



Manuel d'utilisation d'ExtendSim

Version interface française

1Point2

5, rue de la Poste

F-38170 Seyssinet-Pariset

Tél: +33 (0)4 76 27 77 85

Fax: +33 (0)4 7 27 24 67

Email : infos@1point2.com

Site : www.1Point2.com

Copyright © 2018 par Imagine That, Inc. Tous droits réservés.

Copyright © 2018 par UnPointDeux pour la traduction. Tous droits réservés.

Sans autorisation écrite préalable de Imagine That Inc. et de son représentant en France UnPointDeux (1Point2), vous ne pouvez pas copier, diffuser ou traduire, à des fins autres que votre utilisation personnelle, tout ou partie du présent document sous quelque forme ou par quelque moyen que ce soit, tant informatique que mécanique, notamment en photocopiant, en enregistrant ou en stockant des informations ou encore en utilisant des systèmes d'extraction de données. Le logiciel décrit dans ce manuel est fourni selon des conditions spécifiques de licence et ne doit être copié et utilisé que selon les termes des conditions d'utilisation.

Licence, droit de propriété du logiciel, marque déposée et autres informations

Le logiciel décrit dans le présent manuel est fourni selon une licence et un contrat de garantie spécifiques. Ce logiciel peut être utilisé ou copié uniquement dans le respect des conditions dudit contrat. Veuillez noter ce qui suit :

Les blocs et éléments ExtendSim (notamment les icônes, boîtes de dialogue et codes de blocs) sont la propriété de Imagine That Inc. Les blocs et éléments ExtendSim contiennent des avis exclusifs et/ou de marque déposée. Si vous créez des blocs et que vous utilisez tout ou partie des blocs des bibliothèques ExtendSim dans vos propres blocs, ou que vous insérez lesdits blocs ExtendSim (ou tout code provenant de ces blocs) dans vos propres bibliothèques, alors votre droit à vendre, donner ou distribuer de toute autre manière vos blocs et bibliothèques est limité. Dans ce cas, vous êtes autorisé à vendre, donner ou distribuer ledit bloc ou ladite bibliothèque uniquement si le destinataire possède une licence valide pour le produit ExtendSim duquel proviennent vos blocs ou votre code de blocs. Pour plus d'informations, contactez Imagine That Inc.

Imagine That!, le logo Imagine That, ExtendSim, Extend et ModL sont soit des marques de commerce déposées, soit des marques déposées de Imagine That, Inc. aux États-Unis et/ou dans d'autres pays. Mac OS est une marque de commerce déposée de Apple Computer, Inc. Microsoft est une marque de commerce déposée et Windows une marque déposée de Microsoft Corporation. GarageGames, Inc. est propriétaire des droits de Torque Game Engine (TGE) et Geer Mountain Software est propriétaire des droits de Stat::Fit®. Tous les autres noms de produits cités dans ce manuel sont les marques déposées de leurs détenteurs respectifs. TGE et Stat::Fit sont fournis en licence à Imagine That, Inc. pour une distribution avec ExtendSim. Tous les autres produits et portions de produits ExtendSim sont la propriété de Imagine That, Inc. Tout droit, titre et intérêt, notamment tous les droits de propriété relatifs au Logiciel demeurent en tout temps la propriété de Imagine That, Inc.

Remerciements

Extend a été créé en 1987 par Bob Diamond ; il a été rebaptisé ExtendSim en 2007.

Principaux contributeurs pour ExtendSim 10 : Steve Lamperti et Bob Diamond

Ingénieurs simulation : Anthony Nastasi, Cecile Pieper et Peter Tag

Graphisme, documentation et édition : Kathi Hansen, Carla Sackett et Pat Diamond

Imagine That Inc • 6830 Via Del Oro, Suite 230 • San Jose, CA 95119 USA
408.365.0305 • fax 408.629.1251 • info@extendsim.com
www.ExtendSim.com

Table des matières

TABLE DES MATIÈRES.....	2
INTRODUCTION.....	9
Modélisation et simulation.....	9
L'importance de la simulation.....	9
ExtendSim et les tableurs.....	9
Ce que peut faire ExtendSim.....	10
L'utilisation d'ExtendSim.....	10
Les utilisateurs d'ExtendSim.....	11
La documentation.....	11
Remarques sur la version française.....	11
Compléments d'information.....	11
Support et mises à jour.....	11
CONSTRUIRE ET EXÉCUTER UN MODÈLE	13
Les fichiers ExtendSim.....	13
Lancer ExtendSim et créer un modèle.....	13
Généralités sur un modèle.....	17
Lancer une simulation.....	17
Modifier des hypothèses.....	19
LES BIBLIOTHÈQUES.....	21
Les bibliothèques fournies avec ExtendSim.....	21
Ouvrir et fermer des bibliothèques.....	22
La recherche des bibliothèques.....	22
Les fenêtres des bibliothèques.....	23
Placer les blocs dans un modèle.....	24
Types de connecteurs.....	24
Faire évoluer des bibliothèques.....	25
Compiler des bibliothèques.....	26
Protéger des bibliothèques.....	26
Diffuser des bibliothèques.....	26
L'INTERFACE DES MODÈLES	27
Boutons et outils.....	27
Cloner des éléments de dialogue sur le modèle.....	30
Les journaux de bord.....	31
Naviguer dans le modèle.....	31
Des blocs pour un tableau de bord.....	32
Le bloc Buttons.....	32
Le bloc Popups.....	32
Le bloc Model Interface.....	32
Les blocs Run Model et Pause Sim.....	32
Les blocs de contrôle.....	32
Styles et manipulation des liens.....	33
Liens nommés.....	35
Animation.....	35
La hiérarchie.....	36
Rendre une sélection hiérarchique.....	37
Construire un nouveau bloc hiérarchique.....	38
Enregistrer un bloc hiérarchique.....	39
Options complémentaires sur la hiérarchie.....	40

Modifier des blocs hiérarchiques	41
Différentes visualisations d'un modèle	42
Masquer ou voir les connecteurs et les liens	42
Interactions sur un modèle	42
LANCER DES SIMULATIONS.....	44
Le dialogue Paramètres.....	44
Onglet Paramètres	44
Onglet Continu	45
Onglet Aléatoire.....	47
Nombres aléatoires	48
Lancer la simulation	48
Choix de l'unité de temps.....	48
La durée et le nombre de simulations (réplications).....	49
Terminatif ou non ?.....	49
Déterminer la durée et le nombre des simulations.....	49
Accélérer l'exécution.....	49
BIBLIOTHÈQUES DE BASE : MODÈLES CONTINUS ET DISCRETS.....	51
La bibliothèque Value.....	51
Utiliser la bibliothèque Value.....	51
Les blocs de la bibliothèque Value par fonction	51
La bibliothèque Item.....	53
Types de systèmes concernés	53
Utiliser la simulation.....	54
Utiliser la bibliothèque Item	54
Les blocs de la bibliothèque Item par fonction	56
Bibliothèques Utilities et Report.....	58
DONNÉES ET ÉCHANGES DE DONNÉES	60
Les bases de données ExtendSim.....	60
Comment créer une base de données ExtendSim.....	61
Les blocs Read et Write pour accéder à une base de données	65
Gérer les bases de données	66
Importer ou exporter une base de données	66
Dialogues et menus déroulants de la BDD	68
Add-In Excel pour les bases de données ExtendSim.....	70
Echange d'informations avec ExtendSim	72
Copier, Couper et Coller par le Presse-Papiers.....	72
Les images.....	73
Les fichiers texte	73
Partage de données via les matrices globales	74
Les liens dynamiques ("hot links")	74
Les ports série	75
DLL : accéder à du code écrit dans d'autres langages	75
Travail en réseau en utilisant les "Mailslots" (Windows uniquement)	76
Communications entre applications : "inter-process communication"	77
CONSEILS DE MODÉLISATION.....	78
Conseils généraux de modélisation.....	78
Conseils pour la bibliothèque Value.....	78
Bloc Holding Tank : intégrer ou sommer ?.....	78
Utilisation de NoValue pour retarder des calculs.....	79
Animation	79
Copier des portions d'un modèle.....	80
Conseils pour la bibliothèque Item	80
Progression des entités durant la simulation.....	80
Types de blocs.....	81
Blocs continus dans les modèles discrets.....	81

Personnaliser les symboles d'animation.....	82
Traitement préparatoire.....	83
Restreindre des entités dans un système.....	83
Coûts	84
Irrégularités de timing.....	84
Changer d'échelle pour suivre des entités.....	84
Connecteurs et liens.....	84
Mélanger des blocs continus et discrets.....	84
Liens sur des entrées multiples.....	84
LES OUTILS D'ANALYSE	86
Blocs de statistiques.....	86
Analyse de sensibilité.....	87
L'optimisateur	91
Paramètres de l'optimisateur	93
Paramètres d'optimisation.....	93
Analyse de scénarios.....	94
Fonctionnement du Scenario Manager.....	95
Didacticiel s'appuyant sur les paramètres de dialogue.....	95
Didacticiel s'appuyant sur des variables des bases de données.....	100
Comparaison des méthodes d'analyse.....	106
Rapport d'un modèle.....	107
Les graphiques.....	108
Types de graphiques.....	108
Structure des graphiques	108
Onglet Display	109
Onglet Data Collection.....	109
Onglet Graphique	112
Editeur de courbes	113
Propriétés du graphique	113
Onglet Données.....	114
EQUATIONS... SI AFFINITÉS.....	115
Composants des blocs à équations.....	115
Variables d'entrée	115
Variables de sortie	116
Editeur d'équations	118
Fichiers include pour les équations	118
Bloc Queue Equation.....	119
Blocs Query Equation et Query Equation (I)	121
Différences entre les blocs Query Equation.....	124
CONSEILS DE MISE AU POINT ET DÉBOGAGE	127
Animation.....	127
Les blocs de mise au point.....	127
La trace d'un modèle	127
Etapas de la trace.....	128
Profil d'un modèle.....	128
Déboguer des équations.....	129
Commencer le didacticiel.....	129
Définir des points d'arrêt	130
Exécuter le code	130
Définir un point d'arrêt conditionnel.....	132
Autres fonctions de débogage.....	133
La commande Ordre de la simulation.....	133
La commande Rechercher	134
Progression de la simulation.....	134
ENTITÉS: LES GÉNÉRER ET LES QUALIFIER.....	135

Générer des entités	136
Propriétés	139
Attributs	139
Types d'attribut	140
Utiliser des attributs	140
Lire les valeurs d'attribut et repérer les changements	143
Modifier les valeurs d'attribut	143
Priorités	144
Quantités	145
Traitement des entités ayant une quantité différente de 1	145
Autres propriétés	147
FILES D'ATTENTE.....	148
Algorithmes de file d'attente	148
Attente d'un service.....	149
Files selon priorité	149
Blocage	149
Dérivation.....	149
Abandon.....	150
Changement de file.....	150
Classer des entités avec le bloc Queue Equation	151
Queue Matching.....	154
Autres techniques.....	155
ACTIVITÉS ET OPÉRATIONS	158
Déterminer le temps de traitement.....	159
Temps de traitement programmé	159
Temps de traitement aléatoire	159
Temps de traitement personnalisé.....	160
Temps de traitement cumulé : temps partagé.....	160
Traitement en série.....	161
Opérations en parallèle.....	162
Activer un système.....	163
Pour un certain nombre d'entités.....	163
Pour un certain temps.....	164
Opérations interrompues	164
Arrêt programmé	165
Arrêt aléatoire.....	165
Arrêt explicite.....	166
Blocs de transport	167
ORIENTATION DU FLUX	168
Blocs fréquemment utilisés	168
Entités provenant de plusieurs sources	168
Convergence de flux discrets avec les blocs Throw et Catch.....	171
Décisions d'orientation : divergence de flux.....	172
Orientation simple.....	174
Rebut	175
Orientation séquentielle.....	176
Orientation explicite.....	176
Décisions d'orientation basées sur des attributs.....	177
Orientation conditionnelle	178
RESSOURCES.....	182
Trois manières de modéliser des ressources	182
Systèmes fermés et ouverts.....	183
Planification des ressources.....	184
Utiliser le bloc Shift.....	184
LOTTAGE/DÉLOTTAGE.....	187

La mise en lot.....	187
Le délottage.....	188
Conserver l'individualité des entités	188
Réunir des entités pour un traitement	189
Mise en lot simple.....	189
Mise en lot d'entités identiques	189
Association et délottage.....	190
CALCUL DE COÛTS	191
Calcul de coûts.....	191
Combiner des ressources et des accumulateurs de coût	192
Coûts et mise en lot	192
Représentation et visualisation des coûts	192
Statistiques sur les coûts.....	193
MESURES ET STATISTIQUES.....	194
Mesurer, vérifier et documenter les résultats d'une simulation	194
Eléments de statistiques	194
Valeurs constantes ou variables aléatoires ?.....	194
Base aléatoire.....	194
Choix d'une distribution.....	195
Accumuler des données avec des attributs	195
Chronométrer les entités dans une portion du modèle	195
MODÉLISATION AVEC LA BIBLIOTHÈQUE RATE.....	197
Quelques principes à connaître.....	197
Modèles flux.....	198
Comparaison entre modélisation discrète et modélisation continue	198
Composantes des modèles flux.....	198
Terminologie et architecture.....	199
COMMENT CONSTRUIRE UN MODÈLE FLUX.....	201
Créer les liens	202
Ajouter une contrainte dynamique	203
Ajouter une maintenance	204
Refroidissement.....	206
Packaging des pots	206
Qualifier les flux	208
Distinguer les flux	209
Aller plus loin.....	210
SOURCES, RÉSERVOIRS ET UNITÉS	212
Capacité.....	212
Zone de chargement	213
Cuve de stockage	213
Capacité du bloc Convey Flow	213
Définir un contenu initial.....	214
Vide et non vide.....	214
Initialisation du Tank.....	214
Initialisation de l'Interchange	214
Initialisation du Convey Flow	214
Indicateurs	215
Définir les indicateurs.....	215
Connaître les niveaux.....	215
Unités et groupes unitaires	216
Unités de flux.....	216
Unités de bloc	216
Unités de temps	216
Unités de longueur	216

Groupes unitaires	216
Où se déclare l'unité de flux ?.....	217
Gérer les unités de flux dans le bloc Executive.....	217
Définir des unités de bloc	217
Changer le groupe unitaire.....	217
DÉBITS, CONTRAINTES ET MOUVEMENTS.....	219
limiter le mouvement du flux par des débits et des contraintes.....	219
Débits, sections de flux, et zone PL.....	219
Types de débits	219
Sections de flux.....	220
Zone PL	221
Règles de flux	221
Contraintes	221
Contraintes critiques	221
Contraintes relationnelles.....	222
Comparaison de contraintes	222
Définir une contrainte critique.....	222
Tank et Interchange	223
Convey Flow	224
Merge et Diverge	224
Répondre au besoin de contrainte critique.....	224
Valve ou Convey Flow	225
Merge ou Diverge.....	225
Sections de flux.....	227
ORIENTATION DU FLUX.....	229
Emploi des blocs Merge, Diverge, Throw Flow et Catch Flow	229
Mélanger ou séparer les flux.....	229
Mode Select	230
Mode Batch/Unbatch.....	230
Mode Proportional.....	231
Mode Priority	231
Mode Distributional.....	232
Mode Sensing.....	233
Mode Neutral	234
Fonctionnalités des blocs Merge et Diverge.....	234
Paramètres de biais – requêtes en concurrence sur le flux.....	235
Envoi/réception intégré.....	235
Modifier dynamiquement les règles de décision	235
Envoi et réception de flux à distance.....	236
Options de filtre pour faciliter les liens envoi/réception.....	237
RETARDER LE FLUX.....	239
Maintenir le flux à une certaine vitesse ou le stopper.....	239
Définir un objectif de quantité	240
Définir un objectif de durée.....	242
Définir une hystérésis	244
Retarder le flux par le bloc Shift	244
Bloc Convey Flow.....	245
FLUX ET ENTITÉS DISCRÈTES	248
Des entités qui contrôlent le flux.....	248
Le flux qui contrôle des entités.....	248
Contrôles croisés	249
Utiliser bloc Interchange pour mélanger flux et entités.....	250
Les modes de l'Interchange.....	251
Tank only exists... ..	251
Tank is separate.....	252

BIBLIOTHÈQUE RATE : NOTIONS DIVERSES	253
Précision	253
Biaisier le flux.....	253
Options avancées du bloc Executive.....	256
Gérer les unités de flux : Manage flow units.....	257
Options avancées.....	257
Connecteurs courants des blocs Rate	259
Animation.....	260
Animation de l'objectif et de l'hystérésis	263
MODÈLES DED : CONCEPTS AVANCÉS	265
La technologie PL.....	265
La zone PL	265
Types d'informations fournies au bloc Executive.....	267
Fourniture amont et demande aval.....	269
Les messages dans les modèles Rate	272
MENUS	274
Menu Fichier.....	274
Menu Edition.....	276
Menu Bibliothèque.....	282
Menu Modèle.....	283
Menu Base de données.....	287
Menu Développeur	289
Menu Simulation	291
Menu Fenêtres	293
Menu Outils	293
Menu Aide.....	293
Nous contacter.....	295
INDEX.....	296

Introduction

ExtendSim est un logiciel professionnel de simulation de flux. Avec ExtendSim, vous étudiez le comportement dynamique de systèmes dans des domaines très divers. ExtendSim permet de construire des modèles, d'explorer chaque ensemble de traitement et ses interrelations, puis de changer d'hypothèses pour arriver à des solutions optimales.

Modélisation et simulation

Un modèle est une description logique du comportement d'un système. La simulation implique la conception/reproduction d'un modèle et le test d'expériences se déroulant dans le temps sur ce modèle. Par exemple, le jeu de Monopoly est un modèle d'un système réel. Lorsque vous jouez une partie de Monopoly, vous simulez ce système. La simulation avec ExtendSim signifie qu'au lieu d'agir sur un système réel, vous créez un modèle qui lui correspond par certains aspects.

Vous pouvez utiliser un modèle pour décrire le comportement d'une activité réelle. Le premier avantage à utiliser un modèle est que vous pouvez commencer par une approximation sommaire du traitement, puis affiner peu à peu le modèle, au fur et à mesure de votre meilleure compréhension de son fonctionnement. Cet affinage permet d'arriver très vite à des approximations fiables et cohérentes de systèmes très complexes.

Les modèles permettent de tester des hypothèses à un coût infiniment plus faible que le coût d'un prototype ou d'un test réel. Si vous devez concevoir une usine, vous n'aurez pas besoin de bâtir cette usine pour en étudier le fonctionnement.

L'importance de la simulation

Modélisation et simulation sont une méthode pour vérifier votre compréhension des systèmes qui vous entourent. Vous comprenez pourquoi vous obtenez tels ou tels résultats ici, et vous savez comment provoquer ce type de résultat. Lorsque vous savez prédire le déroulement et les résultats d'une simulation, vous pouvez arrêter la phase de modélisation et travailler dans le monde réel.

Souvent, on simule pour confirmer que l'on connaît des réponses et que l'on n'a pas oublié des détails. Dans la conception de systèmes, des phénomènes tels que l'interdépendance et la rétroaction sont difficiles à prévoir et souvent sous-estimés. La simulation permet d'identifier des zones à problème et de faire des modifications avant l'implémentation. Les simulations avec ExtendSim sont aussi d'excellents moyens de communication lorsqu'il s'agit de démontrer aux autres la fiabilité et l'intégrité d'un modèle.

Un programme comme ExtendSim a de nombreux emplois. Le moindre n'est pas d'être le support à des discussions, des échanges d'idées, des modifications constructives, autour d'une table, d'un écran et d'une souris.

Pour vous présenter la démarche de modélisation et simulation, ses enjeux, ses apports et sa mise en œuvre, nous osons vous conseiller un ouvrage : ***Gestion de flux en entreprise : modélisation et simulation***.

Ce livre présente l'intérêt de l'approche par modélisation et simulation de flux pour appréhender précisément et rapidement les effets directs et indirects d'une décision et éclairer les orientations des entreprises. L'ouvrage comporte trois parties :

- Principales problématiques industrielles, enjeux et manières de les traiter par simulation
- Présentation technique de la simulation, ses concepts, ses outils, son fonctionnement
- Guide de conduite d'un projet de modélisation et de simulation.

L'ouvrage propose une approche applicable aussi bien en milieu industriel que tertiaire, et s'adresse aux managers de tous domaines. Auteurs : J-F. Claver, J. Gélinier, D. Pitt. Editeur : Hermès, Paris (1997). Disponible auprès de 1Point2, 28€ port inclus.

ExtendSim et les tableurs

ExtendSim est à la simulation ce que les tableurs sont au traitement des nombres. Vous créez un

diagramme composé de blocs pour décrire un système complexe où chaque bloc représente une partie de traitement. Dans un tableur, vous organisez les chiffres dans un tableau à deux dimensions ; dans ExtendSim, vous composez votre système dans un environnement de dessin à deux dimensions. Alors que le tableur représente une vue instantanée sur un système, ExtendSim est l'équivalent d'un film qui se déroule. La technique itérative d'ExtendSim permet de créer des modèles s'appliquant au monde réel et qui sont trop complexes pour être décrits aisément par un tableur.

Vous élaborez vite vos modèles parce qu'ExtendSim est livré avec les blocs nécessaires pour la plupart des simulations. Ces blocs sont comme des macro-commandes.

Comme vous travaillez à partir de blocs, vous pouvez utiliser plusieurs blocs simples pour représenter un traitement complexe. ExtendSim fournit également un éditeur d'équations (semblable à la barre de formules d'un tableur) où vous pouvez combiner les fonctions de plusieurs blocs en un seul. Vous pouvez aussi créer vos propres bibliothèques de blocs personnalisés pour des applications spécifiques.

Ce que peut faire ExtendSim

ExtendSim propose à la fois puissance et facilité pour répondre au plus grand nombre de cas :

- de nombreux blocs à disposition pour travailler rapidement,
- de l'animation dans les modèles pour mieux faire passer le message,
- une interface graphique que vous personnalisez,
- une hiérarchie dans les modèles pour mieux appréhender les systèmes complexes,
- dialogues, base de données et journal de bord pour modifier rapidement les valeurs du modèle et tester vos hypothèses,
- la modification de paramètres en cours de simulation,
- une excellente communication avec d'autres programmes ou plateformes par fonctions intégrées, liens dynamiques, Copier/Coller, Importer/Exporter, par fichiers texte et autres,
- des techniques de Monte Carlo, un mode batch et des outils d'analyse pour optimiser les systèmes.
- animation 2D et 3D pour des présentations professionnelles.

L'environnement complet d'ExtendSim combine les fonctionnalités les plus audacieuses des outils de simulation existants :

Compatibilité multi-environnement : modèles et bibliothèques peuvent être utilisés indifféremment sous Windows 64-bits (Vista, 7, 8.1, 10), Server 2016, Server 2012, ou Server 2008 R2.

Simulation non dédiée : ExtendSim est un environnement d'applications pluridisciplinaires dans lequel vous modélisez des systèmes discrets, continus ou hybrides, linéaire ou non linéaires.

Langage C intégré : vous adaptez les blocs ExtendSim ou créez de nouveaux blocs pour vos applications spécialisées ; vous concevez une animation personnalisée pour votre modèle.

Support d'autres langages : utilisez les APIs intégrées afin d'accéder à du code créé en Delphi, C++ Builder, Visual Basic, Visual C++, etc.

Structuration en bibliothèques : les blocs créés sont organisés en bibliothèques et facilement utilisables dans d'autres modèles.

Plus de 1000 fonctions : des fonctions à accès direct pour faire du calcul intégral, des statistiques, de la gestion de files d'attente, de l'animation, des mathématiques IEEE, du calcul matriciel, des sons, des transformées de Fourier, de la mise au point, de la manipulation de bits ou de chaînes de caractères, etc. Vous pouvez de plus définir vos propres fonctions.

Envoi de messages : les blocs peuvent envoyer des messages à d'autres blocs interactivement pour lancer des sous-routines.

Echange de données évoluées : les blocs transmettent des valeurs simples, des matrices, ou des structures complexes de matrices de matrices.

Gros modèles : la taille d'un modèle n'est limitée que par la taille de votre système.

L'utilisation d'ExtendSim

Un modèle ExtendSim est un document contenant des copies de blocs avec des liens entre eux. Chaque bloc comporte des procédures et des données. Une fois un modèle créé, il peut être modifié

par ajout de blocs, modification des liens ou des données des blocs.

Les utilisateurs d'ExtendSim

ExtendSim s'utilise à plusieurs niveaux :

- Les nouveaux-venus à la simulation peuvent lancer des simulations sur des modèles qui leur sont préparés, en entrant des données dans des dialogues ou dans des fichiers texte. Leur objectif est de tester des hypothèses, de créer des variantes et d'en analyser les résultats.
- Très vite les utilisateurs seront aptes à assembler leurs propres modèles à partir des blocs fournis avec ExtendSim. Cela implique une bonne connaissance des fondements et rouages des systèmes étudiés, ainsi que la connaissance des blocs livrés dans les bibliothèques.
- Certains utilisateurs auront le besoin ou l'envie de créer leurs propres blocs (ou de modifier les blocs existants) à l'aide du langage ModL. Ces utilisateurs doivent pouvoir écrire un programme informatique.

ExtendSim permet d'organiser les modèles en leur donnant une structure hiérarchique. Vous pouvez ainsi récupérer un tronc commun de blocs ou de parties de modèles pour chaque nouvelle simulation.

La documentation

Ce manuel est le principal pour comprendre comment utiliser ExtendSim, mais il est complété par d'autres. D'une part par des *Guides de prise en main* thématiques, au vous pourrez lire au préalable, dont au moins un correspond à votre type de modélisation :

- *Guide de prise en main ExtendSim continu* :
- *Guide de prise en main ExtendSim discret* :
- *Guide de prise en main ExtendSim DED* (débits à événements discrets) :

D'autre part par un manuel technique pour les programmeurs, *Technical Reference* (en anglais), ainsi que des manuels thématiques (Database, Reliability, ARM).

Remarques sur la version française

L'interface du logiciel, les principaux messages d'erreur ainsi que les manuels essentiels vous sont proposés en français. Les blocs de base restent finalement en anglais, tout comme les symboles d'animation (symboles 2D et objets 3D).

Après avoir traduit les principales bibliothèques, la traduction gagnait en complexité avec la complexité même des blocs, et devenait source possible d'erreur, et source certaine de retards dans la diffusion des versions.

Dans ExtendSim, l'interface utilisateur d'un modèle est en général celle que le modélisateur construit pour le public final du modèle, qui a rarement la connaissance pour ouvrir le dialogue d'un bloc et le paramétrer. Tableaux Excel, blocs hiérarchiques, journaux de bord, animation du modèle et base de données sont les éléments principaux d'une interface qui est de toute façon personnalisée. Les blocs en anglais ne devraient donc pénaliser que quelques utilisateurs, sans gêner la majorité.

Compléments d'information

Des **infobulles** descriptives sont affichées par défaut lorsque le curseur reste quelques secondes sur une icône de bloc, un outil dans la frise ou un élément de dialogue dans la structure d'un bloc. La présence de ces infobulles est contrôlée par le dialogue Options, menu Edition.

L'**aide** complète sur le fonctionnement de chaque bloc (éléments de dialogue, connecteurs, options et animation) est accessible en cliquant sur le bouton Aide dans le dialogue du bloc.

L'aide en ligne par le menu Aide ou la touche F1 donne accès aux manuels en format électronique

Vous trouverez sur internet d'autres sources d'information, en anglais toutefois : des questions-réponses (www.ExtendSim.com/support/advice/faq.html), un forum des utilisateurs accessible via www.ExtendSim.com/support/advice/forum.html.

Vous pouvez contacter 1Point2 et son support technique par l'adresse support@1Point2.com : gratuitement dans les 2 mois suivant votre achat ou selon le contrat de maintenance dont vous disposez.

Support et mises à jour

Un plan de maintenance et de support ExtendSim (MSP) est désormais requis pour tous les produits sauf la version Essai, la version Analyse et la version Player. Le coût du MSP est compris dans le prix

à l'achat de la licence et dure une année. Tant que le MSP est valide, vous pouvez :

- Obtenir une aide sur des questions d'installation, d'utilisation de base, et sur des éventuels bugs.
- Télécharger gratuitement les mises à jour ExtendSim mineures.
- Télécharger gratuitement les mises à jour ExtendSim majeures.
- Déplacer votre licence ExtendSim d'un appareil à un autre (dans le cadre des conditions d'utilisation).
- Obtenir une clé d'activation de remplacement.
- Expérimenter et évaluer de nouvelles fonctionnalités.

Pour continuer de bénéficier de ces services, vous devez renouveler le MSP à la date anniversaire de l'activation de la licence.

Si vous cessez de renouveler le MSP :

- Vous ne bénéficiez plus des services (support technique, mises à jour, remplacement de clé d'activation etc.) listés ci-dessus.
- Vous devrez acheter les mises à jour à prix normal.
- Si vous remettez en place le MSP, ce sera à la date anniversaire d'origine.

Versions d'exécution d'ExtendSim

Si le modèle que vous développez doit être diffusé d'une manière ou d'une autre à des personnes ou entreprises qui seront de simples utilisateurs du modèle, des licences spéciales une diffusion à faible coût de vos modèles.

Exécution active via la version Analyse, où l'utilisateur peut :

- Lancer des simulations, obtenir des résultats et les imprimer.
- Modifier des valeurs dans les dialogues ou la base de données et voir l'animation (si le modèle le prévoit).
- Enregistrer des modifications.

En revanche, l'utilisateur ne peut pas construire des modèles ou des blocs, ni modifier l'agencement du modèle fourni.

Exécution passive via la version Player, où l'utilisateur peut juste lancer la simulation du modèle qu'il ouvre, sans modification ni enregistrement:

Contactez 1Point2 pour davantage de renseignements.

Construire et exécuter un modèle

Cette section présente les grandes fonctions à connaître pour construire et exécuter des modèles, travailler avec des blocs et modifier les paramètres du modèle. À la fin, vous saurez l'essentiel pour exécuter et obtenir des résultats d'une simulation.

Si vous avez lu le manuel *Prise en main* qui s'applique à vos modèles, vous trouverez ici des redites, et vous pouvez survoler ce chapitre.

Bien souvent en simulant on cherche à trouver un système optimal qui puisse traiter le plus grand nombre d'éléments avec le minimum de ressources dans le temps le plus court. Schématiquement, dans un modèle de ce type, on a un poste de travail et une file d'attente.

- nous rencontrons des files d'attente (ou queues) bien souvent dans la vie courante. Les entités en attente peuvent être des personnes, des pièces, des données, ou divers éléments isolables.
- le poste de travail est ce par quoi l'élément doit passer. Cela peut être une caisse, une machine, un ordinateur, ou toute autre ressource qui prend un certain temps pour traiter un élément de la file d'attente.

Nous examinerons un modèle appelé "Car Wash" (lavage de voitures), où l'on retrouve la file d'attente et le poste de travail sous forme de clients qui attendent pour être prendre de l'essence ou faire laver leur voiture. N.B.: il s'agit d'un modèle discret, reportez-vous au manuel *Prise en main ExtendSim continu* si vous utilisez ExtendSim CP.

Les fichiers ExtendSim

Il y a trois types de fichiers ExtendSim : les **modèles**, les **fichiers texte** et les **bibliothèques**. La majeure partie de votre travail se fait sur les modèles. Un modèle est une représentation de blocs et de liens. Une bibliothèque est un stock de blocs. Lorsque vous voulez ajouter une fonction à votre modèle, vous allez chercher un bloc dans une bibliothèque. Les fichiers texte contiennent des données lisibles par de nombreuses applications. Ce sont typiquement des résultats de simulation ou des données en entrée, issues de tableurs, d'outils statistiques ou d'ExtendSim même.

Sous l'Explorateur, l'extension "mox" caractérise les modèles, "lbr" les bibliothèques et "txt" les fichiers texte.

On ouvre et ferme les modèles par le **menu Fichier**. Les bibliothèques s'ouvrent et se ferment par le **menu Bibliothèque**. Vous consultez la liste des bibliothèques ouvertes au bas du menu Bibliothèque.

Les fichiers texte, produits ou non par ExtendSim, sont accessibles depuis le menu Fichier. Ces fichiers sont aussi lisibles par un traitement de texte et par de nombreuses applications.

Lancer ExtendSim et créer un modèle

Vous lancez ExtendSim par un double-clic sur son icône, qui a l'aspect suivant :



Pour construire un nouveau modèle, choisissez **Nouveau** dans le menu **Fichier**. ExtendSim ouvre une fenêtre pour un nouveau modèle.

☞ Si vous ne souhaitez pas construire ce modèle mais seulement le découvrir, choisissez **Ouvrir modèle** dans le menu **Fichier**, trouvez dans `Exemples\Tutorials\Discrete Event` le modèle **Step 1 DE Tutorial** et ouvrez-le. Le modèle apparaît à l'écran.

Ouvrir les bibliothèques

Pour placer un bloc dans un modèle, la bibliothèque où se trouve ce bloc doit être ouverte. Les bibliothèques de base sont normalement ouvertes par défaut, mais au besoin :

- Choisissez **Ouvrir** dans le menu Bibliothèque.
- Dans le dialogue, localisez et sélectionnez la bibliothèque Chart (fichier Chart.lbr) et la bibliothèque Item dans le même répertoire Libraries.

Les blocs

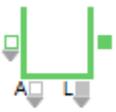
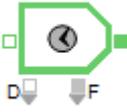
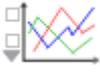
Un bloc représente une action ou un traitement. Chaque bloc est représenté par une icône unique. Une information arrive au bloc et est traitée par le programme qui est dans le bloc. Le bloc transmet

ensuite l'information au bloc suivant dans le modèle.

Il y a six blocs dans le modèle de lavage de voitures. Vous pouvez avoir plusieurs copies du même bloc dans un modèle.

Un bloc est utilisé pour représenter une fonction dans le modèle. Certains blocs représentent seulement des sources d'informations passées aux autres blocs. D'autres modifient l'information en la passant. Certains blocs sont appelés hiérarchiques et représentent des groupes de blocs.

Voici les blocs que nous allons utiliser, avec leur rôle dans le modèle.

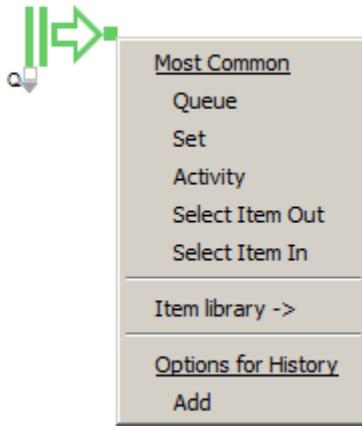
Bloc	Description
	Le bloc Executive de la bibliothèque Item est un bloc spécial qui doit être présent dans toutes les simulations discrètes. La fonction de ce bloc est décrite plus loin. Pour le moment, considérons que ce bloc doit être présent et figurer à la gauche de tous les autres blocs.
	Ce bloc est un Create issu de la bibliothèque Item. Le bloc Create permet de générer des entités ; dans notre cas une "entité" est une voiture qui arrive. Le bloc sera paramétré de sorte qu'un nouveau client arrive à peu près toutes les 4 minutes et moins, selon une distribution exponentielle.
	Les voitures font la queue (ou sont en file d'attente). Si un poste de lavage est libre, la voiture quitte automatiquement la queue et va vers le poste de lavage. Si aucun poste n'est libre, le client attend dans la file qu'un se libère. Le bloc indique la longueur de la queue ainsi que le temps d'attente. Il s'agit du bloc Queue de la bibliothèque Item, avec une logique FIFO (FIFO signifie "first in first out" : le premier entré est le premier sorti).
	La voiture va vers le poste de lavage dès qu'il est disponible. Le postes est représenté ici par le bloc Activity de la bibliothèque Item, où nous indiquerons une <i>capacité</i> 1 pour représenter 1 poste disponible. Le client qui entre dans le bloc sera alors retardé pour la durée attribuée au traitement effectué, soit 6 minutes.
	Après le lavage, les clients vont au bloc Exit qui fait sortir la voiture de la simulation. Le connecteur nb émet le nombre total d'entités qui sont passées par le bloc. Ce bloc provient aussi de la bibliothèque Item.
	Le bloc Line Chart , issu de la bibliothèque Chart, affichera les résultats de la simulation. Tracer des courbes est une manière courante d'obtenir des résultats d'une simulation. Le graphique est décrit en détail plus loin.

Vous remarquerez que par défaut le bloc Executive est déjà placé dans le modèle.

Ajouter les blocs au modèle

Nous allons placer un bloc Create (arrivée des voitures) suivi d'une file d'attente et d'un bloc Activity. Pour ajouter un bloc :

- Ouvrez la fenêtre de bibliothèque de sorte à rendre visibles les blocs de la bibliothèque Item.
- Utilisez la technique « **pointer-cliquer** » pour placer le bloc Create : sélectionnez le bloc dans la fenêtre de bibliothèque, puis à l'emplacement où vous voulez le placer dans le modèle.
- Pour les autres blocs, utilisez la même méthode, ou bien les « **astutiens** », en cliquant du bouton droit de la souris sur le connecteur de sortie du bloc Create. Une liste de blocs vous est proposée, constituée des blocs les plus logiques à placer après celui-ci (ou de ceux que vous avez l'habitude de placer). Cela a l'avantage de requérir moins de mouvements de la souris, et de relier les blocs tout en les plaçant.



Astutiens : bloc proposés en sortie du Create, blocs les plus courants en premier

Relier des blocs

Les connecteurs sont utilisés comme points de liaison des blocs entre eux. Ils sont reliés par des liens, les lignes que vous voyez entre les blocs.

Il existe plusieurs types de connecteurs dans ExtendSim, mais nous en retiendrons deux pour l'instant.

- Entrée continue
- Sortie continue
- Entrée discrete
- Sortie discrete

De nombreux blocs utilisent les connecteurs d'entrée et de sortie continus, qui servent à transmettre des valeurs (couleur grise). Les blocs discrets utilisent des connecteurs spéciaux, servant à transmettre des entités (couleur verte).

Dans le modèle de lavage de voitures, les blocs utilisent des connecteurs d'entrée et de sortie discrets pour passer des entités, et des connecteurs continus pour passer des valeurs.

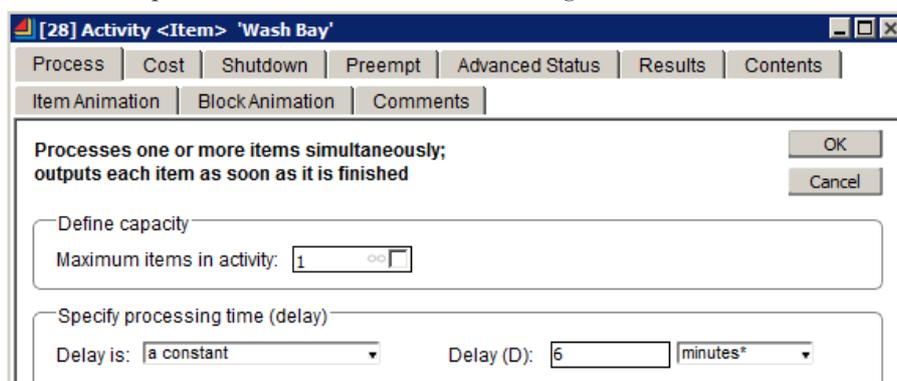
Par exemple pour relier le connecteur de sortie du bloc Queue au connecteur d'entrée du bloc Activity :

- Cliquez sur le connecteur de sortie du bloc Queue, puis cliquez sur le connecteur d'entrée du bloc Activity. Vous savez que vous êtes sur le connecteur parce que la ligne tracée s'épaissit et devient bleue.
- ☞ D'autres manières de relier des blocs existent, elles sont examinées dans [Placer les blocs dans un modèle](#)

Les dialogues

À chaque bloc est associée une boîte de dialogue. Vous utilisez les dialogues pour entrer des valeurs et des paramètres lorsque vous exécutez les simulations. Pour ouvrir un dialogue, faites un double-clic sur l'icône du bloc. Notez que pour les blocs hiérarchiques, qui contiennent d'autres blocs, le double-clic permet de voir les blocs du niveau inférieur.

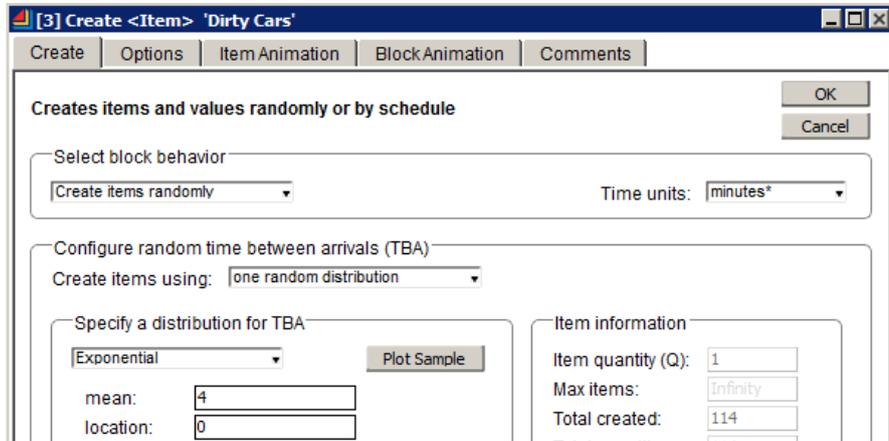
Par exemple, double-cliquez sur l'icône de l'activité. Le dialogue de ce bloc s'ouvre :



dialogue de l'activité Lavage de voitures

Ce dialogue permet de paramétrer le nombre de postes de lavage et leur temps de travail. Spécifiez 3 pour indiquer le nombre de postes, et pour indiquer un lavage standard, indiquez un délai de 6 par exemple, puis cliquez sur OK. Vous pouvez obtenir plus d'informations sur le bloc en cliquant sur le bouton d'aide en bas à gauche du dialogue. À la droite du bouton d'accès à l'aide, une case permet de saisir un label pour le bloc, ce qui est fort utile pour rendre plus clair le modèle. Vous pourriez ainsi donner le label « Lavage » au bloc.

Le **Create** permet d'établir l'intervalle d'arrivée des voitures, ainsi que la méthode avec laquelle elles sont distribuées. Le dialogue affiche :



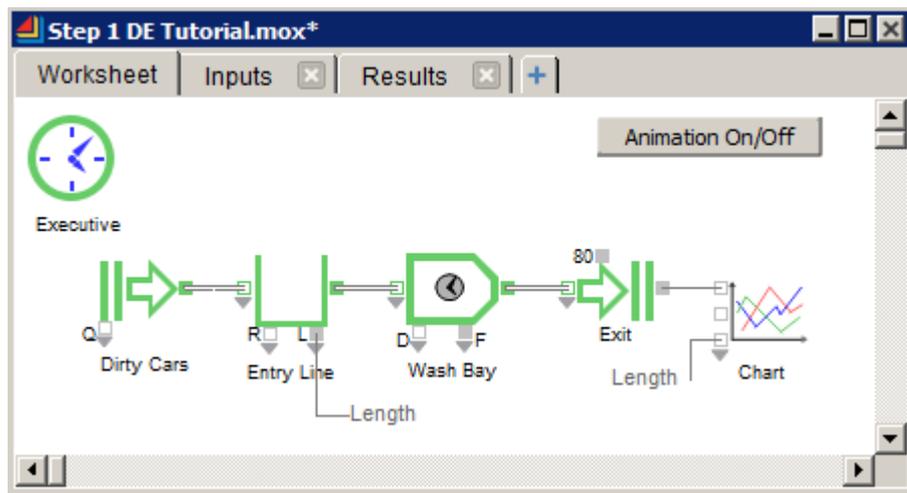
dialogue du bloc Create

Pour ce modèle, l'unité de temps sera la minute, mais nous allons définir une loi d'arrivée dans une unité libre, et l'on indiquera au bloc de générer un nouveau client en moyenne toutes les 4 minutes. Ces paramètres simples suffiront pour que le modèle puisse s'exécuter.

De nombreux dialogues sont composés de plusieurs onglets, organisés thématiquement. Vous pouvez redimensionner un dialogue comme n'importe quelle fenêtre. Si vous lui donnez une petite taille, vous pourrez utiliser les barres de défilement pour accéder aux portions cachées.

Dans les dialogues où vous devez entrer des informations, vous tapez soit des chiffres soit des lettres. Si vous entrez du texte, comme par exemple dans le champ de l'onglet **Comments**, ExtendSim effectue automatiquement le passage à la ligne du texte que vous tapez. Vous pouvez imposer un saut de ligne en appuyant sur Entrée. Si vous entrez des nombres avec des décimales, ExtendSim utilise le format du symbole décimal de votre pays (la virgule en France).

Vous pouvez laisser un dialogue ouvert durant la simulation, bien que cela ralentisse le traitement. Certains dialogues montrent des valeurs du modèle, donc vous pourrez utiliser les dialogues pour voir les valeurs que vous souhaitez durant la simulation.



le modèle "Lavage de voiture" ou Step 1 DE Tutorial

Voici ce à quoi ressemble une fenêtre de modèle typique. Comme avec toute fenêtre, vous pouvez utiliser les barres de défilement pour faire défiler horizontalement et verticalement, et vous pouvez modifier la taille de la fenêtre par la case de grossissement en bas à droite. Vous fermez la fenêtre par

la case de fermeture (ou case du menu Système), et pouvez l'agrandir par la case de zoom (ou les boutons Agrandissement/Réduction) en haut à droite.

Généralités sur un modèle

La simulation elle-même est constituée d'une série de calculs et d'actions qui s'effectuent sur les blocs du modèle, et ceci de manière répétitive. Chaque répétition est appelée événement ; votre modèle peut s'exécuter aussi longtemps que vous le voulez.

Dans le modèle de lavage de voiture, le bloc qui ressemble à une montre sur la gauche du modèle est traité en premier, ensuite globalement de gauche à droite, en suivant les liens entre blocs. Après ce premier événement, le modèle se recalcule. Lorsque vous lancez une simulation, vous indiquez combien de temps (en temps simulé) vous voulez qu'elle s'exécute.

Les bibliothèques

Les blocs sont rangés dans des bibliothèques. Une bibliothèque contient la définition complète de ses blocs (programme, icône, dialogue, etc.). Lorsque vous placez un bloc dans un modèle, le bloc lui-même n'est pas copié dans le modèle, mais une référence aux informations contenues dans la bibliothèque. ExtendSim enregistre avec le modèle les données entrées dans le dialogue du bloc.

Lorsque vous enregistrez un modèle sur disque, ExtendSim enregistre les noms des blocs et l'emplacement des bibliothèques d'où ils proviennent.

Lorsque vous ouvrez un modèle, ExtendSim ouvre automatiquement les bibliothèques nécessaires. Par exemple, si vous avez ouvert le modèle de lavage de voiture plutôt que de le construire, ExtendSim a ouvert deux bibliothèques : "Item" et "Chart". Pour le vérifier, déroulez le menu Bibliothèque et observez le nom des bibliothèques au bas du menu.

Il y a deux manières d'ouvrir une bibliothèque. Comme vous venez de le voir, une bibliothèque peut s'ouvrir parce que vous avez ouvert un modèle qui l'utilise. Vous pouvez aussi ouvrir une bibliothèque par l'option **Ouvrir** du **menu Bibliothèque**. Pour voir le contenu d'une bibliothèque, ouvrez la bibliothèque et sélectionnez-la dans le menu Bibliothèque. Vous voyez apparaître un menu hiérarchique à la droite du nom de la bibliothèque, contenant la liste de tous les blocs dans la bibliothèque.

Exploration du modèle de lavage de voiture

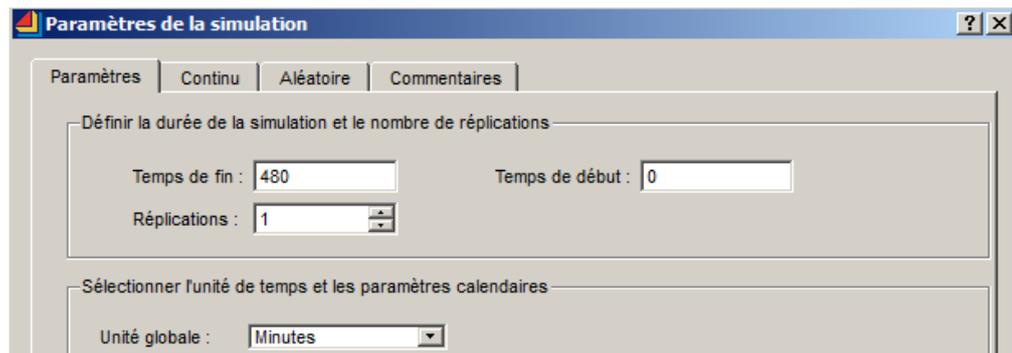
Maintenant que vous avez vu les composants de base d'un modèle, vous êtes prêt à voir comment il fonctionne. Après son exécution, vous verrez combien il est facile de modifier le modèle quand vous modifiez les hypothèses de votre simulation.

Le modèle de lavage de voiture simule des voitures qui arrivent à une station de lavage, font la queue, passent au poste de lavage puis quittent les lieux. L'objet du modèle est de voir le nombre de clients traités durant une heure type, dans des conditions différentes, et combien de personnes font la queue à chaque instant.

Dans de nombreux modèles, votre objectif sera de savoir combien de temps prend une entité (dans notre cas un client en voiture) pour passer d'un bout à l'autre du modèle. Les blocs retardent l'entité ou non. Par exemple dans le modèle de lavage de voiture, la plupart des blocs ne retardent pas le client, mais il y a bien sûr un délai au poste de lavage, puisque l'opération prend un certain temps.

Lancer une simulation

Vous allez maintenant lancer la simulation, après avoir précisé son cadre temporel par le dialogue **Paramètres** de la simulation du menu **Simulation**.



paramètres de la simulation du modèle de lavage de voiture

Cliquez sur l'icône  ou choisissez **Lancer la simulation** dans le menu **Simulation**.

Tandis que le modèle de lavage de voiture s'exécute, ExtendSim affiche les résultats sur le graphique. Les courbes montrent la longueur de la file et le nombre de voiture qui quittent la station.

Regarder des courbes est le premier moyen d'examiner ce qui s'est passé durant la simulation.

Lorsque vous lancez une simulation, ExtendSim peut afficher quelques messages d'initialisation, puis une barre de statut tout au bas de la fenêtre ExtendSim:

Le chiffre à gauche est une estimation du temps restant (exprimé en "heures:minutes:secondes") pour la simulation, ce qui donne une idée de la durée totale. Ensuite est affiché le temps courant (ou CurrentTime) de la simulation (en unités de temps simulées). Le numéro de la réplication (0 dans notre cas) est pertinent si vous lancez plusieurs simulations. Ces valeurs sont déterminées par le dialogue **Paramètres** de la simulation.

Notez que le temps restant affiché n'est qu'une estimation. Dans les simulations continues, l'estimation est souvent exacte. Mais dans les simulations discrètes, cela peut être très imprécis, spécialement en début de simulation où il est impossible de savoir encore quand interviendront des événements et donc combien de temps prendra la simulation.

Plusieurs outils sont disponibles dans la frise d'outils en haut de l'écran, dont ceux concernant le modèle et son exécution :



Pour interrompre une simulation en cours, cliquez sur le bouton rouge **Stop** ou appuyez sur Contrôle-. (la touche Ctrl et le point simultanément). ExtendSim vous demande si vous voulez vraiment arrêter la simulation.



Si vous souhaitez seulement faire une pause dans la simulation, cliquez sur le bouton **Pause** (c'est le bouton Lancer qui a changé d'aspect). Si vous êtes en pause le bouton change encore et devient alors le bouton "Continuer". Vous pouvez aussi utiliser le menu Simulation pour demander une pause ou un arrêt de la simulation.

Le curseur entre la tortue et le lièvre agit sur la vitesse d'affichage de l'animation. L'animation est activée/désactivée à votre guise par le bouton 2D à gauche de la tortue.

Le bouton à la droite du lièvre ouvre une fenêtre de navigation, qui est une autre manière de parcourir les blocs d'un modèle.

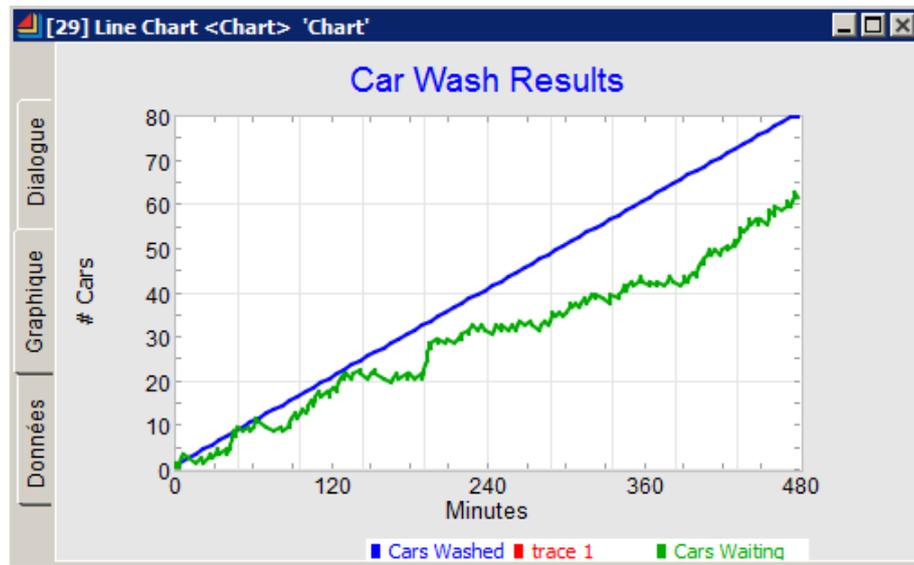
Vous pouvez être amenés, si vous devez exécuter plusieurs modèles, à préférer une exécution multitâche des modèles. Dans la plupart des cas vous travaillez sur un modèle à la fois, et le moteur ExtendSim donnera toute sa puissance à ce modèle. C'est le bonhomme à gauche du bouton Lancer qui détermine dans quel mode vous êtes :



Mode « Un modèle » Mode « Plusieurs modèles »

Les graphiques

Les bibliothèques d'ExtendSim comportent des traceurs de graphiques très souples que vous utiliserez dans les modèles. Les graphiques affichent à la fois une représentation graphique des nombres observés et un tableau de toutes ces valeurs numériques. Le graphique du modèle de lavage de voiture est le bloc Line Chart de la bibliothèque Chart, très employé pour suivre l'évolution d'une valeur.



Lorsque vous lancez une simulation, ExtendSim n'affiche pas le graphique à l'écran car cela ralentit l'exécution. Double-cliquez sur le bloc pour voir les courbes : pour notre modèle il y a deux courbes. La ligne bleue représente le nombre de clients qui ont terminé depuis le début de la simulation. La courbe verte représente la longueur de la file d'attente aux guichets. Vous pouvez observer les valeurs en déplaçant le curseur sur les courbes : les valeurs correspondantes s'affichent en haut à droite du graphique.

L'onglet *Données* affiche les données individuelles ayant produit les courbes. Vous pouvez faire défiler cette table pour voir toutes les valeurs.

L'onglet *Dialogue* permet de modifier des paramètres de calcul et d'affichage. Les modifications que l'on peut apporter aux tracés sont examinées [plus loin](#).

Modifier des hypothèses

Le reste de cette rubrique examine les méthodes pour modifier les hypothèses du modèle de lavage de voitures. Vous modifiez des hypothèses en modifiant des valeurs de paramètres dans des dialogues ou en ajoutant ou supprimant des blocs.

Modifier des dialogues

Les dialogues contiennent beaucoup des paramètres en entrée de vos simulations. Pour ouvrir un dialogue, double-cliquez sur l'icône du dialogue. Certains dialogues offrent peu d'intérêt. Par exemple, il n'y a rien d'important dans le dialogue du bloc Exit.

Cliquez dans le champ que vous voulez modifier. La simulation fait une pause. Lorsque vous voulez continuer, cliquez sur le bouton de continuation dans la barre de statut.

L'**Activité** permet de spécifier le temps pris (6 minutes) pour laver une voiture. Vous pouvez vérifier que l'altération du nombre de postes de lavage peut modifier radicalement la longueur de la queue.

[28] Activity <Item> 'Wash Bay'

Process | Cost | Shutdown | Preempt | Advanced Status | Results | Contents

Item Animation | Block Animation | Comments

Processes one or more items simultaneously;
outputs each item as soon as it is finished

OK
Cancel

Activity statistics

	Current	Average	Maximum
Length:	1	1	1
Wait:	6	6	6
Preemptions:	0		
Shutdowns:	0		
Off-Shifted items:			
Blocked items:	0		

Utilization

Utilization: 1

Items are in a utilized state when: Processing Blocked Off shift Down

Activity's capacity is considered to be in a "utilizable" state when: Off shift Down

Activity state statistics (as a percentage of time and capacity)

Busy: 1 Idle: 2e-08 Blocked: 0 Shutdown: 0 Offshift: 0

Overall statistics

Arrivals: 81 Departures: 80 Total cost:

Aide | Wash Bay | Left to right

dialogue Activity (onglet Results)

Tous ces champs (Longueur, Attente, Arrivées, Départs, Coût total, Utilisation etc.) figurent encadrés de gris. Cela signifie qu'ils affichent des valeurs calculées par la simulation, mais que l'utilisateur n'a pas à modifier lui-même.

Vous pouvez modifier la constante qui définit la durée du lavage ou la capacité (nombre de postes de lavage), et voir comment cela affecte la file d'attente et le nombre de clients servis.

La **File** (Queue) ne vous permet pas de modifier de manière significative le comportement de la simulation si vous gardez une logique FIFO. Son onglet Results affiche des informations intéressantes. Vous pouvez examiner le contenu de l'onglet à la fin de la simulation, ou durant la simulation (en laissant le dialogue ouvert durant l'exécution).

Queue statistics

	Current	Average	Maximum
Length:	51	28,286545	52
Wait:	184,34784	110,20888	195,14688
Arrivals:	132		
Departures:	81		
Reneges:	0		
Utilization:	0,9993286		
Total Cost:	0		

Include items in queue at simulation end

dialogue Queue (onglet Results)

Les bibliothèques

Comme vous l'avez vu dans les rubriques précédentes, les documents de base d'ExtendSim sont des modèles. Nous n'avons examiné que très partiellement les bibliothèques et leur utilisation. Nous décrivons ici comment utiliser les bibliothèques, les types de bibliothèques qui sont fournis avec ExtendSim et comment gérer des bibliothèques.

Les bibliothèques fournies avec ExtendSim

ExtendSim inclus plusieurs bibliothèques, qui répondent à des besoins spécifiques, et qui ne seront pas forcément disponibles avec la version du produit que vous utilisez. Toutes les bibliothèques figurent dans le répertoire ExtendSim/Libraries. La version de base d'ExtendSim (CP) comporte les bibliothèques **Value**, **Chart**, **Animation**, **Report**, **Utilities** et **Electronics**. La bibliothèque **Item** est fournie si votre licence est du type DE ou Pro. Les bibliothèques **Rate** et **Reliability** se sont présentes qu'avec la version Pro.

Bibliothèque Value

La bibliothèque Value permet de résoudre presque tous les problèmes de simulation continue grâce à quelques blocs standard. La bibliothèque Value comporte des blocs de traitement mathématique de base, des blocs de décision et d'entrée/sortie. Ces blocs s'appliquent à la simulation financière, électronique, économique, démographique, biologique, physiologique, chimique, etc.

Bibliothèque Item (non fournie avec ExtendSim CP)

La bibliothèque Item possède tous les blocs de base pour représenter des files d'attente, des entités avec attributs et priorités pour les qualifier. Elle s'applique à la simulation de réseaux, de systèmes logistiques, d'ateliers de production, de flux de papier, aux files d'attente dans tous domaines, etc.

Bibliothèque Chart

La bibliothèque des graphiques comporte les graphiques courants permettant de visualiser des résultats. Certains graphiques sont spécifiques à certains domaines, d'autres sont généraux. Tous les graphiques sont décrits en détail dans la rubrique Graphiques.

☞ La bibliothèque Chart remplace la bibliothèque Traceurs/Plotter des anciennes versions – laquelle devient une bibliothèque « legacy », fournie afin de permettre l'utilisation d'anciens modèles.

Bibliothèque Animation

Les blocs de la bibliothèque Animation servent à l'animation 2D des modèles. L'animation 3D n'est pas encore disponible dans cette version d'ExtendSim 10.

Bibliothèque Utilities

Contient des blocs utilitaires pour tous modèles, tant pour l'interface utilisateur, l'aide à la mise au point que pour diverses actions sur un modèle.

Bibliothèque Rate (non fournie avec ExtendSim CP ou DE)

La bibliothèque Rate sert à décrire des systèmes où il y a à la fois des débits et des événements discrets. Elle s'occupe de flux continus (ou d'éléments discrets tellement petits ou en tel nombre qu'on les décrit par des débits, comme des liquides), mais obéit à un temps discret. S'emploie avec les bibliothèques Item et Value.

Bibliothèque Report

La bibliothèque **Report** contient des blocs qui stockent des informations dans divers formats sur diverses valeurs qui vous intéressent dans un modèle.

Bibliothèque Reliability (non fournie avec ExtendSim CP ou DE)

Combine la simulation de process avec des diagrammes de fiabilité (reliability block diagramming ou RBD). Permet de décrire de manière graphique et statistique les maintenance planifiées, horaires, pannes pouvant affecter des ressources, et l'impact que cela a sur la disponibilité du système global..

Bibliothèque Electronics

Des blocs d'électroniques permettant de simuler des systèmes analogiques et digitaux, du traitement de signal et du contrôle de systèmes.

Le répertoire **Example Libraires** contient des bibliothèques spécifiques servant à illustrer un point précis et utilisées par un modèle d'exemple. Elles illustrent des types d'utilisation d'ExtendSim dans des disciplines variées. Regardez le texte d'aide de ces blocs pour en savoir plus.

Le répertoire **Legacy** contient des bibliothèques qui existaient dans les anciennes versions d'ExtendSim (et notamment ExtendSim9), et ne servent désormais qu'à permettre d'exécuter d'anciens modèles. C'est là que figure notamment la bibliothèque Plotter (Traceurs).

Ouvrir et fermer des bibliothèques

ExtendSim ouvre automatiquement les bibliothèques courantes au démarrage (c'est contrôlé dans l'onglet Bibliothèques de la commande Edition>Options), et ouvre les bibliothèques nécessaires lorsque vous ouvrez un modèle. Sauf si vous voulez travailler sur une bibliothèque qui n'a pas été ouverte ou si vous souhaitez démarrer un modèle, vous n'avez pas à ouvrir par vous-même de bibliothèque. L'ouverture manuelle d'une bibliothèque se fait par l'option **Ouvrir** du menu Bibliothèque.

Lorsque vous ouvrez une bibliothèque, son nom s'ajoute à la fin du menu Bibliothèque. Pour voir les blocs de la bibliothèque, allez jusqu'à son nom : un menu hiérarchique s'affiche avec les blocs de cette bibliothèque. Vous pouvez ouvrir jusqu'à 30 bibliothèques.

Par défaut la fenêtre de la bibliothèque est ouverte à l'ouverture de la bibliothèque (aussi contrôlé dans l'onglet Bibliothèques de la commande Edition>Options)

Vous fermez une bibliothèque en choisissant **Fermer** dans le menu Bibliothèque : un dialogue permet alors de choisir les bibliothèques à fermer. C'est une opération que vous faites rarement, parce qu'une bibliothèque ouverte prend peu de place en mémoire et qu'il est pratique d'avoir les bibliothèques courantes ouvertes. ExtendSim ne permet pas de fermer une bibliothèque dont les blocs sont utilisés dans un modèle ouvert.

La recherche des bibliothèques

Généralement un modèle trouve très facilement les bibliothèques dont il a besoin. Mais si vous avez modifié ou déplacé un modèle ou une bibliothèque, ExtendSim ne pourra peut-être pas trouver la bibliothèque. Il vous demandera alors de localiser le bloc et sa bibliothèque.

À la recherche de la bibliothèque manquante, ExtendSim parcourt d'abord le répertoire où figure le modèle (sans ses sous-répertoires), puis le répertoire Libraries où se trouvent les bibliothèques, ainsi que les sous-répertoires.

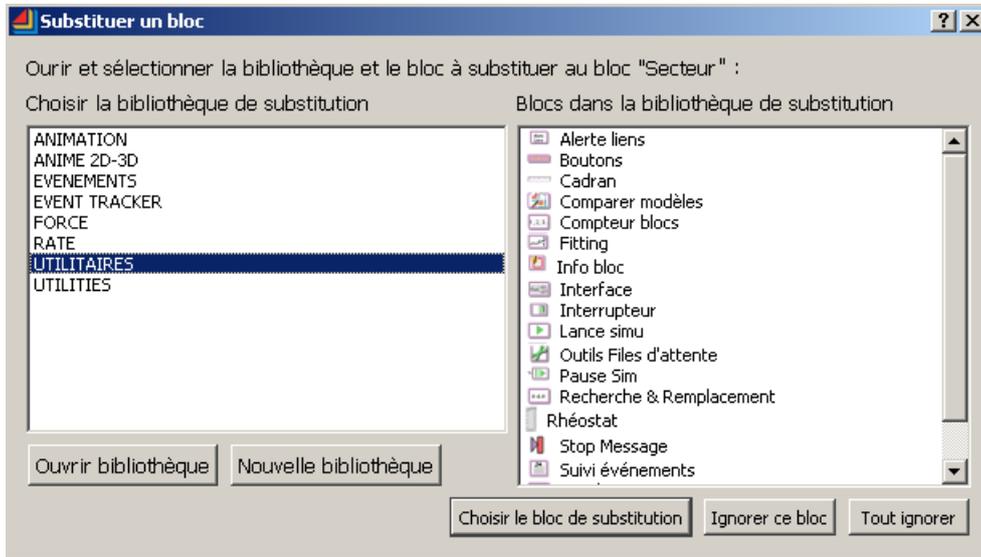
Vous pouvez interrompre cette recherche à tout moment en appuyant sur CTRL-Point.

Un répertoire de préférence peut être spécifié par la ligne « Chemin alternatif » de l'onglet Bibliothèque du dialogue Options : il sera parcouru en premier, avant même le répertoire Libraries.

Dans ce cas ou en cas d'échec de sa recherche, ExtendSim affiche un dialogue de sélection de fichier. Vous pouvez soit Annuler la recherche de bibliothèque (et ExtendSim proposera de trouver le bloc autrement), soit balayer jusqu'à trouver l'emplacement de la bibliothèque à ouvrir.

La prochaine fois que le modèle sera enregistré, l'emplacement de la bibliothèque sera mémorisé, et une seconde recherche ne sera pas nécessaire.

Lorsqu'il a localisé les bibliothèques, ExtendSim cherche les blocs. Si un bloc n'est pas trouvé (il a été supprimé ou renommé, ou sa bibliothèque n'a pas été ouverte), on vous demande le nom de la bibliothèque qui contient ce bloc. La possibilité de substituer un bloc à un autre est offerte à ce stade : en le choisissant dans une bibliothèque ouverte, ou en ouvrant une autre bibliothèque.



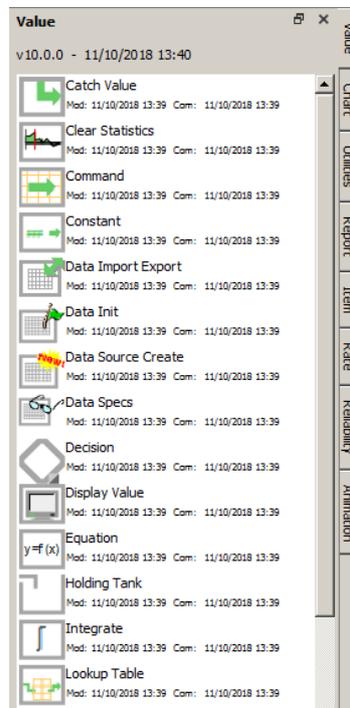
dialogue Substituer un bloc

Si vous cliquez sur Annuler, cela signale que le bloc n'existe plus ou que vous ne trouvez pas de substitution, et ExtendSim ouvrira le modèle 'sans titre' et affichera un texte à la place du bloc dans le modèle, qui sera peut-être inutilisable.

Les fenêtres des bibliothèques

Par défaut les bibliothèques ouvertes sont visibles par une fenêtre indépendante qui liste tous les blocs de la bibliothèque. C'est souvent via cette fenêtre que l'on ajoute les blocs à un modèle.

Vous faites la même chose manuellement si vous sélectionnez l'option **Ouvrir fenêtre** de chaque menu spécifique à une bibliothèque : ExtendSim ouvre alors une fenêtre pour cette bibliothèque. Par exemple voici une fenêtre avec la bibliothèque Value au premier plan, et les autres bibliothèques ouvertes accessibles par des onglets :



fenêtre de bibliothèque

L'ordre des onglets est l'ordre d'ouverture des bibliothèques ou celui du dialogue Options : vous pouvez changer la position des onglets en les faisant glisser. Le haut de la fenêtre fournit des informations sur la version et la date de dernière modification de la bibliothèque. Les blocs sont ensuite listés, avec une représentation de leur icône et (si vous l'avez demandé dans les **Options**, accessibles par le menu **Édition**) la date de leur dernière modification et de leur dernière compilation,

avec le type de compilation. Vous pouvez faire défiler la liste, déplacer et redimensionner la fenêtre comme toute autre fenêtre.

La fenêtre de bibliothèque est ancrée par défaut. Si vous voulez une fenêtre flottante pour une bibliothèque, amenez-la au premier plan et cliquez dans la barre de titre pour la détacher. Vous pourrez la faire glisser pour l'ancrer à nouveau.

Si un bloc a été compilé avec du code source externe, il est listé dans la fenêtre avec la mention *CM* (code management) à la droite de l'icône. Si le bloc a été compilé avec du **code de débogage**, les informations le concernant sont écrites en rouge dans la fenêtre, et son icône est encadrée de rouge.

Placer les blocs dans un modèle

Il y a quatre manières de placer un bloc dans un modèle :

1. **Pointer-cliquer** : sélectionnez le bloc dans la fenêtre de bibliothèque, puis à l'emplacement où vous voulez le placer dans la modèle. Fonctionne avec tout type de bloc.
 - ☞ Pour placer plusieurs copies d'un même bloc, maintenez enfoncée la touche Alt tandis que vous cliquez en plusieurs emplacements. Utilisez la touche Echap pour annuler.
2. **Astulien** : placez le premier bloc, puis du bouton droit de la souris sur le connecteur de sortie de ce bloc. Une liste de blocs vous est proposée, constituée des blocs les blocs logiques à placer après celui-ci (ou de ceux que vous avez l'habitude de placer). Cela a l'avantage de requérir moins de mouvements de la souris, et de relier les blocs tout en les plaçant.

Les astulien utilisent la fonctionnalité de script d'ExtendSim ainsi qu'un algorithme d'apprentissage adaptatif, de sorte que la liste reflète votre usage des blocs.

 - ☞ Les astulien ne sont disponibles qu'avec les blocs des bibliothèques Item et Rate. Le bloc Executive doit être déjà placé dans le modèle.
3. **Insertion automatique** : pour insérer un bloc Discret ou Rate entre deux autres déjà reliés, cliquez sur le bloc à placer, puis cliquez sur le lien de sorte à ce que les connecteurs du bloc figurent par-dessus le lien. Le nouveau bloc est inséré entre les autres.
 - ☞ Cette technique n'est disponible qu'avec les blocs des bibliothèques Item et Rate. Elle est contrôlée par l'onglet Modèle du dialogue Options.
4. **Lien par contact** : pour insérer un bloc Discret ou Rate à la suite d'un bloc existant, cliquez sur le bloc à placer, puis faites contact entre le connecteur itemIn ou inFlow et le connecteur de sortie correspondant. Le bloc est placé et relié au bloc mis en contact.
 - ☞ Cette technique n'est disponible qu'avec les blocs des bibliothèques Item et Rate.

Types de connecteurs

Le tableau ci-dessous présente les différents types de connecteurs et leur fonction.

Type	Entrée	Sortie	Emploi
Continu			Passes des valeurs (des nombres) d'un bloc à un autre. Les blocs des bibliothèques Item et Rate utilisent aussi ces connecteurs parfois.
Discret			Passes des entités (individus de base du flux) d'un bloc à un autre. Les blocs de la bibliothèque Rate utilisent aussi ces connecteurs parfois.
Flux			Passes le débit réel du flux d'un bloc à un autre (bib. Rate). Ces connecteurs peuvent être reliés en directs à des connecteurs d'entrée continus.
Universel			Connecteur universel plutôt utilisé en entrée : des connecteurs continus, discrets, flux et utilisateur peuvent s'y relier (pour utilisateurs avancés).
Matrice			Connecteur matrice servant à passer uniquement des informations sur des matrices d'un bloc à un autre (pour utilisateurs avancés).
Utilisateur			Connecteur ayant un usage spécifique programmé par l'utilisateur (pour utilisateurs avancés).

Connecteurs variables

Indépendamment de son type, chaque connecteur peut être simple ou variable, si le bloc l'a prévu comme tel. Les connecteurs variables sont comme un ensemble de connecteurs simples similaires : on peut en augmenter ou réduire le nombre. Une flèche signale un connecteur variable

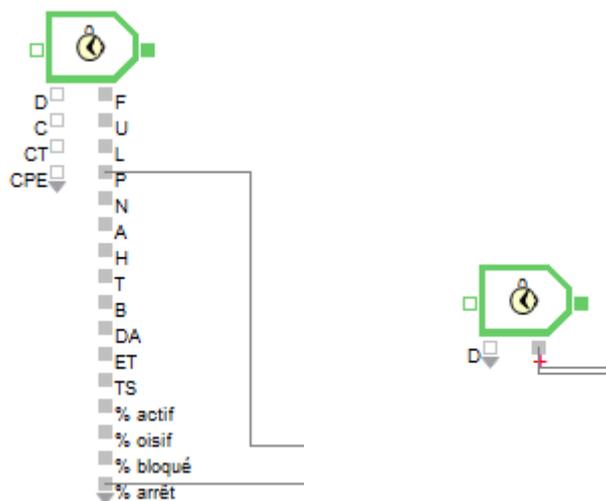


bloc Constant avec connecteurs simples

bloc Math avec un connecteur variable

En faisant glisser la flèche à deux pointes qui sert de curseur lorsque vous placez la souris sur le triangle, vous déployez le connecteur, ou bien vous le ramassez. Le choix de certaines options dans le dialogue peut aussi immédiatement se refléter par l'ajustement du nombre de connecteurs variables.

Certains blocs, notamment des blocs discrets, peuvent posséder des connecteurs variables très nombreux, qui s'ils sont déployés gêneraient la lisibilité du modèle. Il est possible en double-cliquant sur le triangle de ramasser d'un coup les connecteurs (sans couper les liens existants). Le triangle devient une croix rouge, qui par double-clic déploie à nouveau les connecteurs.



connecteurs déployés

connecteurs ramassés par double-clic

- Vous ne pouvez pas déployer un connecteur variable au-delà de ce qui est considéré comme raisonnable pour le bloc, compte-tenu de sa fonction et de son paramétrage. Certains blocs, comme les blocs Equation, ne permettant pas à l'utilisateur d'utiliser la souris pour augmenter le nombre de connecteurs : ils reflètent des paramètres du dialogue.

Faire évoluer des bibliothèques

- Ne modifiez pas les bibliothèques fournies avec ExtendSim, vous compromettriez le bon fonctionnement de vos modèles avec les évolutions du logiciel.

ExtendSim facilite le déplacement de blocs entre bibliothèques, et la suppression de blocs. Pour copier un bloc d'une bibliothèque à l'autre, ouvrez chaque fenêtre de bibliothèque, cliquez du bouton droit pour **Copier les blocs** sélectionnés dans la bibliothèque source, et similairement *Coller les blocs* dans la cible.

Pour avoir une copie d'un bloc dans la même bibliothèque, sélectionnez le bloc dans la fenêtre et cliquez du bouton droit pour **Dupliquer les blocs**. La copie porte le même nom que l'original suivi d'un numéro séquentiel. C'est utile lorsque vous voulez utiliser le script d'un bloc pour base d'un bloc différent, ou lorsque vous souhaitez deux blocs de même comportement mais d'icône différente.

Pour changer le nom d'un bloc, sélectionnez le bloc dans la fenêtre, cliquez du bouton droit pour **Renommer le bloc** puis entrez le nouveau nom du bloc. Si vous changez le nom d'un bloc, les modèles qui utilisent ce bloc ne pourront plus le trouver, puisqu'ils le rechercheront par son nom original. ExtendSim affichera un message d'erreur et vous aurez la possibilité de substituer un nouveau bloc à l'ancien (voir plus haut).

Il est rare que vous vouliez supprimer un bloc d'une bibliothèque, mais vous pouvez le faire en cliquant du bouton droit pour **Supprimer le bloc** ou par la touche d'effacement arrière. ExtendSim

ne vous laissera pas supprimer un bloc qui est employé dans un modèle ouvert. Si vous supprimez un bloc employé dans un modèle et que vous ouvrez ensuite le modèle, ExtendSim affichera un message d'erreur et une image remplaçante dans le modèle.

Compiler des bibliothèques

Chaque fois que vous modifiez quelque chose dans une bibliothèque, ExtendSim enregistre automatiquement la bibliothèque. Lorsque vous modifiez quelque chose au programme d'un bloc, vous devez le recompiler, par les commandes de compilation des menus **Développeur** et **Bibliothèque** (menu hiérarchique **Outils bibliothèques**).

- La commande **Compiler** compile le bloc sans fermer la fenêtre de structure. C'est pratique pour tester que le code ne comporte pas d'erreur de syntaxe et continuer de travailler sur le bloc. La commande n'est accessible que lorsque la fenêtre de structure est ouverte.
- La commande **Compiler des bibliothèques** (menu hiérarchique Outils bibliothèques) compile intégralement les bibliothèques que vous sélectionnez. Vous l'utiliserez rarement, puisque la compilation est proposée chaque fois que vous modifiez un bloc individuel.
- La commande **Compiler sélection** compile les blocs sélectionnés d'une même bibliothèque. Vous l'utiliserez rarement aussi, puisque la compilation est proposée chaque fois que vous modifiez un bloc individuel. La commande n'est accessible que lorsque la fenêtre de bibliothèque est ouverte.

Protéger des bibliothèques

Une bibliothèque protégée est utilisable comme une bibliothèque normale, sauf qu'on ne peut visualiser ou modifier les scripts. Cela signifie que quelqu'un qui utilise cette bibliothèque bénéficie des fonctionnalités des blocs, mais ne peut voir comment ils ont été réalisés. Cela permet une confidentialité de la programmation, et protège le script, l'icône et le texte d'aide d'une modification non souhaitée.

Vous pouvez empêcher l'accès à votre code en écrivant l'essentiel dans des fichiers *include* (voir pour cela le manuel Technical Reference), ou en supprimant le code des blocs, ce qu'on appelle protéger la bibliothèque.

Pour protéger une bibliothèque, vous devez utiliser l'option **Protéger une bibliothèque** sous la commande **Outils bibliothèques** dans le menu Bibliothèque. Cela crée une copie de la bibliothèque où le code source est supprimé. Conservez la bibliothèque originale, car une bibliothèque protégée ne peut être déprotégée. N'empêchez pas l'outil de conversion de renommer la bibliothèque.

En quelques instants la bibliothèque est protégée. Vous pouvez le vérifier en ouvrant la bibliothèque et en essayant de modifier un bloc. Si vous faites Alt-double-clic sur un bloc, ExtendSim affichera un dialogue indiquant que la bibliothèque est protégée et qu'elle ne peut être ouverte. Les fenêtres de structure et de dialogue sont inaccessibles.

Pour protéger la structure d'un bloc H et empêcher l'utilisateur de double-cliquer sur le bloc pour voir le sous-modèle, utilisez la commande **Modèle>Verrouiller le modèle** examinée au paragraphe **Verrouiller le modèle**.

Diffuser des bibliothèques

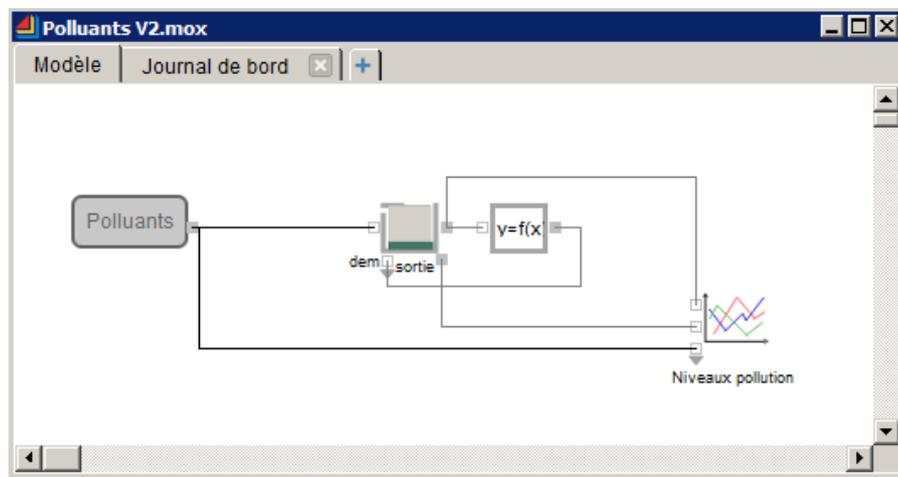
S'il vous plaît, n'oubliez pas qu'il faut l'application ExtendSim pour exécuter les bibliothèques. Vous pouvez distribuer vos bibliothèques personnelles à d'autres utilisateurs ExtendSim, mais il est explicitement interdit de diffuser des copies de l'application ExtendSim (consultez les Conditions d'utilisation incluses dans le package). Une version limitée d'ExtendSim appelée Analyse permet une large diffusion des applications ExtendSim : consultez-nous. Cette version peut ouvrir les bibliothèques en format natif, ou converties en un format spécial qui évite que la bibliothèque puisse ensuite être ouverte par les versions développeur d'ExtendSim, ce qui empêche que votre travail serve à d'autres pour construire leurs modèles et protège votre savoir-faire. Voir la commande **Bibliothèques> Outils bibliothèques > Convertir bib. pour runtime**.

L'interface des modèles

Maintenant que vous savez comment construire et lancer un modèle, vous vous demandez sans doute ce qu'il vous reste à apprendre... ExtendSim n'est pas uniquement un simulateur : il vous permet de présenter vos résultats de manière lisible et efficace.

Certains modèles peuvent devenir très complexes. En lisant cette rubrique, vous saurez comment améliorer la construction, la simulation et la présentation de vos modèles. Vous saurez à la fois faire que le modèle exécute ce que vous voulez, et qu'il l'exprime comme vous le voulez : c'est en cela qu'ExtendSim est aussi un outil de communication.

Dans cette rubrique nous examinerons un modèle fictif "Polluants v2", qui est une version améliorée du modèle "Reservoir" examiné dans le *Guide de prise en main ExtendSim continu*, et se présente comme suit :



modèle "Polluants V2"

Boutons et outils

Vous l'avez déjà constaté : vous agissez sur le modèle par les boutons de la frise d'outils en haut de l'écran. Nous présentons ici les outils disponibles lorsqu'un modèle est ouvert. D'autres outils sont disponibles lorsque vous construisez ou modifiez un bloc, mais ne vous sont pas utiles à ce stade.

Vous contrôlez la taille des boutons par l'onglet Divers du dialogue Options. Chaque groupe d'outils peut être ancré ou non, déplacé en bas, à gauche ou à droite, rendu visible ou non (menu Outils).

Fichiers



Un premier groupe de 4 boutons standard concerne les opérations de base sur les fichiers, à savoir Nouveau modèle, Ouvrir modèle, Enregistrer modèle et Imprimer.

Edition



Le second groupe est tout aussi classique, avec les boutons Couper, Copier, Coller, Annuler et Refaire. Plusieurs niveaux d'annulation sont enregistrés.

Les trois boutons suivants gèrent le zoom sur le modèle, et le dernier ouvre le dialogue permettant de rechercher des blocs, des objets ou du texte.

Simulations



Le groupe suivant comporte huit boutons agissant sur les simulations du modèle courant : Mode de simulation, Lancer la simulation, Stopper la simulation, Lancer une Optimisation/Scénarios, Activer/Désactiver l'animation 2D, curseur pour accélérer/ralentir l'animation, Afficher le Navigateur.

Sélection

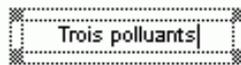
Ces quatre outils permettent de sélectionner des éléments dans le modèle. Le premier curseur est l'outil de base qui sert à manipuler les menus et les blocs. Les autres curseurs servent respectivement aux objets graphiques, aux clones et le dernier à tous les types d'éléments. L'outil le plus utilisé, **Sélection bloc/texte**, est activé par défaut. Les outils plus sélectifs sont utiles lorsque des éléments de genres différents sont très proches ou se chevauchent. Les coordonnées indiquent la position du curseur.

Ajouter et manipuler du texte

Pour commenter le modèle par du texte, double-cliquez directement à l'endroit désiré : vous créez ainsi une zone de texte dans laquelle vous tapez du texte. Pour signaler la fin de la saisie, appuyez sur Entrée ou cliquez n'importe où ailleurs dans le modèle. Vous pouvez aussi utiliser le bouton *Zone de texte* dans les outils Formes.

Pour accéder ensuite à la zone de texte, vous devez utiliser les outils Sélection bloc/texte ou Tout sélectionner puis double-cliquer dans le texte.

Lorsque vous tapez, la zone de texte se présente sous forme d'une case grise entourée de poignées. Par exemple, tapez le texte suivant :



zone de texte

Avant d'appuyer sur Entrée, vous pouvez cliquer sur une poignée et faire glisser pour modifier la taille ou la forme de la zone.

Vous pouvez déplacer la zone de texte en la sélectionnant et la faisant glisser à l'emplacement désiré. Vous supprimez un texte en le sélectionnant avec le curseur puis en appuyant sur la touche Suppr ou par l'option Effacer du menu Edition.

Vous pouvez formater le texte via les **outils texte**



Pour cela, sélectionnez le texte puis choisissez un bouton, par exemple Gras ou Centré.

Si vous voulez taper un nouveau texte qui ait d'emblée certains attributs, sélectionnez ces attributs avant de créer la zone de texte. ExtendSim mémorise ce format pour chaque nouveau texte que vous créez désormais. Si vous modifiez les attributs texte pour une zone de texte déjà créée, ExtendSim ne mémorisera pas ces attributs pour une nouvelle zone de texte.

Vous pouvez encore modifier la couleur du texte, ou lui ajouter un cadre, via les derniers boutons des outils Formes.

Le texte peut être copié et déplacé par les boutons d'édition, par des raccourcis clavier (indispensable Ctrl-D pour dupliquer) mais aussi par les techniques de "glisser-déposer", et ce y compris dans un fichier texte ou un tableur.

☞ Lorsque vous copiez du texte dans le modèle ou dans une zone de texte existante, cela reste du texte éditable. Mais lorsque vous sélectionnez une zone de texte entière et que vous la copiez dans le journal de bord, cela devient une image. Pour que cela reste du texte, sélectionnez seulement le texte dans la zone de texte.

Lorsque d'autres fenêtres que la fenêtre du modèle sont actives, l'outil Sélection bloc/texte reste l'outil par défaut, mais son comportement s'adapte à l'emploi le plus courant. Par exemple, lorsqu'un journal de bord est ouvert, l'outil par défaut permet de sélectionner des éléments de dessin aussi bien que du texte.

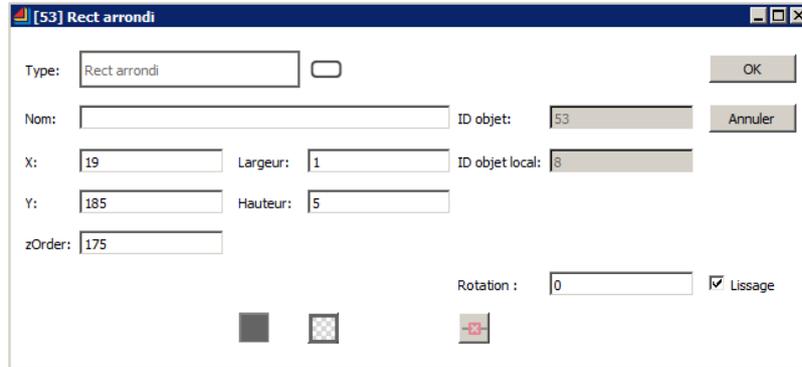
Formes

Ce sont divers outils associés au dessin des icônes de blocs et éléments de modèles. Ils permettent de dessiner des formes géométriques dans le modèle. L'objectif de ces outils est d'augmenter la lisibilité et l'esthétique des modèles. Ils permettent aussi de choisir le type d'objet à Copier. Ils n'ont aucun rôle fonctionnel.

Les outils **Rectangle**, **Rectangle arrondi**, **Ovale**, **Polygone** et **Ligne** permettent d'ajouter ces formes géométriques dans votre modèle. Par exemple, pour placer un rectangle, sélectionnez l'outil Rectangle, cliquez à un emplacement choisi pour l'un des coins du rectangle et faites glisser en diagonale.

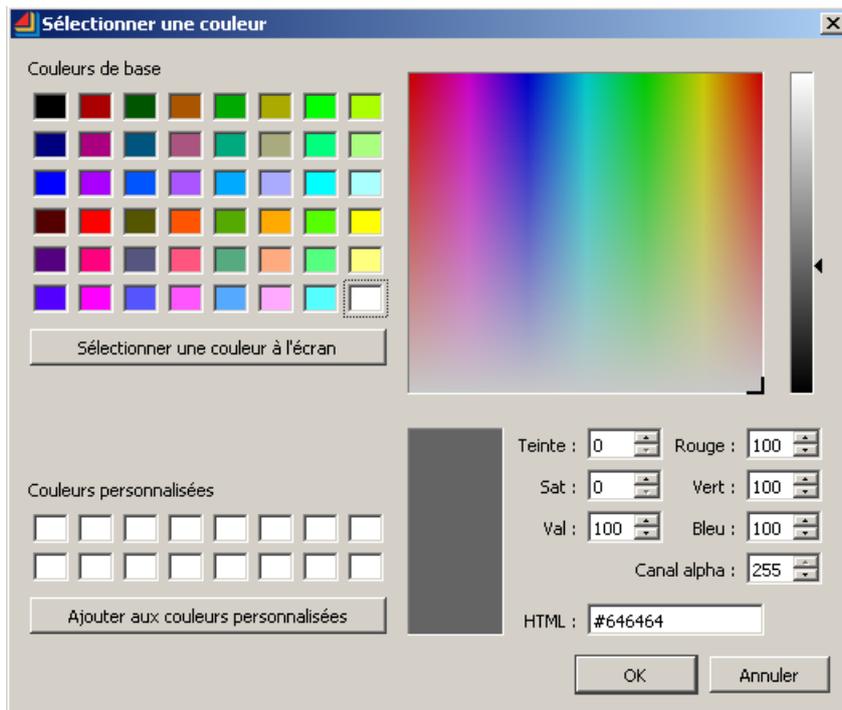
Si vous maintenez enfoncée la touche Maj lorsque vous redimensionnez un rectangle, un rectangle arrondi ou un ovale, la forme dessinée est un carré ou un cercle, et tout changement de taille se fait de manière proportionnelle. Les touches fléchées du clavier peuvent s'employer pour déplacer ces éléments de 1 pixel par clic (0,1 pixel si la touche Alt est enfoncée).

Toute figure géométrique, ainsi que le texte, est associée à des **propriétés** visibles dans une fenêtre accessible par le menu du bouton droit, ou par la commande Fichier>Propriétés. Vous pouvez modifier les propriétés, notamment de taille ou d'emplacement, ce qui est parfois plus précis qu'avec la souris.



Cadres et couleurs

Tout élément dessiné, y compris le texte, possède une couleur choisie parmi le nuancier Couleur remplissage. Lorsque vous cliquez sur le bouton des couleurs, la palette des couleurs s'affiche. Vous pouvez taper du texte en couleur ou colorer un texte existant en le sélectionnant et en choisissant une couleur dans la palette. Si vous changez de couleur avant de créer la zone de texte, le texte sera dans cette couleur, comme expliqué page précédente.



palette de couleurs

Notez que lorsque vous sélectionnez une couleur, ses caractéristiques HSV (teinte, saturation, valeur), HTML et RVB sont aussi indiquées au bas de la palette. Si vous devez donner des couleurs à des objets d'animation, ces paramètres vous seront utiles.

Les lignes ont par défaut une épaisseur de 1 pixel, qui peut être augmentée jusqu'à 5 pixel par l'outil Épaisseur ligne/cadre, le dernier de la palette.

Les objets, à l'exception des lignes, peuvent avoir un cadre dont l'épaisseur se définit par ce même outil, et dont la couleur est définie par l'icône Couleur bordure. Les lignes n'ayant pas de cadre, changer cette couleur pour une ligne revient à changer la couleur de la ligne.



Outils d'alignement

Ces outils permettent d'aligner du texte ou des objets graphiques, de gérer les plans des objets qui se superposent, de faire pivoter des objets.

Le changement de plan concerne autant les objets graphiques que les blocs, le texte et les clones de dialogue.

Les options de **Rotation** ou de **Bascule** s'activent lorsque vous sélectionnez un objet graphique.

Cloner des éléments de dialogue sur le modèle

Dans des gros modèles, devoir passer par des dialogues pour tous vos paramétrages peut devenir un inconvénient. Vous voudriez avoir un accès facile à divers paramètres dispersés dans les dialogues de plusieurs blocs du modèle. ExtendSim permet pour cela de cloner des éléments de dialogues et de les placer à un endroit plus adapté.

Par défaut, les éléments de dialogue figurent dans les dialogues. Vous pouvez faire glisser un clone (une copie) d'un élément de dialogue dans la fenêtre du modèle, dans un journal de bord ou dans la fenêtre d'un bloc hiérarchique. Le [journal de bord](#) est souvent l'endroit où l'on place ces clones.

Vous pouvez cloner un élément vers plusieurs emplacements. Par exemple, en deux endroits de la fenêtre du modèle. Chaque clone se comporte exactement comme l'original : si vous modifiez l'original ou un clone, toutes les instances sont mises à jour immédiatement.

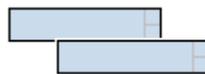
Cet emploi de clones vous donne un contrôle direct sur le modèle. Vous pouvez utiliser les éléments de dialogue en cours de simulation (cliquer sur des boutons ou entrer des paramètres dans des zones), pour tester des valeurs en temps réel. Vous pouvez aussi dans une portion du modèle rassembler plusieurs indicateurs numériques.

☞ Les clones qui affichent des valeurs en temps réel durant la simulation en ralentissent l'exécution.

Le clonage

Ouvrez le dialogue voulu et sélectionnez l'outil de clonage, . Sélectionnez l'élément de votre choix puis cliquez à l'emplacement voulu. Vous pouvez aussi utiliser le menu du bouton droit. Pour sélectionner plusieurs éléments, faites une marquise de sélection ou appuyez sur la touche Maj tout en sélectionnant des éléments. Fermez ensuite le dialogue.

Par exemple, vous voulez pouvoir modifier un taux de pollution durant la simulation. Clonez l'élément correspondant dans un des blocs Random Number. Double-cliquez sur le bloc pour l'ouvrir. Sélectionnez l'outil de clonage, cliquez sur l'élément puis cliquez à l'emplacement (même provisoire) où placer le clone.



clonage en cours

Vous pouvez bien sûr ajouter un label de texte à côté de cet élément.

Lancez la simulation. Pour modifier le taux durant la simulation, cliquez dans la zone et tapez une nouvelle valeur. ExtendSim fait une pause pour que vous puissiez entrer le nombre. Cliquez ensuite sur le bouton Continuer en haut de l'écran pour reprendre la simulation avec la nouvelle valeur.

Vous pouvez faire plusieurs copies d'un clone, à partir de l'original ou d'un clone.

Utiliser des éléments clonés

Lorsqu'un clone est dans un modèle, vous pouvez le déplacer comme vous le voulez. Il est souvent pratique de réunir les clones à un même endroit (vous verrez l'utilité du journal de bord dans ce cas). Pour déplacer un clone, sélectionnez l'outil de clonage, cliquez sur le clone et faites-le glisser.

Les clones peuvent être redimensionnés. Pour cela, cliquez avec l'outil Clone de sorte qu'une poignée apparaisse dans le coin inférieur droit. Vous pouvez alors faire glisser la poignée pour changer la taille du clone. Vous pouvez aussi utiliser les coordonnées dans le dialogue Propriétés relatif au clone.

Vous ouvrez directement le dialogue d'origine d'un clone en double-cliquant sur le clone avec l'outil de clonage. Notez que les graphes des graphiques et leurs tables de chiffres peuvent aussi être clonés, tout comme les boutons ou zones de saisie.

Pour enlever des clones d'un modèle, sélectionnez-les avec l'outil de clonage et utilisez la touche Suppr ou la commande Edition>Effacer. Si vous supprimez un bloc d'où provient un élément cloné,

le clone est automatiquement supprimé. Mais si vous modifiez seulement le bloc pour changer l'ordre des éléments de dialogue ou supprimer un élément cloné, le clone sera grisé et représenté par trois points d'interrogation. Double-cliquez sur le clone avec l'outil de clonage, pour rétablir la liaison avec le dialogue d'origine, ou supprimez le clone "orphelin".

Les journaux de bord

Un journal de bord est une fenêtre que vous personnalisez pour organiser les données importantes d'un modèle. Chaque modèle a posséder au moins un onglet journal de bord qui peut contenir des clones d'éléments de dialogue, du texte, des images et des éléments de dessin. Vous utilisez le journal pour contrôler des paramètres du modèle, pour obtenir un rapport mis en forme des résultats de la simulation, et pour documenter votre modèle.

Vous pouvez utiliser un journal pour des valeurs saisies (en entrée) ou observées (en sortie). Autrement dit, pour modifier dynamiquement des paramètres, ou pour réunir des résultats d'une manière pratique à observer et à imprimer. Vous ajouterez des onglets en fonction de vos besoins, pour regrouper thématiquement ces divers éléments.

L'emploi le plus courant du journal est de réunir tous les éléments qui vous intéressent dans un même endroit. Comme vous pouvez laisser le journal ouvert durant la simulation, il peut devenir une fenêtre de résultats complète regroupant toutes les valeurs importantes des dialogues. C'est aussi un vrai tableau de bord, dans lequel vous clonez tous les paramètres que vous souhaitez modifier en cours de simulation.

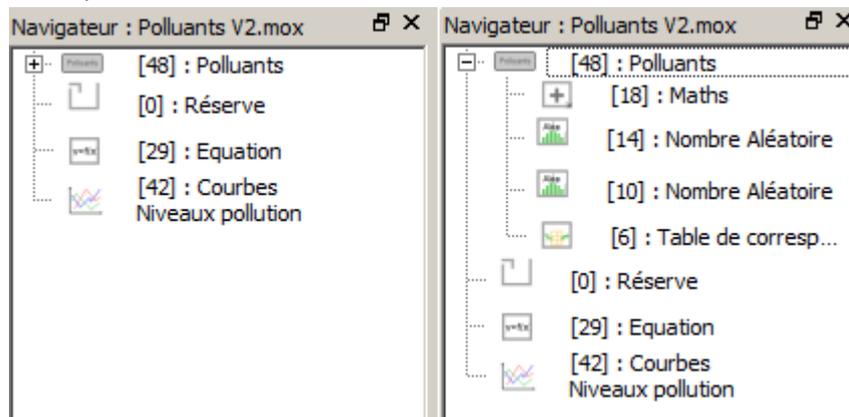
Considérez le journal de bord comme une boîte de dialogue. Lorsque vous modifiez une valeur clonée provenant d'un dialogue, ExtendSim répercute la modification dans le dialogue d'origine.

Pour créer un onglet de journal supplémentaire, cliquer sur le bouton à la droite de l'onglet Journal de bord. Pour cloner un élément dans le journal, choisissez l'outil de clonage, , cliquez sur l'élément de dialogue puis dans le journal. Pour sélectionner plusieurs éléments, utilisez une marquette de sélection ou maintenez la touche Maj appuyée en faisant vos sélections. Le clonage des éléments est décrit page précédente. Ensuite vous arrangez les divers éléments dans la fenêtre du journal.

Un journal peut comporter plusieurs pages en largeur et en longueur. Les ruptures de pages sont affichées si l'option correspondante du menu Fichier est sélectionnée. Si vous avez de nombreux éléments dans un journal de bord et que vous voulez savoir de quel dialogue provient tel élément, choisissez l'outil de clonage et double-cliquez sur l'élément : le dialogue associé s'ouvre automatiquement.

Naviguer dans le modèle

Si besoin cliquez sur l'outil navigateur pour ouvrir la fenêtre de navigation. Il s'agit d'une fenêtre de type Explorateur qui permet de naviguer facilement dans un grand modèle. En l'ouvrant maintenant, vous voyez une liste de blocs :



Navigateur sur le modèle Polluants V2, avec bloc hiérarchique fermé et ouvert

Cliquez sur le signe + à côté d'un bloc hiérarchique pour afficher le contenu du bloc. Si vous cliquez sur l'un des blocs qui est apparu, cela ouvrira son bloc hiérarchique supérieur, et sélectionnera le bloc. En double-cliquant sur le bloc (par exemple Equation), le dialogue du bloc s'ouvrira aussi. Quel que soit le nombre de niveaux, seuls ces deux éléments s'ouvriront. Cela permet de ne pas surcharger l'écran de fenêtres ouvertes.

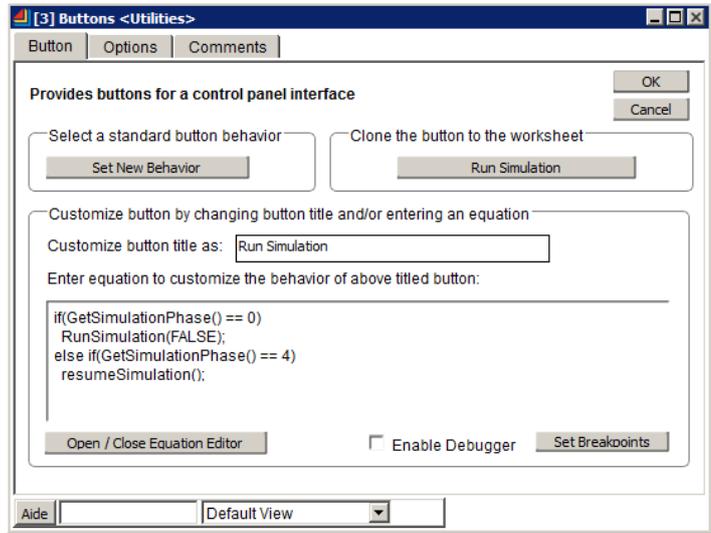
Des blocs pour un tableau de bord

Le bloc Buttons

Ce bloc de la bibliothèque Utilities permet à l'utilisateur de déclencher des actions courantes. Des actions prédéfinies sont proposées dans un menu du dialogue du bloc, par exemple :

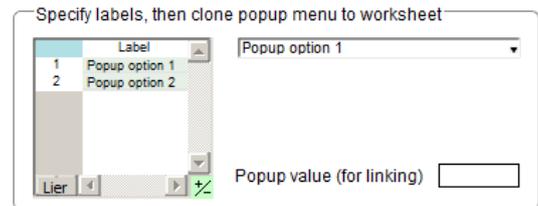
- Sans animation
- Avec animation
- Journal de bord
- Lancer simulation
- Pause simulation
- Enregistrer le modèle
- Ouvrir bloc
- Ouvrir bloc H
- Ouvrir table
- Ouvrir BDD
- Lancer Optimisation de scénarios

Vous clonez le bouton à l'emplacement de votre choix, et en cliquant dessus l'utilisateur déclenchera l'action correspondante (vous pouvez aussi préciser des conditions d'exécution automatique). Vous pouvez modifier les actions ou en créer de nouvelles, avec leur label et leur équation propres. Chaque bouton doit avoir son propre bloc Buttons associé (il ne peut y avoir plusieurs actions par bloc).



Le bloc Popups

Ce bloc de la bibliothèque Utilities permet à l'utilisateur de choisir une option parmi plusieurs. Sert de menu personnalisé à placer sur le modèle et qui lui injectera des valeurs en entrée. Le bloc *Popups* contrôle un autre bloc dans le modèle, par son connecteur de sortie, mais l'utilisateur ne voit généralement que le clone du menu.

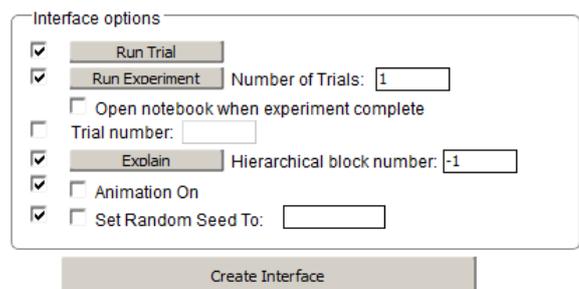


Le bloc Model Interface

Ce bloc de la bibliothèque Utilities est une aide à la création rapide d'une interface personnalisée. Les cases à cocher permettent de spécifier des éléments qui seront clonés sur le modèle.

Les blocs Run Model et Pause Sim

Les boutons de la barre d'outils Lancer, Pause, et Stop sont utilisés de manière interactive, mais les blocs Run Model et Pause Sim permettent un plus grand contrôle sur l'exécution du modèle.



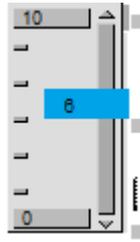
Les blocs de contrôle

La bibliothèque Utilities comporte trois blocs dits de « Contrôle », qui s'emploient pour ajouter un contrôle interactif direct des modèles. Ils permettent de contrôler d'autres blocs et affichent des valeurs directement durant la simulation.

Le Slider (ou rhéostat) et l'interrupteur s'emploient pour affecter des valeurs et n'ont que des connecteurs de sortie. Le Meter (ou cadran) est employé pour voir des valeurs durant la simulation et possède trois connecteurs d'entrée.

Slider

Le Slider ressemble à une glissière de niveau sur un appareil hifi. L'utilisateur pourra faire glisser le pointeur pour qu'il émette un nombre. Vous déterminez les valeurs de minimum et maximum en sélectionnant chaque nombre et en tapant la valeur désirée.



Slider

La valeur du Slider est émise sur le second connecteur du haut. Les connecteurs supérieur et inférieur indiquent le maximum et le minimum. Vous modifiez la sortie du Slider en déplaçant l'indicateur de niveau. La valeur est affichée et émise en sortie au fur et à mesure.

Le Slider est utile lorsque vous voulez sortir un nombre à l'intérieur d'une plage sans devoir être très précis. Par exemple, vous avez un bloc de délai pour lequel vous savez que 8 signifie lent, et 2 signifie rapide. Placez un rhéostat avec un minimum de 2 et un maximum de 8, et durant la simulation, déplacez le curseur pour faire évoluer la valeur entre ces deux bornes.

Interrupteur

L'interrupteur (bloc Switch) ressemble à un interrupteur standard, et sert à activer/désactiver quelque chose dans le modèle :



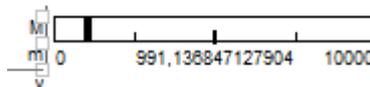
l'interrupteur Off et On

Lorsque vous cliquez sur la partie qui n'est pas enfoncée, elle prend l'autre valeur, et l'interrupteur "s'allume" et "s'éteint".

Ce bloc s'emploie pour contrôler des blocs ayant une entrée booléenne (logique), puisqu'ExtendSim admet 0 comme faux et 1 comme vrai. Il peut être manipulé par un clone, et associé à une structure de données d'ExtendSim.

Meter

Vous utilisez le Meter (cadran) pour visualiser des valeurs qui varient à l'intérieur de bornes connues. Vous déterminez le maximum et le minimum en reliant d'autres blocs (par exemple Constant) aux connecteurs du haut et du bas, ou par le dialogue. Voici le cadran :



le Meter

Cela permet de voir des valeurs évoluer sans pour autant utiliser un graphique de courbes.

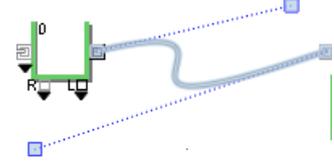
Styles et manipulation des liens

Les lignes qui représentent les liens peuvent recevoir des styles qui les rendent plus expressives. L'option **Liens** du menu **Modèle** permet de modifier le style des liens. Choisissez une option dans le menu hiérarchique puis tracez un nouveau lien, ou inversement:

menu hiérarchique Liens

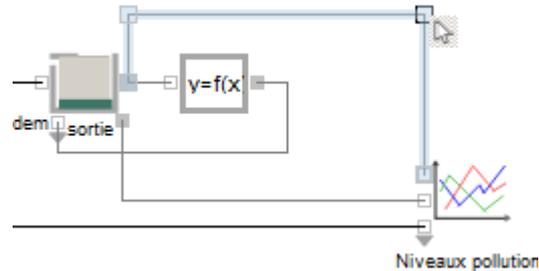
	Intelligents	
	À angle droit	Ctrl+Maj+D
	Droit	Ctrl+Maj+A
	Libres	
	Pas de flèche	
	Sortie vers entrée	Ctrl+Maj+E
	Entrée vers sortie	Ctrl+Maj+F
	Ligne pleine	Ctrl+Maj+N
	Ligne pointillée	Ctrl+Maj+O
	Couleur des liens	Ctrl+Maj+T
	Epaisseur par défaut	Ctrl+Maj+G
	1 Pixel	Ctrl+Maj+H
	2 Pixel	Ctrl+Maj+J
	3 Pixel	
	4 Pixel	Ctrl+Maj+K
	5 Pixel	
	Ligne double	Ctrl+Maj+M

Les liens « intelligents » sont des liens à angle droit qui s'adaptent au déplacement des blocs. Si vous choisissez les angles droits, ExtendSim transformera le lien sélectionné en trois segments de droite liés par des points d'ancrage. Le lien Droit est une diagonale entre les deux connecteurs. Les liens libres ne sont pas pré-formatés : vous pouvez en tracer tous les points constitutifs comme pour une courbe de Bézier. Vous pouvez indiquer quel type de ligne adopter par défaut dans l'option **Options** du menu Edition.



Chaque angle sur un lien à angle droit est appelé Point d'ancrage.

Sauf pour les liens intelligents où l'utilisateur n'intervient pas, vous pouvez cliquer sur un des points d'ancrage et le faire glisser pour agencer le lien à votre guise. Notez qu'ExtendSim conserve toujours des angles droits, où que vous placiez les points d'ancrage :



point d'ancrage déplacé

Le second groupe du menu hiérarchique permet d'ajouter une pointe de flèche à vos liens. Les flèches ne sont utilisables qu'avec les lignes à angles droits, et la direction de la flèche suit le sens de votre tracé du lien.

Le troisième groupe de choix permet que les lignes à angle droit soient en pointillés, plutôt qu'en continu. Il n'est pas disponible pour les lignes en diagonale.

Les liens (nommés ou non) qui sont mal connectés sont affichés en pointillés et en rouge. Cela permet de voir rapidement les liens mal faits.

Le quatrième groupe du menu concerne la couleur des liens, qui, comme les textes ou les objets de dessins, peuvent être colorés. Le choix Liens noirs prévaut sur la couleur sélectionnée dans la palette de couleurs. Si vous choisissez Liens couleurs, tout nouveau lien tracé aura la couleur sélectionnée dans la palette de couleurs. Pour modifier la couleur d'un lien, choisissez Liens couleurs, sélectionnez le lien puis choisissez une couleur dans la palette.

Les derniers choix déterminent l'aspect des liens. Les lignes peuvent être fines, moyennes ou épaisses. ExtendSim permet en plus de différencier automatiquement les types de liens en fonction du type de connecteur.

Les types de liens par défaut sont :

Continu	
Discret	
Flux	

Par exemple, la plupart des modèles discrets observent des flux d'entités tout en calculant des valeurs. Vous pouvez différencier au premier coup d'œil le flux matière des signaux continus grâce au type de lien par défaut.

Vous pouvez sélectionner tous les segments d'un lien, afin de pouvoir supprimer le lien ou en modifier les propriétés. Cela s'applique aux liens standard comme aux liens nommés. En double-cliquant sur n'importe lequel des segments d'un lien, tout le lien est sélectionné entre les deux blocs reliés.

Pour sélectionner tous les segments issus d'une même sortie, double-cliquez sur un des segments en gardant la touche Alt appuyée.

Liens nommés

Pour éviter des modèles embrouillés, utilisez des *liens nommés*. Un lien nommé est un label de texte qui représente une sortie à plusieurs emplacements dans le modèle. Vous pouvez imaginer en dessous du modèle un réseau de liens correspondant, mais qui n'est pas visible dans le modèle. Pour créer un lien nommé, entrez le label de texte près de la sortie désirée, puis tirez un lien depuis le connecteur de sortie vers le label. Lorsque vous tapez le même texte près d'un connecteur d'entrée et que vous tirez un lien depuis ce connecteur d'entrée vers le label, le lien nommé peut fonctionner. Choisissez *Voir les liens nommés* dans le menu Edition pour voir les liens nommés.

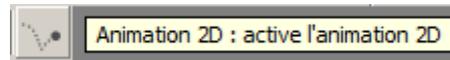
Le texte des liens nommés doit être strictement le même. La différence entre majuscules et minuscules est ignorée, mais des espaces ou un retour-chariot seront pris en compte. Si vous utilisez des liens nommés, il peut advenir que vous accrochiez un bloc à un mauvais label. Par la commande **Voir les liens nommés** du menu Edition, ExtendSim affiche les liens de bloc à bloc, sans tenir compte des labels. Cela vous permet de vérifier la justesse des liens.

Les liens nommés ne sont reconnus que dans un même niveau de hiérarchie. Il faudra prévoir un connecteur dans un bloc hiérarchique pour permettre un lien nommé allant du modèle général au sous-modèle hiérarchique. En revanche, les blocs Throw et Catch (bibliothèque Item) ou Throw Value et Catch Value (bib. Value) ne sont pas limités par la hiérarchie, et peuvent rendre le même service qu'un lien nommé.

Animation

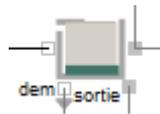
En regardant une courbe dans un graphique en fin de simulation, vous voyez comment les valeurs ont évolué dans le temps. Vous pouvez aussi laisser le graphique ouvert durant la simulation pour voir les valeurs évoluer. Mais parfois vous souhaiteriez en savoir plus sur des valeurs en progression. Pour cela, utilisez les possibilités d'animation d'ExtendSim.

De nombreux blocs des bibliothèques Value, Item et Rate comportent déjà des possibilités d'animation. Le texte d'aide de ces blocs dit s'ils sont animés ou non, et comment. Une bibliothèque supplémentaire, appelée **Animation**, permet d'ajouter des blocs simples d'animation dans le modèle.



Pour voir l'animation, cliquez sur le bouton d'animation, ou sélectionnez l'option **Animation** dans le menu **Simulation** (l'option doit être cochée) et lancez la simulation. L'animation n'est pas montrée par défaut parce que cela ralentit la simulation, et que vous n'en avez pas toujours besoin.

Le bloc Holding Tank est un bon exemple de bloc animé. Nous avons vu ce bloc sans animation. Lancez plusieurs fois le modèle avec animation. Vous voyez le niveau de la réserve s'élever et s'abaisser en fonction du remplissage. Lors de la première simulation, le niveau évolue entre 0 et une valeur maximale supposée ; dans les simulations suivantes, c'est une moyenne des maxima précédents qui est utilisée comme maximum.



bloc Holding Tank avec animation

Si vous avez demandé une animation et qu'elle va trop vite, vous pouvez la ralentir en faisant glisser le curseur vers la **Tortue** dans la barre de statut durant la simulation. La vitesse du ralentissement dépend du type de votre ordinateur. En allant vers le **Lièvre** vous accélerez l'animation.

Les blocs utilisés dans les modèles discrets permettent de représenter les éléments du flux par une icône que l'on voit se déplacer de bloc en bloc au fur et à mesure de sa progression dans le modèle, si l'animation est sélectionnée dans le menu Simulation. Par défaut, le symbole utilisé est un cercle vert, mais vous pouvez modifier les symboles, y compris vos propres images graphiques. Votre image doit être dans l'un des formats suivants : BMP, GIF, JPG/JPEG, PNG, PBM (Portable Bitmap), PGM (Portable Gray Map), PPM (Portable Pix Map), XBM (x11 Bitmap), XPM (x11 Pixmap) et placée dans le répertoire Extensions\Pictures. Les images sont prises en compte au moment où ExtendSim démarre.

- ☞ ExtendSim parcourt les images au niveau supérieur du répertoire Pictures et les ajoute à sa liste interne d'images. Elles sont accessibles à la fois par les fonctions Modl et les menus des symboles d'animation des blocs.

La hiérarchie

Un bloc hiérarchique contient d'autres blocs liés entre eux. Lorsque vous ouvrez un bloc hiérarchique, vous voyez un groupe de blocs (y compris d'autres blocs hiérarchiques), qui représente une portion du modèle, ou sous-système. Les blocs hiérarchiques permettent d'avoir des sous-systèmes dans l'objectif d'une modélisation descendante ou ascendante.

Il y a beaucoup d'emplois pour des blocs hiérarchiques :

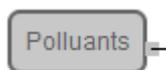
- Si vous avez un modèle complexe avec plusieurs dizaines de blocs, vous pourrez utiliser des blocs hiérarchiques pour simplifier le modèle en faisant des regroupements fonctionnels de blocs. Ces sous-ensembles peuvent ensuite être réutilisés dans d'autres modèles.
- Au lieu de montrer tous les détails d'une opération, vous pouvez représenter des étapes essentielles. Pour montrer l'intérieur d'une étape, il suffit de double-cliquer sur le bloc hiérarchique.
- En construisant un modèle, vous pouvez partir des hypothèses les plus simples jusqu'aux règles les plus fines en créant de nouveaux niveaux hiérarchiques. Cela vous aide à structurer votre pensée et facilite la compréhension de modèles alors même qu'ils sont de plus en plus complexes.

Un bloc hiérarchique possède certaines des caractéristiques d'un bloc et certaines des caractéristiques d'un modèle. En fonction de la manière dont il est enregistré, un bloc hiérarchique sera qualifié de "pur" ou de "physique". Un bloc hiérarchique pur est un bloc dans lequel toutes les instances de ce bloc sont identiques et toute modification effectuée serait répercutée dans toutes les instances du bloc : il est **enregistré dans une bibliothèque** et en cela ressemble à tout autre bloc. Un bloc hiérarchique physique est un bloc dans lequel une modification ne concerne que cette instance du bloc : il est **enregistré avec le modèle** et en cela n'est qu'une portion d'un modèle. Le statut du bloc hiérarchique provient de la manière dont vous l'enregistrez dans une bibliothèque et dont vous enregistrez des modifications ultérieures dans la bibliothèque.

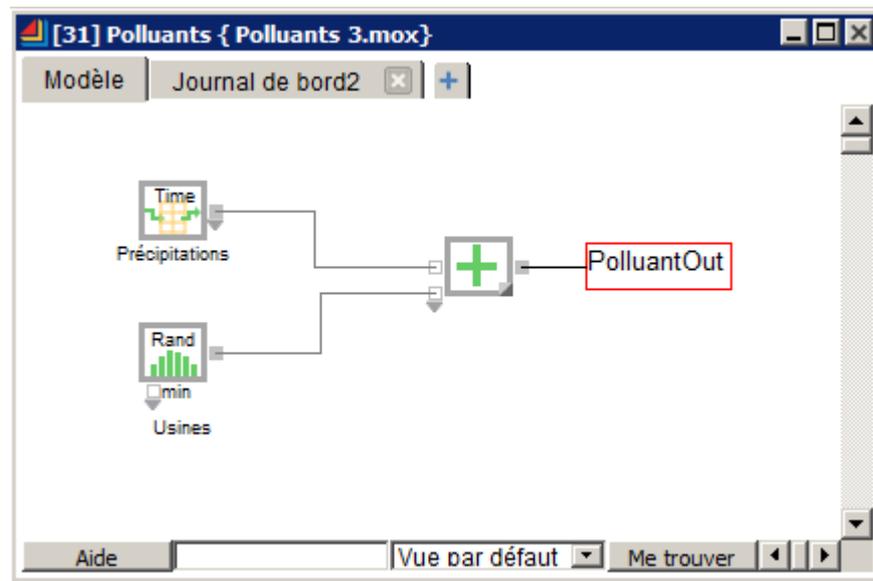
Tous les blocs hiérarchiques ont certains traits en commun :

- Vous pouvez les copier d'un modèle à l'autre ou d'un emplacement à l'autre, comme les autres blocs.
- Vous modifiez les paramètres d'un bloc contenu dans un bloc hiérarchique en double-cliquant sur le bloc une fois que vous avez ouvert la fenêtre hiérarchique.
- Une fenêtre hiérarchique est comme un modèle : vous ajoutez, supprimez, déplacez des blocs ; dessinez des objets, entrez du texte, clonez des éléments de dialogue, etc.
- Vous modifiez l'icône, les connecteurs ou le texte d'aide d'un bloc hiérarchique par Alt-Double-clic sur l'icône.
- À l'encontre des autres blocs, les blocs hiérarchiques sont enregistrés directement dans le modèle sous forme de copies. Cela leur permet d'être traités comme une copie d'une portion d'un modèle. Vous pouvez copier un bloc hiérarchique à un autre emplacement du modèle et modifier sa fenêtre sans affecter le bloc hiérarchique d'origine. C'est cela la hiérarchie physique.
- Vous pouvez aussi enregistrer un bloc hiérarchique dans une bibliothèque, auquel cas il est considéré comme un bloc. Toute modification du bloc affecte toutes les instances de ce bloc. C'est la hiérarchie pure.

Lorsque vous ouvrez un bloc hiérarchique par double-clic, au lieu de voir un dialogue, vous voyez une fenêtre avec le sous-modèle. Dans notre modèle, le bloc Polluant est un bloc hiérarchique :



le bloc Polluant



fenêtre de bloc hiérarchique

Les liens du bloc hiérarchique avec le reste du modèle sont représentés par des cadres de texte (liens nommés avec un cadre). Dans cet exemple, "PolluantOut" est le connecteur de sortie du bloc, comme vous pouvez le voir d'après le texte à la droite de la fenêtre hiérarchique. Les deux nombres figurant dans le dialogue d'un "sous-bloc" sont le numéro de bloc global et le numéro de bloc local



. Le bouton « Me trouver » sert dans les modèles volumineux où il n'est pas évident de localiser chaque bloc dans l'interface.

- ☞ Il n'est possible d'utiliser des liens nommés qu'à l'intérieur d'un même niveau hiérarchique. Pour relier deux niveaux, ou deux blocs hiérarchiques entre eux, utilisez les blocs Throw/Catch (existent pour signaux continus comme pour signaux discrets).

Il y a deux manières de créer un bloc hiérarchique :

1. Sélectionnez quelques blocs dans le modèle et choisissez l'option **Hiérarchiser la sélection** dans le menu Modèle.
2. Créez un nouveau bloc hiérarchique par l'option **Nouveau bloc hiérarchique** du menu Modèle.

Si vous utilisez les blocs hiérarchiques pour réduire la complexité d'un modèle existant, vous utiliserez surtout la première méthode. Si vous construisez un nouveau modèle, vous incluez souvent dès le départ une hiérarchie.

Rendre une sélection hiérarchique

Pour que plusieurs blocs deviennent un bloc hiérarchique :

- Sélectionnez les blocs ainsi que des objets graphiques associés en traçant une marquée de sélection ou en cliquant sur les blocs avec la touche Maj enfoncée.
- Sélectionnez Hiérarchiser la sélection dans le menu Modèle. Vous voyez :



dialogue Hiérarchiser la sélection

- Entrez un nom pour le bloc et cliquez sur le bouton Créer un bloc H.

Notez que cette action ne peut être annulée. Si vous joignez plus de blocs que nécessaire, vous devrez supprimer les blocs dans le sous-modèle et les replacer dans le modèle de niveau supérieur.

Par l'option *Hiérarchiser la sélection*, ExtendSim fait les liens à votre place et remplace les blocs

hiérarchisés par le nouveau bloc hiérarchique. Vous pouvez tout de suite double-cliquer sur l'icône et regarder le sous-modèle.

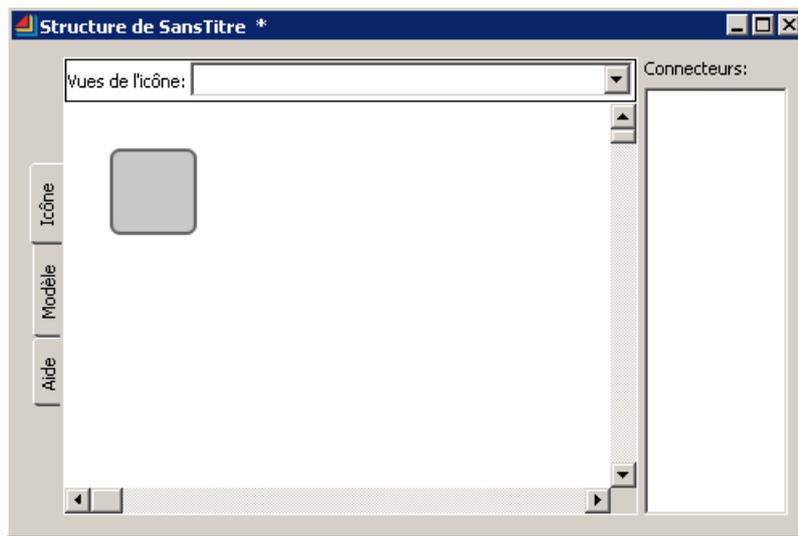
Vous pouvez modifier l'aspect du bloc en déplaçant des connecteurs ou en ajoutant du dessin.

Notez que lorsque vous hiérarchisez une sélection, le bloc hiérarchique est enregistré uniquement dans le modèle. Pour l'enregistrer dans une bibliothèque, vous devez utiliser la commande Enregistrer bloc sous, décrite plus bas.

Construire un nouveau bloc hiérarchique

Si vous créez un bloc hiérarchique par l'option *Nouveau bloc hiérarchique* du menu Modèle, ExtendSim vous demande un nom puis ouvre une fenêtre de structure.

Voici la fenêtre de structure d'un nouveau bloc hiérarchique :



fenêtre de structure d'un nouveau bloc hiérarchique

Chaque onglet rend accessible une zone de travail. Affichée par défaut l'onglet Icône sert pour dessiner l'icône et placer les connecteurs. Un onglet permet de saisir un éventuel texte d'aide pour ce bloc. L'onglet Modèle affiche le sous-modèle : c'est ce que vous voyez lorsque vous double-cliquez sur un bloc hiérarchique.

Voici les étapes pour construire un bloc hiérarchique :

- Construisez le sous-modèle
- Ajoutez des connecteurs à l'icône
- Enregistrez le bloc hiérarchique. Vous pouvez l'enregistrer dans une bibliothèque si vous le souhaitez.
- Reliez le bloc hiérarchique aux autres blocs du modèle.

Construire le sous-modèle

Vous procédez comme pour construire un modèle normal. Si vous préférez, vous pouvez utiliser un Copier/Coller pour copier une portion d'un modèle existant. Vous pouvez aussi utiliser les outils pour ajouter du texte ou des éléments graphiques.

Ajouter les connecteurs

Le bloc hiérarchique doit être lié à l'extérieur par des connecteurs. Certains blocs hiérarchiques ont des connecteurs d'entrée et de sortie, certains n'ont qu'un seul type de connecteur. Vous pouvez mélanger des connecteurs discrets et continus.

S'ils ne sont pas visibles, faites apparaître les outils d'icônes par le menu Outils:



connecteurs des outils d'icône de la fenêtre de structure

Les six premières icônes ajoutent des connecteurs : discret, continu, flux, universel, matrice et utilisateur (respectivement). Les connecteurs utilisateur en losange sont utilisés par les programmeurs. Les options suivantes concernent les connecteurs variables. Le dernier outil s'emploie pour ajouter

des objets d'animation.

L'ajout d'un connecteur se fait en quatre temps :

1. Décidez de la nature du connecteur (le plus souvent discret ou continu). Il doit correspondre au type de l'entrée ou de la sortie principale du sous-modèle.
2. Ajoutez le connecteur en cliquant sur l'outil connecteur puis sur la portion icône à l'emplacement voulu. Le connecteur est créé, il entre dans la liste des connecteurs et ajoute un objet texte connecteur dans la portion modèle. Si vous vous êtes trompé de type de connecteur, sélectionnez-le puis cliquez sur le bon outil connecteur : le nouveau type est automatiquement adopté.

La portion icône comporte une grille magnétique. Pour ne pas dépendre de cette grille, appuyez sur la touche Alt lorsque vous placez le connecteur.

3. Nommez le connecteur. Chaque connecteur a un nom unique dont le suffixe détermine s'il s'agit d'un connecteur d'entrée ou de sortie ("In" ou "Out"). Initialement, un connecteur est un connecteur d'entrée. Pour qu'il devienne un connecteur de sortie, modifiez son nom de sorte qu'il se termine par "Out".

Pour cela, sélectionnez le nom du connecteur dans la portion connecteurs. Entrez un nouveau nom ou modifiez l'ancien, puis appuyez sur Entrée ou cliquez ailleurs pour valider le nouveau nom.

4. Une fois que le connecteur est ajouté à l'icône, vous devez dans la portion modèle lier l'objet connecteur texte aux blocs appropriés. Vous procédez pour cela comme avec les liens nommés : tracez le lien depuis le connecteur vers le bord du texte. Lorsque la ligne devient bleue, le lien est fait.

Enregistrer un bloc hiérarchique

Pour enregistrer un bloc hiérarchique, il y a deux méthodes :

- cliquez sur la **case de fermeture** pour enregistrer le bloc hiérarchique avec le modèle, mais pas dans une bibliothèque.
- choisissez **Enregistrer bloc sous** dans le menu Fichier pour enregistrer le bloc dans une bibliothèque.

Vous pouvez savoir si le bloc hiérarchique est enregistré dans une bibliothèque en le sélectionnant et en choisissant Propriétés dans le menu Fichier.

Enregistrement hors des bibliothèques

Il n'est pas nécessaire qu'un bloc hiérarchique figure dans une bibliothèque. Vous pouvez malgré tout le copier à d'autres emplacements du modèle, et même dans d'autres modèles. Chaque instance du bloc est unique et évolue de manière autonome.

Enregistrement dans une bibliothèque

La version enregistrée dans une bibliothèque reflète l'état du bloc au moment de l'enregistrement. Si vous modifiez le bloc, certains changements ne seront pas reflétés par le bloc source dans la bibliothèque. Par exemple, si vous modifiez le sous-modèle, le bloc de la bibliothèque ne répercutera pas automatiquement les modifications.

Pour enregistrer un bloc hiérarchique dans une bibliothèque :

- Placez-vous devant la fenêtre de structure du bloc.
- Choisissez **Enregistrer le bloc H dans une bib...** dans le menu Fichier. Dans le dialogue, sélectionnez une bibliothèque et installez le bloc dans cette bibliothèque. Puis fermez le bloc hiérarchique. Le nom du bloc figure en italiques dans la liste des blocs d'une bibliothèque.

La répercussion des modifications est examinée plus loin.

Relier un bloc hiérarchique au modèle

Pour relier un bloc terminé au modèle, fermez la fenêtre de structure si elle est ouverte. Puis reliez les connecteurs du bloc hiérarchique aux autres blocs du modèle, comme d'habitude.

Attention : il n'est possible d'utiliser de lien nommé que dans un même niveau hiérarchique. Ainsi, un lien nommé au niveau supérieur du modèle ne peut pas communiquer avec un lien nommé situé dans un bloc hiérarchique. Il faudrait créer un connecteur supplémentaire au bloc hiérarchique, ou utiliser les blocs Throw/Catch.

Options complémentaires sur la hiérarchie

Dans un bloc hiérarchique, vous ajoutez du texte, des images ou des éléments graphiques comme dans n'importe quel modèle. Vous pouvez aussi :

- Cloner des éléments de dialogue
- Modifier l'icône
- Renommer le bloc
- Ajouter une aide
- Ajouter de l'animation

Le clonage d'éléments se fait par la portion du sous-modèle, à laquelle vous accédez par un double-clic sur le bloc hiérarchique. Pour changer d'icône, renommer le bloc ou ajouter de l'animation, vous passez par la fenêtre hiérarchique, à laquelle vous accédez par Alt+double-clic sur le bloc ou par le menu du bouton droit.

Cloner des éléments de dialogue dans la portion de sous-modèle

Vous pouvez cloner des éléments de dialogue directement dans la portion de sous-modèle. Cela permet de voir ces éléments dès que vous double-cliquez sur l'icône du bloc.

Changer l'icône

Vous souhaitez sans doute modifier l'icône de vos blocs hiérarchiques, dans la mesure où ExtendSim vous propose un simple rectangle en guise d'icône. Pour cela :

- Ouvrez la fenêtre de structure par Alt-Double-clic sur l'icône du bloc.
- Cliquez sur l'icône dans l'onglet icône, et modifiez-la ou supprimez-la (mais veillez à ne pas supprimer les connecteurs).
- Utilisez les outils graphiques pour dessiner l'icône. Vous pouvez aussi coller des images conçues avec d'autres logiciels.

Il est fondamental de ne pas détruire les connecteurs. Si vous essayez de le faire, ExtendSim vous avertit. Si cela vous arrive pourtant, vous pouvez annuler l'action ou ajouter un nouveau connecteur, mais il vous faudra vérifier que le lien avec le modèle est correct.

Renommer le bloc

Lorsque la fenêtre de structure d'un bloc hiérarchique est ouverte, vous pouvez renommer le bloc par l'option *Renommer le bloc* du menu Développeur.

Ajouter une aide

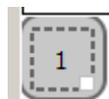
Vous éditez le texte d'aide directement dans la portion correspondante de la fenêtre de structure. Les outils Texte vous permettent de modifier les styles des caractères.

Ajouter de l'animation

Vous pouvez ajouter de l'animation aux blocs hiérarchiques en utilisant les blocs Animate Item et Animate Value de la bibliothèque d'animation (Animation.lbr). Pour utiliser l'un de ces blocs dans un bloc hiérarchique :

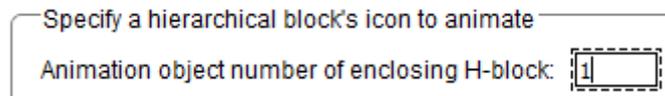
- Ajoutez une zone d'animation dans l'onglet icône de la fenêtre de structure (vous pouvez avoir plusieurs zones d'animation si vous le souhaitez).
- Reliez le bloc Animate Value ou Animate Item à la sortie du bloc que vous voulez animer dans le sous-modèle. Avec le bloc Animate Item, vous devez aussi lier la sortie de ce bloc à l'entrée d'un autre bloc.
- Spécifiez le numéro de zone d'animation dans le dialogue du bloc.

Des zones d'animation sont des zones rectangulaires dans une icône. Pour inclure une zone d'animation, cliquez sur l'outil Objet d'animation dans la barre d'outils, , puis dessinez un rectangle dans l'onglet icône. La zone d'animation reçoit un numéro, en commençant par 1. Par exemple, une icône de bloc hiérarchique comportant une zone d'animation ressemblerait à :



ajout d'une zone d'animation

Reliez le bloc d'animation à la zone que vous voulez animer. Par exemple, pour animer le résultat d'une multiplication, vous pouvez utiliser le bloc Animate Value. Entrez dans le dialogue le numéro de zone d'animation que le bloc doit contrôler. Ainsi pour animer la zone numéro 1, entrez :



spécification de la zone d'animation n°1

Modifier des blocs hiérarchiques

Il y a trois façons de modifier des blocs hiérarchiques :

- Vous modifiez des paramètres des blocs contenus dans un bloc hiérarchique en double-cliquant sur les dialogues.
- Vous modifiez le sous-modèle comme n'importe quel autre modèle : ajout de blocs, de texte, etc. C'est la méthode conseillée si vous voulez modifier le sous-modèle uniquement pour cette instance du bloc hiérarchique.
- Vous modifiez l'icône, les connecteurs et l'aide dans la fenêtre de structure. Vous pouvez aussi modifier la portion sous-modèle.

Pour ouvrir la structure d'un bloc hiérarchique (comme pour tout bloc), faites Alt-Double-clic.

Si votre bloc hiérarchique est dans une bibliothèque, vous ne pouvez pas modifier sa structure en double-cliquant sur son icône dans la fenêtre de bibliothèque. Vous devez travailler directement sur le modèle et utiliser un bloc hiérarchique copié depuis le menu Bibliothèque ou la fenêtre de bibliothèque.

La modification de blocs hiérarchiques donne différents résultats, selon la façon dont vous avez modifié la fenêtre hiérarchique ou la fenêtre de structure, et selon que le bloc est enregistré dans une bibliothèque :

- Si vous modifiez le sous-modèle d'un bloc dans la fenêtre hiérarchique, les modifications ne s'appliquent qu'à ce bloc. Cela équivaut à modifier les paramètres d'un bloc standard : les modifications n'affectent que cette instance du bloc et elles sont enregistrées avec le modèle. C'est vrai y compris pour des blocs hiérarchiques qui avaient originalement été enregistrés dans des bibliothèques.
- Si vous modifiez la structure d'un bloc hiérarchique qui n'est pas enregistré dans une bibliothèque, les modifications ne s'appliquent qu'à ce bloc. Par exemple, vous pouvez avoir plusieurs copies d'un bloc hiérarchique dans un modèle, mais lorsque vous modifiez une des copies, les autres restent telles quelles.
- Si vous modifiez la structure d'un bloc hiérarchique qui est enregistré dans une bibliothèque, vous pouvez choisir la manière dont les modifications seront prises en compte :
 - seulement dans cette instance du bloc
 - également dans le bloc mère dans la bibliothèque (n'affectera que les nouveaux blocs placés dans le modèle depuis la bibliothèque)
 - également dans toutes les instances du bloc pour tous les modèles ouverts (hiérarchie pure)

Par exemple, lorsque vous fermez la fenêtre de structure d'un bloc hiérarchique issu d'une bibliothèque, ExtendSim affiche le dialogue suivant :



fermeture de la fenêtre de structure

Choisissez "de ce bloc" pour enregistrer les modifications seulement dans ce bloc, ou "Aussi dans la bibliothèque" pour répercuter les modifications dans la bibliothèque contenant ce bloc, ou "Aussi les blocs des modèles ouverts" pour affecter tous les blocs issus de cette bibliothèque figurant dans des modèles ouverts (hiérarchie pure).

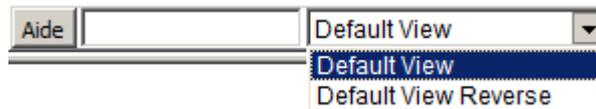
- ☞ Le bloc Constant de la bibliothèque Value a une option *Retain constant if updated from hierarchy* qui, si elle est cochée, fait que le bloc Constant conserve sa valeur même lorsque le bloc H qui le contient est mis à jour dans sa bibliothèque. Les différentes instances du bloc ne sont alors pas modifiées à l'intérieur de blocs H. Cela sert fort utilement lorsque chaque bloc hiérarchique doit être distingué des autres par un identifiant, lequel ne doit pas être affecté si le bloc H est mis à jour dans sa bibliothèque.
- ☞ Les modifications faites à la structure d'un bloc hiérarchique enregistré dans une bibliothèque ne seront pas reflétées dans les modèles qui n'étaient **PAS** ouverts au moment où la modification a eu lieu. Pour qu'elles soient prises en compte, vous devez ouvrir le modèle et remplacer le bloc H par celui de la bibliothèque.

Différentes visualisations d'un modèle

ExtendSim propose plusieurs manières de visualiser les blocs d'un modèle.

Vues de l'icône

Chaque bloc peut avoir plusieurs vues, ou icônes différentes. C'est le cas pour la plupart des blocs discrets, qui ont une vue "inverse", destinée aux représentations de droite à gauche. Les vues sont sélectionnées dans un menu déroulant figurant dans le dialogue du bloc à côté du label, ou par clic du bouton droit sur un bloc :



Menu des vues dans le dialogue

ExtendSim facilite la création des vues et des familles, lorsque vous travaillez dans la structure du bloc.

Masquer ou voir les connecteurs et les liens

Dans le menu **Modèle**, vous pouvez demander à masquer les connecteurs. Le modèle est épuré, parfois rendu plus lisible pour les néophytes.

Vous pouvez encore aller plus loin dans la simplification et afficher le modèle sans liens (menu **Modèle**).

Interactions sur un modèle

En plus de fonctionnalités évidemment très importantes pour comprendre et faire comprendre un modèle, telles l'animation, la hiérarchie ou le journal de bord, ExtendSim propose plusieurs manières d'aider à l'interaction avec le modèle. Nous en citons quelques-unes qui pourront vous être utiles.

Verrouiller le modèle

La commande *Verrouiller modèle* du menu **Modèle** permet d'éviter toute modification d'un modèle autre que le changement de valeurs dans des dialogues. La commande masque aussi la plupart des outils dans la frise d'outils, de sorte que l'utilisateur ne puisse modifier des liens, des aspects graphiques, etc. En verrouillant un modèle, vous pouvez sans risque le confier à une personne qui connaît mal ExtendSim et qui pourrait par erreur déplacer ou supprimer des blocs. Pour déverrouiller le modèle, choisissez à nouveau la commande *Verrouiller modèle*.

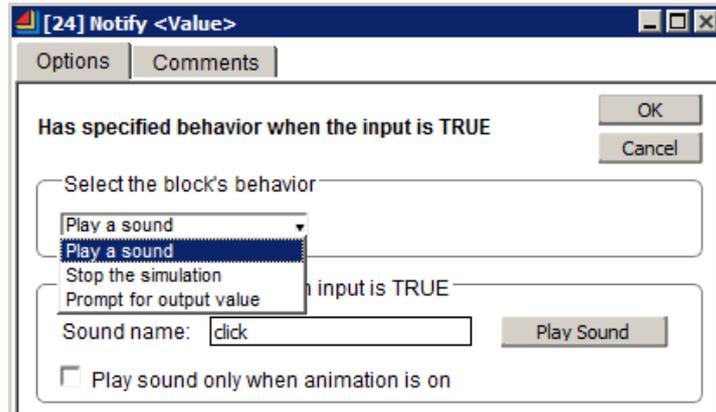
Envoyer des messages à l'utilisateur

Le bloc **Notify** est pratique pour communiquer avec l'utilisateur du modèle, l'impliquer dans le déroulement de la simulation et lui signaler des valeurs notables.

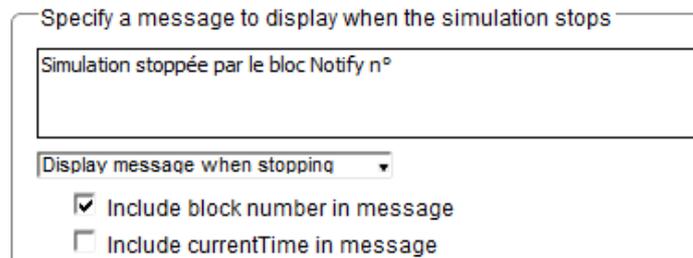
Le bloc propose trois comportements, qui réagissent à un signal VRAI :

Jouer un son mesure un certain paramètre dans le modèle et émet un son lorsque la valeur franchit le seuil critique. Vous entrez le nom du son dans le dialogue :

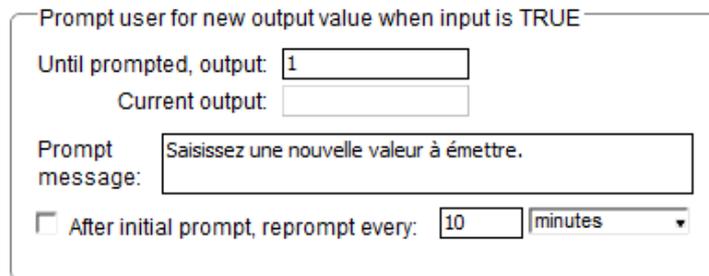
Vous pouvez entrer le nom d'un son du système Windows ou figurant dans le répertoire Extensions d'ExtendSim. Un fichier son doit être en format .WAV. Pour voir la liste des sons, ouvrez le module Son depuis le Panneau de configuration.

*mode Son*

Stop the simulation permet de stopper la simulation et d'alerter l'utilisateur. Le bloc peut alors afficher n'importe quel message jusqu'à 255 caractères (ou éventuellement aucun message). Voici le dialogue du bloc :

*mode Stop*

Prompt for output value (Demander une valeur à émettre) avertit l'utilisateur, et surtout lui demande une action, par exemple de donner une valeur. La valeur critique est fournie dans le dialogue ou calculée dynamiquement en fonction de conditions du modèle. Le dialogue du bloc est :

*mode Demander une valeur*

Utilisez par exemple ce bloc pour effectuer une pause dans la simulation, demander une valeur à l'utilisateur, puis reprendre la simulation en tenant compte de cette valeur.

Lancer des simulations

Le dialogue Paramètres

Pour tout modèle vous devez spécifier le cadre temporel dans lequel il sera étudié. La commande Paramètres du menu Simulation contrôle la manière dont se déroule votre simulation. Voici ce dialogue *Paramètres de la simulation* et ses différents onglets :

Onglet Paramètres

Onglet *Paramètres* -(dialogue *Paramètres*)

Les options sont :

Temps de fin	L'heure à laquelle se termine la simulation. Ce temps moins le temps de début constitue la durée simulée.
Temps de début	L'heure à laquelle démarre la simulation. Généralement c'est 0. Vous pouvez lui donner une autre valeur si votre modèle utilise la valeur du temps dans des calculs.
Réplications	Le nombre de fois que doit s'exécuter la simulation. Les réplications sont numérotées dans la barre d'état en commençant par zéro.
Unité de temps globale	L'unité de temps pour tout le modèle. Les unités de temps locales sont définies dans les dialogues des blocs concernés.
Dates calendaires	Permet que le modèle obéisse à un vrai calendrier, dont vous spécifiez la date de début. L'unité de temps doit être autre que <i>Générique</i> . L'option <i>Format Européen</i> place le jour avant le mois. L'option <i>Date système Macintosh</i> ne s'applique que dans l'environnement Macintosh.
Calendrier personnalisé	Si <i>Dates calendaires</i> n'est pas sélectionné, vous pouvez définir ici la composition de chacune des unités de temps. Ces définitions sont utilisées par ExtendSim pour convertir automatiquement les unités de temps locales en unité de temps globale.

Toutes les simulations se déroulent durant un certain temps et se déroulent selon un temps soit **continu**,

soit **discret**. ExtendSim détermine la durée de la simulation en fonction des valeurs de début et de fin entrées dans le dialogue. Du point de vue de la gestion du temps, si le modèle contient le bloc Executive, c'est un modèle discret ; sinon c'est un modèle continu.

Unités de temps

Dans les modèles discrets, ExtendSim progresse du début à la fin de la simulation par une série d'événements. On ignore à l'avance combien il y aura d'événements, en général.

Lorsque vous créez un modèle, vous pouvez utiliser deux sortes d'unités de temps.

- Une unité de temps générique, et alors il faut veiller à utiliser la même partout dans le modèle. Comme il n'y a qu'une seule unité utilisée dans ce dialogue, toutes vos données temps doivent être conformes à cette unité.
- Une unité de temps spécifique globale, qui permet d'utiliser dans les différents blocs du modèle l'unité locale de votre choix.

Les nouveaux modèles ont par défaut une unité de temps générique. Même si votre modèle utilise partout la même unité de temps, cela peut aider à sa compréhension que de choisir une unité spécifique. Si vous choisissez une unité dans le dialogue Paramètres de la simulation, cette unité devient proposée par défaut par tous les blocs du modèle.

Dates calendaires

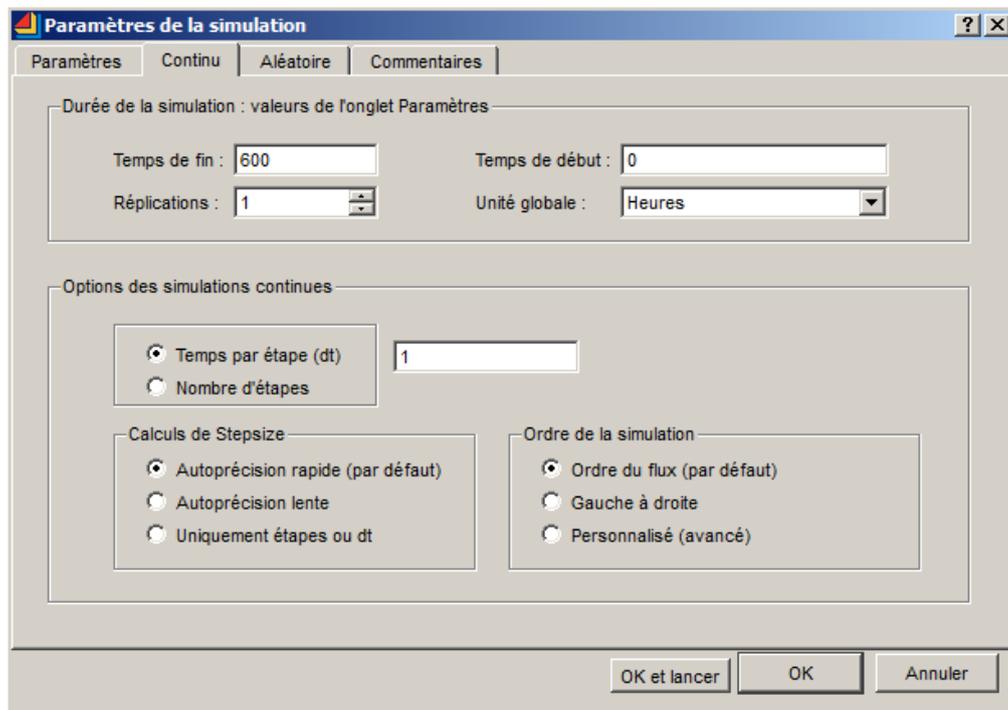
Les dates calendaires ne sont pas disponibles si l'unité de temps globale est générique ou Milliseconde, (et aussi Mois ou Ans pour un modèle Rate). Si l'option est sélectionnée, les blocs Rate ne pourront plus choisir le mois ou l'année pour unité de temps locale.

Le choix des dates calendaires rendra visible une option similaire dans les blocs Create, History, Shift, Lookup Table, certains blocs de la bib. Chart, et les tables de BDD.

Exemple discret

Supposons que vous ayez une simulation discrète dont le temps de début est 0, le temps de fin 480, et dont l'unité de temps est la minute, comme le modèle de lavage de voitures. Comme il s'agit d'un modèle discret, la durée de l'étape et le nombre d'étapes sont ignorés. Les données vont être calculées durant 480 minutes ou huit heures, le dernier événement aura lieu un peu avant ou juste au moment du temps de fin.

Onglet Continu



Onglet **Continu** (dialogue Paramètres)

Dans les modèles continus, la durée de simulation est divisée en étapes, où le temps de début est la première étape, et le temps de fin la dernière étape. La durée de chaque étape est connue sous le nom de

delta time, ou dt. Le delta time détermine le nombre de fois où le modèle est recalculé.

Les options sont :

Temps de fin, Temps de début, Réplifications, Unité de tps globale	Les valeurs sont reprises de l'onglet Paramètres discrets, par commodité. Modifier un onglet revient à modifier l'autre pour ces valeurs.
Temps par étape (dt)	Le delta time, ou durée d'une étape, ou "granularité" de la simulation. ExtendSim utilise ce paramètre pour déterminer le temps passé pour chaque étape dans les simulations continues. Une valeur par défaut du nombre d'étapes est automatiquement calculée en fonction de ce paramètre, selon la formule : $\text{floor}((\text{EndTime}-\text{StartTime})/\text{DeltaTime} + 1,5)$ Vous pouvez le voir en sélectionnant l'option "Durée de l'étape (dt)" après avoir entré un nombre.
Nombre d'étapes	Une autre manière de déterminer la "granularité" de la simulation. Généralement ce nombre sera la durée totale de la simulation, et sous-entendra que chaque étape durera une unité de temps. Une valeur par défaut de delta time est automatiquement calculée en fonction de ce paramètre, selon la formule : $(\text{EndTime}-\text{StartTime})/(\text{NumSteps} - 1)$ Vous pouvez le voir en sélectionnant l'option "Durée de l'étape (dt)" après avoir entré un nombre.
Calculs de StepSize	Ces paramètres ne sont employés que dans les modèles continus qui modifient la variable système "deltaTime", tels les modèles électroniques. Généralement, vous utilisez toujours <i>Auto-Précision rapide</i> . Si votre modèle modifie <i>deltaTime</i> et requiert plus de précision, utilisez <i>Auto-Précision lente</i> (qui divise par 5 la valeur calculée pour <i>deltaTime</i>). Si vos blocs personnalisés modifient <i>DeltaTime</i> mais que vous souhaitez ignorer ces modifications, utilisez <i>Uniquement étapes ou dt</i> .
Ordre de simulation	Utilisez généralement <i>Ordre du flux</i> . Voyez le Guide de prise en main d'ExtendSim continu pour plus de précisions.

Exemple continu

Supposons que vous ayez une simulation continue dont le temps de début est 0, le temps de fin 36, la durée de l'étape (dt) est 1 et l'unité de temps est le mois, comme dans le modèle Reservoir. Les données seront calculées pour chacune des 37 étapes, depuis l'étape 0 jusqu'à l'étape 36. Chaque calcul dans une étape couvre un mois, la période qui commence à cette étape et se poursuit jusqu'à l'étape suivante exclusivement. Ainsi le modèle calcule 36 mois d'activité. Si vous aviez spécifié un temps de début de 1 et un temps de fin de 37, la durée aurait aussi été de 36 mois.

Avec ce même exemple, si vous spécifiez un *deltaTime* de 0,5, la simulation se déroulera du temps 0 au temps 36 en calculant des données pour 79 étapes d'une durée de 15 jours chacune.

Si vous spécifiez un *deltaTime* qui n'est pas le résultat de la durée divisée en segments égaux, ExtendSim va ajuster le temps de fin à une valeur supérieure. Cela permet de garantir que tous les segments sont d'une durée égale, et que la simulation se terminera au temps de fin annoncé. Vous pouvez aussi sélectionner un nouveau *deltaTime* ou un nouveau temps de fin. Dans notre exemple où le temps de fin est 36, vous pouvez choisir une durée d'étape de 2 mois. Il en résulte un modèle qui calcule un mois sur deux, depuis le temps 0 au temps 36, pour un total de 18 étapes.

Généralement, vous choisirez un *deltaTime* de 1, c'est à dire une unité de temps par étape.

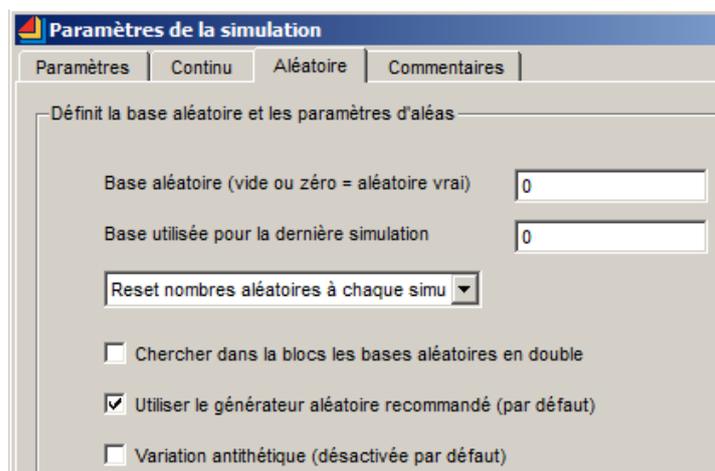
Déterminer le nombre correct d'étapes d'un modèle continu

Les modèles discrets, parce qu'ils sont mus par des événements, ignorent le nombre d'étapes et la durée des étapes.

Les modèles de traitement du signal doivent avoir leur propre durée d'étape spécifiée dans le dialogue Paramètres, ou la calculer avec leurs blocs. Par exemple, les blocs de Filtre du modèle Noisy FM System calculent la durée de l'étape en fonction des paramètres entrés dans le dialogue.

Les modèles d'équations différentielles (modèles avec des intégrateurs dans des boucles rétroactives, tels que vous en trouverez dans le répertoire Continuous) doivent avoir une durée d'étape explicitée. Si vous n'êtes pas sûr de savoir si le nombre d'étapes choisi est assez grand ou si la taille d'une étape (le nombre d'étapes divisé par la durée) est assez petite, exécutez la simulation avec vos meilleures valeurs, puis relancez-la avec deux fois plus d'étapes. Comparez les résultats pour voir s'il y a des changements significatifs.

Onglet Aléatoire



Onglet **Aléatoire** (dialogue Paramètres)

Les options sont :

Base aléatoire (vide ou zéro = aléatoire vrai)	La "base" ou "seed" du générateur de nombres aléatoires. Une valeur 0 ou vide produit une base réellement aléatoire ; toute autre valeur est le numéro d'une séquence reproductible de nombres pseudo-aléatoires. Notez que chaque bloc qui génère des séquences aléatoires utilise sa propre séquence.
Base utilisée pour la dernière simulation	Indique la dernière base aléatoire utilisée. Pour répéter la dernière séquence aléatoire produite, recopiez ce nombre dans la zone au-dessus.
<ul style="list-style-type: none"> Reset nombres aléatoires à chaque simu 	Avec cette option les nombres aléatoires sont initialisés au début de chaque simulation. Si la base aléatoire est vide ou à 0, un nouveau tirage aléatoire a lieu à chaque réplification. Si la base aléatoire est un entier positif, le même tirage aléatoire sera utilisé à chaque réplification.
<ul style="list-style-type: none"> Continuer la séquence aléatoire 	Avec cette option, la séquence de bases aléatoires pour chaque réplification dans une série se poursuit à la suite de la séquence précédente.
<ul style="list-style-type: none"> Utiliser les valeurs de la table <code>_Seed</code> 	Avec cette option, la base aléatoire utilisée pour chaque réplification dans une série est lue dans la table <code>__Seed</code> de la BDD <code>ExtendSim</code> sélectionnée sur la droite. Vous choisissez ainsi explicitement chaque base pour chaque réplification. La table doit avoir autant d'enregistrements que de réplifications prévues dans la série.
Chercher dans les blocs les bases aléatoires en double	En début de simulation, tous les blocs utilisant des nombres aléatoires sont vérifiés pour garantir qu'ils n'utilisent pas la même base aléatoire. Non nécessaire si vous n'avez pas explicitement demandé à ce que les blocs aient leur propre base aléatoire.

Utiliser le générateur aléatoire recommandé	Nous conseillons vivement d'utiliser le générateur aléatoire par défaut. Si l'option n'est pas cochée, c'est le générateur « Schrage » qui sera employé, ce qui était le cas dans de très anciennes versions d'ExtendSim.
Variation antithétique	Génère une variation antithétique destinée à limiter la variance lors de multiples répliquions. Se contrôle par exemple dans le bloc Optimizer, ou dans des blocs programmés. Ne pas cocher a priori.

Nombres aléatoires

Plusieurs blocs ExtendSim font appel à des nombres aléatoires, par exemple le bloc Random Number de la bibliothèque Value, ou le bloc Create de la bibliothèque Item. La fonction d'analyse de sensibilité permet aussi d'utiliser des valeurs aléatoires.

Les fonctions aléatoires utilisées par ExtendSim produisent des nombres pseudo-aléatoires qui sont calculés par un algorithme reproductible par l'un des deux types de générateurs aléatoires.

Le générateur aléatoire par défaut, celui recommandé, s'appuie sur Lewis, Goddman, Miller et Dwyer, "A Portable and Reasonable Fast Minimum Standard Random Number Generator" ACM Transactions on Mathematical Software, vol.33, 1990.

Un autre générateur aléatoire est fondé sur Schrage, "A More Portable Fortran Random Number Generator", ACM Transactions on Mathematical Software, vol.5, 1979. Ce second générateur est utilisé surtout pour la rétro-compatibilité avec d'anciens modèles ExtendSim.

Vous pouvez entrer une base pour le générateur aléatoire dans le dialogue Paramètres de simulation ou dans le dialogue Sensibilité. Une Base de 0 (ou vide) utilisera un aléatoire "vrai" à chaque fois différent ; toute autre valeur provoquera une séquence reproductible de valeurs pseudo-aléatoires. Chaque bloc qui émet des valeurs aléatoires génère sa propre séquence indépendante. Une base spécifique à chaque bloc peut être donnée par l'option « Use block seed : » dans son dialogue.

Si vous avez des données empiriques à utiliser dans vos modèles pour représenter des distributions aléatoires, des utilitaires associés à ExtendSim pourront vous aider à déterminer quelle est la loi de distribution qui se rapproche le plus de vos données ("fitting"). Contactez-nous pour des renseignements complémentaires sur ces outils.

Lancer la simulation

Le plus souvent dans ExtendSim vous exécutez la simulation sur un modèle. C'est l'option par défaut, et toute la puissance de calcul est dédiée au modèle en cours.

Mails il est aussi possible d'exécuter plusieurs modèles en même temps, grâce aux capacités de multi-threading d'ExtendSim : un thread (ou fil d'exécution) est alors utilisé pour l'interface utilisateur, et les autres threads traitent les simulations en cours.

Le choix du mode d'exécution se fait par l'icône du bonhomme, et le bouton « Lancer la simulation » s'adapte en fonction :



Exécution normale



multi-threading

Choix de l'unité de temps

Il existe deux manières de gérer les unités de temps dans ExtendSim :

- Utiliser une unité de temps générique implicite dans tout le modèle et ses blocs. Vous êtes alors responsable de la cohérence des valeurs temporelles que vous saisissez.
- Utiliser une unité de temps choisie dans le dialogue Paramètres du menu Simulation : cette unité vous est alors proposée par défaut dans tous les blocs concernés, mais vous pouvez aussi choisir localement des unités plus adaptées. Par exemple un modèle dont l'unité de temps soit l'heure, mais dont certaines opérations soient spécifiées en minutes et d'autres en secondes.

La durée et le nombre de simulations (réplications)

Quelle période allez-vous simuler ? Vos choix pour le Temps de début et le Temps de fin dépendent de quatre facteurs :

- Est-ce que le système est "terminatif" (comporte une fin naturelle) ou "non terminatif" (n'a pas de fin évidente).
- La période intéressante modélisée.
- Vos objectifs de modélisation (estimer des performances, examiner des alternatives, etc.)
- Comment ont été obtenues vos données statistiques.

Terminatif ou non ?

Certains systèmes ont un temps de fin évident. Par exemple, beaucoup de services atteignent un stade où les activités se terminent. Si vous modélisez un magasin où les clients sont admis durant 8 heures par jour, vous ferez naturellement une simulation sur huit heures.

Ces systèmes atteignent rarement un état stable durable, et vous y recherchez plus des tendances que des moyennes statistiques sur des longues périodes de temps. Ainsi dans notre magasin, il est plus intéressant de repérer les pics et les creux d'activité que de calculer des moyennes.

Dans la mesure où les conditions initiales de ces modèles ont un impact sur les résultats, il est important qu'elles soient très représentatives de la réalité. La simulation se fera souvent sur des périodes courtes en utilisant des bases aléatoires différentes à chaque réplication, pour conforter votre confiance dans les résultats obtenus.

Un système "non terminatif" atteint au bout d'un certain temps un état stable et ses résultats ne changent plus guère. C'est le cas de certains systèmes de production qui fabriquent en continu, ou de salles d'urgence, ou de centraux téléphoniques.

Un système peut observer des horaires de travail et rester "non terminatif", comme par exemple un atelier qui fonctionne sur deux postes par jour : si la seconde équipe reprend là où la première avait terminé, la période intermédiaire sera juste ignorée pour la simulation.

Ce qui importe dans ces systèmes, c'est d'éliminer les erreurs possibles dues à la période initiale de montée en régime, celle qui précède le "régime stable", où les indicateurs ne sont pas représentatifs de l'état réel du système. Pour éviter tout problème, vous pouvez attendre la fin de la montée en régime et demander une mise à zéro statistique par le bloc **Clear Statistics** (bib. Value).

Déterminer la durée et le nombre des simulations

Dans un système "terminatif", la durée se détermine naturellement et facilement. Pour une banque, ce sera les 8 heures de sa journée d'ouverture.

En théorie, un modèle "non terminatif" peut être simulé indéfiniment, mais on l'arrêtera lorsque les résultats témoigneront d'un état bien stabilisé. Dans un modèle de production, vous simulerez une période assez longue pour être sûr que tous les types d'événements se sont produits plusieurs fois chacun.

Le nombre des simulations (réplications) dépend de votre objectif de modélisation et de l'impact de données aléatoires dans votre modèle. Moins de réplications sont nécessaires pour comparer des alternatives que pour déterminer une "meilleure performance possible".

Accélérer l'exécution

Plus votre modèle devient complexe, plus vous devenez sensible à la vitesse d'exécution. La durée dépend de la complexité (niveau de détail) du modèle et de la longueur de la période simulée. Tout ce qui demande à ExtendSim un travail d'écriture à l'écran (alors que sa fonction de base est du calcul) a une influence directe sur la vitesse d'exécution. Quelques conseils :

Diminuez l'affichage des données

- N'utilisez l'animation que lorsque vous en avez besoin (généralement pour les premières exécutions du modèle), car elle peut ralentir beaucoup la simulation.
- Ne gardez pas les graphiques ouverts durant la simulation (voir les outils des graphiques pour déterminer quand la fenêtre du graphique doit s'ouvrir), ainsi que les blocs hiérarchiques.
- Fermez les dialogues restés ouverts (vous en voyez la liste dans le menu Fenêtre).
- Limitez le nombre de clones qui se mettent à jour souvent.

Traquez les paramètres inefficaces

- Avec les modèles continus, choisissez au plus juste le pas de calcul = `deltaTime` (si une étape représentait 1 heure, essayez par exemple 1 étape = 2 heures et voyez si le modèle ainsi adapté vous donne des résultats pertinents).
- `ExtendSim` propose deux manières de tracer le profil des blocs dans un modèle :
 - Placez un bloc *RealTimer* (bib Utilitaires) tout à fait à droite du modèle. Dans son onglet *Block Profile*, activez le profil et choisissez un emplacement où stocker le rapport. Après la simulation, vous trouverez une table de BDD contenant des informations sur chaque bloc : nom, numéro, label, temps d'exécution, pourcentage de temps d'exécution, mémoire allouée au bloc en début et en fin de simulation. Cela vous permet de repérer les blocs « gourmands » en temps d'exécution.
 - Lancez la commande Simulation>Débogage du modèle>Profil des blocs avant de lancer la simulation : un fichier texte est généré, contenant des informations sur chaque bloc : nom, numéro, label, temps d'exécution, pourcentage de temps d'exécution. Cela vous permet de repérer les blocs « gourmands » en temps d'exécution.
- Si un bloc est lié à une source de données dans une matrice globale ou une BDD, il recevra un message à chaque fois que la source de données change. Essayez d'adapter l'activation de ces liens (idéalement en début de simulation).
- N'utilisez le bloc *Display Value* de la bibliothèque Value qu'à bon escient, car il ralentit aussi la simulation si vous n'adaptez pas les paramètres de son dialogue. Quand c'est possible, essayez de remplacer plusieurs blocs continus de calcul par un bloc Equation.
- Avec les modèles discrets, ajustez les paramètres d'allocation des entités dans le bloc Executive (onglet Control). La procédure d'allocation de mémoire pour de nouvelles entités, si elle a lieu souvent, peut faire perdre du temps. Estimez le nombre d'entités qu'utilise le modèle et allouez-les initialement. N'en allouez pas trop non plus, car vous surchargeriez la mémoire ! En fin de modèle, supprimez bien les entités par des blocs Exit.

Autres facteurs pouvant avoir un impact

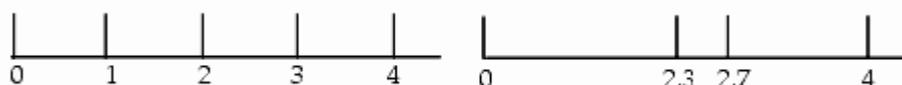
- Augmentez la mémoire (RAM) de votre ordinateur, afin de l'empêcher d'utiliser de la mémoire virtuelle, plus lente.
- Avec les modèles discrets, essayez de réduire le nombre d'entités manipulées. Si vous avez utilisé une entité par objet fabriqué dans l'atelier, voyez si vous ne pouvez pas faire un regroupement pour qu'une entité représente deux objets. Regardez aussi du côté de la bib. Rate.
- Lorsque vous utilisez des blocs de type Equation, limitez autant que possible les comparaisons de chaînes de caractères.

Bibliothèques de base : modèles continus et discrets

Cette rubrique explique le contenu des blocs des bibliothèques Value et Item et comment les employer. Pour chaque bibliothèque, un modèle illustratif de difficulté croissante a été examiné dans le Guide de prise en main correspondant. On se reportera utilement au chapitre [Techniques de modélisation](#).

Continu et discret

Les modèles continus (parfois appelé process) servent à décrire des flux de valeurs insécables homogènes, tandis que les modèles discrets suivent des individus uniques appelés "entités". Dans les deux types de simulations, ce qui nous intéresse, ce sont les changements d'état du modèle, mais on pourrait pour cela représenter le temps de deux manières différentes :



Ligne du temps continu

Ligne du temps discret

Les simulations continues sont comme un fluide s'écoulant dans un tuyau. Le volume peut augmenter ou diminuer, mais le flux est continu. Les changements dans le temps se mesurent par incréments égaux. L'état du modèle est observé à un moment particulier, puis par étapes, régulièrement.

Avec les simulations discrètes, le tuyau peut être vide, ou faire passer un certain nombre de bidons d'eau. Le simple fait que le temps passe ne change rien au modèle : ce sont des événements (arrivée de bidons, présence ou absence d'un bidon) qui font les changements d'état, à des moments rarement réguliers. Le temps avance par sauts d'un événement à un autre.

La bibliothèque Value

Les blocs de la bibliothèque Value sont utiles à la fois pour les modèles continus et pour l'exécution de tâches simples dans tous les autres modèles. L'utilisation de la bibliothèque Value permet de modéliser de nombreux phénomènes sans devoir programmer avec ModL. Lorsque vous concevrez des modèles complexes comportant un foisonnement de blocs génériques, vous apprendrez peut-être à utiliser la hiérarchie, l'éditeur d'équation ou le langage ModL afin de combiner en un seul bloc les fonctions de plusieurs blocs génériques.

Utiliser la bibliothèque Value

Tous les blocs de la bibliothèque Value (Value.lbr) ont un dialogue associé. Vous utilisez le dialogue pour entrer des valeurs ou pour examiner des valeurs courantes. Si vous avez lié une valeur en entrée et aussi entré une valeur dans le dialogue, c'est toujours la valeur du connecteur qui prévaut sur celle du dialogue. Le dialogue de chaque bloc comporte un bouton Aide qui fournit une explication sur le bloc, ses connecteurs, ses paramètres et ses options.

Les icônes des blocs ont une couleur de base grise.

Les blocs de la bibliothèque Value par fonction

Nous avons regroupé les blocs par leur type, comme indiqué dans le menu hiérarchique de la bibliothèque : accès aux données (Data Access), entrées (Inputs), mathématiques (Math), orientation (Routing), retards (Holding), sorties (Outputs) et statistiques. Chaque élément de la liste indique le nom du bloc, sa fonction et des exemples d'emplois possibles dans un modèle.

Note importante : le détail du fonctionnement de chaque bloc et de ses connecteurs figure dans le texte d'aide de chaque bloc ; référez-vous y pour toute information complémentaire.

Accès aux données

Bloc	Fonction	Exemple
Data Source Create	Crée, visualise et modifie des matrices globales ou des fichiers texte.	Structuration des données

Write	Ecrit des données dans un fichier texte, un tableur, une base de données, une matrice globale.	Paramètres de contrôle ou de résultat pour plusieurs blocs d'un modèle.
Import Export	Importe et exporte des données depuis des sources externes dans des matrices globales et BDD ExtendSim.	Echanges avec applications
Data Init	Initialise (et parfois définit) des matrices globales et des tables de bases de données.	Initialisation
Read	Lit des données à partir d'un fichier texte, d'un tableur, d'une base de données, d'une matrice globale.	Paramètres pour plusieurs blocs d'un modèle.
Data Specs	Affiche un rapport sur des données sélectionnées dans une base de données ou une matrice globale.	Présentation des résultats

Entrés

Bloc	Fonction	Exemple
Constant	Génère une valeur constante à chaque étape.	Un flux régulier de fluides, de valeurs.
Pulse	Émet des impulsions périodiques (valeurs VRAI) séparées par un certain délai.	Signaux périodiques
Random Number	Génère des nombres aléatoires entiers ou réels.	Entrées nécessitant une distribution statistique.
Simulation Variable	Permet d'utiliser une variable système dans la simulation.	Pour mesurer quand interviennent les événements.

Mathématiques

Bloc	Fonction	Exemple
Decision	Compare la valeur de A à la valeur de B.	Règles de décision
Equation	Permet d'entrer des équations	Règles de décision complexes
Integrate	Intègre l'entrée en fonction du temps.	Equations différentielles.
Math	Propose diverses fonctions arithmétiques, logiques, trigonométriques ou financières	
Max & Min	Détermine le maximum et le minimum à partir des connecteurs d'entrée.	Choix d'une plus grande priorité, redirection.
Lookup Table	Contient une table de valeurs qui calcule une sortie pour chaque valeur en entrée, ou en fonction du temps.	Table de correspondance de prix/unité ou client/serveur.
Time Unit	Convertit des valeurs d'une unité en une autre.	

Orientation

Bloc	Fonction	Exemple
Select Value In	Sélectionne la sortie à partir des deux entrées en fonction de la valeur du connecteur T.	Variation des sources d'information.
Select Value Out	Passe la donnée en entrée à l'une des deux sorties en fonction de la valeur du connecteur T.	Redirection, catégories.
Throw Value	Transmet sans liaison une valeur à un bloc Catch Value	.
Catch Value	Reçoit sans lien une valeur provenant d'un bloc Throw Value	

Retards

Bloc	Fonction	Exemple
Wait Time	Retient l'entrée un certain temps avant de la passer en sortie.	Délai, temps de construction.
Holding Tank	Accumule le total des valeurs entrées et sort une quantité demandée.	Réservoir, file d'attente, compte en banque.

Sorties

Bloc	Fonction	Exemple
Notify	Demande une valeur de sortie si l'entrée est en-deçà ou au-delà d'un seuil critique.	Pause dans la simulation pour demander de l'information.
Command	Envoie une commande à exécuter dans un tableur.	Commande
Display Value	Affiche et transmet la valeur du connecteur d'entrée à chaque étape de la simulation.	

Statistiques

Bloc	Fonction	Exemple
Clear Statistics	Remet à zéro les statistiques de divers blocs dans le modèle.	Montée en régime.
Mean & Variance	Calcule la moyenne, la variance et l'écart-type des valeurs reçues durant la simulation.	
Statistics	Enregistre des informations sur certaines catégories de blocs des bibliothèques Value, Item, Rate.	

Analyse

Bloc	Fonction	Exemple
Optimizer	Ce bloc particulier sert à rechercher des solutions optimales par convergence. Voir les paragraphes spécifiques qui lui sont consacrés dans le chapitre Analyse	
Scenario Manager	Permet la description et l'optimisation de scénarios différents à appliquer à un modèle. Voir les paragraphes spécifiques qui lui sont consacrés dans le chapitre Analyse .	

La bibliothèque Item

La bibliothèque Item réunit des blocs correspondant à des activités, des opérations et des ressources telles qu'on les trouve dans de nombreux environnements. Tout ce qui concerne la gestion des événements liés au système est intégré aux blocs eux-mêmes.

La modélisation discrète fait partie intégrante des démarches Six Sigma, de restructuration d'entreprise, d'analyse de risque, de planification de capacités, d'analyse de débit et d'étude de fiabilité. Les modèles discrets sont également utilisés pour étudier les effets de certains changements tels que les pénuries de main d'œuvre, l'acquisition d'équipements ou les pannes de transmission. Ils fournissent aux entreprises une perspective interfonctionnelle sur leurs procédés fondamentaux et les amènent à se demander « pourquoi ? » et « que se passerait-il si ? ».

Types de systèmes concernés

Bien que les entreprises soient souvent examinées en termes d'unités fonctionnelles (par exemple la

comptabilité, la production, les ressources humaines...), elles sont en fait composées de process. Nous appelons process une série de tâches logiquement liées que l'on entreprend avec un objectif spécifié, généralement un produit ou un service. Chaque process est une combinaison d'éléments (des personnes, des procédures, de la matière, des équipements, de l'information, de l'espace et de l'énergie) impliqués dans une activité devant atteindre un certain but.

Les process représentent la manière dont l'entreprise travaille effectivement. Il s'agit souvent de cycles complets, tels que la sortie d'un nouveau produit (R&D et marketing), les flux de fabrication (achats, réception et transformation) ou un cycle de produit (arrivée des commandes, livraison et facturation)

Dans l'entreprise, les process sont organisés autour d'événements, par exemple la réception d'une commande, une demande d'achat ou l'idée d'un nouveau produit. Les événements arrivent de manière aléatoire, mais souvent à des cadences que l'on connaît à peu près. Ces événements peuvent être économiques ou non-économiques. Ce sont de toute façon eux qui déclenchent les mouvements de l'entreprise.

Nous allons pouvoir représenter des process par l'utilisation et le flux d'éléments (appelés entités dans la terminologie ExtendSim), pilotés par des événements. Voici certains exemples de process classiques, avec les événements qui les motivent et les entités qui interviennent dans le process :

Process	Evénement	Entités
Développement d'un nouveau produit	Nouvelle idée de produit	Le produit
Fabrication d'un produit	Réception de matière première	Matière première, Opérateurs
Salle d'urgence d'hôpital	Arrivée de patients	patients, personnel médical
Traitement d'un cas d'accident par une compagnie d'assurance	Réception de la demande	Demande
Réseau informatique	Transmission d'un paquet	Paquets de données
Fourniture d'assistance technique	Appel d'un client	Appels téléphoniques
Création d'un plan marketing	Introduction du nouveau produit	Plan marketing

Utiliser la simulation

L'objectif de toute entreprise ou institution est désormais de développer sa compétitivité. La réduction des coûts et l'amélioration de la qualité ne suffisent plus pour gagner des parts de marché. La rapidité à produire, à proposer de nouveaux produits, à réagir au contexte, sont de nouvelles clés de la compétitivité.

Pour cela, les entreprises doivent modifier les process existants dans l'optique de :

- éliminer les étapes non essentielles, celles qui n'apportent aucune valeur ajoutée.
- implémenter des technologies lorsque c'est utile.
- améliorer le déroulement des opérations pour mettre l'accent sur les fonctions ayant une valeur ajoutée.
- fournir des outils de mesure pour mieux analyser et planifier les process.

La simulation propose des outils de mesure qui aident à analyser les process et imaginer leur évolution.

Utiliser la bibliothèque Item

Les simulations discrètes passent des entités de bloc en bloc durant la simulation. Ces entités sont souvent générées selon une loi de distribution aléatoire, ou selon un planning des événements qui doivent se produire. Ces entités peuvent avoir attributs et priorités, ce qui permet une meilleure ressemblance avec des pièces, des tâches ou des clients réels. Les entités sont traitées par des activités et le temps de traitement dépend souvent de la disponibilité des ressources.

La plupart des blocs de la bibliothèque Item ont des connecteurs discrets et des connecteurs continus. Les connecteurs discrets servent à transmettre au connecteur suivant une entité et toutes les

informations qui la qualifient. Les connecteurs continus, ainsi que les paramètres de dialogue, donnent des informations sur les propriétés de l'entité et sur ses effets sur le modèle.

L'objet de la simulation est généralement de déterminer l'emplacement de goulets d'étranglement durant le traitement et de voir quelles parties du système pourraient être améliorées. Chaque branche du flux d'entités alimente un autre bloc ou se termine dans un bloc Exit, qui permet de compter les entités qui sortent du modèle.

Un modèle peut comporter à la fois des blocs continus et des blocs discrets. Si vous intégrez des blocs discrets à un modèle, celui-ci sera alors considéré comme étant un modèle discret et vous devrez donc lui ajouter le bloc Executive (bibliothèque Item).

L'objet des simulations discrètes est généralement de déterminer l'emplacement de goulets d'étranglement durant le traitement et de voir quelles parties du système pourraient être améliorées. Chaque branche du flux d'entités alimente un autre bloc ou se termine dans un bloc Exit, qui permet de compter les entités qui sortent du modèle.

Tous les blocs de la bibliothèque Item ont un dialogue associé, qui sert à entrer des paramètres ou à visualiser des valeurs qui évoluent durant la simulation. Si vous avez lié un connecteur d'entrée et en même temps donné la valeur correspondante dans le dialogue, c'est la valeur du connecteur qui surpasse celle du dialogue. Le dialogue de chaque bloc comporte un bouton Aide qui indique comment utiliser le bloc, ses connecteurs et ses paramètres.

Les icônes des blocs sont de couleur verte, avec des motifs significatifs.

La composition d'un modèle

Vous pouvez placer les blocs n'importe où dans un modèle, sachant qu'ils sont évalués dans l'ordre de leur flux. La seule exception à cette règle est : le bloc Executive (nécessaire dans toute simulation discrète) qu'il faut placer à la gauche de tous les autres blocs.

Entités

L'unité de base qui est passée d'un bloc à un autre est une **entité** (en anglais **item**). Elle représente un élément du système modélisé. Chaque entité a un cycle de vie : elle est d'abord créée, puis transformée et enfin détruite. Son état change (elle se déplace, est retardée ou ses propriétés peuvent être modifiées) lorsqu'un événement se produit, comme par exemple l'assemblage d'une pièce ou l'arrivée d'un client. Dans les modèles de production, ces entités représenteront des pièces dans une chaîne de production ; dans des modèles de réseaux, une entité représentera un paquet d'information. Les entités passent d'un bloc à un autre par des connecteurs discrets. Le bloc Create peut générer des entités selon une distribution aléatoire, à intervalles réguliers, selon un planning ou sur demande. Le bloc Resource Item fournit une quantité déterminée d'entités.

Événements

ExtendSim ne fait passer des entités de bloc en bloc que lorsqu'a lieu un événement. Les événements sont contrôlés par le bloc Executive et n'interviennent que lorsque certains blocs le demandent. Lorsque vous démarrez une simulation, un seul événement a lieu.

Les blocs qui ne génèrent pas d'événement permettent que les blocs après eux demandent les entités durant un seul événement. Ainsi, une entité peut passer par plusieurs blocs en un seul événement si les blocs ne la bloquent pas.

Notez que chaque événement permet potentiellement à chaque bloc de faire passer des entités. Ainsi, un événement peut causer l'avancée de plusieurs entités indépendantes.

Connecteurs discrets

La plupart des blocs discrets font passer par leurs connecteurs un ensemble d'informations appelé élément discret, item ou **entité**. Cet ensemble transporte des informations sur les entités, leurs attributs, leur priorité et leur quantité. C'est une différence d'avec les blocs continus qui ne passent que des valeurs simples. Le connecteur d'entrée discret est un carré vert  et le connecteur de sortie discret est un carré vert plein . Il existe un connecteur d'entrée mixte, , qui accepte des entrées discrètes ou continues.

Lorsque vous combinez des blocs d'autres bibliothèques avec des blocs discrets, assurez-vous de ne relier que des connecteurs compatibles. Les connecteurs des blocs génériques ne peuvent être reliés qu'à des connecteurs continus ou au connecteur mixte. On ne peut les relier à des connecteurs discrets.

Propriétés

Les propriétés sont un élément important dans les simulations discrètes. Ce sont des caractéristiques qu'une entité transporte avec elle lors de sa progression dans le modèle. Il y a quatre propriétés : les attributs utilisateur, la priorité, la quantité, les attributs système.

Attributs

Au premier rang parmi les propriétés, l'**attribut** consiste en un nom et une valeur (numérique ou chaîne). Par exemple, les attributs peuvent être employés pour diriger une entité, pour indiquer des temps d'opération ou des qualités durant le traitement ("couleur=rouge", "Tps = 23"). Voyez les blocs Set, Get et Activity entre autres pour des informations sur la manipulation des attributs. Vous utilisez ces blocs pour créer et contrôler les attributs durant la simulation.

Un type d'attribut spécial est l'attribut DB Address (Adresse BDD) : il contient une valeur qui représente un emplacement dans une base de données. Cette adresse se compose de quatre nombres, chacun étant un index pour la BDD, la table, le champ et l'enregistrement. Ensemble ils pointent sur une information spécifique dans la BDD (des adresses incomplètes sont autorisées). L'adresse est décodée par les blocs Get, Read(I) et Write(I).

Priorité

La priorité permet de spécifier l'importance d'une entité. Par exemple, à une étape de la production, il se peut qu'un employé examine tous les travaux en attente et choisisse le plus urgent. La plus grande priorité a la plus basse valeur de priorité (-1 passe avant 2). Le bloc Set affecte une priorité à une entité. Le bloc Queue possède une logique Priorité qui fait sortir l'entité de plus haute priorité d'abord. Vous visualisez la priorité d'une entité par les blocs Get, History et Information.

Quantité

Chaque entité peut être une entité unique ou un groupe de clones (doublons). Si la quantité d'une entité est 1, elle représente un élément ; si elle est autre que 1, elle représente un groupe. Les entités ont par défaut une quantité de 1, sauf si vous la modifiez par les blocs Set ou Create. Généralement, vos entités auront une quantité de 1.

Les entités ayant une quantité autre que 1 sont traitées différemment en fonction du connecteur d'entrée qui les lit. Elles resteront unies, et seront traitées comme une unique entité, tant qu'elles n'atteindront pas un bloc Exit, une file d'attente, une mise en lot ou une ressource, ou un connecteur mixte.

Vous souhaitez souvent que la quantité d'une entité soit autre que 1. Par exemple, vous pouvez souhaiter modéliser un changement d'équipe constituée de 5 ouvriers qui partent en même temps. Ou encore votre modèle peut recevoir une caisse de 300 écrous destinés à l'atelier.

Modifier des blocs de la bibliothèque

Les blocs de la bibliothèque Item utilisent des structures de données et des méthodes de programmation spéciales. Si vous êtes programmeur et que vous souhaitez ajouter des blocs à la bibliothèque Item, utilisez les blocs *MYO* de la bibliothèque ModL Tips pour modèle. Son script est commenté en détail pour vous permettre de comprendre les traitements effectués. Lisez attentivement le guide de programmation avant de modifier des blocs discrets.

Les blocs de la bibliothèque Item par fonction

Les blocs sont regroupés ici par fonction ou type : Data access (accès aux données), Activities (activités), Queues (files d'attente), information, Batching (lots d'entités), Routing (orientation), Properties (propriétés) et ressources. Chaque élément de la liste indique le nom du bloc, son emploi et des exemples d'utilisation.

Note importante : le détail du fonctionnement de chaque bloc et de ses connecteurs figure dans le texte d'aide de chaque bloc ; référez-vous y pour toute information complémentaire.

Accès aux données

Bloc	Fonction
Write(I)	Ecrit des données dans la BDD à l'arrivée d'une entité
Read(I)	Lit des données dans la BDD à l'arrivée d'une entité

Activités

Bloc	Fonction
Activity	Retient une ou plusieurs entités et les fait sortir en fonction de leur délai et de leur temps d'arrivée
Convey Item	Convoyeur qui déplace des entités d'un point à un autre.
Workstation	Représente une station de travail pouvant faire attendre et traiter des entités.
Transport	Transporte des entités d'un point à un autre en fonction d'informations de vitesse et distance.

Files d'attente

Bloc	Fonction
Queue	File d'attente pouvant adopter plusieurs logiques pour faire sortir les entités.
Queue Matching	File d'attente qui fait sortir les entités seulement si elles possèdent un attribut donné. Sert à assembler ou apparier un type d'entité avec un autre.
Queue Equation	File d'attente qui libère des entités en fonction du résultat d'une équation. spécifiée.

Information

Bloc	Fonction
Cost By Item	Observe le coût accumulé par chaque entité qui passe.
History	Affiche des informations sur les entités ou les valeurs passant par le bloc.
Information	Affiche l'information sur les entités qui passent par lui.
Cost Stats	Observe les coûts calculés par tous les blocs concernés par les coûts.

Lots d'entités

Bloc	Fonction
Unbatch	Génère plusieurs entités à partir d'une seule, selon les spécifications du dialogue.
Batch	Réunit en une seule entité des entités venant de plusieurs sources.

Orientation

Bloc	Fonction
Throw	Envoie une entité à un bloc Catch sans lien ni connecteur.
Create	Fournit des entités à une simulation discrète à certains temps d'arrivée.
Catch	Reçoit, sans lien ni connecteur, une entité provenant d'un bloc Throw.
Gate	Forme de sas, qui ne laisse passer une entité dans une section que lorsque la précédente a quitté cette section.
Select Item In	Sélectionne les entrées selon une décision.
Select Item Out	Sélectionne les sorties selon une décision.
Exit	Fait sortir des entités de la simulation.

Propriétés

Bloc	Fonction
Set	Donne des propriétés aux entités.
Equation(I)	Calcule une équation lorsqu'une entité passe par le bloc.
Get	Affiche et/ou enlève des propriétés sur des entités.

Ressources

Bloc	Fonction
Shift	Pilote les horaires de travail de différentes ressources ou activités.
Resource Pool Release	Libère des ressources en augmentant le nombre d'éléments disponibles dans un pool.
Shutdown	Génère des arrêts pour les blocs Activity et Valve.
Resource Pool	"Stock" de ressources devant être utilisées par des blocs Queue.
Resource Item	Similaire à une file, mais sous forme d'une ressource pour laquelle on peut spécifier une quantité initiale.
Resource Manager	Gestionnaires de ressources complexes, utilisé dans le cadre du gestion spéciale des ressources.

Le bloc **Executive** permet que la simulation se termine selon un temps de fin ou après un nombre d'événements. Ce bloc est le noyau d'une simulation discrète et doit figurer à la gauche des autres.

Bibliothèques Utilities et Report

Les blocs de ces deux bibliothèques ne servent pas pour décrire le fonctionnement d'un système, mais pourront fournir de très utiles compléments pour l'interface des modèles, leur mise au point, la gestion des ressources système, les statistiques, etc. Les blocs peuvent en général s'employer aussi bien dans des modèles discrets que continus. Certains blocs sont présentés tout au long de ce manuel en fonction des besoins, mais nous en rappelons la liste complète ici.

Utilitaires

Bloc	Type	Fonction
Buttons	Contrôles	Permet de paramétrer des boutons servant d'interface utilisateur pour déclencher des actions.
Slider		Manière de visualiser des valeurs entre deux bornes
Feedback		Contrôle l'ordre de calcul en feedback dans les modèles continus.
Model Interface		Aide pour agencer une interface utilisateur
Popups		Permet de paramétrer des menus pour les choix de l'utilisateur.
Pause Sim		Contrôles des arrêts pendant le déroulement de la simulation.
Meter		Manière de visualiser des valeurs entre deux bornes
Model Compare	Informations	Compare deux états d'un modèle.
Count Blocks		Compte les blocs d'un modèle.
Block Info		Indique toutes les propriétés d'un bloc.
Memory Usage		Indique la taille prise par les divers blocs et éléments.
Fitting	Math	Permet de faire du fitting statistique.
Link Alert	Outils discrets	Alerte en cas de modification d'une BDD liée.
Switch		Interrupteur booléen utilisable dans un modèle.
Item Messages		Affiche les messages échangés entre deux blocs discrets.
Queue Tools		Permet d'initialiser et d'observer des files d'attente.

Stop message		Permet de bloquer des échanges entre deux blocs.
Event Monitor		Affiche le calendrier des événements d'une simulation
Run Model	Outils développeur	Permet de paramétrer le lancement de la simulation
Record message		Affiche les messages entre deux blocs continus.
Object Mapper		Donne des informations sur les contrôles ActiveX et les objets inclus (en anglais)
Find & Replace		Effectue des recherches/remplacements sur des éléments de dialogue.
RealTimer	Temps	Chronomètre la durée d'une simulation
TimeSync		Synchronise un modèle pour qu'il s'exécute en temps réel.

Report

Bloc	Fonction
Item Log Manager	Centralise dans une BDD des infos sur les entités.
Statistics	Recueille des statistiques sur des blocs sélectionnés.
Statistics BDD	Recueille des statistiques sur des portions de BDD sélectionnées.
Cost Stats	Enregistre les informations de coût.
Report Manager	Centralise dans une BDD des infos sur les entités, événements, et blocs.

Données et échanges de données

Ce chapitre examine les diverses structures et manières de stocker des données dans ExtendSim, ainsi que divers blocs ou techniques pour échanges des données avec d'autres applications.

Les bases de données ExtendSim

Une base de données est un emplacement centralisé et structuré pour stocker des données. La fonctionnalité ExtendSim « graphical simulation database » (GSDB) permet de créer des bases de données relationnelles internes pour stocker, gérer et afficher les données d'un modèle. Les bases de données ExtendSim constituent une interface idéale entre des modèles et des applications externes, telles que des tableurs ou d'autres bases de données.

Chaque modèle peut avoir plusieurs bases de données associées, et chaque base de données peut contenir plusieurs tables de données. Chaque table comporte des champs qui représentent des colonnes de données, des enregistrements qui représentent des lignes et des cellules qui sont à l'intersection d'un champ et d'un enregistrement.

Selon la convention standard sur les bases de données, les bases de données ExtendSim sont organisées en champs et enregistrements, tandis que les tables de données, les matrices et les tableurs externes sont organisés en lignes et colonnes.

Les bases de données ExtendSim sont stockées avec le modèle. Elles s'ouvrent automatiquement lorsque le modèle s'ouvre et sont enregistrées automatiquement en même temps que le modèle.

Par des blocs spécifiques ou par l'interface de liaison de données, vous pouvez associer des paramètres du modèle à une base de données sans devoir écrire de code.

Cette section est divisée en plusieurs points :

- Avantages à utiliser une base de données interne.
- Manières de créer une base de données ExtendSim et méthodes d'échange de données.
- Comment créer une base de données ExtendSim.
- Etablir une relation parent/enfant entre deux champs.
- Gérer des bases de données, tables et champs.
- Dialogues et menu déroulants.
- Utiliser un Add-In Excel pour les bases de données ExtendSim.

Avantages à utiliser une base de données interne

Utiliser une base de données interne présente plusieurs avantages :

- Séparer les données du modèle pour une meilleure gestion du projet et des expériences.
- Organiser l'information de manière logique, via une base de données ou plusieurs bases de données.
- Pouvoir manipuler les données par type, emplacement, composants, ou toute autre caractéristique.
- Permettre une centralisation des données utilisées partout dans le modèle.
- Utiliser les tables de base de données pour interface d'entrée et de sortie du modèle.
- Réutiliser des données d'un modèle à un autre.
- Faciliter l'import et l'export de données avec d'autres applications.

Recourir à une base de données est souvent indispensable pour les gros modèles, mais l'interface permet d'y faire appel même pour de petits modèles.

Créer une base de données

Il y a trois manières de créer une base de données ExtendSim:

- 1) Via l'interface utilisateur, en utilisant des commandes de menus pour créer une nouvelle base de données, ou en important une un fichier texte de base de données depuis une BDD ExtendSim ou MS Excel.
- 2) Par des fonctions de bases de données dans un bloc discret ou continu faisant appel à une équation. Voyez la section sur les [blocs d'équation](#).
- 3) En programmant du code ModL.

L'interaction avec les bases de données se fera de plusieurs manières également :

- 1) Via l'interface utilisateur, par liens dynamiques entre éléments de dialogue et bases de données. Voir les liens avec les structures de données internes.
- 2) Par les blocs Read et Write (bib. Value) ou Read(I) and Write(I) (bib. Item) pour l'échange de données entre un modèle et une base de données.
- 3) Par des fonctions de bases de données dans un bloc discret ou continu faisant appel à une équation.
- 4) En programmant du code ModL.

Comment créer une base de données ExtendSim

L'exemple suivant montre comment utiliser des commandes de menus pour créer une base de données pour le modèle Reservoir présenté dans le Tutorial. Evidemment une base de données n'est pas requise pour le modèle Reservoir, mais notre objet est de savoir créer une base de données interne.

Voici les étapes à suivre :

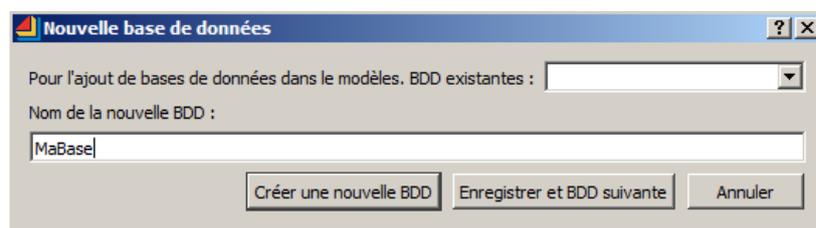
- 1) Ouvrez un modèle existant
- 2) Créez une nouvelle base de données pour ce modèle.
- 3) Ajoutez des tables et des champs à la base de données.
- 4) Ajoutez des enregistrements.
- 5) Si la base de données doit servir pour des valeurs en entrée du modèle : saisissez, copiez ou importez des valeurs pour les cellules de la base.

Commencez par ouvrir le modèle Reservoir 1 dans le répertoire \Exemples\Tutorials. Afin de ne pas écraser le fichier d'origine, enregistrez le modèle sous Reservoir BDD.

Démarrer une nouvelle base de données

- Avec la fenêtre de modèle active, choisissez la commande Base de données > Nouvelle base de données.

Le dialogue Nouvelle BDD s'ouvre, permettant de nommer la base de données.

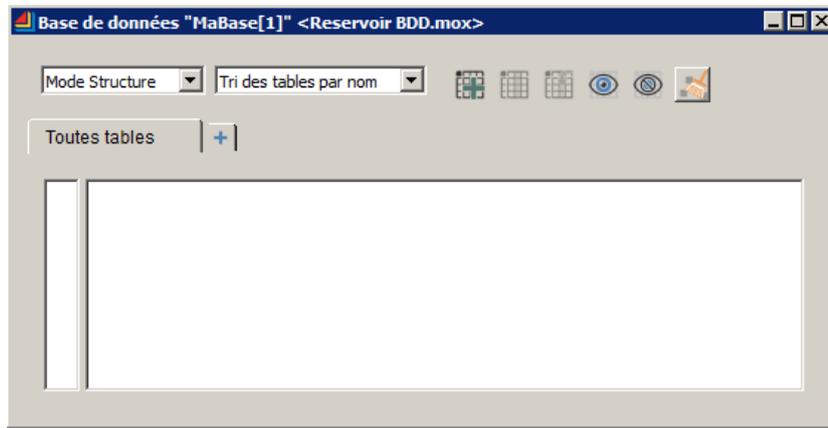


- Nommez la nouvelle base de données *MaBase*.

Le nom ne doit pas dépasser 63 caractères, ni être utilisé par une autre base de données. Il peut contenir des espaces, et ne différencie pas majuscules et minuscules.

- Cliquez sur Créer une nouvelle BDD pour fermer le dialogue et enregistrer la nouvelle base de données.

La fenêtre de base de données s'ouvre en mode Structure, comme montré ci-dessous.



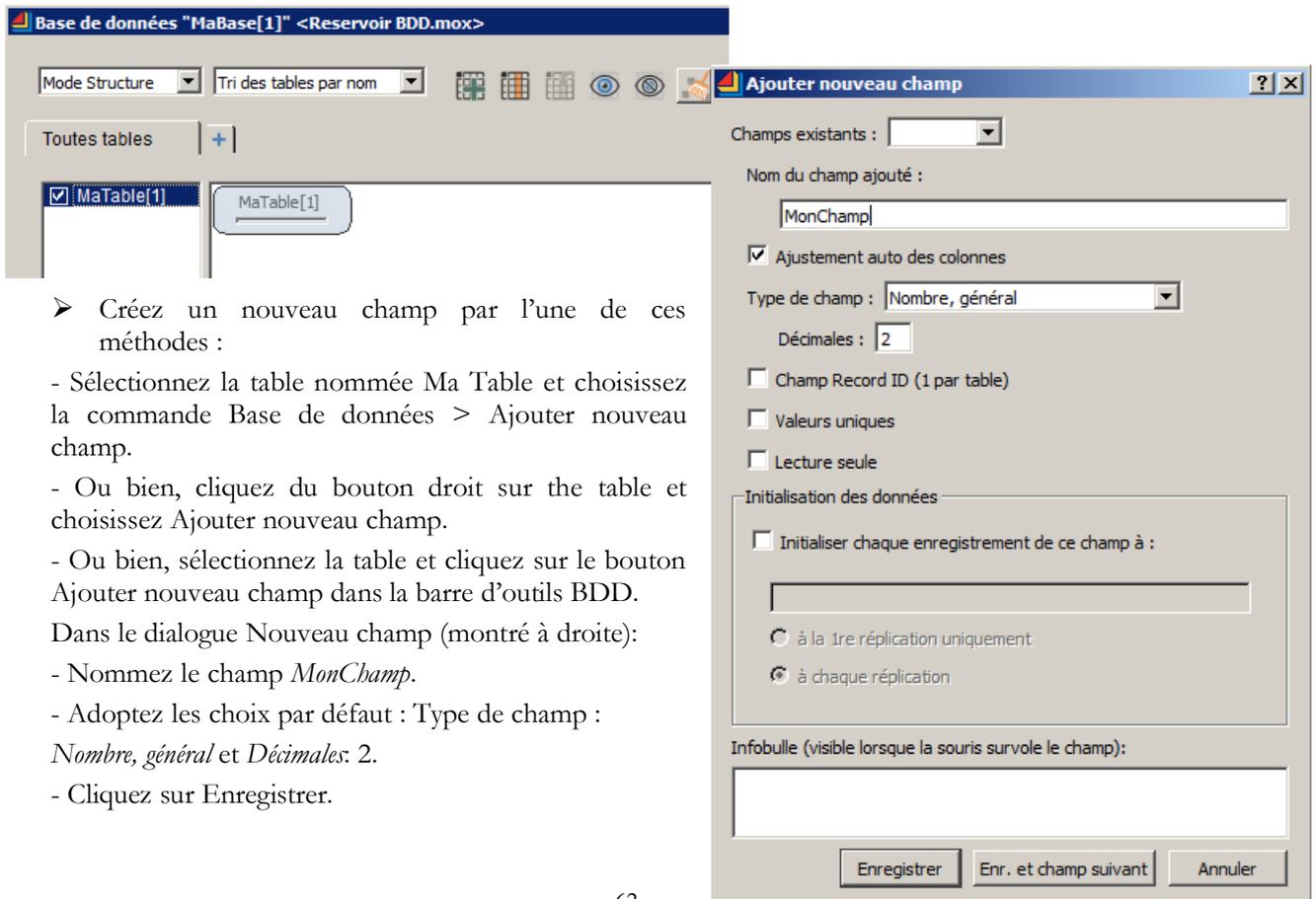
L'en-tête de fenêtre affiche le nom de la base de données et son modèle associé. La fenêtre de structure est la fenêtre où vous créez les tables de base de données, ajoutez des champs aux tables, et créez des relations entre les champs.

- ☞ ExtendSim affecte à chaque base de données un numéro d'index unique, en commençant par 1. On le voit avec la base en exemple, ce numéro est ajouté au nom de la base de données, entre crochets. Ce numéro d'index sert pour faire référence à la base de données dans des équations ou du code ModL.

Ajouter des tables et des champs

- Avec la fenêtre de BDD active, créez une nouvelle table par l'une de ces méthodes :
 - Choisissez la commande Base de données > Nouvelle table.
 - Ou bien, cliquez du bouton droit sur la fenêtre de BDD.
 - Ou bien, cliquez sur le bouton Nouvelle table dans la barre d'outils BDD.
- Dans le dialogue qui apparaît, nommez la table Ma Table et cliquez sur Enregistrer.

La nouvelle table est placée dans l'onglet Toutes tables. Le nom de la table figure dans la portion Tables sur la gauche, et la structure de la base de données apparaît dans la portion tables sur la droite. Comme pour les bases de données, après le nom de chaque table figure son numéro d'index entre crochets.



- Créez un nouveau champ par l'une de ces méthodes :
 - Sélectionnez la table nommée Ma Table et choisissez la commande Base de données > Ajouter nouveau champ.
 - Ou bien, cliquez du bouton droit sur the table et choisissez Ajouter nouveau champ.
 - Ou bien, sélectionnez la table et cliquez sur le bouton Ajouter nouveau champ dans la barre d'outils BDD.
- Dans le dialogue Nouveau champ (montré à droite):
 - Nommez le champ *MonChamp*.
 - Adoptez les choix par défaut : Type de champ : *Nombre, général* ; Décimales: *2*.
 - Cliquez sur Enregistrer.

Créer des enregistrements

Pour accéder à la table pour y créer des enregistrements, effectuez une des actions suivantes :

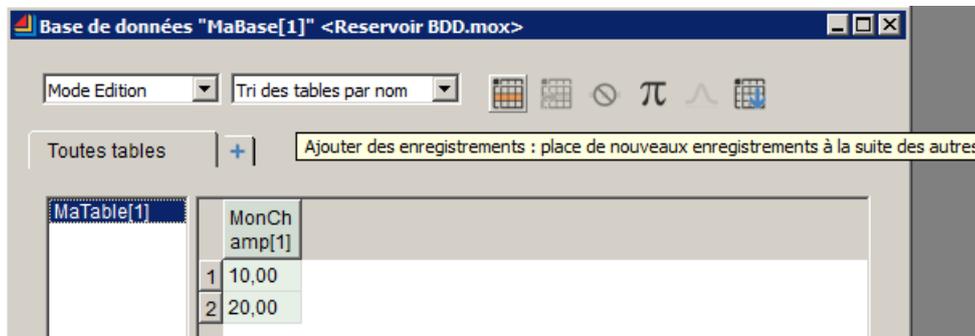
- Double-cliquez sur la table dans la fenêtre de BDD pour ouvrir le mode Edition pour cette table.
- Ou bien, double-cliquez sur le nom de table dans la liste des tables pour ouvrir le mode Edition.
- Ou bien, bouton droit sur the table et choisissez *Ajouter nouveau champ*.
- Ou bien, sélectionnez la table et passez du mode Structure au mode Edition dans le menu déroulant en haut à gauche de la fenêtre de BDD. Cela ouvre la portion d'édition dans la fenêtre de BDD.

Ajouter des enregistrements

- Cliquez sur l'outil *Ajouter nouveaux enregistrements* dans la barre d'outils BDD, ou choisissez la commande Base de données > Ajouter nouveaux enregistrements.
- Dans le dialogue qui apparaît, entrez 2 pour le nombre d'enregistrements et cliquez sur Ajoutez enregistrements.

Entrer des valeurs pour les cellules

- En mode Edition, entrez 10 pour l'enregistrement 1 et 20 pour l'enregistrement 2.
- Si la fenêtre Edition est ouverte, fermez-la.
- Fermez la fenêtre de BDD et enregistrez le modèle. Les bases de données sont enregistrées avec le modèle quand le modèle quand lui-même est enregistré.



Remarquez que MaBase figure au bas du menu Base de données.

Etablir des relations parent/enfant

Pour illustrer une importante fonction des bases de données, l'exemple suivant ajoute une nouvelle table à la base de données créée dans la section précédente et caractérise comme champ parent un champ de la table originale. Cela va établir une relation parent/enfant, faisant que MaTable obtienne ses valeurs de la nouvelle table.

Il n'y a pas d'obligation à établir une relation parent/enfant entre champs. Cela sert dans plusieurs situations, par exemple pour obtenir une liste de composants pour un produit, un choix de couleurs pour un modèle de voiture, ou les propriétés d'un matériel. Avoir une relation parent/enfant limite l'ensemble de valeurs possible à celles présente chez le parent. Au lieu d'entrer les valeurs en direct dans le champ enfant, vous les sélectionnez dans une liste déroulante qui affiche toutes les valeurs possible présente dans le champ parent.

Les champs Parent ont un fond rouge et les tables comportant des champs parent figurent en rouge dans la fenêtre de BDD ; les champs enfant ont un fond vert et les tables comportant des champs enfant sont listées en vert.

Comment créer une relation parent/enfant

Ce qui suit s'appuie sur le modèle ReservoirDB et suppose que vous savez comment créer des tables, champs et enregistrements dans une base de données.

- Ouvrez le modèle ReservoirDB que vous avez créé.
- Ouvrez la fenêtre de *MaBase* en choisissant MaBase au bas du menu Base de données.
- Créez une nouvelle table et nommez-la *Parent*.
- Sélectionnez la table Parent puis :
- Ajoutez un nouveau champ et nommez-le Contenu.
- Laissez les valeurs par défaut pour les autres paramètres du dialogue Propriétés de champ et cliquez

sur *Enregistrer*.

- Double-cliquez sur la table Parent pour ouvrir le mode Edition.
- Ajoutez 4 enregistrements à la table Parent.
- Entrez les valeurs 5, 8, 10 et 15, une valeur par enregistrement.
- Fermez le mode Edition de sorte que la fenêtre de BDD soit active.

Remarquez que chaque champ d'une table comporte des connecteurs sur sa droite et sa gauche. Ils servent à établir la relation parent/enfant.

- Tracez une ligne d'un connecteur de MonChamp à un connecteur de Contenu.
- Tracez la ligne de l'enfant vers son parent, la source des données.

Comme le champ enfant comporte déjà des données, le dialogue *Relations entre tables* s'ouvre avec des options pour la gestion des données du champ enfant. Sélectionnez le choix par défaut (en haut) et cliquez sur Etablir la relation.

Ceci définit le champ Contenu comme parent du champ MonChamp. Les champs parent se reconnaissent à la couleur rouge, les champs enfant à la couleur verte.



- Passez en mode Edition de sorte à sélectionner des valeurs pour les enregistrements de MonChamp.
- Sélectionnez Ma Table dans la portion Tables sur la gauche de l'onglet Toutes tables.
- Cliquez sur la flèche dans l'enregistrement 2, celui dont la valeur (20) a été supprimée car ne correspondant pas aux valeurs du champ Parent.
- Dans le choix déroulant qui apparaît, sélectionnez la valeur 15.

La cellule enfant du champ 1 de l'enregistrement 2 vaut maintenant 15 ; sa valeur provient du champ parent. Si la valeur du parent change, la valeur changera aussi chez l'enfant (mais pas vice versa.)

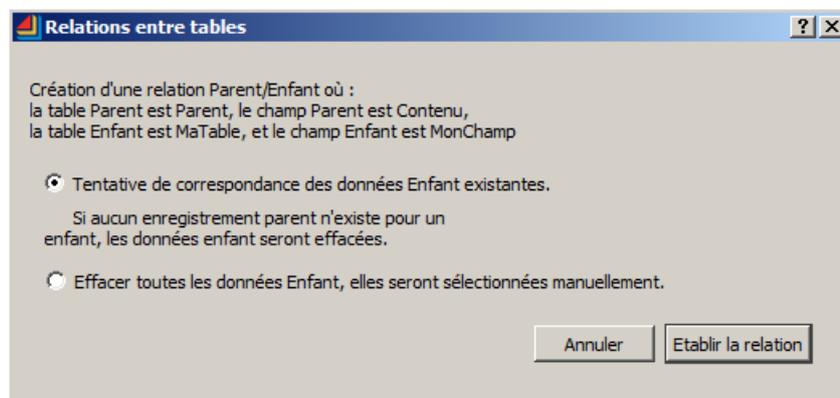
MaTable[1] Parent[2]	MonChamp[1]
1	10,00
2	[aucune valeur encore]

[aucune valeur encore]
[ajouter nvelle valeur parent]
<> : 5,00
<> : 8,00
<> : 10,00
<> : 15,00

Le menu sélecteur Enfant

En mode Edition, lorsque vous cliquez sur la flèche descendante d'une cellule enfant, cela affiche un menu déroulant avec 3 types d'options. Vous pouvez sélectionner une valeur dans la liste, ajouter une "nouvelle valeur" ou choisir "aucune valeur encore". Le choix "nouvelle valeur" permet d'ajouter une valeur nouvelle dans la cellule, et qui sera ajoutée au champ parent. L'option "aucune valeur encore" laisse la cellule vide.

Le dialogue Relation entre tables



S'il existe déjà des données dans un champ enfant, ce dialogue s'ouvre pour permettre de choisir ce qui se passera à l'établissement de la relation. Le dialogue n'apparaît pas s'il n'y a pas de données. Les options sont :

- *Faire correspondre les données enfant à l'enregistrement du champ Parent.* Si des données enfant correspondent aux données parent, elles sont conservées. Sinon, elles sont effacées et l'enregistrement est vide de sorte à permettre la sélection de valeurs.
- *Effacer les données enfant.* Toutes les données enfant sont effacées.

Lier une base de données à des données

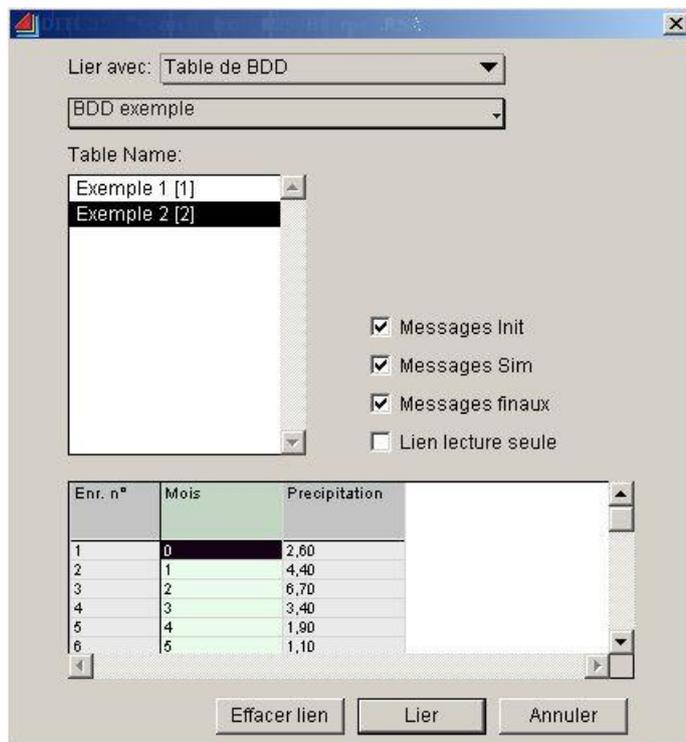
Les bases de données servent à stocker des données utilisées dans le modèle ou à conserver des résultats. Un modèle peut accéder aux informations d'une base de données par un lien dynamique, par des blocs de type équation ou par la programmation ModL.

Lien dynamique

Voici les étapes pour établir un lien dynamique entre un élément de dialogue de bloc et une base de données ExtendSim :

- 1) Créez une base de données.
- 2) Liez un paramètre ou une table de données.
- 3) Dans le dialogue Lier, sélectionnez Lier avec: Table de BDD.
- 4) Choisissez une base de données dans le menu déroulant des bases de données.
- 5) Sélectionnez une table dans la liste Noms de table.
- 6) Pour les paramètres numériques uniquement, choisissez un champ dans le menu Nom de champ, puis entrez un nombre dans le champ Enr. N°. Ou bien, sélectionnez le champ et l'enregistrement en cliquant sur une cellule dans la portion de visualisation au bas du dialogue.

☞ Cette portion affiche le contenu de la table à laquelle vous vous liez. Cela permet de choisir rapidement le champ et l'enregistrement pour un paramètre.



Attention avec les liens dynamiques

Evitez de surcharger un modèle de liens dynamiques vers des sources de données qui sont souvent modifiées : cela ralentirait les performances du modèle car des messages sont envoyés aux blocs liés chaque fois que la source est modifiée.

Les blocs Read et Write pour accéder à une base de données

Tout comme un lien dynamique, les blocs Read et Write (bib. Value) ainsi que Read(I) et Write(I) (bib. Item) permettent l'accès à cellule, une ligne ou une colonne issue d'une base de données.

Les avantages à utiliser ces blocs plutôt qu'un lien dynamique sont les suivants :

- Les blocs permettent un meilleur contrôle sur les valeurs de la base de données utilisées, quand et où elles sont utilisées.
- L'emploi de la base de données est visible au niveau du modèle par la présence des blocs (et pas seulement par le liseré bleu lorsqu'on ouvre le dialogue, comme avec le lien dynamique).
- Le lien dynamique précise la source et la cible. Tandis que vous pouvez modifier les valeurs de la source de données, vous ne pouvez pas contrôler dynamiquement quelle base de données ou table est accédée durant la simulation.
- Dans certains blocs il n'y a pas de paramètre ou de table de données pour établir le lien dynamique,
- L'accès aux données à partir de plusieurs endroits du modèle est facilité.
- La méthode s'adapte mieux aux blocs hiérarchiques, permettant d'accéder à un enregistrement différent pour chaque instance du bloc H dans la modèle.

Le modèle Monte Carlo, situé dans \Exemples\Continuous\Standard Block Models est un exemple d'utilisation des blocs Read et Write pour l'échange de données avec une base de données ExtendSim.

Gérer les bases de données

Cette section indique comment effectuer des opérations générales sur les bases de données ExtendSim, les tables, champs, enregistrements et données qu'elles contiennent.

Ouvrir une fenêtre de base de données

Pour ouvrir la fenêtre d'une base de données afin de modifier sa structure ou ses données, vous pouvez :

- Sélectionner la base parmi les bases de données listées au bas du menu Base de données.
- Ou bien, sélectionner Fenêtre > Liste BDD pour ouvrir le Navigateur en mode *Bases de données*, puis double-cliquer sur le nom de la base de données dans la liste.

Ouvrir une liste de bases de données

Pour accéder à la liste des bases de données, choisissez la commande Fenêtre > Liste BDD. La liste des bases de données sert à ouvrir une base de données, ou à copier, renommer ou effacer une base de données.

Copier, renommer ou effacer une base de données

Pour copier ou dupliquer une base de données, sélectionnez-la dans la liste des bases de données et choisissez la commande Edition > Copier (ou Dupliquer) BDD. Appliquez la commande Coller dans la liste des bases de données ou dans le modèle ; vous pouvez coller dans le même modèle ou dans un autre. Coller ou Dupliquer place une base de données dans la liste des bases de données, avec un numéro d'index et un nom différent (si dans la Liste BDD du même modèle) ou avec le même nom (si la base de données est copiée dans un autre modèle).

- La commande Coller appliquée dans le modèle place une base de données copiée dans la Liste BDD qui devient disponible dans le modèle. Aucune représentation visuelle de la base ne peut être remarquée.

Pour déplacer une base de données d'un modèle à une autre, sélectionnez-la dans la Liste BDD, faites Copier BDD dans le menu Edition puis Coller dans l'autre modèle.

Pour renommer une base de données, sélectionnez-la dans la Liste BDD et choisissez la commande Base de données > Renommer BDD. Pour supprimer une base de données, sélectionnez-la dans la Liste BDD et choisissez la commande Edition > Effacer BDD.

Importer ou exporter une base de données

La commande Base de données > Importer nouvelle BDD crée une nouvelle base de données ExtendSim en l'important entièrement d'une base de données ExtendSim exportée. Dans le dialogue, sélectionnez le fichier texte de base de données à prendre pour source. Cette commande importe toutes les tables, champs, enregistrements et crée une nouvelle base de données. Si vous choisissez le nom d'une base de données existante, elle sera intégralement remplacée. Pour importer des tables dans une base de données existante, de sorte que les tables soient ajoutées à la fin de la base de données, utilisez la commande Importer des tables.



La commande Base de données > Exporter BDD n'est active que lorsqu'une fenêtre de base de données ExtendSim est active. Cette commande exporte toute la base de données ExtendSim dans un fichier texte. Cela vous permet d'importer la base de données dans un autre modèle, d'envoyer la base de données à un autre utilisateur ou de préparer le fichier texte d'une base de données afin de l'utiliser avec l'Add-In ExtendSim DB pour Excel (examiné plus loin). Pour n'exporter que certaines tables depuis une base de données, voir la commande Exporter les tables, plus loin. Les bases de données exportées ne peuvent être importées que dans ExtendSim ou dans l'Add-In ExtendSim DB pour Excel. Pour activer cette commande, placez une fenêtre de BDD au premier plan.

Gérer les tables de base de données et utiliser les onglets

Les onglets permettent d'organiser les tables par catégories. L'onglet *Toutes tables* contient toujours la liste de toutes les tables d'une base de données. D'autres onglets pourront ne montrer que certaines tables. La fenêtre de base de données doit être en mode Structure pour ajouter un onglet. Pour créer

L'onglet, double-cliquez sur la zone grisée vide dans la barre des onglets (à la droite de l'onglet *Toutes tables*) ou choisissez la commande Base de données > Nouvel onglet de BDD. Clonez une table dans le nouvel onglet en la sélectionnant dans la portion tables et en appelant la commande Base de données > Cloner les tables sur l'onglet. Dans le dialogue qui s'affiche, sélectionnez l'onglet où cloner la table.



Vous importez des tables qui s'ajoutent à la base de données courante au moyen de la commande Base de données > Importer des tables lorsque la fenêtre de base de données est ouverte. Ceci importe un fichier texte de tables de base de données issu d'ExtendSim.

Vous importez des tables par la commande Base de données > Exporter des tables. Le fichier résultant pourra être importé dans une base de données ExtendSim.

Pour masquer/afficher des tables dans la portion Tables, désélectionnez la table dans la liste des tables dans n'importe quel onglet de la fenêtre de BDD.

Pour copier une table d'une base de données à une autre, sélectionnez la table dans la fenêtre de BDD, choisissez la commande Edition > Copier tables, et collez la table dans une fenêtre de base de données. Pour déplacer une table d'une base de données à une autre, utilisez la commande Edition > Effacer tables suivie de Coller.

Gérer les champs

Les champs sont formatés dans les dialogues *Propriétés de champ* et *Type de champ* examinés plus haut.

Pour changer l'ordre des champs dans une table, avec la fenêtre de BDD en mode Structure, cliquez et faites glisser les champs. Pour redimensionner la largeur de colonne d'un champ, placez la fenêtre de BDD en mode Edition. Puis placez le curseur au-dessus du diviseur de colonne qui devient alors un curseur de dimensionnement, que vous déplacez pour changer la taille.

Utilisez l'outil Trier Table dans la barre du mode Edition pour trier les enregistrements en fonction de trois champs; chacun en ordre ascendant ou descendant. Par exemple, vous pouvez trier les enregistrements d'abord par nom, puis par prénom, puis par ville.

Vous pouvez trier la liste des tables par nom ou par index.

- Pour créer un champ avec plusieurs formats (par exemple à la fois des nombres et des lettres), donnez-lui le type Chaîne dans le dialogue Propriétés de champ. Entrez des chiffres ou des lettres dans les cellules.

Enr. n°	Mois	Precipitation
1	0	2,60
2	1	4,40
3	2	6,70
4	3	3,40
5	4	1,90
6	5	1,10
7	6	0,70
8	7	0,50
9	8	0,40
10	9	0,70
11	10	2,60
12	11	3,40

Editer des données

Pour éditer les données, depuis la fenêtre de BDD vous devez :

- Double-cliquer sur une table dans la fenêtre de BDD pour ouvrir le mode Edition.
- Ou bien, sélectionner la table dans la portion tables et mettre la fenêtre de BDD en mode Edition. Les enregistrements et les champs de la table sélectionnée s'afficheront.

Par défaut, les valeurs dans les cellules sont des constantes. Pour qu'elles puissent être aléatoires, sélectionnez la cellule et choisissez l'outil Rendre cellule aléatoire dans la barre d'outils.

Interrogation des données

Pour rechercher une chaîne ou un nombre à l'intérieur d'une base de données, avec la fenêtre de structure ouverte, choisissez la commande Edition > Rechercher. Cette commande ouvre le dialogue Recherche/Remplacement. Pour ne parcourir qu'une seule table, Sélectionnez la table avant d'utiliser la commande.

Dialogues et menus déroulants de la BDD

La plupart des dialogues de bases de données sont explicites. Nous détaillons les éléments qui requièrent une explication.

Dialogue Propriétés de champ

Lorsque vous ajoutez un champ à une table, le dialogue Propriétés de champ propose des options décrites dans le tableau qui suit.

Option	Description
Nom de champ :	entrez un nom unique de 63 caractères maximum. Les espaces sont autorisés.
Type :	Format du champ, voir tableau ci-dessous.
Décimaux :	pour le format Général et Scientifique, nombre de chiffres. Pour devise et pourcentage, le nombre de chiffres après la virgule.
Utiliser séparateurs :	insère des virgules et des points pour séparer les milliers dans les nombres.
Champ index :	champ servant d'index pour rechercher un enregistrement dans la table.
Valeurs uniques :	si cochée, le contenu de chaque enregistrement pour ce champ doit être unique.
Lecture seule :	si cochée, empêche de modifier les données après la première saisie.
Initialiser chaque enregistrement de ce champ à :	si cochée, initialise ce champ pour tous les enregistrements à la valeur entrée. Si aucune valeur d'initialisation n'est donnée, initialise les enregistrements à vide (blank).
Première répllication...Chaque répllication :	spécifie quand doit avoir lieu l'initialisation.

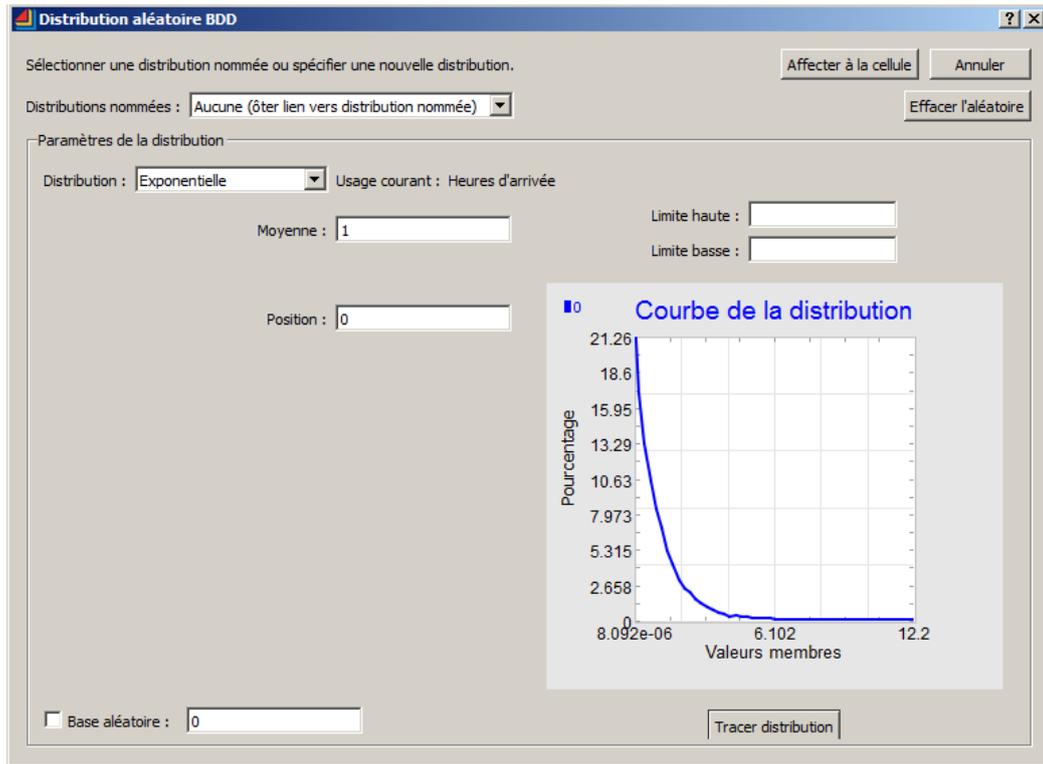
Menu déroulant Type de champ

Type	Description
Nombre	Un nombre entier ou décimal à point flottant double-précision. Choisissez le format général, scientifique, Pourcentage, devise ou entier.
Chaîne	Une chaîne alphanumérique jusqu'à 255 caractères
Date Heure	Date et heure calendaires
Adresse de données	Décrit les coordonnées d'une cellule dans une table de base de données
Case à cocher	Place une case à cocher pour indiquer oui ou non, vrai ou faux, on ou off, etc.
Liste des tables	Place un menu déroulant de toutes les tables de la base de données, permettant de sélectionner une table.

Dialogue Distribution aléatoire BDD

Par défaut toutes les cellules de base de données sont des constantes, mais on peut leur donner une nature aléatoire. Pour cela, sélectionnez les cellules en mode Edition, et choisissez l'outil *Rendre cellule aléatoire* dans la barre d'outils. Ou bien cliquez du bouton droit sur la cellule et choisissez Rendre aléatoire. Ceci ouvre le dialogue Distribution aléatoire BDD présenté ci-contre.

Les distributions aléatoires de ce dialogue sont les mêmes que celles du bloc Random Number (bib. Value).



- Ce dialogue permet notamment d'assigner un nom à une distribution dans la base de données, permettant que la distribution soit sélectionnée par un nom spécifique dans la base de données. Par exemple, entrez des valeurs dans un tableau empirique et nommez la distribution Temps Machine. Ou donnez un nom spécial à une distribution exponentielle avec certains arguments.

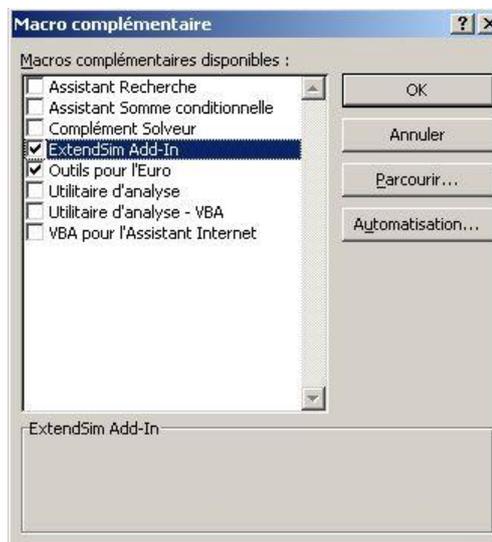
Add-In Excel pour les bases de données ExtendSim

Ce module complémentaire permet qu'un fichier texte résultant de l'exportation d'une base de données ExtendSim puisse être importé dans Excel. Les données et la structure de la base de données ExtendSim peuvent ainsi être modifiées sous Excel, hors de l'application ExtendSim. Après modifications, la base de données peut être exportée depuis Excel sous forme d'un fichier texte et importée dans ExtendSim.

L'Add-In Excel pour les bases de données ExtendSim **n'est fourni qu'avec ExtendSim Pro**. Il peut toutefois être acheté séparément.

Pour installer ce module additif dans Excel:

- Suivez les instructions Excel pour ajouter un Add-In (programme auxiliaire). Elles dépendent de votre version d'Excel et de votre configuration. Avec Excel 2003, c'est la commande Macro complémentaire dans le menu Outils.
- Dans le dialogue *Macro complémentaire*, cliquez sur le bouton Parcourir et naviguez jusqu'à trouver le fichier ExtendSim DB Add-In.xla; dans le répertoire ExtendSim\Extensions.
- Sélectionnez le fichier.xla file et cliquez sur OK. Cela vous ramène dans le dialogue Macro complémentaire où ExtendSim DB Add-In a été ajouté dans la liste des Add-In disponibles. (Assurez-vous que la case est cochée à côté de ExtendSim DB Add-In.)
- Cliquez sur OK pour fermer le dialogue *Macro complémentaire*. Cette action installe une nouvelle commande de menu Add-In appelée "ExtendSim DB".

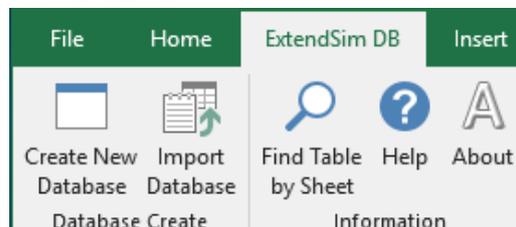


Pour exporter une base de données ExtendSim de sorte qu'Excel puisse l'importer, voir la section correspondante [plus haut](#). Le fichier exporté doit être au format texte. Pour l'importer depuis Excel, choisissez la commande ExtendSim DB > Import Database dans Excel.

Pendant l'importation, la barre d'état au bas de la feuille signale la table en cours ainsi que le nombre d'enregistrement en cours d'importation et le nombre total d'enregistrements qui seront importés.

Après importation, vous pouvez modifier les données dans les tables, changer le nombre d'enregistrements (lignes) ou de champs (colonnes), changer le type des champs, modifier des distributions. Vous pouvez même ajouter ou modifier les relations dans la base de données, bien que ce soit plus facile à faire sous ExtendSim.

- Pour modifier les caractéristiques d'un champ, cliquez sur le bouton Dégroupier (+) à la gauche de la table. Ceci déploie les propriétés du champ, que vous pouvez modifier.



	19	Table	Exemple 1
	20	Table Index	1
	21	Table in Tabs	1
	22	Tab Locations	"(20;20)"
	23	Table Format	0
	24	Access	read_write
	25	Format	general
	26	Decimal	2
	27	Comma	no
	28	Unique	no
	29	Record ID	no
	30	Do Initialization	no
	31	Init First Run Only	no
	32	Initializer	
	33	Notes	
	34	Fields	Valeurs initiales
	35		1 15
	36		2 25
	37		

Champ groupé et non groupé

- Pour ajouter des lignes ou des colonnes à une table qui n'est pas vide, copiez une ligne/colonne existante et collez les lignes/colonnes au bas/sur la droite de la table. Cela agrandira la table. Modifiez ensuite les données des nouvelles lignes/colonnes. Si la table est encore vide, entre le nombre de lignes sous l'identifiant Field.
- Pour modifier les relations, entrez les noms des table et de champs appropriés dans la table Indexed Fields au bas de l'onglet Toutes tables.
- Lorsque vous changez le nombre de lignes pour une distribution empirique, veillez à modifier aussi le nombre de lignes dans la table Empirique.

Champs parent/enfant

- Les champs parent et enfant sont maintenant codés en couleur dans la feuille Excel, correspondant au codage d'ExtendSim : rouge pour les champs parent et vert pour les champs enfant.
- Quand la BDD est importée et/ou que la cohérence de la BDD est vérifiée, les noms des champs enfant possèdent maintenant des hyperliens qui renvoient à leurs champs parent. Par exemple, cliquer sur l'en-tête vert d'un champ enfant vous amène au champ lié dans la table parent. Ceci permet à l'utilisateur de trouver rapidement les tables parent et de voir la liste des valeurs enfant valides.

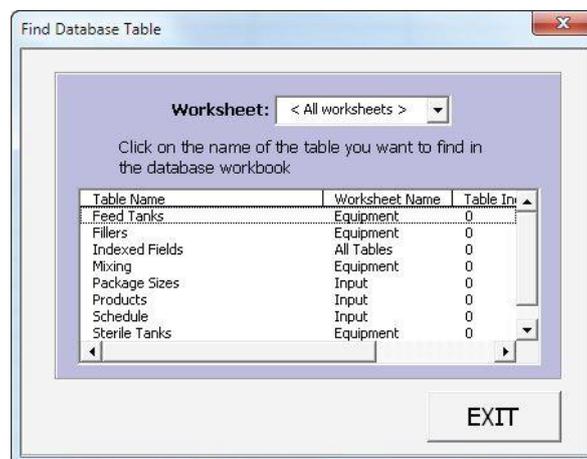
Find Database Table

Cette commande permet à l'utilisateur de trouver et d'aller rapidement dans une table de BDD dans leur feuille Excel. Sélectionner la commande ouvre le formulaire Find Database Table qui affiche une liste de toutes les tables de la feuille sélectionnée.

→ Chaque feuille est l'équivalent d'un des onglets de la BDD importée.

Pour sélectionner un des onglets de la BDD, cliquez sur la liste du menu déroulant de la feuille, qui affiche toutes les tables sur la feuille sélectionnée. Pour voir une liste de toutes les tables présentes dans la BDD, sélectionnez l'option « Toutes les feuilles ».

Quand vous sélectionnez un nom de table dans la liste des tables, la cellule active de la feuille sera placée à l'endroit où la table de BDD a été définie.



Vérification de la cohérence de la BDD

Une des nouvelles options du menu de l'Add-In est « Check Database Consistency ». Cette commande accomplit les validations de la BDD et de ses relations parent/enfant. Ceci permet de s'assurer que la structure de la BDD et le contenu des données soient cohérents et valides.

Si des erreurs sont trouvées elles sont enregistrées dans un Rapport d'erreurs et un message s'affiche à la fin de la vérification. Le Rapport d'erreurs contient des hyperliens qui renvoient à ces erreurs pour que vous puissiez les corriger.

Les quatre validations

La commande Check Database Consistency effectue les vérifications suivantes.

1) Contenu des données parent/enfant

Pour chaque champ enfant, les contenus sont vérifiés pour s'assurer que les valeurs correspondantes existent dans les champs parents des tables parents. Chaque cas d'une valeur de champ enfant n'ayant pas de valeur de champ parent correspondante est enregistré dans le Rapport d'erreurs.

2) Relations parent/enfant

Les relations parent/enfant définies par l'utilisateur dans la table Indexed Fields sont validées. Cette vérification permet de s'assurer que chaque table enfant et parent et que chaque champ parent et enfant référencé dans la table Indexed Fields existe vraiment dans le classeur de la BDD.

3) Nom unique des tables et des champs

Ceci permet de s'assurer que chaque nom de table est unique dans la BDD et que chaque nom de champ est unique pour une table donnée. Chaque cas de nom de table et de champ qui n'est pas unique est enregistré dans le Rapport d'erreurs.

→ L'Add-In ne fonctionne que sur une BDD à la fois. Il est donc possible que d'autres BDD puissent contenir une table avec le même nom qu'une autre table de la BDD de la feuille Excel.

4) Chaque valeur unique

Cette fonction regarde les contenus de l'enregistrement pour chaque champ dont la case *Chaque valeur unique* a été cochée dans son dialogue de Propriétés du champ. La validation permet de s'assurer que toutes les valeurs dans les tables correspondantes sont uniques. Des valeurs qui ne seraient pas uniques sont signalées dans le Rapport d'erreurs.

Après avoir effectué vos modifications, exportez la base de données par la commande Excel ExtendSim DB > Export Database. Importez ce fichier dans ExtendSim par la commande Bases de données > Import Nouvelle BDD.

☞ L'Add-In ne vous permettra pas d'exporter une BDD à partir d'un classeur Excel tant que la structure et le contenu de la BDD ne sont pas validés avec succès.

Modèle Monte Carlo

Le modèle Monte Carlo (située dans le répertoire \Exemples\Continuous\Standard Block Models) utilise une base de données ExtendSim nommée "Scenarios" pour stocker les entrées et sorties du modèle. La base de données permet au modèle d'expérimenter différents scénarios.

Plutôt que de lier dynamiquement les paramètres de données et les tables à une base de données, le modèle utilise les blocs Read et Write (bib. Value) en guise d'interface à la base de données. Les blocs Read obtiennent des valeurs aléatoires depuis une base de données et les utilise pour représenter les paramètres du modèle lorsque les valeurs ne sont pas connues avec certitude.

Echange d'informations avec ExtendSim

Il est facile d'échanger des informations entre ExtendSim et d'autres programmes : ExtendSim s'appuie sur les techniques standard de son environnement, et transite par des fichiers texte pour les programmes externes à l'environnement. Vous pouvez aussi connecter ExtendSim à des périphériques divers.

Cette section décrit les fonctions de Copier/Coller et d'Import/Export d'ExtendSim, ainsi que les fichiers texte. Les autres transferts de données, plus spécifiques à chaque environnement, tels que ODBC, les Mailslots, les DLL, les communications entre applications, les fonctions de gestion du port série et de drivers de périphériques sont examinés en détail plus loin.

Copier, Couper et Coller par le Presse-Papiers

Le Presse-Papiers est utile pour passer des informations en restant dans ExtendSim ou vers d'autres applications. Vous copiez et collez les données des tables et blocs d'ExtendSim, et même des portions de modèles. Blocs et portions de modèles peuvent être passés vers des applications graphiques, des traitements de textes ou d'autres modèles ExtendSim. Les données des tables de données seront transférées vers n'importe quelle application du même environnement, traitement de textes ou tableur. ExtendSim connaît les commandes Copier, Couper et Coller comme les autres applications. Vous pouvez examiner le contenu du Presse-Papiers par la commande *Voir le Presse-Papiers* du menu Edition.

En restant à l'intérieur d'ExtendSim, le Presse-Papiers sait reconnaître les paramètres de blocs et les connexions : cela permet de coller des portions de modèles, y compris les paramètres de dialogue, à un autre endroit du modèle, dans un autre modèle ou dans l'Album. Lorsqu'un bloc est collé dans une autre application, il devient un graphique. À l'inverse, si vous collez une image dans ExtendSim en provenance d'un autre programme, elle devient un dessin qui peut être déplacé et redimensionné. Les fichiers graphiques sont des fichiers au format Windows metafile.

Si un graphique est la fenêtre active, la commande Copier place une image du tracé dans le Presse-

Papiers. Vous pouvez copier les données de la table en sélectionnant les données de votre choix et en les copiant. Dans ce cas, le contenu du Presse-Papiers devient du texte. Vous pouvez faire de même pour le texte contenu dans des dialogues. Notez que les données des tables de données sont directement exploitables par les tableurs courants.

Si vous voulez copier des portions de modèle qui ne soient pas directement copiables depuis ExtendSim (tels qu'un dialogue entier ou une fenêtre de tracé), vous pouvez utiliser un programme de capture.

Les images

Vous pouvez copier dans ExtendSim des images en noir et blanc ou en couleurs pour les utiliser dans les modèles, les blocs hiérarchiques ou les icônes de blocs standard. Vous pouvez créer l'image avec un outil de dessin puis la copier dans le Presse-papiers, et enfin la coller dans ExtendSim par l'option Coller image du menu Edition. Les images collées dans ExtendSim, comme des objets graphiques créés avec ExtendSim, se placent toujours derrière les blocs et le texte.

Les images et les objets graphiques (rectangles, rectangles arrondis et ovales) peuvent être redimensionnés proportionnellement. Pour cela, maintenez la touche Maj appuyée lorsque vous faites glisser la souris.

Vous pouvez exporter ce qu'affiche un bloc Chart vers d'autres applications afin de l'inclure dans des rapports ou des présentations. Pour copier l'affichage du graphique, choisissez simplement la commande Copier tracé du menu Edition alors que la fenêtre du graphique est active. Pour copier les données de la table, vous devez d'abord les sélectionner puis actionner la commande Copier.

Si vous voulez copier des portions de modèle qui ne soient pas directement copiables depuis ExtendSim (tels qu'un dialogue entier ou une fenêtre de tracé), vous pouvez utiliser un programme de capture d'écran (il suffit même de la combinaison de touches CTRL-Imp Ecran) pour sauvegarder ce que vous avez à l'écran. Certains programmes de capture vous permettent de sélectionner la portion d'écran à capturer.

Les fichiers texte

Un fichier texte est un fichier sans format spécifique créé par un tableur, une base de données ou un traitement de textes. Ce format est généralement proposé lorsque vous choisissez l'option Enregistrer sous du menu Fichier dans une application. Dans de nombreux cas vous voudrez lire et écrire sur des fichiers texte (ou fichiers ASCII). Par exemple :

- Vous voulez partager des données avec une base de données ou un tableur. La plupart des programmes savent enregistrer leurs données sous forme de fichier texte.
- Les fichiers partagés avec d'autres environnements sont souvent en format texte.
- La plupart des fichiers provenant de minis ou de mainframes sont aussi en format texte.

ExtendSim peut manipuler des fichiers texte standard de plusieurs manières :

- La façon la plus simple est d'utiliser les blocs Importer et Exporter de la bibliothèque Value. Reportez-vous à l'aide en ligne de ces blocs.
- Vous pouvez utiliser les commandes Importer et Exporter du menu Edition pour lire et écrire sur des fichiers texte dans un dialogue ou dans une table de données d'un graphique. Vous devez sélectionner les colonnes de données de la table avant d'importer ou d'exporter.
- Si vous utilisez l'analyse de sensibilité, vous pouvez créer un fichier texte dans ExtendSim ou dans une autre application, constituant les valeurs variables dans votre plan d'expérience.
- Si vous créez vos propres blocs, vous pouvez utiliser les fonctions d'entrée/sortie décrites avec le langage de programmation.
- Le Rapport et la Trace génèrent aussi des fichiers texte. Ces fonctions sont expliquées en détail plus haut.

Utilisation des fichiers texte

Vous pouvez lire, écrire, modifier et imprimer des fichiers texte depuis ExtendSim. Cela permet de visualiser des fichiers de rapport, de modifier des sources de données, et ainsi de suite, sans devoir charger un autre programme.

Par exemple, vous pouvez créer un nouveau fichier texte et entrer des valeurs qui seront utilisées dans une analyse de sensibilité ou par les blocs Read et Import Export.

Les fichiers texte sont manipulés depuis le menu Fichier.

- Pour créer un nouveau fichier texte, choisissez la commande **Nouveau fichier texte**.
- Pour ouvrir un fichier texte, choisissez la commande **Ouvrir**.
- Pour enregistrer un fichier texte, choisissez la commande **Enregistrer texte sous**.
- Pour fermer un fichier texte, choisissez la commande **Fermer** ou cliquez sur la case de fermeture dans le coin supérieur gauche de la fenêtre.

Lorsqu'un fichier texte est la fenêtre active, de nombreuses commandes d'édition sont accessibles par les menus Edition et Définir. Par exemple, vous pouvez utiliser la commande Couper pour supprimer une portion de texte, ou la commande Trouver pour trouver une chaîne de texte particulière.

Les commandes du menu Texte permettent aussi d'améliorer la lisibilité d'un fichier texte, par exemple en augmentant la taille des caractères. Toutefois, ces modifications de style sont abandonnées lorsque vous fermez le fichier. Pour modifier de façon permanente des caractéristiques typographiques du texte, utilisez l'option Script/Texte via la commande Options.

Lorsque vous créez des fichiers texte pour les utiliser comme source de données pour des blocs ExtendSim ou d'autres logiciels, vous devez tenir compte de la structure de données demandée. Par exemple, certains blocs admettent plusieurs caractères pour délimiter les données (tabulation, espace, virgule), alors que l'analyse de sensibilité ne reconnaît que la tabulation. Les lignes sont toujours séparées par des retours chariot.

Note : les programmes qui requièrent un délimiteur Tabulation n'en attendent qu'un seul. Deux tabulations perturberaient le programme récepteur.

Partage de données via les matrices globales

Les matrices globales sont des matrices (tableaux) utilisables n'importe où dans le modèle. Elles peuvent s'employer pour transférer des valeurs d'un point à un autre ou pour accumuler des informations. Les matrices globales sont créées par le bloc Data Source Create (bib. Value) et manipulées par divers blocs, notamment Read et Write. Elles peuvent être de type Réel ou Entier. Chaque MG possède un nom unique, au plus 255 colonnes et autant de lignes qu'on veut.

Ces matrices contiennent des nombres pouvant être définis, modifiés ou réinitialisés. Plusieurs blocs d'un modèle peuvent accéder à la même matrice, pour définir des valeurs dans la matrice, et pour lire ces valeurs à différents endroits du modèle. Si l'accès aux données est plus rapide qu'avec les bases de données, la structure de données reste plus primaire que les BDD (pas de type aléatoire, pas de relation entre matrice comme la relation parent-enfant, interface de gestion moins adaptée).

Une fois créées, les matrices globales peuvent être remplies de valeurs par les blocs Data Import Export et Data Init (bib. Value), ou remplies manuellement dans le dialogue du bloc Data Source Create lui-même.

Le bloc Data Import Export peut aussi importer et exporter des informations de/vers une source ODBC (open database connectivity) compatible (sous Windows). Définissez la source ODBC (depuis le Panneau de configuration Source de données ODBC) et sélectionnez la base de données, la table, et les colonnes à importer dans l'onglet Base de données. Les informations peuvent être transférées en réponse à un clic utilisateur (boutons Importer et Exporter) ou en début et fin de simulation.

Le bloc Data Import Export peut encore importer et exporter des informations de/vers une feuille MS Excel. Sélectionnez le fichier puis la feuille à l'intérieur du fichier. Vous pouvez spécifier une plage à importer ou exporter. Lors de plusieurs répliques l'option "Feuille suivante à chaque réplique" passera d'une des feuilles numérotées à l'autre.

Le bloc Data Import Export peut enfin importer et exporter des informations de/vers un fichier texte en utilisant le protocole File Transfer Protocol (FTP) via internet. Entrez les informations FTP (adresse, nom, mot de passe, etc.) et lancez le transfert en cliquant sur un bouton Importer/Exporter ou en cochant la case "Lire/Ecrire en début/fin de simulation".

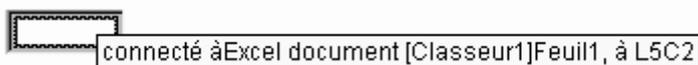
Les liens dynamiques ("hot links")

Il est possible à ExtendSim et à une autre application de partager des données au moyen d'un lien dynamique ("hot link"). Par ce lien, lorsque vous modifiez des données dans une application, elles sont automatiquement modifiées dans l'autre. Il existe deux types de liens directs dans ExtendSim : 1) les liens qui importent des données dans votre modèle, 2) les liens qui exportent des données du modèle vers une autre application.

Lier des données d'une autre application vers ExtendSim

Les données d'une autre application peuvent servir pour des paramètres d'entrée de votre modèle (champ numérique ou table de données). Voici les étapes pour créer un lien dynamique :

1. Dans l'application serveur, copiez les données dans le Presse-papiers.
2. Dans ExtendSim, sélectionnez le champ dans lequel vous voulez copier les données. S'il s'agit d'une table de données, vous devez sélectionner toutes les cellules concernées.
3. Si l'application serveur supporte les liens dynamiques, ExtendSim le reconnaîtra et la commande Coller un lien du menu Edition sera activée. Sélectionnez cette commande pour lier les données à ExtendSim.



Si le lien dynamique a été correctement créé, le champ numérique aura un cadre dans le dialogue. Une fois le lien établi, toute modification effectuée dans l'autre application sera immédiatement répercutée dans ExtendSim. Pour effacer le lien, sélectionnez le paramètre et sélectionnez Supprimer le lien dans le menu Edition.

Lier des données ExtendSim vers une autre application

Pour exporter des données de votre modèle vers une autre application au moyen d'un lien dynamique, voici les étapes à suivre :

1. Dans ExtendSim, sélectionnez le champ numérique ou les cellules de la table à lier.
2. Sélectionnez la commande Copier dans le menu Edition.
3. Reportez-vous à la documentation de l'autre application pour savoir comment créer le lien dynamique. Généralement, vous sélectionnez une commande Coller spéciale dans le menu Edition.

Si le lien dynamique a été correctement créé, le champ numérique aura un cadre dans le dialogue. Une fois le lien établi, toute modification effectuée dans ExtendSim sera immédiatement répercutée dans l'autre application. Pour effacer le lien, reportez-vous à la documentation de l'autre application.

Les liens dynamiques sont enregistrés avec le modèle. ExtendSim essaie de les rétablir lorsque le modèle s'ouvre.

Les ports série

Si vous créez vos propres blocs avec ModL et que vous souhaitez transférer des données vers/ depuis des périphériques série, plusieurs fonctions sur le port série sont disponibles. Ces fonctions peuvent lire ou écrire tout type de données (même des données en temps réel) sur le port série. Cela permet par exemple le transfert de données par modem. Les trois fonctions sont très simples d'emploi et sont utilisables par les programmeurs ModL débutants. Voir PORTS_S.MOX dans le répertoire PROG pour des exemples de blocs utilisant ces fonctions.

DLL : accéder à du code écrit dans d'autres langages

Les DLL sont des bibliothèques de code écrit et compilé dans un langage. Leur interface standardisée permet un lien entre ExtendSim et d'autres langages que ModL. Elles doivent être stockées sous forme de fichiers dans le répertoire Extensions d'ExtendSim.

DLL d'interface

Un premier ensemble de fonctions DLL permet à l'utilisateur d'accéder à des DLL Windows existantes. Parce qu'elles ont des listes d'arguments variables, ces fonctions permettent d'appeler presque n'importe quelle DLL existante. Cette interface inclut les fonctions suivantes : DLLLongFunction, DLLDoubleFunction, DLLBoolCFunction, DLLVoidCFunction, DLLLongPascalFunction, DLLDoublePascalFunction, DLLBoolPascalFunction, DLLVoidPascalFunction et DLLMakeProcInstance.

Un second ensemble de fonctions DLL a été conçu pour la compatibilité avec la version Macintosh d'ExtendSim. Il n'est utile que pour appeler des DLL écrites spécifiquement pour ExtendSim. Comme il utilise le même format d'appel et les mêmes arguments que les fonctions XCMD pour la version Macintosh, certaines modifications doivent être apportées au code. Cet ensemble regroupe trois fonctions : DLLXCMD, DLLParam et DLLArray.

Les trois fonctions XCMD sont les équivalents des fonctions XCMD, XCMDParam et XCMDArray pour Macintosh. Si vous utilisez un groupe ou l'autre de ces fonctions, vous n'aurez pas besoin de modifier votre code si vous transférez des blocs d'un environnement à l'autre. Cependant, l'extension doit être reconstruite.

Transformer du code en DLL

Vous devez écrire le code DLL avec un compilateur Windows capable de compiler des DLL. La DLL doit être construite pour une exécution 32 bits.

Vous devez sélectionner l'une des deux interfaces présentées ci-dessus. L'interface multi-plate-forme rend facile le transfert d'applications, mais est plus limitée et moins souple d'emploi.

Vous devez modifier légèrement le code pour ajouter une interface d'appel DLL, là encore en fonction de l'interface choisie.

Après avoir compilé le code, vous obtenez un fichier DLL. Placez le fichier dans le dossier **Extensions** et relancez ExtendSim.

Points à noter

S'il est possible que vos blocs soient utilisés sur Macintosh comme sous Windows, le code de vos blocs doit vérifier la plate-forme d'exécution pour éviter toute erreur.

Les variables réelles et entières passées depuis du code ModL à une DLL sont passées par valeur. Les variables réelles sont sur 8 octets à double précision, et les entiers sur 4 octets (entiers longs).

Les chaînes et les matrices sont passées aux DLL sous forme de pointeurs sur des données allouées par ExtendSim. Des modifications sur ces données affecteront les informations originales dans ExtendSim. Il est possible d'accéder et de modifier des données de ce type, mais n'essayez pas de redimensionner le pointeur. Si vous le faites, ExtendSim ne sera pas capable d'accéder aux données et risque de se planter.

Les chaînes sont passées aux DLL sous forme de chaînes Pascal, pas C. Cela signifie que la chaîne est précédée d'un octet de taille et ne se termine pas par un zéro. Par exemple, si vous passez une chaîne à une DLL, la DLL obtiendra un pointeur sur 256 octets de données dans lesquels le premier octet contiendra le nombre de caractères dans la chaîne.

Travail en réseau en utilisant les "Mailslots" (Windows uniquement)

La fonction mailslots d'ExtendSim permet à des applications ExtendSim fonctionnant sur différents ordinateurs d'un même réseau local de communiquer. La copie d'ExtendSim fonctionnant sur l'ordinateur récepteur ouvre un mailslot. L'ordinateur émetteur peut envoyer une chaîne de 255 caractères à un bloc de réception sur un autre ordinateur.

Le destinataire du message est indiqué par deux chaînes de caractères : le nom de l'ordinateur destinataire (qui peut être trouvé sur le réseau Windows) et le nom de mailslot (spécifié par le bloc de réception lorsque le mailslot est créé). Par exemple le bloc de réception d'un ordinateur appelé "guillaume" créerait un mailslot appelé "mymail" et stockerait l'index de ce mailslot avec le code suivant :

```
MailslotIndex = MailSlotCreate("mymail");
```

Un bloc d'un autre ordinateur pourrait envoyer un message au mailslot "mymail" sur l'ordinateur "guillaume" avec la fonction ModL suivante :

```
MailSlotSend("guillaume", "mymail" "Hello à tous");
```

Lorsque le message est reçu par "guillaume", la fonction "On MailSlotReceive" est appelée. Cela signifie qu'un mail attend dans la file d'attente mailslot. Le message continuera d'être périodiquement appelé par ExtendSim jusqu'à ce que le mail mis en attente soit enlevé par la fonction MailSlotRead. L'ordinateur destinataire peut obtenir le message avec la commande :

```
Message = MailSlotRead(MailSlotIndex); // ItemString is a 255 character string
```

Les blocs de simulation "distribuée" (plusieurs modèles calculent et communiquent en même temps)

ExtendSim contient un exemple utilisant les mailslots pour envoyer une entité d'un modèle à un autre. Deux blocs de la bibliothèque "Custom Blocks" (Répertoire Distributed Computing) sont utilisés pour transférer l'entité. Le modèle récepteur utilise le bloc "Receive Item from model". Le modèle émetteur utilise le bloc *Sent item to model*. Un nom de mailslot commun est utilisé pour lier entre eux les deux blocs. Avant que la communication puisse commencer le modèle récepteur doit être ouvert avec l'option "Open Mailslot communication automatically" activée ou alors le bouton "Open mailslot"

doit être pressé. Cela paramètre le bloc de réception de manière à ce qu'il puisse recevoir un message. L'horloge de simulation dans le modèle émetteur est toujours en avance par rapport à l'horloge du modèle récepteur.

Les entités arrivant dans le modèle récepteur ont une date (heure à laquelle elles ont été expédiées par le modèle émetteur) et sont stockées dans le bloc `Receive item from model` jusqu'à ce que l'heure du modèle récepteur coïncide avec l'heure d'émission de chaque entité.

Il y a trois types de messages qui sont envoyés d'un modèle à l'autre :

- Un contenant la commande "Run" (démarré la simulation).
- Un contenant des informations sur chaque entité transférée (numéro de série pour le message, heure de transfert, valeurs d'attribut). Seules les valeurs d'attribut sont transférées (pas les noms). Cela à cause de la longueur de 255 caractères qui limite la taille du message !
- Un contenant la commande "End" qui stoppe la simulation.

Communications entre applications : "inter-process communication"

Ce qu'on appelle "interprocess communication" ou IPC est un standard de communication entre applications. Dans ExtendSim, vous faites appel à IPC par des fonctions IPC.

Les applications qui communiquent entre elles sont qualifiées en tant que Serveur ou Client. Une application cliente demande un service à une autre application ; une application serveur répond à la demande d'une application cliente. De nombreuses applications, comme ExtendSim, prennent le rôle de client ou de serveur selon les circonstances.

Que vous utilisiez les fonctions IPC, les fonctionnalités IPC d'ExtendSim sont communes à tous les environnements et ne se différencient que par l'implémentation. Sous Windows, IPC est supporté par le DDE (Data Dynamic Exchange).

En tant qu'application serveur, ExtendSim peut recevoir et exécuter du code ModL qui est envoyé par une application cliente. Cela signifie que d'autres applications peuvent demander à ExtendSim d'effectuer une tâche que supporte le langage ModL. Sur Macintosh ExtendSim supporte les AppleEvents `OpenApplication`, `OpenDocument`, `PrintDocument`, `Quit` et `DoScript`. Sous Windows, ExtendSim supporte la fonction `DDEExecute`.

Si vous prévoyez d'écrire du code utilisable sur plusieurs plates-formes, il vous sera nécessaire de définir les arguments des fonctions IPC en relation avec la plate-forme d'utilisation. Les variables `PlatformMacintosh`, `PlatformPowerPC` et `PlatformWindows` vous aideront à rendre votre code adaptatif. Par exemple, le code qui suit établit une conversation entre ExtendSim (le client) et Excel (le serveur) aussi bien sur Macintosh que Windows :

```
if (PlatformMacintosh or PlatformPowerPC)
  Nomserveur = "XCEL";
if (PlatformWindows)
  nomServeur = "Excel";
IPCConnect (nomserveur, feuille);
```

Voici la liste des fonctions dont un argument (celui en italiques) a une syntaxe qui dépend de l'application serveur :

```
IPCConnect (nomServeur, élément)
IPCExecute (conversation, commande, élément)
IPCpoke (conversation, données, élément)
IPCRequest (chaineRet, conversation, élément)
```

Le manuel de programmation décrit en détail l'emploi de chacune des fonctions.

Les blocs concernés

La bibliothèque Value comporte des blocs qui implémentent les principales fonctions IPC du langage ModL. Les blocs sont **Command**, **Read**, **Write**, **Data Specs**, **Data Import Export**.

Conseils de modélisation

Vous avez déjà vu beaucoup d'aspects d'ExtendSim que vous utiliserez chaque jour. Une méthode de travail à mémoriser est de commencer par un modèle grossier, de tester cette première réalisation, d'affiner le modèle, de le valider, de l'affiner encore, et ainsi de suite. Votre objectif doit être d'obtenir un ensemble d'informations utilisable, et non une réplique exacte de la réalité. Aucun modèle n'est jamais terminé dans la mesure où vous pouvez toujours affiner vos prédicats.

Cette rubrique traite des sujets directement liés à la manière de réaliser un modèle. Certaines suggestions concernent l'amélioration esthétique du modèle, d'autres l'optimisation de tâches spécifiques. Toutes nous semblent utiles.

Conseils généraux de modélisation

Les étapes de la création d'un modèle

Plutôt que de vous lancer à placer tout de suite des blocs à l'écran, il est préférable de réfléchir sur ce qu'il faut faire, et suivre certaines étapes :

- Formulez le problème. Définissez le problème et formulez vos objectifs.
- Décrivez les flux. Déterminez d'où viennent et où vont les flux, dans quelle portion du modèle l'arrivée simultanée de l'information est nécessaire.
- Construisez le modèle, en utilisant les blocs ExtendSim.
- Obtenez les données. Identifiez, spécifiez et rassemblez les données nécessaires. Cela comprend des données numériques et aussi des formules mathématiques telles que la distribution d'événements aléatoires.
- Lancez la simulation. Déterminez éventuellement la précision de vos observations.
- Vérifiez les résultats de la simulation. Comparez-les à ce que vous attendez ou prévoyez.
- Validez les résultats. Comparez le modèle au système réel si c'est possible.
- Analysez les résultats. Faites vos recommandations sur la façon de modifier le système.
- Appliquez vos décisions.

Affiner un modèle

Il est important de comprendre qu'un modèle ne vous fournit pas une unique réponse "juste". Bien souvent, vous commencez par vous rendre compte d'erreurs dans la modélisation du système. Ces erreurs peuvent être des étapes manquantes, trop de simplification d'une étape, des postulats faux ou des liens manquants. Il est donc souvent nécessaire d'affiner un modèle.

Tout modèle peut être complexifié en augmentant la précision dans la description de certaines étapes, en ajoutant des hypothèses et des connexions. Il faut commencer par un modèle simple, voire simpliste. Après analyse et validation de ce modèle, vous ajoutez de la complexité puis validez à nouveau le modèle, et ainsi de suite. La complexité peut être apportée de deux manières :

- En remplaçant un bloc (un traitement simple) par plusieurs blocs (un traitement complexe).
- En ajoutant un lien entre deux blocs isolés auparavant, généralement via une opération mathématique (trouver un lien entre deux traitements).

A chaque étape, vérifiez les résultats et leur cohérence. Si un résultat est incorrect, faire du pas à pas (en utilisant par exemple la Trace) pour trouver à partir d'où le résultat est devenu faux.

Conseils pour la bibliothèque Value

Bloc Holding Tank : intégrer ou sommer ?

Dans le dialogue du bloc Holding Tank, l'utilisateur choisit si les entrées sont sommées ou intégrées.

- Si la valeur reçue en entrée se fonde sur le temps durant la simulation (un taux), il faut intégrer. Par exemple si l'entrée représente des K€ par an ou des litres à l'heure.

- Si la valeur reçue en entrée doit être additionnée à chaque pas de temps ou étape, il faut sommer. C'est le cas lorsque l'entrée représente des bons de commande ou des personnes.

La somme ajoute la valeur en entrée telle quelle. L'intégration considère que l'entrée doit être répartie sur chaque unité de temps, et n'ajoute qu'une portion de l'entrée à chaque pas de temps ou étape (ou plusieurs fois l'entrée si le pas de temps est supérieur à une unité de temps).

Le tableau suivant considère une entrée de valeur 2000 où l'unité de temps est l'année, et le pas de calcul 0,25 (trois mois).

Heure	Etape	Somme	Intégration (délai)	Intégration (sans délai)
0	0	2000	0	500
0,25	1	4000	500	1000
0,50	2	6000	1000	1500
0,75	3	8000	1500	2000
1	4	10000	2000	2500

Notez que si vous changez le pas de temps, le contenu final de la Réserve sera très différent si les entrées sont sommées, mais restera le même si les entrées sont intégrées.

Le choix de la méthode d'intégration dépend du modèle :

- Vous choisirez l'intégration avec délai lorsqu'il n'existe qu'un seul bloc intégrateur dans le modèle, ou que les blocs intégrateurs sont indépendants. C'est le cas du modèle Reservoir.
- Lorsqu'un modèle comporte plusieurs blocs d'intégration, on utilise souvent le signal émis par chaque bloc comme un facteur de correction (feedback). Ajouter un délai de réaction perturberait alors la régulation du modèle, en provoquant une instabilité souvent visible par l'allure oscillatoire d'amplitude croissante des courbes tracées.

Utilisation de NoValue pour retarder des calculs

Parce que la plupart des blocs ignorent la valeur NoValue (signifie l'absence de valeur), cela permet de modéliser facilement certaines situations. Par exemple, si vous souhaitez commencer un calcul de Moyenne/Variance après la période de montée en régime d'un modèle, vous pouvez envoyer des NoValues au bloc jusqu'à l'instant où vous voulez faire débiter les calculs. L'envoi de zéros fausserait le calcul, car ils seraient pris dans l'échantillon, tandis que NoValue est simplement ignorée.

Animation

De nombreux blocs des bibliothèques discrètes et continues comportent de l'animation, comme décrit plus haut. L'animation est visible lorsque vous activez l'option **Animation** du menu Simulation.

Vous voudrez peut-être ajouter votre propre animation à un modèle. Il existe pour cela deux blocs, qui figurent dans la bibliothèque Animation : Animate Item et Animate Value. Les icônes de ces blocs sont très simples et semblables :

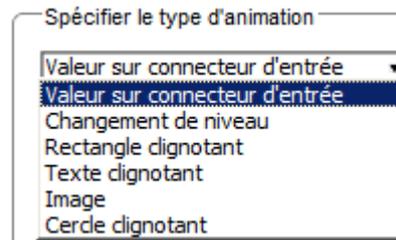


blocs d'animation

Ces blocs permettent de choisir dans le dialogue la couleur et le motif qui vont clignoter, et aussi de montrer des images et des films. Vous pouvez également utiliser ces blocs pour animer des blocs hiérarchiques, comme décrit plus loin.

Le bloc Animate Item peut être lié à des connecteurs discrets. Lorsqu'une entité entre dans le bloc, le bloc clignote avec la couleur et le motif choisis.

Le bloc Animate Value peut être lié à des connecteurs continus. Vous spécifiez dans le dialogue quand et comment le bloc doit s'animer :



dialogue de Animate Value

Quatre choix principaux existent pour déterminer le type d'animation :

- c'est la valeur reçue en entrée qui est affichée (comme le bloc Valeur de la bibliothèque Value)
- c'est un niveau qui se déplace entre deux limites
- c'est un clignotement lorsqu'un certain seuil est franchi (rectangle, cercle ou texte)
- c'est une image qui est rendue visible

Si vous choisissez un clignotement lorsqu'une entrée est supérieure ou égale à une certaine valeur, vous pouvez préciser la forme de l'animation : un cadre, un texte ou un cercle. À cause de la petite taille des icônes de bloc, le texte doit rester court. Le seuil par défaut de 0,5 sert à déterminer les valeurs vrai/faux.

Si vous choisissez une animation par un niveau, assurez-vous de spécifier des valeurs de minimum et maximum qui couvrent bien toute la plage possible de valeurs. Le bloc déplace un niveau entre le haut et le bas du bloc.

Pour les images, il faut entrer le nom d'un fichier d'une des formats suivants :

- BMP (Windows Bitmap)
- GIF (Graphical Interchange Format)
- JPG/JPEG (Joint Photographic Experts Group)
- PNG (Portable Network Graphics)
- PBM (Portable Bitmap)
- PGM (Portable Gray Map)
- PPM (Portable Pix Map)
- XBM (x11 Bitmap)
- XPM (x11 Pixmap)

Les fichiers doivent figurer dans le répertoire Extensions sous le répertoire ExtendSim AVANT de lancer l'application.

Copier des portions d'un modèle

Dans certains modèles, des associations de blocs se répètent à plusieurs endroits. ExtendSim facilite la réutilisation d'un groupe de blocs par les options Copier et Coller : sélectionnez les blocs désirés, choisissez Copier blocs dans le menu Edition, cliquez à l'emplacement de destination et choisissez l'option Coller blocs. L'option Dupliquer effectue le copier et le coller en une seule opération.

Conseils pour la bibliothèque Item

Progression des entités durant la simulation

La plupart des blocs discrets demandent une entité en entrée, traitent l'entité, puis attendent que le bloc qui les suit demande à son tour l'entité, et enfin demandent une nouvelle entité. Par exemple, un bloc tel le bloc Activity va chercher une entité dans le bloc précédent, traite l'entité, et attend qu'elle soit prélevée par un autre bloc. Il est important de comprendre la progression des entités dans le modèle afin d'éviter deux pièges : perdre des entités et immobiliser des entités.

Retenir et pousser (en sortie)

Les blocs discrets traitent de deux manières leurs entités en sortie :

- L'entité est **retenue** dans le bloc jusqu'à ce qu'un autre bloc la tire (c'est le comportement général).
- L'entité est **poussée** hors du bloc, qu'un bloc soit prévu ou non pour la ramasser.

Les blocs Create et Shutdown poussent toujours leurs entités. Tous les autres blocs retiennent leurs

entités et attendent qu'elles soient tirées.

Tirer et constater (en entrée)

Le connecteur d'entrée d'un bloc discret accède de deux manières à une entité : il peut la tirer depuis le bloc précédent (ce que fait la majorité des blocs), ou simplement constater qu'une entité attend sur le connecteur de sortie du bloc précédent. Si un connecteur d'entrée tire une entité, il peut effectuer un traitement dessus. Mais si le connecteur est passif, il n'a pas accès à l'entité, il ne fait que constater sa disponibilité.

Notez qu'un connecteur passif doit être lié à un connecteur pousseur. S'il est lié à un connecteur qui retient, il pourra percevoir l'entité mais pas l'accepter. De plus, le bloc précédent restera en attente permanente et causera un blocage.

Les connecteurs particuliers qui ne font que constater la présence d'entités sont :

- les connecteurs *sensor* et *demand* du bloc Gate.
- le connecteur d'entrée du bloc History lorsque ce dernier est ajouté en parallèle du flux (petite icône).

Pièges à éviter

Lorsqu'un bloc pousse une entité et qu'elle n'est pas ramassée, l'entité disparaît de la simulation, ce qui est rarement souhaité. Lorsqu'un bloc retient une entité et que l'entité n'est que constatée passivement (et jamais tirée), l'entité ne progresse pas dans le modèle, ce qui n'est jamais souhaité.

Bien que généralement vous ne rencontriez pas ces problèmes, vous devez être prudent dans l'association de blocs de fonctionnement différent. Voici les associations sûres et risquées :



combinaisons de blocs

Types de blocs

Chaque bloc discret manipulant des entités est identifié dans son dialogue comme étant un bloc de type stockeur, passeur ou décision.

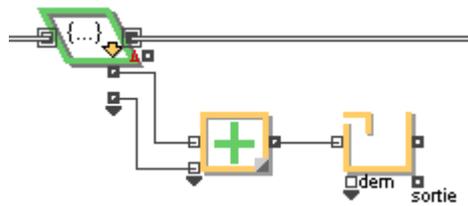
- Les blocs **stockeurs** peuvent contenir ou retenir des entités pendant un certain temps au cours de la simulation. Certains blocs stockeurs déclenchent des événements, d'autres pas. Exemples de blocs stockeurs : les blocs Queue et Activity.
- Les blocs **passeurs** passent des entités sans les retenir. Ces blocs exécutent des opérations qui ne dépendent pas du temps. Généralement, ils ne déclenchent pas d'événements. Les blocs Set et Equation(I) (qui effectue un calcul lorsqu'une entité passe) sont des exemples de blocs passeurs.
- Les blocs de **décision** orientent les entités dans le modèle. Ils choisissent un chemin en fonction de la propriété d'une entité, d'une valeur aléatoire, d'une séquence ou d'une valeur reçue par un connecteur. Selon les options sélectionnées dans le bloc, celui-ci peut retenir des entités. Par exemple, les blocs Select Item In et Select Item Out sont des blocs de type décision.

Ainsi, une entité n'entrera pas dans un bloc passeur tant que le bloc stockeur situé en aval n'aura pas de place pour l'accueillir. Pour déboguer un modèle, il est important de comprendre où les entités peuvent être stockées, ainsi que le temps nécessaire à une entité pour se déplacer d'un endroit à un autre. De plus, certaines options des blocs renvoient à des types de blocs spécifiques. Par exemple, dans le bloc Transport, vous pouvez spécifier que la distance jusqu'au bloc suivant est celle qui sépare le bloc Transport du prochain bloc non-passeur (stockeur ou de décision).

Blocs continus dans les modèles discrets

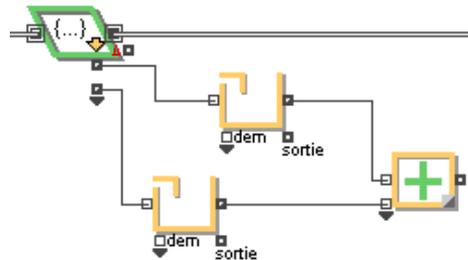
Dans un modèle discret, les blocs continus peuvent être considérés comme passifs, c'est à dire qu'ils ne se recalculent que lorsqu'un bloc discret le leur demande. Lorsque cette demande se fait, c'est par un message émis par le bloc discret, et qui est instantanément retransmis à tous les éventuels blocs continus relié au premier bloc continu (propagation du message).

Nous cherchons ici à accumuler le résultat de la somme de deux attributs, et la modélisation faite rapidement et intuitivement donnerait probablement ceci :



Mauvaise manière d'accumuler la somme de deux attributs

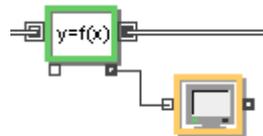
Le bloc Holding Tank aura un résultat incorrect. Chaque fois qu'une entité passe par l'un des blocs *Get*, un message est envoyé au bloc Math, et la somme est accumulée dans le Holding Tank. Cela signifie que pour une entité qui passe, deux valeurs sont envoyées dans le Holding Tank. La bonne modélisation serait celle-ci :



Bonne manière d'accumuler la somme de deux attributs

Dans cette solution, les valeurs d'attributs sont accumulées séparément, ce qui évite le "double comptage" du cas précédent, car le Holding Tank ne reçoit qu'un message par passage d'entité.

Vous pourriez de manière encore plus compacte utiliser le bloc *Equation(I)* pour effectuer d'un coup la lecture et l'accumulation des attributs.



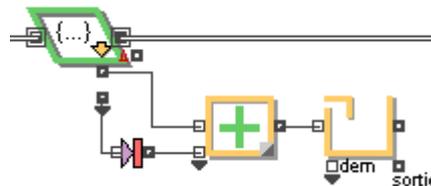
Recours au bloc Equation(I)

En entrée l'équation lit les deux attributs, et dans le corps de l'équation l'on trouve :

```
Accumule = Accumule + a + b ;
Resultat = Accumule
```

Où *Accumule* est une variable statique, et *a* et *b* les deux valeurs d'attribut.

Vous pourriez au besoin garder la première architecture et utiliser le bloc **Stop message** de la bibliothèque Utilities, pour éviter que deux messages ne soient reçus pour chaque entité.



Autre manière d'accumuler la somme de deux attributs

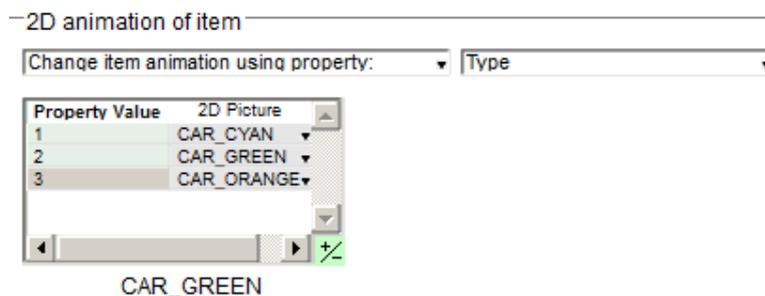
Personnaliser les symboles d'animation

Le symbole qui représente les entités dans l'animation des modèles discrets est le cercle vert. Vous pouvez choisir d'autres symboles parmi ceux proposés par ExtendSim, ou en créant les vôtres (voir liste des formats) et en les plaçant dans le répertoire Extensions\Pictures.

Dans l'emploi des symboles d'animation, il faut distinguer les blocs qui créent les entités (Create, Resource Item, etc.) de ceux qui les voient passer (files d'attente, activités, etc.). Avec les blocs créateurs, vous spécifiez le symbole dans l'onglet Item Animation, et celui-ci sera conservé pour toute la progression de l'entité, à moins que vous ne changiez de symbole via un des blocs "passeurs".

Vous pouvez également faire évoluer les symboles de manière "intelligente" en vous appuyant sur des

valeurs d'attributs, grâce à l'option **Change item animation using properties** de l'onglet Item Animation. Vous décrivez une correspondance entre valeurs d'attribut et symboles, afin de mieux distinguer visuellement les flux.



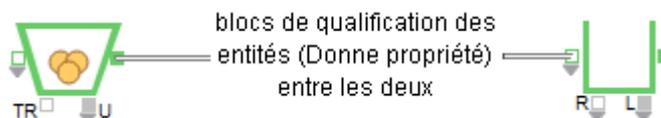
Onglet Item Animation

Utilisez le menu déroulant pour spécifier la propriété à lire. Le tableau comporte des cellules qui sont elles-mêmes des menus déroulants pour choisir parmi les symboles d'animation disponibles.

Traitement préparatoire

Parfois vous souhaitez avoir toutes vos entités disponibles au début d'une simulation, plutôt que générées au fur et à mesure. Par exemple si vous désirez un ensemble aléatoire de nombres, mais que vous vouliez les injecter triés dans le modèle. Dans les circonstances normales, la première entité parcourrait le modèle avant même la génération du second. Il faut donc créer de nombreuses entités, les stocker et les traiter, puis les relâcher ensuite.

Utilisez les blocs suivants au début de la simulation :



exemple de traitement préparatoire

Ouvrez le bloc Resource Item et attribuez pour valeur initiale le nombre d'entités que vous voulez générer. À la place du texte, placez les blocs Set ou autres, afin de traiter les entités.

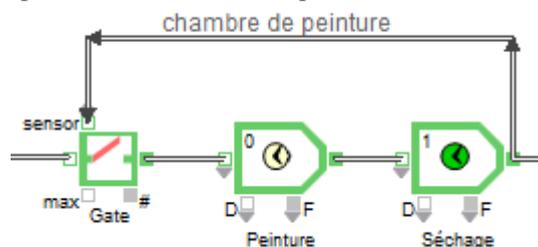
Placez les entités dans une file qui les triera selon un certain ordre.

Toutes les entités vont aller du bloc Resource Item à la file d'attente au temps zéro. Elles recevront toutes les valeurs prévues par vos blocs de qualification d'entités puis entreront dans le modèle. S'il y a de nombreuses entités dans le bloc Resource Item, vous remarquerez dans la barre de statut le mot "Initialisation". Dès que le traitement préparatoire sera terminé, le chrono de statut adoptera une valeur plus adéquate.

Restreindre des entités dans un système

Dans un modèle vous pouvez souhaiter que certains blocs formant un groupe ne puissent accueillir qu'une seule (ou un nombre limité) entité en même temps. Par exemple, dans votre atelier, la chaîne de peinture est représentée par plusieurs activités, mais dans la réalité une seule entité peut entrer dans cette chaîne à la fois. Vous pouvez faire en sorte qu'une nouvelle entité n'entre que lorsque la précédente est sortie.

Pour cela utilisez le bloc **Gate** dans son mode **Area gating** (Sas sur une portion). Ainsi le bloc fait réellement office de sas, car il laisse passer sa première entité, mais ne libère les suivantes que lorsqu'il observe la sortie d'une entité sur son connecteur *témoin*. Placez le bloc Gate au début du système, et à la fin du système faites revenir un lien depuis la sortie du dernier bloc vers le connecteur *témoin*. Le connecteur témoin n'absorbe pas les entités, il ne fait que les voir. Vous auriez par exemple :



exemple de Gate

Coûts

Dans de nombreux modèles discrets, il est primordial de suivre le coût associé à la production d'un produit, pour notamment déterminer aussi précisément que possible les facteurs qui contribuent à ce coût. En analysant ce qui contribue à ce coût (main d'œuvre, matière première, travail administratif, contrôle qualité), vous savez sur quels facteurs porter en premier vos efforts de réduction des coûts.

ExtendSim sait calculer les coûts pour trois types de facteurs, identifiables dans les dialogues des différents blocs concernés : 1) générateurs d'entités, 2) ressources, 3) activités. Dès qu'une de ces composantes de coût est définie quelque part dans le modèle, ExtendSim calcule automatiquement les coûts accumulés par chaque entité dans le modèle. Le coût accumulé est stocké dans un attribut appelé "_cost". Le coût de stockage ou de ressource est stocké dans l'attribut "_rate". Voyez plus loin des exemples.

Irrégularités de timing

La fenêtre de timing d'ExtendSim, au bas de la fenêtre du modèle, suppose que la simulation se déroule régulièrement, avec un intervalle de temps à peu près égal entre les événements. La valeur du temps restant est parfois incertaine, et pourrait même augmenter si la simulation ralentit parce que des entités sont retardées ou que de plus en plus d'événements sont générés. La barre de statut affichera parfois les mots "Initialisation" durant la période initiale d'une simulation s'il y a une forte activité. Le temps affiché peut être très important durant ces périodes transitoires dans les modèles discrets : ne vous en préoccupez pas car ce n'est qu'une estimation qui n'affecte en rien la façon dont se déroule le modèle.

Changer d'échelle pour suivre des entités

Dans de nombreux modèles discrets, le nombre d'entités à traiter dans une simulation est très important. Cela ralentit souvent l'exécution du modèle et entraîne des problèmes de place mémoire. Dans de nombreux cas, il est possible, sans fausser le modèle ni invalider les résultats, de traiter les entités sur une échelle différente : traiter par exemple des lots de 10 plutôt que 10 fois une unité. Une activité qui prenait 2 unités de temps auparavant prend désormais 20 unités de temps. Lorsque vous faites des modifications de cette nature, assurez-vous d'être homogène partout. Une autre solution est d'utiliser la bibliothèque Rate pour ces systèmes comportant beaucoup d'entités.

Connecteurs et liens

Mélanger des blocs continus et discrets

Comme vous l'avez vu plus haut, vous pouvez utiliser des blocs continus (ceux de la bibliothèque Value) dans vos modèles discrets. Tous les blocs de la bibliothèque Item nécessitent la présence du bloc Executive qui va gérer le timing du modèle de sorte qu'il soit adapté à des blocs discrets. Cela signifie que tout modèle contenant un bloc discret est un modèle discret.

Les connecteurs portent sur eux les signes de leur compatibilité. Vous pouvez relier ensemble des connecteurs discrets, ou ensemble des connecteurs continus. Vous pouvez aussi relier des connecteurs de sortie continus ou discrets à des connecteurs d'entrée mixtes.

Vous ne pouvez pas relier une sortie discrète à une entrée continue. Vous ne pouvez pas non plus relier une sortie continue à une entrée discrète.

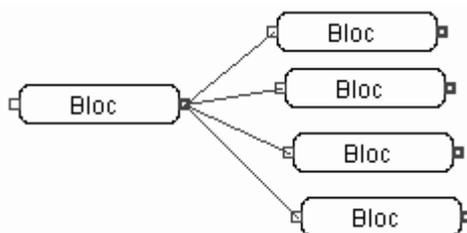
Si vous créez des blocs avec des connecteurs utilisateur, ces connecteurs ne peuvent être liés qu'à d'autres connecteurs utilisateur ou à des connecteurs mixtes.

Liens sur des entrées multiples

Vous pouvez alimenter autant de connecteurs d'entrées que vous le souhaitez à partir d'un seul connecteur de sortie dans les modèles continus et discrets. Mais le résultat est très différent dans les deux types de modèles : une valeur continue peut être transmise à plusieurs blocs, mais une entité est transmise au premier bloc qui la demande.

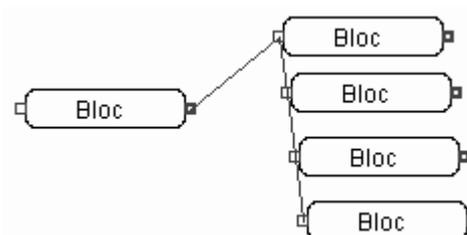
Connecteurs continus

La divergence en sortie est utile lorsque plusieurs blocs ont besoin du résultat d'un bloc précédent. Par exemple, si vous avez un connecteur de sortie relié à quatre connecteurs d'entrée, votre modèle ressemblera à :



un connecteur de sortie relié à quatre entrées

Pour des raisons esthétiques, vous pouvez souhaiter n'avoir qu'un seul lien en provenance du connecteur de sortie, par exemple parce que les blocs sont éloignés les uns des autres. Au lieu de quatre liens, vous pouvez lier la sortie à une seule entrée, puis lier les entrées entre elles (on appelle cela *en guirlande*). Votre modèle ressemblerait à :

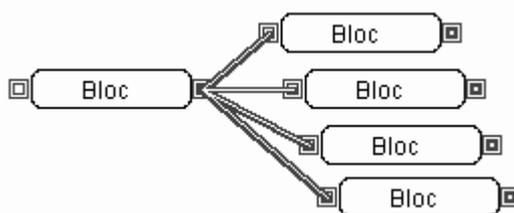


quatre entrées liées ensemble

Connecteurs discrets

Dans un modèle discret, la connexion d'une sortie discrète à plusieurs entrées discrètes a une fonction différente, à savoir le traitement d'entités en parallèle. Dans ce cas, les entités vont au premier bloc lié ; si celui-ci est occupé, elles vont au second bloc lié, et ainsi de suite.

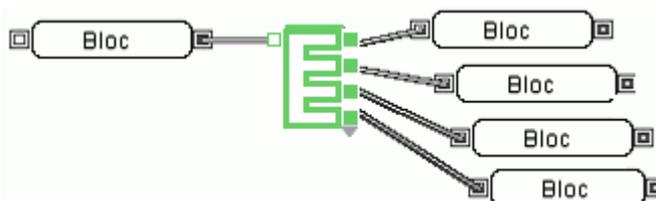
Ainsi, dans l'exemple suivant, supposons que le bloc du haut a été le premier lié, puis celui en-dessous, et ainsi de suite. Un élément ira toujours vers le bloc du haut d'abord. Si celui-ci ne peut accepter l'entité, elle sera proposée au second bloc. Cela permet de simuler plusieurs machines qui fonctionnent en parallèle, les entités étant affectées à la première machine qui peut les accepter. Vous pouvez voir l'ordre des blocs par l'option **Ordre de la simulation** du menu Modèle.



une sortie liée à 4 entrées dans un modèle discret

Vous pouvez aussi effectuer des liens en guirlande, comme expliqué plus haut. Le premier bloc relié reste celui qui reçoit l'entité.

Si l'ordre des blocs récepteurs a une quelconque importance, placez un bloc Select Item Out pour définir les préférences au cas où plusieurs blocs récepteurs sont disponibles. Préférez de toute façon cet ordre explicite aux liens directs.



blocs en parallèle avec Select Item Out

Les outils d'analyse

Il faut être conscient que la simulation en elle-même ne fournit pas des réponses exactes ni n'optimise un système. En fait, un modèle bien construit mesure un process et indique des résultats statistiques. Ces diverses mesures vous donnent l'information nécessaire pour votre analyse et votre procédure de décision.

Bien que la simulation soit une technique quantitative, elle laisse la place à l'estimation et au jugement. C'est notamment le cas lorsque vous devez construire un modèle grossier qui vous donne rapidement des tendances pour prendre une décision immédiate pouvant être affinée par la suite. La précision statistique de vos modèles dépend de la nature du problème, de l'importance de la décision et de la sensibilité du système aux données qui lui sont fournies.

Un des bénéfices d'ExtendSim est qu'il propose plusieurs méthodes d'analyse qui pourront se combiner pour prendre en compte tous les résultats : statistiques, intervalles de confiance, scénarios, analyse de sensibilité, optimisation, Stat::Fit, et outils graphiques.

☞ Gardez à l'esprit que lors que vous employez des méthodes statistique pour analyser les données issues du modèle, ce sont les résultats du modèle que vous analysez, et non le système réel. La validité de votre analyse dépend de la qualité des données ("garbage in = garbage out").

Blocs de statistiques

De très nombreux blocs fournissent des informations statistiques dans leur onglet Results.



Le bloc **Statistics** (bibliothèque Report) accumule et note des informations statistiques sur un modèle discret (bibliothèques Item et Rate). Selon son mode il observe un type particulier de blocs dans un modèle : les activités, les files, les blocs de ressources, pools, stations, réservoirs de type flux, etc. Parmi les informations notées : le label du bloc (ou son numéro), l'heure d'observation, l'utilisation, des temps moyens, etc.

Les informations sont réunies dans des tables des dialogues. Une ligne est mesurée pour chaque bloc concerné à chaque fois qu'une observation est faite. Au besoin, les nouvelles observations effacent les anciennes, pour un maximum de 250 lignes.

Pour utiliser ce bloc, placez-le n'importe où dans un modèle. Vous choisissez dans le dialogue si les observations doivent être faites continûment (ce qui réduit le temps d'exécution de la simulation) ou en fin de simulation. Vous pouvez aussi demander une observation périodique en liant un bloc de type Programme (Pulse, Create) au connecteur *go*. Pour lire en temps réel les statistiques courantes, cliquez sur le bouton *Update now*.

Activities observe les arrivées, départs et taux d'utilisation de chaque bloc de la catégorie. La manière de calculer le taux d'utilisation dépend de chaque bloc (se reporter au texte d'aide du bloc). Concerne les blocs suivants : Activity, Convey Item et Transport.

Queues observe la longueur moyenne et maximum de la file, le temps d'attente moyen et maximum pour chaque bloc de la catégorie. Concerne les blocs suivants : Queue, Queue Equation, Queue Matching.

Resource Item observe le nombre d'entités disponibles et utilisées ainsi que l'utilisation du bloc. La manière de calculer le taux d'utilisation dépend de votre paramétrage (se reporter au texte d'aide du bloc). Concerne le bloc Resource Item.

Resource Pool enregistre : utilisation, nombre disponible, nombre en utilisation et nombre en attente.

Mixed blocks concerne l'élément dont vous faites glisser un clone sur l'icône du bloc. Enregistre : nom variable, valeur et numéro de simu, pour les variables que vous sélectionnez. Faites glisser sur l'icône du bloc des clones de champs de résultat issus des dialogues de blocs. Les informations viendront s'inscrire dans le tableau.

Workstation observe les arrivées, départs et taux d'utilisation, ainsi que la longueur moyenne et maximum de la file.

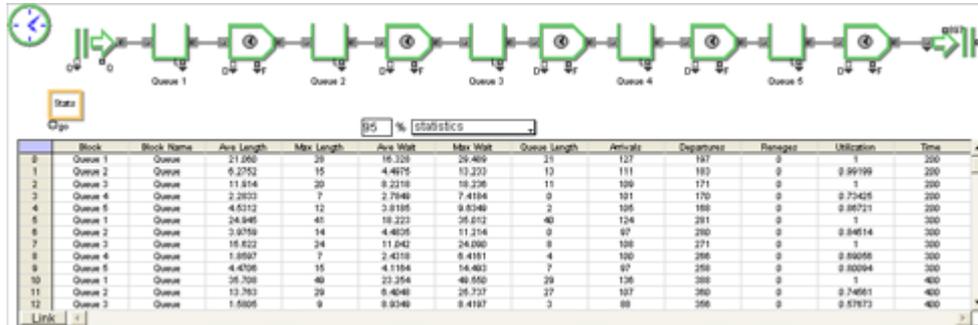
Convey flow concerne ce bloc de la bib. Rate.

Tanks fait référence aux blocs Tank et Interchange de la bib. Rate.

Le bloc **Statistics** permet de générer un intervalle de confiance à partir des données enregistrées dans la table. Pour cela il faut au moins deux observations pour chaque bloc du modèle. Le niveau de l'intervalle de confiance est spécifié sous forme d'un pourcentage (entre 80 et 99.999) dans le dialogue.

Les informations collectées peuvent être exportées : voir pour cela les options proposées dans l'onglet Export.

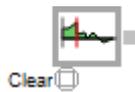
Le modèle *Queue Statistics* illustre l'emploi d'un bloc **Statistics** qui réunit des informations sur plusieurs files d'attente d'un modèle.



Le bloc **Cost By Item** (bib. Item) s'insère dans le flux et lit (et stocke dans un tableau) les valeurs des attributs "_cost" des entités qui passent par lui, en calculant le coût moyen et le coût total. Le tableau utilise soit une ligne par entité passée, soit une ligne par groupe d'entité d'un même type (défini par un attribut), ce qui permet d'obtenir des coûts et des moyennes pour des familles de produits.



Le bloc **Cost Stats** réunit l'information *Total Cost* issue de tous les blocs qui génèrent des coûts (create, ressource, activité, file), ce qui permet de déterminer rapidement quels blocs ont le plus contribué au coût de l'entité.



Le bloc **Clear Statistics** (bib. Value) remet à zéro les statistiques de divers blocs dans le modèle. C'est parfois utile après la période de montée en régime d'un modèle, quand la mise en place des flux et des processus pourrait brouiller l'analyse statistique du système en régime normal.

Intervalle de confiance

L'intervalle de confiance indique, avec un certain degré de probabilité ou de confiance, combien les résultats moyens de la simulation sont proches de la moyenne réelle du modèle. L'intervalle de confiance est la plage à l'intérieur de laquelle il est considéré que la vraie moyenne va se trouver. C'est exprimé sous la forme de la probabilité que la vraie moyenne se situe dans l'intervalle $x \pm y$ où « x » est la moyenne obtenue sur plusieurs observations, et « y » définit les bornes de la plage. Le niveau de confiance est la probabilité que la moyenne réelle soit située à l'intérieur de la plage. Le niveau de confiance sera typiquement de 90%, 95% et 99%.

Le bloc Mean & Variance (bib. Value) et le bloc Cost Stats (bib. Report) proposent des options pour spécifier un niveau de confiance dans leur dialogue.

Pour générer un intervalle de confiance où chaque échantillon est le résultat d'une unique simulation, effectuez une des actions suivantes dans l'onglet *Nombres aléatoires* du dialogue Paramètre de la simulation :

- Demandez un aléatoire vrai en saisissant 0 ou en laissant vide la base aléatoire.
- Ou bien demandez *Continuer la séquence aléatoire* dans le menu déroulant.
- Ou bien sélectionnez *Utiliser les valeurs de la table __seed* dans le menu déroulant.

Pour obtenir un échantillon suffisamment large pour déterminer l'intervalle de confiance, chaque statistique doit être ajoutée à la table des données. Vous en trouverez un exemple dans le modèle "Run for CI" situé dans ExtendSim/Exemples/Tips/Modeling Tips, s'appuyant sur le bloc Mean & Variance.

Analyse de sensibilité

L'analyse de sensibilité permet de voir l'effet de la variation d'une variable en permettant de lancer plusieurs simulations de suite en modifiant à chaque fois la valeur d'un paramètre. Cela permet de tester complètement l'implication d'une variable sur le système et de repérer les résultats extrêmes ainsi que la moyenne d'apparition de certaines valeurs critiques.

Chaque fois que vous utilisez une distribution aléatoire pour déterminer une valeur et que vous lancez

plusieurs fois la simulation, vous faites de l'analyse de sensibilité sans le savoir. ExtendSim permet de spécifier explicitement la manière dont certains paramètres doivent changer.

L'analyse de sensibilité fonctionne avec tous les champs numériques de saisie (les rectangles dans lesquels vous entrez vos paramètres numériques), ainsi qu'avec les clones de ces champs. Vous entrez les paramètres de sensibilité pour une valeur en utilisant une touche spéciale lorsque vous cliquez sur l'élément ou son clone. Vous pouvez faire varier autant de valeurs de dialogue que vous le voulez. Nous conseillons cependant de ne travailler que sur une ou deux valeurs à la fois pour plus de clarté dans l'analyse des résultats.

Lorsque vous avez associé un "scénario" à une valeur numérique, vous devez spécifier combien de simulations vous voulez lancer.

Voici les étapes à suivre pour faire une analyse de sensibilité :

- Ouvrez le dialogue du bloc contenant la valeur que vous voulez faire varier.
- Cliquez sur la valeur en maintenant la touche Contrôle (Ctrl) enfoncée. Vous pouvez aussi utiliser la commande Paramètres de sensibilité du menu clic droit, ou Paramètres de sensibilité du menu Modèle. Le dialogue suivant s'affiche :

Dialogue Sensibilité

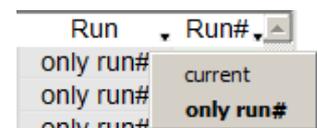
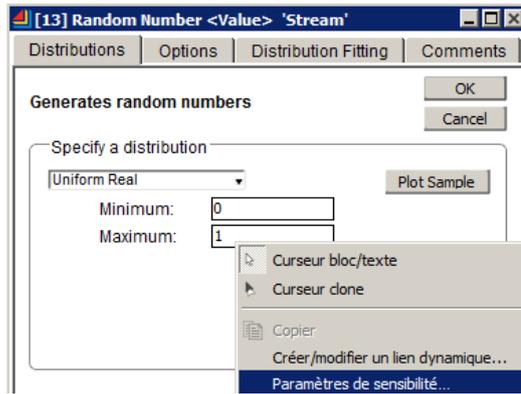
- Assurez-vous que la case "Activer la sensibilité" est cochée.
- Spécifiez le nombre de réplifications nécessaires pour l'analyse.
- Choisissez une méthode d'analyse de sensibilité (un fichier, des bornes, une distribution aléatoire ou des valeurs d'un BDD, comme décrit plus loin) et entrez les valeurs nécessaires.
- Assurez-vous que la commande Analyse de sensibilité du menu Simulation est cochée. Sinon, sélectionnez la commande pour qu'une marque de cochage figure à sa gauche.
- Lancez la simulation.

Exemple d'analyse de sensibilité

Considérons que vous vouliez faire varier l'apport d'une des sources de liquide dans l'exemple "Reservoir 1".

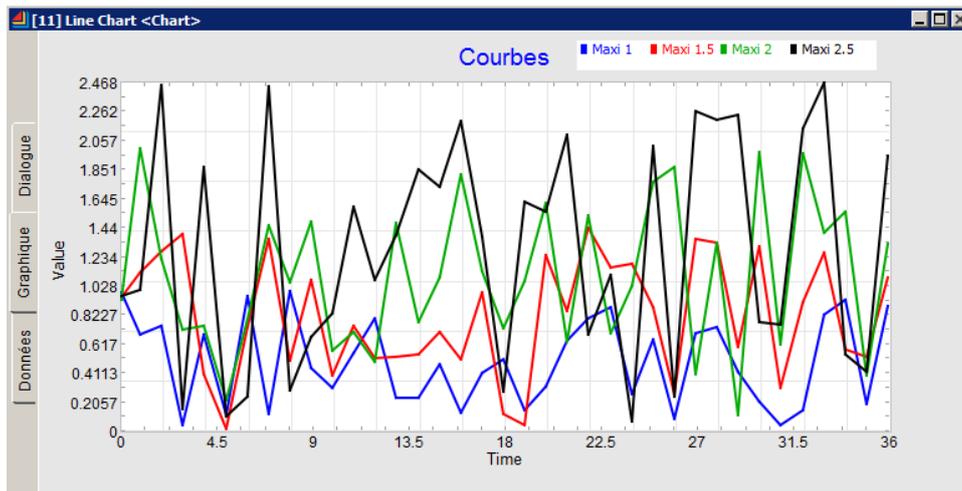
- Ouvrez le modèle " Reservoir 1".
- Utilisez la commande Enregistrer sous du menu Fichier pour enregistrer le modèle sous le nom "Reservoir Sensible".
- Ouvrez le bloc Random Number par un double-clic.

- En vous plaçant sur le champ correspondant au maximum de la distribution, choisissez la commande Paramètres de sensibilité du menu clic droit. Le dialogue de sensibilité s'ouvre. Dans le dialogue, pour « Définir Paramètres= » indiquez 4 réplifications. Ce chiffre sera repris dans le dialogue Paramètres de la simulation.
- Choisissez "Initialement", laissez la valeur initiale de 1, et demandez "modifié par" 0.5. Ainsi, la borne haute augmente de 0.5 à chaque simulation.
- Cliquez sur OK pour accepter ces valeurs, puis à nouveau sur OK pour valider le bloc. Observez que le champ « sensible » a maintenant un cadre vert.
- Ajoutez un Line Chart à la droite du bloc Random Number, déployez 4 connecteurs que vous relierez chacun au connecteur de sortie du bloc Random Number. L'objectif est de voir plusieurs courbes de plusieurs réplifications sur un même graphe, et donc de voir l'impact de l'augmentation du débit sur le niveau du réservoir.
 - Dans l'onglet Data Collection du dialogue du graphique, choisissez 'only run#' dans le menu proposé par la colonne Run. Chaque tracé sera associé à une réplification et une seule.
 - Dans la colonne Run# saisissez les numéros de 0 à 3 (la première réplification a le numéro 0, et pas 1). Fermez le dialogue du Line Chart.



Data collection options				
Trace Na...	Con	Offset B)	Run	Run#
0 trace 0	<input checked="" type="checkbox"/>	0	only run#	0
1 trace 1	<input checked="" type="checkbox"/>	0	only run#	1
2 trace 2	<input checked="" type="checkbox"/>	0	only run#	2
3 trace 3	<input checked="" type="checkbox"/>	0	only run#	3

- Sélectionnez la commande Lancer dans le menu Simulation. ExtendSim exécute la simulation quatre fois (0 à 3).
- Le graphique avec une courbe par réplification s'affiche. Vous pouvez voir la variation entre chaque tirage aléatoire sur les quatre courbes.



Lin Chart avec les résultats de l'analyse de sensibilité

Notez que l'allure de la courbe est globalement la même.

Spécification de l'analyse et du nombre de réplifications

Vous pouvez indiquer qu'un nombre sur lequel porte la variation est modifié de façon incrémentale, aléatoire ou d'après des valeurs contenues dans un fichier. Les choix sont :

Option	Description
Depuis la	Affecte des valeurs lues dans un fichier texte. C'est l'option que vous utiliserez le

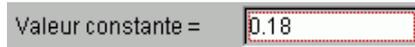
colonne X du fichier	plus souvent lors de vos expérimentations. Le fichier texte peut avoir été créé par ExtendSim ou tout autre programme. S'il comporte plusieurs colonnes séparées par des tabulations, vous pouvez spécifier la colonne de votre choix. Cliquez sur le bouton "Parcourir" pour entrer le nom du fichier à lire.
Initialement	Spécifie la valeur de début et le pas du changement. Vous pouvez augmenter la variable en spécifiant un nombre positif, ou la diminuer par un nombre négatif.
Distribution aléatoire	La variation est soumise à une loi de distribution. C'est une manière facile de faire varier une valeur sur plusieurs simulations tout en conservant la même valeur lors d'un seul lancer. Choisissez l'un des cinq types de distribution et entrez les paramètres de la distribution sur la droite. La "Base" (ou Seed) est le nombre à utiliser dans le générateur de nombres aléatoires.
Champ de BDD	Sélectionnez successivement la BDD, Table et Champ, dont les valeurs d'enregistrement seront prises à chaque réplication.

Vous pouvez spécifier le nombre de simulations soit dans le dialogue de sensibilité, soit dans le dialogue Paramètres de la simulation. Le dernier nombre entré dans l'un de ces deux dialogues est celui qui est retenu pour la prochaine simulation. Si vous utilisez la programmation, la variable CurrentSense vous permettra de modifier le nombre de jets depuis un bloc.

Activer/désactiver la sensibilité

Une commande de menu permet d'activer/désactiver la sensibilité pour tout un modèle. Pour agir sur une valeur particulière, vous devez passer par le dialogue où figure cette valeur. Lorsque vous entrez les paramètres de sensibilité pour une valeur, la variation est active tant que la case "Activer la sensibilité" est cochée dans le dialogue Sensibilité. La case non cochée désactive de manière temporaire la variation, mais vous n'aurez pas à entrer à nouveau les paramètres si vous réactivez la sensibilité.

Une valeur numérique ayant des paramètres de sensibilité associés comporte un cadre supplémentaire. Par exemple :



Cadre dans l'élément, indiquant la présence de paramètres de sensibilité

Si la variation est activée pour un élément (c'est à dire que le choix "Activer la sensibilité" est coché), le cadre est vert. Si elle est désactivée (pour cet élément ou pour tout le modèle), le cadre est rouge. Pour supprimer toute capacité de variation pour un élément, ouvrez le dialogue Sensibilité puis cliquez sur le bouton Effacer.

La commande **Ouvrir blocs sensibles** du menu Modèle affiche tous les dialogues des blocs comportant des paramètres de sensibilité, quel que soit l'état de la case "Activer scénario". C'est une commande très utile si vous avez entré de nombreux paramètres de sensibilité dans un grand modèle.

Note : modifier la valeur d'un élément "variable" dans le dialogue d'un bloc désactive la sensibilité pour ce bloc. Lorsque c'est le cas, le choix "Activer la sensibilité" est désélectionné. ExtendSim considère que si vous modifiez la valeur, c'est que vous voulez utiliser une nouvelle valeur, et non plus celle donnée par l'analyse de sensibilité. Pour désactiver durablement l'analyse de sensibilité pour un nombre, ouvrez le dialogue Sensibilité de l'élément et cliquez sur le bouton Effacer. Cela empêchera la modification accidentelle de la valeur lors d'une prochaine simulation.

Variation multidimensionnelle

Vous pouvez activer la sensibilité sur plusieurs valeurs simultanément. Par exemple, vous pouvez faire varier la valeur de deux blocs Constant afin de voir l'interaction entre les deux éléments. Si vous déterminez les deux valeurs dans l'option "Initialement", les deux éléments vont s'incrémenter de la même manière. Par exemple, si vous avez un élément qui débute à 5 et s'incrémente de 1, que le second élément commence à 100 et s'incrémente de 50, et que vous lancez la simulation sept fois, les couples de valeurs prises en compte seront les suivants :

Réplication numéro	Variable 1	Variable 2
0	5	100
1	6	150
2	7	200
3	8	250

4	9	300
5	10	350
6	11	400

Souvent pourtant, vous voulez examiner toutes les paires possibles pour les deux variables (ou davantage). Dans cet exemple, vous voulez exécuter le modèle 36 fois, avec les couples de valeurs suivants :

Réplication numéro	Variable 1	Variable 2
0	5	100
1	5	150
2	5	200
...
7	6	100
8	9	150
...
35	11	400

Afin de pouvoir effectuer ce type d'analyse multidimensionnelle, vous devez lire les valeurs dans un fichier. La manière la plus simple est de créer un fichier comportant deux colonnes séparées par une tabulation, chaque couple de valeur occupant une ligne. Ainsi, le fichier de notre exemple commencerait par :

```
5 100
5 150
5 200
...
```

Dans le dialogue Sensibilité, vous devez sélectionner le bouton radio *Depuis la colonne X du fichier* et choisir le nom de fichier. Pour la première variable, entrez "1" pour numéro de colonne ; pour la seconde variable, entrez "2".

Lorsque vous lancez une analyse multidimensionnelle, vous connectez généralement un bloc Write (bib. Value) pour écrire les valeurs que vous voulez comparer ensuite. Dans le bloc Write, sélectionnez dans l'onglet Options "Record index is equal to run number" pour incrémenter les lignes écrites en fonction de la réplication. Si le volume de données est important, vous pouvez demander à ce que les valeurs soient écrites à la fin de chaque simulation en cochant "Store data as a local copy and export to (destination) at end of run" dans l'onglet principal.

L'optimisateur

L'optimisateur d'ExtendSim est un outil performant qui permet déterminer automatiquement des paramètres correspondant à une solution optimale. Il accomplit ceci en lançant la simulation à plusieurs reprises jusqu'à atteindre une solution acceptable. L'optimisateur livré avec ExtendSim utilise un algorithme génétique afin de réduire le nombre de simulations nécessaires.

Introduction à l'optimisation

L'optimisation, également connue sous le nom de "goal seeking", est une technique utile pour trouver automatiquement la meilleure réponse à un problème. Le "problème" est habituellement décrit par une fonction objective, généralement équivalente à des coûts ou profits, qu'on cherche à maximiser ou minimiser.

La contrepartie de l'optimisation est la nécessité de lancer beaucoup d'instances de la simulation pour trouver une solution qu'on aurait pu trouver dans certains cas manuellement avec moins de simulations. De plus il se peut que l'optimisateur converge vers une solution optimale localement. En effet, il n'existe pas d'algorithme d'optimisation qui garantisse de trouver une solution optimale en un temps fini. Plus on accorde de temps à l'optimisateur, plus on aura de chance de converger vers une solution optimale.

L'algorithme

La plupart des algorithmes capables de résoudre des problèmes stochastiques (c'est à dire avec une composante aléatoire) utilisent une population initiale de solutions possibles. Chaque solution est évaluée en lançant la simulation à plusieurs reprises, ce qui permet de générer des moyennes et de trier

les solutions. Les meilleures solutions sont alors utilisées pour générer des solutions légèrement différentes qui pourraient s'avérer meilleures. Chaque nouvelle solution dérivée est appelée **génération**. Cette démarche est répétée jusqu'à ce que probablement plus aucune meilleure solution ne soit en vue.

Le problème de tout algorithme d'optimisation est l'incapacité de dire si une solution optimale a été atteinte ou même envisagée. Une bonne approche est d'autoriser l'algorithme à continuer la génération de nouvelles solutions "assez longtemps" pour déterminer si les solutions convergent. De plus, lancer plusieurs fois une même optimisation permet de déterminer si l'algorithme aboutit toujours à des solutions semblables, ce qui permet de réduire le risque d'avoir abouti à une solution sous-optimale.

Mise en place de l'Optimisateur

Voici les étapes nécessaires à la mise en place de l'optimisateur :

- Ouvrir le modèle dans lequel vous souhaitez utiliser l'optimisateur.
- Placer le bloc Optimizer de la bibliothèque Value dans le modèle.
- Définir la fonction objective à minimiser ou maximiser.
- Déterminer les paramètres nécessaires pour cette équation.
- Avec l'outil de clonage, faire glisser les variables depuis les dialogues des blocs sur le bloc Optimizer.
 - Pour les éléments de tables de données, définir la ligne et la colonne (ex. 0,1) dans la table des variables.
- Définir les limites pour ces variables dans l'onglet *Objectives*.
- Entrer la fonction de profit ou de coût dans le dialogue de l'optimisateur.
- Si une variable est sujette à une contrainte fonction d'autres variables, ajouter les équations dans l'onglet *Constraints*.
- Choisir les réglages par défaut de l'Optimisateur pour un modèle aléatoire ou non-aléatoire.
- Dans l'onglet *Results* appuyer sur le bouton **New Run** pour lancer l'optimisation.

Onglet Run Parameters

Paramètres de l'optimisateur

Table des variables

Lorsqu'une variable d'un dialogue est déposée sur un bloc Optimizer fermé (en utilisant l'outil de clonage), l'Optimisateur place cette variable dans le Tableau des variables. Cette table contient les variables utilisées dans la fonction de coût ou de profit. Ces variables peuvent être de deux types :

- Variables de décision à optimiser. Celles-ci nécessitent des limites inférieures et supérieures.
- Variables de sortie du modèle. Pour celles-ci les limites ne devraient pas être définies.

Remarque : s'il s'agit d'une variable entière, n'utilisez pas de point décimal dans les limites. En contrepartie, les limites pour des variables réelles doivent être entrées avec des points décimaux (par exemples 1.0 à 2.0).

Utilisation d'éléments de tables de données

Si une table de données est déposée sur le bloc Optimizer, il est nécessaire d'indiquer les indices de ligne et de colonne, séparés par une virgule. Comme la numérotation se fait à partir de zéro il faudra entrer par exemple 0,1 pour utiliser l'élément à la première ligne de la deuxième colonne du tableau.

Fonctions objectives

L'optimisateur peut maximiser un profit, minimiser des coûts ou approcher une constante, selon la forme de l'équation. Par exemple, on peut optimiser le nombre de machines utilisées pour maximiser le profit. Dans ce cas on peut définir une équation du type :

```
ProfitMax = production * eurosParUnité - nombreMachines * eurosParMachine ;
```

Ou bien, cela peut se traduire sous forme de coûts qu'on cherchera à minimiser :

```
CoutMini = nombreMachines * eurosParMachine - production * eurosParUnité ;
```

Les noms des variables utilisées dans ProfitMax ou CoutMini sont définis dans la colonne

Label de la Table des variables

Selon l'utilisation de ProfitMax ou CoutMini, l'Optimisateur cherchera à maximiser ou minimiser la fonction objective.

Comme dans les blocs Equation, les équations peuvent être écrites sur plusieurs lignes, utiliser des structures (IF-ELSE, FOR ...) et faire appel à des fonctions ModL. Pour plus d'informations sur les équations, reportez-vous au bloc Equation.

Fonctions objectives spéciales – approcher une constante

Un cas spécial pourrait être la nécessité de s'approcher le plus possible d'une constante. Dans ce cas on peut utiliser une équation de profit sous l'une des formes suivantes :

Dans le cas où l'on cherche à s'approcher d'une constante K non nulle :

```
ProfitMax = 1.0 - RealAbs(1.0 - équation / K) ;
```

Dans le cas où on cherche à s'approcher de zéro :

```
ProfitMax = ValeurConstante - RealAbs(équation) ;
```

Remarque : dans ce cas ValeurConstante devra être petite mais suffisamment grande pour que la plupart des solutions soient initialement positives. Si ValeurConstante est trop petite, le calcul de la convergence ne sera pas correct car il y aura des valeurs positives et négatives dans la population finale des solutions. Si ValeurConstante est correcte, toutes les valeurs vont devenir positives. Si ValeurConstante est trop grande, le calcul de la convergence sera insensible et élevé, entraînant un arrêt prématuré de l'optimisation.

Paramètres d'optimisation

Dans la plupart des cas, les paramètres par défaut selon le type de modèle (aléatoire ou non-aléatoire) devraient donner des résultats satisfaisants. L'explication de tous les paramètres peut être trouvée dans l'aide en ligne du bloc.

Contraintes

Les **contraintes individuelles** permettent de définir la valeur d'une variable en fonction d'autres variables. Elles sont définies comme des équations, utilisant généralement la structure IF ou IF-ELSE:

```
If (TailleFileMax > 7)
NouvTailleFileMax = NombreActivités + 3 ;
```

Dans certains cas il peut juste être nécessaire de modifier la valeur d'une variable. Par exemple si on souhaite contraindre une variable à une valeur multiple de 0,5. On réalise ceci en multipliant la variable par 2, puis en ajoutant 0.5 avant d'appeler la fonction Int() pour tronquer la valeur obtenue et la convertir en entier ; finalement on divise le résultat par 2. On obtient ainsi uniquement des multiples de 0.5.

```
NouvVar2 = Int (Var2*2.0+0.5) /2.0 ;
```

Dans tous les cas la valeur de NouvVarN remplace la valeur de l'ancienne variable VarN après l'évaluation de chaque contrainte. Ainsi les contraintes individuelles dépendent les unes des autres dans l'ordre de calcul de haut en bas.

Les **contraintes globales** sont utilisées pour rejeter un cas tout entier si toutes ou certaines des variables de décision ne satisfont pas un critère spécial. Les contraintes globales sont définies sous forme d'équation, pour assigner la valeur TRUE à la variable REJET si le cas doit être rejeté.

```
If (Var3+Var5>7)
REJET = TRUE ;
```

Quelquefois il peut être nécessaire d'utiliser des formes plus complexes :

```
If (Var4+Var5>7 || Var4<2)
REJET = TRUE ;
```

Dans tous les cas, l'optimisateur tentera jusqu'à 500 fois de générer des cas non rejetés. Si on n'obtient pas de solution possible alors l'optimisateur interrompt l'algorithme avec un message d'erreur. Dans ce cas il est fort probable que la contrainte globale contienne une erreur.

Résultats

L'onglet Results donne la population entière de solutions, avec la meilleure dans la ligne 0. Au cours de l'optimisation, de nouvelles et meilleures solutions remplacent les solutions dans le tableau.

Si l'optimisateur finit l'algorithme, que ce soit à cause de la condition de convergence ou parce qu'il a atteint le nombre maximum de cas, le modèle est paramétré avec la meilleure solution trouvée.

Analyse de scénarios

L'analyse de scénarios est une méthode pour examiner systématiquement et stratégiquement le résultat de différentes configurations d'un modèle. Le but est de permettre l'exploration et l'analyse d'alternatives, de se faire une idée de pourquoi votre système se comporte de cette façon et de comment il peut être amélioré et contrôlé. ExtendSim facilite l'analyse de scénarios grâce au bloc **Scenario Manager** (bibliothèque Value) qui peut être ajouté à n'importe quel modèle pour contrôler tous les aspects de l'analyse. Le Scenario Manager garde une trace de plusieurs hypothèses, toutes basées sur le même modèle. Il propose une structure très flexible pour l'expérimentation et l'analyse.

Le Scenario Manager d'ExtendSim est un système de stockage des données et d'analyse qui aide les analystes dans leur spécification, visualisation, gestion et analyse de divers scénarios. Il permet d'utiliser un seul modèle pour explorer et évaluer un nombre illimité d'options tout en apportant un contrôle précis sur les changements du modèle ainsi qu'un accès facile aux résultats. Il fournit également des méthodes automatisées pour modifier rapidement des informations variables et pour générer des rapports sur les scénarios. Il fait ceci en assignant un nom de scénario à chaque combinaison de variables que vous choisissez d'examiner. Par exemple, vous pouvez créer les hypothèses de valeurs en entrée pour le meilleur cas, le pire cas et le cas typique. Ensuite rappelez chacun de ces scénarios pour voir comment le modèle réagit suivant différentes conditions. Vous pouvez aussi les rappeler en les combinant, comme par exemple meilleur cas pour les ventes et pire cas pour les coûts.

La BDD d'ExtendSim est une partie centrale du stockage des données et de l'organisation du Scenario Manager. Une base de données appelée Scenario DB est automatiquement créée quand un bloc Scenario Manager est ajouté au modèle. Cette BDD est utilisée pour stocker les données à traiter et les résultats pour les scénarios. Le Scenario Manager facilite aussi l'utilisation d'une méthodologie de type « Plan d'expérience » (Design Of Experiments).

Le Gestionnaire de scénario est un des trois outils d'analyse d'ExtendSim ; voir « Les options pour JMP et Minitab requièrent d'avoir les applications JMP de SAS Corporation ou Minitab de Minitab Inc. pour générer le plan et affiche les résultats (version compatible 64-bits).

☞ Les méthodes JMP custom et Minitab optimal produisent bien moins de scénarios que le plan factoriel complet.

Comparaison des méthodes d'analyse » pour les différencier. Le GS répondra aux besoins de la plupart des modélisateurs d'ExtendSim, et son code est ouvert, comme pour la majorité des blocs ExtendSim. Ainsi un programmeur peut le modifier et l'améliorer si le besoin s'en fait sentir.

Fonctionnement du Scenario Manager

Pour gérer les scénarios vous évaluez et comparez un nombre de configurations du modèle ou *scénarios*, où chaque scénario est un ensemble de données en entrée ou *facteurs*. Ces facteurs sont en général des paramètres de dialogue (tel que le nombre maximum d'entités dans une Activité) mais ils peuvent aussi être des BDD, des tables, des champs ou des enregistrements. Un scénario peut avoir n'importe quel nombre de facteurs et chaque facteur peut prendre n'importe quel nombre de valeurs, ou *niveaux*. (Par exemple, définir le nombre maximum d'entités dans l'Activité à 2, 6 ou 10 ferait 3 niveaux).

Au début de chaque série de simulation, ou *réplications*, le Scenario Manager copie les facteurs pour le scénario en cours aux endroits appropriés dans le modèle. Généralement chaque scénario est simulé plusieurs fois pour prendre en compte tout comportement stochastique. À la fin de chacune de ces réplications, les résultats appelés (réponses) sont générés, enregistrés et résumés. Une fois toutes les réplications terminées, les résultats du scénario sont mémorisés dans des tables de la BDD Scenario DB. Vous pouvez aussi générer un rapport et exporter les données dans une BDD ExtendSim différente, une feuille de travail Excel ou dans JMP.

☞ Il vaut mieux prévoir du temps pour simuler tous les scénarios. Selon la taille et la complexité cela peut prendre de plusieurs minutes à quelques heures pour terminer l'ensemble des simulations. Faites attention à ne pas créer trop de scénarios : c'est particulièrement tentant si vous créez un Plan factoriel complet.

Didacticiel s'appuyant sur les paramètres de dialogue

La plupart des analystes n'utilisent que des paramètres de dialogue dans leurs scénarios, donc ce didacticiel se concentre sur ceux-ci. Pour une analyse impliquant les BDD, les tables, et ainsi de suite, voir [plus loin](#).

L'exemple suivant utilise le modèle du didacticiel sur les événements discrets pour observer comment des changements dans un certain nombre de facteurs affectent le temps d'attente moyen et le débit.

La table ci-dessous montre les facteurs du modèle et leurs variations :

Facteur	Valeurs
Temps pour le lavage seulement	5, 6, 7
Temps pour le lavage + polissage	6, 7, 8
Intervalle d'arrivée des voitures	3, 3.5, 4, 4.5, 5

Chaque scénario sera simulé 5 fois pour prendre en compte la variation des résultats entre une simulation et la suivante. Pour évaluer toutes les combinaisons possibles il faut faire 225 réplications (45 scénarios ((3 x 3 x 5 niveaux)) simulés 5 fois chacun).

☞ Heureusement, la simulation de ce modèle est très rapide donc le nombre de réplications n'est pas un problème. Cependant, si le modèle prenait beaucoup plus de temps à simuler ou s'il y avait plus de facteurs, il faudrait utiliser un Plan d'expérience (PE) pour réduire le nombre de scénarios.

Etapas

Les étapes pour effectuer une analyse de scénarios sont :

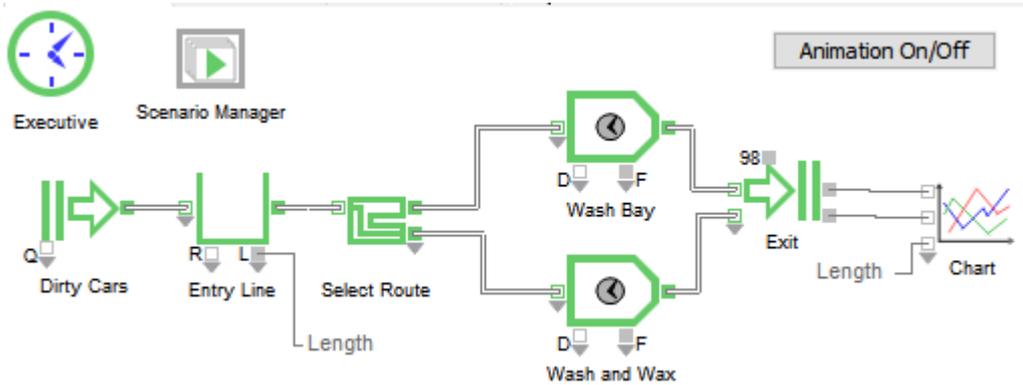
- 1) Ajouter un bloc Scenario Manager (bibliothèque Value) à un modèle
- 2) Identifier et ajouter les facteurs (valeurs en entrée) à inclure dans l'analyse
- 3) Identifier et ajouter les réponses (résultats) à analyser
- 4) Déterminer ce que vous voulez dans le rapport
- 5) Déterminer le plan d'expérience, puis générer et simuler les scénarios
- 6) Analyser les résultats
- 7) Optionnel. Exporter les scénarios dans JMP, Excel ou une BDD ExtendSim, pour une analyse plus poussée

Ajouter le bloc Scenario Manager

- Ouvrez le modèle nommé *Step 4 DE Tutorial* (\Exemples\Tutorials\Discrete Event)
- Enregistrez le modèle sous le nom « Avec scénarios » pour ne pas écraser le modèle original
- Placez un bloc Scenario Manager dans le modèle. Le bloc peut être placé n'importe où mais il sera plus facilement accessible en haut du modèle.

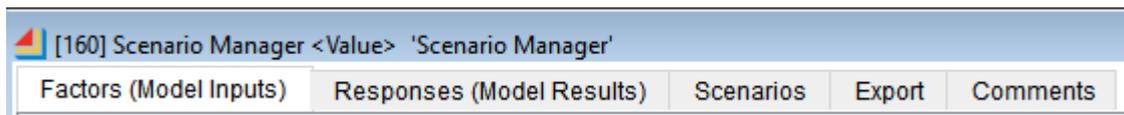
Comme indiqué en bas du menu BDD, ajouter un bloc Scenario Manager dans un modèle crée automatiquement une BDD nommée Scenario DB. C'est ici que les résultats de chaque scénario, ainsi que les facteurs du modèle, sont mémorisés individuellement.

Votre modèle devrait maintenant ressembler à la capture d'écran suivante :



Modèle avec un bloc Scenario Manager

- Ouvrez le dialogue du bloc Scenario Manager.



Dialogue du bloc Scenario Manager

Comme indiqué ci-dessus, le Scenario Manager possède des onglets pour les Facteurs, les Réponses, les Scénarios et l'Exportation. Ceux-ci représentent les étapes d'utilisation du Gestionnaire de Scénarios.

Identifier et ajouter des facteurs et des valeurs

L'étape suivante sert à déterminer et à ajouter à la table des Facteurs tous les facteurs (paramètres du modèle) que vous voulez inclure dans l'analyse. Les informations se trouvent dans la table suivante.

Nom du bloc (Label)	Paramètre	Nom du facteur	Valeurs
Activity (Wash)	Délai (D)	Temps pour servir l'essence	5, 6, 7
Activity (Wash+Wax)	Délai (D)	Temps pour servir l'essence et laver	6, 7, 8
Create (Dirty cars)	Moyenne	Heure d'arrivée des voitures	3, 3.5, 4, 4.5, 5

Méthodes pour référencer des variables de dialogue comme facteurs

Il y a 3 façons d'ajouter une variable en tant que facteur à la table des Facteurs :

- Maj-clic. Maintenez Maj et cliquez sur le paramètre cible dans le dialogue du bloc cible puis sélectionnez dans le menu où elle devrait aller.
- Clone. Faites glisser un clone du paramètre cible et déposez-le sur l'icône du Scenario Manager.
- Manuel. Entrez manuellement un nom de variable dans la table des facteurs.

Facteur sur l'intervalle inter arrivées

→ Cette section utilise Maj-clic pour associer les paramètres avec le Scenario Manager.

- Dans le dialogue du bloc Create, tout en appuyant sur Maj, cliquez sur la variable après *Mean* (4)
- Dans le menu déroulant qui apparaît, sélectionnez **Scenario Manager: Add Factor**.

mean: 4
location: 0

Scenario Manager: Add Factor
Scenario Manager: Add Response

- Dans le dialogue qui apparaît, nommez le facteur **Voitures**. Ensuite cliquez sur OK

Paramètre utilisateur

Enter factor name for dialog parameter Rnd_Arg1_prm from block Create number 3

Voitures

OK Annuler

- Allez dans l'onglet Factors du Scenario Manager

Remarquez que la table des Facteurs contient maintenant toutes les informations sur le facteur Voitures (le bloc d'où il provient, la valeur actuelle de variable, et ainsi de suite).

Par défaut, la valeur minimum de Voitures est définie à sa valeur actuelle, la valeur maximum est de 1 de plus et le pas est 1. Ces nombres seront changés plus tard dans ce didacticiel.

Autres facteurs

- Répétez les mêmes étapes pour placer les deux facteurs restants (délai lavage, délai polissage+lavage) dans la table des Facteurs du Scenario Manager. Voir ci-dessus pour les paramètres à utiliser et dans quels blocs ils se trouvent.

Entrer des valeurs

Maintenant que les facteurs ont été entrés, il est temps d'entrer également leurs valeurs envisagées. Il y a deux méthodes :

- Directement dans les colonnes de la table des Scénarios. Pour faire cela, entrez une valeur pour chaque facteur et chaque scénario dans la table des Scénarios de l'onglet Scénarios.
- Automatiquement en utilisant la méthode de Plan d'expérience. Pour utiliser les caractéristiques du PE du Scenario Manager, vous devez choisir le minimum, le maximum et le pas (l'incrément entre les valeurs minimum et maximum) pour chaque variable de facteur du dialogue.

Puisque cet exemple utilise le plan factoriel complet :

- Allez dans l'onglet Facteurs du bloc Scenario Manager
- Entrez les nouvelles valeurs Minimum, Maximum et Step pour chacun des facteurs comme indiqué ci-dessous

Nom du facteur	Minimum	Maximum	Pas
Délai lavage	5	7	1
Délai lavage +polissage	6	8	1
Intervalle d'arrivée	3	5	0.5

La table des Facteurs devrait maintenant ressembler à ceci :

Factor Name	Block Name	Block Number	Block Label	H-Block Name	H-Block Label[num]	Dialog Var	Row,Column	Var Value	Min Value	Max Value	St
1 Voitures	Create	3	Dirty Cars		[-1]	Rnd_Arg1_prm	-1,-1	3	5	5	0.5
2 Lavage	Activity	28	Wash Bay		[-1]	WaitDelta_prm	-1,-1	5	7	7	1
3 Lavage+Pol	Activity	16	Wash and Wax		[-1]	WaitDelta_prm	-1,-1	6	8	9	1

Portion de la table des Facteurs

Identifier et ajouter des réponses ciblées

L'étape suivante consiste à ajouter des réponses dans le Scenario Manager.

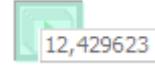
- ☞ Bien que vous puissiez utiliser une des 3 options indiquées page précédente, cette section utilise la méthode de l'outil clone pour associer les paramètres avec le Scenario Manager.

Les réponses (résultats ayant un intérêt) se trouvent dans la table suivante :

Nom du bloc (Label)	Onglet	Paramètre	Nom de la réponse
Queue (File d'attente)	Résultats	Longueur: Moyenne	Durée moyenne

Queue (File d'attente)	Résultats	Attente: Moyenne	Attente moyenne
------------------------	-----------	------------------	-----------------

- Dans l'onglet Results du bloc Queue, cliquez sur la variable correspondant à *Length : Average*
- Effectuez un clic-droit dans le champ et sélectionnez l'*Outil de clonage*
- Tout en gardant le curseur sur le champ, faites-le glisser sur l'icône du Scenario Manager
- Quand l'icône est sélectionné, relâchez le curseur.
- Dans le dialogue qui apparaît, sélectionnez **Reponse**
- Nommez la réponse **Longueur Moy**



Remarquez que la table Réponses dans l'onglet Responses du Scenario Manager contient maintenant toutes les informations sur la variable Longueur moyenne.

- De la même façon, utilisez soit Maj-clic ou soit l'outil clone pour ajouter la réponse nommée Attente moyenne dans le Scenario Manager

La table Réponses dans l'onglet Responses devrait maintenant ressembler à cette capture d'écran :

	Resp Name	Block Name	Block Number	Block Label	H-Block Name	H-Block Label[num]	Dialog Var	Row,Column
1	Longueur moy	Queue	10	Entry Line		[-1]	AveLength_prm	-1,-1
2	Attente Moy	Queue	10	Entry Line		[-1]	AveWait_prm	-1,-1

Portion de la table des réponses

Déterminez ce qui sera ajouté au rapport

La quatrième étape sert à déterminer quel type de rapport vous voulez et ce que vous voulez voir dans le rapport.

La table Responses est l'endroit où vous spécifiez quel niveau de statistiques vous voulez que le rapport calcule et quelles réponses vous voulez inclure dans le rapport.

- Pour lancer un Plan d'expérience JMP, définissez le type d'analyse JMP désiré dans la colonne *Mini/Maxi*: Maximize (Maximiser), Match Target (Conforme cible), Minimize (Minimiser) ou None (Aucun).
- La colonne *Report set* détermine quelle statistique est retenue pour la réponse spécifiée. Par défaut ne sont montrées que la valeur moyenne de la réponse pour chaque scénario, mais des statistiques plus détaillées peuvent être ajoutées en sélectionnant parmi celles proposées.

Mean (M)
Mean, Standard Deviation (M,SD)
Mean, Confidence Interval (M,CI)
Mean, Variance, Standard Deviation, Confidence Interval (M,V,SD,CI)
Maximum, Minimum (Max,Min)
Mean, Maximum, Minimum (M,Max,Min)
All Results

- La colonne *Include in report* sert à choisir quelles réponses seront incluses dans le rapport.

Pour cet exemple :

- Garder les valeurs par défaut

Sélectionner le Plan d'expérience et simuler des scénarios

L'onglet Scénarios du bloc GS sert à choisir une méthode pour le Plan d'expérience (design of experiment ou DOE) ainsi qu'à créer et simuler des scénarios. Comme expliqué [plus loin](#), il y a quatre principales méthodes de PE que vous pouvez choisir :

- Création manuelle
- Plan factoriel complet
- Minitab optimal
- Création personnalisée JMP, qui a deux options.

Ces deux options vous permettent soit de créer puis simuler des scénarios (et plus tard exporter le rapport dans JMP), soit de créer et simuler immédiatement les scénarios, avec l'exportation automatique du rapport dans JMP et sa mémorisation dans ExtendSim. Ces deux options requièrent l'outil JMP de la société SAS Corporation pour générer la création et analyser les résultats.

- Dans la section *Run control*, sélectionnez Choose DOE Method : Full factorial design (Plan factoriel complet)
 - Cliquez sur le bouton Create scenarios
- Ceci génère toutes les combinaisons des valeurs des facteurs. Il y aura un total de 45 lignes (3 x 3 x 5 niveaux) dans la table des Scénarios. Par défaut tous les scénarios sont sélectionnés ; pour ne pas simuler un scénario particulier, décochez sa case *Select*.
- Puisque vous voulez simuler tous les scénarios, laissez toutes les cases **Select** cochées
 - Entrez dans **Runs** (réplications) **per scenario : 5**
 - Suggestion : Utilisez le bouton Animation ou décochez la commande Simulation > Montrer animation 2D pour désactiver l'animation 2D
 - Cliquez sur le bouton **Run Scenarios**

Ce processus peut prendre du temps puisque 225 réplications sont requises pour terminer l'analyse du scénario !

☞ Les scénarios seront simulés **BEAUCOUP** plus rapidement si l'animation n'est pas activée.

Décochez *Son à la fin de la simulation* dans l'onglet Edition > Options > Modèle, avant la simulation.

Quand la simulation des scénarios est terminée, vous devriez voir ceci dans la table des Scénarios :

Scenarios								
	Select	Scenario Name	Voitures	Lavage	Lavage+Pol	(M) Longueur moy	(M) Attente Moy	Details
1	<input checked="" type="checkbox"/>	Scenario 01	3	5	6			Show
2	<input checked="" type="checkbox"/>	Scenario 02	3	5	7			Show
3	<input checked="" type="checkbox"/>	Scenario 03	3	5	8			Show
4	<input checked="" type="checkbox"/>	Scenario 04	3	6	6			Show
5	<input checked="" type="checkbox"/>	Scenario 05	3	6	7			Show
6	<input checked="" type="checkbox"/>	Scenario 06	3	6	8			Show
7	<input checked="" type="checkbox"/>	Scenario 07	3	7	6			Show
8	<input checked="" type="checkbox"/>	Scenario 08	3	7	7			Show
9	<input checked="" type="checkbox"/>	Scenario 09	3	7	8			Show
10	<input checked="" type="checkbox"/>	Scenario 10	3,5	5	6			Show
11	<input checked="" type="checkbox"/>	Scenario 11	3,5	5	7			Show
12	<input checked="" type="checkbox"/>	Scenario 12	3,5	5	8			Show

Portion de la table des scénarios

Résultats d'analyse

La BDD Scenario DB, dans la liste du menu BDD, mémorise automatiquement toutes les réponses de chaque scénario dans une série de tables. Les tables correspondent aux lignes de la colonne Scenario Name dans la table Scénarios. La table All Scenarios de Scenario DB contient des informations sur à peu près tous les facteurs et réponses du modèle. Vous pouvez utiliser Scenario DB pour accomplir une analyse, ou pour exporter les informations sélectionnées dans une BDD différente, dans Excel ou dans JMP.

Remarques supplémentaires :

- Décochez *Include in Report* dans l'onglet Responses pour ne pas inclure certaines réponses dans le rapport
 - Chaque réponse aura le type de statistique indiqué dans la colonne *Report Set* de l'onglet Responses. Par défaut la valeur moyenne est affichée pour chaque variable de réponse mais vous pouvez aussi afficher l'intervalle de confiance, l'écart-type et ainsi de suite.
- Changer un Paramètre du rapport après que les scénarios ont été simulés fait que le Scenario Manager recalcule les résultats avec le nouveau paramètre. C'est utile si vous changez d'avis à propos du rapport que vous désirez.
- Selon ce qui a été sélectionné dans *Report Set* de l'onglet Réponse, chaque réponse aura une colonne ou plus dans la table des Scénarios de l'onglet Scénarios. Ces colonnes résument les résultats de chaque série de réplication pour chaque scénario. Le type de rapport, par exemple IC pour Intervalle de confiance, est indiqué entre parenthèses dans l'intitulé de chaque colonne.
 - Dans l'onglet Scénarios, la colonne Détails de la table des Scénarios ouvre une table de BDD qui contient les données originelles qui ont été utilisées pour calculer toutes les statistiques. La table provient de la BDD Scenario DB qui a été créée quand le bloc Scenario Manager a été initialement placé dans le modèle.
 - Par défaut tous les scénarios sont sélectionnés dans la table des Scénarios ; pour ne pas simuler

un scénario particulier, décochez sa case Sélectionner.

Exporter les résultats (optionnel)

La forme des tables de la BDD Scenario DB peut ne pas correspondre exactement à ce que vous voulez. Pour faire des analyses supplémentaires en utilisant un format différent, exportez un rapport via l'onglet Export. Par exemple, vous pourriez exporter les résultats complets, y compris les valeurs de chaque scénario, dans JMP. Utilisez ensuite les capacités d'analyse de JMP pour comparer les différents scénarios. Pour exporter les résultats :

- Choisissez le type de rapport.
 - Complete Results (Résultats complets) exporte le contenu de la BDD Scenario DB (les facteurs et les réponses pour chaque simulation dans chaque scénario).
 - Displayed Statistics (Statistiques affichées) exporte les statistiques pour chaque scénario comme indiqué dans la table des Scénarios. Ce qui est affiché dépend de ce qui est sélectionné dans la colonne *Report Set* de l'onglet Responses.
 - Complete Statistics (Statistiques complètes) exporte toutes les statistiques disponibles pour chaque scénario. Ce sont toutes les statistiques possibles permises par la colonne *Report Set* de l'onglet Responses.
- Sélectionnez la destination pour le rapport : une BDD ExtendSim, Excel ou JMP.
- ☞ Pour ne pas écraser accidentellement les données, il vous est conseillé de ne pas exporter dans la BDD Scenario DB.
- Définissez les paramètres de destination pour les données exportées. Ceci varie selon la destination sélectionnée pour l'exportation.
- Cliquez sur Export Report

Une fois que vous avez exporté les résultats, analysez les scénarios en classant, en traçant des courbes ou en générant des statistiques de comparaison.

Didacticiel s'appuyant sur des variables des bases de données

☞ Dans cette section il est supposé que vous comprenez comment créer des BDD ExtendSim, y compris comment les lier et comment établir des relations parent/enfant.

Jusqu'ici vous n'avez fait que contrôler des variables de dialogue avec le Scenario Manager. La plupart des analystes n'utiliseront que les paramètres de dialogue comme facteurs et réponses, mais certains d'entre vous voudront faire des analyses plus poussées. Avec le Scenario Manager vous pouvez utiliser des BDD entières ou un sous-ensemble d'une BDD (tables, champs ou enregistrements), comme facteurs et/ou réponses. Ces variables de BDD peuvent être utilisées en plus ou à la place des variables de dialogue.

☞ Utiliser des variables de BDD est plus compliqué que d'utiliser des paramètres de dialogue. Cependant, c'est très utile quand vous voulez utiliser pour chaque scénario des séries de données, plutôt que de simples variables. Par exemple, plusieurs programmes de production pour un modèle de fabrication.

Pour utiliser le Scenario Manager avec une variable de BDD, vous devez créer une BDD ExtendSim ainsi que les tables, champs et enregistrements nécessaires (dans ce didacticiel il est supposé que vous savez déjà faire cela). Vous devez aussi spécifier les cibles et sources, comme indiqué ci-dessous.

Cibles et sources

Le concept de cibles et de sources est spécifique au Scenario Manager et permet des analyses plus sophistiquées.

Définitions

- Une *variable BDD* est une variable qui pointe sur une BDD entière ou un sous-ensemble d'une BDD, tels une table ou un champ.
- Une *source* est une variable BDD qui est copiée dans une cible
 - Les *sources des facteurs* (factor sources) sont l'ensemble des valeurs qu'une variable BDD peut prendre pour tous les différents scénarios. Elles contiennent toutes les informations de tous les scénarios.

- Une *source de réponse* (response source) est la valeur d'une variable BDD en sortie pour une réplication particulière. Chaque source de réponse n'a que les informations pour la série de réplifications en cours.
- Une *cible* (target dans l'interface du bloc) est une variable BDD dont la valeur est définie par une source
 - Une *cible de facteur* (factor target) est une variable BDD en entrée dont la valeur a été définie par une source de facteur particulière. Chaque cible de facteur n'a que les informations pour la série de réplifications du scénario en cours.
 - Les *cibles de réponses* (response targets) sont une collection de variables BDD en sortie où chaque valeur de variable est définie par une source de réponse particulière. Elles possèdent tous les résultats pour tous les scénarios.

Vue d'ensemble

- Les architectures des cibles et des sources doivent correspondre. Par exemple, si la source est une table de BDD, la cible doit aussi être une table de BDD avec le même nombre de champs.
 - Un nombre minimum de sources et de cibles doivent être créées pour le modèle :
 - Pour les *facteurs* BDD, au moins deux sources et au moins une cible. (Chaque ensemble de données est une source donc avoir une seule source n'a aucun sens dans le cas de séries de données).
 - Pour les *réponses* BDD, une source et un certain nombre de cibles. Le nombre de cibles dépend si ExtendSim crée des sources et des cibles ou si vous le faites manuellement. Si *Auto-Create Target* est sélectionné dans la table Responses issues de la BDD, ExtendSim va créer une nouvelle BDD cible pour chaque simulation lancée. Si vous le faites manuellement, créez une source et une cible par scénario.
 - Si vous simulez le modèle dans le but de le tester sans utiliser le Scenario Manager, les variables identifiées peuvent être utilisées dans le modèle. Cependant, ces variables sont juste des paramètres fictifs et leur contenu changera quand vous utiliserez le Scenario Manager.
 - *Cible de facteur*. Quand vous testez le modèle, utilisez la *cible* de facteur BDD dans vos blocs. (Liez un paramètre ou une table de données à la variable cible ou utilisez un bloc Read(I) ou Write(I) pour pointer sur la variable cible).
 - *Source de réponse*. De la même façon, construisez le modèle comme si vous alliez utiliser un paramètre fictif de réponse BDD. Cependant, utilisez la *source* plutôt que la cible.
 - Les sources sont toujours copiées dans les cibles. Quand et comment cela arrive dépend de si la variable BDD est un facteur ou une réponse.
 - *Facteur*. Au début de chaque série des réplifications du scénario, le Scenario Manager copie le contenu de la variable de facteur BDD source dans la variable de facteur BDD cible. Ceci est réalisé automatiquement par le Scenario Manager.
 - *Réponse*. À la fin de chaque simulation les variables BDD utilisées dans le modèle sont copiées des variables BDD source aux variables BDD cible pour ce scénario.
- ☞ Le contenu de la table, plutôt que la table elle-même, est copié. Ainsi l'index de la variable de la BDD cible ne change pas.

Diagramme des sources et des cibles

Le diagramme suivant montre comment les sources et les cibles sont en relation avec les facteurs et les réponses.



→ Les facteurs BDD sont copiés des sources à la cible au début de chaque réplication. Les réponses BDD sont copiées de la source aux cibles à la fin de chaque simulation.

Étapes à suivre pour une analyse de scénario

Les étapes pour effectuer une analyse de scénario en utilisant des variables de BDD sont les suivantes :

- 1) Ouvrez un modèle
- 2) Créez la BDD, les tables, les champs et les enregistrements requis pour le modèle
- 3) *Optionnel*. Si vous pensez utiliser les plans d'expérience, créez une table supplémentaire pour chaque variable de BDD, où chaque table a une liste des valeurs possibles pour cette variable
- 4) Créez des liens entre le modèle et les variables de BDD
- 5) Entrez les informations dans la table des Facteurs BDD (Database Factors)
- 6) Entrez les informations dans la table des Réponses BDD (Database Responses - cette étape n'est pas présente dans cet exemple)
- 7) Générez et simulez les scénarios
- 8) Optionnel. Exportez le rapport

☞ Les étapes ci-dessus supposent que les tables de BDD sont utilisées pour les facteurs et les réponses. Vous pouvez aussi utiliser les champs ou les enregistrements, ou même des BDD entières, en tant que cibles et/ou réponses.

Ouvrir le modèle

Dans le didacticiel précédent vous avez construit un modèle en utilisant des variables de dialogue comme facteurs et réponses. Au lieu de vous faire commencer avec le modèle précédent, ce didacticiel utilise un modèle similaire qui possède déjà la structure de BDD requise.

Ouvrez le modèle *Scenario Manager Final Car Wash* (ExtendSim/Examples/How To/Scenario Manager).

Similairement au modèle que vous avez construit lors du didacticiel précédent, le bloc Scenario Manager a dans ce modèle :

- Trois facteurs dans la table des Facteurs de dialogue : *Wash Time*, *Wash&Wax Time*, et *Cars Ratee*
- Deux réponses dans la table Réponses du dialogue : *Ave Length 1* et *Ave Length 2*

Ce didacticiel va garder ces facteurs de dialogue et ces réponses mais va aussi y ajouter un quatrième facteur (le pourcentage de voitures qui veulent *Essence* ou *Essence+Lavage*) dans la table Database Factors (Facteurs de BDD).

Dans le modèle, la proportion de voitures voulant soit le lavage seulement ou soit le lavage et le polissage est définie dans la table empirique du bloc Random Number. La table est montrée ci-contre.

	Preference	Probability
0	Wash only	0.75
1	Wash and Wax	0.25

Pour ce didacticiel le facteur Wash ou Wash & Wax aura trois combinaisons de pourcentages, ce qui représente 3 niveaux :

- 1) 75% Wash et 25% Wash & Wax – le cas de base
- 2) 90% Wash et 10% Wash & Wax – le cas avec le plus de lavage
- 3) 50% Wash et 50% Wash & Wax – le cas avec le moins de lavage

Comme dans le didacticiel précédent, chaque scénario sera simulé 5 fois pour prendre en compte la variation dans les résultats d'une simulation à une autre. Evaluer toutes les combinaisons possibles demande maintenant un nombre de 675 réplifications (135 scénarios ((3 x 3 x 5 x 3 niveaux)) avec 5 simulations chacun).

Créer une structure de BDD pour le modèle

Afin que vous puissiez vous concentrer sur le didacticiel, la BDD requise a déjà été créée pour ce modèle, comme indiqué en bas du menu Base de données.

☞ La BDD système *Scenario DB* a été automatiquement créée quand le bloc Scenario Manager a été ajouté au modèle. Elle n'est pas visible par défaut.

- La BDD Model Data mémorise dans 6 tables les différents options et probabilités pour le facteur de type de service :

- Une table Parent, Wash Type, qui liste les options pour les deux types de service : Wash seulement ou Wash & Wax. Son contenu est montré ci-contre.

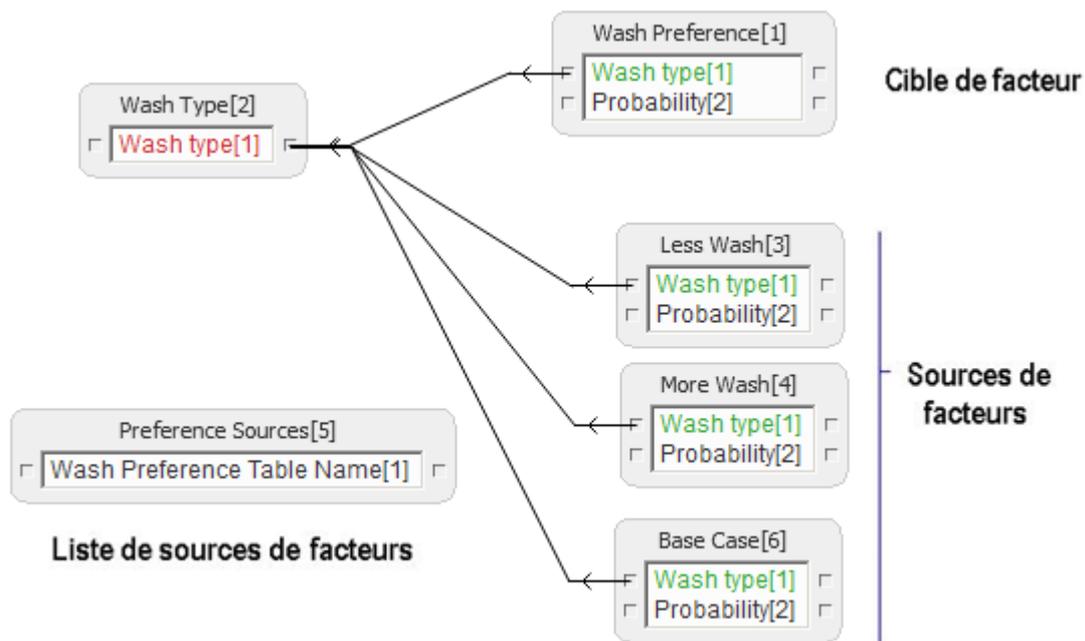
	Wash type[1]
1	Wash only
2	Wash and wax

- Trois tables sources de facteurs, chacune ayant une des combinaisons de probabilité : Cas de base (75% Wash /25% Wash & Wax), Cas – de lavage (50%/50%) et Cas + de lavage (90%/10%). Celles-ci sont des tables enfant qui prennent leur information concernant les deux services possibles dans la table Wash Type.
- Une table cible de facteurs nommée Wash Preference. Chaque enregistrement dans cette table a un champ de Wash Type et un champ de probabilité qui lui est associé. Cette table est utilisée par le modèle est cours d'exécution.
- Une table avec la liste des sources de facteurs, nommée Preferences Sources, qui est une liste des trois tables sources de facteurs. Le contenu de cette table est montré ci-contre. Cette table est utilisée lors de l'exécution d'un plan d'expérience, afin que vous sachiez quelles tables sélectionner lors de la création du plan. Voir « Créer une liste source (optionnel) » ci-dessous pour plus d'informations.

	Wash Preference Table Name[1]
1	Base Case ▼
2	More Wash ▼
3	Less Wash ▼

☞ La BDD comporte deux tables qui n'étaient pas strictement requises pour l'exemple, mais qui devraient être généralement utilisées : la table Parent (Wash Type) et la liste des tables sources (Preferences Sources).

La structure de la BDD Model Data est montrée ci-dessous.



Créer une liste source (optionnel)

Si vous prévoyez d'effectuer un plan d'expérience (DOE), créez une table supplémentaire pour chaque variable facteur de BDD, où chaque table mentionne une liste des valeurs possibles pour cette variable. C'est une table des tables, ou une liste des facteurs sources.

Utiliser une liste de facteurs sources vous permet d'utiliser un sous-ensemble de variables BDD plutôt que toutes les variables BDD du modèle. Ceci rend plus facile la création manuelle de scénarios et l'utilisation du Plan d'expérience. Sans cette liste de facteurs sources, n'importe quelle table dans la BDD serait listée dans le tableau *Database Factors* comme table possible de sources de facteurs.

Comme évoqué plus tôt, une liste de sources de facteurs, nommée Preferences Sources, a déjà été créée pour cet exemple. Comme vous le verrez plus tard, cocher *Use Source List* dans le tableau *Database Factors* permet de limiter les entrées possibles à celles de la liste source (dans ce cas, à la table Preferences Sources).

Lien vers la table cible

Pour contrôler les probabilités de préférence du modèle de lavage, la table de probabilité empirique du bloc Random Number doit être liée à la table cible (Wash Preference) :

- Cliquez sur le bouton **Link** dans la table des Probabilités du bloc Random Number

- Dans le dialogue du menu déroulant **Lier**, sélectionnez **Lier avec : Table de BDD**
- Sélectionnez la BDD appelée **Model Data**
- Choisissez la table nommée **Wash Preferences**, puis cliquez sur **Lier**
- Fermez le dialogue Random Number

Entrer les facteurs de BDD

Vous ne pouvez pas utiliser Maj-clic ou l'outil clone sur une variable de BDD. Cependant, vous pouvez entrer les informations manuellement. Le tableau *Database Factors* possède aussi des menus déroulants pour sélectionner les informations appropriées.

- Cochez la case *Enter database Factors* pour activer le tableau correspondant
- Utilisez le bouton de redimensionnement +/- pour ajouter une ligne à la table
- Entrez les informations suivantes dans la table:

Titre de la colonne	Entrez ou sélectionnez ces informations
Target Name	Wash ou Wash & Wax
Target Type	Table
Target DB	Model Data
Target Table	Wash Preferences
Target Field	(laissez cette cellule vide)
Target Record	(laissez cette cellule vide)
Use source list	X (cochez la case)
Source BD	Model Data
Source Table	Preference Sources
Source Field	Wash Preference Table Name

Générer et simuler des scénarios

Générez et simulez les scénarios comme vous l'avez fait [plus haut](#). Si vous utilisez la méthode du plan factoriel complet et avez 5 réplifications par scénario, 135 scénarios devraient être créés avec 575 réplifications.

Méthodes de Plan d'expérience

L'onglet Scenarios du bloc Scenario Manager vous permet de choisir la méthode de Plan d'expérience.

Méthode de Plan d'expérience	Description	Séquence
Manual	Entrez manuellement les facteurs pour chaque scénario	Créer, Simuler, Exporter
Full factorial (Plan factoriel complet)	<i>Create Scenarios</i> crée un scénario pour chaque combinaison de valeurs des facteurs. Puisque toutes les combinaisons de facteurs possibles sont générées, cette méthode est la plus longue.	Créer, Simuler, Exporter
JMP Custom	<i>Create Scenarios</i> ouvre JMP et lui fait créer un plan Custom ou D-optimal, puis affiche les facteurs et les réponses à la fois dans ExtendSim et JMP.	Créer, Simuler, Exporter
JMP custom - - create and run	<i>Create and Run Scenarios</i> ouvre JMP, lui faisant créer le plan, simuler immédiatement les scénarios, et afficher les facteurs et les réponses dans ExtendSim et JMP.	Créer et Simuler
Minitab optimal	<i>Create Scenarios</i> ouvre Minitab et lui fait créer un plan Optimal ou D-optimal, puis affiche les facteurs et les réponses à la fois dans ExtendSim et Minitab.	Créer, Simuler, Exporter

Minitab optimal – create and run	<i>Create and Run Scenarios</i> ouvre Minitab, lui faisant créer le plan, simuler immédiatement les scénarios, et afficher les facteurs et les réponses dans ExtendSim et Minitab.	Créer et Simuler
----------------------------------	--	------------------

Les options pour JMP et Minitab requièrent d'avoir les applications JMP de SAS Corporation ou Minitab de Minitab Inc. pour générer le plan et affiche les résultats (version compatible 64-bits).

☞ Les méthodes JMP custom et Minitab optimal produisent bien moins de scénarios que le plan factoriel complet.

Comparaison des méthodes d'analyse

Le Scenario Manager est juste un des outils d'analyse d'ExtendSim. Chaque outil a un but différent, comme le décrit la table suivante.

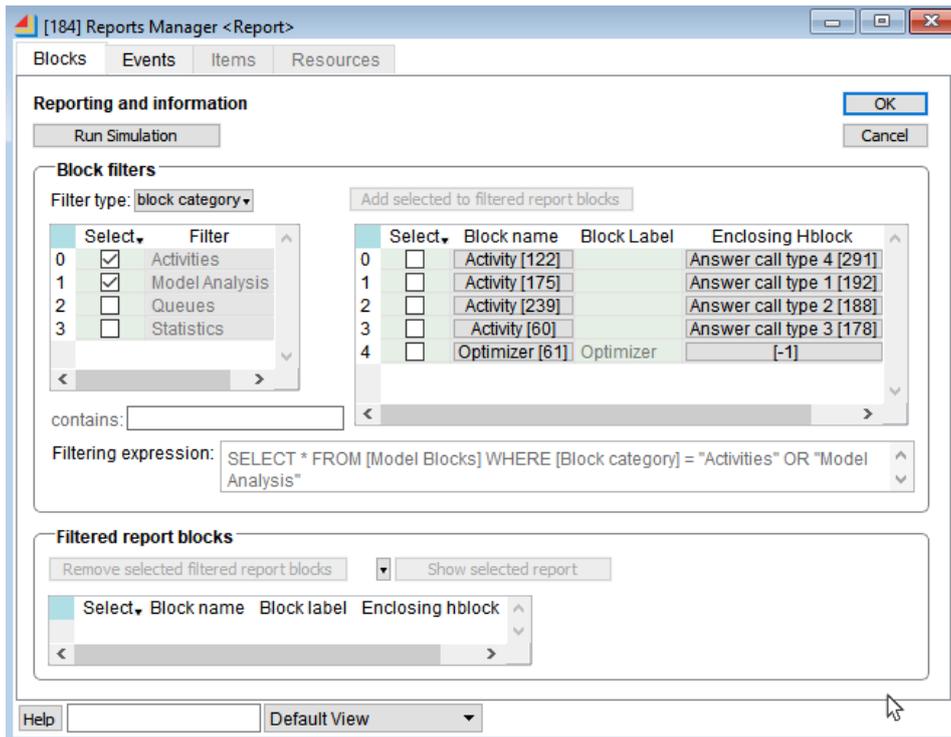
Technique	Analyse de sensibilité	Scenario Manager	Optimisation
Objectif	Etudier le comportement du modèle en réponse à des changements dans un seul paramètre d'entrée	Etudier et comparer le comportement du modèle en réponse à des changements dans plusieurs paramètres d'entrée. Apprendre comment le système se comporte dans différentes conditions.	Déterminer la meilleure configuration des paramètres du modèle, selon un but spécifique (habituellement minimiser les coûts ou maximiser le profit)
Question	Si un paramètre avait cette valeur, comment cela affecterait-il les résultats du modèle ?	Si plusieurs paramètres avaient ces valeurs, comment cela affecterait-il les résultats du modèle ?	Quelle configuration permet à un modèle de se comporter de façon optimale ?
Avantages	Le plus facile à spécifier	Le plus complet pour comprendre comment le modèle réagit à différents facteurs. Les facteurs peuvent provenir de BDD comme de variables de dialogue. Enregistre automatiquement chaque réponse pour chaque réplication Les Plans d'expérience peuvent réduire le nombre de scénarios requis pour étudier les interactions de facteurs	Détermine la configuration optimale du modèle Des contraintes peuvent être ajoutées pour éliminer les configurations de modèle impossibles
Désavantages	Evaluer plus d'un facteur est lourd De la modélisation supplémentaire est requise pour enregistrer les résultats du modèle.	Une série de scénarios doit être créée par le modélisateur Peut obliger à un grand nombre de réplifications pour évaluer tous les scénarios.	Une fonction d'objectif est requise N'enregistre pas les résultats de chaque simulation Il y a une limite pratique sur le nombre de facteurs et leurs valeurs possibles pouvant être examinées Peut obliger à un grand nombre de réplifications pour déterminer la configuration optimale.

Rapport d'un modèle

Le bloc Report Manager (bib. Report) sert d'interface pour créer un rapport personnalisé durant la simulation, concernant des blocs, événements, entités ou ressources. Le dialogue propose plusieurs filtres possibles pour sélectionner les éléments pertinents.

- Le rapport sur les blocs rassemble des informations issues des onglets *Results* des blocs présentés par catégories.
- Le rapport sur les événements rassemble des informations sur des événements sélectionnés par des conditions de filtrage.

En fin de simulation les informations réunies sont stockées dans les tables d'une BDD ExtendSim. Ces tables sont accessibles depuis le dialogue du Reports Manager et depuis l'interface de la base de données.



Filtres sur les blocs

Dans l'onglet Blocks les filtres de sélection des blocs sont les suivants :

- Catégories définie à la création du bloc, telle Inputs ou Math
- Nom du bloc, tel Activity ou Lookup Table
- Label du bloc
- Bloc hiérarchique parent.

Plusieurs filtres peuvent être utilisés ensemble, par exemple catégorie et label.

Event names

Dans l'onglet Events vous sélectionnez une ou plusieurs catégories d'événements:

- Creation (creation)
- Arrival (arrivée)
- Downstream blocking (blocage aval)
- Resources unavailable (ressources disponibles)
- Departure (départs)
- Exit (sortie du modèle)
- Route (orientation)
- Batch (lot)
- Unbatch (délotage)

- Spawn (memo)
- Preempt (preemption)
- Shutdown (arrêt)
- Item allocation (allocation d'entité)

Les [autres blocs](#) de la bibliothèque Report servent également à documenter un modèle ou rassembler des résultats statistiques.

Les graphiques

L'ancienne bibliothèque Plotter a pris le statut de Legacy, ce qui signifie qu'elle reste utilisable par souci de compatibilité avec les anciens modèles. Mais pour tout nouveau modèle, nous recommandons d'utiliser uniquement les blocs de la bibliothèque Chart.

Les graphiques de la bibliothèque Chart d'ExtendSim permettent d'avoir une représentation graphique des résultats, notamment l'évolution de valeurs, et ont souvent des points communs. Les modèles des tutoriaux ont montré comment utiliser un graphique Line Chart.

Les graphiques permettent de tracer jusqu'à 20 signaux simultanément. Si vous construisez votre propre traceur avec le langage ModL, vous pouvez tracer jusqu'à 100 signaux.

Vous pouvez avoir plusieurs graphiques dans un même modèle, et les placer où vous le voulez. Les graphiques s'adaptent au type de modèle (continu, discret, DED) dans lesquels vous les placez : dans certains cas des options seront inaccessibles dans les modèles continus.

Types de graphiques

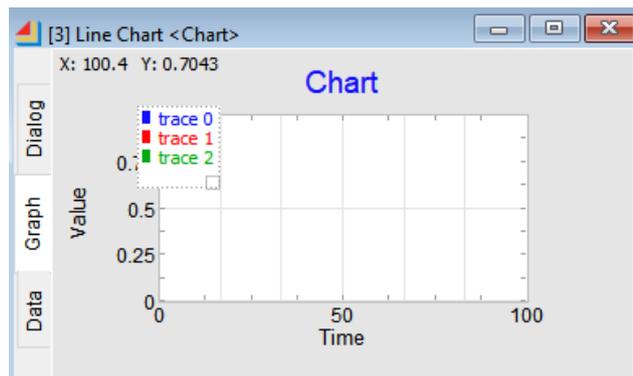
ExtendSim propose quatre types de graphiques dans la bibliothèque Chart, chacun avec une fonction différente.

Bloc	Description
Bar Chart	Affiche des comparaisons entre des catégories de données pour une ou plusieurs séries. Les données sont assemblées en groupes discrets (les catégories) comme par exemple les mois de l'année, des groupes d'âge, des types de machines, etc. Plusieurs séries peuvent s'y référer : ainsi des données de précipitation de 4 emplacements (4 séries) pour chaque mois de l'année (12 catégories). Les barres peuvent être empilées ou groupées côté à côté.
BD Line Chart	Affiche des données issues de la base de données ExtendSim sous forme de graphique tout en écrivant ces données dans une table. Peut aussi tracer l'historique de valeurs reçues dans une cellule de BDD durant la simulation, ou encore le contenu de tables. Peut s'utiliser sur plusieurs réplifications
Line Chart	Trace des valeurs dans le temps, jusqu'à 20 courbes et leur données associées. Peut s'utiliser sur plusieurs réplifications et dans un tracé de type "escargot".
Scatter Chart	Permet de tracer deux ensembles de données sous forme de paires x,y (en nuage de points). Vous devez lier au moins une des paires x,y en entrée, et jusqu'à 20. Peut s'utiliser sur plusieurs réplifications et dans un tracé de type "escargot"

Structure des graphiques

Que le bloc s'ouvre automatiquement ou par un double-clic, c'est toujours le graphique qui est d'abord visible. Quasiment tous les graphiques ont certains points communs :

- trois onglets sur la gauche ;
- la légende du graphique et les valeurs du curseur, visibles par défaut ;
- les dialogues Editeur de courbes et Propriétés du graphique qui sont appelés par clic du bouton droit, ou double-cli sur la légende ou les valeurs du curseur.



Onglet	Description
Dialog	C'est l'onglet de paramétrage du graphique. Il se décompose avec trois autres onglets : <i>Data Collection</i> qui détermine quelles données sont tracées, et quand. <i>Display</i> qui définit le moment de l'affichage du graphique et son échelle. <i>Comments</i> pour les habituels commentaires.
Graph	Affichage du graphique lui-même, avec le contrôle de son apparence via les dialogues complémentaires Editeur de courbes et Propriétés du graphique
Data	Données à la source du graphique, sous forme d'un tableau

Onglet Display

L'onglet Display est l'un des trois onglets qui apparaissent en haut de la fenêtre lorsque l'onglet Dialog est actif.

Section General display

Cette section détermine si et quand le graphique doit s'ouvrir :

- Never = jamais, par défaut
- At the beginning of the simulation = en début de simulation
- At the end of the simulation = en fin de simulation

Selon le bloc d'autres options figurent dans cette section, comme on le voit pour le bloc Line Chart.

Option Background picture

Par exemple, pour insérer une image de fond:

- 1) Ajoutez le caractère @ au début du nom du fichier
- 2) Placez le fichier dans le répertoire ExtendSim/Extensions/Pictures
- 3) Relancez ExtendSim1.
- 4) Dans la section General display de l'onglet Display, cochez la case *Insert background picture*
- 5) Sélectionnez le fichier via le menu

Option d'empilement (Bar Chart)

Si plusieurs Séries sont spécifiées via l'onglet Data Collection, le graphique Bar Chart devient des barres groupées. Dans ce cas, les barres de chaque série peuvent être affichées soit côte à côte (par défaut), soit empilées l'une sur l'autre. Voyez [plus loin](#) la configuration du graphique Bar Chart.

Section Scaling

Cette section détermine si et quand le graphique sera mis à l'échelle:

- Never = jamais
- At the beginning of the simulation = en début de simulation
- At the end of the simulation = en fin de simulation, par défaut

D'autres options indiquent ce qui se passe durant la mise à l'échelle et comment définir les bornes des axes.

Onglet Data Collection

L'onglet Data Collection est l'un des trois onglets qui apparaissent en haut de la fenêtre lorsque l'onglet Dialog est actif. Il détermine quels points sont utilisés et stockés.

☞ Les modèles continus affichent moins d'options que celles présentées.

Data collection options									
	Trace Na...	Con	Filtering	Start Time	End Time	Blank	Offset By	Cross Msg	Run
0	trace 0	<input type="checkbox"/>	no duplicate	do nothing	write point	ignore	0	auto	current
1	trace 1	<input type="checkbox"/>	no duplicate	do nothing	write point	ignore	0	auto	current
2	trace 2	<input type="checkbox"/>	no duplicate	do nothing	write point	ignore	0	auto	current

Le cadre comporte un tableau avec plusieurs colonnes. Chaque ligne représente un tracé sur le graphique, et les colonnes sont des propriétés personnalisables. La plupart ne s'appliquent qu'aux modèles discrets et DED.

- *Trace Name*. Vous pouvez donner un nom à chaque tracé, dans ce tableau ou dans l'[Editeur de courbes](#).
- *Series* (Bar Chart). Vous pouvez donner un nom à chaque tracé, dans ce tableau ou dans l'éditeur des séries qui apparaît par le menu du bouton droit sur le graphique.
- *Con* (Line Chart and Bar Chart) ainsi que *X* et *Y* (Scatter Chart). Ces case à cocher indiquent s'il y a un lien relié au connecteur d'entrée correspondant.
- *Filtering*. Ces options de filtrage permettent d'ajuster quels points seront tracés. Une colonne supplémentaire 'Duration' apparaît si l'option de filtrage 'Scheduled' est choisie.
- *Start Time* and *End Time*. Contrôle si un point est trace pour le temps de début et le temps de fin de la simulation. S'emploie dans les modèles discrets, où il est plus exact parfois que faire s'arrêter le tracé au moment du dernier événement plutôt qu'en End Time.
- *Blank*. Si une valeur Blank (vide) est reçue, le bloc peut l'ignorer ou interrompre le trace.
- *Offset By*. Décale l'affichage des valeurs sur l'axe Y ou Y2 (Line Chart), ou bien X, Y, ou Y2 (Scatter Chart) de sorte que les traces ne se superposent pas.
- *Cross Msg* (Line Chart et Bar Chart). Auto est l'option par défaut. Si vous ne voyez pas des points que vous considérez devoir figurer sur le graphique, essayez une des autres options. Elles changent la manière dont sont interprétés les messages au cas où le bloc relié au graphique n'avertirait pas d'un changement de valeur. Un exemple serait le cas d'une entité qui entre et sort d'une file au même événement (un événement de durée zéro). Cette option n'est activée que si le filtrage n'est pas 'scheduled', pour les modèles discrets et DED.
- *Auto* signifie que le connecteur est dépendant par défaut mais devient indépendant s'il reçoit un message.
- *Dependent* fait que le point est écrit si la valeur sur le connecteur d'entrée correspondant a changé et que le connecteur n'a pas reçu de message mais qu'un autre connecteur lui a reçu un message.
- *Independent* signifie qu'un point n'est écrit que si le connecteur d'entrée a reçu un message direct.
- *X Msg* (Scatter Chart). Ce type de graphique requiert des valeurs de l'axe X et Y. Lorsque l'entrée pour X reçoit un message, le bloc a trois options pour savoir comment obtenir une valeur pour l'axe opposé (Y):
 - *Pull from the opposite connector*, qui fait que l'entrée Y va chercher activement une valeur à fournir au bloc.
 - *Read from the opposite connector*, qui fait que l'entrée Y va passivement indiquer au bloc la valeur disponible.
 - *Ignore message*, qui fait que ce message sera ignoré.
- *Y Msg* (Scatter Chart). Comme pour l'option *X Msg* ci-dessus, en inversant les connecteurs.
- *Run* (Line Chart et Scatter Chart). Utilisez cette option avec des simulations multiples. Par défaut l'option "Current" est sélectionnée et chaque simulation écrasera le trace de la simulation antérieure. Si "only run#" est sélectionné et que différentes réplifications sont indiquées par leur numéro dans la colonne Run# qui s'affiche, chaque réplification écrira un tracé différent, tous les connecteurs étant reliés au même signal. Voyez [une application](#) avec l'analyse de sensibilité.

Data recording options

Data recording options

Disable recording (memory saving)

Data collection window: Start time: End time: time units

Only record the latest points on the chart (worm).

Pour les graphiques Line Chart, Scatter Chart et DB Line Chart, le cadre du bas de l'onglet Data Collection contrôle le recueil et l'enregistrement des données, et permet :

- D'inhiber entièrement le recueil des données (fait gagner du temps et de la mémoire)
- De ne recueillir les données que Durant une fenêtre de temps spécifique
- De ne recueillir que les N derniers points spécifiés ("graphique escargot")

Configurer les options du Bar Chart

En haut de l'onglet Data Collection du Bar Chart figure un cadre pour définir les catégories et séries.

- Les données sont regroupées en un ou plusieurs groupes discrets (catégories) comme par exemple les mois de l'année, des groupes d'âge, des types de machines, etc. Plusieurs séries peuvent s'y référer : ainsi des données de précipitation de 4 emplacements (4 séries) pour chaque mois de l'année (12 catégories). Les catégories que l'on compare sont listées le long de l'axe horizontal; et la hauteur d'une barre est proportionnelle aux valeurs que les barres représentent. Par défaut il y a une série (et donc une barre) pour chaque catégorie.
- Spécifier deux séries ou plus permet d'utiliser le Bar Chart avec des barres groupées. Dans ce cas chaque groupe catégoriel comporte deux barres ou davantage, chacune représentant une série. Chaque série peut avoir sa propre couleur et est mentionnée en légende. Par exemple, des données de précipitation de 4 emplacements (4 séries) pour chaque mois de l'année (12 catégories).

Par défaut les barres sont côte à côte, une option de l'onglet Display permet de les empiler.

☞ Le graphique Bar Chart peut comporter jusqu'à 250 barres, 20 Catégories, et 20 Séries. Cependant ces contraintes sont limitées par le fait que le nombre de barres est égal au nombre de Catégories multiplié par le nombre de séries. Ainsi si vous avez 20 Catégories, vous êtes limités à 12 Séries.

Configurer les options du DB Line Chart

Le graphique DB Line Chart trace et affiche des informations issues de la base de données ExtendSim. Au lieu de lire des valeurs depuis des connecteurs d'entrée, le block trouve des données dans la BDD. Son onglet Data Collection est très différent de celui décrit ci-dessus.

☞ Pour utiliser ce bloc, le modèle doit comporter une base de données ExtendSim.

Le cadre *Data collection type* en haut de l'onglet Data Collection détermine ce qui doit être montré dans le graphique.

Ce que représente le tracé

- *One cell per trace*. Une cellule par tracé : c'est l'option la plus simple pour voir l'évolution des valeurs dans une ou plusieurs cellules de la BDD. Les valeurs de chaque cellule sélectionnée sont affichées dans un tracé et remplissent une colonne dans l'onglet Data. Les cellules source peuvent provenir d'une ou de plusieurs BDD. Chaque point du tracé représente la valeur de la cellule à cet instant de la simulation.
- *One field per trace*. Un champ par tracé : trace les données de différents champs d'une même table de BDD. Les informations temporelles sont celles d'un champ d'une et une seule table, et les champs à tracer doivent provenir de la même table. Ainsi chaque enregistrement est un point d'un tracé et une valeur dans la table de données du bloc.
- *One table per trace*. Une table par tracé : trace les données de différents champs de différentes tables de BDD. Chaque table doit avoir son propre champ temporel et au moins un champ de données à tracer. Si tous les champs proviennent de la même table de BDD, cela revient à l'option précédente.

Quand est lue la source de données

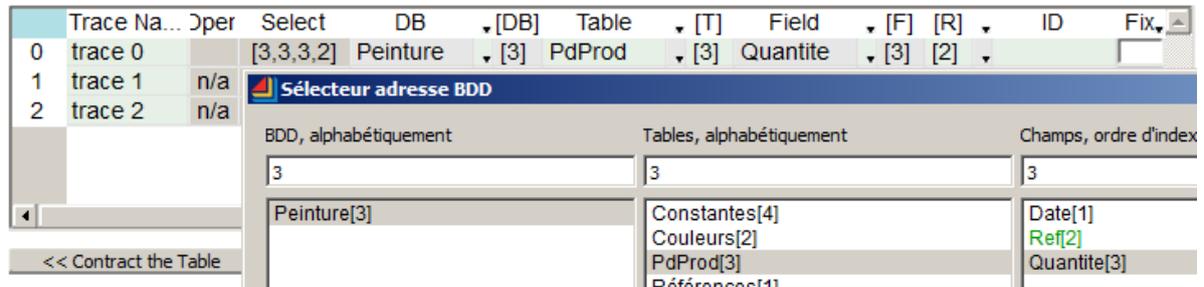
Lorsque l'option "One cell per trace" est choisie, les données sont toujours recueillies durant la simulation.

Avec les deux autres modes, les options sont :

- During the simulation. Note : cette option ralentit notablement la simulation.
- After the simulation. Le graphique est calculé en fin de simulation
- Only when the Plot Now button is clicked. Utilisez cette option si vous voulez juste visualiser ce que contient la BDD en cet instant – en évitant de lancer la simulation, ou pour voir des résultats intermédiaires en cours de simulation.

Data collection options pour DB Line Chart

Le cadre *Data collection options per trace* au milieu de l'onglet Data Collection du bloc DB Line Chart sert à choisir la source des données à tracer, une ligne par tracé. Les options sont assez explicites, nous expliquons celles qui pourraient poser un problème.



Sélection de la source en mode 'Une cellule'

Database:

Table:

Time Field:

Trace Name	Field	[F]	Fix	Blank	Offset	Run
0 trace 0	Quantite	[3]		ignore		current

Sélection de la source en mode 'Un champ'

☞ En fonction des données prises en compte, les options ne seront pas toutes visibles.

- *Open*. Après avoir spécifié les emplacements des données source (base de données, table, champ et éventuellement enregistrement), cliquez sur le bouton de la colonne Open pour ouvrir la BDD en mode Edition. Le bouton n'est actif que si la sélection de la source est faite.
- *Select*. Si cette colonne est visible, permet de spécifier l'adresse de BDD plutôt que de sélectionner chaque composant intermédiaire via le Sélecteur.
- *Fix*. Si cochée, le nom du tracé reprend le nom du champ.

Onglet Graphique

L'onglet Graphique est l'un des trois onglets figurant à gauche d'un bloc graphique, et montre le graphique lui-même, tel qu'il a été configuré comme expliqué plus haut.

Vous pouvez directement nommer le graphique, changer les limites des axes, et changer le texte des labels. Pour changer d'autres propriétés, ouvrez le dialogue Propriétés du graphique par clic du bouton droit ou double-clic sur la case des valeurs du curseur.

Par défaut les cases avec *Légende* et *Valeurs du curseur* figurent sur le graphique.

Légende

Chaque graphique a par défaut une case pour la Légende du graphique, qui indique le nom et la couleur de chaque tracé.

L'apparence de la légende (cadre, couleur, affichage) est contrôlé dans le dialogue [Propriétés du graphique](#).

Pour redimensionner la légende, sélectionnez-la et utilisez le carré en bas à droite. Pour la déplacer, il suffit de la sélectionner et de la déplacer.

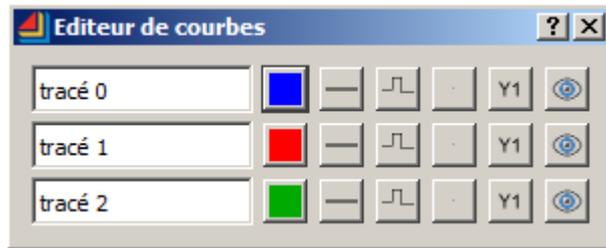
☞ Double-cliquer sur la Légende ouvre l'éditeur de courbes.

Valeurs du curseur

Chaque graphique a par défaut une case pour les valeurs à l'emplacement du curseur. Vous déplacez et redimensionnez cette case comme avec la légende.

☞ Double-cliquer sur la case du curseur ouvre l'éditeur de courbes.

Editeur de courbes



Editeur de courbes

Pour chaque tracé 7 colonnes permettent de modifier 7 aspects du tracé des graphiques.

Le texte sur la gauche **décrit les tracés**. C'est le texte affiché en légende et en haut des colonnes de données. Pour modifier un label, cliquez sur la zone et tapez votre texte. Pour activer d'autres éléments, chaque fois que vous cliquez sur la case, l'option suivante s'affiche.

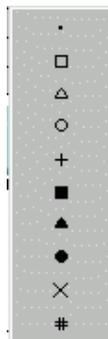
Les zones **couleur**, **épaisseur** et **motif** décrivent l'aspect du tracé. Les couleurs sont celles de la palette Windows, 5 épaisseurs de lignes, 3 styles et 4 motifs.

Les zones style et symbole affectent la façon de tracer. Voici les choix de **style** :



styles de tracé

Le style du haut indique que les points de données sont reliés par des lignes diagonales (interpolées), le style du milieu indique que les points de données sont reliés par des lignes à angle droit (échelonnées), et le style du bas ne trace rien entre les points.



Symboles de tracé

Vous avez plusieurs choix de **symboles** : points, carrés, croix, ainsi qu'un choix numérique (#). Si vous choisissez un symbole non numérique, il sera tracé à chaque point de la courbe. Si vous choisissez le symbole #, les nombres seront espacés de manière régulière pour cette courbe. Chaque symbole est tracé avec la couleur et l'épaisseur spécifiées. Les nombres sont tracés en noir.

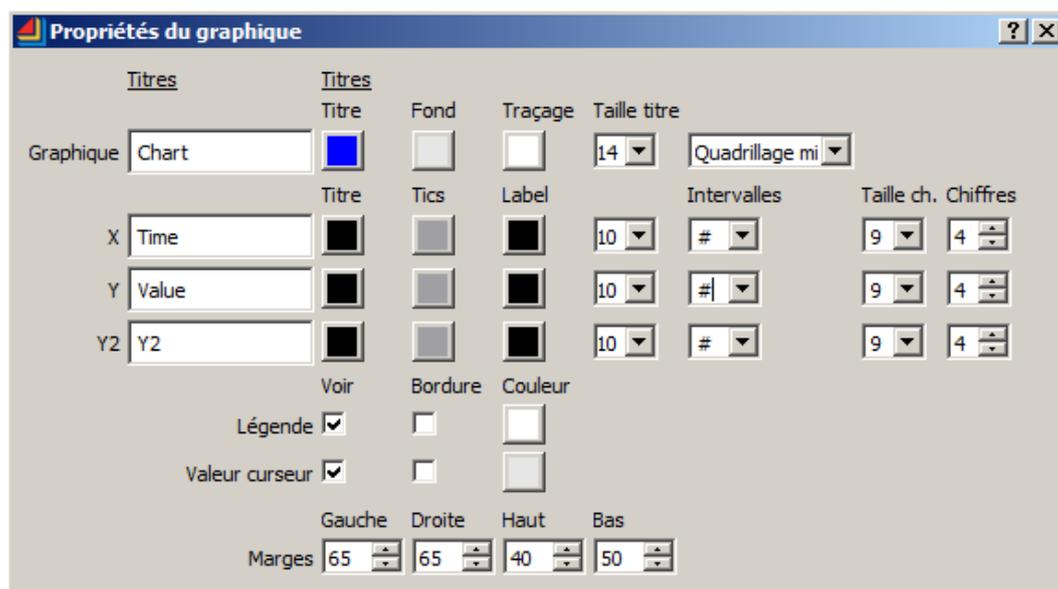
Les zones **Y1/Y2** spécifient sur quel axe tracer la courbe. Le choix par défaut utilise l'axe Y1, à gauche. En cliquant dessus vous passez à l'axe Y2, à droite.

Le choix **Voir** (l'oeil) détermine si le tracé est visible ou non. Si vous cliquez sur ce choix, il devient un oeil fermé et la courbe n'est pas dessinée. Cette option est utile lorsque vous avez beaucoup de tracés qui se superposent, car en masquant une courbe vous pouvez mieux en voir une autre. Vous pouvez aussi utiliser l'option Offset By [plus haut](#).

Propriétés du graphique

Pour voir le dialogue Propriétés du graphique montré ci-dessous, utilisez le menu par clic droit, ou double-cliquez sur la légende.

Le dialogue Propriétés du graphique sert au formatage des titres, couleurs et autres propriétés du graphique. (L'éditeur de courbes sert à formater les courbes du graphique lui-même.)



Les possibilités de personnalisation sont nombreuses. Les options sont assez explicites, nous expliquons celles qui pourraient poser un problème.

Dans la ligne Graphique, la couleur de *fond* est celle du dialogue, celle de *traçage* est de fond de la portion de tracé (celle du *quadrillage*).

Pour les lignes X, Y et Y2, les *tics* sont les petites marques tout au long des axes.

Intervalles indique combien d'espaces entre le premier et le dernier chiffre sont utilisés pour séparer les chiffres sur l'axe (l'intervalle est l'espace entre deux chiffres). La valeur par défaut (#) signifie que l'intervalle varie en fonction de la taille de la fenêtre. Toute autre option fixe le nombre d'intervalles quelle que soit la taille de la fenêtre.

Chiffres détermine combien de positions de chiffres (y compris décimaux) sont affichées sur l'axe correspondant.

Onglet Données

L'onglet Graphique est l'un des trois onglets figurant à gauche d'un bloc graphique, et montre les données ayant généré le graphique. Vous sélectionnez des cellules, des lignes ou des colonnes comme dans n'importe quelle table de données.

Si vous modifiez des données de la table, le tracé reflétera les modifications. Vous pouvez modifier les nombres à la main, coller des lignes, etc. Utilisez cette fonctionnalité pour voir l'effet de certaines données, ou pour tracer une courbe de référence.

La couleur des en-têtes du tableau est la même que celle des tracés, et elle est affectée par les choix de [l'éditeur de courbes](#).

Equations... si affinités

ExtendSim comporte plusieurs blocs dont la logique se décrit sous forme d'une équation, que ce soit dans des modèles continus ou discrets. De deux lignes ou de deux cents lignes, ces équations sont comme des macros ou de petits programmes. Ce sont des outils très puissants pour accéder à des fonctions non représentées explicitement par les blocs ExtendSim, ou lorsque vous voulez combiner en un bloc des opérations diverses.

Cinq blocs permettent d'utiliser la puissance des équations :

- **Equation** (bib. Value) et son pendant discret, **Equation(I)** : équation de base.
- **Query Equation** (bib. Value) et son pendant discret, **Query Equation(I)** : équation en lien avec la base de données.
- **Queue Equation** (bib. Item) : une file d'attente dont la logique s'appuie sur une équation.

Composants des blocs à équations

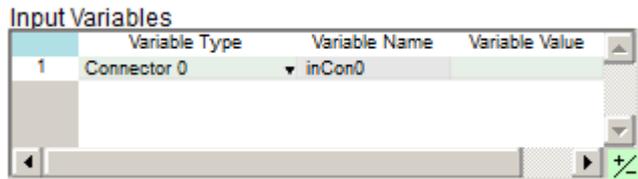
Les composants d'une équation sont les variables d'entrée, l'équation et les variables de sortie.

Tous les blocs basés sur des équations comportent une zone de défilement pour saisir l'équation. La plupart de ces blocs ont aussi un Editeur d'équation séparé pour visualiser et éditer l'équation.

Alors que vous pouvez définir des variables locales (temporaires) en entrée et en sortie dans la zone d'équation ou dans l'Editeur d'équation, la plupart des blocs basés sur les équations possèdent aussi deux tables, Input Variables et Output Variables, pour spécifier les variables statiques (permanentes). Comme décrit ci-dessous, ExtendSim fournit plusieurs types de variables prédéfinies en entrée et en sorties que vous pouvez utiliser dans les équations.

Variables d'entrée

Les variables d'entrée sont les valeurs du modèle utilisées dans une équation. À l'exception des blocs Buttons et Optimizer, les blocs basés sur les équations possèdent une table pour créer des variables d'entrée. Chaque ligne de la table Input Variables (montré à droite) comporte un menu déroulant pour sélectionner un type de variable d'entrée prédéfini et un champ pour son nom et sa valeur. Les variables créées ici sont statiques (permanentes).



Ajoutez des lignes à cette table en cliquant sur le bouton de redimensionnement +/- dans le coin en bas à droite de la table ; vous pouvez sélectionner une option différente pour chaque ligne.

Le menu déroulant *Type* fournit plusieurs options pour les variables d'entrée, comme montré ci-dessous. Pour plus d'informations sur chaque option, voir l'aide des blocs.

Type de variable d'entrée	Equation	Equation(I)	Queue Equation	Query Equation	Query Equation (I)
Connector – connecteur d'entrée continu	X	X	X	X	X
DB read value – lit une valeur à partir d'un emplacement fixe dans une table de BDD. Spécifiez son emplacement dans la colonne Valeur (Value).	X	X	X	X	X
DB read value using attrib – lit une valeur à partir d'un emplacement variable de la BDD spécifié par un attribut DB Address		X	X		X

DB read PRI – lit le Parent Record Index (PRI = Index d'enregistrement parent ou IEP) à partir d'une cellule dans un champ Enfant. L'emplacement de lecture est fixé et spécifié dans la colonne Value. Voir note ci-dessous.	X	X	X	X	X
DB read PRI using attrib – lit le PRI à partir d'un emplacement variable de BDD spécifié par un attribut DB Address. Voir note ci-dessous.		X	X		X
DB address – une adresse BDD choisie dans la colonne Value.	X	X	X	X	X
DB database index – un index BDD choisi dans la colonne Value.	X	X	X	X	X
DB table index – une table BDD choisie dans la colonne Value.	X	X	X	X	X
DB field index – un champ BDD choisi dans la colonne Value.	X	X	X	X	X
DB record index – un enregistrement BDD choisi dans la colonne Value.	X	X	X	X	X
Static first run init – une variable statique qui n'est initialisée à sa valeur de départ qu'au début de la première simulation. Spécifiez une valeur de départ dans la colonne Value.	X	X	X	X	X
Static mult run init – une variable statique qui est initialisée à sa valeur de départ au début de chaque simulation. Spécifiez une valeur de départ dans la colonne Value.	X	X	X	X	X
Static open model init – une variable statique qui est initialisée à sa valeur de départ quand le modèle est ouvert. Spécifiez une valeur de départ dans la colonne Value.	X			X	
Attribute – la valeur de l'attribut nommé dans Variable Name.		X	X		X
Quantity – la valeur de la quantité de l'entité.		X	X		X
Priority – la valeur de la priorité de l'entité.		X	X		X
Item index – la valeur de l'index de l'entité issue de la matrice des entités de l'Executive.		X	X		X
3D object ID – l'index de l'objet 3D de l'entité.		X	X		X

Pour la colonne *Variable Name*, vous pouvez utiliser les noms par défaut, assigner des noms qui sont les plus pertinents pour le modèle ou sélectionner un nom depuis un menu déroulant, selon le type de variable sélectionné. Le type de variable d'entrée sélectionné détermine également les options pour le champ Value : entrer une valeur directement, sélectionner la valeur depuis un emplacement de la BDD et ainsi de suite.

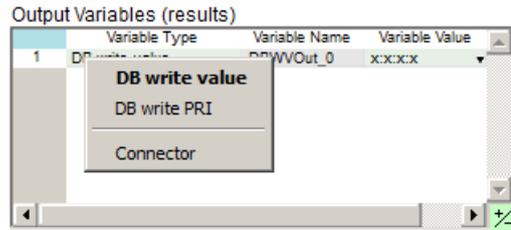
→ La notation PRI (Parent Record Index = index enregistrement parent ou IEP) est utilisée dans les blocs qui sont reliés à la BDD ExtendSim. Elle décrit le type de valeur qui est lu ou écrit lorsqu'il s'agit d'un champ Enfant.

→ Les blocs Queue Equation, Query Equation et Query Equation (I) possèdent des types de variables d'entrée spécifiques supplémentaires. Voir la suite de ce chapitre.

Variables de sortie

Les variables de sortie sont l'endroit où les résultats sont enregistrés quand l'équation est calculée. À

À l'exception des blocs Buttons et Optimizer, les blocs basés sur les équations comportent une table pour créer les variables de sortie. Chaque ligne dans la table Output Variables (montré ci-contre à droite) possède un menu déroulant pour sélectionner un type de variable de sortie prédéfini et un champ pour son nom et sa valeur. Les variables créées ici sont statiques (permanentes).



Ajoutez des lignes à cette table en cliquant sur le bouton de redimensionnement +/- dans le coin en bas à droite de la table ; vous pouvez sélectionner une option différente pour chaque ligne.

Le menu déroulant *Variable Type* fournit plusieurs options pour les variables de sortie, comme montré ci-dessous. Pour plus d'informations sur chaque option, voir l'aide des blocs.

Type de variables de sortie	Equation	Equation(I)	Queue Equation	Query Equation	Query Equation (I)
Connector – connecteur sortie continu	X	X	X	X	X
DB write value – écrit une valeur dans un emplacement fixe d'une table de BDD. Spécifiez son emplacement dans la colonne Value.	X	X	X	X	X
DB write value using attrib – écrit une valeur dans un emplacement non-fixe d'une BDD spécifié par un attribut DB Address		X	X		X
DB write PRI – écrit un Parent Record Index (PRI) dans un emplacement fixe d'une table de BDD. Spécifiez son emplacement dans la colonne Value. Le PRI est l'index de la cellule parent qui est écrite à l'enfant.	X	X	X	X	X
DB write PRI using attrib – écrit le PRI (voir ci-dessus) dans un emplacement non-fixe d'une BDD spécifié par un attribut DB Address. Voir la note ci-dessous.		X	X		X
Attribute – la valeur de l'attribut nommé dans Variable Name.		X	X		X
Quantity – la valeur de la quantité de l'entité.		X	X		X
Priority – la valeur de la priorité de l'entité.		X	X		X
3D object ID – l'index de l'objet 3D de l'entité.		X	X		X

Pour la colonne *Variable Name*, vous pouvez utiliser les noms par défaut, assigner des noms qui sont les plus pertinents pour le modèle ou sélectionner un nom depuis un menu déroulant, selon le type de variable sélectionné. Le champ Value présente les résultats pour la variable de sortie sélectionnée. Dans le bloc Equation(I) il y a aussi une colonne pour spécifier ce qu'il devrait arriver au résultat si des données de sortie sont demandées mais qu'il n'y a pas d'entité pour déclencher le nouveau calcul de l'équation.

→ La notation PRI (Parent Record Index = index enregistrement parent ou IEP) est utilisée dans les blocs qui sont reliés à la BDD ExtendSim. Elle décrit le type de valeur qui est lu ou écrit lorsqu'il s'agit d'un champ Enfant.

- ☞ Les blocs Queue Equation, Query Equation et Query Equation (I) possèdent des types de variables d'entrée spécifiques supplémentaires. Voir la suite de ce chapitre.

Editeur d'équations

Comme évoqué précédemment, tous les blocs basés sur des équations comportent une zone de défilement pour entrer l'équation. À l'exception des blocs Optimizer (bibliothèque Value) et Buttons (bibliothèque Utilities), les blocs basés sur les équations possèdent une fenêtre Editeur d'équation séparée pour voir et éditer les équations.

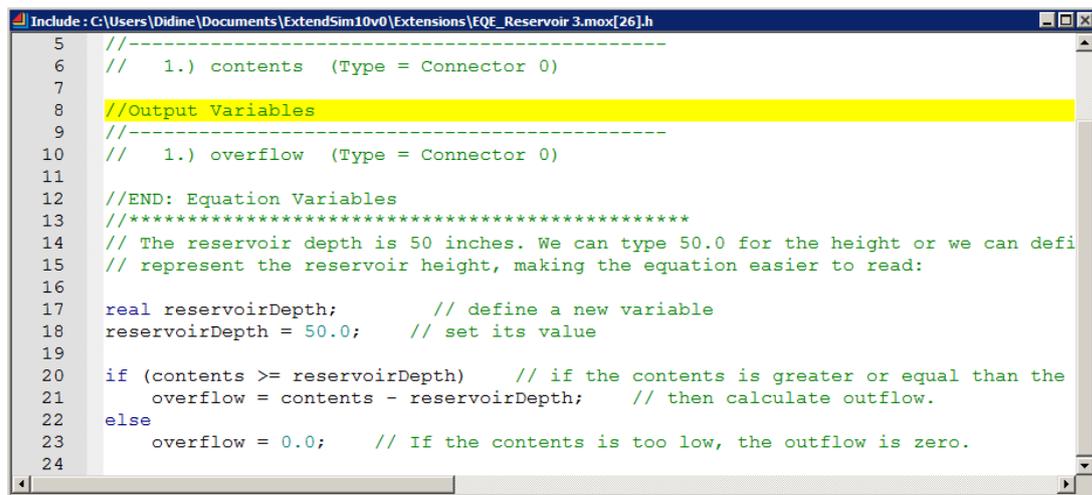
Bien que l'équation puisse être éditée dans le dialogue du bloc, si l'équation est un peu longue, il est avantageux de l'éditer dans la fenêtre Editeur d'équation pour plusieurs raisons :

- La fenêtre Editeur d'équation peut être redimensionnée pour s'adapter à de plus longues équations
- Les tabulations peuvent être utilisées pour l'indentation
- La colorisation du code est autorisée, pour que le code soit plus compréhensible
- Les autres commandes d'édition, telles que les accolades, sont supportées

→ La fonction de suggestion de code est activée dans la zone d'équation et la fenêtre de l'Editeur d'équation. L'affichage de suggestions au fur et à mesure de la saisie rend plus facile le rappel des noms des fonctions et de leurs arguments.

Pour voir l'Editeur d'équations :

- Ouvrez le modèle Reservoir 3 situé dans \Exemples\Tutorials\Continuous
- Dans le modèle, ouvrez le dialogue du bloc Equation
- Dans le dialogue du bloc, cliquez sur Open/Close Equation Editor. La fenêtre qui s'ouvre est montrée par la capture d'écran ci-dessous



```

5 //-----
6 // 1.) contents (Type = Connector 0)
7
8 //Output Variables
9 //-----
10 // 1.) overflow (Type = Connector 0)
11
12 //END: Equation Variables
13 //*****
14 // The reservoir depth is 50 inches. We can type 50.0 for the height or we can defi
15 // represent the reservoir height, making the equation easier to read:
16
17 real reservoirDepth; // define a new variable
18 reservoirDepth = 50.0; // set its value
19
20 if (contents >= reservoirDepth) // if the contents is greater or equal than the
21     overflow = contents - reservoirDepth; // then calculate outflow.
22 else
23     overflow = 0.0; // If the contents is too low, the outflow is zero.
24

```

Fenêtre de l'Editeur d'équations montrant une équation

Comme vous pouvez le voir, le contenu de la fenêtre utilise automatiquement les caractéristiques d'indentation et de colorisation du code. Utilisez la suggestion de code pour rechercher une fonction et obtenir la liste des arguments corrects. Pour cela, tapez la première partie du nom de la fonction dans la portion du code, puis faites dérouler la liste qui s'affiche jusqu'à que la fonction souhaitée apparaisse.

Fichiers include pour les équations

Il est fréquent que plusieurs blocs utilisent des fonctions et définitions de variables similaires ou identiques. Les fichiers include simplifient les tâches de programmation qui se répètent dans plusieurs blocs. Ils sont particulièrement utiles quand vous définissez vos propres fonctions pour un bloc et si vous voulez que les autres blocs aient accès à ces fonctions.

Pour ceux qui construisent des équations complexes, la plupart des blocs basés sur les équations ont une interface pour ajouter des fichiers include. Cette caractéristique est implémentée dans les blocs Equation et Query Equation (bibliothèque Value), Equation(I) ainsi que Queue Equation et Query Equation (I) (bibliothèque Item).

Les fichiers include dans les équations sont similaires aux fichiers include ModL standards utilisés lors de la création d'un bloc et examinés dans le Developer Reference. Ils peuvent contenir des définitions de symboles de ModL comme *#define*, des opérateurs de compilation conditionnels comme *#ifdef* et des définitions de fonctions.

Les principales différences entre les deux manières d'utiliser les fichiers include sont :

- Les fichiers include utilisés avec les équations sont normalement enregistrés au même endroit que le modèle qui les utilise ; ceci rend plus facile le déplacement du modèle et des include qu'il utilise dans un emplacement différent. Les fichiers include utilisés dans le code du bloc devraient être enregistrés dans le dossier Extensions/Includes.
- Les fichiers include utilisés dans les équations ne supportent que les déclarations variables locales. Ils ne devraient pas avoir de déclarations variables statiques (variables déclarées hors des limites d'une fonction ou d'une définition de procédure). Les fichiers include utilisés dans le code des blocs ont parfois des variables statiques ; celles-ci sont habituellement placées en haut du code du bloc ModL.
- Les fichiers include sont ajoutés au code des blocs via la syntaxe *#include*, comme par exemple *#include « fichier.h »*. Cependant, les fichiers include dans les équations sont ajoutés via une interface (*Utiliser fichiers include*) dans le dialogue du bloc Equation. Vous ne pouvez pas utiliser la syntaxe *#include* dans l'équation d'un bloc.

Pour créer un nouveau fichier include pour un bloc basé sur les équations :

- Cochez **Use include files** dans l'onglet Equation du bloc
 - Cliquer sur le bouton **Create new**
 - Donnez un nom au nouveau fichier include
 - Dans la fenêtre qui apparaît, entrez la procédure de la fonction. Par exemple :


```
//*****
//START: Equation Include Prototypes
Procédure myproc() ;//ceci sera affiché dans la liste des prototypes
//END: Equation Include Prototypes
//*****
Procédure myproc()
{
  userError("Bonjour tout le monde") ;
}
```
 - Utilisez la procédure dans l'équation. Dans ce cas, quand l'équation est appelée le bloc affichera une erreur utilisateur qui dit Bonjour tout le monde.
- Si vous possédez ExtendSim DE ou ExtendSim PRO, regardez aussi le modèle Equation Include File/Fichier include de l'équation ; il se situe dans \Exemples\Item Event\Tips.

Bloc Queue Equation

Le bloc Queue Equation (bibliothèque Item) est utilisé pour libérer des entités dans un ordre basé sur une règle de classement. Les entités entrent dans la file et sont mémorisées selon leur ordre d'arrivée (FIFO). Cependant, au lieu que les entités partent dans l'ordre FIFO, une équation dans le dialogue du bloc détermine quelle serait la prochaine entité (s'il y en a) qui devrait quitter la file. L'équation est calculée automatiquement une fois pour chaque entité dans la file et les résultats sont utilisés pour assigner un classement à chacune de ces entités. L'entité la mieux classée est celle qui est autorisée à partir ensuite.

Fonctionnement du bloc

Il est important de savoir que seuls les résultats des équations de l'entité gagnante sont utilisés pour ce cycle de calcul particulier. Par exemple, si la file contient actuellement 10 entités, l'équation sera calculée 10 fois et 10 ensembles individuels de résultats d'équation seront collectés. Mais seuls les résultats d'équation de l'entité la mieux classée seront utilisés ; les résultats des 9 autres sont rejetés.

Afin que la prochaine entité à partir soit choisie convenablement, un nouveau cycle de calcul est initié par défaut chaque fois qu'une entité entre ou sort de la file ou quand un connecteur d'entrée reçoit un message. Pour passer outre le comportement par défaut, les facteurs qui déclenchent les cycles de calcul peuvent être contrôlés grâce à des cases à cocher dans l'onglet Option du bloc.

Les tables ci-dessous décrivent un nombre de variables d'entrée et de sortie pour les équations qui

sont propres au bloc Queue Equation. En plus des variables d'entrée et de sortie habituelles décrites dans [Composants des blocs à équations](#), ces variables ont été créées pour aider les utilisateurs à écrire des équations qui sélectionneront correctement la prochaine entité qui doit partir. Plus particulièrement, la variable *QEQ item rank* (=rang de l'entité dans la file) est utilisée pour assigner un classement à chaque entité dans la file.

→ Au moins une variable de sortie QEQ item rank doit être définie pour que ce bloc puisse fonctionner correctement.

Une fois qu'un rang a été attribué à chaque entité, celles-ci sont libérées de la file selon la règle de classement sélectionnée dans le menu déroulant Release de l'onglet Options ; les règles de classement sont listées ci-dessous. Le bloc garde en mémoire la position attribuée à chaque entité.

→ Attribuer un rang vide (Blank) à une entité fait de celle-ci une candidate inéligible pour quitter la file pendant ce cycle particulier d'évaluation.

Variables

Le bloc Queue Equation propose plusieurs types de variables d'entrée et de sortie. Le début de ce chapitre a décrit les variables habituellement partagées par tous les blocs basés sur des équations. Les variables suivantes sont particulières au bloc Queue Equation.

Variables d'entrée

Variable d'entrée	Utilisations
QEQ arrival time (Temps d'arrivée)	L'heure à laquelle l'entité est arrivée dans la file
QEQ FIFO position (Position FIFO)	La position FIFO (first-in, first-out) dans la file
QEQ current best item rank (Meilleur rang actuel de l'entité)	Le meilleur (le plus haut ou le plus bas) résultat de rang pour le cycle de calcul en cours
QEQ attrib last item to exit (Attribut dernière entité à partir)	Fournit la valeur d'attribut choisi de la dernière entité sortie du bloc
QEQ num items in queue (Nombre d'entités dans la file)	Le nombre d'entités actuellement dans la file
QEQ static calc cycle init (Début du cycle de calcul statique)	Une variable statique qui est initialisée à sa valeur de départ au début de chaque cycle de calcul. Les changements apportés à cette variable restent fixes (statiques) à travers les calculs de l'équation.

Variables de sortie

Variable de sortie	Utilisation
QEQ item rank (Rang de l'entité)	Définit chaque rang (position) de l'entité dans la file. <i>Au moins une des variables de sortie doit être de ce type.</i>
QEQ halt calculation cycle (Interruption du cycle de calcul)	Si cette variable vaut Vrai, elle va prématurément arrêter le cycle de calcul en cours.

Comme il a été évoqué précédemment, vous pouvez spécifier d'autres résultats d'équation mais au moins une des variables de sortie doit être de type *QEQ item rank*. De plus :

- Seuls les résultats pour l'entité ayant le rang le plus haut seront émis.
- Vous pouvez utiliser plus d'une variable QEQ item rank ; la variable de classement secondaire sera utilisée pour départager les entités ayant le même rang.

→ Ces variables ne sont disponibles que pour l'entité libérée.

Règles de classement

Le menu déroulant Release de l'onglet Options propose les options suivantes pour déterminer les entités qui doivent être libérées en premier :

- **Highest ranked item** (Entités de plus haut résultat d'abord). La valeur de rang de chaque entité est calculée à partir de l'équation et l'entité ayant le plus haut résultat est autorisée à partir.

- **Lowest ranked item** (Entités de plus bas résultat d'abord). La valeur de rang de chaque entité est calculée à partir de l'équation et l'entité ayant le plus bas résultat est autorisée à partir.
- **The first True ranked item** (Entité de premier résultat Vrai d'abord). L'équation sera calculée une fois pour chaque entité jusqu'à qu'une entité reçoive un rang Vrai ou qu'il n'y ait plus d'entité restante à évaluer.
- **All True ranked items** (Toutes les entités de résultat Vrai). Toutes les entités de résultat Vrai sont déplacées dans une liste interne de départs où elles sont immédiatement et définitivement disponibles pour quitter le bloc.
 - Les résultats sont enregistrés pour chaque entité dans la liste des départs. Ces résultats sont utilisés (ou diffusés) au moment où l'entité sort de la file.
 - Les entités dans la liste des départs ne sont plus évaluées par l'équation. Ceci peut réduire le nombre de fois qu'une entité est évaluée, réduisant le temps de simulation.
 - Les entités sont libérées dans l'ordre FIFO.

Blocs Query Equation et Query Equation (I)

Les blocs Query Equation sont utilisés pour classer les enregistrements dans une table de BDD ExtendSim et sélectionner intelligemment un enregistrement, en fonction des règles de classement. Une équation définie par l'utilisateur dans le dialogue du bloc est calculée une fois pour chaque enregistrement dans la table ; les résultats servent à assigner un rang à chaque enregistrement. L'enregistrement ayant le plus haut rang est celui qui est sélectionné.

Les blocs Query Equation sont utilisés quand une BDD contient des informations requises pour prendre des décisions dans un modèle. L'évaluation et le classement pourraient être accomplis en utilisant les blocs Equation, mais les équations seraient longues et compliquées. En outre, les blocs Query Equation fournissent des informations spécifiques permettant d'écrire des requêtes plus complexes.

Il existe deux blocs Query Equation :

- Query Equation (bibliothèque Value)
- Query Equation (I) (bibliothèque Item)

Les différences majeures entre ces deux blocs sont présentées plus loin.

Fonctionnement des blocs

Un *cycle de requête* est le point temporel où un bloc Query Equation exécute son équation pour choisir le prochain enregistrement « gagnant ».

- Dans le cas du bloc Query Equation (I), ceci se produit chaque fois qu'une entité le traverse.
- Pour le bloc Query Equation, l'onglet Options propose des choix pour contrôler l'initiation du cycle de requête.

Il est important de savoir que seuls les résultats des équations pour l'enregistrement gagnant sont utilisés pour un cycle de requête particulier. Par exemple, si la table de requête contient 10 enregistrements, l'équation sera calculée 10 fois et 10 ensembles individuels de résultats d'équation seront collectés. Cependant, seuls les résultats de l'enregistrement le mieux classé seront utilisés et les résultats des 9 autres sont oubliés.

Les types de variables d'entrée et de sortie des blocs sont spécialement conçus pour aider les modélisateurs à écrire des équations qui vont classer les enregistrements correctement et sélectionner l'enregistrement gagnant. L'équation définie par l'utilisateur est calculée une fois pour chaque enregistrement afin que chacun obtienne un classement.

Les enregistrements sont classés selon la règle de classement sélectionnée dans l'onglet Options du bloc ; ces règles sont listées au paragraphe [Règles de classement](#). La structure interne des données du bloc garde une trace du classement et de tout autre résultat de l'équation pour chaque enregistrement.

→ Au moins une variable de sortie *DBQ record rank* (*rang de l'enregistrement*) est requise pour que chaque enregistrement puisse avoir un classement.

Entités Mémo (Spawn)

Une caractéristique unique du bloc Query Equation (I) est la capacité de pouvoir créer des entités supplémentaires (optionnel). Ces entités Spawn (Mémo) peuvent ensuite être utilisées dans le modèle

à des fins spéciales. C'est un système indépendant et parallèle à la façon dont les entités qui traversent le bloc sont créées.

Généralement les entités sont créées hors du bloc puis le traversent. Ces entités sont appelées *entités de passage*. Dans certains cas, vous voudrez peut-être créer des entités supplémentaires, appelées entités Spawn (ou Mémo), basées sur la logique d'équations et ce qui a été trouvé dans la table de requête. Si un Spawn est créé pour un enregistrement particulier, l'équation peut être définie de telle sorte que les résultats précis de l'enregistrement Spawn puissent être donnés comme attributs de l'entité Spawn.

Pour utiliser cette caractéristique, cochez *Enable the spanning of items* dans l'onglet Options. Ceci provoque l'apparition d'un nouveau connecteur de sortie discret tout en haut de l'icône du bloc ainsi que la disponibilité de types de variables d'équation supplémentaires, avec le préfixe DBQS (pour database query spawn). Ces variables sont montrées dans la table ci-dessous.

→ Les résultats d'équation pour les entités de passage sont calculés indépendamment de ceux pour les Spawn.

Les résultats associés à l'entité de passage peuvent être différents des résultats de la génération d'entités Spawn. Par exemple, les résultats pour l'entité de passage peuvent provenir de l'enregistrement 6 (parce que 6 possédait le meilleur rang d'enregistrement de passage), tandis que le Spawn pourrait utiliser les résultats de l'enregistrement 2 (parce que 2 possédait le meilleur rang d'enregistrement de Spawn).

Quand une entité Spawn est créée, elle représente un enregistrement de la table de requête. Par conséquent, vous pouvez imaginer cette entité nouvellement créée comme « un enregistrement Spawn ».

Variables

Les blocs Query Equation ont plusieurs types de variables d'entrée et de sortie. Le début de ce chapitre a décrit les variables habituellement partagées par tous les blocs basés sur des équations. Les variables suivantes sont propres aux blocs Query Equation.

Variables d'entrée

Variable d'entrée	Query Equation	Query Equation (I)	Utilisations
DBQ read value	X	X	Demande à l'utilisateur de choisir un champ dans la table de requête. Au fur et à mesure que le bloc Query Equation commence à parcourir chaque enregistrement, cette variable prend automatiquement la valeur trouvée dans ce champ pour l'enregistrement en cours. (C'est plus facile que d'appeler <code>DBDataGetAsNumber()</code> pour faire référence à des informations situées dans la table de requête pour l'enregistrement en cours.)
DBQ read PRI	X	X	La valeur du Parent Record Index (Index d'enregistrement parent) depuis le champ spécifié pour l'enregistrement en cours. Voir note ci-dessous.
DBQ start record	X	X	L'enregistrement où commencer le cycle de requête. (Par défaut commence à l'enregistrement 1).
DBQ num reports	X	X	Le nombre d'enregistrements dans la table de requête.
DBQ current record index	X	X	L'index de l'enregistrement actuellement évalué lors du cycle de requête.
DBQ static query init	X	X	Une variable d'entrée statique qui est initialisée au début de chaque cycle de requête.
DBQ current best	X	X	Meilleur résultat de rang actuel en cours dans le cycle de

rank result			requête en cours.
DBQ num non-blank ranks	X	X	Nombre d'enregistrements classés non-nuls à ce stade dans le cycle de requête en cours.
DBQS current best rank result		X	Meilleur résultat de rang Spawn actuel en cours dans le cycle de requête en cours.
DBQS num non-blank ranks		X	Nombre d'entités Spawn classées non-nulles à ce stade dans le cycle de requête en cours.

→ La notation PRI (Parent Record Index = index enregistrement parent ou IEP) est utilisée dans les blocs en relation avec la BDD d'ExtendSim. Elle décrit le type de valeur qui est lu ou écrit quand il s'agit d'un champ Enfant.

Variables de sortie

Variable de sortie	Query Equation	Query Equation (I)	Utilisations
DBQ record rank	X	X	Résultat du rang pour l'enregistrement en cours
DBQ halt query	X	X	Une valeur Vrai interrompt le cycle de requête en cours
DBQ next record	X	X	Indique au bloc à quel enregistrement suivant appliquer la requête
DBQS record rank		X	Génère un rang pour l'enregistrement en cours
DBQS attribute		X	Pour assigner des valeurs d'attributs à une entité Mémo. Si le rang de génération de l'enregistrement est suffisant pour mériter une génération, alors les résultats de cet enregistrement peuvent être attachés à l'entité Spawn générée sous la forme d' <i>attributs DBQS</i> . Voir Entités Mémo (Spawn)
DBQS item quantity		X	Quantité à placer sur les entités Spawn. Voir aussi DBQS attribute ci-dessus.
DBQS item priority		X	Priorité à placer sur les entités Spawn. Voir aussi DBQS attribute ci-dessus.
DBQS 3D object ID		X	ID de l'objet 3D à placer sur les entités Spawn. Voir aussi DBQS attribute ci-dessus.

Comme mentionné plus tôt, même si vous pouvez spécifier d'autres résultats d'équation, au moins une des variables de sortie doit être de type *DBQ record rank*. De plus :

- Seuls les résultats de l'enregistrement ayant le meilleur classement seront émis.
- Vous pouvez utiliser plus d'une variable *DBQ record rank* ; la variable secondaire de classement sera utilisée pour départager ces valeurs en cas d'égalité au classement. C'est utile si vous êtes dans le cas où vous devez briser une égalité entre deux enregistrements ou plus possédant la même variable primaire. Ce qui définit une égalité peut aussi être défini dans l'onglet Options avec le paramètre *Records ranked within +/- are equal*.
- Si l'option de Génération d'entités Spawn est activée, au moins deux variables de sortie sont requises : une variable de type *DBQ record rank* plus une de type *DBQS record rank*.

Règles de classement

Les règles qui déterminent quel enregistrement est sélectionné sont indiquées dans l'onglet Options du bloc :

Pour une entité de passage

- **Highest pass-through record rank** (Rang de l'enregistrement de passage le plus élevé). L'équation calcule une valeur de rang pour chaque enregistrement. Celle qui a le rang le plus élevé est choisie.
- **Lowest pass-through record rank** (Rang de l'enregistrement de passage le plus bas). L'équation calcule une valeur de rang pour chaque enregistrement. Celle qui a le rang le plus bas est choisie.
- **First True pass-through record rank** (Premier enregistrement ayant le rang Vrai). L'équation sera calculée une fois pour chaque enregistrement jusqu'à que :
 - Un enregistrement reçoive un classement Vrai
 - Ou qu'il ne reste plus d'enregistrements à évaluer

Pour une entité Spawn

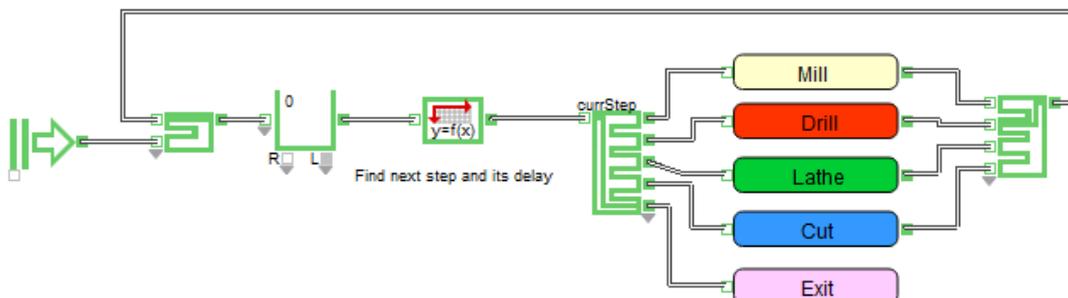
- **The highest spawn record rank** (Rang d'enregistrement Mémo le plus élevé). L'équation calcule une valeur de rang pour chaque enregistrement. Celle qui a le rang le plus élevé est choisie.
- **Lowest spawn record rank** (Rang d'enregistrement Mémo le plus bas). L'équation calcule une valeur de rang pour chaque enregistrement. Celle qui a le rang le plus bas est choisie.
- **Premier enregistrement Mémo de rang Vrai.** L'équation sera calculée une fois pour chaque enregistrement jusqu'à que :
 - Un enregistrement reçoive un classement Vrai
 - Ou qu'il ne reste plus d'enregistrements à évaluer
- **First True spawn record rank** (Tous les enregistrements Mémo Vrai). Tout enregistrement qui reçoit un rang mémo $\geq 0,5$ sera généré. Ceci rend possible la création de n'importe quel nombre d'entités Spawn pour une entité de passage.

Différences entre les blocs Query Equation

Description	Query Eq	Query Eq (I)
Cycle de requête initié seulement quand une entité arrive (similaire au bloc Equation(I)).		X
L'initiation du cycle de requête est contrôlée en utilisant les options de l'onglet Options (similaire au bloc Equation).	X	
Supporte la génération d'entités Spawn		X
Fonctionne dans les modèles continus	X	
Fonctionne dans les modèles discrets	X	X
Fonctionne dans les modèles Rate (hybrides)	X	X

Modèle DB Job Shop Query(I) Lookup

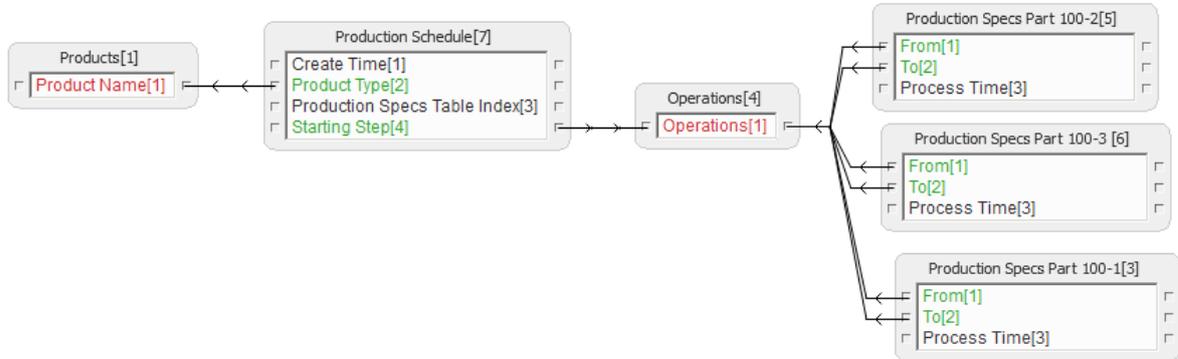
Le modèle DB Job Shop Query(I) Lookup est situé dans \Exemples\Item Event\Routing.



Modèle DB Job Shop Query(I) Lookup

Le modèle simule trois types de pièces, chacun ayant sa propre séquence opératoire, orientées à travers cinq process potentiels. Selon la pièce, une séquence est composée d'une combinaison de fraisage, de perçage, de lattage, de coupage et de sortie. Par exemple, la séquence pour la pièce 100-3 est découpe, laminage, tournage, perçage et fin, alors que la séquence pour la pièce 100-1 ne contient qu'une étape de laminage avant la sortie. Chaque étape du process a un délai aléatoire qui est unique pour chaque type de pièce.

Toutes les informations dont vous avez besoin pour diriger ce modèle sont situées dans la BDD ExtendSim nommée Database 1. La structure de cette BDD est montrée ici.



Structure de la Database 1

- La table Products contient les noms des trois pièces qui sont modélisées.
- La table Production Schedule est utilisée par le bloc Create (grâce à un lien) pour créer des pièces selon un programme.
- La table Operations contient une liste des cinq différentes opérations potentielles : laminage, tournage, perçage, découpe et fin.
- Chacun des trois types de pièces a une table Production Specs associée. Cela sert à définir la gamme opératoire d'une pièce et les délais associés à chaque étape. Les champs From et To sont utilisés pour définir la destination appropriée selon une origine particulière : « Si tu viens d'ici, va ensuite là-bas ».

Dans le modèle, un bloc Query Equation (I) est utilisé pour trouver la prochaine étape du process de la pièce en cours et le délai qui lui est associé. Les deux informations sont ensuite mémorisées en tant qu'attributs pour l'entité de passage. Quand l'entité/la pièce suivante arrive, le bloc Query Equation commence à parcourir la table Production Specs associée à cette pièce.

Dans cet exemple, le bloc Query Equation cherche l'enregistrement dont la valeur *From* correspond à l'attribut *currStep* trouvé pour cette entité. Une fois trouvé, le bloc trouve ensuite les valeurs *To* et *Process Time* associées à cet enregistrement et les mémorise en tant qu'attributs de cette entité.

L'équation dans le bloc Query Equation est :

```
if(currStep == fromField)
{
    rank = TRUE;
    currStep = toField;
    activityDelay = delayField;
}
else
{
    rank = False;
}
```

Etant donné la complexité de ce que le bloc Query Equation peut faire, le code est très simple.

L'équation est entrée dans le bloc sous la forme "résultat = formule;". La partie droite de l'équation est la formule. Une formule peut combiner des valeurs, des noms, des variables, des opérateurs, des fonctions, des procédures et des structures de contrôle.

- Les valeurs sont des constantes, telles que 31 ou 5000.2
- Les noms et les variables servent à représenter des valeurs dans la formule. Par exemple, vous pouvez nommer la première entrée "revenu", puis utiliser "revenu" dans la formule comme vous

auriez utilisé une valeur. Vous avez un accès direct à des variables système, des informations de la base de données, en plus de valeurs reliées au bloc lui-même.

- Les opérateurs ExtendSim sont +, -, etc. Ils sont listés dans l'aide du bloc Equation.
- Le bouton Fonctions ModL du dialogue affiche la liste des fonctions et procédures intégrées à ExtendSim, en indiquant comment utiliser leurs arguments. Vous pouvez utiliser le Copier/Coller pour prendre la fonction de votre choix dans la liste. Vous devez bien entendu utiliser de vrais arguments, et les séparer par une virgule (,). Les parenthèses sont obligatoires pour indiquer où commencent et finissent les arguments. Fonctions et procédures sont expliquées dans le manuel de programmation.
- Les structures de contrôle de type "if-else" donnent une plus grande souplesse au bloc.

Vous devez observer les règles suivantes en utilisant ce bloc :

- Les connecteurs du bloc étant tous continus, il faut relier le bloc à des connecteurs continus.
- La forme doit toujours être `résultat = équation;` (notez le point-virgule).
- Une équation est limitée à 32767 caractères. Le passage à la ligne se fait automatiquement où à la demande par Maj-Entrée.
- Vous devez donner un nom aux connecteurs utilisés dans la formule. Si une entrée est liée, vous devez utiliser son nom dans votre équation. Lorsque vous lancez la simulation, ExtendSim vous prévient si vous avez lié des entrées qui ne sont pas utilisées dans l'équation. ExtendSim vous prévient aussi si vous avez utilisé un connecteur sans nom ou non lié. Vous pouvez utiliser chaque nom plusieurs fois dans l'équation.
- Les noms peuvent être alphanumériques, mais doivent commencer par une lettre ou le caractère souligné (). Vous ne pouvez inclure des espaces, des tirets ou d'autres caractères non-alphanumériques. Un nom ne peut avoir plus de 30 caractères.
- Aucune différence n'est faite entre majuscules et minuscules, donc "ACOS" = "AcOs" = "acos".
- Les espaces entre opérateurs sont ignorés :
`sortie = entrée1 + entrée2;`
est équivalent à :
`sortie=entrée1+entrée2;`
- Vous ne pouvez pas utiliser de fonctions ou de procédures utilisateur dans le bloc Equation.
- Vous devez utiliser le point comme symbole décimal, la virgule ayant déjà une fonction dans le langage ExtendSim.

Si vous connaissez un peu la programmation, ce bloc sera d'autant plus puissant entre vos mains. Voyez le langage ModL dans le manuel de programmation : vous pouvez l'utiliser directement dans le bloc Equation.

Conseils de mise au point et débogage

Il est rare qu'un modèle s'exécute correctement du premier coup. ExtendSim fournit plusieurs outils pour pister les erreurs du modèle lorsqu'il donne des résultats inattendus.

Animation

L'animation est souvent utile pour la mise au point. Si vous savez comment un bloc est animé, en le regardant vous pouvez vous rendre compte de son bon ou mauvais fonctionnement, par exemple en ne constatant aucun signe visible du passage d'entités.

Si l'animation va trop vite pour l'œil, le curseur permet de ralentir la simulation entre "Lent" (la tortue) et "Vite" (le lièvre).

Les blocs de mise au point

Les bibliothèques Value, Utilitaires et Item comportent des blocs utiles pour la mise au point, à savoir :

Bloc	Bibliothèque	Emploi
Display Value	Value	Affiche la valeur du connecteur d'entrée à chaque étape de la simulation. Vous pouvez voir les nombres sortis par un autre bloc dans le dialogue Value.
Notify	Value	Stoppe la simulation ou joue un son ou demande une valeur lorsqu'il reçoit une entrée supérieure ou inférieure à un certain seuil, et affiche un message spécifié dans le dialogue.
History	Item	Affiche des informations sur les entités qui ont transité par lui.
Information	Item	Affiche des informations sur une entité ou des valeurs provenant d'un autre bloc. Comme le bloc Display Value, sauf qu'il affiche la valeur, la priorité et les attributs des éléments discrets.
Record message	Utilitaires	Enregistre les messages passés par les connecteurs continus.
Item Messages	Utilitaires	Affiche des informations détaillées sur les messages envoyés lors de la transmission d'entités.

Souvenez-vous que vous pouvez placer des graphiques à n'importe quel point du modèle : cela permet de suivre l'évolution de certaines valeurs. Les blocs de statistiques fourniront aussi des informations utiles.

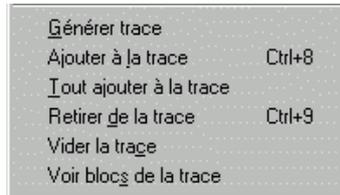
La trace d'un modèle

Les commandes de trace sont similaires aux commandes de rapport, mais le texte en sortie est bien plus important car les blocs écrivent des valeurs pour chaque étape de la simulation. Demander la trace d'un modèle permet de trouver des anomalies qu'on ne peut identifier qu'en exécution pas à pas.

Parce que la trace génère des masses d'informations, on hésite parfois à l'utiliser. En réduisant la trace aux seuls blocs qui vous intéressent, vous en ferez un outil très pratique.

En fin de simulation, le fichier Trace s'ouvre automatiquement. Si vous lancez la simulation plus d'une fois, chaque trace est ajoutée à la fin du fichier, et vous pouvez ainsi comparer plusieurs simulations.

Les commandes de trace, figurant dans le menu hiérarchique **Debuggage** du menu Simulation, sont :



Commandes de trace

Vous sélectionnez certains blocs pour la trace par la commande **Ajouter à la trace**. Tout ajouter à la trace permet d'inclure tous les blocs du modèle. Pour savoir quels blocs sont prévus dans la trace, sélectionnez la commande Voir blocs de la trace : le mot "Trace" s'affiche alors sur l'icône des blocs inclus dans la trace.

Etapes de la trace

- Pour inclure des blocs dans la trace, sélectionnez-les et choisissez la commande Ajouter à la trace. Si pour une prochaine simulation vous voulez moins de blocs, sélectionnez les blocs qui ne vous intéressent plus et choisissez la commande Retirer de la trace. Pour recommencer à zéro votre choix de blocs, choisissez la commande Vider trace.
- Choisissez la commande Générer la trace. ExtendSim vous demande le nom du fichier de trace.
- Lancez la simulation pour générer la trace.

Par exemple, si vous sélectionnez les deux blocs Queue et Activity d'un modèle, le début de la trace pourrait être:

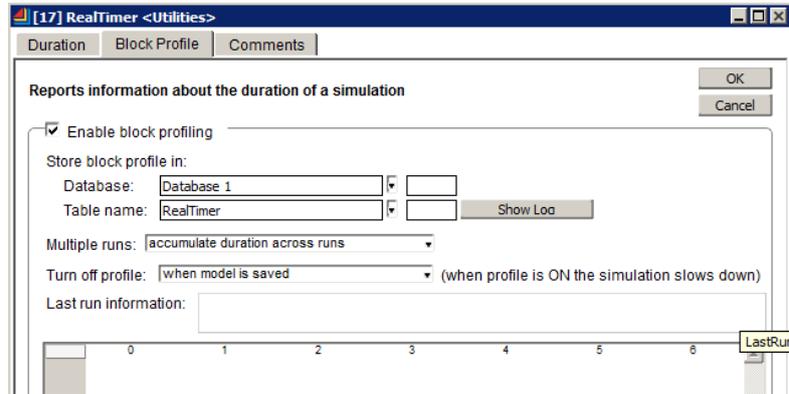
```
--> Item (1) departure from Queue number 12. CurrentTime:1.
Length: 1
--> Item (1) arrival to Activity number 9. CurrentTime:1.
Block Label: Mill
Delay (D): 0,818060245682408
Length: 0
--> Item (1) departure from Activity number 9. CurrentTime:1,81806024553953.
Block Label: Mill
Delay (D): 0,818060245682408
Length: 0
--> Item (4) departure from Queue number 12. CurrentTime:3,9999999994179.
Length: 1
--> Item (4) arrival to Activity number 9. CurrentTime:3,9999999994179.
Block Label: Mill
Delay (D): 1,24603935883896
Length: 0
--> Item (4) departure from Activity number 9. CurrentTime:5,24603935884079.
Block Label: Mill
Delay (D): 1,24603935883896
Length: 0
etc.
```

Si vous créez vos propres blocs, ils doivent comprendre une partie de script spéciale pour être inclus dans le fichier de Trace.

Profil d'un modèle

Vous pouvez demander le profil d'un modèle pour analyser la portion de temps utilisée par chaque bloc dans une simulation. C'est une information souvent utilisée par les programmeurs qui veulent déterminer s'il serait souhaitable d'optimiser des blocs spécifiques, mais qui renseignera aussi les non programmeurs sur les portions du modèle qui sont le plus utilisées.

Pour générer un fichier texte de profil, placez le bloc **RealTimer** (Utilities) dans le modèle, et activez le profilage des blocs.



RealTimer utilisé pour le profil des blocs

Voici par exemple le profil sur le modèle Car Wash Final exécuté sur 4800 minutes :

DB	Block name	Block label	Block #	Time (seconds)	Percent	Memory start	Memory diff at end
0	Executive	Executive	0	0	44,037%	119858,00	0,00
1	Activity	Wash Only	109	0	14,879%	8658,00	0,00
2	Queue	Wash & Wax Q	91	0	13,781%	47944,00	0,00
3	Resource Pool	Attendant Re...	28	0	13,781%	4094,00	0,00
4	Select Item Out	Route	77	0	13,781%	3540,00	0,00
5	Activity	Wash & Wax	136	0	0,000%	8658,00	0,00
6	Buttons		66	0	0,000%	2222,00	0,00
7	Create	Dirty Cars	3	0	0,000%	5098,00	0,00
8	Equation(I)		129	0	0,000%	8100,00	0,00
9	Exit	Exit	48	0	0,000%	2012,00	0,00
10	Line Chart	Chart	57	0	0,000%	30498,00	768,00
11	Queue	Wash Q	10	0	0,000%	47944,00	0,00
12	Random Num...		72	0	0,000%	2052,00	0,00
13	RealTimer		17	0	0,000%	2044,00	0,00
14	Resource Poo...	Release Reso...	37	0	0,000%	27986,00	0,00
15	Resource Poo...	Release Reso...	46	0	0,000%	27986,00	0,00

profil des blocs Car Wash Final

Les données sont enregistrées dans des bases de données ExtendSim, et un résumé pertinent affiché dans le dialogue du bloc.

Vous pouvez utiliser les informations du profil pour affiner le modèle ou repérer des résultats incohérents. Par exemple, si l'un des deux blocs Activity délai utilise un pourcentage de temps notablement supérieur aux deux autres, alors qu'ils sont censés être similaires, cette information vous sera utile pour repérer une différence.

Notez que les pourcentages sont approximatifs, donc leur somme peut ne pas être égale à 100%.

Déboguer des équations

Les blocs basés sur les équations vous permettent d'entrer des équations qui sont de simples opérations logiques (if x, then y) ou des calculs complexes qui contrôlent le modèle en utilisant les fonctions ModL. Alors que dans beaucoup de cas vous n'aurez pas besoin de déboguer l'équation, ExtendSim fournit un débogueur pour déterminer si les équations s'exécutent comme prévu. Il fonctionne comme le Débogueur ExtendSim pour développeurs, mais il est spécifique aux blocs basés sur les équations.

Le débogueur d'équation vous permet :

- De définir des points d'arrêt pour les équations. Un point d'arrêt est l'endroit où le débogueur stoppe l'exécution de l'équation pour que vous puissiez exécuter le code en pas à pas.
- De définir des conditions pour les points d'arrêt. La définition d'une condition de point d'arrêt fait que celui-ci n'arrête l'exécution que dans certaines circonstances.
- D'exécuter le code et examiner en pas à pas les variables.

Commencer le didacticiel

- Ouvrez le modèle Reservoir 3 situé dans \Exemples\Tutorials\Continuous
- Pour ne pas écraser le modèle original, enregistrez le modèle sous Reservoir 3 Débogage.

- Dans le modèle, ouvrez le dialogue du bloc Equation ; il est appelé *Calculate Overflow*
- ➔ Bien que le didacticiel suivant utilise un modèle continu, le débogueur d'équation peut être utilisé avec n'importe quel bloc basé sur les équations.

Définir des points d'arrêt

Pour intervenir sur chaque ligne de code, il est d'abord nécessaire de stopper l'exécution du code. Dans cet exemple, cela demande de placer un point d'arrêt avant l'opérateur « if » au début de l'équation :

Dans le dialogue du bloc Equation, cochez la case **Enable debugger**

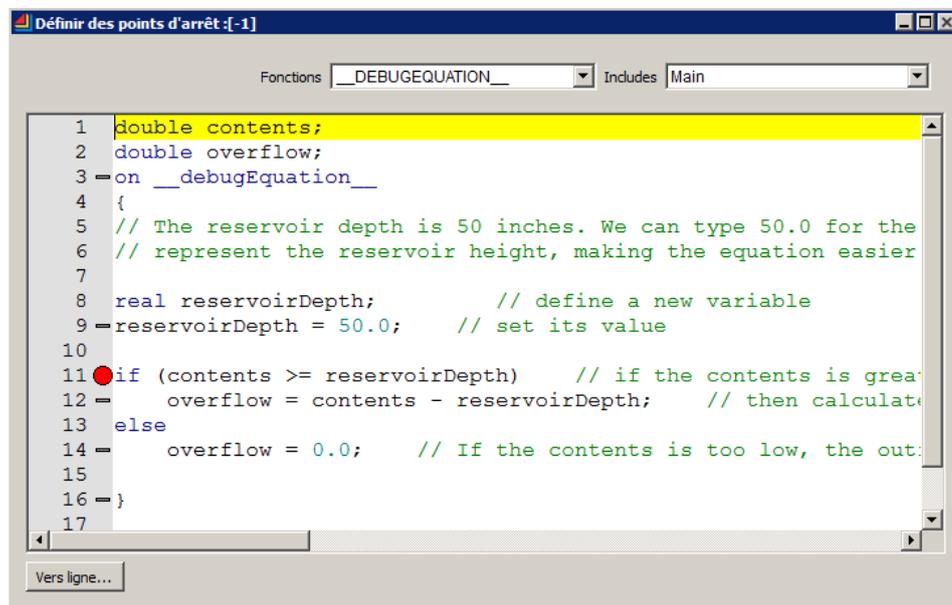
Cliquez sur le bouton **Set breakpoints**

L'équation est alors recompilée en mode débogage et deux fenêtres s'ouvrent :

- La fenêtre *Définir les points d'arrêt* est l'endroit où vous définissez les points d'arrêt
- La fenêtre *Points d'arrêt* liste les points d'arrêts qui ont été définis. Nous verrons comment l'utiliser pour [Définir un point d'arrêt conditionnel](#).

➔ Le code dans la fenêtre Définir les points d'arrêt ne peut être édité.

Les lignes grises dans la marge de gauche de la fenêtre Définir les points d'arrêt indiquent les seuls endroits où les points d'arrêt peuvent être placés.

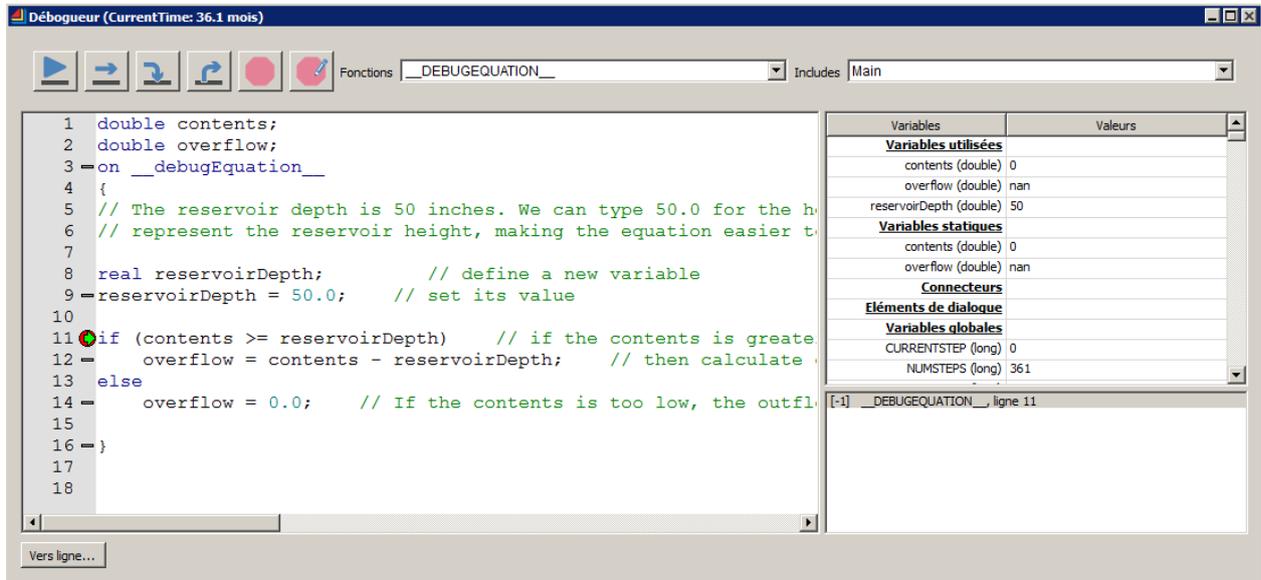


Fenêtre Définir les points d'arrêt montrant un point d'arrêt à « if »

- Dans la marge de gauche de la fenêtre Définir les points d'arrêt, cliquez sur la ligne de point d'arrêt qui est à gauche de l'opérateur *if*. Ceci transforme le tiret gris en rond rouge.
- Si vous cliquez sur la mauvaise ligne du code, cliquez simplement dessus à nouveau pour effacer le point d'arrêt.
- Fermez la fenêtre Définir les points d'arrêt.
- Lancez la simulation. La simulation va se dérouler jusqu'à que le point d'arrêt soit atteint.

Exécuter le code

Quand le point d'arrêt est atteint, la fenêtre de débogage s'ouvre.



Fenêtre de débogage montrant le point d'arrêt et les variables

- Dans la portion Source à gauche de la fenêtre, une flèche verte se trouve sur le point d'arrêt. Ceci indique que le code avant cette flèche a été exécuté.
- La flèche verte indique quelle ligne de code sera exécutée ensuite, une fois que la simulation aura repris.
- La portion Variables en haut à droite montre les variables utilisées dans l'équation et la valeur de chaque variable à l'état d'avant l'exécution de la ligne de code du point d'arrêt. Lorsque le curseur survole une variable dans la portion Source, cela affiche aussi sa valeur.
 - En haut de la fenêtre se trouvent six boutons. Ceux-ci servent à continuer la simulation, à Passer par-dessus, Rentrer dans ou Remonter jusqu'à un appel de fonction, et à Abandonner la simulation, en allant éventuellement éditer le code. (Ces boutons sont expliqués plus en détail dans le *Developer Reference*).
 - La portion Enchaînement des appels en bas à droite indique où se trouve le contrôle dans l'enchaînement des commandes. Elle n'est pas utilisée pour déboguer des équations à moins qu'il y ait des blocs dans le modèle qui ont un code de débogage.

L'opérateur « if » n'a pas encore été exécuté. Pour que l'opérateur s'exécute :

- Cliquez plusieurs fois sur le bouton **Passer** en haut de la fenêtre du Débogueur ; ce bouton est montré ci-contre. Autrement, utilisez la flèche de droite pour passer par-dessus la fonction. 

La flèche verte dans la portion Source indique l'avancement du code ; la portion Variables montre le changement de la valeur des variables.

- Faire glisser le contrôle sur une variable dans la portion Source montrera également sa valeur en cours.

Obtenir le calcul de dépassement

L'étape suivante est la définition d'un point d'arrêt pour découvrir quand la valeur de dépassement est d'abord calculée.

Dans la fenêtre du Debugger :

- Enlevez le premier point d'arrêt en cliquant sur le rond rouge qui le représente
- Ajoutez un nouveau point d'arrêt à la ligne « overflow = contents ... »
- Cliquez sur le bouton Continuer en haut de la fenêtre du Débogueur. Ou utilisez Maj+flèche de droite.

Bien que le point d'arrêt soit situé à la ligne de calcul du dépassement, la flèche verte indique que le calcul n'a pas encore été effectué. Ainsi la valeur de dépassement indiquée ne sera pas la valeur en cours.

Afin de voir la valeur de dépassement en cours :

- Cliquez sur le bouton Passer pour exécuter la ligne de code. La valeur de dépassement calculée est montrée dans la portion Variables.

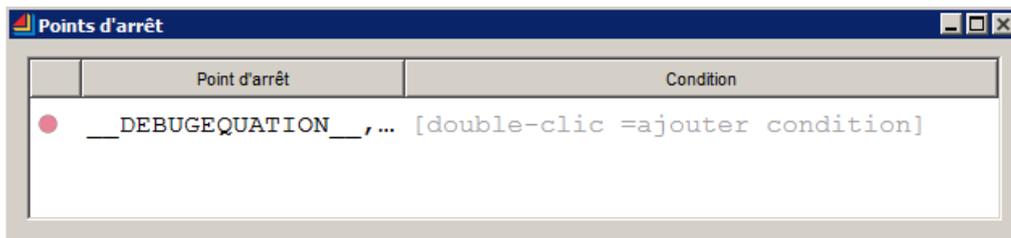
The screenshot shows a code editor on the left and a variables window on the right. The code is in C++ and defines a reservoir depth of 50.0. It checks if the current contents are greater than or equal to the reservoir depth. If so, it calculates the overflow as the difference between contents and reservoir depth. The variables window shows the current state of variables: 'contents' is 50.6550363200575759, 'overflow' is nan, and 'reservoirDepth' is 50. The current line of code is highlighted in yellow: `overflow = contents - reservoirDepth;`

Variables	Valeurs
Variables utilisées	
contents (double)	50.6550363200575759
overflow (double)	nan
reservoirDepth (double)	50
Variables statiques	
contents (double)	50.6550363200575759
overflow (double)	nan
Connecteurs	
Éléments de dialogue	
Variables globales	
CURRENTSTEP (long)	151
NUMSTEPS (long)	361

Fenêtre du Débogueur s'arrêtant au premier dépassement

Définir un point d'arrêt conditionnel

Le point d'arrêt conditionnel cause l'arrêt du calcul seulement quand la valeur de dépassement est calculée et comprise dans une gamme définie de valeurs spécifiques. La fenêtre Points d'arrêt sert à définir des conditions d'arrêts. Elle vous permet de :



Fenêtre Point d'arrêt

- Voir quels points d'arrêt ont été définis et où
- Ajouter des conditions à un point d'arrêt
- Cliquer sur le rond rouge d'un point d'arrêt pour désactiver ou activer tout point d'arrêt, sans le supprimer. (À la place du rond rouge, les points d'arrêt désactivés ont un rond ouvert).
- Supprimer un point d'arrêt en sélectionnant son nom et en appuyant sur la touche Supprimer ou Effacement arrière.

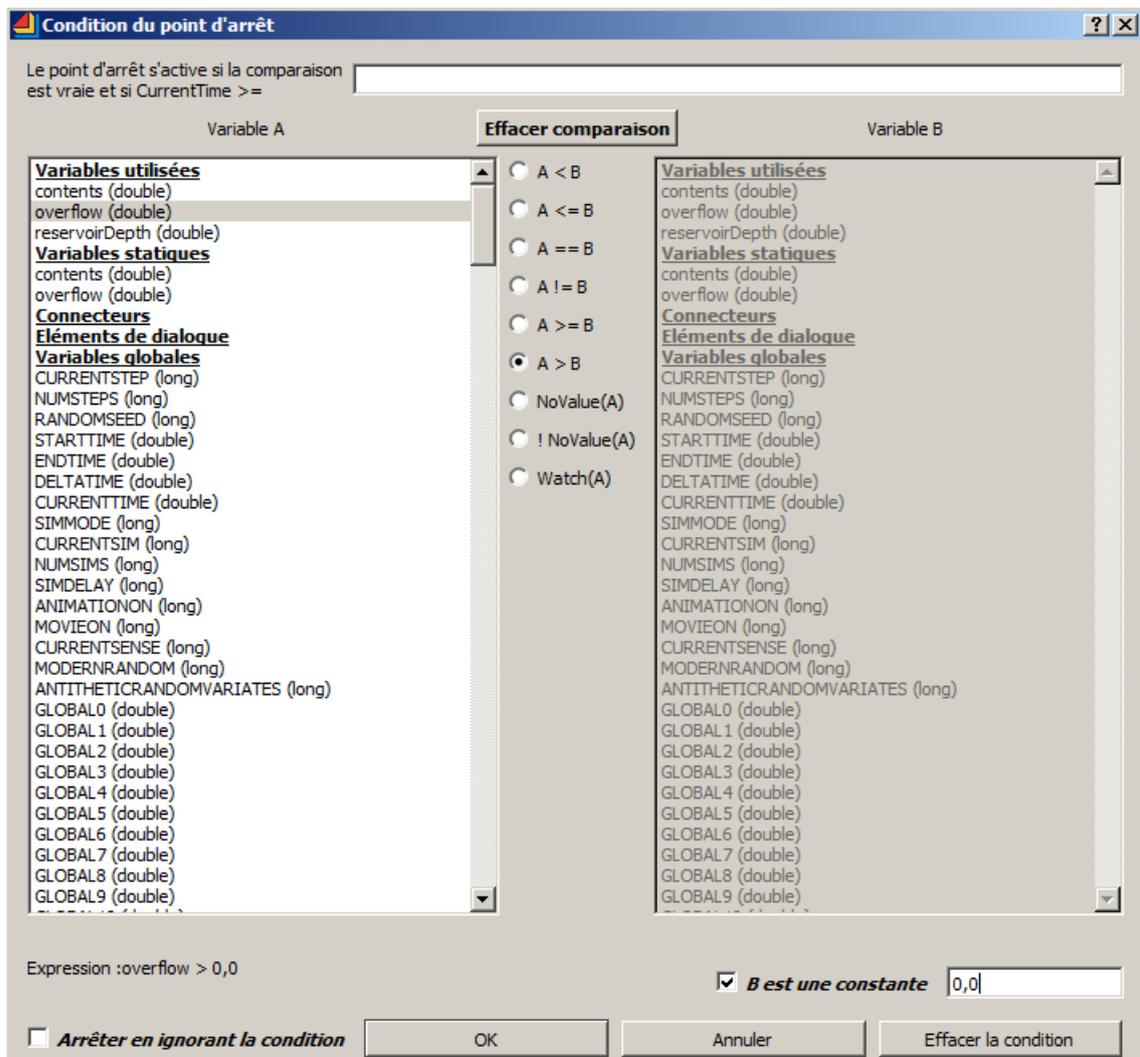
Nous allons définir une condition d'arrêt afin que l'équation ne s'arrête que si le dépassement est supérieur à 0,0 :

Dans la fenêtre de débogage :

- Supprimez les points d'arrêt existants en cliquant sur les ronds rouges correspondants.
 - Ajoutez un nouveau point d'arrêt après les expressions entières « if » et « else » ; dans ce cas au niveau de l'accolade de fin qui est sous l'instruction « overflow = 0,0 ».
 - L'étape suivante consiste à écrire une opération de comparaison qui va comparer la valeur de dépassement par rapport à zéro.

Dans la fenêtre Points d'arrêt :

- Double-cliquez dans la zone à droite du point d'arrêt. Ceci ouvre la fenêtre Points d'arrêt conditionnel, montrée ci-dessous.
 - Dans la colonne Variable A de cette fenêtre, sélectionnez **overflow**.
 - Puisque vous voulez que la condition détermine quand le dépassement est supérieur à zéro, choisissez **A>B** comme comparaison.
 - Cochez tout en bas la case **B est une constante** et entrez 0,0 comme constante.
- Cliquez sur OK



Fenêtre montrant le point d'arrêt conditionnel défini

Remarquez que le rond rouge représentant le point d'arrêt dans la fenêtre de débogage est devenu bleu, pour indiquer qu'une condition existe pour ce point d'arrêt.

Puisque vous avez défini des conditions, et vous voulez trouver à quel moment cette condition est remplie pour la première fois, le calcul doit être recommencé depuis le début. Pour recommencer la simulation :

- Cliquez sur le bouton rouge Stop dans la fenêtre de débogage. Ceci ferme la fenêtre et arrête la simulation.
- Lancez la simulation. La première valeur de dépassement sera affichée dans la portion Variables.

En tant qu'exercice supplémentaire, essayez d'utiliser différentes valeurs pour la condition de dépassement, tel que 2.0, pour voir quand cette valeur apparaît pour la première fois.

Autres fonctions de débogage

La commande **Pause au début** (menu hiérarchique Débogage du menu Simulation) effectue automatiquement une pause de la simulation après l'initialisation (avant la première étape). Cela permet de faire du pas à pas dès de début de la simulation, sans avoir à deviner à quel moment appuyer sur le bouton Pause.

Si l'animation est activée, la commande Pas à pas animation indique à ExtendSim de faire un saut jusqu'au prochain changement de l'animation. Dans les modèles où il se passe beaucoup de choses entre deux modifications de l'animation, cette option permet d'aller beaucoup plus rapidement d'une modification visible à une autre.

La commande Ordre de la simulation

ExtendSim détermine généralement l'ordre d'exécution des blocs en suivant les liens. Si les entrées de

plusieurs blocs sont liées à la sortie d'un seul, ExtendSim considère le premier bloc lié. Si le modèle ne fait pas ce que vous supposiez, vous voudrez sans doute voir explicitement l'ordre d'exécution (menu Modèle).

La commande Rechercher

Lorsque votre modèle est gros, il est parfois difficile de trouver un bloc en un seul coup d'œil. Par exemple, vous avez constaté d'après la Trace que tel bloc ne reçoit pas les données voulues. La commande Edition>Rechercher permet d'entrer un numéro de bloc (ID objet), un label ou un nom : le défilement se fait automatiquement vers ce bloc qui est sélectionné. Il peut également s'agir d'un bloc texte ou autre objet, et non d'un bloc de bibliothèque.

Les blocs et autres objets d'un modèle sont numérotés de 0 à n-1 dans l'ordre où ils sont ajoutés au modèle. Les numéros correspondant à des blocs supprimés sont ré-affectés à de nouveaux blocs.

Deux numérotations existent pour les blocs : le **numéro global**, unique et existant pour tous les blocs ; le **numéro local**, partagé par tous les blocs à l'intérieur d'un bloc hiérarchique. Ces deux numéros figurent dans la barre de titre du dialogue du bloc.

Progression de la simulation

La barre d'outils en haut de l'écran comporte des boutons très utiles en phase de mise au point. Lorsque la simulation tourne, ces boutons sont :



boutons en cours de simulation

Lorsque vous cliquez sur le bouton Pause, la simulation s'interrompt et les boutons deviennent :



boutons en pause

Les options **Pas à pas par bloc** et **Pas à pas animation** du menu Simulation>Outils de débogage indiquent à ExtendSim jusqu'où aller lorsque vous avez demandé la pause puis le Pas à pas. Pas à pas animation place le pas suivant au prochain événement d'animation. Pas à pas par bloc fait que le bouton Pas progresse uniquement vers le bloc suivant.

L'option **Pause au début** (menu hiérarchique Débogage) permet de commencer facilement le pas à pas en début de simulation, après toutes les initialisations.

Entités: les générer et les qualifier

Les entités (en anglais 'item') sont ce qui circule à travers le modèle discret, les propriétés contiennent des informations sur les entités et les valeurs fournissent des informations sur les conditions du modèle. Ce chapitre traite des entités, de leurs propriétés et de la façon dont les informations sont transmises sous forme de valeurs.

La plupart des modèles décrits dans ce chapitre se trouvent dans le dossier Exemples\Discrete Event\Items and Properties. Pour les autres modèles, leur chemin d'accès sera donné au début de la section les concernant.

☞ Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Blocs discrets concernés

Créer et retirer des entités



Create (Item > Routing)

Génère des entités de façon aléatoire, planifiée ou infinie. Génère des valeurs aléatoires ou planifiées. Initialise des entités nouvellement créées avec des propriétés, comme les attributs ou les priorités.



Exit (Item > Routing)

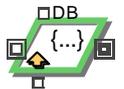
Fait sortir des entités de la simulation. Donne le nombre total d'entités sorties et le nombre d'entités provenant de chaque connecteur d'entrée du bloc.

Propriétés



Get (Item > Properties)

Affiche la valeur des propriétés d'une entité (attribut, priorité, quantité, indice).



Set (Item > Properties)

Donne une propriété (attribut, priorité et quantité) aux entités.



Equation(I) (Item > Properties)

Donne, modifie ou lit des attributs sur des entités. Calcule l'équation lorsque l'entité arrive.



Executive

Son onglet Item Attributes permet de gérer les attributs : les renommer, les supprimer ou savoir où ils sont utilisés dans un modèle. Permet également de définir les équivalents chaîne/valeur pour les attributs chaîne.

Blocs qui tiennent compte des propriétés

La propriété d'une entité peut être un attribut, une priorité ou une quantité. En plus des blocs listés ci-dessus, les blocs suivants (bibliothèque Item) permettent de visualiser, de sélectionner ou de modifier les propriétés des entités existantes ou d'en attribuer de nouvelles.

Activity	Read(I)
Batch	Resource Item
Cost By Item	Select Item Out
History	Shutdown
Information	Throw
Query Equation	Equation(I)

Queue	Unbatch
Queue Equation	Workstation
Queue Matching	Write(I)

Générer des entités

Généralement, les entités requises dans un modèle sont générées par le bloc Create. Une distribution aléatoire ou un planning définissent à quel moment une entité est générée.

Avec une distribution aléatoire, les entités sont générées à intervalle aléatoire ou régulier. La distribution détermine le temps entre deux arrivées d'entité : si l'intervalle est court, des entités arriveront plus souvent.

Lorsque la création des entités est planifiée, celles-ci seront générées à intervalle régulier. Le programme définit *quand* l'entité arrivera et l'intervalle de temps entre deux arrivées ne change pas.

Dans un modèle, une entité peut représenter soit une entité individuelle, soit un groupe d'entités individuelles. Par exemple, 50 entités qui entrent dans un modèle peuvent représenter 50 personnes, ou 50 bus transportant chacun plusieurs personnes. C'est vous qui choisissez la façon dont vous définissez les entités.

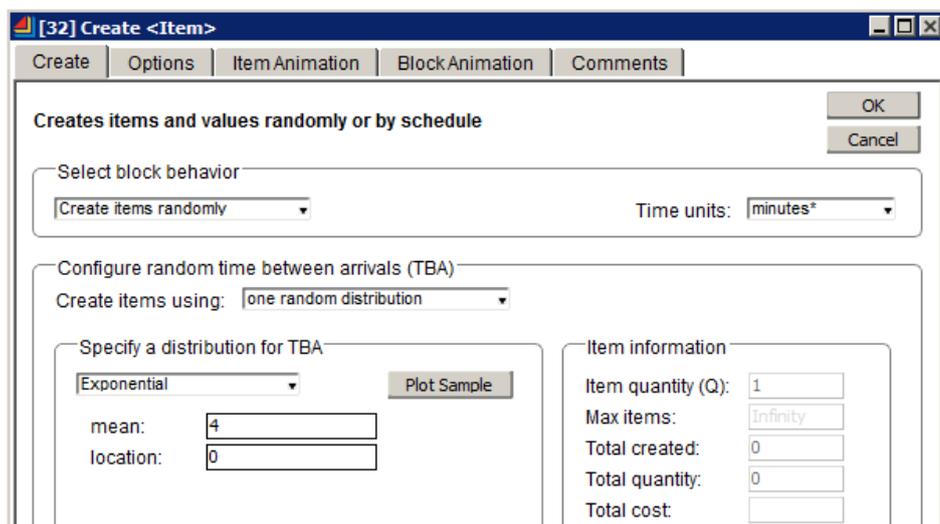
- ☞ Le bloc Create peut aussi générer des entités indéfiniment, afin de disposer à la demande d'un nombre indéfini d'entités : avec par exemple un bloc Gate placé derrière le Create pour le réguler. Attention : ne jamais placer une file d'attente de capacité infinie derrière le Create avec ce comportement.

Générer des entités à intervalles aléatoires

Le bloc Create génère des entités qui arrivent dans le modèle à des heures aléatoires. En sélectionnant l'option *Create items ransomly*, le bloc Create génère des entités à intervalles aléatoires. Les paramètres de la distribution définissent la durée de l'intervalle.

Comme nous l'avons vu précédemment, le modèle de station-service est un exemple d'utilisation du bloc Create pour générer des entités à intervalles aléatoires.

Choisir une distribution dans le bloc Create



Le dialogue du bloc Create comporte plusieurs lois de distribution. Dans le cas d'une distribution empirique, les données peuvent être entrées dans une table. Chaque distribution est décrite dans l'aide du bloc Create.

Dans le bloc Create, lorsque vous spécifiez une distribution, vous indiquez à la fois l'intervalle entre deux arrivées et les caractéristiques de la fréquence d'arrivée.

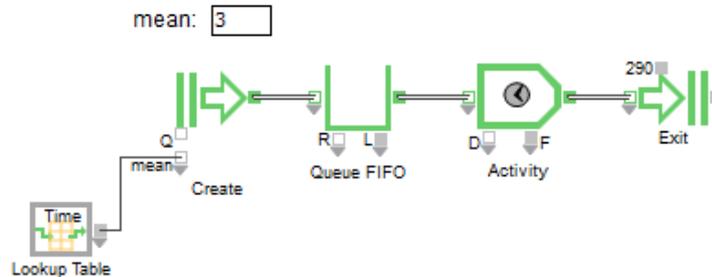
Par exemple, dans le modèle de station de lavage, avec une distribution exponentielle de moyenne 4, les voitures arrivent environ toutes les 4 minutes, pendant toute la durée de la simulation. L'intervalle entre arrivées est donc de 4. Cependant, la distribution exponentielle précise qu'il est plus probable que l'intervalle entre arrivées se situe entre 0 et 4 qu'entre 4 et 8.

Intervalle aléatoire avec paramètres dynamiques

Vous pouvez modifier les paramètres d'une distribution aléatoire au cours de la simulation. Ceux-ci peuvent être fonction du temps ou du statut du modèle. Pour une distribution donnée, les paramètres peuvent être contrôlés de façon dynamique par le connecteur d'entrée continu du bloc Create.

Modèle Random Interval

Dans le modèle Random Intervals, la fréquence d'arrivées dépend du temps. Elle est modélisée en reliant le bloc Lookup Table (bibliothèque Value) au connecteur d'entrée continu du bloc Create. Dans le bloc Lookup Table, une table donne les valeurs moyennes de la distribution normale définie dans le bloc Create. Ainsi, les arrivées d'entités sont basées sur les heures de la journée.



Spécifier les paramètres dynamiques

Comme nous l'avons vu dans le dialogue du bloc Create, les entités arrivent selon une distribution normale. Celle-ci a un paramètre « Moyenne » (Mean). En reliant la sortie du bloc Lookup Table au connecteur d'entrée *mean* du bloc Create, la moyenne de la distribution sera prise dans le bloc Lookup Table pendant la simulation, prévalant ainsi sur toute valeur entrée dans le dialogue. Cela constitue un changement dynamique de l'intervalle entre arrivées moyen.

La distribution détermine l'intervalle entre arrivées. Une valeur moyenne plus petite indique que l'intervalle entre deux arrivées est plus court et que des entités arrivent donc plus souvent.

Specify behavior

Lookup the:

Output is:

Time units:

If input is out of range:

Enter values in the table

	Hour	Mean
0	0	12
1	8	6
2	10	3
3	11	0.8
4	12	3
5	18	6
6	20	3

Lier

Sort Plot

Repeat table every:

Dans le dialogue du bloc Lookup Table, la moyenne est plus petite entre la 10^{ème} heure et la 12^{ème} heure. Les entités arrivent donc plus souvent pendant cette période.

Choisir la bonne unité de temps

Lorsqu'il est paramétré pour regarder le temps, le bloc Lookup Table regarde l'heure qu'il est dans la simulation et émet une valeur correspondante. Sa table permet de déterminer la valeur émise (par défaut la colonne Output 1, ici renommée Mean) à un moment donné de la simulation (par défaut la colonne Time). Le menu déroulant des unités de temps (dans ce modèle : « heures ») représente l'unité de temps pour la valeur figurant dans la colonne Time/Hour. Néanmoins, cela ne contrôle pas le temps auquel obéit la valeur de la colonne Mean. L'unité de temps pour cette colonne est déterminée par le bloc qui reçoit ces valeurs émises, donc le Create.

Lorsqu'une valeur transmise au connecteur d'entrée d'un bloc est utilisée pour définir un paramètre

de temps (comme c'est le cas ici), la valeur envoyée doit être définie dans l'unité de temps attendue par le bloc receveur. Dans ce modèle, les valeurs de la colonne Mean sont envoyées au bloc Create et sont utilisées pour calculer l'intervalle moyen entre deux arrivées. Puisque le bloc Create utilise les minutes comme unité de temps, les valeurs de la colonne Mean dans le bloc Lookup Table sont aussi des minutes. Par exemple, la valeur 6 dans la colonne Mean représente une moyenne de 6 minutes entre deux arrivées, même si dans la colonne Hour, l'unité de temps pour les événements est l'heure.

La moyenne d'une distribution ne peut pas être 0. Le bloc Create vous avertira de cette erreur.

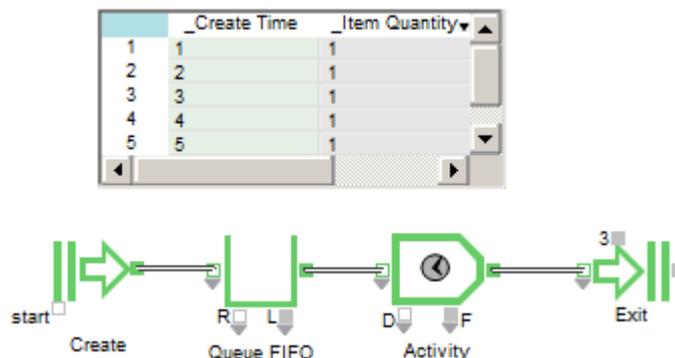
Pour éviter des résultats inattendus, il est important de comprendre ce qui se passe dans le bloc Create lorsque vous changez la moyenne des intervalles entre arrivées dans le temps. Par défaut, le bloc publie un événement, appelé « nextTime », pour la prochaine génération d'entité en fonction des paramètres d'entrée actuels. Lorsque le temps de simulation atteint « nextTime », le bloc Create libère une entité et génère un nouveau nextTime, en fonction des paramètres à cet instant. Pendant la durée entre deux entités libérées, le bloc Create ne réagira pas aux modifications des paramètres d'entrée. Un changement radical des entrées peut donner des résultats inattendus.

Arrivées programmées

Planifier les arrivées d'entités apporte plus de souplesse et de précision que de les générer de façon aléatoire. Vous pouvez paramétrer le comportement du bloc Create. En choisissant l'option *Créer des entités par programme*, les entités arriveront à des intervalles fixes spécifiés dans une table.

Modèle Scheduled Intervals

Par exemple, si l'on veut que 5 entités soient générées, soit une par minute, mais que la simulation dure 10 minutes. En paramétrant le bloc Create pour *Create items by schedule* et en choisissant une distribution constante, une entité sera générée toutes les minutes mais il y aura 10 entités. En planifiant la génération d'entités, c'est vous qui déterminez à quel moment les entités seront créées et combien d'entités entreront dans le modèle.



Dans le modèle d'exemple, le dialogue du bloc Create est paramétré pour planifier la création d'entités. Le planning des heures d'arrivée, qui est cloné sur la feuille de travail, indique qu'une entité sera générée à *_Create Time* (Heure) 1, une autre à l'heure 2, etc. jusqu'à l'heure 5. Avec ce planning, le modèle comporte 5 événements d'arrivée.

Les entités se dirigent vers une file d'attente en attendant d'être traitées par une activité. Pour chaque entité, le temps de traitement est de trois minutes. Etant donné que la simulation dure dix minutes, seules trois entités sortent ; il reste une entité dans la File et une entité en cours de traitement dans l'Activité.

Cette méthode est particulièrement utile lorsque l'intervalle entre deux arrivées est connu mais irrégulier. Par exemple, la première entité pourrait être générée à l'heure 1, la deuxième à l'heure 3 et la troisième à l'heure 3,5.

Notez que la colonne *_Item Quantity* (Quantité) a une valeur par défaut de 1 pour chaque entité générée. Cela signifie que chaque entité générée représente 1 entité.

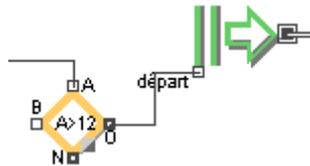
Une autre méthode serait d'utiliser le paramètre *Create items randomly* et de choisir une distribution **constante** de valeur **1** dans le dialogue du bloc Create. Ensuite, dans l'onglet Options, sélectionnez **Maximum items generated : 5**. Cette méthode offre moins de flexibilité/souplesse car il faudrait que l'intervalle entre arrivées soit toujours le même.

Le connecteur Start

Lorsque la création d'entités est planifiée, le bloc Create a un connecteur d'entrée continu *start* qui

permet de contrôler l'exécution du planning. Le programme dépend de si l'entrée *start* est reliée ou pas :

- Si le connecteur *start* n'est pas relié, les temps de création d'entités du planning sont synchronisés avec le temps absolu de la simulation. Par exemple, si le planning indique que la première entité doit être créée à un temps t_2 et que la simulation commence à t_0 , une entité sera créée lorsque le temps de simulation arrive à 2. cependant, si le temps de début spécifié dans le dialogue des paramètres de la simulation est 4, l'entité programmée pour être créée à t_2 ne sera jamais générée.
- Si le connecteur *start* est relié, par exemple à un bloc Decision (bibliothèque Value), les temps de création dépendent de quand le connecteur est activé. Par exemple, dans le cas où le temps de début de simulation est t_0 et que la première entité est programmée pour être créée à t_{11} , si le connecteur *start* est activé à t_5 , alors la première entité sera créée à t_{16} ($5 + 11$).



un bloc Decision qui contrôle un bloc Create

Le connecteur *start* est activé à chaque fois qu'il reçoit un message avec une valeur True (supérieure ou égale à 0,5). L'onglet Options du bloc Create permet de définir le comportement du connecteur *start* :

- *Follow schedule* : c'est l'option par défaut et qui devrait être utilisée dans la plupart des cas. Une fois que le connecteur *start* est activé, tout le planning est exécuté. Tout signal d'activation arrivant avant que le planning en cours n'ait été entièrement exécuté sera ignoré.
- *Generates one item per message (une entité par message)* : il s'agit d'une option avancée pour les situations particulières. En général, elle est utilisée lorsqu'un bloc personnalisé est relié au connecteur *start* et que vous voulez qu'une entité soit générée instantanément à chaque message. Avec cette option, le planning est limité à une ligne. À chaque fois que le connecteur *start* est activé, la ligne est exécutée.
- *Generates one item per event (une entité par événement)* : il s'agit d'une option avancée pour les situations particulières. Dans ce cas, le planning est limité à une seule ligne. À chaque fois que le connecteur *start* est activé, un événement de durée zéro est planifié. Dès que le bloc Create reçoit le message de l'événement durée zéro, il exécutera le planning.

Propriétés

Une propriété est une caractéristique qu'une entité transporte avec elle lors de sa progression dans le modèle. Certaines propriétés peuvent être attribuées aux entités par celui qui construit le modèle ; d'autres sont attribuées automatiquement par le système.

Attributs

Les attributs jouent un rôle important dans les simulations discrètes car ils permettent de différencier les entités. Ils peuvent être employés pour indiquer des temps de traitement (pour les blocs de type Activity) ou pour diriger une entité avant ou après le traitement.

☞ Les attributs de la bibliothèque Rate sont assez similaires.

Chaque attribut consiste en un nom et une valeur :

Le **nom** d'un attribut identifie une caractéristique générale d'une entité. Ainsi, un nom d'attribut peut être « taille », « orientation », « typeVoiture » ou « capacite_reservoir ».

La **valeur** d'un attribut indique une dimension de cette caractéristique. Par exemple, l'attribut « taille » d'une entité peut avoir une valeur de « 8 » ou une valeur de « 12 », alors qu'un attribut dont le nom est « typeVoiture » peut avoir une valeur de « 1 » (pour Peugeot), « 2 » (pour Renault) ou « 3 » (pour Citroën). Une valeur d'attribut n'est pas obligatoirement un chiffre : cela dépend de l'emploi de la valeur. Elle peut aussi être l'adresse de données dans une base de données.

L'intérêt des attributs est leur unicité. Si vous donnez deux attributs du même nom, ExtendSim vous avertira que le nom choisi existe déjà. Les noms d'attribut ne sont pas sensibles à la casse (« Type » est la même chose que « type ») mais les espaces sont à éviter.

Les noms et valeurs d'attribut sont stockés dans deux matrices globales dynamiques.

Le modèle de station essence utilisait des attributs chaîne. Les modèles avec des attributs chaîne utilisent du texte pour représenter la valeur d'attribut correspondante. Cependant, l'architecture sous-

jacente est que les valeurs d'attribut restent des nombres.

Nombre d'attributs dans un modèle

Dans un modèle, chaque entité peut avoir jusqu'à 500 attributs qui la rendent unique. Chaque entité contient l'ensemble des attributs qui ont été définis dans le modèle. L'onglet Item Attributes du bloc Executive affiche tous les attributs du modèle.

Chaque attribut comporte une valeur qui peut représenter un nombre pouvant être utilisé pour l'orientation, le planning, etc. ou l'adresse de données dans une base de données ou une matrice globale. Les données auxquelles on fait référence peuvent contenir un simple chiffre ou une quantité illimitée de données qui décrivent l'entité, son orientation, ses propriétés, etc.

Une utilisation efficace des attributs permet de représenter presque tout. Si vous vous rapprochez de la limite des 500 attributs, pensez à utiliser des attributs DB Address pour référencer des informations dans la base de données ExtendSim.

Types d'attribut

ExtendSim accepte trois types d'attributs :

- Un *attribut numérique* a un nombre réel comme valeur d'attribut.
- La valeur d'un *attribut chaîne* est un nombre mais elle est représentée dans le modèle par une chaîne. Les attributs chaîne permettent d'entrer un label sous forme de texte descriptif (chaîne) pour chaque valeur d'attribut potentielle dans une table de correspondance située dans l'onglet Attributs du bloc Executive. La chaîne peut ensuite être utilisée dans le modèle à la place du nombre correspondant. Par exemple, un attribut chaîne dont le nom est « typeVoiture » a trois valeurs possibles : 1, 2 ou 3. Une fois que la table de correspondance pour cet attribut a été correctement configurée, les blocs dans lesquels l'attribut « typeVoiture » est référencé afficheront les chaînes « Peugeot », « Renault » ou « Citroën » au lieu des nombres 1, 2 et 3.
- La valeur d'un *attribut adresse BDD* (ou DB Address) se compose d'une valeur simple qui représente quatre nombres séparés. Chaque nombre est l'indice d'un élément de la base de données ExtendSim. Pris ensemble, les nombres renvoient à une cellule spécifique dans la base de données. C'est-à-dire que chaque nombre correspond à l'index d'une base de données, d'une table, d'un champ et d'un enregistrement. (Les adresses BDD incomplètes sont autorisées. Par exemple, une entité peut avoir un attribut adresse BDD où seuls l'index de la base de données et de la table sont définis.)

La valeur d'un attribut adresse BDD ne peut pas être utilisée directement. Elle doit être décodée au moyen d'un bloc Get, Read(I) ou Write(I) ou en accédant aux fonctions d'attribut BDD dans un des blocs qui utilisent une équation.

Utiliser des attributs

Le tableau suivant présente quelques techniques de modélisation utilisant des attributs, ainsi que les blocs à utiliser pour les appliquer (bibliothèque Item).

Technique	Blocs à utiliser
Initialiser des entités nouvellement créées avec des attributs.	Create (option « Create items by schedule »)
Définir des attributs par défaut pour les ressources représentées par des entités	Resource Item
Définir ou modifier des valeurs pour les attributs existants	Set, Equation(I), Queue Equation
Vérifier les attributs d'entités existantes	Tous les blocs qui tiennent compte des propriétés.
Orienter des entités selon leurs attributs	Select Item Out (option « Sélectionner la sortie selon l'attribut »)
Classer et libérer des entités d'une file en fonction des attributs	Queue (classement par valeur d'attribut) Queue Matching
Classer des entités selon leur valeur d'attribut et les libérer en fonction du résultat d'une équation	Queue Equation
Regrouper des entités en lots en fonction de leur valeur d'attribut	Batch (option « regrouper les entités en une seule entité »)

Utiliser les valeurs d'attribut pour spécifier un délai ou un temps de traitement	Activity (option « Délai : valeur d'attribut d'une entité »)
Définir la correspondance valeur/chaîne pour les attributs chaîne	Executive (mode « Déclarer des valeurs d'attributs chaîne », onglet Item Attributs)
Trouver le bloc qui utilise un attribut	Executive (mode « Gérer tous les attributs », onglet Attributs)
Gérer des attributs et leur nom (renommer ou supprimer un attribut)	Executive (mode « Gérer tous les attributs », onglet Attributs)
Calculer le temps de cycle d'une entité	Donnez un attribut correspondant au temps courant dans un bloc Set ou utilisez la fonction <i>Attribut de timing</i> dans l'onglet Options du bloc Create.

Ajouter des attributs à un modèle

Les blocs discrets concernés par les attributs comportent un menu déroulant qui permet de sélectionner des propriétés existantes ou d'en ajouter de nouvelles.

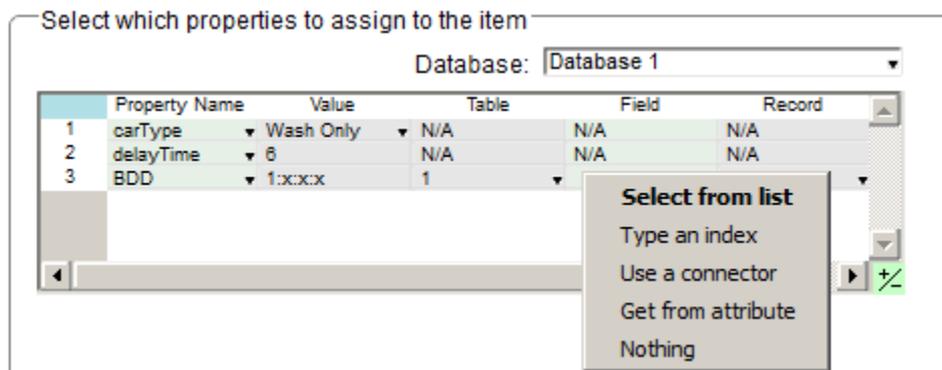
☞ Selon le bloc, le menu déroulant propose une liste différente de types d'attribut. Par exemple, le bloc Set permet de créer des attributs numériques, chaînes ou adresse BDD.

Lorsqu'un nouvel attribut est créé, il apparaît dans la liste des propriétés dans le dialogue du bloc et rend l'attribut accessible par tous les blocs qui en tiennent compte dans le modèle.

Les informations suivantes expliquent comment créer différents types d'attributs dans la colonne *Property* du tableau figurant dans le dialogue du bloc Set. Les options soulignées sont celles qui permettent de créer un nouvel attribut :

- Pour créer un nouvel attribut numérique, cliquez sur le menu déroulant, puis sélectionnez **New Value Attribute**. Entrez le nom du nouvel attribut dans le dialogue qui s'affiche et cliquez sur OK. Par exemple, le nom d'un attribut numérique peut être « taille ».
- Pour définir un nouvel attribut chaîne, sélectionnez **New String Attribute** dans le menu déroulant de la colonne Nom. Entrez un nom et cliquez sur OK. L'onglet Attributs du bloc Executive s'ouvre automatiquement, affichant la table dans laquelle sont précisées les valeurs d'attribut (et les chaînes correspondantes) pour les attributs chaînes. Par exemple, un attribut chaîne peut être « type de voiture » et les combinaisons chaîne/valeur correspondantes peuvent être Peugeot/1, Renault/2 et Citroën/3.
- La création d'un nouvel attribut adresse BDD requiert une base de données ExtendSim existante. Dans le dialogue du bloc Set, sélectionnez une base de données ExtendSim dans la liste qui s'affiche. Pour créer l'attribut adresse BDD, cliquez sur le menu déroulant et sélectionnez **New DB Address Attribute**. Entrez le nom du nouvel attribut dans le dialogue qui apparaît puis cliquez sur OK. Par exemple, le nom d'un attribut adresse BDD peut être « bdd ».

Avec les trois différents types d'attributs, le dialogue du bloc Set ressemble à ceci :



Une fois que les attributs sont créés, ils doivent être affectés aux entités présentes dans le modèle et des valeurs uniques doivent leur être attribuées.

Sélectionner des attributs et les attacher aux entités

Pour qu'un attribut puisse être utilisé, il faut le définir et lui attribuer une valeur. Pour cela, vous pouvez utiliser le bloc Set ou Create.

La méthode la plus simple pour affecter un attribut à une entité est de sélectionner un attribut dans le dialogue du bloc Set, puis de faire passer l'entité par ce bloc. La valeur de l'attribut peut être définie dans le dialogue du bloc Set ou par son connecteur d'entrée continu.

Un autre bloc fréquemment utilisé est le bloc Create en mode « Programme ». Cette méthode est pratique pour initialiser de nouvelles entités avec un ensemble d'attributs au moment où elles entrent dans le modèle. Dans la table de planification du bloc Create, vous pouvez sélectionner un attribut dans le menu déroulant d'une des colonnes, et entrer une valeur pour cet attribut. Une entité générée aux heures spécifiées aura le nom et la valeur d'attribut indiqués dans le tableau. D'autres blocs, comme le bloc Resource Item, peuvent aussi être utilisés pour donner des attributs aux entités.

Attribut numérique

Pour donner un attribut numérique, sélectionner un attribut dans le menu déroulant de la colonne Property. (Les attributs numériques existants apparaissent dans la liste.) Entrez un nombre dans la colonne Value.

Attribut chaîne

Pour donner un attribut chaîne, sélectionner un attribut chaîne et cliquez sur la cellule dans la colonne Value pour faire apparaître un menu avec toutes les chaînes définies pour cet attribut. Dans l'exemple de droite, l'attribut TypeVoiture a été sélectionné et le menu Value contient les chaînes Peugeot, Renault et Citroën. Si vous sélectionnez Renault, la valeur correspondante affectée à l'entité pour l'attribut Type de voiture sera le chiffre 2.

Si vous reliez un bloc Random Number (bibliothèque Value) utilisant un tableau Empirique à un bloc Set affectant un attribut chaîne, une liste avec les chaînes de cet attribut s'affichera dans la colonne Value de la table empirique.

Attribut adresse BDD

Chaque bloc Set ne fait référence qu'à une seule base de données ExtendSim, qui devient un élément de l'adresse BDD. Pour affecter un attribut DB Address dans le dialogue du bloc Set, choisissez d'abord une base de données dans le menu déroulant, et l'attribut DB Address.

Une fois que l'attribut adresse BDD est sélectionné ou créé, le tableau s'agrandit et affiche les autres éléments de l'adresse (Table, Champ et Enregistrement). Pour définir la valeur de l'attribut adresse BDD, cliquez sur les menus déroulants appropriés dans le tableau, choisissez si les informations doivent être prises dans une liste, entrées sous forme d'index ou obtenues depuis un connecteur. La figure ci-dessous montre l'attribut DB Address « ProcessTime », dont la valeur provient du dernier enregistrement, dans le champ « Times » de la table « Process time », dans la base de données « Process ».

Database: Process

	Property Name	Value	Table	Field	Record
1	ProcessTime	1:1:1:2	Process time	Times	2 : 5.04

Il n'est pas nécessaire de sélectionner tous les éléments pour un attribut adresse BDD. Par exemple, vous pouvez spécifier les indices de base de données, de table, de champ et ignorer l'indice d'enregistrement.

Pour un attribut DB Address, la colonne Valeur affiche l'adresse de la base de données, déterminée par les indices des paramètres dans les colonnes Table, Champ et Enregistrement. Dans la figure ci-dessus, la valeur est 1:2:2:1. Ce sont une suite d'index : 1 est l'index de la base de données, 2 est l'indice de table, 2 est l'indice de champ « précipitations » et -1 est la marque du dernier l'enregistrement.

Propriété	Valeur	Table
type voiture	Peug...	
	Peugeot	
	Renault	
	Citroën	

Une fois que l'attribut est défini, les informations sur l'attribut indiquées dans le dialogue du bloc Set seront affectées à chaque entité au moment où elles entrent dans le bloc. Les valeurs d'attribut peuvent aussi être définies de façon dynamique avec les connecteurs d'entrée continus du bloc, qui prévalent sur des valeurs déterminées dans le dialogue.

Chaque modèle dispose d'une liste interne de tous les noms des attributs donnés aux entités dans le modèle. Cependant, chaque nom d'attribut ne sera pas utilisé par toutes les entités. Pour qu'une entité utilise un nom d'attribut, la valeur de l'attribut doit être définie de manière explicite au moyen d'un bloc permettant de modifier les attributs (par exemple le bloc Set).

Lire les valeurs d'attribut et repérer les changements

Pour manipuler une entité qui a un attribut (l'orienter, la traiter, etc.), il faut lire la valeur d'attribut de l'entité. Pour cela, le plus simple est de sélectionner le nom de l'attribut dans le menu déroulant Attribut d'un bloc qui lit les attributs (bloc Activity ou Get).

Blocs Activity et Workstation

Dans le dialogue d'un bloc Activity ou Workstation, vous pouvez spécifier que la valeur d'attribut d'une entité doit être utilisée comme temps de traitement de l'entité.

Bloc Get

Lorsque des entités passent par ce bloc, il va chercher les informations concernant les attributs spécifiés dans son dialogue. Ensuite, il reporte ces informations dans la table et sur ses connecteurs de sortie continus. Les informations et l'endroit où elles sont reportées dépendent du type d'attribut :

- *Attributs numériques.* La valeur de l'attribut est affichée dans la colonne Valeur du tableau de l'attribut et envoyée sur le connecteur de sortie continu correspondant à l'attribut.
- *Attributs chaîne.* La chaîne est affichée dans la colonne Value du tableau de l'attribut et le nombre qui correspond à cette chaîne est envoyé sur le connecteur de sortie continu approprié.
 - ☞ Note : Si vous reliez un bloc Lookup Table (bibliothèque Value) avec une correspondance sur la valeur **d'entrée**, à un bloc Get qui fait référence à un attribut chaîne, une liste des chaînes de cet attribut s'affichera dans la colonne de gauche du bloc Lookup Table.
- *Attributs DB Address.* Vous pouvez obtenir soit un élément d'une adresse de BDD, soit l'adresse entière. C'est pourquoi vous devez spécifier l'élément auquel le bloc Get doit faire référence. Pour cela, dans le menu déroulant de la colonne « DB attrib = » du tableau, sélectionnez lequel des 5 éléments sera observé (index bdd, index table, index champ, index enr. ou adresse bdd). L'information sera reportée dans la colonne Value et sur le connecteur de sortie continu utilisé pour cet attribut.

Pour prendre en compte les 5 éléments d'un attribut DB Address, ajoutez cinq lignes au tableau. Pour chaque ligne, le même attribut DB Address doit apparaître dans la colonne Name (nom de propriété), mais des éléments différents doivent être sélectionnés dans la colonne « DB attrib reports ». Cela est utile lorsque le bloc Get travaille conjointement avec les blocs Read ou Write (bibliothèque Value). Cela permet de varier le point de lecture ou d'écriture en fonction des informations qui circulent sur l'entité.

En plus des sorties continues pour reporter la valeur d'un attribut, le bloc Get possède un connecteur Δ (delta) qui permet de signaler lorsque la valeur d'un attribut change. Le connecteur Δ émet 1 lorsque la valeur d'attribut d'une entité (pour le premier attribut spécifié dans le dialogue) est différente de la valeur d'attribut de l'entité précédente. Sinon, il émet 0. Cela permet de détecter l'arrivée d'un nouveau type d'entité ou un changement de la valeur d'attribut utilisée pour le temps de traitement.

Modifier les valeurs d'attribut

La meilleure manière de modifier la valeur de l'attribut d'une entité est d'utiliser le bloc Equation(I) (bibliothèque Item). Ce bloc peut consulter des informations sur les propriétés d'une entité et les modifier en appliquant une formule mathématique, et utiliser le résultat comme nouvelle valeur d'attribut de l'entité. Par exemple, si une entité arrive avec une valeur de 5 pour l'attribut « prochain enregistrement », vous pouvez ajouter 1 à la valeur 5 et ainsi créer une nouvelle valeur d'attribut de 6 pour l'attribut « prochain enregistrement » de cette entité.

Pour modifier les propriétés, vous pouvez également établir un lien entre la sortie continue du bloc Get et un bloc Math ou Equation (bibliothèque Value). Le bloc applique une formule mathématique puis envoie les résultats vers l'entrée continue d'un bloc Set.

Matrices des attributs

Les noms et valeurs d'attribut sont stockés dans deux matrices globales dynamiques :

- La *matrice Names* est une matrice unidimensionnelle qui stocke le nom de chaque attribut utilisé dans le modèle. Les noms d'attribut peuvent comporter jusqu'à 15 caractères et ne sont pas sensibles à la casse.

- La *matrice Values* est une matrice bidimensionnelle qui stocke la valeur de chaque attribut pour chaque entité sous forme de nombres réels.

Lorsque de nouveaux noms d'attribut sont ajoutés au modèle, de nouvelles cellules (éléments de matrice) sont ajoutées à la matrice Names et de nouvelles colonnes sont ajoutées à la matrice Values. Jusqu'à 500 colonnes peuvent être ajoutées.

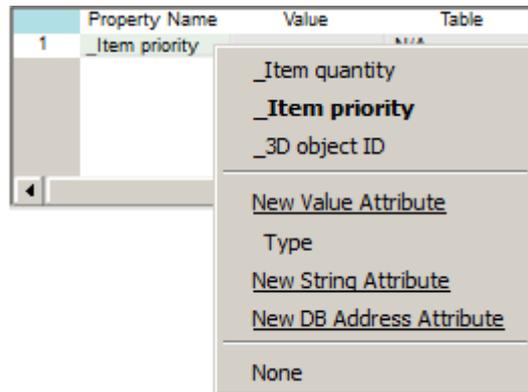
À mesure que des entités sont créées pendant la simulation, de nouvelles lignes sont ajoutées à la matrice Values. Le nombre de lignes dans la matrice Values est illimité et restera égal au nombre d'entités dans le modèle.

Notez que pour chaque attribut nommé dans le modèle, une cellule est réservée dans la matrice Values pour chaque entité. Néanmoins, tous les attributs ne sont pas utilisés par toutes les entités. Pour qu'une entité puisse utiliser un attribut, il faut donner une valeur à cet attribut avec un bloc qui manipule les attributs (par exemple le bloc Set). Si aucune valeur n'est affectée à l'attribut, celui-ci ne sera pas utilisé par l'entité.

Priorités

Comme les attributs, une priorité est un type de propriété qui peut être affectée à une entité. La priorité permet de spécifier l'importance d'une entité. Vous pouvez donner des priorités aux entités puis les manipuler en fonction de ces priorités.

Les priorités sont particulièrement utiles pour examiner un ensemble d'entités en attente et déterminer l'ordre dans lequel elles seront traitées. Par exemple, à une étape de la production, il se peut qu'un employé examine tous les travaux en attente et choisisse le plus urgent.



Sélectionnez *_Item priority* (Priorité) dans le menu des propriétés. Les entités ne peuvent avoir qu'une seule priorité. Pour avoir plusieurs niveaux de priorité, utilisez plutôt les valeurs d'attribut. Si une priorité est donnée à une entité qui en possède déjà une, c'est la nouvelle priorité qui prédomine. Lorsque des entités sont regroupées en lots, c'est la plus haute priorité qui prédomine.

La plus petite valeur (y compris les valeurs négatives) donne la plus haute priorité.

Donner, lire et utiliser des priorités

Le tableau suivant présente quelques-unes des techniques de modélisation basées sur les priorités ainsi que les blocs à utiliser pour les appliquer (bibliothèque Item).

Technique	Bloc(s) à utiliser
Initialiser des entités nouvellement créées avec des priorités.	Create (option « Create items by schedule » = selon un planning »)
Définir des priorités par défaut pour les ressources (d'entités)	Resource Item
Donner, modifier ou vérifier des priorités sur des entités existantes	Set, Get, Equation(I)
Sélectionner des entités entrantes selon leur priorité.	Select Item In (option « Select item based on priority »)
Classer et libérer des entités selon leur priorité	Queue (classement par priorité)
Classer des entités selon leur priorité et	Queue, Equation

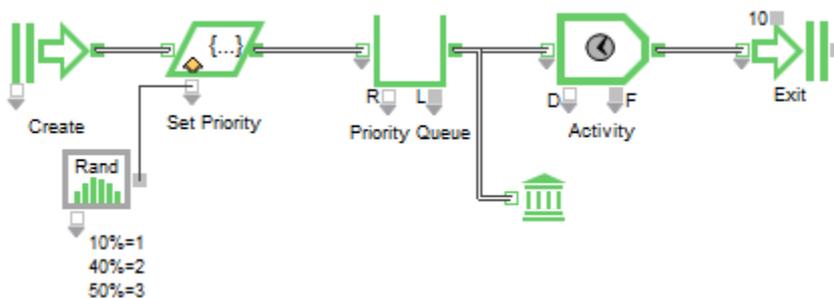
les libérer en fonction du résultat d'une équation	
Visualiser la priorité d'une entité	Get, History
Allouer des ressources en priorité à l'entité ayant le plus haut rang	Resource Pool

☞ Le bloc Select Item Out n'affecte et n'utilise pas de priorités mais son connecteur de sortie peut être rendu prioritaire pour permettre à une entité d'être orientée vers le premier connecteur disponible ayant la priorité la plus haute. Le bloc Select Item Out privilégie le *chemin* emprunté par une entité plutôt que l'entité elle-même.

Modèle Priorities

Dans l'exemple, un bloc Random Number (bib. Value) envoie des valeurs à un bloc Set de la façon suivante :

- 10 % du temps, il émet 1
- 40 % du temps, il émet 2
- 50 % du temps, il émet 3



Dans le dialogue du bloc Set, le tableau indique qu'une priorité sera affectée aux entités entrantes. En reliant la sortie du bloc Random Number au connecteur d'entrée continu *ItemPriority* du bloc Set, les priorités seront données en fonction des valeurs issues du bloc Random Number. Puisque le plus petit nombre représente la plus haute priorité, 10 % du temps les entités recevront la plus haute priorité.

Le bloc Queue est paramétré de façon à classer les entités par priorité. Cela signifie que les entités présentes dans la file qui ont la plus haute priorité seront envoyées vers le bloc Activity avant les autres. Un bloc History, ajouté au modèle en faisant un clic-droit sur le connecteur de sortie de la File, montre que seules les entités ayant la plus haute priorité sont traitées ; l'Activité ne parvient pas à faire face à la demande.

La section « [Opérations interrompues](#) » (chapitre sur les activités) explique comment utiliser les valeurs de priorité pour déterminer si une entité doit être prioritaire par rapport à une autre.

Quantités

La quantité est une autre sorte de propriété pouvant être attribuée à une entité. Chaque entité peut être une entité individuelle ou un groupe d'entités similaires. Comme pour les priorités, une entité ne peut avoir qu'une seule quantité à la fois ; la quantité par défaut est 1. Si la quantité d'une entité est 1, elle représente une seule entité. Si la quantité est un autre nombre, elle représente un groupe. Généralement, les quantités sont définies dans les blocs Create et Set.

La quantité d'une entité peut être n'importe quel nombre. Si la quantité est 0 ou moins, l'entité disparaît dès qu'elle atteint la file.

Dans la plupart des cas, l'entité garde sa quantité par défaut, 1. Cependant, pour modéliser un changement d'équipe au cours duquel cinq travailleurs quittent leur poste en même temps, pour simuler la réception d'un paquet de 300 lettres ou d'autres situations semblables, définissez la quantité de l'entité avec une valeur autre que 1.

Traitement des entités ayant une quantité différente de 1

Les entités ayant une quantité différente de 1 sont considérées différemment selon la nature du bloc qui les traite. Elles circulent ensemble, forment une unité et sont traitées comme une seule entité jusqu'à ce qu'elles atteignent un bloc Exit, Queue, Batch ou Resource Item ou jusqu'à ce qu'elles soient envoyées sur un connecteur mixte (Demand ou Start.)

- Si une entité avec une quantité autre que 1 atteint un bloc Exit, Queue, Batch ou Resource Item, elle est décomposée en entités identiques séparées. Par exemple, lorsqu'elle entre dans une file d'attente, une entité ayant une quantité de 10 sera transformée en 10 entités distinctes. Chacune aura alors une quantité de 1 et les mêmes propriétés (attributs, priorité, etc.) que l'entité d'origine. Une entité avec une quantité de 0 disparaît lorsqu'elle atteint une file.
- Si des entités ayant une quantité autre que 1 sont envoyées vers un connecteur mixte (comme *demand*), elles seront traitées comme une seule entité mais la quantité de l'entité peut être utilisée par le bloc comme information de contrôle.

Tous les autres blocs traitent les entités ayant une quantité différente de 1 comme une entité individuelle, en ignorant la quantité qui lui est associée.

Le bloc Activity considère une entité avec une quantité supérieure à 1 comme une entité individuelle et la traite comme une seule entité. Pour que les entités soient traitées séparément, placez une file en amont du bloc Activity. En effet, les files d'attente décomposent les entités qui ont une quantité supérieure à 1.

Attribuer une quantité à une entité

Une quantité est affectée à une entité dans les blocs Create, Set et Equation(I).

Bloc Set

Dans le tableau du dialogue du bloc Set, sélectionnez **_Item quantity** dans la colonne **Property Name** et entrez la quantité dans la colonne Value ou envoyez une valeur dans le connecteur d'entrée continu **_Item Quantity** du bloc. Cette quantité sera alors affectée à chaque entité qui passera par le bloc Set.

Bloc Create

Par défaut, le bloc Create est paramétré de sorte qu'à chaque événement, une entité entre dans le modèle ; c'est la situation de modélisation la plus courante. Le comportement du bloc Create détermine comment spécifier que plusieurs entités sont libérées plutôt qu'une seule :

- Mode *Create items randomly*. Dans les Options, remplacez **Item quantity (Q)** par un nombre entier différent de 1. Sinon, envoyez une valeur sur le connecteur d'entrée continu *ItemQuantity (Q)* du bloc Create.
- Mode *Create items by schedule*. Entrez des valeurs dans la colonne **Quantity** du tableau pour chaque ligne **_Create Time** ayant des heures d'arrivée.

Par exemple, si vous voulez montrer qu'une entité arrive de façon aléatoire toutes les 4 minutes environ, vous devez utiliser les mêmes paramètres que dans le modèle de Station essence vu précédemment. Dans le bloc Create, choisissez une distribution exponentielle et entrez **Mean : 4**. Dans l'onglet Options, laissez **Item quantity (Q) : 1**. Pour montrer que 2 voitures/entités arrivent toutes les 4 minutes, conservez les mêmes paramètres (distribution Exponentielle, Moyenne : 4) mais entrez **Item quantité (Q) : 2**. Le bloc générera une entité de quantité 2 toutes les 4 minutes environ.

Comme nous l'avons vu dans la section précédente, le bloc situé en aval du bloc Create détermine la façon dont est traitée une entité avec une quantité avec une quantité supérieure à 1. Par exemple, si une entité avec une quantité de 2 va directement dans un bloc Queue ou Resource Item, elle sera divisée en 2 entités ayant chacune une quantité de 1. Cependant, si l'entité va directement dans un bloc Activity, elle sera traitée comme une entité individuelle ayant une quantité de 2. Dans la plupart des cas, le bloc Create sera suivi d'une Queue qui décomposera l'entité en deux entités distinctes.

Souvent, vous ne voudrez générer qu'une seule entité à chaque événement. Par exemple, plutôt que de générer 2 entités toutes les 4 minutes, vous ne voudrez générer qu'une entité toutes les 2 minutes. En effet, à moins qu'elles ne se trouvent dans un conteneur quelconque, il est rare que deux entités arrivent exactement au même moment ; il est plus probable qu'elles arrivent à des moments différents.

Modèle Quantities

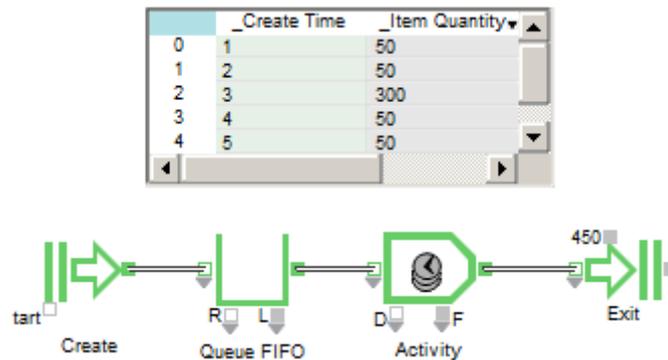
Par exemple, vous recevez 500 commandes par semaines dont l'essentiel arrive le mercredi. Dans ce cas, il y a cinq événements (une arrivée par jour du lundi au vendredi), chacun avec une quantité de 50 ou 300.

Ce modèle Quantities est similaire au modèle Scheduled intervals mais chaque entité générée par le bloc Create a une quantité supérieure à 1 et l'activité traite 5 entités à la fois.

Le bloc Create est en mode **Create items by schedule**. Les heures d'arrivée (**_Create Time**) et le nombre d'entités qui arrivent à l'heure programmée (**_Item Quantity**) sont entrés dans la table, qui a

été clonée sur la feuille de travail.

La table indique que 300 entités arrivent le troisième jour (mercredi) et que 50 entités arrivent les autres jours.



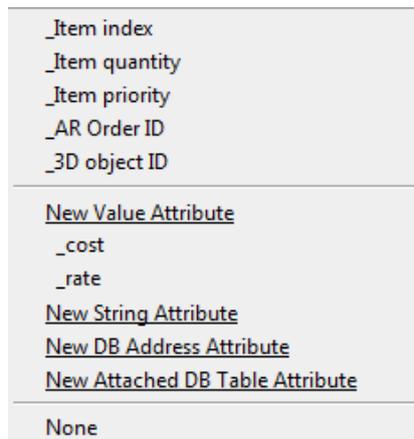
Le bloc Create génère une entité avec une quantité supérieure à 1 comme s'il s'agissait d'un groupe d'entités arrivant toutes au même moment. Lorsque l'entité arrive dans un bloc Queue, elle se démultiplie. Dans cet exemple, en passant du bloc Create au bloc Queue, chaque entité se transformera en 50 ou 300 entités, en fonction de sa quantité.

Si vous exécutez le modèle, vous verrez que le bloc Create crée 5 entités mais que ce sont 500 entités qui sont arrivées dans la file.

Autres propriétés

Outre les propriétés que nous venons de voir, telles que la quantité d'une entité ou les attributs définis par l'utilisateur, ExtendSim peut affecter des propriétés aux entités. Comme le montre le menu déroulant ci-contre, ces propriétés données par le système sont précédées du signe « _ » et sont les suivantes :

- **_Item index** (index de l'entité). Cette propriété est disponible dans les blocs Get, Equation(I) et History et indique l'endroit où se situe l'entité dans la matrice d'entités stockée dans le bloc Executive. Elle est utilisée en débogage par les programmeurs.
- **_AR Order ID**. Cette propriété n'est utilisable que si le modèle comporte un bloc Resource Manager (bib. Item). Voyez le bloc pour plus de détails.
- **_3D object ID** (ID de l'objet 3D). Quand vous sélectionnez un objet d'animation 3D pour représenter cette propriété stocke les indices des objets. Actuellement non utilisable.
- **_Cost** ou **_Rate** (coût ou taux). Si des coûts doivent apparaître quelque part dans le modèle, ExtendSim ajoutera les attributs _Cost et _Rate aux menus déroulants des propriétés.
- **_Animation**. Si vous sélectionnez un objet d'animation 2D pour représenter une entité, cette propriété stocke l'indice de l'objet.



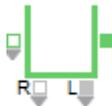
Files d'attente

Une file d'attente stocke les entités en attente de traitement. Les files peuvent avoir un comportement simple, comme le stockage des entités selon un ordre FIFO (premier entré, premier sorti) ou plus complexe, libérant les entités en fonction de leurs attributs. Vous pouvez également définir une option dans le dialogue du bloc Queue afin de spécifier combien de temps une entité attendra avant qu'elle n'abandonne ou ne quitte la file prématurément.

Les exemples se trouvent dans le dossier Exemples\Item Event\Queueing.

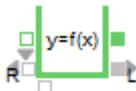
☞ Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Les blocs File d'attente



Queue (Item)

Stocke des entités jusqu'à ce qu'il y ait de la disponibilité en aval. Peut retenir les entités selon un ordre FIFO (premier entré, premier sorti), LIFO (dernier entré, premier sorti) ou selon leur priorité ou valeur d'attribut. Utilisé comme pool de ressources, le bloc Queue retient les entités selon un ordre FIFO.



Queue Equation (Item)

Stocke des entités. Calcule une équation au moment où il reçoit une entité où lorsqu'on une valeur lui est envoyée. Dès qu'il y a de la disponibilité en aval, il libère les entités en fonction du résultat de l'équation.



Queue Matching (Item)

Contient plusieurs files pour retenir les entités dans des groupes séparés. Libère un groupe d'entités quand une machine est disponible en aval et que toutes les conditions requises pour ce groupe sont remplies. Permet d'associer différentes entités.



Queue Tools (Utilitaires > Outils discrets)

Lorsque ce bloc est relié au connecteur de sortie L (Longueur) d'une file, il visualise et initialise le contenu de la file. Affiche une table d'informations sur les propriétés des entités. Peut ajouter à une file un nombre initial d'entités ayant des propriétés spécifiques.

Algorithmes de file d'attente

ExtendSim propose plusieurs algorithmes de file d'attente dans les blocs Queue.

FIFO : file classée selon un ordre « premier entré, premier sorti ».

LIFO : file classée selon un ordre « dernier entré, dernier sorti ».

Priorité : le bloc Queue peut lire des priorités et libérer en premier les entités qui ont la plus haute priorité (la plus petite valeur). Pour cela, une priorité doit avoir été attribuée aux entités. Si ce n'est pas le cas, elles reçoivent une priorité par défaut et sont reléguées à la fin de la file d'attente.

Attribut : le bloc Queue peut utiliser des valeurs d'attribut pour classer des entités dans la file. De plus, le bloc Queue permet de définir des algorithmes personnalisés basés sur les attributs donnés aux entités. Il regroupe les entités selon certains attributs et les libère par groupe une fois que toutes les conditions sont remplies. Pour ce type de classement, il faut donner des attributs aux entités avant qu'elles n'entrent dans la file. Les entités qui n'ont pas d'attribut reçoivent un attribut par défaut avec une valeur Blank et sont orientées vers la fin de la file d'attente.

Décision : le bloc Queue Equation utilise une équation pour déterminer la règle de classement dans la file (par exemple le Plus petit résidu, le Moindre changement ou le Niveau de service maximum.)

Note : Sauf pour une file FIFO, il doit y avoir plusieurs entités dans une file pour que les algorithmes de file d'attente puissent fonctionner correctement et modifier l'ordre des entités. Par exemple, si

vous paramétrez un bloc Queue de façon à classer les entités selon leur priorité, et qu'il n'y a qu'une seule entité à la fois dans la file, la règle de classement n'aura aucun effet.

Attente d'un service

Les entités attendent dans une file avant d'être traitées par un ou plusieurs services. Pour représenter ces systèmes, on utilise les blocs suivants :

Le bloc *Create* génère des entités à des intervalles de temps exponentiels par exemple.

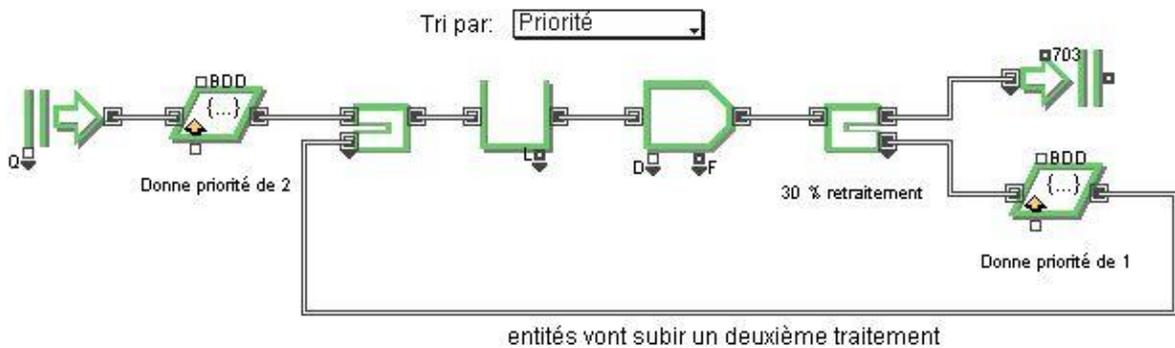
Le bloc *Queue*, paramétré pour classer les entités par ordre FIFO, LIFO ou autre, retient les entités et les libère dans l'ordre voulu. La longueur de la file peut être spécifiée dans le dialogue du bloc.

Le bloc *Activity* représente les services : vous pouvez spécifier un temps de service selon une distribution exponentielle ou autre dans le dialogue du bloc.

Files selon priorité

Lorsqu'il est paramétré selon la règle de classement de la priorité, le bloc Queue libère l'entité qui a la priorité la plus haute. Pour cela, il faut attribuer une priorité aux entités avant qu'elles n'entrent dans la file. Les entités sans priorité reçoivent une priorité *Blank* ou vide par défaut et sont envoyées à la fin de la file d'attente.

Modèle Priority



Dans l'exemple de Priority, les entités entrent dans le modèle et reçoivent immédiatement une priorité de valeur 2. Ensuite, elles entrent dans un bloc Queue paramétré pour un **Sort by : priority**. Après l'usinage, chaque entité est inspectée pour détecter d'éventuels défauts. Si l'entité comporte un défaut, une nouvelle priorité de valeur 1 lui est attribuée et elle est renvoyée vers le bloc Queue en attendant de subir un deuxième usinage. Dès que la machine peut recevoir une nouvelle entité, le bloc Queue libère l'entité qui a la plus haute priorité. Dans ce cas, toutes les entités en attente d'un second usinage seront libérées en premier. Lancer le modèle avec l'animation pour regarder les entités ayant une priorité de 1 (billes rouges) doubler celles qui ont une priorité de 2 (billes vertes) dans la file d'attente.

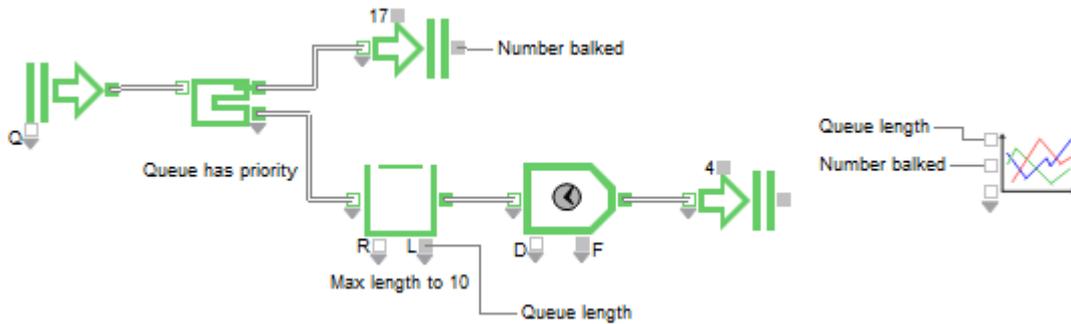
Blocage

Une situation de blocage arrive lorsqu'une entité a été traitée mais qu'elle ne peut pas quitter le bloc parce que l'opération suivante ne peut l'accueillir ou qu'il y a saturation. C'est une situation que l'on rencontre souvent dans des traitements en série, où plusieurs postes se suivent, ou bien lorsque des charges de travail ont été mal dimensionnées.

Des exemples présentent des situations de blocage dans les sections « Traitement en série », et « Classement séquentiel ».

Dérivation

Parfois, un client entre dans un magasin, observe qu'il y a une longue file d'attente, et s'en va. On parle alors de **dérivation**. En fonction de certaines conditions (file trop longue ou qui avance trop lentement), le bloc Select Item Out dirige l'entité hors de la simulation avant qu'elle n'entre dans la file.



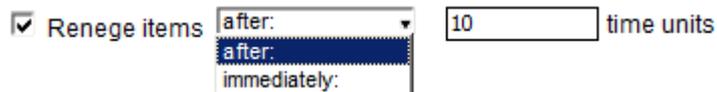
Modèle Balking

Dans le modèle Balking illustrant le phénomène de dérivation, le bloc Select Item Out dirige en priorité les entités vers la file. Si la longueur maxi de la file est atteint le seuil de 10 défini dans le dialogue du bloc, le bloc Select Item Out fera passer l'entité par son connecteur du haut pour la diriger vers le bloc Exit.

Abandon

On parle d'abandon lorsqu'une entité, en attente dans une file, quitte la file sans attendre son tour. Par exemple, une personne qui appelle au téléphone est mise en attente trop longtemps et raccroche.

Pour simuler une situation d'abandon, sélectionnez une des options suivantes dans l'onglet Options du bloc Queue :

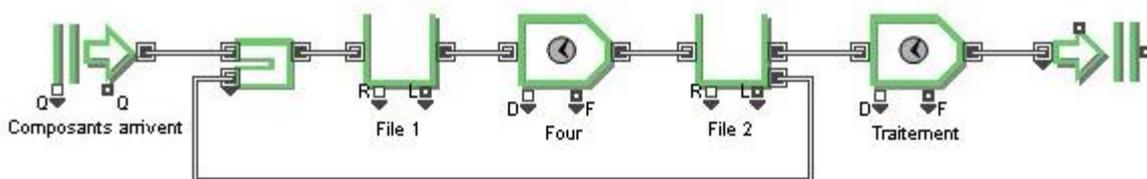


- **Renege items after...** Abandon après un certain nombre d'unités de temps. Le nombre d'unités de temps peut être défini dans le dialogue du bloc ou par son connecteur R.
- **Renege items immediately** Abandon immédiat quand le connecteur R reçoit une valeur True (supérieure ou égale à 0.5).

Lorsque l'une de ces options est sélectionnée, un second connecteur de sortie apparaît sur la droite du bloc Queue. Les entités qui abandonnent quittent la file par cette sortie RenegeItemOut. Elles peuvent être soit réorientées vers la file du début (comme le montre l'exemple ci-dessous), soit dirigées vers un autre endroit du modèle, soit sortir du modèle.

Modèle Reneging

Dans le modèle d'abandon, des composants attendent dans une première file avant de passer dans un four. Elles attendent ensuite dans une seconde file avant d'être traitées. Si une entité attend trop longtemps avant et refroidit avant d'être traitée, elle est redirigée vers la première file pour être réchauffée dans le four.



Dans l'onglet Options de la deuxième file, il est spécifié qu'un composant attendra 5 minutes avant d'être réchauffé. Ces informations ont été clonées sur la feuille de travail. Comme nous le voyons dans l'onglet Results, le bloc Queue compte et indique automatiquement le nombre d'entités qui ont abandonné.

Changement de file

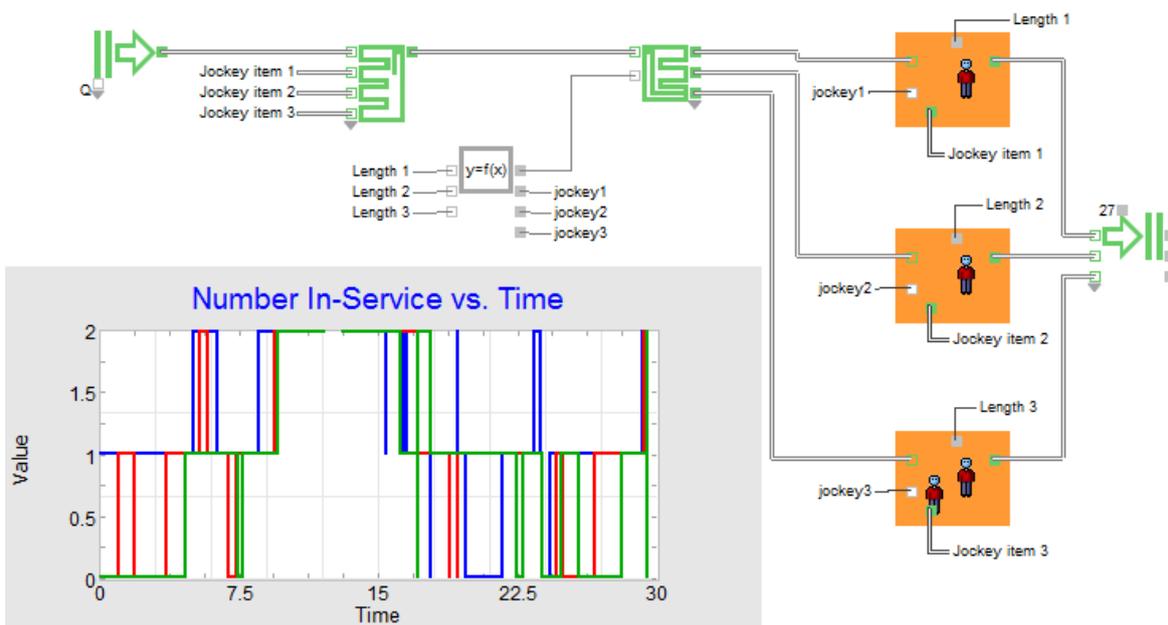
Une situation dite de *Jockeying* ou changement de file intervient lorsque des entités changent de file d'attente dans le but d'obtenir un avantage. Par exemple, dans un supermarché, il n'est pas rare de voir une personne quitter une file d'attente qui avance trop lentement et se diriger vers une autre file.

L'option Renege (onglet Options du bloc Queue) permet de construire des modèles représentant ce type de comportement. Normalement, les entités abandonnent si elles ont passé trop de temps dans une file d'attente. Le bloc Queue dispose d'un connecteur qui peut contraindre la dernière entité de la

file à l'abandon. Le modèle Jockey.mox illustre bien cette situation.

Modèle Jockey

Dans ce modèle, les clients arrivent du bloc Create et passent par le bloc Select Item Out qui les oriente vers la file la moins longue parmi trois files d'attente. Les files peuvent avancer à des vitesses différentes. Le dernier client de chaque file peut changer de file si une file moins longue est disponible.



Classer des entités avec le bloc Queue Equation

Le bloc Queue Equation utilise une équation pour déterminer la position de chaque entité dans la file. Pour déterminer le classement des entités, à chaque fois qu'il y a un changement éventuel dans le classement, l'équation est évaluée pour chaque entité présente dans la file. Ensuite, ce calcul est effectué à chaque fois qu'une entité entre dans la file ou la quitte, ou lorsque le connecteur d'entrée continu du bloc reçoit une nouvelle valeur.

Plusieurs variables peuvent être employées pour déterminer le tri dans la file ; pour attribuer un rang à chaque entité, on utilise la variable « _Queue rank » (rang).

Pour que ce bloc fonctionne correctement, au moins une variable de sortie de « Rang » doit être définie.

Une fois qu'un rang a été attribué à chaque entité, celles-ci sont classées dans la file selon la règle de classement sélectionnée dans l'onglet Options du bloc. La structure de données interne du bloc garde en mémoire la position attribuée à chaque entité et les entités ayant le rang le plus élevé sont placées en tête de file. Note : Si une valeur vide a été attribuée à une entité, celle-ci ne quittera jamais la file.

Bien qu'une seule variable Rang ne soit requise, vous pouvez aussi définir d'autres variables de Rang (variables « secondaires »). Elles sont utiles si vous devez départager des entités ayant le même rang. Dans l'onglet Options, le paramètre +/- permet également de définir les critères donnant lieu à une égalité. Le bloc Queue Equation permet de calculer des règles de classement par plus petit résidu, moindre changement, niveau de service maximum et autres situations complexes.

Variables et règles de classement

Le bloc Queue Equation propose plusieurs types de variables d'entrée et de sortie ainsi que quelques règles de classement prédéterminées.

Variables d'entrée

Plusieurs variables d'entrée peuvent être utilisées pour déterminer le classement :

Variables d'entrée	Utilisation
_Attribute	Un attribut de l'entité évaluée
_Last item to exit	Donne la valeur d'attribut choisie de la dernière entité à quitter le bloc
_Item Quantity	Quantité de l'entité évaluée
_Item Priority	Priorité de l'entité évaluée
_Item Index	Valeur de l'index de l'entité
3D Object ID	Identité de l'objet utilisée pour représenter l'entité dans la fenêtre 3D
DB Value	Renvoie à une valeur dans une base de données ExtendSim
DB Address	Adresse d'une position spécifique dans une base de données ExtendSim
DB Index	Index d'une base de données, d'une table, d'un champ ou d'un enregistrement ExtendSim
Static Variables	Variables statiques, qui gardent leur valeur entre deux calculs d'équation.
Arrival Time	Heure d'arrivée de l'entité dans la file
Best Result	Le meilleur (plus haut ou plus bas) résultat de l'équation
Connector	Un nombre variable de connecteurs d'entrée continus sont disponibles

Variables de sortie

Une fois que le bloc a déterminé la prochaine entité à être libérée, plusieurs variables de sortie peuvent être calculées pour cette entité :

Variables de sortie	Utilisation
Attribute	Stocke les informations de l'attribut de l'entité libérée.
_Item Priority	Change la priorité de l'entité libérée
3D Object ID	Change l'identité 3D de l'entité libérée
DB Value	Entre des informations dans la base de données ExtendSim
_Result Rank	Définit le rang (la position) de chaque entité dans la file. Au moins une des variables de sortie doit être de ce type.
Connector	Un nombre variable de connecteurs de sortie continus est disponible
Select con	Conçu pour être relié au connecteur Select du bloc Select Item Out, ce connecteur oriente l'entité libérée.

Comme il a été évoqué précédemment, vous pouvez sélectionner une ou plusieurs variables de sortie mais au minimum une d'entre elles doit être du type « Rang ». Vous pouvez aussi avoir plusieurs variables de classement. La variable de classement « secondaire » sera utilisée pour départager des entités ayant le même rang.

Note : ces variables ne sont disponibles que pour l'entité libérée.

Règles de classement

L'onglet Options du bloc Queue Equation propose les options suivantes pour déterminer les entités qui doivent être libérées en premier :

- Entité ayant le rang le plus haut
- Entité ayant le rang le plus bas
- Entité ayant la première valeur VRAI

La valeur du rang de chaque entité est calculée à partir de l'équation utilisant les variables d'entrée. Les entités ayant le rang le plus élevé seront libérées en premier.

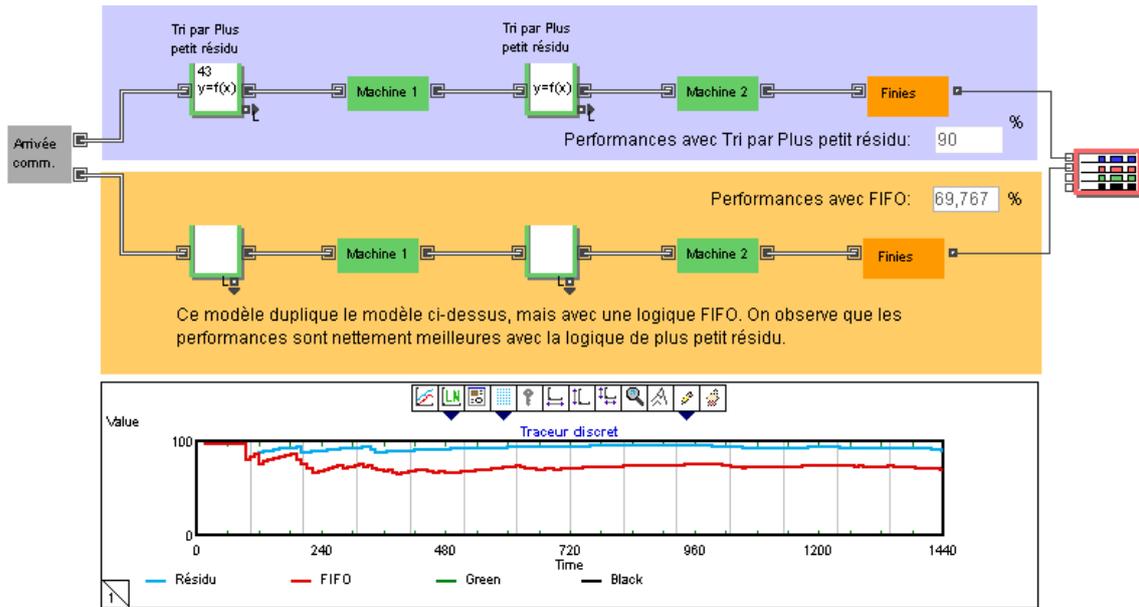
Plus petit résidu

Une règle de classement par plus petit résidu utilise le temps de traitement restant et la date de fin prévue pour l'entité pour classer les entités dans la file. Le résidu est la date de fin moins le temps

restant. Autrement dit, c'est ce qu'il manque pour que l'entité soit terminée. Utiliser cette information pour classer les entités donne priorité à celles qui vont être en retard. Dans un modèle représentant des commandes de biens et services, un classement par plus petit résidu tend à réduire le nombre de commandes en retard.

Modèle Least Dynamic Slack

Le modèle Least Dynamic Slack montre l'amélioration sur le respect des délais que l'on peut obtenir en classant les entités de cette manière plutôt que dans l'ordre FIFO. Les deux modèles sont identiques mais le premier utilise des blocs Queue Equation pour calculer le plus petit résidu, alors que le deuxième utilise des blocs Queue qui classe les entités dans un ordre FIFO. Dans le modèle, les équations des blocs Queue Equation calcule le résidu pour chaque entité. L'entité qui a le plus petit résidu (qui sera la première à être en retard) sera sélectionnée en premier. Comme le montre le graphique cloné sur la feuille de travail, les délais sont mieux respectés lorsqu'on utilise un classement par plus petit résidu (courbe du haut) plutôt que dans un ordre FIFO (courbe du bas).

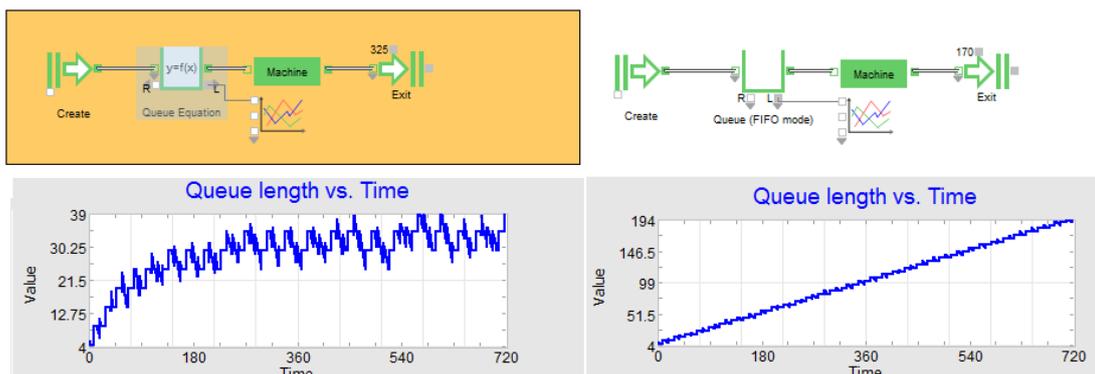


Moindre changement

Dans certains systèmes, le temps de changement, c'est-à-dire les nettoyages, réglages ou changements d'outils en passant d'une pièce à une autre, ajoutent des durées importantes au temps de cycle. Dans un tel cas, il est parfois utile de traiter toutes les pièces d'un même type tant qu'il en figure dans la file, avant de changer de type. Cela réduit le nombre de changements entre différents types de produits. Pour cela, on utilise une règle de classement par moindre changement.

Modèle Minimize Setup

Le modèle Minimize Setup compare l'attribut « Produit » de chaque entité dans le bloc Queue Equation à l'attribut « Produit » de l'entité qui a terminé son traitement. La première entité de même Produit est libérée. Si aucune entité n'a la même valeur pour l'attribut « Produit », la première entité dans la file est libérée. Le graphique monte les effets de cette règle de classement : la file commence par se remplir jusqu'à ce qu'elle puisse constituer assez de lots d'entités du même type pour tirer un avantage de la réduction des temps de changement.



Niveau de service maximum

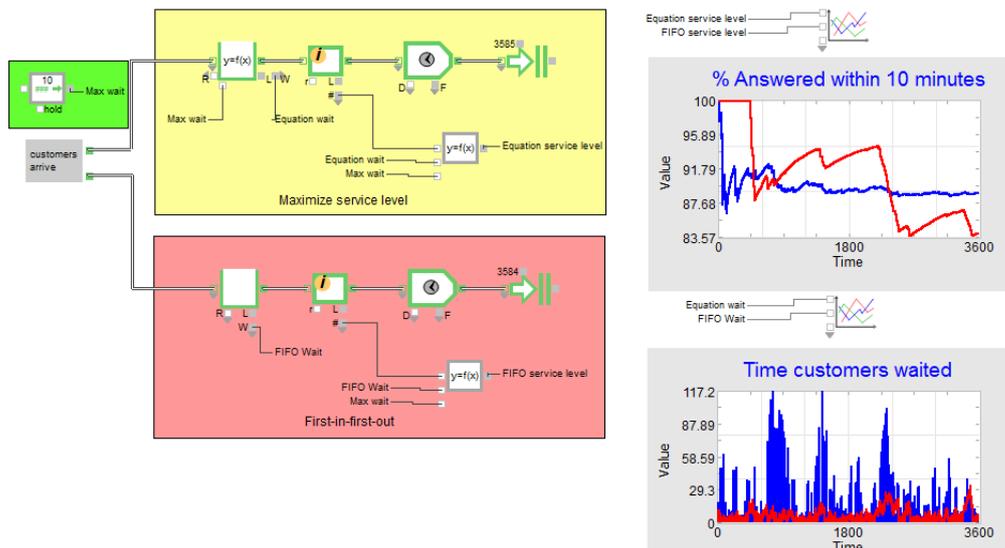
Dans un système tertiaire, le niveau de service peut être défini comme le nombre de clients servis pour une période donnée. Par exemple, les personnes peuvent être évaluées en fonction du pourcentage de demandes client satisfaites pour cette période. Pour maximiser le niveau de service, une file qui applique la règle du niveau de service maximum donne la priorité aux clients ayant attendu moins longtemps, ce qui entraîne une très nette amélioration du niveau de service. Cependant, dans une situation réelle, ce type de système peut causer des mécontentements chez certains clients qui peuvent attendre très longtemps avant d'être servis alors que d'autres clients arrivés plus tard attendront moins longtemps.

Modèle Maximize Service Level

Dans ce modèle, la partie du haut utilise un bloc Queue Equation pour classer la file en deux niveaux de priorité.

- La priorité est donnée au client qui a attendu le plus longtemps (mais moins de 10 minutes).
- Les clients ayant attendu plus de 10 minutes sont placés en queue de file.

La partie du bas est identique à celle du haut mais utilise un bloc Queue avec un classement de type FIFO. Pendant la simulation, deux graphiques montrent les effets sur le niveau de service et le temps d'attente. En comparant les deux modèles, vous pouvez voir que l'utilisation de cette règle de classement entraîne une forte amélioration. Néanmoins, le graphique du bas montre qu'avec la règle du niveau de service maximum, certains clients attendent beaucoup plus longtemps que d'autres.



Combinaisons complexes

Le bloc Queue Equation permet de modéliser des situations complexes. La première règle de classement permet de départager des entités égales. La seconde règle permet de déterminer l'entité qui a la plus haute priorité.

Modèle Combined Rule

Ce bloc utilise le plus petit résidu comme première règle de classement. Cependant, si le plus petit résidu est inférieur à deux unités de temps, le rang (ordre des entités dans la file) est utilisé.

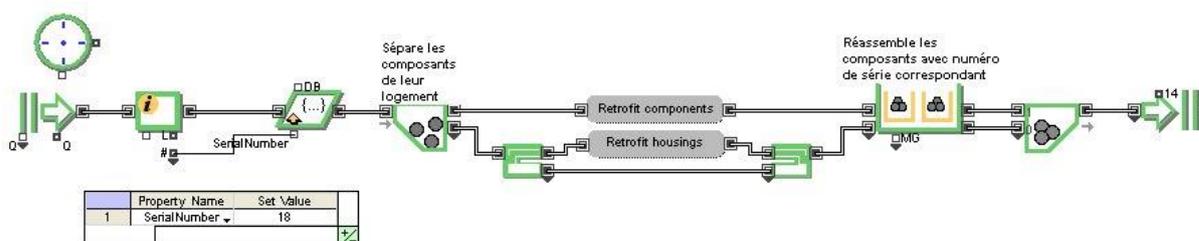
Queue Matching

Le bloc Queue Matching possède un nombre variable de connecteurs représentant chacun une file interne distincte. Dans chaque file, le bloc classe les entités en différents groupes selon leur valeur d'attribut. Les entités ne sont libérées que lorsque le nombre requis d'entités est atteint dans chaque groupe et dans chaque file. Ce bloc est particulièrement utile pour vérifier que les entités ont une caractéristique donnée à un moment donné. Par exemple, cette file permet d'assembler des composants dans le bon ordre ou de s'assurer que des sous-ensembles sont compatibles entre eux.

Modèle Queue Matching

Dans l'exemple suivant, des systèmes électroniques arrivent, sont démontés pour nettoyage et vérification de leurs composants, puis sont réassemblés. Dans cette opérations, tous les composants mais seulement 40 % des logements doivent être examinés. De plus, il est important que les

composants retrouvent leur logement d'origine.



Le bloc Information compte les systèmes électroniques qui arrivent et donne le nombre total de systèmes qui sont passés par ce bloc. Le bloc Set utilise la valeur comptée comme un « numéro de série » affecté à chaque système. Après vérification, le système est recomposé.

Autres techniques

Cette section explique comment visualiser, initialiser et animer le contenu d'une file. Ces techniques permettent de créer un modèle plus précis et de déboguer et/ou valider un modèle.

Visualiser et manipuler le contenu d'une file

Une des fonctionnalités avancées d'ExtendSim est la possibilité de manipuler le contenu d'une file. Le bloc *Queue Tools* (bibliothèque Utilities) permet de visualiser les entités présentes dans une file, de modifier le classement de ces entités et d'initialiser le contenu de la file. Pour cela, reliez le connecteur d'entrée continu du bloc *Queue Tools* au connecteur de sortie L (longueur) d'un bloc Queue ou Queue Equation. Le bloc *Queue Tools* possède deux onglets, **View** et **Options**.

Onglet View

Cet onglet permet de manipuler des entités dans un bloc Queue ou Queue Equation et d'afficher des informations concernant ces entités. En lançant la simulation, chaque entité de la file correspondante aura une entrée dans la table.

Les menus déroulant situés en haut des colonnes permettent de choisir l'information que vous souhaitez visualiser parmi les différentes propriétés de l'entité (attribut, quantité, priorité, etc.). Des boutons permettent de changer la place d'une entité dans la file ou de supprimer une entité. Par exemple, la figure ci-contre montre que les entités ont repris leur numérotation à 1 après l'entité 15, car certaines entités sorties par un bloc Exit ont rendu libres ces numéros.

	timing	Item quantity	Item index	Aucun
1	6,66666666667	1	11	
2	7,33333333333	1	12	
3	8	1	13	
4	20	1	14	
5	20,6666666667	1	15	
6	21,3333333333	1	1	
7	40	1	2	
8	40,6666666667	1	3	
9	41,3333333333	1	4	
18				

Lier

Note: les entités de quantité >1 utiliseront une ligne dans la table

Pour manipuler les entités dans une file, lancez la simulation, mettez-la en pause au moment voulu, sélectionnez une entité et utilisez les boutons du dialogue pour la déplacer ou la supprimer. (Pour mettre la simulation en pause, cliquez sur une cellule quelconque dans la table Queue Tools ou sur le bouton Pause dans la barre d'outils ExtendSim.) Attention, si vous laissez le dialogue ouvert ou que vous avez placé un clone sur la feuille de travail, la simulation est nettement ralentie.

L'onglet View du bloc *Queue Tools* est cloné sur la feuille de travail du modèle Initializing and Viewing, commenté dans la section suivante.

Initialiser une file

Il est souvent utile d'introduire des entités dans le modèle en début de simulation. L'onglet Options du bloc Queue Tools permet de remplir une file d'attente pour les besoins suivants :

- Réduire les erreurs dues à la montée en régime, en faisant que la simulation commence en état quasi stable.
- Prendre en compte un état courant connu dans un modèle. par exemple, dans un modèle d'ordonnement, il est important de démarrer la simulation dans l'état de charge réel.

L'onglet Options comporte trois options :

No Queue initialization (File non initialisée) : pas d'entités dans la file en début de simulation.

Initialize with... (File initialisée) : la file est initialisée avec le nombre d'entités et les valeurs de propriété spécifiées.

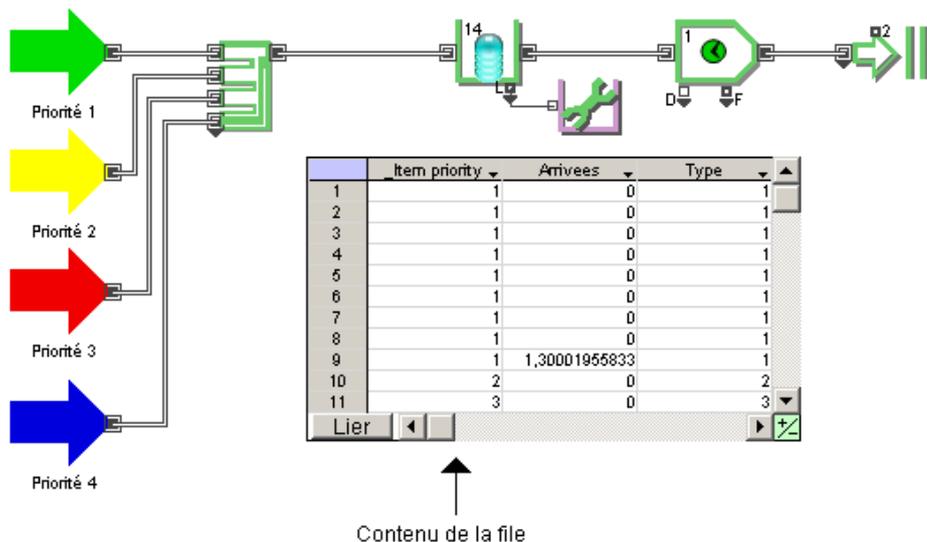
Initialize from global array (Initialisation depuis une matrice globale) : le nombre d'entités et les valeurs de propriété sont lues dans une matrice globale. Cette méthode est très utile pour importer le contenu d'une file depuis une source externe car les matrices globales peuvent elles-mêmes rechercher leurs données dans les fichiers de bases de données, Excel ou Internet.

Modèle Initializing and Viewing

Dans cet exemple, l'onglet Options du bloc Queue Tools est utilisé pour introduire des entités dans un bloc Queue au début de la simulation. L'onglet View a été cloné sur la feuille de travail.

Dans ce modèle, les entités sont ajoutées à la file au temps 0. Comme le montre la table Propriétés (onglet Options), les entités d'origine ont une priorité de 1, une quantité de 1, une valeur de 1 pour l'attribut Type et une valeur de 0 pour l'attribut Arrivée.

L'onglet indique également que les *entités* sont représentées par une bille bleue dans l'animation.



Animer le contenu d'une file

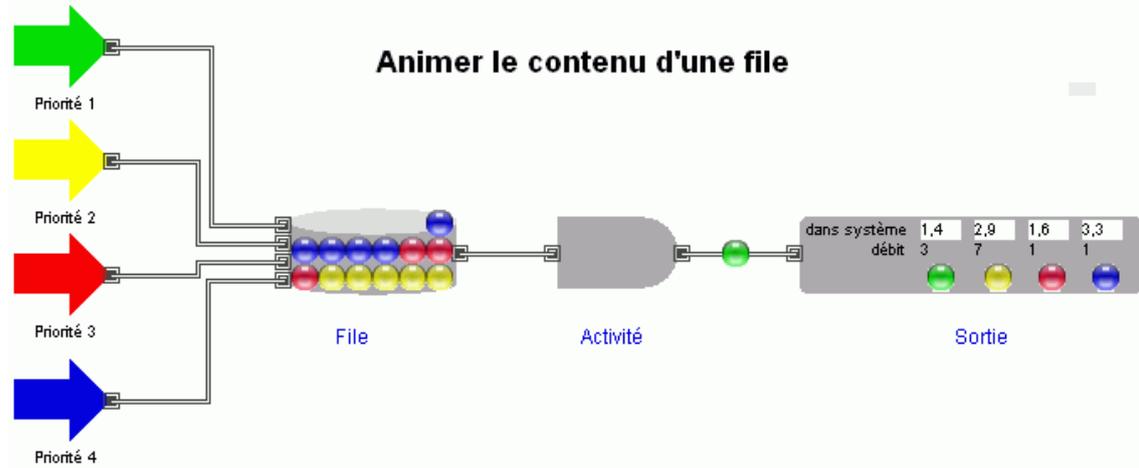
Par défaut, le bloc Queue affiche le nombre d'entités qu'il contient. Cependant, il est possible d'obtenir une animation plus détaillée de chaque entité en plaçant le bloc Queue à l'intérieur d'un bloc hiérarchique et en animant l'icône du bloc hiérarchique. Pour cela :

- Placez une Queue à l'intérieur d'un bloc hiérarchique (faites un clic droit sur le bloc Queue et sélectionnez **Hiérarchiser la sélection**.)
- Ouvrez la structure du bloc hiérarchique (faites un clic droit sur le bloc hiérarchique et sélectionnez **Ouvrir structure de bloc H**) et ajoutez des symboles d'animation de taille identique à partir des outils d'animation de la barre d'outils.
- Ouvrez l'onglet *Item Animation* du bloc Queue et entrez les numéros du premier et du dernier symbole d'animation dans les champs **Animate H bloc objects x to y**.

Modèle Animating Queue Contents

Cet exemple montre l'animation d'une file priorité. Comme le montre les flèches de couleur situées à gauche, il y a quatre priorités possibles. Les billes vertes représentent des entités avec une priorité de 1, les billes jaunes indiquent les entités ayant une priorité de 2, etc.

Les entités ayant la plus petite valeur de priorité (priorité la plus haute) seront placées en début de file. L'animation montre que les entités qui entrent dans la file et qui ont une plus petite valeur de priorité doublent les autres entités.

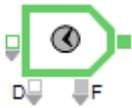


Dans ce modèle, il y a quatre lignes et six symboles d'animation par ligne sur l'icône du bloc hiérarchique situé à droite des flèches. Dans l'onglet Item Animation du bloc Queue situé à l'intérieur de ce bloc hiérarchique, l'option **Animate H bloc objects 1 to 24** est sélectionnée. Ainsi, les 24 symboles seront animés sur l'icône du bloc hiérarchique en fonction de ce qui se passe dans la file.

Activités et opérations

Les activités ou opérations sont des actions portant sur des ressources ou des groupes de ressources, telles que le traitement, la transformation ou le déplacement. Les activités impliquent un **délai** qui est le temps pris pour effectuer l'action. Tous les blocs d'opérations de ExtendSim régulent le flux d'éléments par ces opérations. Les opérations retiennent une entité (ou plusieurs) jusqu'à ce qu'elle soit demandée par le bloc suivant.

Les blocs liés aux activités



Activity (Item)

Contient une ou plusieurs entités et les transmet en fonction de leur heure d'arrivée et du délai subi.



Convey Item (Item)

Transporte des entités d'un point à un autre en fonction d'informations de vitesse et distance, avec des capacités d'accumulation.



Shutdown (Item)

Génère des arrêts pour les blocs Activity et Valve.



Workstation (Item)

Représente une station de travail pouvant faire attendre et traiter des entités. Dans l'onglet Behavior, vous définissez combien d'entités peuvent être traitées à la fois, et combien de temps dure le traitement.

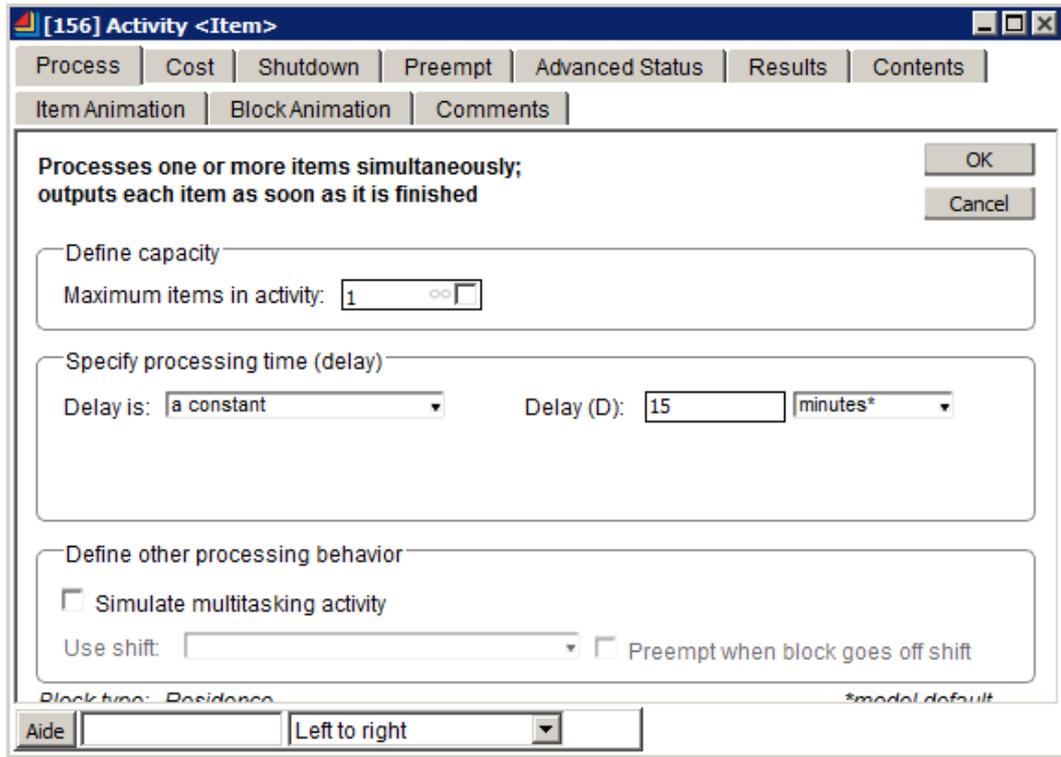


Transport (Item)

Transporte des entités d'un point à un autre en fonction d'informations de vitesse et distance.

☞ Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Voici le dialogue typique du bloc Activity, le bloc d'opération le plus courant :



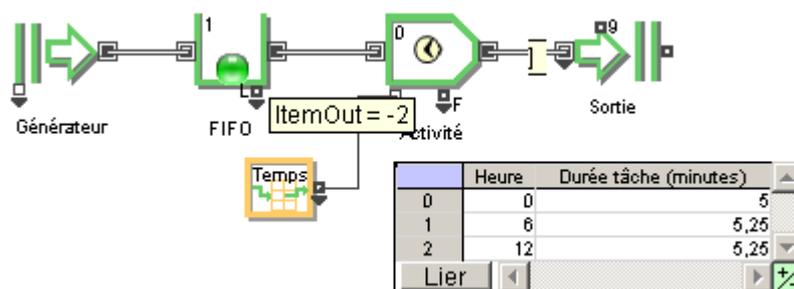
Dialogue d'une activité

Déterminer le temps de traitement

Les opérations impliquent une certaine durée de traitement. Ce délai peut être spécifié dans le dialogue s'il s'agit d'un délai fixe et connu. Si le délai est mal connu ou variable, vous pouvez l'indiquer par le connecteur D, ou utiliser des attributs pour spécifier un délai.

Temps de traitement programmé

Si le temps de traitement évolue selon des plages horaires connues, vous pouvez utiliser un planning horaire. C'est le cas lorsque vous simulez la productivité d'un opérateur, qui diminue avec la fatigue. Par exemple, le temps de traitement est de 5 minutes, mais après 6 heures de travail, il passe à 5,25 minutes. Pour cela, liez un bloc Lookup Table au connecteur D d'un bloc Activity, comme ci-dessous :

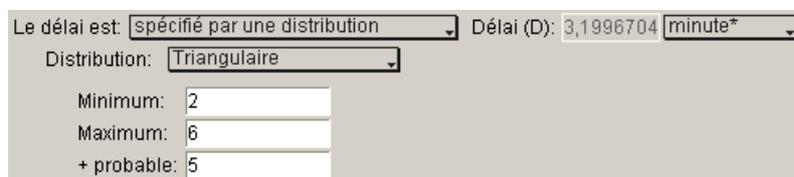


Temps de traitement programmé : modèle Scheduled Time

Le temps pris pour effectuer la tâche est spécifié par des plages horaires dans le bloc Lookup Table.

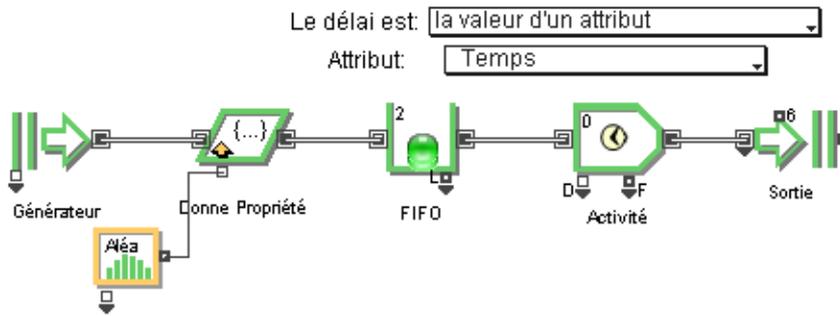
Temps de traitement aléatoire

Il est très courant qu'on ne connaisse qu'un temps moyen pour effectuer une tâche. Il faut alors choisir une distribution aléatoire, par exemple dans le modèle Random Activity :



Temps de traitement personnalisé

Des attributs peuvent être employés pour connaître le temps de traitement de telle entité spécifique. Dans le cas le plus simple, vous donnez une valeur 0 à l'attribut pour signifier que l'entité ne doit pas être traitée par cette machine, et autre valeur qui sera lue comme un temps de traitement. Exemple :



Temps de traitement personnalisé : modèle Custom Time

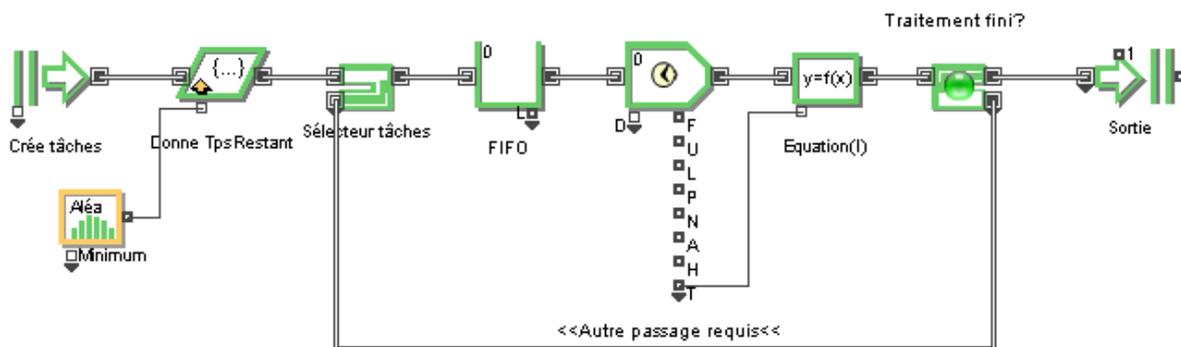
Le bloc Set sert à affecter un attribut. Les entités contrôlées reçoivent une valeur de 5 pour cet attribut, celles qui ne sont pas contrôlées reçoivent une valeur de 0. Cette valeur est donnée par le bloc Random Number dans lequel on a choisi une distribution empirique où 25% des entités ont une valeur 0 et 75% ont une valeur 5.

Toutes les entités passeront par le poste de contrôle. Les entités ayant un attribut Temps de valeur 0 passeront tout droit, les autres seront retenues.

Temps de traitement cumulé : temps partagé

Vous pouvez donner comme valeur à un attribut le temps total des traitements qu'il doit subir, puis diriger l'entité vers une série d'opérations, chaque opération soustrayant à l'attribut la part de traitement qu'elle a effectuée. C'est un cas courant lorsqu'on simule des traitements multi-tâche dans des modèles de réseaux informatiques ou de télécommunications. En temps partagé, plutôt que chaque tâche soit effectuée en séquence, de nombreuses tâches demandent à être traitées en même temps, et il n'y a qu'un nombre limité d'opérations pour y faire face. Chaque tâche se voit attribuer une petite fraction de temps puis patiente jusqu'à ce qu'une autre fraction de temps lui soit accordée. Comme il s'agit de temps très courts, cela donne une impression globale de traitement en continu.

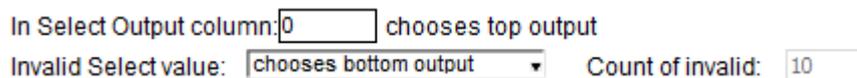
Voici un exemple où chaque tâche est traitée pendant une courte période puis renvoyée dans une file d'attente pour être traitée à nouveau, jusqu'à ce que tout le temps prévu pour la tâche ait été accordé :



temps de traitement partagé : modèle Cumulative Time

L'attribut spécifié dans le dialogue du bloc Set détermine le temps de traitement total requis pour chaque tâche, dans notre cas 3 millisecondes. Le bloc Activity traite la tâche durant un temps fixe de 1 milliseconde, et juste après, le bloc Equation(I) soustrait ce temps (observé sur T) à la valeur de l'attribut.

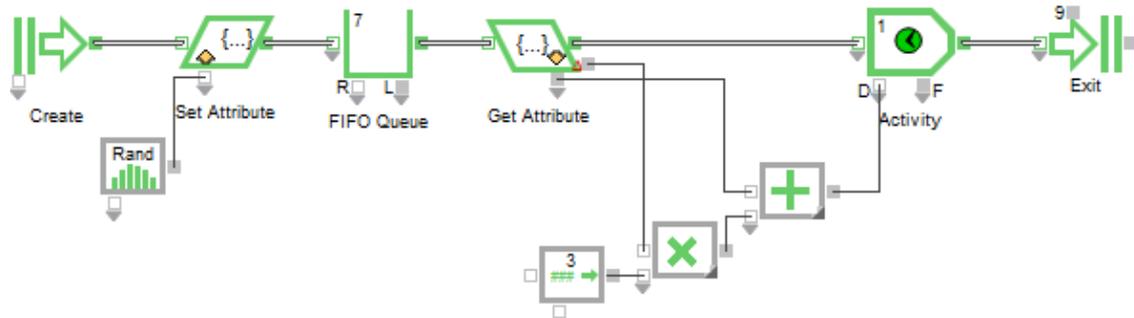
Dans le dialogue du bloc Select Item Out, l'attribut est lu : on spécifie qu'un 0 envoie l'entité sur la sortie du haut. Ainsi, toute tâche qui n'est pas entièrement terminée (attribut TpsRestant vaut 1 ou davantage, compris par le bloc comme « Invalid Select value ») est envoyée sur la sortie du bas et effectue un nouveau cycle.



portion du dialogue du Select Item Out

Ajout d'un temps de réglage

Bien souvent dans un atelier, une machine doit être reconfigurée lorsque le type de pièce change. Ce temps de reconfiguration doit s'ajouter au temps de traitement normal. Dans cet exemple, le temps de traitement ainsi que le type de la pièce sont indiqués par un attribut. Lorsqu'un changement de valeur d'attribut est perçu dans un bloc Get, le connecteur Delta prend la valeur 1. Cette indication est utilisée pour ajouter un temps de réglage au temps de traitement normal :

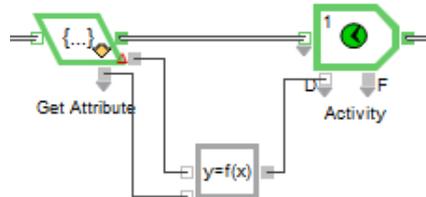


ajout d'un temps de réglage : modèle Setup Time 1

Par défaut, tant que la valeur de l'attribut ne change pas, le connecteur Delta émet 0 : ainsi le bloc Constant qui spécifie le temps de réglage est multiplié par zéro, ce qui n'ajoute rien au temps de traitement normal. Mais lorsque la valeur de l'attribut change, la Constante est multipliée par 1, et le temps de réglage est ajouté au temps de traitement.

Notez que chaque entité conserve son attribut original : l'attribut n'est pas modifié, mais il est utilisé pour déterminer un délai envoyé sur la machine sur son connecteur D.

Pour exprimer l'arithmétique de ce modèle, vous pourriez aussi utiliser un bloc Equation(I), et éventuellement faire appel à une valeur d'attribut pour indiquer aussi le temps de réglage :



temps de réglage par le bloc Equation(I) : modèle Setup Time 2

Le dialogue du bloc serait :

Input Variables			
	Variable Type	Variable Name	Variable Value
1	Connector 0	delta	0
2	Connector 1	process	1

Output Variables (results)			
	Variable Type	Variable Name	Variable Value
1	Connector 0	delay	1

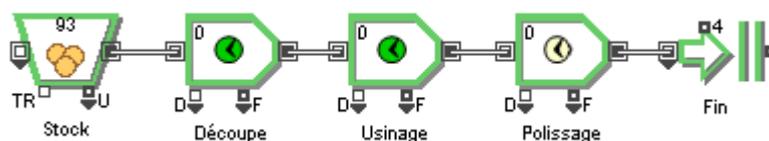
Enter the equation in the form "result = formula;"

delay=delta*3+process;

détermination du temps de réglage

Traitement en série

On parle de traitement en série lorsque plusieurs tâches obligatoires qui se suivent en séquence affectent une entité. Voici un exemple simple :



traitement en série : modèle Serial Processing

Comme les postes se suivent, il est probable que parfois les entités ne pourront pas quitter un poste car le poste suivant est occupé.

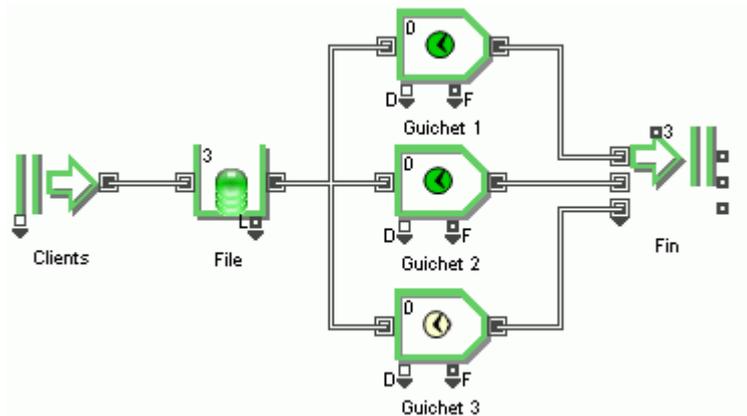
Opérations en parallèle

Il existe plusieurs possibilités pour modéliser plusieurs opérations effectuant la même tâche (traitements parallèles).

N'oubliez pas que, à moins d'être clonées, les entités ne suivent qu'un chemin à la fois. Le bloc Transaction permet de traiter plusieurs entités simultanément et peut souvent représenter des opérations en parallèle. Ici nous examinons des opérations parallèles dont chacune est explicitée par un bloc.

Liens simples en parallèle

La méthode la plus simple pour gérer des traitements parallèles est de créer des liens entre la sortie du flux et chaque entrée d'opération en parallèle. Ainsi, ExtendSim passe les entités au premier poste disponible. Si plusieurs postes sont disponibles en même temps, l'entité va sur le poste qui a été lié en premier. Par exemple, voici une file qui sert trois guichets de cette façon :

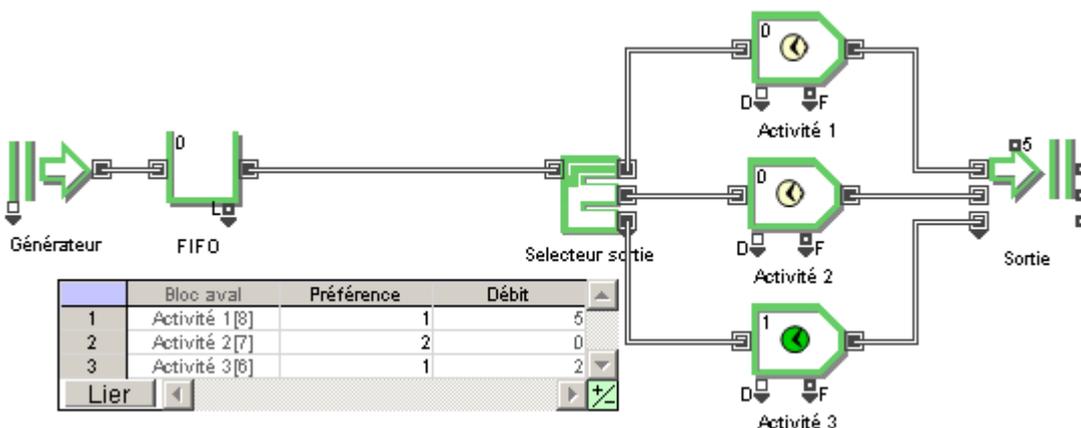


traitement parallèle simple

Ordre explicite

Si vous préférez que certaines opérations soient actives plutôt que d'autres, vous pouvez spécifier quelles opérations auront priorité pour traiter les entités. C'est un cas courant lorsque vous voulez éviter une opération moins efficace (sur une vieille machine) ou coûteuse (utilisant plusieurs opérateurs). Notez qu'il existe de toutes façons un ordre implicite : mieux vaut donc l'expliciter.

Le bloc **Select Item Out** (bib. Item) permet de spécifier dans le dialogue ou par des connecteurs quelle sortie est prioritaire. C'est bien différent de la notion de priorité attachée aux entités, car le Sélecteur distingue des chemins, pas des entités. Par exemple, vous voulez que les postes A et C reçoivent le plus d'entités, et le poste B uniquement en dernier recours :



traitement parallèle avec ordre explicite ; modèle Explicit Ordering

Dans le dialogue du Select Item Out, spécifiez le mode **Connector Priority** et donnez les plus hautes préférences (le plus petit nombre) à la première et troisième sortie :

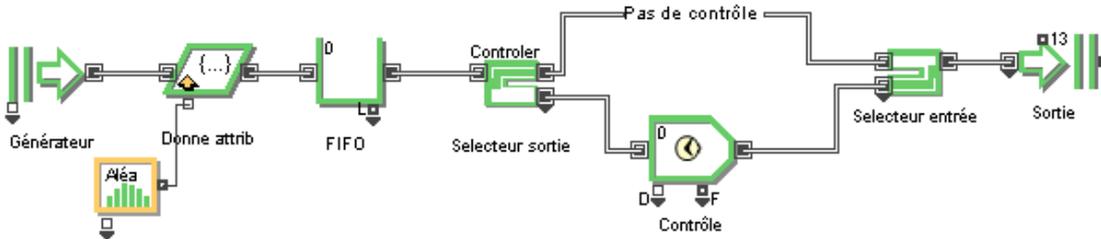
L'ordre de priorité est donc A, C, B. Vous pouvez donner la même préférence à plusieurs sorties (les sorties de même niveau sont traitées comme des postes parallèles simples).

Ordre de succession

Pour orienter les entités successivement à trois machines, sans tenir compte de la disponibilité, mais afin de répartir la charge, spécifiez le mode **Sequential** pour utiliser le système de bascule prévu dans le dialogue du bloc Select Item Out.

Choix d'un poste selon l'attribut ou la priorité

Le bloc Select Item Out est adapté pour orienter une entité selon un attribut ou une priorité. Par exemple, l'attribut peut dire qu'il doit y avoir un contrôle ou non, comme dans l'exemple ci-après :



traitement parallèle sélectionné selon l'attribut : modèle Attributes for Routing

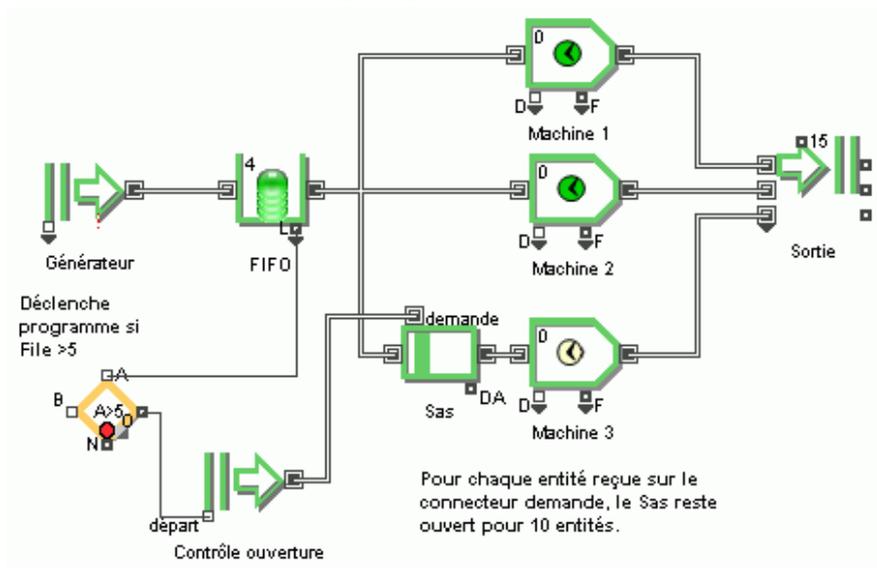
Vous affectez aux entités un attribut de valeur 1 ou 0, qui est lu par le Sélecteur qui effectue sa décision.

Activer un système

Nous avons vu des manières d'activer un système en fonction de l'heure ou d'un état du modèle. Mais la méthode employée peut provoquer des mises en route et des arrêts courts et fréquents sur les postes ouverts. Or activer un système (un ou plusieurs postes) est souvent coûteux, il faut que cela "en vaille la peine".

Pour un certain nombre d'entités

Si des conditions sont réunies pour l'activation d'un système, il vous est possible de maintenir le système actif pour un certain nombre d'entités, même si l'encours qui a déclenché l'activation est repassé en-deçà du seuil critique. Un désengorgement durable de l'encours peut ainsi avoir lieu, et le système est activé moins souvent. Pour cela, ajoutez un Create en mode **Create items by schedule** avant le bloc Gate en mode **Conditional gating with items**:



Activation pour un certain nombre d'entités : modèle Fixed items

Lorsqu'il est activé, le planning émet un élément discret de quantité 10. Cela fait que le bloc Gate reste ouvert tant qu'il n'a pas vu passer 10 entités. Voici le dialogue du Create :

—Enter a schedule of arrival times—

	_Create Time	_Item Quantity
1	0	10
2	5	0

activation du connecteur Demand par une entité

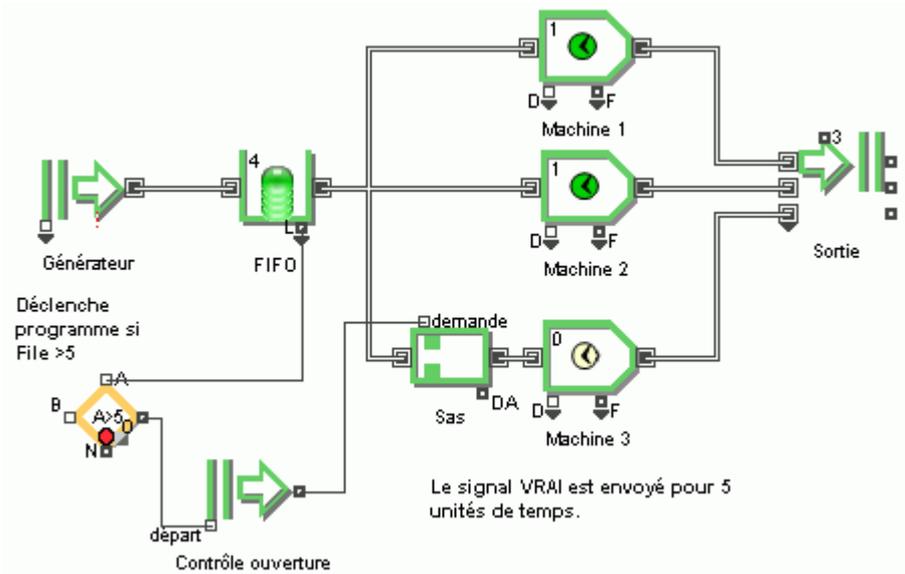
Notez bien les points suivants :

- Comme le connecteur *start* est relié, le Create est déclenché de manière relative. Tout le planning s'exécute alors : la seconde ligne permet que le programme ignore tous les messages sur *start* pendant 5 unités de temps. Ce temps doit être calculé pour permettre à la machine de traiter les entités. Si cela ne figurait pas dans le planning, le bloc Gate pourrait être activé constamment.
- C'est le connecteur *demand* qui active le bloc Gate, selon la logique, c'est à dire qu'une demande est créée pour laisser passer autant d'entités que le spécifie la quantité de l'entité émise.

Connecteur Start : le bloc Create possède un connecteur *start* qui s'emploie pour déclencher le planning entré dans le dialogue. Si ce connecteur n'est pas utilisé, les temps du planning sont lus de manière absolue par rapport au temps de la simulation. S'il est lié, par exemple au bloc Decision de la bibliothèque Value, le temps entré dans le planning devient relatif par rapport au temps où est activé le connecteur Start. Par exemple, si le planning mentionne le temps 2 et que le connecteur start est activé au temps 3, le planning va démarrer au temps 5 (2+3). Le connecteur Start est activé par une entité ou la valeur *True* (une valeur > 0,5).

Pour un certain temps

Une variante est de laisser le système activé ouvert pendant une certaine durée. Pour cela, nous utilisons les mêmes blocs, mais en activant le bloc Gate en mode **Conditional gating with values**:



activer un système pour un certain temps : modèle Fixed time

Notez que c'est le connecteur continu qui sort du Create, en mode **Create values by schedule**. Le programme est rempli différemment, par exemple pour que l'activation dure 5 unités de temps :

- Enter a schedule of arrival times -

	_Create Time	Value 1
1	0	1
2	5	0

activation du connecteur Demande par une valeur

Une fois activé, le connecteur demande reçoit une valeur *vrai* (1), ce qui ouvre la Gate. Après 5 unités de temps, la valeur passe à faux (0), ce qui ferme le bloc Gate.

Opérations interrompues

De multiples causes peuvent provoquer des arrêts sur des process : pause des opérateurs, pannes, tâche prioritaire, etc. C'est un des aspects réalistes de la simulation de prendre en compte ces arrêts, pour ne pas avoir une vision trop optimiste de la réalité.

Vous pouvez provoquer des arrêts selon des plannings (nettoyage journalier ou pause-café) ou selon des hasards (pannes). Vous pouvez encore observer tel signal dans le modèle (saturation de tel poste) pour décider d'un arrêt. Dans tous les cas, la durée de l'arrêt peut être constante ou aléatoire.

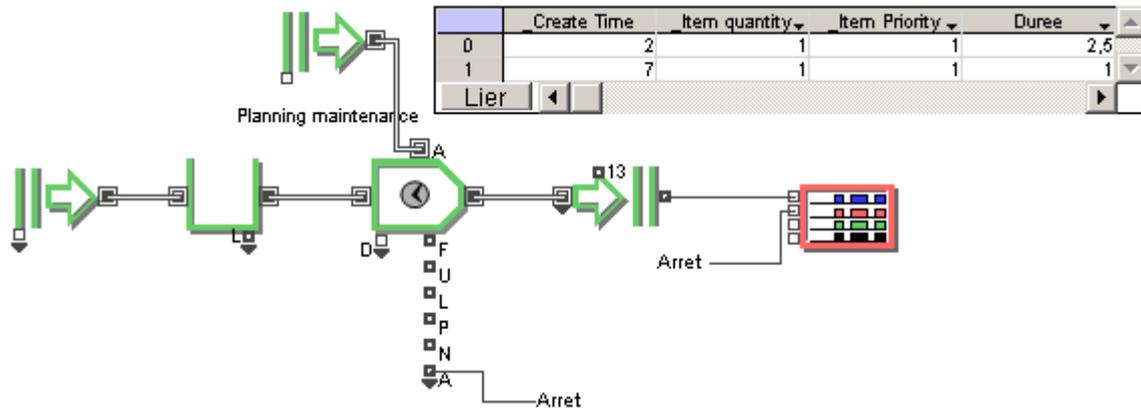
C'est le connecteur d'entrée **SD** ou **shutdown** (arrêt) de l'Activité (voir également le bloc Valve de la bib. Rate) qui est utilisé pour gérer des arrêts. Le connecteur de sortie Sd montre le nombre d'entités

arrêtées, et DT la durée en arrêt.

L'activation du connecteur provoque l'arrêt du poste. Vous spécifiez dans le dialogue si la valeur reçue sur le connecteur doit être interprétée comme une durée ou comme un interrupteur. Durant un arrêt, tous les messages d'arrêt sont ignorés.

Arrêt programmé

Utilisez un bloc Create en mode **Create items by schedule** directement lié au connecteur SD :



arrêt d'un poste par programme : modèle Scheduled Shutdown

Pour prévoir par exemple un nettoyage à 2 heures et un à 7 heures, remplissez le Programme comme ci-dessus. Au temps 2, l'activité reçoit une entité dont l'attribut DownTime donne la durée de l'arrêt. Vous pouvez aussi spécifier dans le dialogue que le planning doit être répété. Dans l'onglet Shutdown de l'activité, spécifiez comment doit s'interpréter ce signal.

Enable shutdown

SD (shutdown) input is from:

When signal is received at SD input, shutdown...

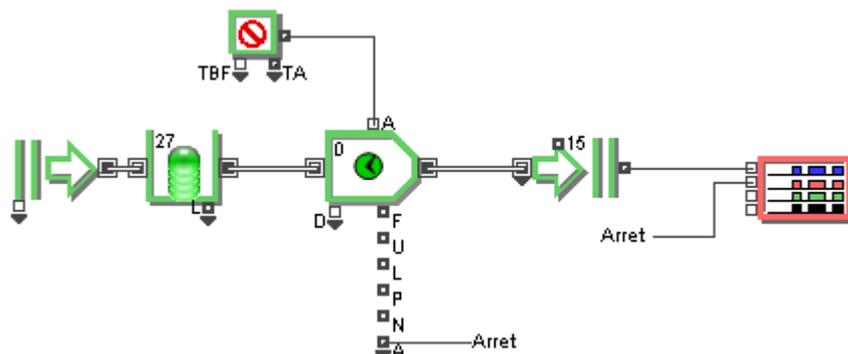
Shutdown duration specified by:

When activity shuts down:

dialogue de l'activité (partiel)

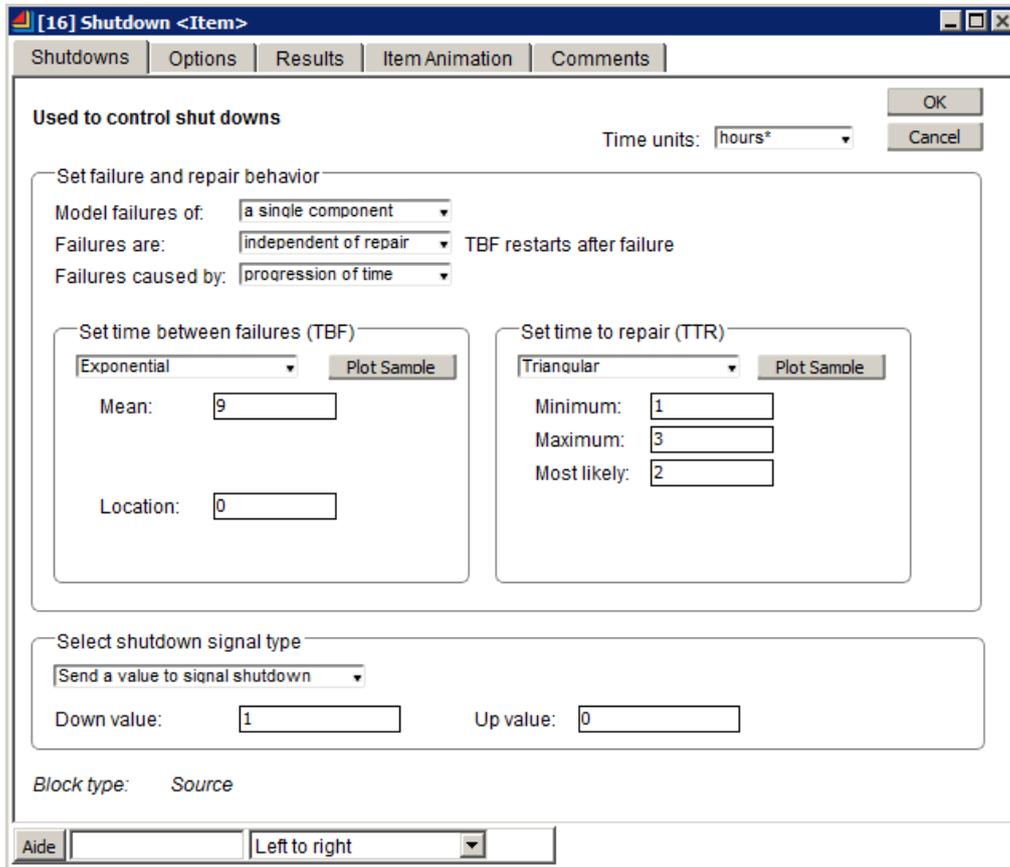
Arrêt aléatoire

Vous pouvez provoquer des arrêts aléatoires de durée fixe, et des arrêts aléatoires de durée aléatoire. Cependant, ne liez pas le bloc Random Number ou tout autre bloc de la bibliothèque Value au connecteur d'arrêt d'une activité. Vous devez passer par un bloc Shutdown de la bib. Item pour envoyer des valeurs au connecteur A.



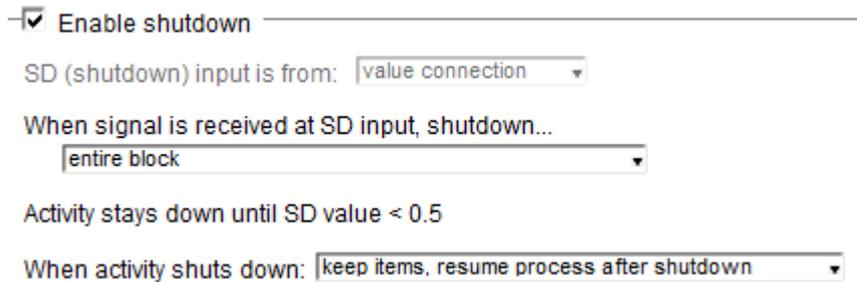
pannes aléatoires sur un bloc activité : modèle Random Shutdown

Dans le dialogue du bloc Shutdown, la partie gauche permet de décrire la loi d'arrivée des pannes (leur fréquence), et la partie droite la durée des pannes. Ici les pannes surviennent selon une loi exponentielle de moyenne 9, et elles durent entre 1 et 3 heures, la durée + probable étant de 2h.



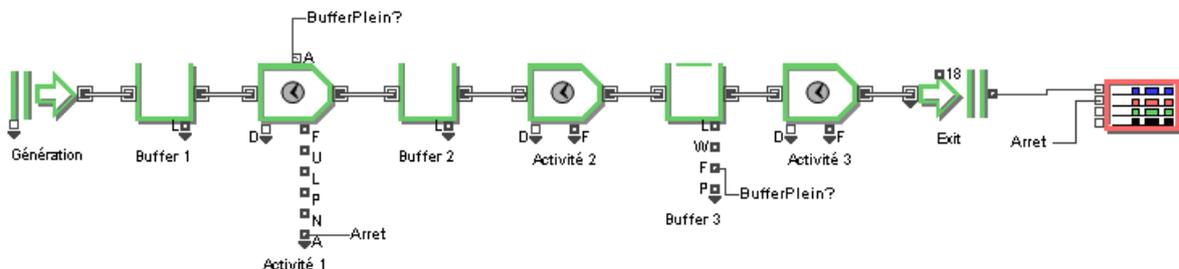
dialogue de Shutdown

L'onglet Shutdown de l'activité est bien différent de l'exemple précédent :



Arrêt explicite

Les exemples précédents ont montré comment arrêter une machine de manière autonome, non fonction de l'environnement. Si vous voulez lier l'arrêt à un état du modèle, par exemple un goulet d'étranglement en aval de l'activité, liez le connecteur F d'une file au connecteur SD :



arrêt explicite : modèle Explicit Shutdown

Vous spécifiez la capacité du buffer dans son dialogue ; lorsqu'il devient plein, le signal F passe à 1. Pour que l'activité soit arrêtée tant que le buffer est plein, choisissez de lire un signal continu.

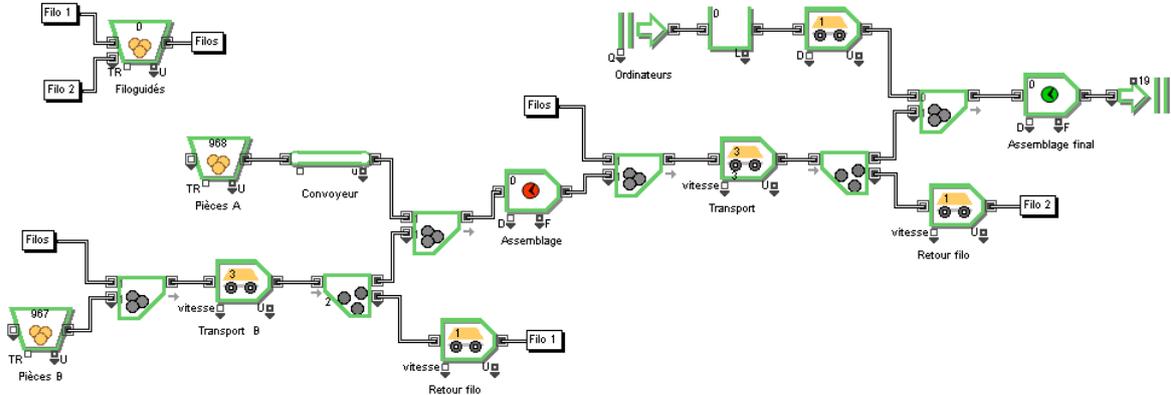
Ainsi des facteurs en amont peuvent influencer des opérations en aval. De toute façon, si une file d'attente est saturée, elle engorgera par conséquent la machine qui la précède, et il y aura bien arrêt du traitement. Mais en demandant un arrêt explicite, vous choisissez quel poste doit réagir à un goulet

d'étranglement, et où doit se former l'encours.

Blocs de transport

Les blocs Convey Item et Transport servent à représenter des transports d'entités d'un point à un autre. Vous utilisez ces blocs n'importe où dans le modèle et vous attribuez à chacun une vitesse et une longueur en fonction de votre système. En conjonction avec les blocs Resource Item et Batch/Unbatch, vous représenterez en outre des véhicules se déplaçant sur des trajets.

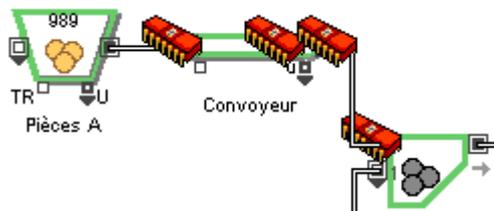
Exemple d'emploi de blocs de transport :



Modèle de transport : modèle Transportation 1

Ce modèle utilise plusieurs blocs Transport, les premiers pour prendre les chariots filoguidés et les autres pour les ramener après usage. Vous spécifiez dans le dialogue le nombre maximum de véhicules pouvant circuler sur le même trajet (ici 3). Dans notre cas, les filoguidés sont libérés à la fin de chaque trajet, et l'on prend en compte le temps d'un retour à vide des filoguidés.

Le Convoyeur utilisé est assez similaire au bloc Transport, mais présente des possibilités d'animation plus complètes encore, même en 2D. Dans l'onglet Animation Convoyeur, on a coché la case *2D Animation shows simultaneous item movements along connections*.



Dans le dialogue du bloc, notez à la fois les options d'accumulation, et leur corollaire, qui est la place prise par chaque entité.

— Specify conveyor behavior and how item length is determined —

Accumulating conveyor

Travel time: move time

Move time: 1 time units

Length: 70 meters

Capacity: 4

Item length is: a constant 1

Orientation du flux

En construisant des modèles, vous vous apercevrez que dans bien des cas, vous aurez besoin de manipuler des entités venant de plusieurs sources ou de les envoyer vers plusieurs destinations possibles. Plusieurs méthodes sont possibles en fonction de l'objectif à atteindre.

☞ Les modèles commentés dans ce chapitre se trouvent dans le dossier `Exemple\Item Event\Routing`. Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Blocs fréquemment utilisés

Blocs qui orientent des entités



Throw (Item > Routing)

Envoie des entités à distance vers un bloc Catch.



Catch (Item > Routing)

Reçoit des entités envoyées à distance par un bloc Throw.



Select Item Out (Item > Routing)

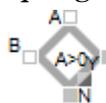
Dirige chaque entité vers une sortie choisie.



Select Item In (Item > Routing)

Choisit une entrée et fait sortir les entités du bloc relié à cette entrée.

Blocs qui agissent sur le flux d'entités



Decision (Value > Math)

Peut être utilisé avec des blocs de la bibliothèque Item pour contrôler le flux d'entités dans une partie du modèle.



Gate (Item > Routing)

Contrôle le flux d'entités dans une portion du modèle ou en fonction des conditions du modèle.



Math (Value > Math)

Effectue un calcul mathématique (addition, soustraction, etc.) qui peut être utilisé avec des blocs de la bibliothèque Item pour effectuer des calculs.



Max & Min (Value > Math)

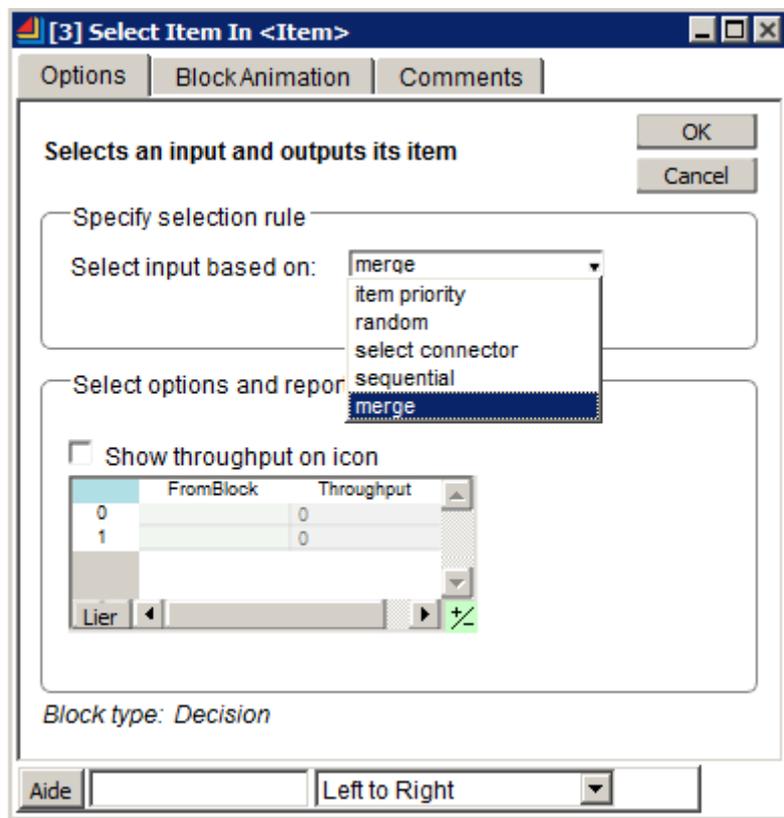
Emet la valeur minimum ou maximum lue sur ses différents connecteurs d'entrée. Peut être utilisé dans les modèles discrets pour prendre des décisions.

Entités provenant de plusieurs sources

Selon vos besoins de modélisation, vous pouvez avoir besoins de réunir plusieurs flux d'entités en un seul flux d'entités individuelles, de sélectionner une entité parmi d'autres pour l'orienter ou la traiter, ou d'assembler plusieurs entités pour n'en faire qu'une.

- Pour réunir en un seul flux des entités provenant de plusieurs sources, où chaque entité garde son individualité, utilisez les blocs **Select Item In**, **Throw** et **Catch**. Généralement, les entités finissent par aboutir dans une file d'attente. Cela représente exactement plusieurs routes qui fusionnent en une seule route, ou l'arrivée de clients par plusieurs portes d'entrée. Le bloc Select Item In permet de :
 - Réunir des flux d'entités (cf. modèle Merging Inputs)

- Diriger des entités qui nécessitent un temps de traitement plus long (cf. modèle Orientation par attribut)
- Réorienter des entités (cf. modèle Prémption)
- Pour sélectionner une entité à traiter parmi plusieurs sources en fonction d'une certaine décision, utilisez le bloc **Select Item In**. La décision peut être une décision logique (choisissez d'orienter une entité sur deux vers la file du haut) ou être basée sur une caractéristique de l'entité (obtenir l'entité qui a la plus haute priorité). Les spécifications qui permettent de prendre la décision dépendent des données que vous entrez dans le dialogue du bloc Select Item In et sont modifiées par les blocs reliés à son connecteur d'entrée « Select ».
- Pour réunir des entités provenant de plusieurs sources afin de les traiter comme une seule unité, utilisez un bloc **Batch**. C'est ce qui représente l'opération d'assemblage de composants de base, ou l'association momentanée de deux entités pour un traitement ou un trajet. La mise en lot diffère tout à fait de l'utilisation des blocs Select Item In, Throw et Catch qui ne réunissent que des flux d'entités où chaque entité reste unique et est traitée seule.
- Pour réunir des flux d'entités issus d'un même niveau hiérarchique en un seul flux sur un niveau hiérarchique différent, vous pouvez ajouter des connecteurs au bloc hiérarchique ou utiliser les blocs Throw et Catch.



Dialogue du bloc Select Item In

Le bloc Select Item In choisit une entité sur un de ses connecteurs d'entrée et envoie cette entité vers son connecteur de sortie. Le choix dépend des paramètres et options du dialogue.

Options du Select Item In

Le bloc Select Item In propose plusieurs règles pour sélectionner une entité parmi ses connecteurs d'entrée :

- **Item priority** (*Priorité de l'entité*). Choisit le connecteur d'entrée qui a une entité disponible et avec la plus haute priorité (plus petite valeur numérique). Par exemple, vous pouvez choisir cette option pour sélectionner une entité parmi plusieurs files d'attente. La file qui possède l'entité ayant la plus haute priorité sera sélectionnée. La sélection se fait toujours en commençant par l'entrée du haut.
- **Random** (*Aléatoire*). Les entrées sont sélectionnées de façon aléatoire en fonction de probabilités entrées sous forme décimale dans la table de sélection du bloc (0.75 pour 75 %). Si le total des nombres entrés n'est pas égal à 1, la somme apparaîtra en rouge en bas de la colonne Probability.

Si l'option *Select from : all inputs* (Sélectionner parmi : toutes les entrées) est choisie, une entrée sera sélectionnée au hasard, même si aucune entité n'est disponible sur cette entrée. Cette situation peut entraîner un arrêt temporaire de l'activité en aval. Si l'option *Select from : only inputs with available item* (Sélectionner parmi : entrées avec entités disponibles) est choisie, seules ces entrées seront prises en compte dans la sélection.

- **Select connector (Connecteur Select).** L'entrée choisie est déterminée par la valeur lue sur le connecteur *Select*. Dans le dialogue du bloc, une option permet de spécifier quelle valeur choisit l'entrée du haut ; la valeur par défaut est 0. Les connecteurs situés en dessous seront numérotés de façon séquentielle après le connecteur du haut. Si l'entrée du haut est choisie par la valeur 1, les autres entrées sont sélectionnées en ajoutant 1 à la valeur du connecteur au-dessus. Ainsi, la deuxième entrée sera choisie par la valeur 2, l'entrée suivante par la valeur 3, etc. Dans ce cas, si une valeur de 3 est lue sur le connecteur *Select*, c'est l'entité provenant du troisième connecteur en partant du haut qui sera sélectionnée. Notez que même si des entités sont disponibles sur d'autres entrées, le bloc attendra une entité à l'entrée numéro 3, même si cela doit entraîner un ralentissement de l'activité en aval.
- **Sequential (Séquentiellement).** Sélectionne les entrées selon un ordre séquentiel strict en partant du haut. Cette possibilité, aussi appelée sélection « circulaire », peut entraîner un manque d'activité en aval car le bloc attend qu'une entité soit disponible à chaque entrée sélectionnée.
- **Merge (Convergence).** Les entités sont prises dès qu'elles sont disponibles, quelle que soit l'entrée d'où elles proviennent. Généralement, cette option est utilisée pour réunir des flux d'entités lorsqu'il n'y a pas de blocage à l'entrée du bloc *Select Item In*. Les entrées sont choisies selon une méthode circulaire en commençant du haut ; une fois qu'un choix a été fait, la recherche reprendra à partir du connecteur suivant.

Conditions d'inactivité

Si aucune entité n'est disponible sur l'entrée sélectionnée sur le bloc *Select Item In*, les options suivantes entraîneront l'inactivité :

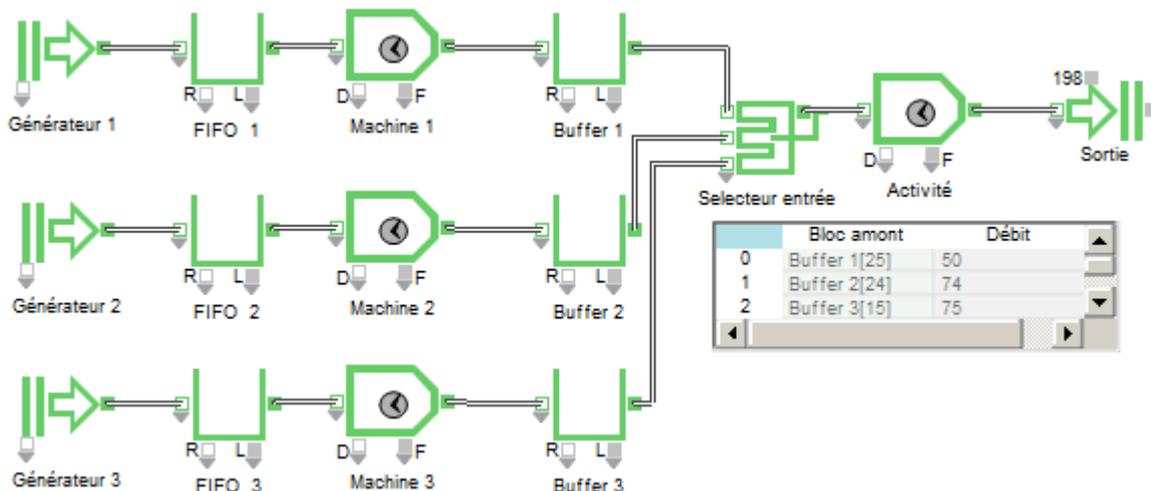
- Aléatoire (si l'option *Select from : all inputs* est choisie)
- Connecteur *Select*
- Séquentiellement

Convergence de flux

Le bloc *Select Item In* peut réunir les entrées de plusieurs sources en un seul flux d'entités sorties.

Modèle Merging Inputs

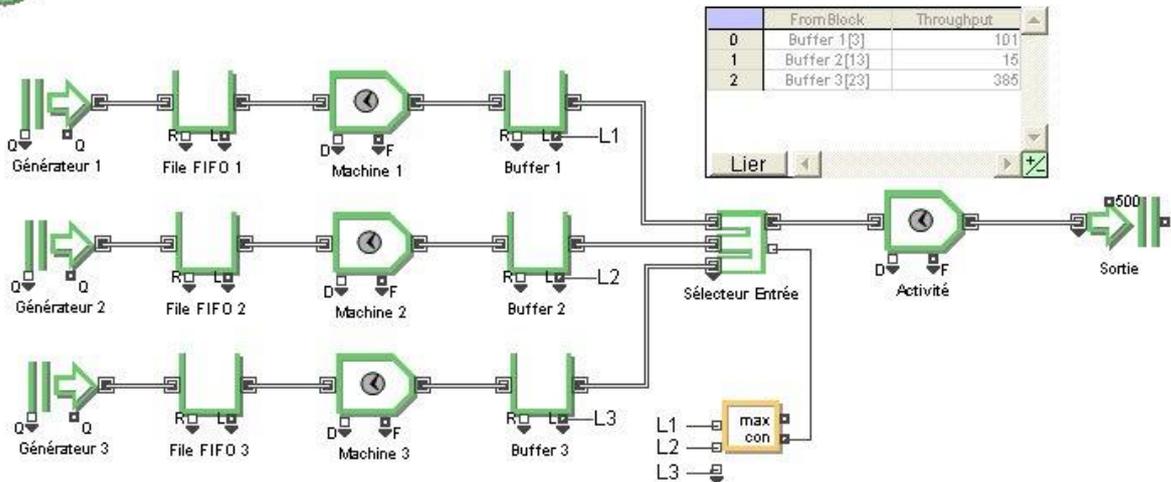
Dans cet exemple, le bloc *Select Item In* accepte des entités provenant de l'une des trois entrées. Dans le dialogue, l'option *Select input based on : merge* est choisie. Si la sortie du bloc *Select Item In* est bloquée, celui-ci obligera les entités à attendre dans les files (nommées *Buffer 1*, *2* et *3*). Lorsque le sélecteur se débloque, il vérifie chaque entrée pour essayer de faire passer une entité dans l'activité pour être traitée. Comme le montre la table du bloc *Select Item In*, dès que l'activité est prête à traiter des entités, elle reçoit la première entité disponible. Cela peut entraîner des files d'attente plus longues que d'autres, comme le montre leur onglet *Results*.



Équilibre des lignes

Pour équilibrer la longueur des files d'attente sur différentes lignes, utilisez un bloc Select Item In contrôlé par un bloc *Max & Min* (bibliothèque Value) qui vérifie la longueur de chaque file. Par exemple, vous voulez décharger en premier le quai de livraison qui est le plus plein.

Modèle Input Line Balancing



Sur le bloc Max & Min, le connecteur de sortie *con* indique au bloc Select Item In quelle est la file la plus longue, et donc dans laquelle il doit aller chercher la prochaine entité. Dans le dialogue du Select, les options *Select input based on : Select* et *Top input is chosen by Select value: 1* sont sélectionnées. La Table de débit clonée sur la feuille de travail montre que les entités sont tirées de chaque file de façon équilibrée et la longueur des différentes files est presque identique, contrairement au modèle de convergence de flux vu précédemment.

Convergence de flux discrets avec les blocs Throw et Catch

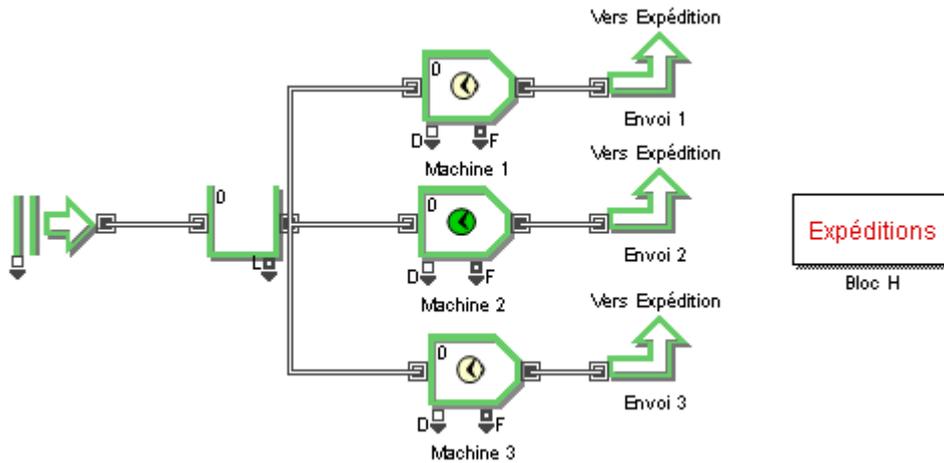
Les exemples précédents traitaient de l'orientation des entités au moyen de blocs situés à proximité et sur le même niveau hiérarchique. Parfois, surtout dans les grands modèles, une entité doit être envoyée vers un autre niveau hiérarchique. Les blocs Throw et Catch sont particulièrement utiles pour envoyer des entités se trouvant à divers endroits du modèle (ou sur des niveaux hiérarchiques différents) vers un autre endroit. Notez que ces blocs sont complémentaires aux méthodes d'orientation expliquées ci-dessus mais ne les remplacent pas.

Les blocs Throw et Catch passent des entités sans besoin d'être liés et permettent également d'envoyer des entités vers d'autres blocs hiérarchiques. C'est pourquoi ils sont parfois utilisés pour remplacer les blocs Select Item In et Select Item Out.

- N'utilisez les blocs Throw et Catch que lorsque les liens nommés ne suffisent pas (par exemple, pour passer des entités d'un niveau hiérarchique à un autre ou pour utiliser les options d'orientation sur ces blocs).

Modèle Throw & Catch

Les blocs Throw et Catch permettent également de réunir plusieurs flux d'entités en un seul. Dans l'exemple suivant, trois blocs Throw orientent les entités vers un bloc hiérarchique Transport, comprenant deux blocs : un bloc Catch et un bloc Exit. Le bloc Catch s'appelle *Vers Expédition* et appartient au *Catch Groups 1*.



Dans le dialogue du bloc Throw, un menu déroulant affiche le label des blocs Catch candidats. Vous avez la possibilité d'orienter toutes les entités vers le bloc Catch spécifié dans le menu déroulant ou vers différents blocs Catch en fonction de la valeur d'un attribut spécifié (cf. modèle Throw & Catch Attributes)

- Vous devez donner un label à chaque bloc Throw (dans le champ situé en bas à gauche du dialogue de chaque bloc). Seuls les blocs Catch ayant un label apparaîtront dans le menu déroulant du bloc Throw.

Groupes de blocs Catch

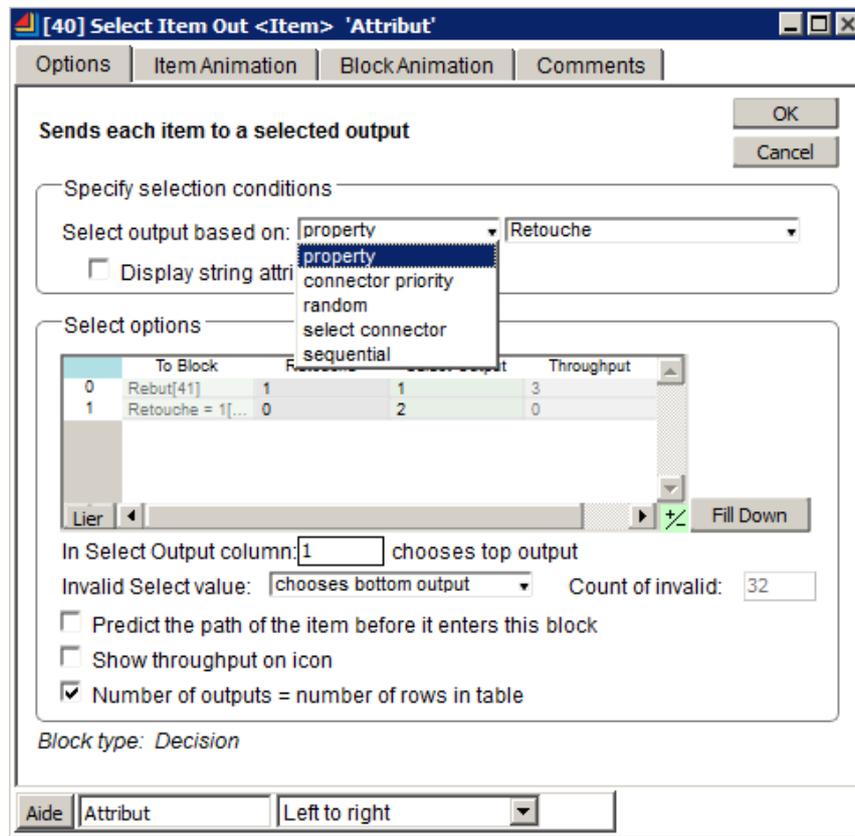
Si vous travaillez avec un grand nombre de blocs Catch, vous pouvez les organiser en groupes. Pour cela, sélectionnez ou créez un nom de groupe en utilisant le menu déroulant *Catch Item Group* dans le bloc Catch. Ensuite, sélectionnez le groupe désiré dans le menu déroulant *Catch Item Group* du bloc Throw. Une fois qu'un groupe est sélectionné dans le bloc Throw, le menu déroulant *Specify by : label* ne contient que les labels des blocs de réception du groupe sélectionné.

- ☞ Les groupes ne peuvent être définis que dans un bloc Catch.

Décisions d'orientation : divergence de flux

Dans bien des cas, vous devrez orienter des entités vers un flux en particulier.

- Prendre un flux d'entités et les diriger vers un groupe d'opérations ou de machines est appelé *traitement parallèle*. Dans un traitement parallèle, chaque entité est envoyée à l'un des postes. La logique qui détermine sur quel poste sera envoyée chaque entité peut être simple (la pièce est usinée sur la première machine disponible) ou complexe (la bouteille de type 4 est remplie sur la Machine 3). Le traitement parallèle a été [examiné](#).
- Quand une entité doit être décomposée en plusieurs entités, utilisez le bloc **Unbatch**. Par exemple, vous recevez une livraison de meubles composée de 8 bureaux, 20 chaises et 7 micros. Utilisez un bloc Unbatch pour décomposer l'entité en ses composants individuels, puis dirigez chaque entité à sa destination prévue.
- Pour sélectionner le chemin qu'une entité doit prendre, utilisez les blocs **Select Item Out** ou **Throw**. Le bloc Select Item Out est utile pour diriger un flux d'entités vers plusieurs chemins, en fonction d'une décision. Par exemple, vous envoyez une certaine portion des pièces au contrôle technique, et envoyez le reste aux postes de conditionnement.



Dialogue du bloc Select Item Out

Le bloc Select Item Out permet d'orienter des entités vers un chemin ou un autre. Son dialogue comporte plusieurs options permettant de déterminer ce chemin.

Options du Select Item Out

Dans le dialogue du bloc Select Item Out, la logique choisit le connecteur de sortie vers lequel une entité entrante doit être dirigée. Les options sont les suivantes :

- **Property** (*Selon propriété*). La sortie appropriée est déterminée en fonction de l'attribut ou de la priorité de l'entité. Les valeurs d'attribut sont entrées dans la colonne 2 du tableau. Par défaut, la valeur 0 choisit la sortie du haut. Pour chaque entité, le bloc trouve la valeur de l'attribut spécifié dans la colonne 2 de la table de correspondance et détermine le connecteur de sortie correspondant à cette valeur dans la colonne *Select Output*. Cette option peut entraîner une situation de blocage car le bloc retient l'entité jusqu'à ce qu'une activité puisse l'accueillir.
 - **Connector priority** (*Préférence des sorties*). Le bloc essaie de faire sortir l'entité par chaque connecteur, en suivant l'ordre de priorité des connecteurs, jusqu'à ce que l'entité soit acceptée par un bloc lié. La priorité de chaque connecteur est entrée dans un tableau dans le dialogue du bloc. La plus haute priorité correspond à la plus petite valeur. Ainsi, une sortie ayant une valeur de priorité de 1 est plus prioritaire qu'une sortie avec une valeur de priorité de 3.
- ☞ Notez que cela n'est pas la même chose qu'attribuer une priorité à une entité. En effet, le bloc Select Item Out donne essentiellement la priorité au chemin de sortie, pas à l'entité.
- **Random** (*Aléatoirement*). Les sorties sont choisies de façon aléatoire selon des paramètres définis dans la table de probabilité du bloc. Les probabilités doivent être entrées sous forme décimale. Par exemple, entrez 0.75 pour 75 %. Si la somme des nombres entrés n'est pas égale à 1, celle-ci apparaîtra en rouge au bas de la colonne *Probability*. Si l'option *If output is blocked : try unblocked output* est sélectionnée, le bloc essaiera de trouver une sortie aléatoire pouvant accepter l'entité. Quand l'option *If output is blocked : wait* est sélectionnée, le bloc choisit une sortie et l'entité attend jusqu'à ce que cette sortie puisse l'accepter. Cela peut provoquer un blocage.
 - **Select connector** (*Connecteur Select*). La valeur lue sur le connecteur Select détermine la sortie choisie. Le dialogue du bloc comporte une option permettant de définir quelle valeur choisit la sortie du haut (0 par défaut). Les autres sorties sont sélectionnées en ajoutant 1 à la valeur du connecteur au-dessus. Ainsi, si l'entrée du haut est choisie par la valeur 1, la deuxième entrée sera choisie par la valeur 2, la troisième par la valeur 3, etc. Dans ce cas, si la valeur 3 est lue sur le connecteur Select,

L'entité ira vers le troisième connecteur en partant du haut. Cette option peut entraîner un blocage car le bloc retient l'entité jusqu'à ce que celle-ci puisse sortir par le troisième connecteur.

- **Sequential** (*Séquentiellement*). Les sorties sont choisies les unes après les autres selon un ordre séquentiel en commençant par le haut. Quand l'option *If output is blocked : try unblocked output* est sélectionnée, le bloc essaie les connecteurs suivants dans l'ordre. Quand *If output is blocked : wait* est sélectionnée, le bloc choisit une sortie et l'entité attend jusqu'à ce que cette sortie puisse l'accepter. Cela peut provoquer un blocage.

Le bloc Select Item Out ne compare que des valeurs entières et tronquera les valeurs décimales. Par exemple, si la sélection se fait en fonction de la valeur lue sur le connecteur Select, les nombres 0,001 et 0,009 seront tronqués et lus comme des 0.

Conditions de blocage

Le bloc Select Item Out est un bloc de décision. Par défaut, il tire l'entité et détermine ensuite le chemin qu'elle prendra. Dans certaines situations, il se peut que le chemin de sortie choisi soit bloqué et que l'entité doive attendre avant de pouvoir sortir. Plusieurs conditions de sélection peuvent entraîner le blocage des entités :

- Property
- Random (option *If output is blocked : wait*)
- Select
- Sequential (option *If output is blocked : wait*)

Dans le cas des sélections Random et Sequential, il est utile de pouvoir choisir ce qui doit se passer si la sortie est bloquée. Par exemple, dans le modèle d'orientation simple décrit ci-dessous, si le bloc Queue du haut est choisi pour recevoir l'entité mais qu'il est bloqué, le bloc Select Item Out orientera l'entité vers une file qui n'est pas bloquée.

Connaitre le chemin de sortie d'une entité avant qu'elle n'entre dans le bloc

Une entité peut très bien être tirée par un bloc Select Item Out mais impossible à traiter car le chemin en aval est bloqué. Une des solutions est de faire attendre l'entité dans une file en amont plutôt que dans le bloc Select Item Out. Pour cela, cocher l'option *Predict the path before item enters*. Le bloc Select Item Out regardera alors en amont afin de déterminer les propriétés de la prochaine entité. Puis il vérifie que le chemin correspondant en aval est libre. C'est uniquement si la sortie voulue peut accueillir l'entité que celle-ci sera tirée. Cela permet de s'assurer que l'entité ne restera pas bloquée dans le bloc Select Item Out.

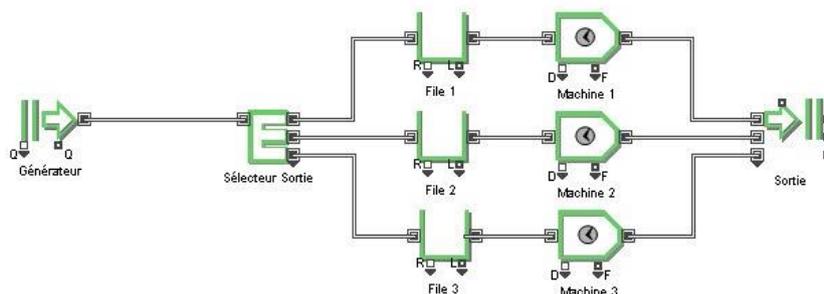
- ☞ Pour utiliser cette option, toutes les propriétés utilisées dans la sélection doivent être attribuées avant que l'entité ne se dirige vers le bloc Select Item Out. Par exemple, il faut placer une Queue entre les blocs Set et Select Item Out.

Orientation simple

Le type d'orientation le plus simple est celui où un certain nombre d'entités doit être dirigé d'un côté du modèle, et le reste d'un autre. C'est le cas lorsqu'à un carrefour, certaines voitures de la file vont tourner à gauche, d'autres à droite, et d'autres vont continuer tout droit.

Modèle Simple Routing

Dans ce modèle, les entités sont orientées de façon aléatoire vers l'une des trois machines, comme l'indique le paramètre *Select output based on : random*. Dans le dialogue du bloc Select Item Out, la table de probabilité montre que 0.50 (50 %) des entités se dirigeront vers la file du haut alors que les autres seront réparties équitablement entre les deux autres files. L'option *If output is blocked : try unblocked output* permet au bloc Select Item Out d'essayer d'autres sorties si la sortie choisie ne peut pas accepter l'entité.

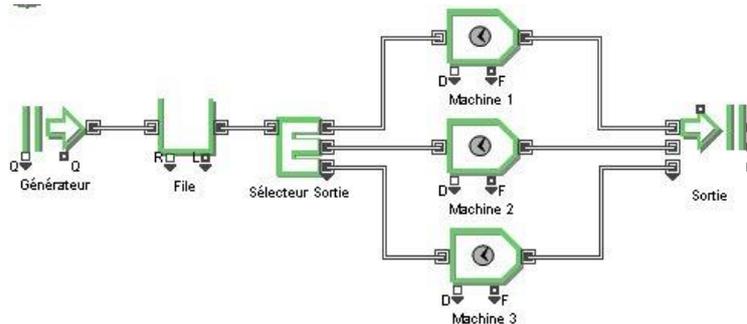


Vous remarquerez que des files d'attente ont été placées en amont des machines. Sans ces files, il est possible que les entités provenant du bloc Create restent bloquées. Par exemple, si la deuxième entité doit passer dans la machine du haut mais que celle-ci n'a pas terminé de traiter la première entité, les autres machines devront attendre que la machine du haut ait fini le traitement et accepte la deuxième entité.

- Dans le modèle précédent, les entités étaient orientées selon des probabilités. Pour diriger une entité entrante vers une sortie disponible, sélectionnez les options **Select output based on : sequential** et **If output is blocked : try unblocked output** dans le dialogue du bloc Select Item Out.

Modèle Simple Routing One Queue

Il est possible de construire un modèle semblable au modèle d'orientation simple mais avec une seule file d'attente placée en amont du bloc Select Item Out. Cependant, comme expliqué ci-dessus, si deux entités qui se suivent doivent être traitées par la même machine, elles bloqueront les entités qui arrivent derrière elles. Dans ce cas, sélectionnez l'option **If output is blocked : try unblocked output**.



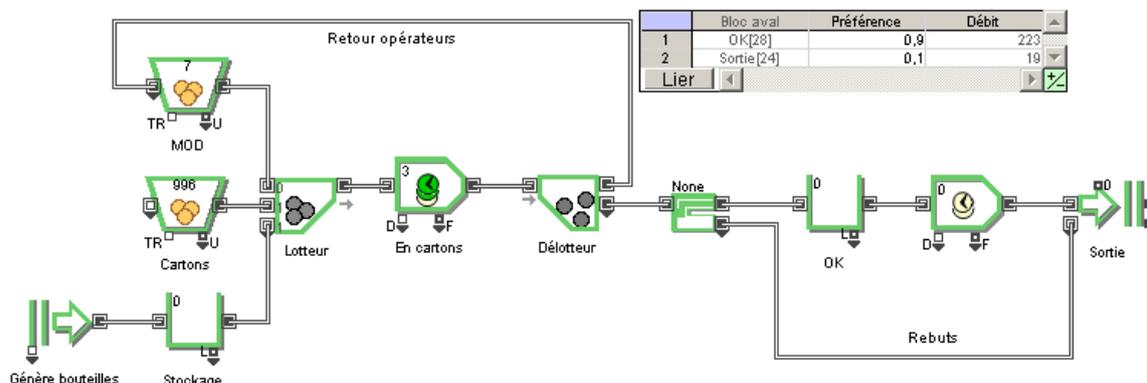
Comme le montre le modèle, même si 50 % des entités sont censées se diriger vers la machine du haut, celles-ci sont réparties de manière quasi égale entre les trois files. Étant donné que toutes les machines ont le même temps de traitement, la machine du haut est souvent occupée, et plutôt que de provoquer un blocage, l'entité qui devait normalement aller vers la machine 1 est orientée vers une autre machine.

Rebut

Dans certains systèmes, modéliser le *rebut* ou simuler un taux de rendement est un aspect important. Par exemple, un grand nombre de procédés de fabrication produisent une quantité variable de déchets ou d'entités défectueuses. ExtendSim permet de modéliser cela en écartant du processus de traitement certaines entités prises au hasard.

Modèle Scrap Generation

Dans ce modèle, un bloc Select Item Out détermine si les entités provenant du bloc Unbatch doivent poursuivre leur traitement ou être mise au rebut. Dans le bloc Select Item Out, l'option **Select output based on : random** est sélectionnée. En définissant une probabilité selon laquelle 0.90 (90 %) des entités sortiront par le connecteur du haut et 0.10 (10 %) par le connecteur du bas (rebut), le bloc Select Item Out mettra une entité sur dix au rebut.



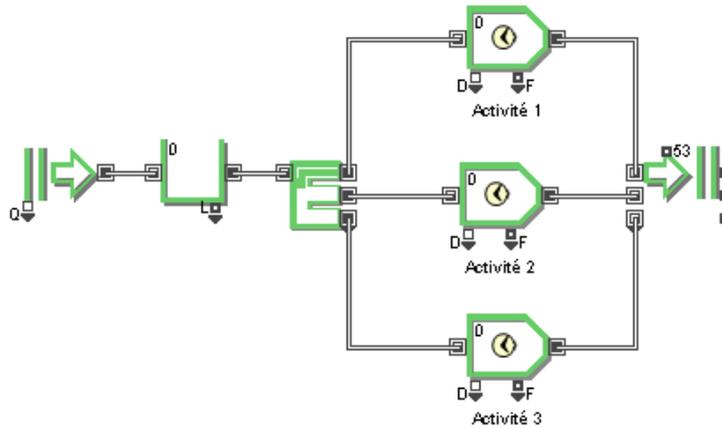
Une autre solution est de donner et lire des attributs pour représenter les entités qui doivent être mises au rebut.

Orientation séquentielle

Pour orienter les entités vers des postes de traitement dans un ordre successif, sans tenir compte du fait qu'une autre activité soit disponible, sélectionnez l'option *Select output based on : sequential* dans le bloc Select Item Out.

Modèle Sequential Ordering

Dans cet exemple, le bloc Select Item Out choisit les sorties dans un ordre successif, en commençant par celle du haut. Dans le dialogue, l'option 'If output is blocked : wait' est sélectionnée.



Dans les deux premiers blocs Activity, le temps de traitement est 1 unité de temps. Dans le troisième bloc Activity, il n'est que de 0,5 unité de temps. Dans ce modèle, même si l'activité 3 est la première à pouvoir accepter une entité, elle n'en acceptera qu'après que les deux premières Activités aient reçu une entité à traiter. Notez également qu'une entité ne sort du bloc Queue que lorsqu'une activité a terminé son traitement, ce qui peut provoquer un blocage du système. Une solution est proposée dans l'exemple «Input Line Balancing».

- ☞ Pour envoyer une entité entrante vers une sortie disponible, sélectionnez *Select output based on : sequential* et paramétrez le bloc en choisissant l'option *If output is blocked : try unblocked output*.

Orientation explicite

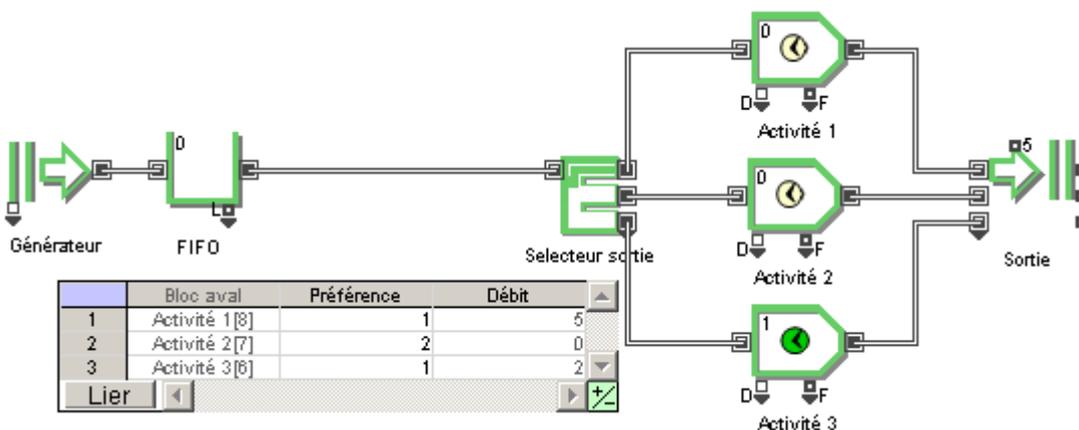
S'il y a plusieurs opérations et que vous voulez que certaines soient plus actives que d'autres, vous pouvez choisir lesquelles sont prioritaires pour les entités. Cela permet d'éviter une opération qui n'est plus effectuée de manière aussi efficace qu'avant (équipement usé ou obsolète), ou parce qu'elle suppose une utilisation plus coûteuse des ressources (lorsqu'un responsable sert les clients).

Vous pouvez spécifier la priorité de chaque sortie en choisissant la condition de sélection *Préférence des sorties* dans le dialogue du bloc Select Item Out.

- ☞ Notez que cela n'est pas la même chose que d'attribuer une priorité à une entité car le bloc Select Item Out donne une priorité au chemin de sortie, pas à l'entité.

Modèle Explicit Ordering

Par exemple, si vous voulez que les Machines 1 et 3 traitent la majorité des entités et que la Machine 2 n'en traite que si les deux autres Machines sont occupées, le modèle sera comme suit :



Dans le dialogue du bloc Select Item Out, l'option *Select output based on : priority* est sélectionnée. Dans le tableau, la plus haute priorité (valeurs les plus basses) est donnée à la sortie du haut et à celle du bas. Une priorité plus basse est attribuée à la sortie du milieu. Dans ce cas, l'activité 1 est prioritaire pour les entités. Si l'activité 1 est occupée, alors c'est l'Activité 3 qui recevra l'entité. Ce n'est que si les activités 1 et 3 sont occupées que l'activité 2 reçoit une entité.

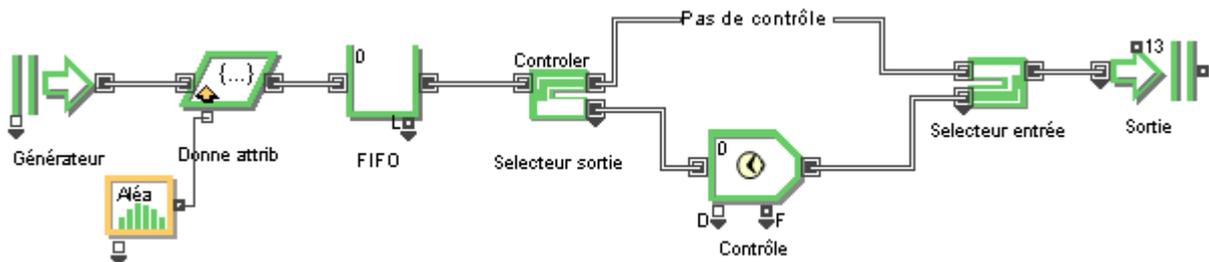
- Comme le montre ce modèle, plusieurs sorties peuvent avoir la même priorité. Néanmoins, l'entité se dirigera vers la sortie la plus en haut ayant la plus haute priorité, et à condition que celle-ci soit disponible. Si ce n'est pas le cas, le bloc regarde si la sortie située en dessous et ayant la même priorité est disponible, etc. Par exemple, vous pouvez attribuer les priorités 1, 2 et 3 au lieu de 1, 2 et 1 comme dans le modèle d'exemple.

Décisions d'orientation basées sur des attributs

Pour orienter des entités en fonction de leurs caractéristiques (taille, qualité, âge, état), vous devez leur donner un attribut dont la valeur sera ensuite lue pour orienter l'entité.

Attributs pour orienter une entité

Pour spécifier si une entité doit être traitée ou pas, donnez une valeur oui-ou-non à l'attribut d'une entité en utilisant le bloc Random Number (bibliothèque Value), comme le montre le modèle Attributes for Routing ci-dessous :



Dans le bloc Random Number, la distribution Empirique spécifie que 75 % des entités n'ont pas besoin d'être contrôlées (valeur 0 pour l'attribut Controler) et que 25 % en ont besoin (1 pour la valeur d'attribut). Le bloc Select Item Out, avec l'option *Select output based on : property*, lit la valeur d'attribut pour déterminer le chemin à emprunter par l'entité, laquelle passera soit par la ligne de contrôle (valeur 1) soit par l'autre ligne (valeur 0). Le bloc Select Item In permet de faire converger les deux lignes en un seul flux, qui sort de la simulation.

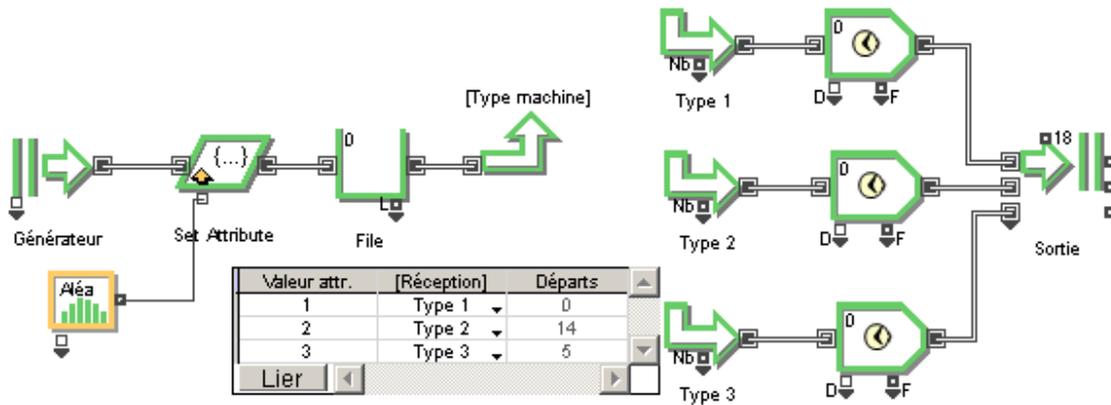
Cette méthode est particulièrement utile si le processus de contrôle se fait en plusieurs étapes. Par exemple, si un contrôle est nécessaire, vous aurez sûrement besoin de transporter une entité vers le poste de contrôle avec des blocs Transport. Toutes ces étapes se situeront entre les blocs Select Item Out et Select Item In.

Dans ce modèle, une entité devant être contrôlée peut entrer dans le bloc Select Item Out et être retenue ici car une autre entité se trouve déjà dans l'activité. Pour éviter cela, vous pouvez demander au bloc de prévoir à l'avance le chemin à emprunter par l'entité avant que celle-ci n'entre dans le bloc.

- Une entité devant être vérifiée et qui se trouve bloquée dans le bloc Select Item Out va également bloquer les entités suivantes, même si celles-ci ne doivent pas être vérifiées. Si votre process ne fonctionne pas de cette manière, ajoutez un bloc Queue avant l'activité pour retenir les entités qui doivent être vérifiées.

Modèle Throw & Catch Attributes

Comme il est expliqué [plus haut](#), le bloc Throw peut être utilisé pour orienter des entités vers un bloc Catch spécifique, identifié par son label. Les blocs Throw permettent également d'orienter des entités vers différents blocs Catch en fonction de la valeur d'un attribut. Dans la figure ci-dessous, le modèle d'exemple « Attributes for Routing » a été modifié à l'aide des blocs Throw et Catch, qui remplacent le bloc Select Item Out :

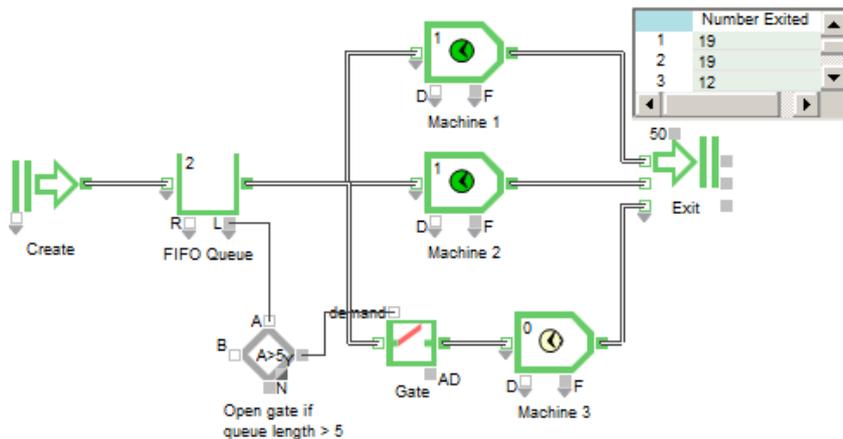


Dans cet exemple, dans le dialogue du bloc Throw, l'option *Specify Catch Item by : Type machine* est sélectionnée. Le bloc Throw lit l'attribut Type machine et oriente l'entité vers le bloc Catch correspondant d'après la table qui se trouve dans le dialogue du bloc, et qui est clonée sur la feuille de travail.

Pour associer une valeur d'attribut à un bloc Catch spécifique, entrez la valeur dans la colonne 'Attribute Value' et sélectionnez le bloc Catch correspondant dans le menu déroulant de la colonne 'Catch block'.

Orientation conditionnelle

Il se peut que vous ayez besoin d'orienter les entités en fonction des conditions actuelles du modèle. Par exemple, surveiller la longueur des files d'attente pour déterminer si oui ou non une activité sera intégrée au système ou équilibrera l'utilisation de files d'attente parallèles.



Modèle Conditional Routing

Par exemple, il est possible que dans une usine, la majorité du traitement soit effectué par deux machines et que le travail supplémentaire soit pris en charge par une troisième machine. Cette situation peut être modélisée facilement avec ExtendSim, en utilisant le bloc Decision (bibliothèque Value) et le bloc Activity (bibliothèque Item).

La sortie L d'une file qui envoie des entités vers une ou plusieurs machines émet le nombre d'entités en attente de traitement. Si cette valeur dépasse un certain seuil, vous pouvez orienter une partie des entités vers une autre machine ou lancer un autre process.

Dans le modèle, le dialogue du bloc Decision spécifie que le connecteur O sort une valeur True (1) lorsque la valeur lue sur l'entrée A est supérieur à 5. Cela active le connecteur *demand* du bloc Gate, qui laisse alors passer des entités vers la troisième machine (qui n'acceptait pas d'entités jusqu'alors). Lorsque la File contient 5 entités ou moins, la Porte se ferme.

Vous pouvez, au contraire, modéliser cette situation de façon à ce que tous les blocs d'opération traitent des entités, puis en mettre une ou plusieurs à l'arrêt sous certaines conditions. Dans ce cas, des entités peuvent restées coincées dans l'opération arrêtée jusqu'à ce que le traitement reprenne.

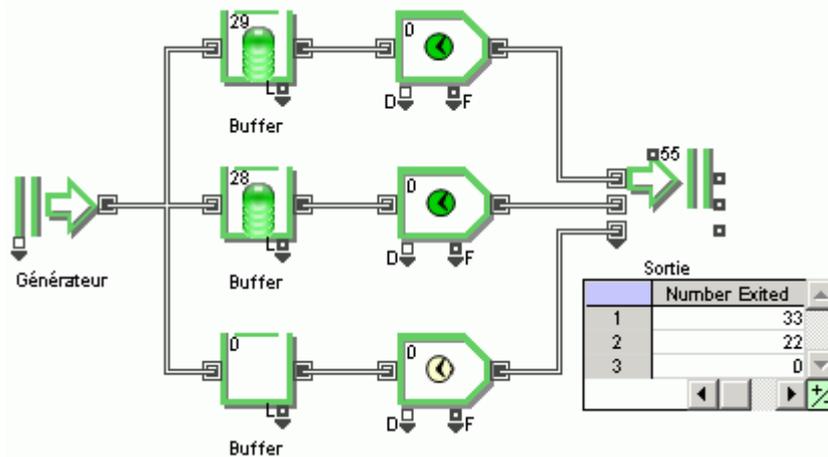
Equilibre des lignes

Les opérations sont souvent précédées de files d'attente, comme une zone d'entreposage pour chaque machine. La position et le classement des entités dans la file peuvent avoir une incidence sur le

fonctionnement du modèle.

Modèle Buffering Operations

Voici un exemple de modèle dans lequel chaque opération est précédée d'une file d'attente :



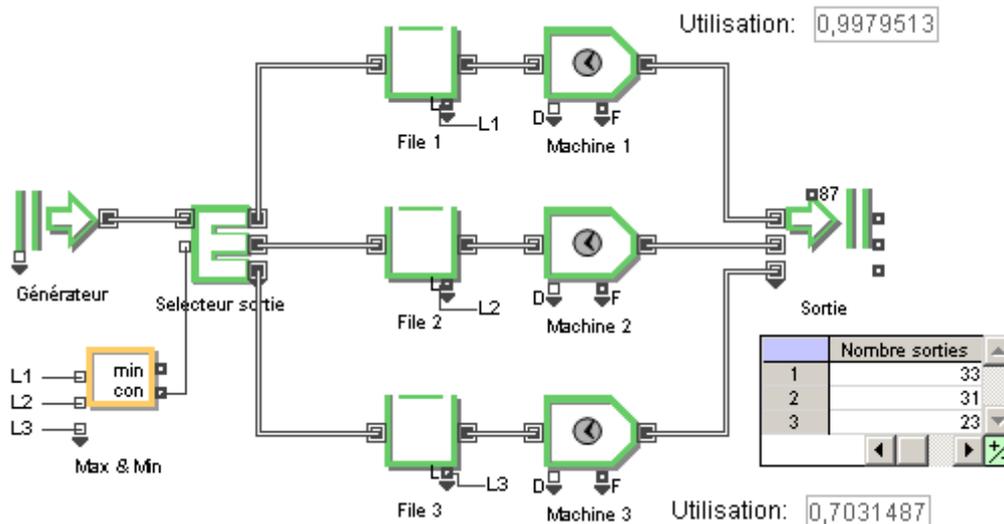
Dans le modèle, chaque file peut contenir 30 entités au maximum. Toutes les entités se dirigeront vers la première file à avoir été reliée jusqu'à ce que celle-ci soit pleine, puis la file suivante commencera à se remplir (sauf si la première machine parvient à absorber le flux d'entité, auquel cas la file suivante ne recevra jamais d'entités). Par exemple, dans ce modèle, la majorité des entités entrent dans la file du haut et sont traitées par la Machine 1, et aucune entité ne se dirige vers la file du bas pour être traitées par la Machine 3. En général, ce n'est pas le comportement souhaité.

Pour mieux répartir l'utilisation des machines, utilisez un bloc Select Item Out paramétré pour choisir les Machines de façon séquentielle et placez une file devant chaque machine. Cependant, notez que si les machines fonctionnent à des vitesses différentes, la file de la machine la plus lente se remplira plus rapidement que les autres.

- ☞ Si une barre verte apparaît en haut de la file, cela signifie que sa capacité est limitée. Par exemple, dans ce modèle, les Files sont paramétrées pour retenir 30 entités au maximum. C'est pourquoi une barre verte apparaît sur chaque Queue.

Modèle Output Line Balance

Une meilleure méthode que les opérations avec buffer serait de vérifier la longueur de chaque file d'attente et d'envoyer l'entité suivante dans la file la moins longue, ce qui équilibrerait les files d'attente. Le bloc Max & Min (bibliothèque Value) convient très bien pour cela.



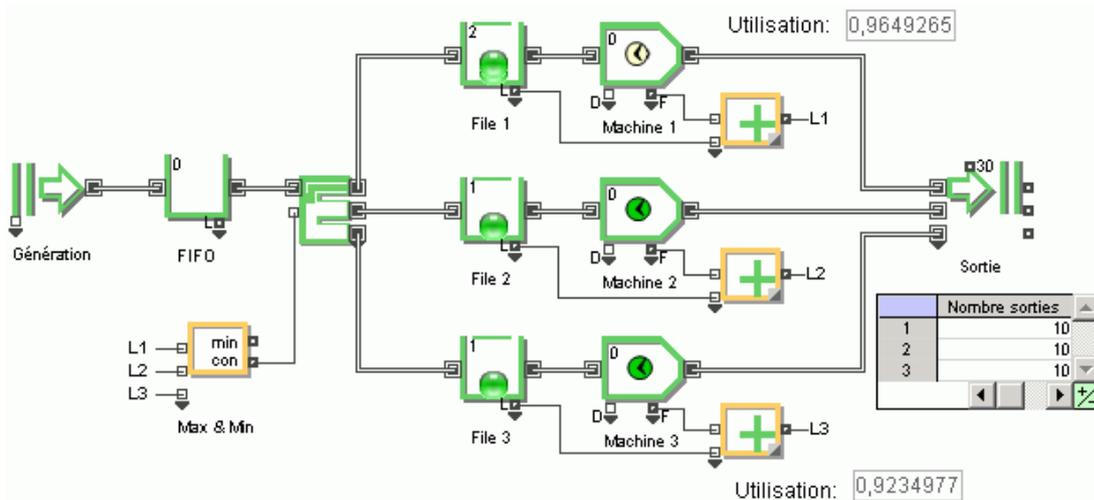
Dans ce modèle, le bloc Max & Min indique au bloc Select Item Out la file la moins longue, laquelle pourra donc accueillir la prochaine entité. Dans le bloc Max & Min, les options *Output the : minimum value* et *Top input connector # starts at : 1* sont sélectionnées. Grâce à ces paramètres, l'entrée du haut (L1) du bloc porte le numéro 1 et la sortie **con** indique laquelle des sorties (L1, L2 ou L3) a la plus petite valeur, en indiquant la file la moins longue. Dans le dialogue du bloc Select Item Out, les options choisies sont *Select output based on : Top output chosen by select value : 1*.

Contrairement au modèle d'opérations avec buffer, dans ce modèle, le nombre d'entités présentes dans chaque file est généralement plus équilibré. Cependant ce système n'est pas aussi efficace qu'il pourrait l'être car une entité se dirige souvent vers la file de la Machine 1, même si la Machine 3 est au repos. Cela est dû au fait que la File 1 est prioritaire pour les entités si les trois files ont la même longueur, or le bloc Max & Min ne regarde que la longueur des files, et non si l'Activité est occupée ou pas.

- La File 1 étant la première à avoir été liée dans le modèle, elle sera choisie en priorité par les entités si toutes les files ont la même longueur.

Modèle Line Balance with Activities

Le modèle précédent montrait comment équilibrer les files d'attente. Il serait plus utile d'inclure des informations sur les entités traitées au moment de choisir la file la moins longue. Pour cela, il suffit d'ajouter la valeur du connecteur F d'un bloc Activity à la longueur de la file. Le connecteur F émet 1 quand l'activité est occupée et 0 sinon. Ainsi, le modèle empêche les entités d'aller dans une file suivie par une activité occupée quand d'autres activités sont disponibles.



Comme dans le modèle précédent, les files d'attente ont tendance à être plus équilibrées que dans le modèle avec buffer. Aussi, cette méthode permet une meilleure utilisation des machines.

- ☞ Une autre possibilité serait de remplacer les blocs Queue et Activity par un bloc Workstation. Son connecteur Longueur indique en effet toutes les entités présentes dans sa file et son activité interne.

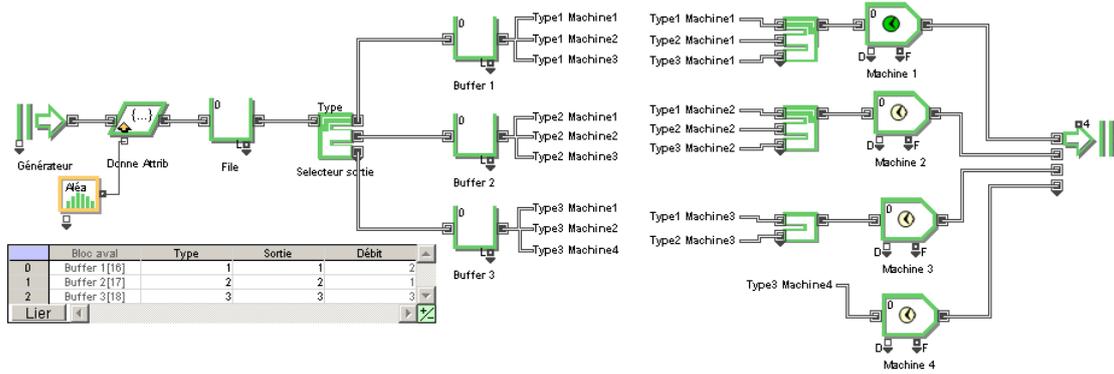
Machines ne pouvant traiter que certains types d'entités

Généralement, une chaîne de montage peut manipuler plusieurs types d'entités en même temps. Par exemple, il est possible d'assembler trois modèles différents sur une même chaîne. L'assemblage est quasiment identique, seuls quelques composants diffèrent à certains endroits. Malheureusement, sur une chaîne de montage aussi hétérogène, certaines machines ne fonctionnent pas sur certains modèles. Pour pouvoir le faire, la méthode consiste à décomposer les entités, à faire passer leurs composants par des chemins différents afin d'établir quel sont les différents « types » d'entités, puis à faire converger les chemins de façon appropriée pour les différentes machines. Dans de telles situations, les blocs Select Item Out et Select Item In sont utiles.

Modèle Processing by Type

Prenons comme exemple une chaîne de montage sur laquelle chaque entité possède un attribut *Type* de valeur 1, 2 ou 3 en fonction du type de l'entité. À une étape du processus d'assemblage, il y a quatre machines. Deux d'entre elles peuvent traiter les trois types d'entités ; la troisième, moins récente, ne peut traiter que les types 1 et 2 et la quatrième ne peut traiter que le type 3.

Un bloc Set donne un attribut *Type* à chaque entité. Dans le bloc Random Number (bibliothèque Value), la table empirique indique que la probabilité qu'une entité soit du type 2 est de 50 %, et 25 % pour les types 1 et 3.



Dans le bloc Select Item Out, l'option choisie est *Select output based on : property*. Le bloc regarde la valeur de l'attribut Type (1, 2 ou 3) et sélectionne la sortie correspondante (1, 2 ou 3) en fonction des entrées dans la table d'options du bloc.

Notez que les files sont utilisées comme buffers dans le modèle ci-dessus. Elles permettent de stocker les entités par type, avec la file du haut pour le Type 1, etc. Sans les files, toute la chaîne pourrait être bloquée, selon l'ordre d'arrivée des entités. Par exemple, si les trois première machines sont toutes en train de traiter une entité du Type 1, et que l'entité suivante est aussi du Type 1, un blocage se produit jusqu'à ce que l'une des machines ait terminé avec son entité. La quatrième machine ne peut pas accepter d'entité avant qu'une des autres machines ait fini son traitement et ne reçoive une nouvelle entité de Type 1. Mais même dans ce cas, elle devra attendre une entité du Type 3 pour pouvoir effectuer un quelconque traitement.

- Les liens nommés permettent de rendre le modèle plus facile à lire. Sans ces liens, il y aurait un trop grand nombre de liens reliant les blocs Queue aux blocs Select Item In. Une autre manière d'organiser le modèle serait d'utiliser les blocs Throw et Catch pour orienter les entités vers les bonnes machines.

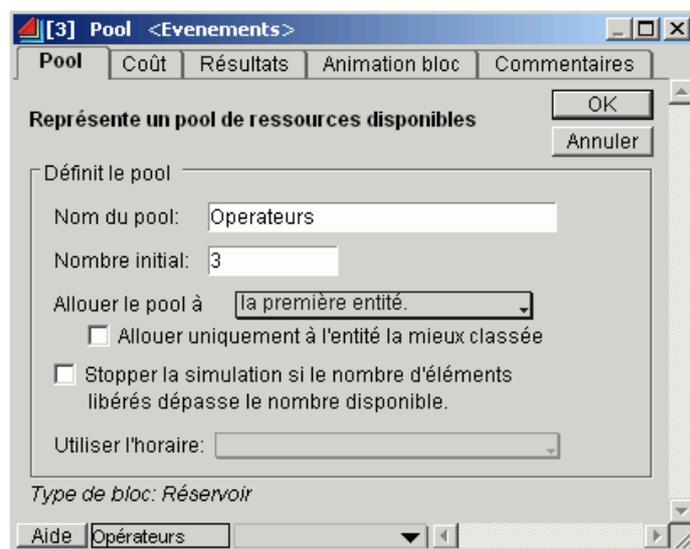
Ressources

Trois manières de modéliser des ressources

Une ressource est un stock fini d'éléments, tels que des outils ou de la main d'œuvre, que l'on utilise en production. Les ressources sont limitées ; vous pouvez spécifier combien d'éléments de telle ressource sont disponibles à un moment donné. Les ressources peuvent être réutilisées si vous le souhaitez. C'est le cas général avec la main d'œuvre, où les personnes sont employées avec d'autres ressources sur certaines tâches, puis redeviennent disponibles pour d'autres tâches.

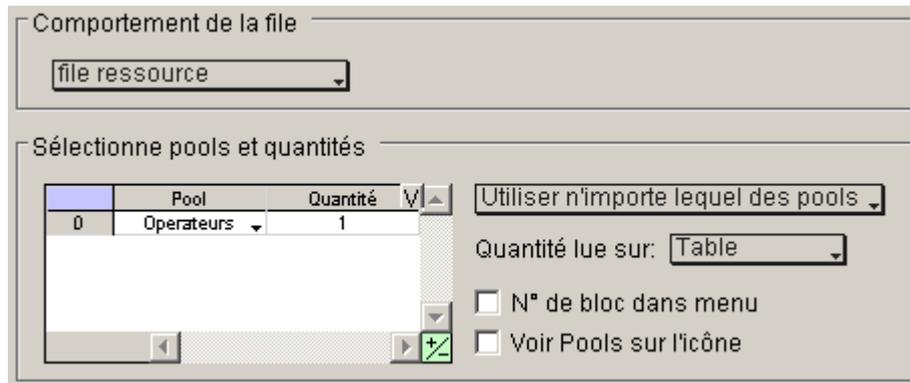
Il existe trois manières de modéliser des ressources dans ExtendSim : Les ressources peuvent être **représentées par des entités**. Celles-ci peuvent exister en stock, être générées au hasard ou selon un planning, comme c'est le cas dans une usine réelle. Par exemple, vous pouvez lier un bloc Create à un bloc Resource Item pour simuler l'arrivée aléatoire de pièces dans un atelier. Si les pièces arrivent à intervalles réguliers, vous pouvez lier le bloc Lookup Table au connecteur TR du bloc. Les ressources seront ensuite mises en lot avec les entités qui les utilisent, au moyen des divers blocs Batch. Vous utilisez pour tout cela le bloc Resource Item de la bibliothèque Item.

Les ressources peuvent aussi être représentées par trois blocs associés (Resource Pool, Queue en mode Ressource et Resource Pool Release). À la différence du bloc Resource Item, les **éléments d'un pool** ne sont pas des entités. Il s'agit de variables qui indiquent l'aspect quantitatif d'un facteur limitant. Le **bloc Resource Pool** s'emploie avec le bloc Queue pour allouer les éléments du pool à des entités, et avec le bloc *Resource Pool Release* pour libérer les éléments du pool. Des entités peuvent attendre pour les éléments d'un pool depuis n'importe quel nombre de blocs Queue. Le bloc Resource Pool détermine comment les éléments sont alloués aux entités : soit l'entité qui est arrivée la première dans le bloc Queue, soit l'entité ayant la plus haute priorité (plus faible valeur numérique de priorité).

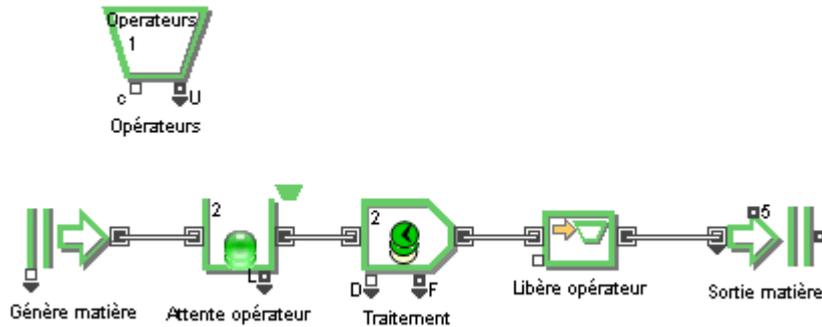


Dialogue d'un bloc Resource Pool

Cette méthode a l'avantage de ne requérir aucun lien, et pour cela s'avère spécialement adaptée lorsqu'une même ressource peut être utilisée à plusieurs endroits. À l'inverse, une même Queue en mode Ressource peut faire appel à des éléments provenant de différents Pools, soit parce que ces éléments sont équivalents et donc interchangeables, soit parce qu'ils sont tous requis.



Dialogue d'un bloc Queue en mode Ressource



Modèle utilisant des Pools

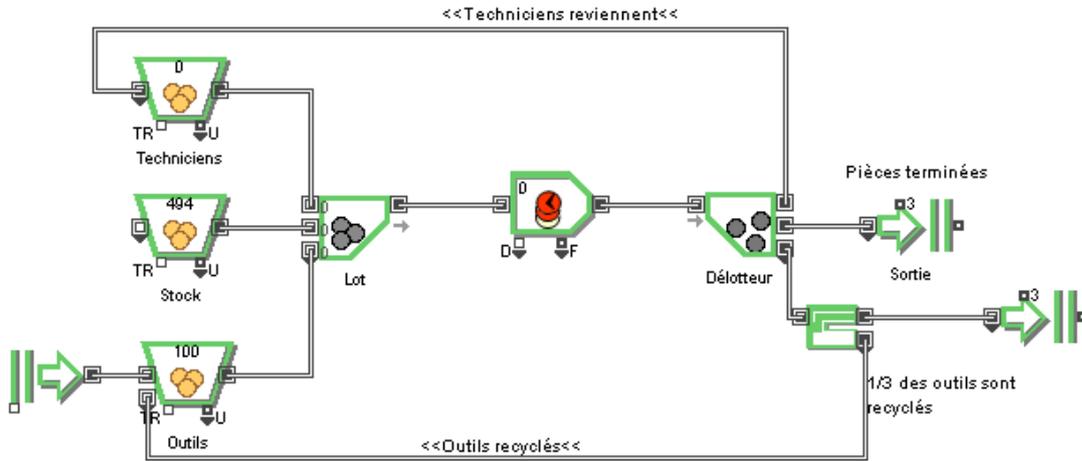
Les ressources peuvent enfin être représentées via la fonctionnalité de « gestion de ressources complexes » qui s'appuie sur le bloc Resource Manager. Cette méthode fait appel aux enregistrements dans une base de données, et où chaque enregistrement représente une ressource avec des propriétés et statistiques propres. Le bloc Resource Manager fournit l'interface de gestion de ressources complexes. Ce système de gestion est un outil évolué à utiliser uniquement lorsque les ressources sont effectivement complexes à gérer et au cœur de la problématique de modélisation. Elle est examinée dans un chapitre à part.

Systemes fermés et ouverts

Les blocs de ressources peuvent s'employer dans des systèmes fermés ou ouverts, selon que les entités sont ramenées ou non dans leur bloc d'origine.

Dans un système fermé, les entités sont sorties du bloc puis recyclées sur le connecteur d'entrée du bloc une fois qu'elles ont servi ailleurs dans le modèle. Par exemple, vous pouvez utiliser un bloc Appareil pour fournir certains appareils, les mettre en lot avec certaines pièces à traiter, puis les délotter afin de rendre les appareils à nouveau disponibles à l'emploi.

Les blocs ressources font partie d'un système ouvert lorsque les entités du bloc ne sont pas recyclées. En fin de traitement, les entités ne sont pas ramenées vers le connecteur d'entrée du bloc. L'exemple le plus évident est celui du stock, dont les entités quittent le système de production en fin de traitement. Un autre exemple est celui des ressources consommables, tel un appareil jetable qui ne sert qu'une fois.

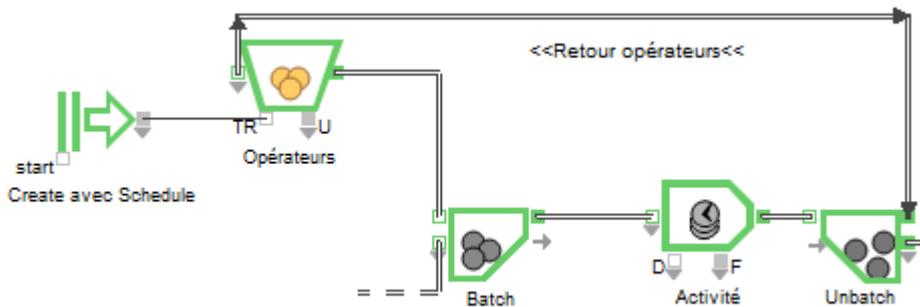


Modèle avec ressources en système ouvert et fermé : modèle Closed and Open Systems

Dans un système fermé, une entité qui revient à sa ressource peut comporter des attributs acquis au cours de son traitement mais qui ne s'appliquent plus à un nouvel élément. Par défaut les blocs de ressource effacent les attributs des éléments en entrée.

Planification des ressources

Le bloc Resource Item a un connecteur d'entrée continu nommé TR., Total resources. Vous utilisez ce connecteur pour indiquer dynamiquement combien d'éléments sont disponibles dans la ressource.



Planning appliqué à une ressource : modèle Scheduling Resources

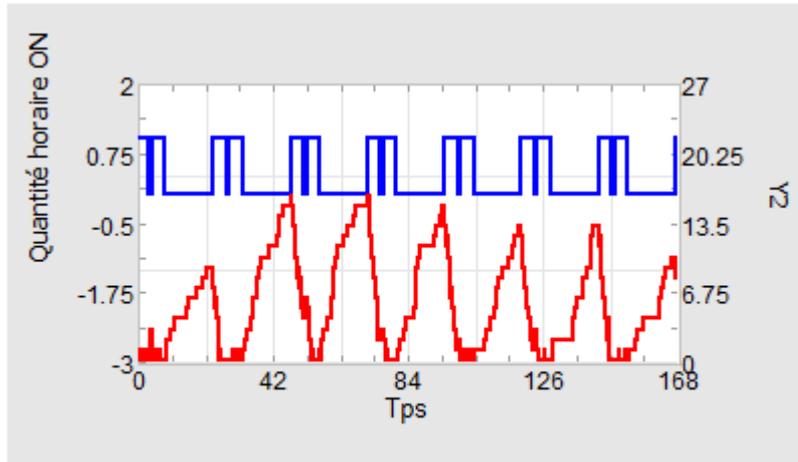
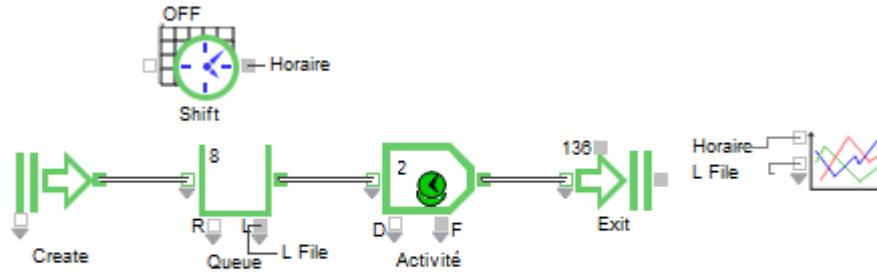
Utiliser le bloc Shift

Le bloc Shift s'emploie pour planifier le nombre et la disponibilité de certains éléments (blocs) d'un modèle, notamment des ressources. Typiquement, il permettra de représenter un planning de main d'œuvre, avec des opérateurs qui arrivent et partent, font des pauses, etc.

L'horaire peut être de type ON/OFF ou représenté par un nombre. Dans le cas ON/OFF, les blocs utilisant cet horaire seront en inactivité (suspens) lorsque l'horaire est inactif, et en fonctionnement normal lorsque l'horaire est actif. Lorsque l'horaire est numérique, les blocs utilisant cet horaire modifieront leur capacité en fonction de la valeur courante de l'horaire (qui figure alors une équipe de travail). Seuls les blocs de capacité multiple (Activity, Resource Item) peuvent utiliser l'horaire numérique. Un bloc Shift peut contrôler les horaires de travail de n'importe quel nombre de blocs dans un modèle. Si l'horaire est modifié dans le bloc Shift, tous les blocs utilisant cet horaire reflèteront la modification.

Sa case à cocher "Repeat schedule every" permet par exemple d'affecter les mêmes horaires quotidiens de travail à un groupe de machines ou d'opérations.

Le connecteur d'entrée du bloc permet de prévaloir sur les paramètres du dialogue. S'il reçoit une valeur inférieure à 0.5, le bloc passera en horaire inactif. Si la valeur est supérieure à 0.5, l'horaire décrit dans le dialogue du bloc sera appliqué.



Shift On and Off.mox

Dans l'exemple, un horaire On/Off sert à contrôler les plages actives du bloc Activity.

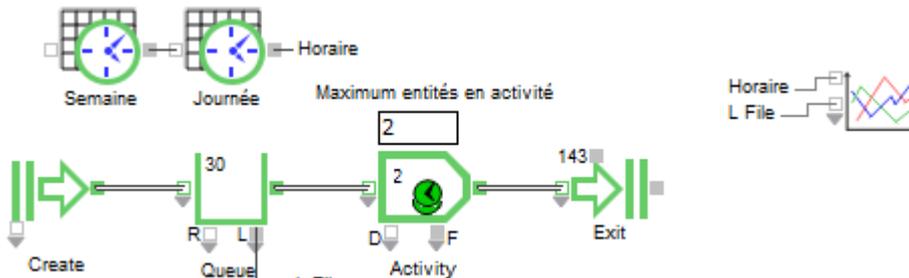
Shift name:

Shift type: Time unit:

	Time	On/off	Description
0	0	On	Work
1	4	Off	lunch
2	5	On	Work
3	9	Off	Overnight

Repeat schedule every:

Dialogue du bloc Shift



Weekly and Daily Shifts.mox

Dans cet exemple, un premier bloc Shift modélise une semaine de 40 heures typique avec 2 jours de week-end. Il agit sur un autre bloc Shift qui modélise un planning journalier qui est directement utilisé par des activités. L'horaire du second bloc est neutralisé pendant les week-ends, c'est-à-dire quand le premier lui envoie des valeurs zéro.

NOTE IMPORTANTE:

Il est assez facile de définir un horaire complexe incluant les pauses, week-ends, vacances, etc. Mais avant d'implémenter une telle complexité dans un modèle, il faut être sûr que ces détails sont pertinents. Par exemple, si rien ne se passe durant les week-ends, il est préférable de créer une semaine de 5 jours, plutôt que d'ajouter un bloc Shift et de définir une inactivité durant les week-ends.

Lottage/délottage

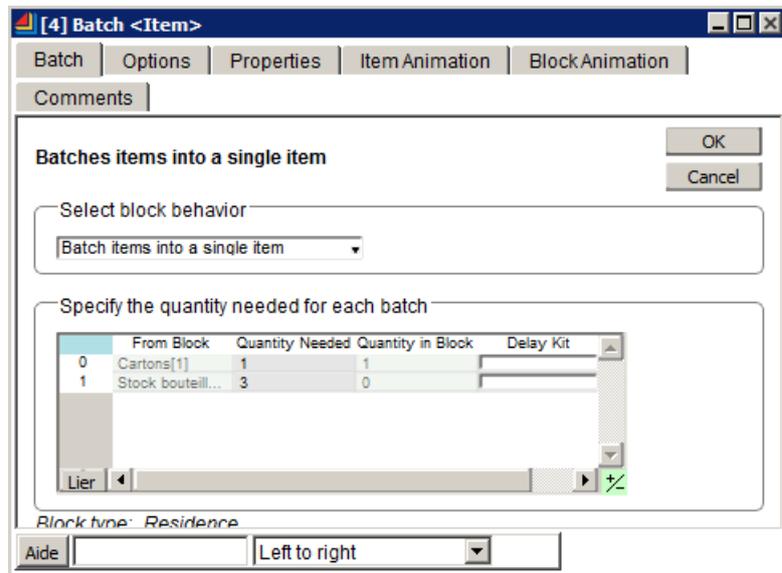
Dans tout système de production, les entités transitent par l'usine et sont traitées. Au début les composants du produit final sont amenés : matières premières, conditionnement, main d'œuvre. Ces éléments sont passés en traitement à certains moments et selon certaines quantités. Ces composants initiaux sont souvent associés à d'autres, et progressent ensemble dans le traitement. Par exemple, dans une chaîne de montage de cartes électroniques, la carte vide et les puces sont très tôt mises en lot.

NB : les blocs Resource Pool, Queue et Resource Pool Release de la bibliothèque Item proposent une autre manière de représenter les ressources et de les associer aux entités qui en ont besoin.

La mise en lot

La mise en lot est la réunion de plusieurs entités en une seule entité, pour les besoins de la simulation (traitement, acheminement, etc.). Le bloc Batch accumule les entités jusqu'à un niveau requis, puis sort une seule entité représentant le lot.

La mise en lot s'utilise pour deux tâches légèrement différentes, qu'on pourrait appeler assemblage et rendez-vous. L'assemblage implique la réunion physique des composants, alors que le rendez-vous fait référence à la disponibilité simultanée de plusieurs entités. Dans le second cas, bien souvent les entités ne restent pas longtemps sous forme de lot.



dialogue d'un bloc Batch

Lors d'une mise en lot, vous pouvez spécifier que les entités sur l'une ou plusieurs des sources ne seront pas amenées au bloc Batch tant que les autres sources n'auront pas rempli leur quota requis. Utilisez cette méthode lorsque l'un des composants est une ressource limitée, par exemple un opérateur, que vous ne voulez pas immobiliser en attendant la disponibilité des autres ressources. C'est en cochant ou non la case "Delay Kit" que vous déterminez si l'entité attendra la constitution du lot à l'intérieur ou hors du bloc Batch. Au moins l'une des sources utilisées doit être accumulée dans le bloc.

Select options

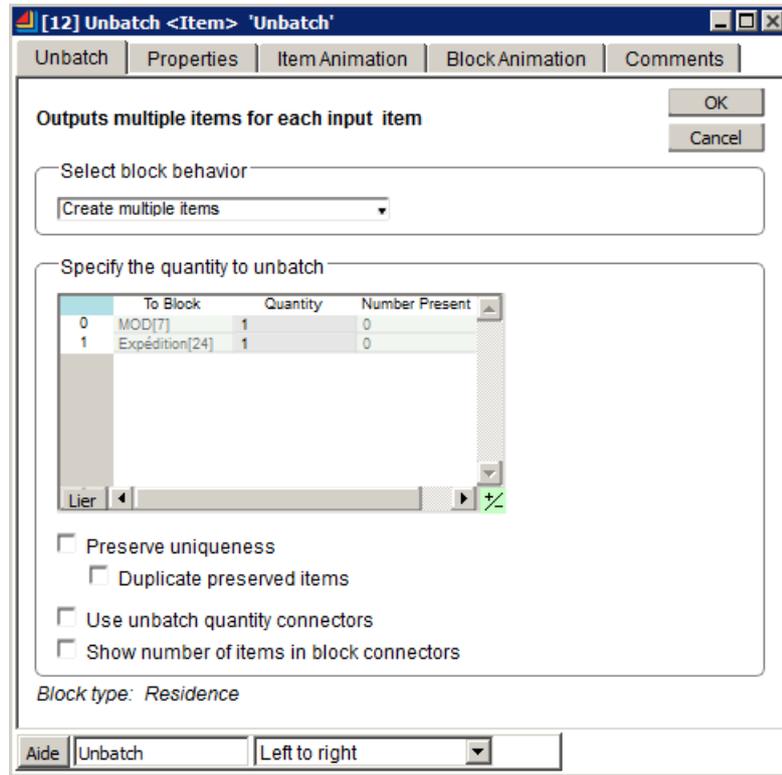
- Preserve uniqueness
- Use quantity input connectors.
Set batch size:
- Show connectors for item count
- Allow zero batch size
- Show demand connector
When value at demand ≥ 0.5 :
- Store number of items in batch in attribute:

Lorsque des entités sont mises en lot, vous choisissez dans l'onglet Properties comment sont combinés attributs et priorité pour qualifier le au lot sortant. Mais vous pouvez demander à ce que les attributs provenant de telle ou telle source ne soient pas conservés dans l'entité résultante. Vous pouvez à l'inverse conserver **l'individualité** (Preserve uniqueness) de chaque composant, pour qu'ils retrouvent leur intégralité lors du délottage, comme expliqué plus loin.

Les blocs Batch et Unbatch peuvent faire apparaître un connecteur *demande* permettant de contrôler le fonctionnement du bloc. Il s'agit d'un connecteur continu.

Le délottage

Le délottage est généralement utilisé pour séparer des entités qui ont été momentanément associées, ou pour cloner des entités.



dialogue du bloc Unbatch

- Dans certains cas, la mise en lot n'est que temporaire : c'est le cas du rendez-vous, expliqué plus haut. Par exemple, vous prenez une palette pour transporter des entités, puis ramenez la palette à son lieu d'origine à la fin du transport. C'est typiquement ce qui est effectué avec la main d'œuvre, que l'on associe aux éléments traités pendant toute la durée prévue pour les opérations, avant de lui rendre sa disponibilité pour une nouvelle tâche.
- Le délottage peut aussi être utile s'il n'y a pas eu de mise en lot. Par exemple, si un camion livre une boîte de vis sous forme d'un seul élément. Un bloc Unbatch pourra servir à cloner 100 fois l'entité pour alimenter un stock. Les mêmes attributs seront affectés à toutes les vis.

Conserver l'individualité des entités

Délotter une entité c'est en faire des copies qui ont toutes les attributs de l'entité initiale. Ainsi, si une boîte de vis est constituée à partir de nombreuses vis, lorsqu'elle sera délottée, les vis auront des attributs de taille et de boîte. À ce moment, lorsque vous considérez les vis individuelles, vous prenez en compte l'attribut de taille et vous ignorez celui de boîte.

Mais s'il vous importe de ne pas combiner des attributs ou des priorités pour constituer un lot que vous savez devoir éclater ensuite, sélectionnez l'option **Preserve uniqueness** dans l'onglet Options du bloc Batch et dans le bloc Unbatch. Cette option marque de manière unique les caractéristiques de chaque entité, afin de les rétablir lors du délottage. Cette option utilise beaucoup de mémoire.

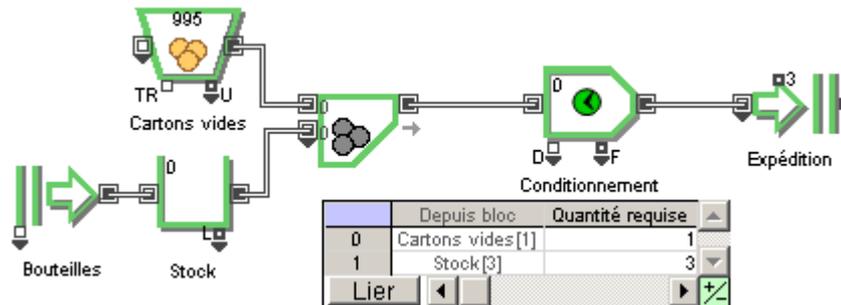
Si vous choisissez de conserver l'individualité des entités mises en lot, vous devez être prudent si vous employez des blocs d'affectation d'attributs ou de priorités dans la portion du modèle entre Lottage et

Délottage. Toute modification de propriété effectuée sur cette portion sera oubliée au moment du délottage, car les entités sont délottées exactement dans l'état où elles ont été mises en lot. Notez aussi que l'individualité des entités ne sera restaurée que lors du délottage. Dans la portion entre Lottage et Délottage, les attributs sont combinés selon vos indications dans l'onglet Priorité.

Réunir des entités pour un traitement

Mise en lot simple

L'exemple de mise en lot qui suit réunit un cartons avec trois bouteilles ; le résultat de cet assemblage circule ensuite dans le modèle sous la forme d'une unique entité :



exemple de lot simple

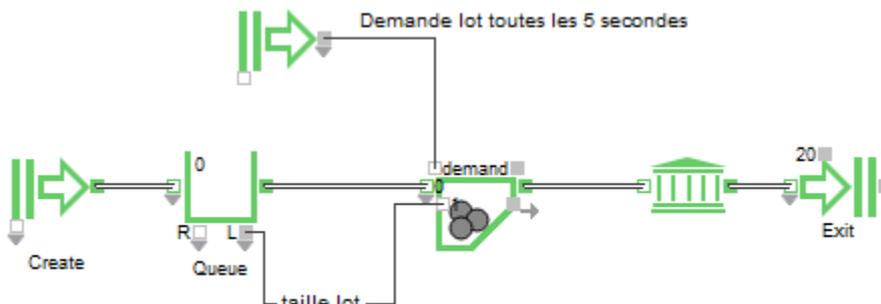
Dans le dialogue du bloc Batch, on spécifie les quantités nécessaires en A et en B. Notez que le bloc Batch ne libère une entité en sortie que lorsqu'il a reçu une entité de la source du haut et trois entités de la source du bas.

L'entité résultante est traitée par le bloc Activity pendant la durée nécessaire à l'assemblage (le bloc Batch n'implique pas de délai), puis quitte le modèle.

Mise en lot d'entités identiques

Certains traitements demandent à ce que plusieurs entités semblables soient réunies : c'est typiquement le cas de pièces terminées qui sont conditionnées dans une boîte. Vous pouvez utiliser pour cela les options plus fines de l'onglet Options.

Si vous demandez à utiliser le connecteur *demand*, le bloc Batch rassemble des entités tant que le signal du connecteur *demand* n'indique pas que le lot doit être libéré. Le lot est ainsi constitué du nombre d'entités qui a été accumulé jusque-là. Vous utiliserez cette méthode lorsque les lots sont constitués en fonction de l'heure ou lorsque des conditions externes déterminent combien d'entités doivent constituer un lot. Par exemple, si vous remplissez des camions avec des caisses, vous pouvez demander l'arrêt du lottage lorsque la journée se termine ou lorsqu'un autre camion se présente sur le quai.



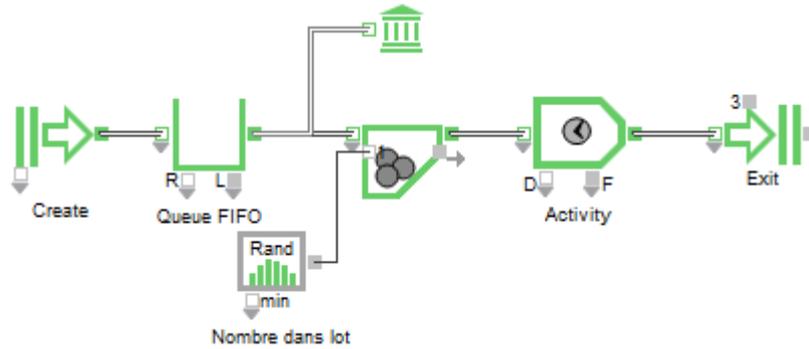
Note: la taille du lot est égale au nombre d'entités dans la file à cet instant.

Batch on Demand.mox

Dans cet exemple, le bloc Create envoie des demandes de lots selon un planning horaire.

Pour contrôler un nombre variable d'entités mises en lot, utilisez la case à cocher **Use quantity item connectors** (Utiliser les connecteurs pour la taille), où la taille du lot est déterminée par le connecteur continu accolé à chaque entrée discrète. Par exemple, si les entités sont mises dans des caisses de taille différente selon le camion qui les expédie, le nombre d'entités contenues dans une caisse (= taille du

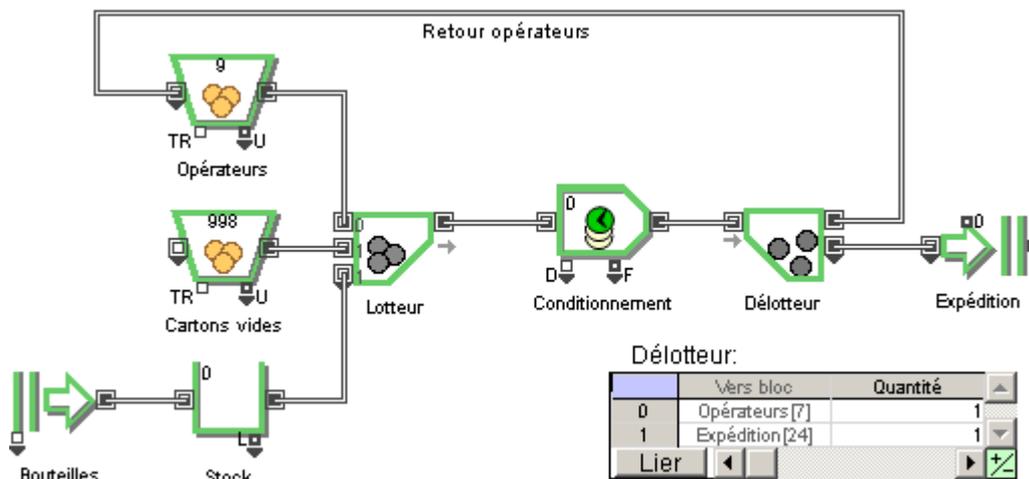
lot) pourra être connu par un calcul. Notez que si vous liez le connecteur *demande* de ce bloc, les entités ne sont aspirées que lorsque ce connecteur est activé (par une valeur continue *vrai*).



mise en lot de taille variable (Batching Variable)

Association et délottage

Nous avons vu plus haut que souvent des entités sont associées de manière temporaire pour un traitement, puis séparées et individualisées. En voici un exemple :



lottage/délottage simple

Avant d'être conditionnées, les cartons sont préparés par un opérateur, représenté par le bloc Resource Item. Il n'y a pas "assemblage" entre les cartons et l'opérateur, mais une association temporaire effectuée par le bloc Batch.

Dans le dialogue, l'option **Delay Kit** indique que l'opérateur ne sera pas demandé par le bloc tant que les autres entités nécessaires ne seront pas présentes. Cela évite de réquisitionner l'opérateur avant qu'il ne soit vraiment utile. Chaque opérateur effectue sa tâche (bloc Activity) de mise en cartons, et lorsqu'il a terminé, place les cartons pour expédition puis retourne pour effectuer une autre tâche. Ici le bloc Unbatch sort deux entités à partir d'une seule : l'une représente l'opérateur qui est ramené vers sa ressource, l'autre représente les cartons qui partent.

Tant que vous n'utilisez pas le calcul des coûts ni l'option *Preserve uniqueness*, vous pouvez très bien mettre en lot une entité depuis la source A et la délotter par la sortie B.

Calcul de coûts

Calcul de coûts

Plusieurs types de blocs permettent d'intégrer automatiquement des calculs de coût dans votre modèle. Dès qu'une information de coût est renseignée dans un bloc, le calcul s'effectue automatiquement. Vous définissez deux types d'information de coût :

- le coût à l'utilisation, ou coût par entité, qui est un coût fixe.
- le coût par unité de temps, qui est un coût variable.

Les entités que l'on définit comme étant des accumulateurs de coût possèdent un taux pour le coût fixe et un taux pour le coût variable. Dans leur progression dans le modèle, elles accumulent des coûts en fonction de ces taux et des blocs par lesquelles elles passent. Vous spécifiez ces taux dans le bloc où est créée l'entité. Le coût fixe est typiquement le coût direct du produit, tandis que le coût variable intervient lorsque l'entité est stockée ou en attente de traitement. Voici un exemple avec le bloc Create :

Coûts dans onglet Options du Create

Le Waiting Cost (coût d'attente) est le taux d'accumulation de coût lorsque l'entité est en attente dans une file. Dans cet exemple, cela coûte 0,15 euros de l'heure de faire attendre l'entité, à laquelle une valeur initiale de 500 euros est donnée.

Ces informations peuvent également être données par le bloc Resource Item. Vous devez alors spécifier dans le dialogue que les entités créées par le bloc sont des accumulateurs de coût, par exemple :

Entités qui accumulent des coûts

Dans les **blocs de ressource**, le taux Cost / time unit (Coût par unité de temps) sert à calculer et ajouter un coût (variable) à tout accumulateur de coût durant le temps où il est associé à la ressource, tandis que le Cost per use (coût à l'utilisation) est le coût (fixe) ajouté à l'accumulateur de coût dès qu'il utilise cette ressource.

Si vos accumulateurs de coûts sont générés par un Create en mode programme, vous devez explicitement définir les attributs correspondant aux coûts dans le programme. L'attribut "_cost" sert pour les coûts fixes, l'attribut "_rate" pour les coûts variables. Ce dernier doit utiliser la même unité de temps que celle définie globalement pour le modèle. Ces attributs se manipulent ensuite comme tous les autres attributs.

-Enter a schedule of arrival times-

	_Create Time	_Item Quantity	_Item Priority	_cost	_rate
1	0	10	1	5	0.1

Coûts définis par un bloc Create

De la même manière, vous pouvez définir dans les **blocs d'activité** un coût variable en fonction de la durée du traitement, ainsi qu'un coût fixe par traitement, ajouté à chaque entité passant par le bloc.

Define processing costs

Processing cost: / time unit

Cost per item:

Total cost:

coûts pour une activité

Combiner des ressources et des accumulateurs de coût

Il existe pour des entités deux manières d'utiliser des ressources : la première est de mettre en lot ressource et entité, la seconde est d'utiliser les blocs Resource Pool pour représenter les ressources.

Lorsque vous faites un lot entre une ressource et un accumulateur de coût, les taux définis pour la ressource sont stockés dans l'entité qui constitue le lot. Pour que ces taux ne soient plus utilisés après le délotage, vous devez spécifier "Release cost resources" (Libérer des ressources de coût) dans le bloc Unbatch. Si vous ne le faites pas (c'est à dire si l'option est "Create multiple items"), le bloc crée plusieurs copies de l'accumulateur de coût, chacune avec les informations de coût issues de la ressource.

Lorsque vous associez plusieurs ressources à un accumulateur de coût, vous pouvez libérer les ressources en une fois, ou incrémentalement, mais toujours en respectant en sortie le connecteur utilisé en entrée. Vous ne pouvez pas associer plus de deux ressources différentes à une même entité si vous demandez un calcul de coûts par ce système (en revanche il n'y a pas de limite si vous utilisez le bloc Resource Pool).

Avec la notion de Pool, lorsque des accumulateurs passent par une File, des ressources leur sont allouées. Les informations de coût du pool sont associées à l'entité et utilisées pour les calculs de coût. Lorsque le pool est libéré, ces informations ne sont plus utilisées par l'entité. Il n'y a pas de limite au nombre de pools associés à une entité pour le calcul des coûts.

Vous pouvez utiliser conjointement ces deux modes de représentation des ressources.

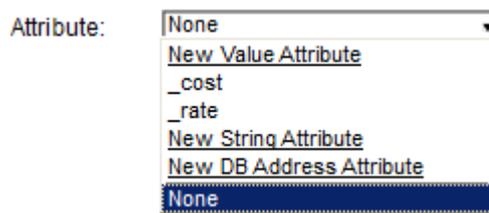
Coûts et mise en lot

Si vous sélectionnez l'option "Preserve uniqueness" (Individualiser) dans le Batch et le Unbatch, les coûts accumulés tant que les entités formaient un lot sont répartis entre les accumulateurs de coûts sortant du délotteur. Par exemple, si le lotteur associait trois entités et que le lot a accumulé 9 euros supplémentaires, chacun des trois accumulateurs de coûts sortira du délotteur en ayant accumulé 3 euros.

Si l'option " Preserve uniqueness " n'est pas sélectionnée, vous choisissez dans l'onglet Properties du bloc Unbatch comment traiter _rate et _cost sur les trois entités identiques produites.

Représentation et visualisation des coûts

Pour gérer les informations de coûts, ExtendSim utilise deux attributs ("_cost" et "_rate") pour chaque entité du modèle. Ils sont considérés comme des attributs "système", et figurent à part dans le menu déroulant des noms d'attributs :



Menu déroulant des attributs avec "_cost" et "_rate"

Ces attributs n'apparaissent que si un coût est défini quelque part dans le modèle. L'information stockée dans ces attributs n'est pas la même suivant que l'entité est un accumulateur de coûts ou une ressource :

Type d'entité	"_cost"	"_rate"
Accumulateur de coûts	Le coût accumulé de l'entité	Le coût d'attente ou de stockage de l'entité (fonction de l'unité de temps globale du modèle)

Ressource	Le coût par utilisation de la ressource	Le coût par unité de temps de la ressource (fonction de l'unité de temps globale du modèle)
-----------	---	---

En plus des blocs qui manipulent les attributs, deux blocs statistiques servent spécifiquement aux coûts.

Statistiques sur les coûts

Le bloc **Cost By Item** (bib. Item) s'insère dans le flux et lit (et stocke dans un tableau) les valeurs des attributs "_cost" des entités qui passent par lui, en calculant le coût moyen et le coût total.

Le tableau utilise soit une ligne par entité passée, soit une ligne par groupe d'entité d'un même type (défini par un attribut), ce qui permet d'obtenir des coûts et des moyennes pour des familles de produits.

Le bloc **Cost Stats** (bib. Report) réunit l'information Coût total issue de tous les blocs qui génèrent des coûts (générateur, ressource, activité, file), ce qui permet de déterminer rapidement quels blocs ont le plus contribué au coût de l'entité.

Mesures et statistiques

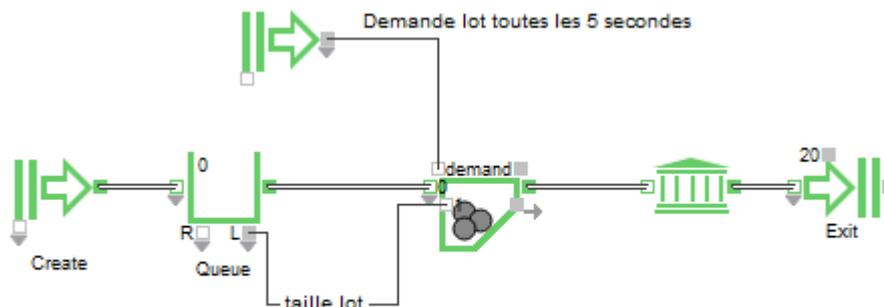
Mesurer, vérifier et documenter les résultats d'une simulation

Nos exemples se sont surtout intéressés à des problèmes de modélisation de cas typiques, et au flux des entités traitées. Vous aurez souvent à rassembler des informations d'autre nature, par exemple des taux d'utilisation, l'emploi de ressources, des débits ou des files d'attente. Cette phase d'exploitation du modèle est facilitée par ExtendSim.

De nombreux dialogues fournissent des informations complètes telles que longueur d'une file d'attente, arrivées et départs d'entités, etc. Une façon simple d'obtenir ces informations est de laisser les dialogues ouverts durant la simulation. Vous pouvez encore cloner les éléments qui vous intéressent dans un journal de bord.

D'autres informations seront plus exploitables si vous en tracez la courbe dans le temps, par exemple le connecteur L des files d'attente, qui indique le nombre d'entités en attente.

Certains blocs sont spécialement conçus pour fournir des informations sur l'état du modèle : par exemple le bloc **Information**, qui donne des informations sur les attributs, les intervalles de temps et le nombre d'entités, ou le bloc **History**, qui fournit un historique complet des entités qu'il a vues passer, avec leur temps d'arrivée, leur priorité et la valeur des attributs choisis.



Note: la taille du lot est égale au nombre d'entités dans la file à cet instant.

utilisation du bloc History

Comme ce bloc utilise beaucoup de mémoire, placez-le dans la section du modèle que vous testez, puis supprimez-le en phase finale.

Vous utiliserez aussi le bloc [Reports Manager](#) pour rassembler des informations sur les blocs dans une base de données et produire des rapports sur des composants sélectionnés.

Éléments de statistiques

Valeurs constantes ou variables aléatoires ?

Une valeur constante ne change pas, alors que les variables aléatoires sont fondées sur des distributions et changent à chaque utilisation. On appelle déterministes les modèles qui n'ont aucun paramètre aléatoire en entrée, et stochastiques ceux qui font appel à des paramètres aléatoires.

- Dans la mesure où tous ses paramètres sont constants, les résultats d'un modèle déterministe sont déterminés. Plusieurs simulations avec les mêmes données fourniront les mêmes résultats, ce qui permet de tester avec précision l'impact de la moindre nouvelle hypothèse.
- Dans la réalité cependant, beaucoup de phénomènes sont aléatoires, notamment lorsqu'ils font référence à des comportements humains. Notez que le fait qu'un phénomène soit aléatoire n'implique pas du tout qu'il soit peu connu, indéfinissable ou même imprévisible. Dans une distribution aléatoire, on peut souvent indiquer précisément les valeurs possibles.

Dans ExtendSim, ce sont les blocs Random Number, Create et Shutdown qui vous permettront de manipuler des valeurs aléatoires.

Base aléatoire

Les valeurs aléatoires sont calculées par un générateur aléatoire interne à ExtendSim qui utilise un

nombre appelé **random seed** (ou *base aléatoire*) afin de produire aussi bien des séquences aléatoires indépendantes que des séquences aléatoires reproductibles.

ExtendSim affecte une base aléatoire différente à chacun des blocs aléatoires d'un modèle. Vous pouvez spécifier votre propre base pour ces blocs, si vous le souhaitez. Dans le dialogue Paramètres de la simulation, vous pouvez donner la même base aléatoire à tous les blocs du modèle, bloquant ainsi le générateur de nombres aléatoires sur une séquence particulière, ce qui permet de voir l'effet d'hypothèses de simulation qui ne seront pas perturbées par des variations aléatoires.

Choix d'une distribution

À moins d'avoir une formation ou des outils statistiques, vous ne savez pas toujours quelle distribution correspond au phénomène aléatoire que vous voulez représenter. Voici quelques conseils pour identifier les principales lois de distribution proposées et leurs cas courants d'application.

Constante : ne produit pas de nombre aléatoire, mais une valeur constante qui ne change jamais. Permet d'imposer un intervalle constant entre des arrivées, ou bien de gommer l'aspect aléatoire d'un phénomène en cours de modélisation.

Empirique (tableau) : permet de décrire un phénomène par des valeurs (dans la première colonne) et entre la probabilité sur 100 d'obtenir cette valeur (dans la deuxième colonne). C'est donc du « sur-mesure ».

Erlang : une distribution utilisée dans le trafic téléphonique et la théorie des files d'attente, et sert à représenter des temps de service.

Exponentielle : la plus couramment utilisée dans l'industrie et les services, pour représenter des intervalles entre des événements indépendants. S'emploie dans le trafic téléphonique et la théorie des files d'attente.

Normale : courbe souvent connue (Gauss), mais plus adaptée à des phénomènes naturels qu'humains, ou pour représenter une distribution d'erreurs. Peu adaptée à des temps de service.

Triangulaire : très adaptée à des temps de service, permet une approximation que l'on peut affiner par la suite. Plus adaptée que la distribution normale parce qu'elle pourra être orientée vers la valeur probable et qu'aucune valeur ne dépassera les bornes.

Uniforme : décrit une valeur comprise entre des bornes spécifiées.

Weibull : distribution utilisée en analyse de fiabilité.

Ce ne sont que quelques distributions courantes parmi les dizaines que propose ExtendSim.

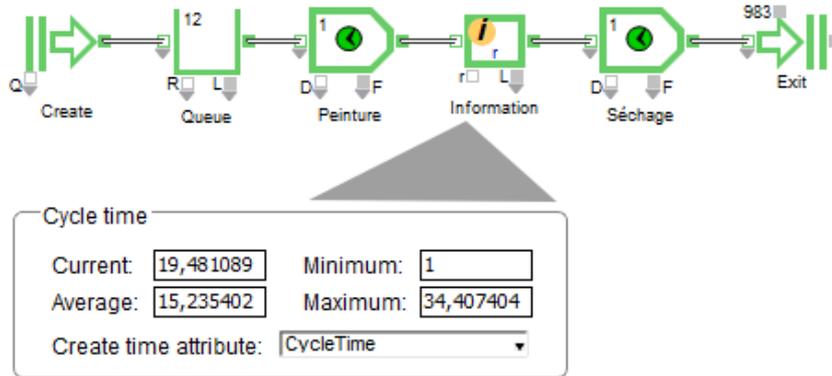
Accumuler des données avec des attributs

Vous pouvez utiliser les attributs pour effectuer des cumuls, par exemple un temps de traitement total, ou bien un poids total.

Vous pouvez accumuler des valeurs à n'importe quelle étape du modèle, même hors d'un traitement. Par exemple, les divers assemblages possibles pour constituer un produit résultent en des poids différents. Vous pouvez avoir un attribut auquel est ajoutée une valeur chaque fois qu'un composant est assemblé. Vous utilisez pour cela un bloc Equation pour modifier l'attribut en sortie d'activité.

Chronométrer les entités dans une portion du modèle

Pour repérer la cause réelle de goulets d'étranglement et effectuer des tests de qualité plus approfondie, il est souvent utile de connaître le temps passé par chaque entité dans telle portion du modèle. Pour cela, créez un attribut de timing : lors de la création par le bloc Create qui propose l'option dans l'onglet Options, ou par un Set au début de la portion à observer. À la fin de la portion à observer, placez un bloc Information (ou Equation(I) si vous voulez faire un calcul en plus) : voyez les modèles Cycle Time 1 et 2.



entités chronométrées

Pour chaque entité, l'Information affiche des statistiques sur les temps de cycle observés (le top de début est donné au moment où l'entité quitte le bloc).

Si vous spécifiez un nom d'attribut sur un seul flux du modèle, seules les entités ayant cet attribut sont mesurées.

Modélisation avec la bibliothèque Rate

Introduction

Quelques principes à connaître

Ce type de modélisation s'applique à des débits de flux qui changent lorsque des événements surviennent. Des quantités de "flux" (matière, produit, données, etc.) figurent dans un ou plusieurs endroits du modèle. Durant la simulation, le flux se déplace d'un emplacement à un autre à une certaine vitesse, appelée débit réel. Le mouvement entre les divers blocs contenant le flux suit des chemins, des règles et des contraintes définies dans le modèle.

Dans certaines situations (listées plus loin), ce type de modélisation est plus adapté que l'orientation purement continue ou discrète. Les process soumis à des événements (et pas uniquement à l'écoulement du temps), se prêtent mal à la simulation continue. Les systèmes où aucune entité n'est identifiable, ou bien où elles sont en tel nombre que leur individualisation n'a pas de sens, sont plus facilement modélisés avec la bibliothèque Rate qu'avec de simples blocs discrets. De plus, les modèles de ce type s'exécutent plus rapidement que les modèles discrets standard, et s'appliquent à une représentation par flux, vannes, débits, silos, etc.

☞ Tout système impliquant une quantité de matière stockée en un endroit, puis se déplaçant en un autre endroit à un certain *débit par unité de temps*, peut être modélisé avec la bibliothèque Rate.

Ce chapitre couvre :

- Domaines d'application
- Modèles flux
- Composantes des modèles flux
- Concepts Rate
- Technologie PL
- Agencement d'un modèle flux
- Le bloc Executive
- Connecteurs et liens
- Unités et groupes unitaires
- Débits
- Organisation d'un modèle Rate

Domaines d'application

La bibliothèque Rate sert à représenter de la matière, par exemple des poudres ou des liquides, des gaz ou d'autres matières fluides, dans les domaines suivants :

- Pétrochimie
- Production
- Mines
- Traitement de l'eau
- Pharmacie
- Métallurgie
- Toute industrie incluant de la matière en vrac et du traitement par batch

Elle sert à modéliser des "choses" en nombre tel qu'il serait décalé ou très volumineux de les modéliser individuellement :

- Nourriture et boissons (sachets de thé, céréales, sodas, etc.)

- Pharmacie, cosmétiques (cachets, lotions)
 - Broyage (laminage, papeteries, filatures)
 - Stockage et traitement de données (échantillons, messages, paquets)
 - Toute industrie qui mélange, remplit ou conditionne des produits en volumes importants sur des lignes à grande vitesse
- ☞ Il n'est pas rare, nous le verrons aussi, qu'un modèle en logique flux comporte des portions discrètes standard.

Modèles flux

Comparaison entre modélisation discrète et modélisation continue

Les blocs des bibliothèques Value et Item agissent individuellement et de manière indépendante pour calculer des valeurs ou déplacer des entités. Ils peuvent communiquer entre eux et envoyer des messages, mais il n'y a pas de lien global entre eux.

Les blocs de la bibliothèque Rate dépendent les uns des autres et agissent sur les autres. Les modèles peuvent se diviser en zones qui font partie d'un système global. Les blocs à l'intérieur d'une zone communiquent par un programme linéaire interne (PL) qui pilote toute la zone. Chaque bloc dans une zone PL contribue pour partie à l'équation PL de la zone; dont le résultat est le débit réel affectant cette portion du modèle. Le système est optimisé de sorte que si une portion n'a pas besoin d'être recalculée, elle ne l'est pas.

Une autre différence majeure tient dans le déplacement de la matière.

- Dans un modèle discret, les entités vont d'un bloc à un autre instantanément. Les contraintes figurant par les blocs, et non sur les mouvements entre les blocs.
- Dans un modèle flux, le mouvement du flux prend du temps. Sinon, le débit du flux pourrait approcher l'infiniment rapide, ce qui n'est pas réaliste. L'essentiel de l'intelligence d'un modèle s'exprime par des contraintes sur le déplacement du flux.

Composantes des modèles flux

L'on rencontre dans ces modèles des flux, des contraintes, des débits; des événements, des capacités de stockage et des orientations.

- Le flux est ce qui est stocké et progresse dans le modèle. Physiquement, ce peut être n'importe quoi, à condition de ne pas avoir à donner des propriétés particulières à chaque individu du flux. Le flux est mesuré en unités de flux, – générique ou sous forme de paquets, litres, transactions, tonnes, boîtes, etc.
- Le flux progresse par les connecteurs flux dans une seule direction, d'un connecteur du flux sortant d'un bloc au connecteur du flux entrant d'un autre. Il avance à un rythme exprimé en nombre d'unités de flux par unité de temps – nombre de paquets par seconds, litres par minute, tonnes à l'heure, etc.
- L'architecture flux met au centre la progression du flux. Les blocs expriment les contraintes qui freinent cette progression : présence ou absence du flux dans un silo, débit maximum traduit par une vanne, règle de division au moment d'une distribution.
- Les contraintes déterminent des débits maximum, mais le débit réel indique la progression effective du flux. Le taux réel dans chaque section est déterminé au moyen d'une programmation linéaire (PL), tenant compte de l'ensemble des contraintes. À chaque événement, la quantité de flux qui s'est déplacée est mise à jour et les contraintes réévaluées.
- L'état d'un modèle flux ne change que lorsqu'un événement se produit. Cet événement, c'est un silo qui devient vide ou plein, une contrainte de débit qui change durant la simulation, des proportions modifiées en sortie d'un bloc, etc. À chaque événement, ExtendSim calcule les nouveaux débits réels du modèle, dans chaque portion qui a pu connaître des modifications.
- Chaque modèle flux doit être virtuellement divisé en groupes unitaires, sections de débit, et zones PL. Ces divisions sont gérées de manière automatique et interne, et sont déterminées par le type de blocs, la manière dont ils sont liés, leurs paramètres, etc.

Une liaison entre deux blocs flux peut être considérée comme un tuyau infiniment petit qui est en

permanence plein d'une matière à pression constante – dès que le débit réel est supérieur à 0, le contenu du tuyau avance avec le débit maximum que permettent les contraintes du système. Lorsque le débit réel est 0, le tuyau reste plein, mais le flux s'arrête.

☞ L'architecture Rate de ExtendSim conserve l'équilibre des masses dans le système, mais aucune considération n'est donnée à la pression, l'énergie, le moment ou la température, qui ne sont pas de son ressort.

Les blocs Rate

La difficulté du calcul du débit réel et la génération des événements qui décident d'un nouveau calcul des taux est automatiquement gérée par les blocs Rate. Les blocs de la bibliothèque Rate entrent dans plusieurs catégories :

- Certains blocs stockent et fournissent le flux matière
- D'autres blocs influencent le débit réel
- Les autres blocs servent à l'orientation du flux

Ces blocs intègrent les concepts de contraintes, cibles, priorités des flux, mélanges, lots, indicateurs de niveau, etc.

Terminologie et architecture

Avant de construire un modèle flux, il est important de se mettre d'accord avec le vocabulaire employé.

Technologie PL

Pour calculer le débit réel dans un modèle flux, ExtendSim utilise une technologie de programmation linéaire (PL). Ce calcul cherche à trouver le meilleur débit réel compte tenu des contraintes et de la structure du modèle. Voyez l'examen avancé "Technologie PL" à la fin du manuel.

Agencement d'un modèle flux

Un modèle flux peut combiner des blocs continus et discrets, et bien sûr ceux de la bibliothèque Rate. Dès que vous placez un bloc Rate dans un modèle, il faut placer aussi un bloc Executive (bib. Item).

A part le bloc Executive, vous placez les blocs où vous voulez, sans oublier qu'ExtendSim évalue les blocs flux en suivant l'enchaînement de leurs liens.

Bloc Executive

Le bloc Executive (bib. Item) planifie les événements et lance les calculs PL. L'onglet *Rate* du bloc permet de définir des options globales, de gérer les unités de quantité, et de sélectionner des options avancées concernant les modes de divergence et convergence. Il est expliqué plus loin.

☞ Dans la majorité des cas vous ne modifierez pas les paramètres du bloc Executive.

Connecteurs et liens

La plupart des blocs de la bibliothèque Rate ont des connecteurs flux et des connecteurs continus; le bloc Interchange ayant en plus des connecteurs discrets.

- Dans un modèle flux, des connecteurs flux indiquent le débit réel du flux à chaque événement. Le flux se déplace dans une direction, d'un connecteur de sortie du flux ("flux sortant") à un connecteur d'entrée du flux ("flux entrant").
- Des connecteurs continus fournissent des informations sur la taille du flux et la capacité du bloc, entre autres.
- Des connecteurs discrets sur le bloc Interchange permettent l'interface entre des portions de type « flux » et des portions discrètes classiques.

Vous devez respecter les types de connecteurs en reliant les différents blocs. Pour représenter le flux matière, un connecteur du flux sortant doit être relié à un connecteur du flux entrant. Chaque connecteur flux n'accepte qu'une seule connexion. Il est cependant possible de relier un connecteur du flux sortant à la fois à un connecteur du flux entrant et à un connecteur d'entrée continu. Le connecteur d'entrée continu lit alors la valeur de débit réel.

Unités et groupes unitaires

Il y a quatre types d'unités dans la bibliothèque Rate :

- L'unité de flux précise ce qui passe d'un connecteur flux à un autre – par exemple, des litres, des

Bias
Catch Flow
Change Units
Convey Flow
Diverge
Get(R)
History(R)
Interchange
Merge
Sensor
Set(R)
Tank
Throw Flow
Valve

bouteilles. L'unité de flux est la même pour tous les blocs d'un groupe unitaire.

Unité du groupe : litres / minute*

- Les unités de temps obéissent aux mêmes règles que dans les modèles discrets.

Les dates calendaires ne sont pas disponibles si les mois ou les années sont l'unité globale d'un modèle flux. Et même, si les dates calendaires ont été sélectionnées, les blocs Rate ne proposeront pas les unités locales Mois ou Ans.

- Les unités de bloc sont des unités de volume internes, spécifiques aux blocs Tank et Interchange. Si vous sélectionnez une unité de bloc différente de l'unité de flux, vous devrez préciser un facteur de conversion.

Un groupe unitaire est un ensemble de blocs liés ensemble et partageant la même unité de flux. Pour visualiser un groupe, cliquez sur le carré gris à la droite du menu déroulant Unités de flux d'un bloc. Tous les blocs du groupe s'afficheront alors en surbrillance.

Unité du groupe : litres 

Débits

L'un des plus importants aspects d'un modèle flux est le débit du flux. Le taux s'affiche dans le dialogue du bloc sous forme de unités/temps, litres/minute, transactions/seconde, tonnes/heure, etc.

Différents types de débits doivent être considéré lors de la construction du modèle :

- Débit maximum – la limite supérieure de débit.
- Débit réel – le débit effectif.
- L'alimentation amont et la demande aval – les débits potentiels.
- Débit infini – Toute valeur supérieure ou égale à un nombre très grand.

Comment construire un modèle flux

L'exemple de ce chapitre montre comment construire un modèle flux portant sur un process de yaourt, et utilisant de nombreux blocs de la bibliothèque Rate. Pas à pas, le chapitre montrera comment :

- Construire un modèle simple
- Faire varier un débit en fonction de l'heure
- Ajouter une seconde source de produit, parfois interrompue pour une maintenance
- Mélanger deux sources selon une certaine proportion
- Créer une opération de remplissage
- Ajouter un convoyeur pour simuler une opération de refroidissement
- Conditionner les pots en cartons
- Créer une zone de palettisation et de stockage
- Créer une seconde zone de palettisation en parallèle

A propos du modèle

Le modèle Yaourt Production représente un process qui prend une quantité de liquide, la convertit en yaourt, puis mélange le yaourt avec des fruits. Le mélange est versé dans des pots de yaourt et refroidi, puis les pots sont conditionnés dans des cartons. L'étape finale place les cartons sur des palettes.

Le modèle s'appuie sur quelques hypothèses :

- Le liquide servant à faire le yaourt provient d'un emplacement, et les fruits d'un autre. Les deux existent en quantité infinie.
- La plus grande partie de la journée, le liquide est transformé en yaourt à la vitesse de 100 litres/minute.
- Au moment du repas, comme il y a moins de personnel, le débit décroît à 60 litres/minute pendant cette heure.
- Après traitement, le yaourt est transféré vers la zone de mélange.
- Les fruits arrivent à la zone de mélange à la vitesse de 8 litres par minute lorsque les équipements ne sont pas en maintenance, et à un débit de 2 litres par minute sinon.
- Pour 10 litres de mélange il y a 1 litre de fruit et 9 litres de simple yaourt.
- On obtient 12 pots de yaourt par litre mélange.
- Le cycle de refroidissement s'opère sur une unité longue de 20 mètres. Le yaourt doit être refroidi au moins 20 minutes avant de mettre les pots en cartons.
- Chaque carton contient 48 pots de yaourt
- Il y a deux zones de palettisation, l'une ayant une priorité supérieure à l'autre.
- Les palettes arrivent toutes les 2 minutes et peuvent supporter 24 cartons.
- L'unité de temps est la minute et la simulation dure 480 minutes.
- Les blocs proviennent des bibliothèques Rate, Item et Chart

☞ Les modèles Yaourt Production illustrant les étapes de ce chapitre figurant dans le répertoire \Exemples\Tutorials\Item Rate. Nous conseillons de construire le modèle par vous-même.

Paramètres de la simulation

Commencez un nouveau modèle. Dans le dialogue Paramètres de la simulation, entrez :

- Temps de fin : 480

- Unité de temps globale : minutes

Commencer petit

Dans un premier temps, construisez une ligne de production, où une Cuve contient le produit qui progresse vers un autre Cuve à un débit constant.

Contenu initial : litres
 Capacité de la cuve : litres

Le tableau ci-dessous montre les blocs et leur emploi. À part le graphique, tous les blocs proviennent de la bibliothèque Rate. Placez le bloc Executive s'il n'est pas présent, puis les autres dans l'ordre du tableau.

Donnez des labels comme indiqué.

Nom (Label)	Fonction du bloc	Rôle dans le modèle
Tank (source liquide)	Sert de source, de stockage intermédiaire ou de cuve pour la matière du modèle.	Contient une quantité illimitée de liquide transformable en yaourt.
Valve (Trait. yaourt)	Impose une contrainte sur le flux. Contrôle et pilote le transfert de flux à un certain débit.	Régule le débit à 100 litres par minute. (Une contrainte est requise)
Tank (Yaourt)	Sert de source, de stockage intermédiaire ou de cuve pour la matière du modèle.	Contient une quantité illimitée de yaourt.
Traceur discret	Affiche des informations sur le flux et des valeurs du modèle.	Indique combien de litres de yaourt par minute sont traités (débit réel) et la quantité totale de yaourt produit.

Créer les liens

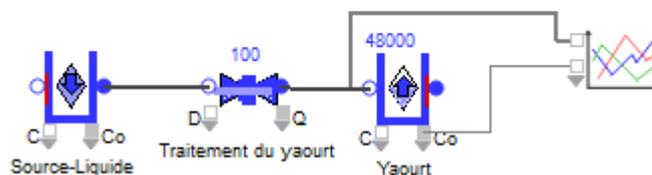
Reliez les connecteurs comme suit :

- Reliez le connecteur du flux sortant du premier Tank (source liquide) vers le connecteur entrant de la Valve.
- Reliez le connecteur du flux sortant de la Valve au second Tank (Yaourt).

Pour obtenir des informations sur la quantité de yaourt traité :

- Reliez le connecteur du flux sortant de la Cuve Yaourt à l'entrée du haut du Line Chart.
- Reliez le connecteur *Co* (niveau du contenu) de la Cuve Yaourt à la seconde entrée du Line Chart.

Quand vous aurez fini, le modèle devrait ressembler à



Modèle de base Production de yaourt

Entrer les paramètres des dialogues

Pour le premier Tank ; dans l'onglet Tank, cochez la case ∞ (infini) pour le champ *Contenu initial*. Le mot "infini" s'écrit dans le champ.

Dans l'onglet Options du bloc, sélectionnez Nouvelle unité dans le menu déroulant de Unité de flux: et entrez litres pour unité de flux.

Dans le dialogue du bloc Valve, entrez Maximum rate: 100 litres/minute.

Il n'y a pas de paramètres à saisir pour la Cuve de Yaourt. Par défaut il n'a pas de contenu initial et sa capacité maximum est infinie, ce que nous souhaitons.

Dans ce modèle, le graphique trace le débit maximum de la Valve sur son entrée du haut, et le nombre de litres de yaourt traités sur la seconde entrée. Comme ces nombres sont sur deux échelles très différentes, demandez dans l'éditeur de courbes du Line Chart à ce que la seconde entrée soit tracée sur l'axe Y2, et choisissez une ligne de type interpolée.

Enregistrez le modèle et lancez simulation.

Vérifier les résultats

Il est toujours bon de vérifier les résultats intermédiaires. Comme il n'y a aucun changement dans les débits, le modèle n'a rien à recalculer. Cela signifie que la simulation se termine en deux événements : celui de début et celui de fin. Le graphique indique un débit réel de 100, et un total de 48,000 litres de produit (visible sur l'axe Y2) a été traité. Cela semble logique si 100 litres de yaourt sont produits par minute et que la simulation dure 480 minutes.

Ajouter une contrainte dynamique

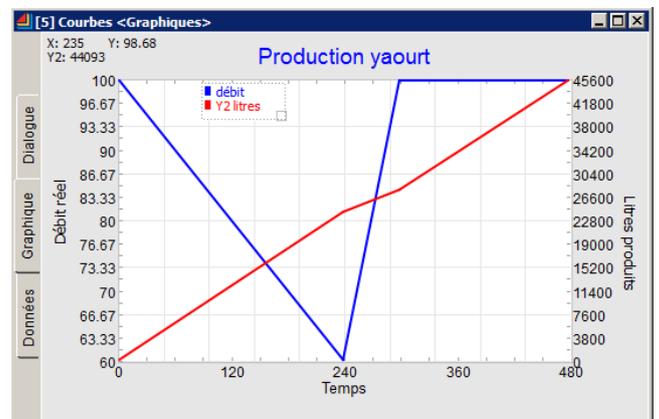
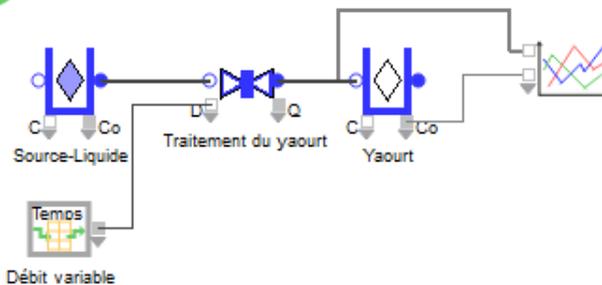
Il n'est pas rare que le débit maximum d'une vanne change en fonction du temps. Pendant l'heure du repas, le débit de yaourt va passer de 100 litres par minute à 60 litres par minute. Pour cela :

- Ajoutez un bloc Lookup Table (bib. Value).
- Reliez la sortie du Lookup Table à l'entrée R (débit maximum) de la Valve.
- Dans le dialogue du bloc Lookup Table, sélectionnez Correspondance sur: temps.
- Dans l'onglet Options du bloc, entrez les labels de colonne Minutes; Litres/Minute.

	Minutes	Litres/Minute
0	0	100
1	240	60
2	300	100
3	480	100

Dans l'onglet Table, entrez les valeurs affichées à droite pour les colonnes Minutes et Litres/Minute. Cela fait que le débit maximum de la Vanne est de 100 litres/minute sauf pendant la période de la minute 240 à la minute 300, où il est de 60 litres/minute.

Donnez au bloc le label *Débit variable*.



Le modèle doit ressembler à l'illustration ci-dessus.

Le graphique montre bien le changement de débit à l'heure du déjeuner, ainsi que la quantité réduite de yaourt, qui n'atteint plus le total de 48,000 litres.

- ☞ Le bloc Lookup Table émet des valeurs à chaque temps spécifié, en fonction de la table, sans besoin d'aller chercher les valeurs.

Ajouter la ligne de fruits

Le yaourt doit être mélangé à des fruits, ce qui requiert une seconde ligne de traitement:

Supprimez le lien entre la Valve et la cuve de Yaourt et déplacez la cuve sur la droite.

Sous la Table de correspondance, ajoutez un autre bloc Tank.

- Dans l'onglet Tank, cochez la case ∞ (infini) pour le champ *Contenu initial*.
- Dans l'onglet Options du bloc, sélectionnez Unité de flux: litres.
- Donnez au bloc le label Source fruit.

Ajoutez une Valve à la droite de la cuve Source fruit.

- Reliez le connecteur du flux sortant de la cuve au connecteur de flux entrant de la Valve.

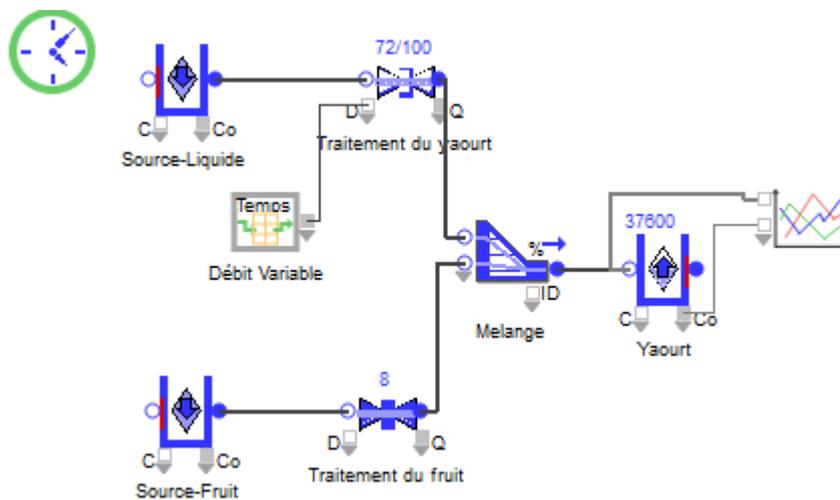
- Dans l'onglet Valve, entrez Maximum rate: 8 litres/minute.
- Donnez au bloc le label Traitement fruit.

Notez que litres est automatiquement sélectionné pour unité dans la Valve. C'est parce qu'elle est reliée au Tank.

Ajouter une contrainte variable

Pour mélanger les deux flux, ajoutez un bloc Merge entre les blocs Traitement yaourt et Traitement fruit.

- Reliez le connecteur sortant de la Valve Traitement yaourt au premier connecteur du Merge.
- Reliez le connecteur du flux sortant de la Valve Traitement fruit au second connecteur du Merge.
- Dans l'onglet Merge du dialogue, sélectionnez *Mode: proportionnel* dans le menu.
- Dans la colonne Proportion de la table, entrez 9 pour le Traitement yaourt et 1 pour le Traitement fruit.
- Donnez au bloc Merge le label *Mélange*.
- Reliez le connecteur du flux sortant du Merge au connecteur du flux entrant de la cuve Yaourt.



Lorsque vous lancez le modèle, le process de mélange doit traiter 80 litres par minute et le process entier produire 37,600 litres de yaourt. Il est intéressant d'avoir l'animation active (à un rythme lent). Vous verrez peut-être que le débit affiché au-dessus du Traitement yaourt indique parfois la fraction 72/100. C'est le ratio du débit réel par rapport au débit maximum. Dans ce modèle, il n'y a parfois pas assez de fruit et ainsi le débit réel est inférieur au débit maximum.

En validant le modèle, notez que le débit réel pour la partie yaourt du process ne peut jamais être supérieur à 72. Comme la sortie maximum du traitement fruit est de 8 litres par minute, et que le mélange requiert un ratio de 9 portions de simple yaourt pour 1 portion de fruit, la quantité maximum de simple yaourt requise ne peut dépasser 72 (8×9) litres par minute.

Ajouter une maintenance

La maintenance aussi va ralentir le débit pendant une période de temps. Le traitement des fruits a un débit lent de 2 litres par minute durant la maintenance, pour un débit normal de 8 litres par minute. Il y a une maintenance environ toutes les 60 minutes, d'une durée aléatoire de 5 minutes minimum, 20 minutes maximum, et le plus souvent 15 minutes. Pour refléter cela :

- Ajoutez un bloc Shutdown (bib. Item) d'une des deux manières suivantes :
- Cliquez sur le bouton Ajouter des arrêts dans le dialogue de la Valve. Cela lie automatiquement un bloc Shutdown au connecteur d'entrée D (débit maximum) et ouvre le dialogue du bloc Shutdown.
- Placez le bloc Shutdown et faites les liens similaires manuellement.

Dans le dialogue du bloc Shutdown :

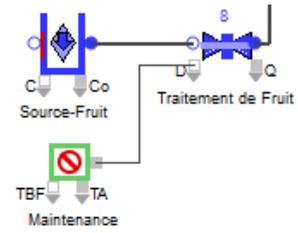
- Pour la configuration de l'arrêt, entrez *Down Value* (Valeur d'arrêt): 2 et *Up Value* (Valeur de marche): 8
- Pour l'intervalle entre arrêts (TBF) choisissez une distribution **Exponential** avec une moyenne

de 60

- Pour la durée de l'arrêt (ITR) sélectionnez une distribution **Triangular** avec pour Minimum: 5, Maximum: 20, et Plus probable: 15.

Donnez au bloc le label Maintenance.

- ☞ Comme le bloc Lookup Table, le bloc Shutdown émet ses informations sans qu'on lui demande. Ainsi la Valve n'a pas besoin d'être définie à *Poll constraint*. Ce concept est examiné plus loin.



Lorsque vous lancez le modèle avec l'animation active, notez que la Valve est partiellement arrêtée pour la maintenance un certain temps et que son débit maximum est réduit à 2 litres/minute durant la maintenance. Il y a aussi une moindre production totale de yaourt qu'à l'étape précédente.

Changer l'unité de flux pour préparer le conditionnement

On considère qu'un litre de mélange de yaourt remplit 8 pots. Pour représenter cela :

- Supprimez le lien entre le bloc Merge et la cuve Yaourt.
- Ajoutez un bloc Change Units entre les deux blocs. Donnez au bloc le label Chge unités 1.

Dans le dialogue du bloc Change Units :

- Ne modifiez pas le premier paramètre (Change units from de: litres)
- Dans le menu pour la seconde unité (to: litres), sélectionnez *New unit* et nommez la nouvelle unité de flux Pots.
- Puisque chaque litre devient 12 pots, entrez pour Facteur de conversion: 12 pots/litres. (Sélectionnez bien pots/litres dans le menu.)

Si vous cochez la case *Afficher les unités sur l'icône* dans le dialogue de Change Units, la zone au-dessus de l'icône affiche litres=>pots.

Après la saisie de ces paramètres, le dialogue du bloc Change Units devrait ressembler à :

Change l'unité de flux, créant un nouveau groupe unitaire

Changer l'unité de : /

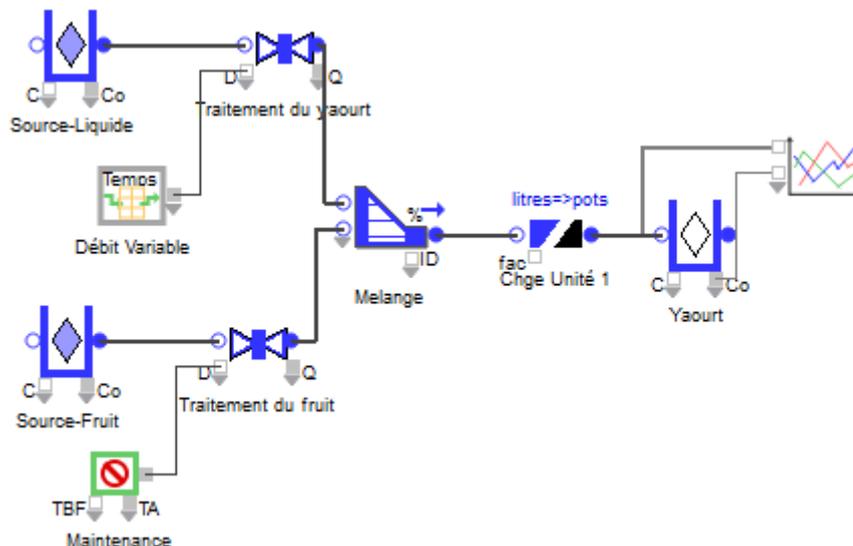
en : /

Facteur de conversion:

Facteur de conversion initial:

Afficher les unités sur l'icône

Lorsque vous lancez la simulation, le nombre de pots de yaourt produits chaque minute, et donc le nombre total de pots, change en fonction du débit maximum de la Vanne et des restrictions du flux lorsqu'il n'y a pas assez de fruit pour le mélange. Le graphique devrait indiquer une production d'environ 370,000 pots de yaourt.



- ☞ Le modèle considère que l'opération de remplissage des pots a le même rythme que l'opération de mélange. Si le conditionnement devait avoir eu un débit différent, il aurait fallu lier une Valve entre les blocs Merge et Change Units.

Refroidissement

Le process inclut une phase de refroidissement de 20 minutes dans une unité de réfrigération de 20 mètres de long, avant que les pots de yaourt puissent être réunis en cartons. Le bloc Convoyeur flux est conçu pour représenter un délai dans le mouvement du flux.

- Effacez le lien entre Change Units et la cuve de Yaourt.
- Ajoutez un bloc Convoyeur flux entre les deux blocs précédents.

Notez que si vous reliez d'abord le bloc Change Units, l'unité de flux appropriée (pots) est sélectionnée automatiquement dans l'onglet Options du nouveau bloc.

- Dans l'onglet Options du bloc Convoyeur flux :

Sélectionnez Nouvelle unité dans le menu *Unité de longueur*. Dans le dialogue qui apparaît, entrez mètres.

Dans l'onglet Convoyeur du bloc Convey Flow :

- Notez que par défaut le bloc est défini à *Accumule-densité maximum*.
- Dans le menu sur la droite, sélectionnez le comportement *Temps de trajet fondé sur le séjour*.
- Entrez Délai: 20 minutes *.
- Entrez Densité maximum: 500 pots/mètres.
- Donnez au bloc le label Refroidissement.

Lorsque vous lancez le modèle, remarquez qu'il n'y a pas de produit arrivant dans la cuve de Yaourt durant les premières 20 minutes. Cela représente le temps qu'il faut pour que les premiers pots quittent l'étape de refroidissement. Cela affame la portion du modèle en aval du bloc Convey Flow. Comme le produit fini prend plus de temps à être produit, la production est moindre qu'à l'étape précédente.

- ☞ Si un bloc Convey Flow est bien requis dans ce modèle, évitez d'en placer un trop grand nombre dans vos modèles car ils génèrent beaucoup de calculs. Souvent, une Valve et un Tank peuvent remplacer ce bloc.

Packaging des pots

L'étape suivante met les pots de yaourt en cartons. Il faut utiliser un nouveau bloc Change Units.

- Effacez le lien entre Refroidissement et la cuve de Yaourt.
- Ajoutez un second Change et liez-le au deux autres;

Dans le dialogue du bloc, ne modifiez pas le premier paramètre (Changer l'unité de: pots).

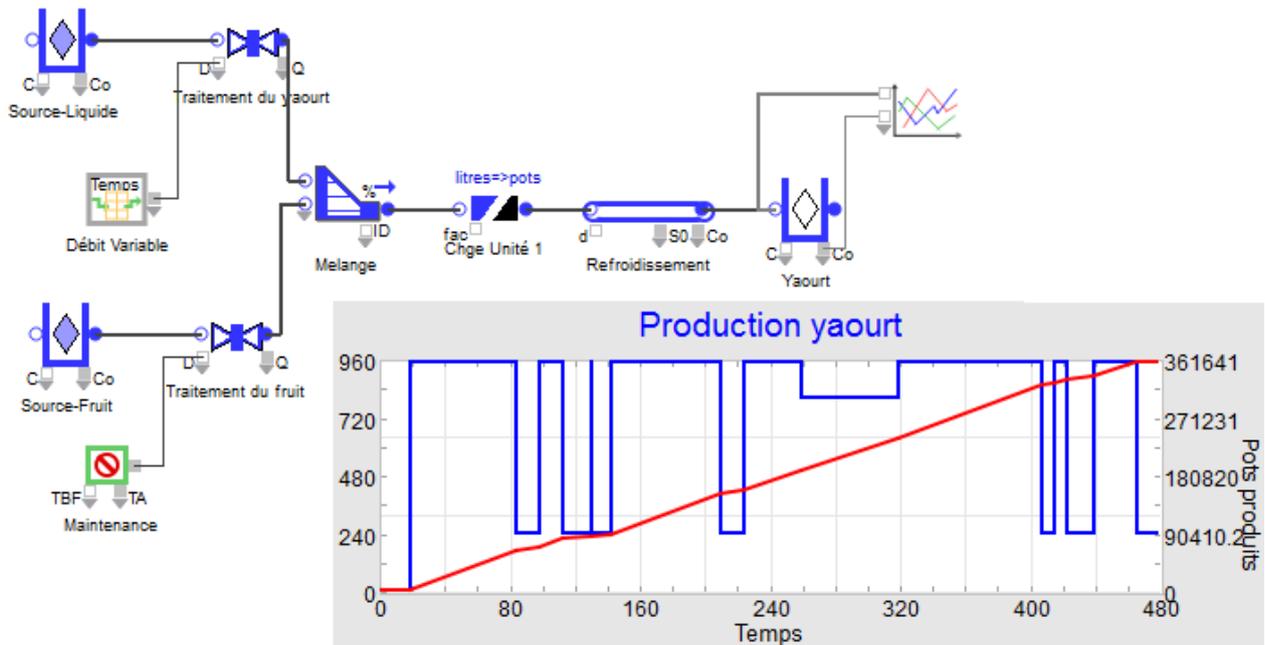
Dans le menu pour la seconde unité (en: pots), sélectionnez *Nouvelle unité* et nommez la nouvelle unité de flux cartons.

Entrez pour Facteur de conversion: 48 pots/cartons.

Cochez *Afficher les unités sur l'icône*, pour voir au-dessus de l'icône pots=>cartons.

Donnez au bloc le label Chge unités 2.

Quand vous aurez fini, le modèle devrait ressembler à



A quelques variations près, le modèle devrait produire environ 8,000 cartons. (Chaque carton contient 48 pots et le process fabrique environ 360,000 à 390,000 pots de yaourt.)

Ajout d'une zone de palettisation

Les cartons sont stockés dans cette zone avant expédition. Une palette vide arrive toutes les 2 minutes et peut contenir 24 cartons. Sans palette vide, le flux de yaourt s'arrête lorsqu'une palette pleine part.

Le bloc Interchange

Le bloc Interchange représente un stock ou une zone de stockage, où le flux peut interagir avec des entités discrètes. Le bloc travaille sur une seule entité à la fois. Par défaut, le bloc ressemble un peu à un interrupteur. Lorsqu'il contient une entité, il peut aussi recevoir le flux matière. En l'absence d'une entité, il ne peut recevoir le flux matière.

Dans ce modèle des entités représentant des palettes vides sont générés aléatoirement. L'arrivée d'une palette fait que le bloc Interchange peut recevoir le flux, avec une capacité maximum de 24 cartons, comme l'indique le dialogue. Lorsque ce maximum est atteint, c'est une entité « palette pleine » qui quitte le bloc. Le bloc Interchange est inactif jusqu'à ce qu'une nouvelle entité arrive.

☞ Le bloc Interchange est examiné en détail dans "Utiliser le bloc Interchange pour mélanger entités et flux" plus loin.

Ajout d'une zone de palettisation

Supprimez la cuve de Yaourt, puis placez les blocs discrets suivants, en les reliant comme indiqué : Create, Queue, Interchange (Rate), Exit

- Reliez le connecteur sortant de Chge unités 2 au connecteur de flux entrant du bloc Interchange 1.

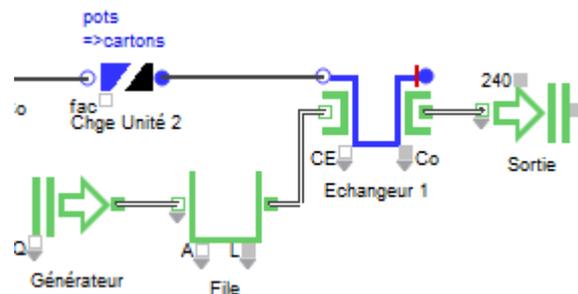
Dans le dialogue du bloc Create, donnez une arrivée Constante d'intervalle : 2.

Les blocs Queue et Exit sont correctement paramétrés par défaut.

Dans l'onglet Entité/Rate du bloc Interchange, définissez les comportements suivants (Tank only exists..) :

- Capacité, quand l'entité est présente: une constante de 24 cartons.
- Libérer entité: quand contenu \geq Cible (chargement).
- Cible: plein.

Ainsi, le bloc Interchange a une capacité de 24 cartons chaque fois qu'arrive une entité représentant une palette vide. Lorsque 24 cartons de yaourt sont arrivés, le niveau du bloc est plein, et l'entité peut partir. Le bloc Interchange attend alors la palette suivante.



Dans le modèle actuel, on manque de palettes, ce qui ralentit le process et parfois l'arrête. Il serait préférable d'avoir deux zones de palettisation.

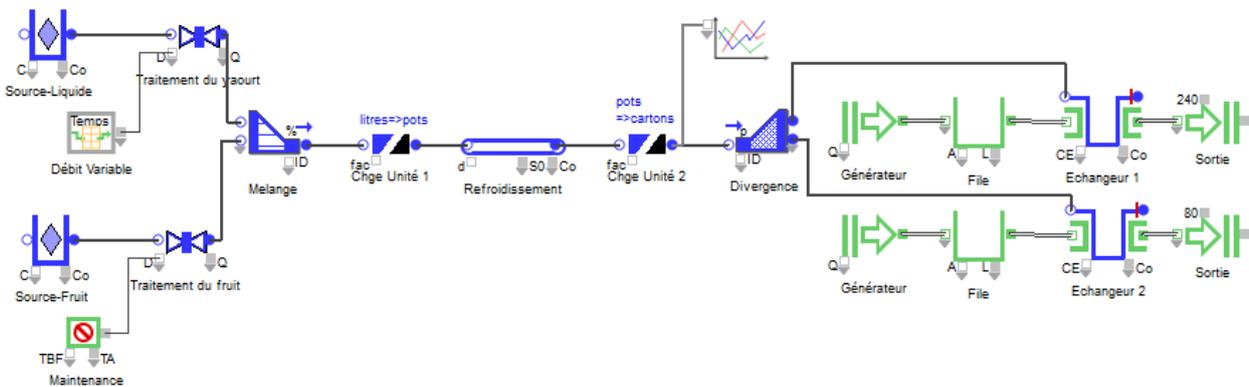
Ajout d'une seconde zone de palettisation

Le plus simple est de dupliquer les blocs de la première zone. Placez les 4 blocs dupliqués sous la première zone, et entrez les labels Générateur 2, Queue 2, Interchange 2 et Sortie 2.

- Ajoutez un bloc Diverge à la droite du Chge unités 2:
- Reliez le Chge unités 2 au Diverge.
- Reliez le premier connecteur du flux sortant du Diverge au connecteur de flux entrant de l'Interchange 1.
- Reliez le second connecteur du flux sortant du Diverge au connecteur de flux entrant de l'Interchange 2.

Dans le dialogue du bloc Diverge, notez que par défaut le Diverge mode: est *priorité des sorties*, et qu'une priorité de 1 est affectée au bloc Interchange du haut et une priorité de 2 au bloc Interchange du bas. Ne changez pas ces paramètres, qui conviennent.

Quand vous aurez fini, le modèle devrait ressembler à



C'est la zone de palettisation du haut qui a la priorité pour traiter les cartons. Désormais, il est très rare qu'il y ait 0 cartons par minute.

Qualifier les flux

Nous allons identifier les diverses saveurs de yaourt produites en ajoutant des attributs flux pour qualifier la source de fruit. Un attribut flux est une qualité ou caractéristique du flux qui est conservée durant la progression du flux dans le modèle, telle que le parfum, la couleur, ou l'ID produit. Les attributs flux permettent d'organiser des quantités ou des volumes de flux en couches. Les valeurs des attributs flux affectent une couche particulière qui se distingue ainsi des autres couches dans les mêmes espaces de stockage. En d'autres termes, toutes les unités de flux d'une couche possèdent le même ensemble de valeurs d'attributs flux. L'utilisation des attributs flux fait partie des fonctionnalités avancées, parfois subtiles à employer.

C'est dans le bloc Executive que nous allons définir les attributs flux, qui seront ici des attributs chaîne permettant d'identifier la saveur du yaourt.

Dans l'onglet Attributs flux, créez l'attribut Parfum, et remplissez le tableau des cinq valeurs possibles, comme ci-dessous :

Choisir une action

Suivi des valeurs chaîne par les blocs distants

Sélectionnez un attribut flux chaîne

ele	Nom attribut flux	Initialisation
1	Parfum	

Entrez les valeurs pour cet attribut flux chaîne

	Parfum
1	Vanille
2	Pêche
3	Abriçot
4	Banane
5	Fruits rouges

Dans le bloc Tank qui re présente la source de fruits, choisissez dans le menu Contenu initial défini

par des attributs flux, et remplissez le tableau comme indiqué. Ainsi la cuve contient initialement 1000 litres de fruits, en cinq couches de parfums différents. Lorsque 1000 litres ont été absorbés par la production de yaourt, les mêmes quantités sont à nouveau disponibles dans les mêmes proportions.

Définit la capacité, le contenu initial et les limitations éventuelles sur les débits

Contenu initial fini défini par des attributs flux: litres

	_Quantity	Parfum
0	200	Vanille
1	200	Pêche
2	200	Fruits rouges
3	200	Abricot
4	200	Banane

Répéter l'initialisation si la cuve devient vide

Distinguer les flux

Au moment du mélange yaourt+fruits, nous allons conserver l'attribut Parfum afin de savoir ultérieurement la saveur du yaourt.

Dans le bloc Merge, onglet Attributs flux, désignez le connecteur 1 comme étant celui dont on conserve l'attribut (ici le connecteur 0 n'a pas d'attribut).

Comportement de mélange par défaut pour les attributs flux :

Attributs couche : valeur du conn. n°

Attributs couche chaîne : valeur du conn. n°

Placez ensuite un bloc Lit Attribut entre le bloc Merge et la Valve, afin de repérer quand a lieu un changement de parfum : nous imposerons alors un temps d'arrêt de la ligne pour nettoyage.

Dans le dialogue du bloc, sélectionnez l'attribut Parfum dans le tableau : sa valeur sera lue au passage du flux. Toutefois ce n'est pas le fait de savoir si le parfum est fraise ou abricot qui importe, mais bien de repérer qu'il y a un **changement** de parfum. C'est le connecteur *delta* du bloc que nous utiliserons : il passe de 0 à 1 lorsqu'il note un changement de valeur de l'attribut. Choisissez l'option d'envoyer True puis False, afin que le bloc envoie un signal furtif, et non pas rémanent.

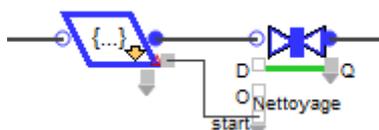
Choisir les attributs à lire

	Nom attribut	Type	Valeur	Heure modif.
0	Parfum	C CHAÎNE		

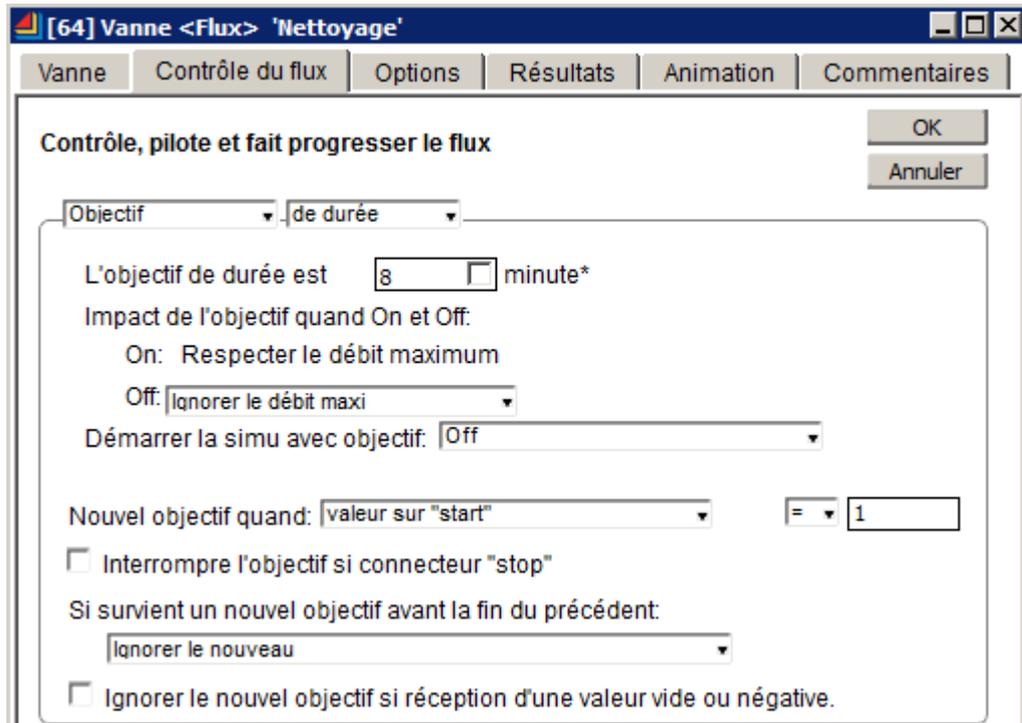
Couche (nouvel attr.)
 C chaîne (nouvel attr.)
Parfum
 Aucun

Si Delta est relié, lorsque la valeur de l'attribut du haut change :

Les deux blocs seront connectés de la manière suivante :

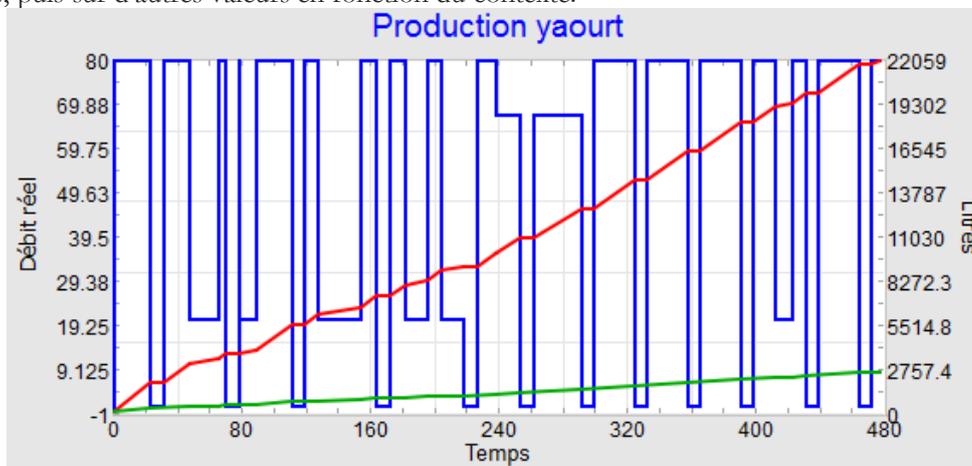


Le connecteur *delta* du bloc Lit attribut est relié au connecteur *start* de la Valve, laquelle sera soumise à un contrôle décrit dans l'onglet Flow Control.



Le contrôle sur la Valve se fait par un objectif de durée. Le nettoyage prévu étant de 8 minutes, c'est la durée fixée pour l'objectif. Un nouvel objectif est lancé lorsqu'une valeur 1 (True) est reçue sur le connecteur *Start* (autrement dit lorsque *delta* signale un changement de parfum). Durant les 8 minutes où l'objectif est 'on', la Vanne est bloquante. Lorsque l'objectif est 'off', la vanne ignore les débits, c'est-à-dire que son impact sur le flux est inexistant (ce sont les blocs alentours qui imposent le débit).

Si l'on mesure le débit en sortie de la Valve, on voit bien les moments où le débit est à zéro pendant 8 minutes, puis sur d'autres valeurs en fonction du contexte.



☞ Les modèles des différentes étapes ainsi que le modèle final figurent dans le répertoire Exemples\Tutorial\Item Rate, dans des version 100% en anglais toutefois.

Aller plus loin

Voici quelques pistes d'amélioration possibles :

- Ajouter une animation et un bloc Pause Sim (bib. Utilities) pour mieux apprécier les évolutions à chaque événement.
- On pourrait ajouter un bloc Shift au Convey Flow, pour arrêter le traitement en fin de journée et vider de son contenu le bloc Convey Flow.
- Les zones de palettisation pourraient inclure un temps de chargement et déchargement.
- Le process de conditionnement pourrait être plus réaliste avec des temps de remplissage et de nettoyage des équipements.

Sources, réservoirs et unités

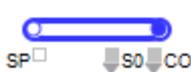
Les blocs Convey Flow, Interchange et Tank sont des blocs de type stockeur – ils ont une certaine capacité et peuvent contenir des quantités de flux. Ils peuvent aussi être “pré-chargés” avec une quantité initiale, servant de source pour le système.

Les unités de flux décrivent ce qui progresse dans un modèle Rate. Des blocs liés par des connecteurs flux et ayant la même unité de flux font partie d’un même groupe unitaire. Le bloc Change Units sert à créer un nouveau groupe unitaire. Les blocs en aval du Change Units sont dans un groupe unitaire différent des blocs en amont.

Ce chapitre examine le stockage du flux et les changements unités de flux, et couvre notamment :

- La capacité d’un bloc flux
- Le contenu initial
- Les indicateurs sur le niveau du flux
- Les unités de temps, de flux et de longueur
- L’emploi du bloc Change Units pour créer des groupes unitaires

Le chapitre s’intéresse spécialement à quatre blocs :



Convey Flow

Retarde le mouvement du flux d’un point à un autre. Peut ou non jouer un rôle d’accumulateur.



Interchange

Représente un stock pouvant interagir avec des entités discrètes. Le bloc a deux comportements : il est zone de chargement si une entité est présente ; ou une cuve de stockage.



Tank

Cuve qui peut être une source, un stockage intermédiaire ou final pour la matière. Le bloc a une capacité et une éventuelle quantité initiale.



Change Units

Change l’unité de flux, produisant ainsi un nouveau groupe unitaire de flux. Le dialogue permet d’entrer le facteur de conversion et la direction du changement.

Capacité

Les blocs Tank, Interchange et Convey Flow sont considérés comme des blocs stockeurs. Cela signifie qu’ils ont une capacité et peuvent contenir de la matière.

La capacité maximum d’un bloc stockeur est spécifiée dans le dialogue, et pour les blocs Tank et Interchange, peut être infinie.

Plein et non plein

Lorsque la capacité d’un bloc stockeur est finie, son état est soit plein, soit non plein. Ce changement d’état a un impact sur les débits réels du modèle :

- Si un bloc stockeur de capacité finie n’est pas plein, son niveau peut s’élever, et par conséquent son débit entrant réel peut être supérieur à son débit sortant réel.
- Si un bloc stockeur de capacité finie est plein, son niveau ne peut s’élever; et par conséquent son débit entrant réel sera inférieur ou égal à son débit sortant réel.

Chaque fois qu’un bloc stockeur de capacité finie change d’état entre plein et non plein, ExtendSim recalcule les débits réels.

☞ Un bloc stockeur de capacité infinie ne peut jamais être plein. Il est assimilable à un bloc stockeur de capacité finie non plein; son débit entrant réel peut être supérieur à son débit sortant réel.

Capacité du Tank

La capacité d’une cuve (Tank) peut être infinie, ou un nombre fini différent de zéro, ou bien zéro.

Capacity: units

Par défaut, la Cuve a une capacité infinie, comme indiqué dans le dialogue. Elle ne sera jamais pleine.

- ☞ Laisser la capacité vide équivaut à cocher infini.
- La capacité maximum d'un Tank peut être modifiée par le champ *Capacity* ou par le connecteur d'entrée continu C (capacité). Si le connecteur C est utilisé, cela prévaut sur le dialogue. Avec une capacité finie, la cuve peut être en état plein ou non plein.
- Si la capacité d'un Tank est zéro, le flux peut transiter par le bloc, mais pas y rester. Dans ce cas, la cuve n'est ni pleine ni non pleine, et le débit entrant réel sera égal à son débit sortant réel.
- ☞ Si la cuve n'est pas liée en sortie, elle est utilisée comme un bassin. Si alors elle atteint l'état plein, son débit entrant sera à zéro pour le reste de la simulation.

Capacité du bloc Interchange

Le flux entre dans l'Interchange non seulement par son connecteur du flux entrant mais aussi par l'arrivée d'une entité. De même, le flux sort du bloc par son connecteur du flux sortant ou par la sortie d'une entité.

Le bloc Interchange a deux options qui affectent son contenu initial et sa capacité maximum :

- **Zone de chargement « Tank only exists when item is in it ».** La cuve n'existe qu'avec une entité. Ce comportement est comparable à un camion (l'entité) qui arrive à un quai de chargement (la cuve) où son chargement s'effectue à un certain débit. Le camion a une certaine capacité, parfois il est déjà partiellement rempli. Tant que le camion est à quai et non plein, il peut se remplir encore.
- **Cuve de stockage « Tank is separate from item ».** La cuve séparée de l'entité. Ce comportement est comparable à un camion (l'entité) qui apporte du produit à un quai de chargement (la cuve) qui peut déjà contenir ou non du produit. Le camion se vide ou même se remplit, plus ou moins complètement. Que le camion soit à quai ou non, le camion comme le quai peuvent contenir du produit. La zone de stockage peut recevoir ou fournir du produit, même s'il n'y a pas de camion à quai.

Zone de chargement

C'est le comportement par défaut. Le bloc Interchange n'a la capacité d'accepter du flux matière que si une entité réside dans le bloc. Sa capacité est alors soit finie soit infinie.

La capacité du bloc est fixée au moment où l'entité entre dans le bloc; elle reste la même jusqu'à ce que l'entité quitte le bloc. La capacité revient alors à zéro. Il ne reste aucune matière.

Capacity (when item is present): units

Pour définir la capacité d'un bloc Interchange dans ce comportement, choisissez une des options du menu affiché à droite.

- Constant. Entrez un nombre (infini par défaut).
- Value at ICO. La valeur sur ce connecteur contrôle la capacité du bloc.
- Value of item attribute. Sélectionnez un attribut. Lorsque l'entité arrive, la valeur d'attribut indique la capacité du Tank.

Dans le modèle Yogurt Production, les blocs Interchange sont paramétrés comme des zones de chargement et leur capacité est une constante de 24.

Cuve de stockage

Avec cette option, le bloc Interchange se comporte comme un Tank – il reçoit la matière de son connecteur du flux entrant, la retient et la libère par son connecteur du flux sortant.

La différence est qu'une entité peut arriver et contribuer, en ajoutant ou soustrayant de la matière, au contenu existant. L'impact de l'entité sur le contenu du bloc est précisé dans la section *Define item behavior* de l'onglet Item/Flow du bloc.

Pour déterminer la capacité du bloc lorsque ce comportement est sélectionné, entrez une valeur dans le champ *Capacity* ou laissez-le vide pour signifier infini.

Capacité du bloc Convey Flow

La capacité maximum d'un bloc Convey Flow est la combinaison de deux facteurs:

- 1) La longueur du bloc et la densité maximum déterminent une capacité *maximisée*. Par défaut,

ExtendSim calcule une capacité maximisée pour le Convey Flow en multipliant sa longueur par sa densité maximum. C'est indiqué dans le champ Capacity par la case à cocher Capacity: maximized. Dans ce cas la capacité maximum et la capacité maximisée sont égales.

- 2) Un nombre dans le champ Capacity de l'onglet Options peut réduire la capacité sous la valeur maximisée. Par exemple, le Convey Flow peut avoir des propriétés structurelles qui limitent le poids qu'il peut transporter. Pour cela, décochez la case et entrez le nombre désiré. Ici, la capacité maximum sera inférieure ou égale à la capacité maximisée.
- ☞ La capacité maximum d'un bloc Convey Flow ne peut jamais dépasser sa capacité maximisée, quelle que soit la valeur saisie dans le champ *Capacity*.

Capacity: maximized units
 Maximized capacity: units

Définir un contenu initial

Ces trois blocs stockeurs (Convey Flow, Interchange et Tank) peuvent ne pas être vides. Le contenu initial d'un bloc stockeur peut être un nombre spécifique ou, pour le Tank et l'Interchange, l'infini.

Une fois donné le contenu initial, il ne change pas durant la simulation. La seule exception est le bloc Interchange quand il est en mode *Tank only exists...* Dans ce cas, chaque entité qui arrive peut établir le contenu initial.

- ☞ Si le contenu initial d'un Tank ou d'un Interchange est l'infini, sa capacité devient automatiquement l'infini.

Vide et non vide

Si le contenu initial d'un Tank ou d'un Interchange est fini, l'état du bloc peut être vide ou non vide. Ce changement d'état a un impact sur les calculs des débits réels :

- Si le Tank ou l'Interchange n'est pas vide, son niveau peut diminuer, et par conséquent son débit sortant réel peut être supérieur à son débit entrant réel.
- Si le bloc est vide, il ne peut fournir plus qu'il ne reçoit, et par conséquent son débit sortant réel sera inférieur ou égal à son débit entrant réel.

Chaque fois qu'un Tank ou un Interchange change d'état entre vide et non vide, ExtendSim recalcule les débits réels.

- ☞ Le bloc Convey Flow a d'autres mécanismes pour calculer un changement d'état entre vide et non vide.

Initialisation du Tank

Le contenu initial de la cuve peut être :

- 0 (par défaut)
- Une valeur saisie
- Infini

- ☞ Si le Tank n'est pas relié en entrée, il est par définition utilisé comme une source. Si cette source atteint l'état vide, son débit sortant sera à zéro pour le reste de la simulation.

Initialisation de l'Interchange

Elle dépend du mode choisi.

Tank only exists... - Ce choix permet d'avoir un contenu initial uniquement lorsqu'une entité arrive. Choisissez une valeur dans le menu :

- Constante. Entrez un nombre (infini par défaut).
- Valeur sur CoE. La valeur sur ce connecteur contrôle la capacité du bloc.
- Valeur d'attribut. Sélectionnez un attribut. Lorsque l'entité arrive, la valeur d'attribut indique la capacité du Tank.

Tank is separate... - Laissez le champ Contenu initial à infini (par défaut) ou entrez un nombre.

Initialisation du Convey Flow

Elle se fait par l'onglet Initialize, où une table permet de personnaliser les divers segments du convoyeur, chacun avec son contenu initial. Chaque ligne de la table représente un segment du convoyeur, d'une densité différente des segments adjacents.

L'onglet Initialize comporte un bouton *Show example* qui place un exemple de paramétrage dans la table, pour vous aider à mieux comprendre comment la remplir. Le convoyeur fait 100 mètres, et la table indique que la densité initiale serait 10 unités de flux par mètre pour le segment qui va de 75 à 100 mètres (un total de 250 unités), et de 5 unités de flux par mètre pour le segment allant de 25 à 50 mètres (un total de 125 unités).

	High Limit	Low Limit	Density	Initial Quantity
0	100	75	10	250
1	50	25	5	125

- ☞ Dans cet exemple, les sections entre 0 et 25 mètres et entre 50 et 75 mètres ne contiennent aucun produit.

Indicateurs

Au cours de la simulation, le niveau du flux dans les blocs stockeurs (Convey Flow, Interchange et Tank) varie dans le temps. Vous pouvez souhaiter avoir une indication de quand le niveau atteint des valeurs particulières. Il est courant de vouloir savoir si le contenu approche ou atteint des seuils remarquables. Par exemple, une procédure d'urgence qui se déclenche si le niveau atteint un seuil « haut », et s'annule lorsque qu'il retrouve un niveau « normal ».

A chaque plage de niveau, on peut affecter un nom, une limite basse et une limite haute. Lorsque le niveau du flux atteint une valeur qui fait changer de plage, le bloc indique sur son connecteur I (indicateur) la valeur, et peut ainsi alerter tout bloc qui est lié à ce connecteur.

- ☞ Alors que les blocs Tank et Interchange indiquent leur niveau courant par leur connecteur I (indicateur), le bloc Convey Flow indique à quelle distance (la *longueur d'accumulation*) se situe le point d'accumulation de la fin du convoyeur. Lorsque la quantité de produit prête à quitter le bloc dépasse la quantité acceptée en aval, le flux s'accumule à partir de la fin du convoyeur.

Définir les indicateurs

Les onglets Indicators des trois blocs stockeurs ont une interface similaire.

Chaque onglet Indicators comporte une table pour spécifier les noms et les plages de valeurs (haute et basse), ainsi qu'une valeur spéciale (un ID pour chaque indicateur) à émettre lorsque le niveau est dans la plage.

Pour créer des indicateurs, entrez vos informations ou cliquez sur le bouton *Show example* pour remplir la table que vous pourrez ensuite modifier. Dans tous les cas, ExtendSim calcule la limite haute en fonction de la limite basse.

- ☞ La ligne du haut spécifie la plage la plus haute; celle du bas la plage la plus basse.
- ☞ Pour ajouter ou supprimer des lignes, utilisez le bouton +/- dans le coin inférieur droit. Par exemple, pour modifier les valeurs d'exemple, changez le nombre de lignes en 0.

	Indicator Name	Low Limit	High Limit	Value to Output
0	Full accum	100 mètres	100 mètres	4
1	High accum	75 mètres	100 mètres	3
2	Medium accum	25 mètres	75 mètres	2
3	Low accum	0 mètres	25 mètres	1
4	No accumul...	0 mètres	0 mètres	0

L'illustration ci-dessus montre le dialogue après avoir cliqué sur le bouton *Show example*. Chaque indicateur correspond à une plage de contenu définie par les limites Haute et Basse pour cette ligne. La colonne High limit est présentée pour plus de clarté seulement, car elle provient du calcul des diverses limites basses.

Sauf si le bloc a une capacité infinie, les limites de l'indicateur peuvent être exprimées en nombres absolus (ci-dessus) ou en pourcentages.

- ☞ Si le bloc a une capacité infinie, les limites doivent être exprimées en nombres absolus. Si son contenu initial est infini, les indicateurs sont désactivés.

Connaître les niveaux

Il y a deux types d'événements qui vont faire qu'un nouvel indicateur est signalé :

- Lorsque le niveau augmente et que le contenu du bloc atteint la Limite basse de l'indicateur suivant.
- Lorsque le niveau décroît et que le contenu du bloc atteint la Limite haute de l'indicateur suivant.

Dans chacun des cas, le nouveau ID est émis en sortie sur le connecteur I, et les blocs liés à ce connecteur sont alertés du changement.

Avec les valeurs de l'exemple, si le niveau de la Cuve passe de 320 à 760 litres, le bloc comparera ce niveau à la limite basse et émettra la valeur 3 sur le connecteur I. Si son niveau descend de 760 à 700 litres, le bloc comparera ce niveau à la limite haute et émettra 2.

	Nom indicateur	Limite basse	Limite haute	Valeur à émettre
0	Plein	1000 litres	1000 litres	4
1	Haut	750 litres	1000 litres	3
2	Moyen	250 litres	750 litres	2
3	Bas	0 litres	250 litres	1
4	Vide	0 litres	0 litres	0

Indicateurs d'exemple d'un bloc Tank

- ☞ La valeur émise sur le connecteur I (indicateur) dépend de si le niveau est ascendant ou descendant, et d'où est le niveau dans la plage où il entre. Le bloc Tank a un connecteur de sortie S (direction status) qui indique si le niveau est ascendant ou descendant.

Modèle Tank Flow Unit

La cuve de Yaourt du modèle Tank Flow Unit émet des valeurs indiquant le niveau du Tank; affichées sur la troisième ligne du graphique.

Unités et groupes unitaires

Cette section s'intéresse à tout ce qui concerne les unités de flux, et notamment le bloc Change Units.

Unités de flux

L'unité de flux indique ce qui circule d'un bloc Rate à un autre. Comme pour les unités de temps ExtendSim, les unités de flux peuvent être génériques (le dialogue du bloc indiquera juste « units ») ou spécifiques. Par exemple, l'unité de flux peut être un paquet, un litre, une transaction, une boîte, etc. Dans l'onglet Options, vous pouvez sélectionner les unités de flux ou en définir de nouvelles. L'onglet *Discrete Rate* du bloc Executive comporte également une section sur les unités de flux, permettant de manière centralisée d'ajouter, supprimer ou renommer des unités.

Unités de bloc

Les blocs Tank et Interchange peuvent avoir une unité de bloc interne, différente de l'unité de flux. C'est une représentation du volume qui n'affecte pas l'unité de flux du groupe unitaire. Si vous sélectionnez une unité de bloc différente de l'unité de flux qui entre et sort du bloc, vous devez saisir un facteur de conversion. Le facteur de conversion représente le ratio de l'unité de bloc par rapport à l'unité de flux.

- ☞ L'emploi des unités de bloc est facultatif ; l'unité de bloc par défaut est l'unité de flux.

Unités de temps

Comme pour un modèle discret classique, des unités de temps spécifiques peuvent être définies pour un modèle, et notamment une unité par défaut. Vous pouvez utiliser une unité de temps locale choisie dans un menu de l'onglet Options d'un bloc Rate. L'unité locale s'applique à ce bloc seulement, sans affecter le groupe unitaire.

Les dates calendaires ne sont pas disponibles si le mois ou l'année a été sélectionnée pour unité de temps globale d'un modèle flux. De même, si Dates calendaires a été sélectionné, les blocs Rate ne pourront pas sélectionner Mois ou Année localement.

Unités de longueur

Par commodité, le bloc Convey Flow permet de nommer une unité de longueur, utilisée pour calculer la vitesse du bloc. Vous pouvez utiliser l'unité générique "length unit" ou déclarer une unité spécifique.

Groupes unitaires

Un groupe unitaire consiste en deux bloc ou plus, liés ensemble et partageant la même unité de flux. En liant le connecteur du flux sortant du premier bloc au connecteur du flux entrant du second bloc du modèle, vous créez un groupe unitaire. À moins de modifier explicitement le groupe unitaire, tous

les blocs ainsi liés partageront la même unité de flux et feront partie du même groupe unitaire. Si une unité de flux est modifiée par un des blocs du groupe unitaire, le groupe unitaire ne change pas mais tous les autres blocs du groupe sont automatiquement mis à jour et adoptent la nouvelle unité de flux.



Pour voir le groupe unitaire, cliquez sur le carré gris à la droite du menu *Flow units* d'un bloc, montré ci-dessus. Tous les blocs de ce groupe unitaire passeront en surbrillance.

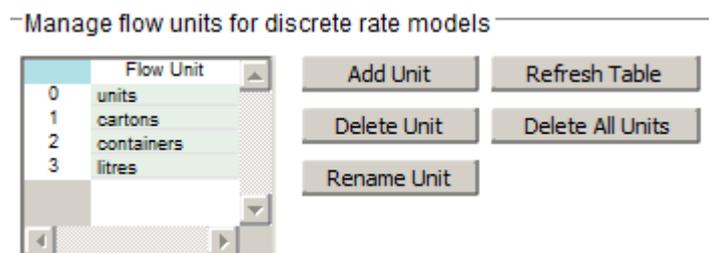
Vous pouvez définir plusieurs groupes unitaires, utilisant différentes unités de flux (les bouteilles dans une section, les cartons dans une autre), au moyen du bloc Change Units.

Où se déclare l'unité de flux ?

Chaque bloc de la bibliothèque Rate permet de sélectionner une unité de flux existante ou d'en créer une nouvelle, via un menu dans l'onglet Options du bloc. Spécifier une unité de flux dans un bloc fait que cette unité est adoptée par tous les blocs du groupe unitaire.

Gérer les unités de flux dans le bloc Executive

L'onglet *Discrete Rate* du bloc Executive comporte une section pour gérer les unités de flux dans un modèle.



La table affiche toutes les unités pour un modèle et permet par des boutons d'ajouter, supprimer ou renommer les unités. Pour utiliser cette fonction, sélectionnez dans la table l'unité de flux que vous voulez modifier, puis cliquez sur le bouton approprié.

Définir des unités de bloc

Nous l'avons indiqué, les blocs Tank et Interchange permettent de définir une unité de bloc différente des unités de flux. Il s'agit d'une représentation interne de volume uniquement, et elle est spécifique aux blocs Tank et Interchange. Cela ne modifie pas les unités de flux ou le groupe unitaire pour le flux entré ou sorti par ces blocs. Si une unité de bloc a été spécifiée, il faut également saisir un facteur pour convertir l'unité de bloc en unités de flux.

Select units for the flow unit group and for this Interchange

Define a flow unit for the group and a block unit for the block

Flow unit: cartons / minute*

Block unit: containers / minute*

Unit factor: 6 containers / cartons

Dans l'onglet Options, sélectionnez *Define a flow unit...* En plus de proposer des champs pour déclarer une unité de flux, cette option propose un champ pour entrer une unité interne au bloc, ainsi que le facteur de conversion entre l'unité de flux et l'unité de bloc.

Changer le groupe unitaire

C'est le bloc Change Units qui permet de créer un nouveau groupe unitaire, en transformant l'unité de flux. Vous pouvez changer l'unité de flux dans de nombreux blocs, et cela modifiera l'unité de flux pour le groupe en entier. Mais pour changer les unités de flux entre un point du modèle et un autre, il faut créer un nouveau groupe par le bloc Change Units. Son connecteur du flux entrant fera partie du groupe unitaire amont, et son connecteur du flux sortant fera partie du groupe unitaire aval.

Le menu *Change units from:* définit l'unité de flux qui entre dans le bloc, le menu *to:* sélectionne ou crée une nouvelle unité de flux. Le bloc comporte un champ pour saisir le facteur de conversion de l'unité

de flux entrante vers l'unité sortante, et un menu permet de sélectionner la direction de la conversion. Le bloc permet également de changer l'unité de temps.

Change the flow unit, resulting in a new unit group

Change units from: litres / minute*

to: containers

Conversion factor: 12 containers / litres

Initial conversion factor:

Show flow unit change on icon

Un exemple de changement d'unités de flux est présenté dans le tutorial Yogurt Production plus haut, où des litres de yaourt sont convertis en pots de yaourt.

Débits, contraintes et mouvements

Limiter le mouvement du flux par des débits et des contraintes

Nous l'avons mentionné, ExtendSim cherche à faire progresser les flux aussi vite que possible. En l'absence de contraintes, le débit peut être quasi infini et le flux aller instantanément d'un bout à l'autre du modèle. Ce serait une erreur de modélisation de n'imposer aucune contrainte au flux. Ainsi certains blocs définiront leurs propres contraintes, et chaque zone d'un modèle comportera ses mécanismes de contrainte critique.

Ce chapitre examine :

- Débits, sections de flux, et zone PL
- Règles de flux observées par un bloc pour la progression du flux
- Les blocs qui spécifient des contraintes critiques
- Répondre au besoin de contrainte critique
- Exemples de contraintes et de sections de flux

☞ La plupart des modèles illustrant ce chapitre figurant dans le répertoire `\Examples\Item Rate\Rates and Constraints`. Les modèles du tutorial figurant dans `\Examples\Tutorials\Item Rate`. Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Débits, sections de flux, et zone PL

Un des aspects le plus important d'un modèle flux est le débit du flux – la vitesse de progression du flux. Il est représenté par le ratio de l'unité de flux par l'unité de temps. Il s'affiche dans le dialogue des blocs par litres/minute, transactions/seconde, boîtes/heure, etc.

Les modèles Rate peuvent être divisés en sections de flux et en zones PL.

Les connecteurs flux à l'intérieur de chaque section de flux ont le même débit réel. La zone PL se compose d'une ou plusieurs sections de flux dont le débit réel pourrait changer durant la simulation.

Types de débits

Les débits suivants sont pris en compte par ExtendSim et par les règles de flux d'un bloc.

Débit maximum

Le débit maximum impose une limite supérieure au mouvement du flux. Six blocs Rate ont la possibilité de définir un débit maximum. Vous pouvez définir un débit maximum explicite dans les blocs Interchange, Tank et Valve. Le débit maximum pour le bloc Convey Flow est calculé mathématiquement. Les débits maximum peuvent aussi être implicites, sous certaines conditions dans les blocs Merge et Diverge.

☞ Le connecteur du flux entrant pour les blocs Convey Flow, Interchange ou Tank peut avoir un débit maximum tandis que le connecteur du flux sortant peut en avoir un autre. Le débit entrant maximum du bloc Convey Flow est dérivé de paramètres du dialogue; le débit sortant maximum est dérivé de paramètres du dialogue et de conditions du modèle. Le débit entrant maximum et le débit sortant maximum des blocs Interchange et Tank peuvent être saisi dans le dialogue.

Pour éviter toute erreur, chaque zone d'un modèle doit comporter un mécanisme restreignant le débit à un nombre inférieur à l'infini. Si ces contraintes minimales n'existent pas, ExtendSim stoppe la simulation et affiche un message d'erreur.

Débit réel

Un des motifs pour créer un modèle flux est de vouloir déterminer le débit réel d'un système. Le débit réel est le résultat de calculs internes qui prennent en compte le débit maximums et toutes les contraintes du process. Dans certaines situations le débit réel est le même que le débit maximum; souvent il est inférieur. Un débit réel est associé à chaque section de flux d'un modèle.

Dans une section de flux, le débit réel ne peut être supérieur au débit maximum de tous les blocs de cette section. Il peut être identique au plus faible débit maximum, ou même égal à zéro (0), selon les conditions du modèle.

Chaque section de flux ne peut avoir qu'un débit réel, mais comporter plusieurs blocs ayant un débit maximum. Il est courant d'avoir plusieurs blocs Valve, chacun avec son débit maximum, dans une section de flux.

Débit infini

Un débit infini est un débit théorique qui ferait que le flux avancerait de manière instantanée d'un point à un autre du modèle. L'onglet *Discrete Rate* du bloc Executive spécifie qu'un débit supérieur ou égal à un certain nombre doit être considéré comme infini. Par défaut, un débit $\geq 1e10$ est considéré comme infini.

— Global options for discrete rate models —

Any rate \geq is considered infinite

Any effective rate \leq is considered to be zero

Vous pouvez modifier ce nombre à votre guise. Cependant, comme il existe une limitation par la précision à 12 chiffres pour les débits réels, choisissez un nombre aussi proche que possible du plus haut débit réel possible qui pourrait être atteint.

Débit réel infini

Comme le mouvement instantané n'est pas possible dans le monde réel, la bibliothèque Rate n'accepte pas de débit réel infini. Le nombre infini spécifié dans le bloc Executive établit une limite haute parmi tous les débits réels. Si la simulation calcule un débit réel égal ou supérieur à ce nombre, ExtendSim stoppe la simulation et affiche un message d'erreur. Cela peut arriver par exemple si une Cuve source est liée directement à une Cuve réceptrice, sans aucune contrainte.

Débit maximum infini

L'infini du bloc Executive (par défaut, $1e10$) peut être utilisé dans des blocs qui n'exercent pas de contrainte sur la vitesse du flux. Par exemple, si vous cochez la case ∞ (infini) pour le débit maximum d'une Valve. Avec un débit maximum infini, la Valve ne limitera pas la progression du flux. Si le champ débit maximum est laissé vide ou avec un nombre $\geq 1e10$, le bloc ne limitera pas non plus la progression du flux.

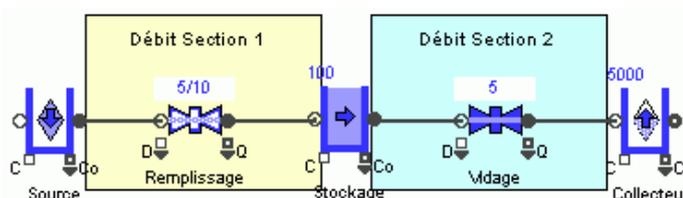
Fourniture amont / demande aval

Ce débit potentiel est considéré lorsque vous utilisez un mode avancé dans les blocs Diverge, Merge et Sensor.

Sections de flux

Une section de flux est un ensemble de blocs liés entre eux et possédant le même débit réel. Chaque section de flux commence par un connecteur du flux sortant et se termine par un connecteur du flux entrant plus loin en aval. Elle contient au moins deux blocs.

- Comme un bloc stockeur (Convoyeur, Interchange ou Tank) peut retenir le flux pendant un certain temps, son débit entrant réel peut être différent de son débit sortant réel.



Comme les limites entre des sections

de flux ne changent jamais, les blocs stockeurs définissent toujours les limites entre deux sections, même si leur débit entrant réel est identique à leur débit sortant réel. Par exemple, le connecteur d'entrée d'un bloc Tank termine une section de flux et son connecteur de sortie en commence une autre, comme ici dans la modèle Tank Constraints.

- Certains blocs passeurs définissent une nouvelle section de flux et d'autres non. Par exemple, une Valve ne fait partie que d'une seule section de flux. Mais par définition débit entrant réel d'un bloc Change Units sera toujours différent de son débit sortant réel, puisqu'il effectue une conversion. Il sera donc à la limite de deux sections de flux.

- ☞ Tous les blocs stockeurs, et certains blocs passants, définissent toujours la limite entre deux sections de flux. Ces limites sont établies de manière interne par ExtendSim en début de simulation; et ne changent pas.

La table ci-dessous indique le rôle de chaque bloc Rate dans la définition d'une section de flux:

Bloc	Définit toujours une section de flux ?	Commentaires
Bias	Non	Le débit entrant réel est identique au débit sortant réel. Le bloc influence le débit réel de sa section.
Catch Flow	Non	Le débit sortant réel est identique au débit reçu réel.
Change Units	Oui	Le débit sortant réel est le débit entrant réel multiplié par le facteur de conversion.
Convey Flow	Oui	Le débit sortant réel peut être différent du débit entrant réel car c'est un bloc stockeur.
Diverge	Oui	Les débits réels peuvent être différents suivants les connecteurs- l'entrée et tous les sortant.
Interchange	Oui	Voir commentaire pour Convey Flow
Merge	Oui	Voir commentaire pour Diverge
Sensor	Non	Débit réel entrant et sortant sont identiques.
Tank	Oui	Voir commentaire pour Convey Flow
Throw Flow	Non	Le débit entrant réel est identique au débit envoyé réel.
Valve	Non	Débit réel entrant et sortant sont identiques.

La précision mathématique pour le débit réel est limitée à 12 chiffres.

Zone PL

Alors que les sections de flux ne changent pas après le début d'une simulation, les limites de la zone PL se modifie dynamiquement durant la simulation. Une zone PL se compose d'une ou plusieurs sections de flux associées parce que leurs débits réels peuvent changer durant la simulation. Lorsque survient un événement qui fait que le débit réel d'une section de flux est réévalué, ExtendSim détermine quelles autres sections de flux peuvent être affectées. Les sections de flux affectées constituent la zone PL et font partie du calcul PL. Ceci sera discuté en détail plus loin.

Règles de flux

Les règles de flux définissent comment un bloc permet au flux d'avancer durant la simulation. Lors du calcul des débits, ExtendSim essaie de maximiser l'avancement du flux, soumis à diverses contraintes. L'ensemble des règles de flux des blocs définit la vitesse de progression du flux.

Les règles de flux d'un bloc dérivent de quatre facteurs :

- le comportement du bloc.
- la configuration du dialogue.
- l'utilisation des connecteurs d'entrée continus.
- les liens sur les connecteurs flux. Un bloc Tank est une source si seul son connecteur de sortie est lié, c'est un bassin si seul son connecteur du flux entrant est lié.

Ces règles de flux sont utilisées par le bloc Executive pour tous les blocs d'une zone PL pour calculer les nouveaux débits réels de la zone.

Contraintes

Il existe deux types de règles de flux : les contraintes critiques et les contraintes relationnelles.

Contraintes critiques

Alors que toutes les règles de flux coopèrent pour contraindre le débit, certains blocs ont des règles que l'on appelle contraintes critiques. Si une section de flux contient une ou plusieurs contraintes

critiques, cela impose une limite supérieure au débit des blocs de cette section.

Une contrainte critique est **inconditionnelle** – quoi qu'il adienne durant la simulation, le débit réel ne pourra être supérieur à la contrainte critique minimum des blocs de la section de flux. Par exemple, le champ Maximum Rate: d'une Valve est une contrainte critique. Si cette valeur est la plus basse contrainte critique de la section de flux, le débit réel pour n'importe quel bloc de la section ne pourra pas être supérieur.

Les blocs capables de définir une contrainte critique sont : Convey Flow, Diverge, Interchange, Merge, Tank et Valve ; la Valve est le plus habituellement utilisé.

Contraintes relationnelles

Les contraintes relationnelles définissent comment les débits réels des différentes sections sont associés, créant par là des dépendances entre sections de flux. Par exemple, la contrainte relationnelle entre une section de flux (débit réel x) et une autre section de flux (débit réel y), peut être définie sous la forme $x \geq y$, $x = y$, $2x - 3 = y$ ou toute autre expression. Les contraintes relationnelles sont mises à jour lorsque le bloc réagit à un nouveau paramètre ou à un changement d'état, mais n'affectent pas les contraintes critiques.

Un exemple de contrainte relationnelle est le bloc Change Units, où l'emploi d'un facteur de conversion fait que le débit sortant réel est différent du débit entrant réel. Le bloc Change Units définit la limite entre deux sections de flux; le facteur de conversion spécifie la relation entre deux débits réels.

— Effective rate —

Inflow rate: litres / minute*

Outflow rate: containers / minute*

☞ Vous ne saisissez pas les contraintes relationnelles, elles sont déterminées par le comportement des blocs.

Comparaison de contraintes

Certains blocs peuvent définir une contrainte critique, d'autres une contrainte relationnelle, et d'autres les deux.

- Les blocs pouvant définir une contrainte critique sont : Convey Flow, Diverge, Interchange, Merge, Tank et Valve.
- Les blocs pouvant définir des contraintes relationnelles sont : Change Units, Convey Flow, Diverge, Interchange, Merge et Tank.

Par exemple, un Tank où le *Maximum Inflow Rate* et le *Maximum Outflow Rate* sont cochés impose des contraintes critiques aux flux entrant et sortant. Si aucune de ces cases n'est cochée, la cuve n'aura aucune contrainte critique mais pourrait avoir des contraintes relationnelles. Si la cuve a une capacité finie mais n'est ni pleine ni vide, elle ne donne pas de contraintes sur le flux. Si elle atteint son état Plein, son débit entrant doit devenir inférieur ou égale à son débit sortant ; et ceci est une contrainte relationnelle.

Le débit réel pour une section de flux ne peut être supérieur à la plus basse contrainte critique définie pour l'un des blocs de la section. De plus, comme l'ensemble des règles de flux comporte généralement des contraintes relationnelles, le débit réel pour la section peut varier entre zéro et la plus basse contrainte critique.

Définir une contrainte critique

- Vous pouvez explicitement définir une contrainte critique par les blocs Valve, Tank et Interchange.
Vous entrez un débit maximum dans le dialogue, lisez une valeur de débit maximum sur un connecteur, ou associez le champ débit maximum à la valeur d'une matrice globale ou à la base de données ExtendSim.
- Pour le bloc Convey Flow, la contrainte critique n'est pas saisie, mais dérive de paramètres du dialogue.
- Une contrainte critique peut être indiquée implicitement dans certaines conditions par les blocs Merge et Diverge.

Valve

Le bloc Valve (Vanne) sert souvent à imposer une contrainte critique. Vous entrez une valeur dans le champ *Maximum rate*, lisez une valeur de débit maximum sur le connecteur R, ou associez le champ débit maximum à la valeur d'une matrice globale ou à la base de données ExtendSim.

Vous pouvez aussi changer dynamiquement le débit maximum durant une simulation :

- Liez le connecteur d'entrée R (débit maximum). Cela prévaut sur toute valeur entrée dans le champ *Maximum rate*. Ce champ affichera alors le débit maximum courant.
- Associez le champ *Maximum rate*: à une BDD ExtendSim ou à une matrice globale.

Define constraints

Maximum rate: units / minute*

Initial maximum rate: units / minute*

Poll constraint only each: minute*

- ☞ Les cases "Initial Maximum rate" et "Poll constraint only each" (Interroger la contrainte chaque), ne sont utilisées que lorsque le débit maximum de la Vanne est configuré pour changer dynamiquement.

Initialiser le débit maximum

Lorsque le débit maximum d'une vanne change dynamiquement, la case *Maximum rate initial* joue un rôle important. Le premier calcul de débit réel dans une simulation a lieu juste avant de premier pas de temps de la simulation. Si le connecteur R est relié ou si le champ *débit maximum* a un lien dynamique, des problèmes peuvent survenir à cette étape car ni le bloc relié à R ni la source de donnée liée n'a pu fournir de valeur initiale. La case *Initial maximum rate* résout ce problème par l'initialisation du débit maximum.

La valeur initiale entrée dans le dialogue sera utilisée tant que la Valve n'obtient pas une valeur différente de son connecteur d'entrée D ou d'une autre source.

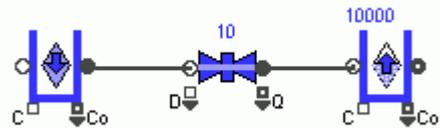
- ☞ Lors de simulations multiples, la case *Initial maximum rate* empêche que le dernier débit maximum serve de débit maximum initial pour la simulation suivante.

Interroger les contraintes

La case à cocher *Poll constraint only each...* peut être utilisée lorsque le débit maximum d'une Valve est configuré pour changer dynamiquement. Cette option indique à la Valve d'aller chercher une nouvelle valeur de débit maximum à intervalles fixes durant la simulation. La Valve ira périodiquement interroger le bloc lié à son connecteur R (débit maximum) ou toute autre source dynamique. Les valeurs reçues entre les requêtes seront ignorées.

Cette case à cocher est optionnelle lorsque le champ débit maximum est associé à un nombre fixe dans une source de données. Elle est requise si la source contient un nombre aléatoire ou si le connecteur d'entrée R est lié à un bloc passif comme le Random Number (bib. Value), car un bloc passif ne va pas de lui-même générer une valeur de maximum.

- ☞ Chaque fois que le débit maximum d'une Vanne change, les débits réels doivent être recalculés sur plusieurs sections. Si vous interrogez la contrainte dans plusieurs blocs Valve, il est bon qu'ils soient mis à jour en même temps. Cela réduira beaucoup le nombre de recalculs.



Contrôler quand et comment la Vanne applique son débit maximum

L'onglet Flow Control de la Valve comporte des options avancées pour gérer comment et quand un bloc applique son débit maximum. En définissant un objectif ou en utilisant une hystérésis, vous contrôlez explicitement quand observer les contraintes et quand les ignorer.

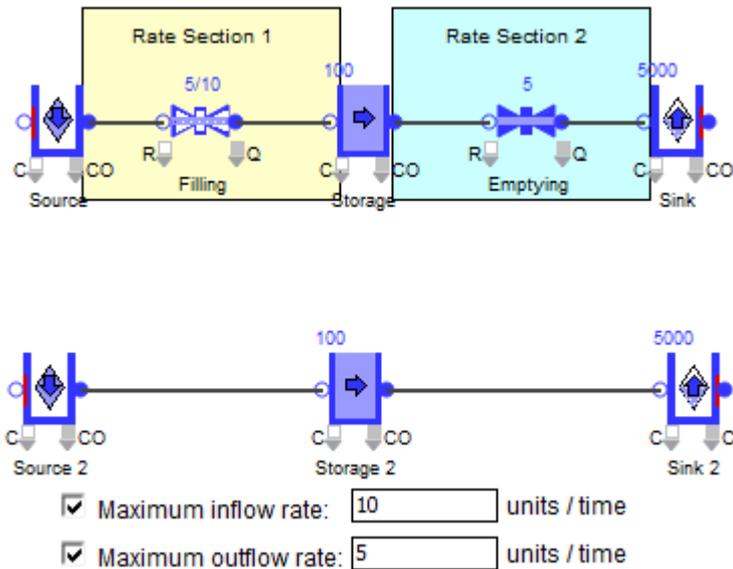
Tank et Interchange

La Tank et l'Interchange permettent de définir explicitement le débit entrant et sortant maximum. À la différence du débit maximum d'une Valve, ces contraintes ne changent pas dynamiquement durant la simulation.

Vous entrez soit un débit entrant maximum soit un débit sortant maximum, soit les deux. L'exemple

Tank Constraint présente deux flux de comportement identique. Dans le flux du bas, la Cuve 2 utilise les options *Maximum Inflow Rate* et *Maximum Outflow Rate* pour remplacer les vannes de remplissage et vidage du flux du haut.

- ☞ Au lieu d'utiliser une Valve pour limiter le flux, définir un débit entrant maximum et/ou un débit sortant maximums dans un bloc Tank ou Interchange suffit pour les besoins de contrainte d'un modèle.

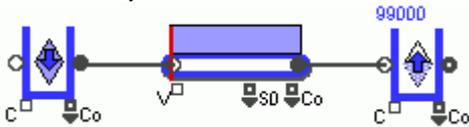


Modèle Tank Constraint

Convey Flow

Le Convoyeur de la bib. Rate calcule des contraintes critiques pour ses connecteurs flux entrant et sortant séparément. Les contraintes critiques dérivent de conditions du modèle et de paramètres dans le dialogue.

La contrainte critique **en entrée** pour le Convey Flow est calculée en multipliant la vitesse réelle du bloc par sa densité maximum.



- ☞ La vitesse réelle peut être inférieure ou égale à celle définie dans le dialogue. Si le bloc ne peut accumuler et que son contenu dépasse la demande aval, la vitesse réelle sera inférieure à la vitesse saisie.

La contrainte critique pour le **flux sortant** du bloc est le résultat de la multiplication du paramètre de vitesse par la densité de flux présente sur le connecteur de sortie du bloc.

- ☞ L'initialisation du contenu initial et de la capacité d'un Convey Flow est examinée plus loin.

Merge et Diverge

La contrainte critique sur une branche ou davantage d'un bloc Merge ou Diverge peut être spécifiée de manière implicite sous certaines conditions. Souvent, le résultat sera un débit de 0 (zéro).

Lorsqu'un bloc Merge ou Diverge est dans certains modes, le flux peut être bloqué sur un ou plusieurs de ses branches. Par exemple, si une branche d'un bloc Diverge en mode Distributionnal reçoit une valeur blank ou ≤ 0 , le flux n'avance plus par cette branche. De même, le flux qui n'est pas sur la branche sélectionné est arrêté si le bloc Merge est en mode Select. Dans ces deux cas, le débit maximum sera de 0 pour les branches affectées.

Répondre au besoin de contrainte critique

Nous avons vu qu'au moins une des sections de flux dans la zone PL doit comporter une contrainte critique pour limiter le flux.

- ☞ Par définition, les blocs stockeurs sont toujours aux limites de deux sections de flux. La règle générale est qu'il faut au moins une contrainte critique entre deux blocs stockeurs. Les contraintes critiques seront fournies par les blocs Convey Flow, Diverge, Interchange, Merge, Tank et Valve. Les blocs stockeurs sont Convey Flow, Interchange ou Tank. On trouve des exceptions à ces règles dans des situations où un bloc Merge ou Diverge se trouve entre deux blocs stockeurs.

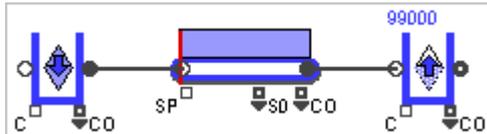
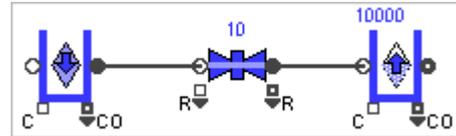
Les exemples suivants montrent comment répondre au besoin de contrainte critique dans des modèles Rate.

Valve ou Convey Flow

Le modèle *Tank Constraint* illustre une contrainte critique entre deux blocs stockeurs (ici, des cuves) qui n'ont pas de bloc Merge ou Diverge entre eux.

L'exemple utilise une Valve pour contraindre le flux entre les deux blocs Tank. C'est le plus courant dans cette situation banale. Pour que le débit maximum de la Valve impose une contrainte critique, il doit être:

- Supérieur ou égal à 0 (zéro)
- Inférieur à $1e10$ (le débit infini)
- Non vide



L'exemple à gauche utilise un convoyeur pour la contrainte critique. Le Convey Flow dérive la contrainte critique pour son flux entrant des paramètres du dialogue et la contrainte critique pour le flux sortant provient des paramètres du dialogue et de conditions du

modèle. Comme il existe une contrainte critique sur les connecteurs de flux entrant et sortant, le Convey Flow limite le débit de la première cuve à la seconde à un nombre inférieur à l'infini.

Tank ou Interchange

Au lieu d'utiliser une vanne pour fournir la contrainte critique entre deux blocs stockeurs, vous pouvez spécifier les débits entrant et sortant maximums par un bloc Tank ou Interchange. Avec ces débits maximums, le Tank ou l'Interchange limiteront le débit entre les deux blocs stockeurs à un nombre inférieur à l'infini. C'est présenté dans le modèle *Tank Constraint* examiné plus haut.

Merge ou Diverge

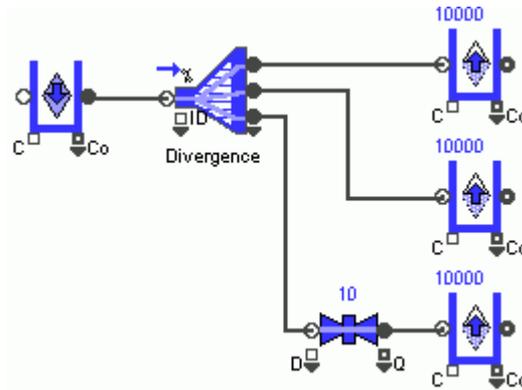
Si un bloc Merge ou Diverge figure entre deux blocs stockeurs, les branches entrante et sortante peuvent ne pas requérir de contrainte critique.

- ☞ Quel que soit le mode d'un Merge/Diverge, si une contrainte critique a été placée sur la branche sortante du bloc Merge, aucune contrainte critique n'est requise sur ses branches entrantes. De même, une contrainte critique sur une branche d'entrée d'un bloc Diverge signifie qu'aucune contrainte critique n'est requise sur ses branches sortantes. Si ces contraintes n'existent pas, le mode du bloc dictera comment placer la contrainte critique.

Le tableau suivant présente brièvement les corollaires de chaque mode vis-à-vis des contraintes critiques lorsque ni la branche de sortie d'un bloc Merge ni la branche d'entrée d'un bloc Diverge n'a de contrainte critique. Dans ce tableau, l'expression "branche variable" signifie une branche d'entrée pour le bloc Merge ou une branche sortante pour un bloc Diverge. Note: pour le mode Proportional, la branche variable portant la contrainte critique ne doit pas avoir une proportion ≤ 0 . Sinon, cette branche sera fermée et les autres branches variables auront potentiellement de débits réels infinis.

Mode Proportional

La section du haut du modèle Minimum Value présente les besoins de contrainte critique lorsqu'un bloc Merge ou Diverge est en mode Proportional. Si le bloc est situé entre deux blocs stockeurs, une seule contrainte critique est requise tant que la proportion de la branche n'est ni 0 ni vide. (Cette contrainte minimale est une exception à la règle générale énoncée plus haut.) Les débits réels des autres branches sont déduits du débit maximum de la Valve.

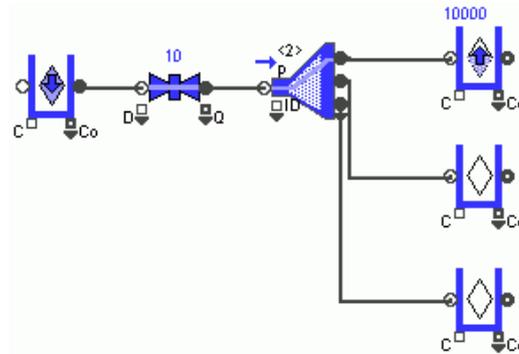


Une seule contrainte requise

Dans l'exemple *Valve Minimum*, une Valve est placée sur la branche de sortie du bas d'un bloc Diverge, et cette branche n'a pas de proportion 0 ou vide.

Mode Priority

La section du bas du modèle *Valve Minimum* présente les besoins de contrainte critique lorsqu'un bloc Merge ou Diverge est en mode Priority. Dans ce cas, le nombre de contraintes critiques à placer sur les branches entre les blocs stockeurs dépend de l'emplacement des contraintes. Leur situation est présentée dans le modèle *Valve Minimum*.



Pas de contrainte sur chaque branche sortante

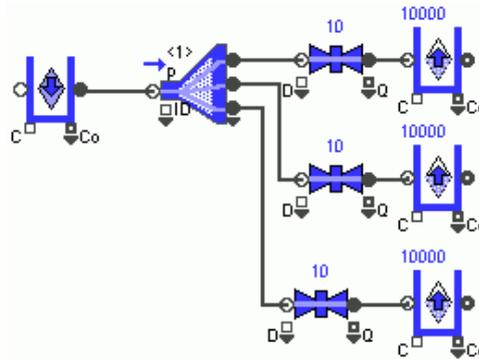
Mode	Besoin de contrainte critique s'il n'y a pas de contrainte critique sur la branche non-variable
Batch/Unbatch	Une seule des branches variables
Distribution	Chaque branche variable
Neutral	Chaque branche variable
Priority	Chaque branche variable
Proportional	Une seule des branches variables (Voir Note plus bas)
Select	Chaque branche variable
Sensing	Chaque branche variable

Si une contrainte critique est placée entre un bloc stockeur et une branche d'entrée d'un bloc Diverge, il n'est pas nécessaire de placer d'autre contrainte critique en sortie du bloc Diverge. De même, si vous placez une contrainte critique sur une branche de sortie d'un bloc Merge, il n'est pas nécessaire de placer d'autre contrainte critique en entrée du bloc. On le voit ci-contre, où une Vanne avec un débit maximum supérieur ou égale à 0 mais inférieur à $1e10$ (le débit infini) est sur une branche d'entrée d'un bloc Diverge et où aucune contrainte critique n'est requise sur les branches sortantes.

Si vous ne placez pas de contrainte critique en entrée du bloc Diverge, vous devez placer au moins une contrainte critique sur chaque branche sortante. De même, si vous ne placez pas de contrainte critique sur une branche de sortie d'un bloc Merge, vous devez placer au moins une contrainte

critique sur chaque branche entrante.

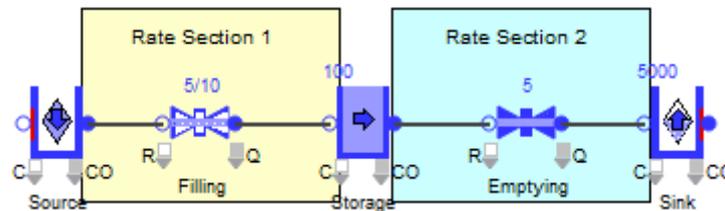
On le voit ci-dessous, où il n'y a pas de contrainte critique sur la branche d'entrée du bloc Diverge. Cela signifie que chaque branche sortante doit avoir une contrainte critique, dans ce cas une Valve avec un débit maximum supérieur ou égale à 0 mais inférieur à $1e10$ (le débit infini).



Contrainte sur chaque branche

Exemple complet

L'exemple qui suit illustre plusieurs concepts examinés dans ce chapitre. La partie haute du modèle *Tank Constraint*, ci-après, comporte deux sections de flux, deux contraintes critiques et une contrainte relationnelle.



☞ Les sections qui suivent utilisent l'abréviation FPT pour indiquer "unités de flux par unité de temps".

Sections de flux

Les sections de flux sont déterminées en interne via une communication entre les blocs Rate et le bloc Executive (bib. Item). Les limites entre sections de flux sont établies en début de simulation et ne changent plus même si les débits évoluent.

En début de simulation:

- La Valve Filling a un débit maximum de 10, reçoit le flux d'une Source infinie, et l'envoie dans un Tank vide d'une capacité de 100 unités de flux. Le débit calculé est un débit de 10 FPT pour la Valve Filling en début de simulation. (Cela changera quand le Tank sera plein.)
- La Cuve Storage a une capacité de 100 unités de flux et renvoie le flux vers une Valve avec un débit maximum de 5 FPT. En début de simulation, son débit entrant réel est de 10 FPT et son débit sortant réel est de 5 FPT.
- La Valve Emptying a un débit maximum de 5 FPT et envoie son flux à un collecteur infini. Son débit réel entrant et sortant est de 5 FPT.

En début de simulation, le Tank Storage n'a pas les mêmes débits en entrée et sortie. La première section de flux du modèle commence au connecteur du flux sortant du bloc Source et se termine au connecteur du flux entrant du Tank Storage. La seconde section de flux commence là et se termine au connecteur du flux entrant du collecteur.

Contraintes critiques

Il y a deux contraintes critiques dans la partie haute de ce modèle. La première contrainte critique est 10 FPT dans le champ débit maximum de la Valve Filling. La seconde est 5 FPT dans le champ débit maximum de la Valve Emptying.

Contraintes relationnelles

Les contraintes relationnelles définissent les relations entre les débits réels des différentes sections. En début de simulation il n'y a pas de contraintes relationnelles – le débit entrant réel est indépendant du

débit sortant réel. Lorsque le Tank (de capacité finie de 100 unités de flux) devient plein, il impose une contrainte critique: le débit entrant doit devenir inférieur ou égal au débit sortant.

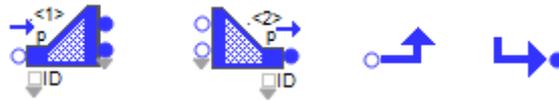
Impact de la simulation sur les débits réels

Comme il est vide au début, les règles de flux initiales du Tank Storage n'imposent aucune restrictions à son débit entrant. Par conséquent, le débit réel initial dans la Section 1 n'est limité que par la contrainte critique de 10 FPT de la Valve Filling. Ce débit réel initial de la première section de flux est temporaire car le débit réel de la Section 2 n'est que de 5 FPT et que la cuve va se remplir. Donc le Tank Storage introduit une contrainte relationnelle imposant que le débit entrant réel (Section 1) soit inférieur ou égal à son débit sortant réel (Section 2).

Orientation du flux

Emploi des blocs Merge, Diverge, Throw Flow et Catch Flow

Pour orienter le flux ou ses parties vous utiliserez les blocs Diverge, Merge, Throw Flow et Catch Flow.



Les blocs Merge et Diverge ont une interface et des possibilités similaires. Ils reçoivent ou envoient le flux par un nombre variable de connecteurs de flux entrant et sortant. Leurs dialogues proposent des options pour mélanger ou faire diverger les flux. Les blocs Throw Flow et Catch Flow servent à envoyer le flux d'un point à un autre, à distance.

Le chapitre examine :

- Des blocs pour mélanger, séparer et orienter les flux
 - Les modes des blocs Merge et Diverge
 - Autres fonctions des blocs Merge et Diverge
 - Emploi des blocs Throw Flow et Catch Flow
- ☞ Les modèles utilisés dans ce chapitre figurant dans le répertoire \Exemples\Item Rate\Merge and Diverge. Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Mélanger ou séparer les flux

Les blocs Merge et Diverge servent à mélanger ou séparer les flux, selon une logique définie par des « modes » dans le dialogue, et que les deux blocs possèdent en miroir. La liste des sept modes et de leurs caractéristiques est résumée dans “Tableau des modes”, ci-dessous.

- ☞ Nous appellerons branche d'entrée ou branche entrante chaque connecteur d'entrée du bloc Merge. De même, branche de sortie ou branche sortante, chaque connecteur de sortie du bloc Diverge. Collectivement nous les appellerons branches variables, car leur nombre n'est pas forcément de deux.

Tableau des modes

Mode	Somme des entrées = somme des sorties ?	Règle fixe ?	Ordre de préférence requis ?	Valeur sur une branche qui bloquerait le flux	Compatible avec le mode Réactif?
Batch/Unbatch	Non	Oui	Non	Aucune	Oui
Distributional	Oui	Non	Oui	vide, <=0	Parfois
Neutral	Oui	Non	Non	Aucune	Parfois
Priority	Oui	Non	Oui	vide	Parfois
Proportional	Oui	Oui	Non	vide, <=0	Oui
Select	Oui	Oui	Non	Aucune	Oui
Sensing	Oui	Non	Oui	vide, <=0	Oui

Caractéristiques

- Certains modes utilisent une règle fixe – peu importe ce qui se passe dans le reste du modèle,

cette règle sera observée. Avec d'autres modes, les règles de flux expriment une *préférence* et ne sont invoquées que dans des situations spécifiques fonction des conditions du modèle.

- Des demandes simultanées pour le flux entre des blocs Merge et Diverge en modes Distributional, Priority et Sensing doivent faire appel au rang de *préférence*, cf. plus loin.
- Certaines valeurs peuvent bloquer en flux en certaines circonstances (par exemple, un nombre hors limites), mais dans certains modes, une valeur vide ou zéro (0) stoppera toujours le flux. Les valeurs hors limites causeront un message d'erreur; un zéro (0) ou vide ne générera pas un message d'erreur.
- Certaines incompatibilités peuvent surgir dans une zone du modèle si des blocs utilisent le mode Sensing et d'autres blocs les modes Distributional, Neutral ou Priority. Ces situations doivent être évitées.

Mode Select

Lorsque les blocs Merge ou Diverge sont en mode Select, seule la branche sélectionnée est ouverte à un instant donné. Une table dans le dialogue du bloc permet d'affecter un ID unique à chaque branche entrante (Merge) ou branche sortante (Diverge). Le connecteur ID du bloc est alors utilisé pour sélectionner la branche passante. Si la valeur sur le connecteur ID ne correspond à aucune branche, vous choisissez le comportement :

- Choisir la branche du haut
- Choisir la branche du bas
- Stopper le flux
- Générer une erreur

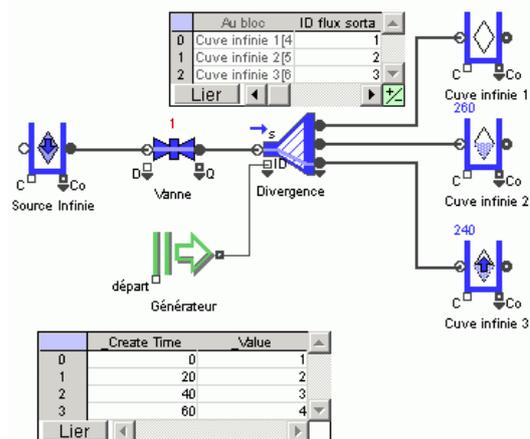
Une valeur vide reçue sur le connecteur ID stoppera toujours le flux, jusqu'à recevoir une valeur valide.

☞ Le mode Select utilise une règle fixe.

Modèle Select mode Diverge

Dans le modèle *Select Mode Diverge*, un bloc Create émet une valeur séquentielle (1, 2, 3 ou 4) toutes les 20 unités de temps. Le bloc Diverge reçoit cette valeur sur le connecteur ID, la compare aux valeurs dans la table et choisit le connecteur du flux sortant correspondant. Trois blocs Cuves, chacun de capacité infinie, sont liés au Diverge. Ici le bloc Create contrôle le bloc Diverge.

Remarquez que 4 est un nombre incorrect et que le bloc Diverge est paramétré de sorte à stopper le flux. À l'envoi de la valeur 4, tout s'arrête et une barre rouge apparaît à la droite du bloc.



Modèle Select Mode Merge

Ce modèle est le miroir du modèle précédent. Trois sources fournissent la matière, et le bloc Merge choisit une des sources. Le Merge est en mode Select, et le bloc Create envoie les mêmes quatre valeurs sur le connecteur ID. Ici, une valeur incorrecte sur ID choisit la branche du haut.

Mode Batch/Unbatch

Ces modes servent à causer un débit sortant total différent de la quantité entrante totale, ou inversement.

- Lorsque le bloc Merge est en mode Batch, chaque unité de flux de chaque branche en entrée est combinée en une unité de flux sortant. Le débit réel de chaque branche en entrée est le même que celui du connecteur de sortie. Dans ce mode, le comportement du bloc Merge est similaire à celui du bloc Batch (bib. Item).
- Lorsque le bloc Diverge est en mode Unbatch, chaque unité de flux de chaque branche en entrée est clonée en une unité de flux pour chaque branche sortante. Le débit réel du connecteur du flux entrant est le même que celui de chaque branche sortante. Dans ce mode, le comportement du bloc Diverge est similaire à celui du bloc Unbatch (bib. Item).

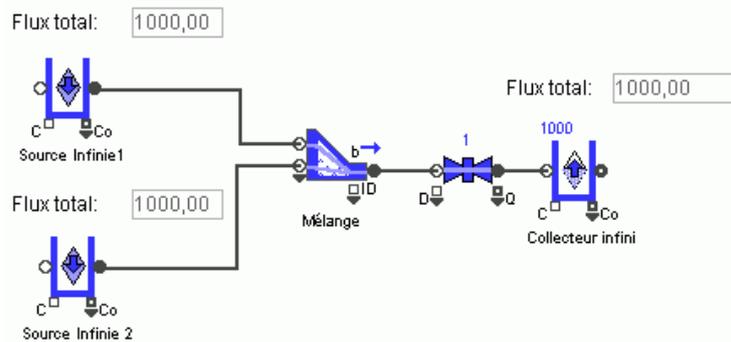
- Les modes Batch/Unbatch sont différents des autres modes car le flux entrant total n'est jamais le même que le flux sortant total.

Modèle Batch Mode Merge

Dans le modèle Batch Mode Merge, à chaque unité de temps le bloc Merge prend une unité de flux d'une Source 1 et une unité de flux d'une Source 2. Il en fait une unité de flux émise en sortie. Les deux Sources fournissent chacune 1,000 unités de flux, mais le bloc Merge ne produit que 1,000 unités de flux, et non la somme des sources.

Modèle Unbatch Mode Diverge

Ici c'est l'inverse : une unité de flux devient deux unités de flux – l'une destinée au Réservoir 1 et l'autre destinée au Réservoir 2. La quantité de flux a doublé.



Modèle Batch mode Merge

Mode Proportional

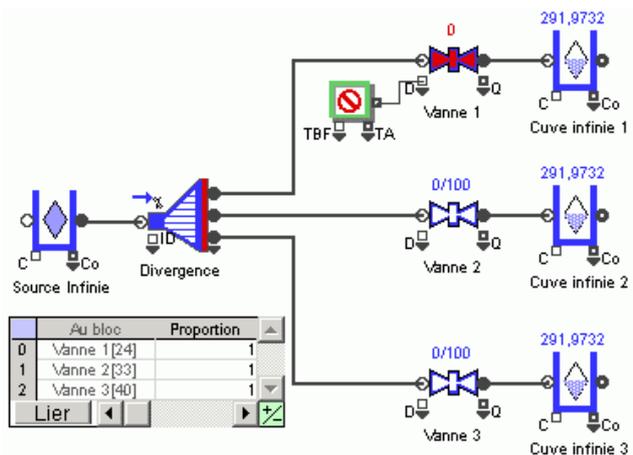
Avec le mode Proportional, vous décrivez dans une table les proportions requises pour chaque branche, de manière relative aux autres branches. Par exemple, une valeur de 2 pour la branche sortante supérieure et de 4 pour la branche sortante inférieure indique que la branche du bas reçoit deux fois plus que la branche du haut. Si le proportion sur une branche est définie à vide ou ≤ 0 , le débit réel de la branche passe à 0 et le flux est stoppé sur cette branche.

Ce mode utilise une règle fixe où le débit réel sur chaque branche doit correspondre à la proportion définie dans la table. Par conséquent, si le flux sur l'une des branches est bloqué ou manquant, les débits réels de toutes les branches passeront à zéro et le flux sera bloqué.

Modèle Proportional Mode Diverge

Dans cet exemple, le flux provenant d'une Source est équitablement distribué entre les trois branches sortantes du bloc Diverge. Les proportions de la table sont à 1:1:1. Avec cette proportion, le débit réel sur les trois branches doit être le même – la même quantité de flux doit passer par chaque branche.

Les trois Vannes imposent un débit de 100. Mais par les arrêts du bloc Arrêts, le débit alterne entre 0 et 100 durant la simulation, ce qui influence le débit réel des trois branches. Quand la Vanne 1 passe à 0, les trois branches passent à 0 quand bien même les Vannes 2 et 3 ont encore un débit de 100. Car le bloc Diverge doit respecter le ratio de 1:1:1.



Modèle Proportional Mode Merge

Ce modèle est le miroir du modèle précédent. Les vannes contrôlent le flux entrant, le bloc Arrêts force la Vanne 1 à alterner entre 0 et 100. Quand la Vanne 1 passe à 0, les trois branches passent à 0 car le bloc Merge doit respecter le ratio de 1:1:1.

Mode Priority

Le mode Priority permet d'affecter des priorités aux branches (entrantes pour Merge, sortantes pour Diverge). Ces priorités n'agissent que sur les débits réels affectés aux branches lorsqu'un écart survient entre la fourniture amont et la demande aval. Sinon elles sont ignorées.

- Dans le cas du bloc Diverge, lorsque la fourniture amont est supérieure ou égale à la demande aval, le bloc fait passer par chaque branche autant que la demande aval le permet, et les priorités sont ignorées. Mais lorsque la demande aval excède la fourniture amont, les priorités données à chaque branche servent à calculer les débits réels appropriés pour les branches sortantes.
- Le bloc Merge fait passer par chaque branche entrante autant que possible quand la demande aval excède la fourniture amont, et ignore les priorités. Mais lorsque la fourniture amont excède la demande aval, les priorités données à chaque branche servent à calculer les débits réels appropriés pour les branches entrantes.

Il faut connaître des cas spéciaux dans l'emploi du mode Priority dans un bloc Merge ou Diverge:

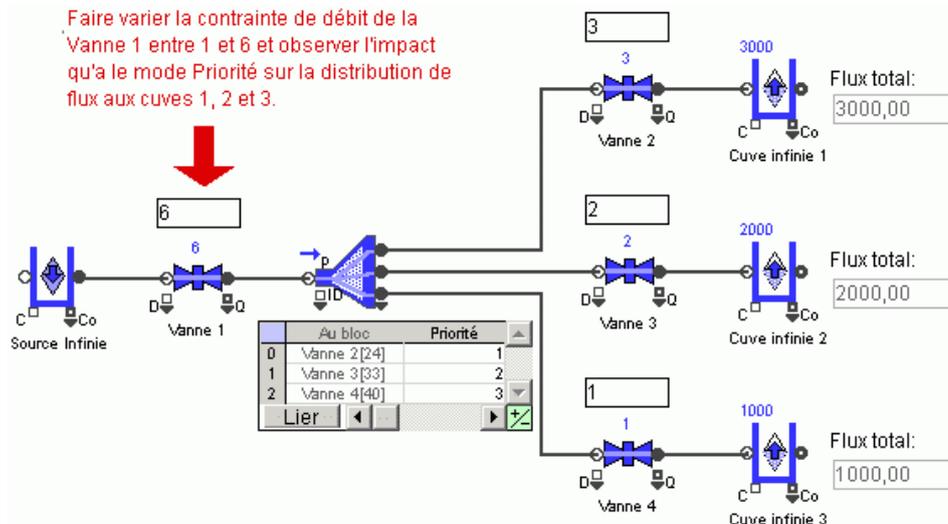
- Si la priorité d'une branche est vide, le débit réel de cette branche sera zéro et le flux arrêté sur la branche.
- Si les priorités de deux branches ou plus sont égales, le flux se divisera entre elles comme en mode Distributional, en proportions égales.

☞ Les priorités d'un bloc Diverge ou Merge ne sont pas des règles fixes, mais contextuelles, puisqu'elles ne servent que lorsqu'il y a un écart entre demande aval et fourniture amont.

Les blocs Merge/Diverge en mode Priority ne sont pas toujours compatibles avec les blocs Merge/Diverge en mode Sensing. Eviter une zone du modèle avec des blocs en mode Sensing et d'autres en mode Priority.

Modèle Priority Mode Diverge

Dans cet exemple, les débits des vannes sont tels que la fourniture amont de 8 unités de flux de la Vanne 1 dépasse la demande aval de 6 (Vannes 2, 3 et 4). En l'état, les priorités sont ignorées.



Si l'on réduit la fourniture en passant la Vanne 1 de 8 à 4, le bloc Diverge va calculer des débits réels qui vont distribuer le flux trop rare selon les priorités. Les priorités étant en ordre descendant du haut vers le bas, la première branche sera la mieux servie, la seconde ensuite si possible, etc.

Modèle Priority Mode Merge

Lorsque les blocs sont en mode Priority, la différence entre Merge et Diverge est que les priorités définies dans le bloc Merge affectent les débits réels des branches entrantes en cas de faible demande aval.

Mode Distributional

Similaire au mode Proportional, le mode Distributional permet de définir des proportions pour chaque branche. Mais à la différence du mode Proportional (mais comme le mode Priority), ces proportions servent de règles de décision pour définir les débits réels sur les branches uniquement en cas d'écart entre la fourniture amont et la demande aval.

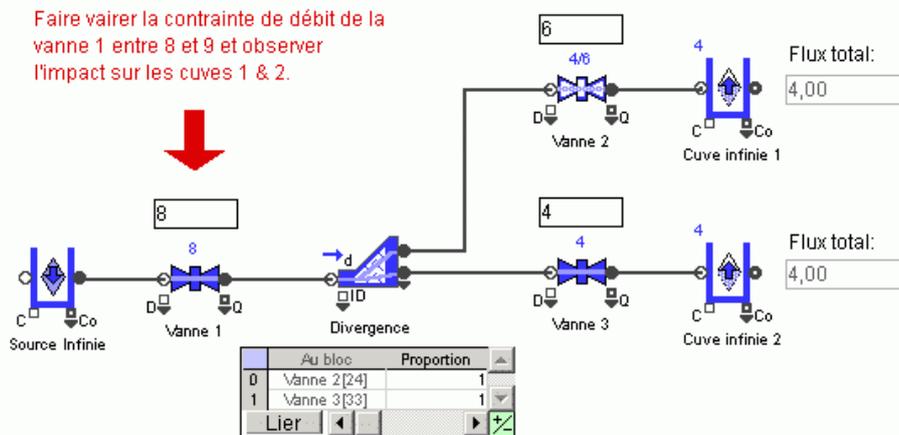
- Dans le cas du bloc Diverge, lorsque la fourniture amont est supérieure ou égale à la demande

aval, le bloc laisse passer autant que la demande aval le permet, et les proportions sont ignorées. Mais quand la demande aval excède la fourniture amont, les proportions servent à déterminer comment distribuer le flux trop rare sur les branches sortantes.

- Le bloc Merge fait passer par chaque branche entrante autant que possible quand la demande aval excède la fourniture amont, et ignore les proportions. Mais lorsque la fourniture amont excède la demande aval, les proportions données dans la table servent à calculer les débits réels appropriés pour les branches entrantes.
- ☞ Les proportions d'un bloc Diverge ou Merge ne sont pas des règles fixes, mais contextuelles, puisqu'elles ne servent que lorsqu'il y a un écart entre demande aval et fourniture amont. Les blocs Merge/Diverge en mode Proportion ne sont pas toujours compatibles avec les blocs Merge/Diverge en mode Adaptatif. Eviter une zone du modèle avec des blocs en mode Sensing et d'autres en mode Priority.

Modèle Distributional Mode Diverge

Dans cet exemple, les proportions des deux branches sont 1:1. La fourniture amont par la Vanne 1 correspond à la demande aval des Vannes 2 et 3. En l'état, les proportions sont ignorées.



Si la fourniture amont devient inférieure à la demande aval:

- Si le débit de la Vanne 1 passe de 10 à 8, le bloc Diverge utilisera les proportions 1:1 pour affecter le flux trop rare aux deux branches en aval. Ici, 4 FPT passeront par les deux branches.
- Si le débit de la Vanne 1 passe à 9 FPT, la situation est différente. Selon les proportions 1:1, le bloc Diverge devrait allouer 4.5 unités à chaque branche. Mais le débit de la Vanne 3 est de 4 FPT, pas davantage. Les 0.5 unités seront envoyées à la branche du haut puisque la demande aval de la branche du bas ne peut accepter la fourniture amont (4.0 vs. 4.5) et que le mode Distributional essaie de faire avancer le flux autant que possible.
- ☞ Le mode Proportional du bloc Diverge sert à résoudre des écarts lorsque la fourniture amont devient inférieure à la demande aval.

Modèle Distributional Mode Merge

En mode Distributional, la différence entre Merge et Diverge est que les proportions du bloc Merge influencent les débits réels des branches entrantes en cas de faible demande aval.

- ☞ Le mode Proportional du bloc Merge sert à résoudre des écarts lorsque la fourniture amont devient supérieure à la demande aval.

Mode Sensing

Similaire au mode Proportional, le mode Sensing utilise des proportions pour calculer les débits réels pour les branches. En revanche les proportions du mode Sensing ne sont pas saisies, mais dérivées dynamiquement de l'état du modèle.

- Dans le cas du bloc Diverge, 'Demand Sensing' implique des proportions en sortie sont calculées en fonction de la demande aval potentielle. Par exemple, la demande aval sur une branche sortante devient la proportion de cette branche.
- De même, le bloc Merge utilise la fourniture amont potentielle pour définir les proportions 'Supply Sensing' pour chaque branche entrante.

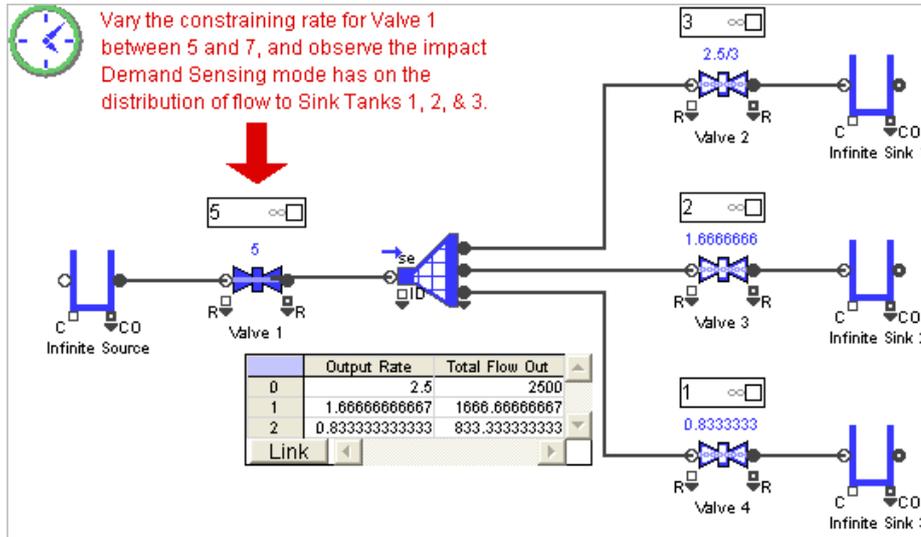
En mode Sensing, vous définissez dans la table le débit maximum possible pour chaque branche. Cette limite haute est utilisée pour limiter le débit de sorte que les proportions puissent être respectées si la fourniture amont ou la demande aval sont infinies.

Le mode Sensing est complexe et doit s'employer avec précautions. On lui préférera généralement le mode Distributional, qui est très similaire.

Modèle Sensing Mode Diverge

Dans cet exemple, les débits des Vannes 2, 3 et 4 définissent la demande en aval du bloc Diverge. Elles définissent ainsi les proportions pour distribuer le flux en sortie du bloc Diverge.

Un débit maximum possible de 1,000 pour chaque branche est saisi dans la table. L'onglet Results (cloné dans le modèle) affiche le débit réel de chaque branche et la quantité totale de flux.



Modèle Sensing Mode Merge

Dans le modèle Sensing Mode Merge, les débits des Vannes 1, 2 et 3 définissent la fourniture amont du bloc Merge. Ils définissent également les proportions utilisées pour distribuer le flux sur les branches en entrée.

Mode Neutral

À la différence des autres modes, le mode Neutral ne permet pas de contrôler les débits réels des branches. C'est un mode passif où aucune branche n'a d'avantage; la branche choisie est imprévisible. Il s'emploie lorsqu'aucun contrôle sur l'orientation du flux n'est requis.

- Dans le cas du bloc Diverge, lorsque la fourniture amont est supérieure ou égale à la demande aval, le bloc passe sur le maximum permis par la demande aval. Mais lorsque la demande aval excède la fourniture amont, la distribution du flux sur les branches ne peut être précisée.
- A l'inverse, le bloc Merge passe sur le maximum possible par chaque branche en entrée lorsque la demande aval excède la fourniture amont. Mais lorsque la fourniture amont excède la demande aval, la distribution du flux sur les branches ne peut être précisée.

Le mode Neutral doit être utilisé avec prudence, mais peut rendre des services. Si aucune branche n'est prioritaire et que vous voulez un flux maximum, vous pouvez envisager ce mode. Le mode Neutral peut également servir à résoudre des règles de décision en conflit. Par exemple, un bloc Merge en mode Neutral en aval d'un bloc Diverge en mode Proportional permet de contrôler les débits réels sur les branches en entrée du Merge.

Les blocs Merge/Diverge en mode Neutral ne sont pas toujours compatibles avec les blocs Merge/Diverge en mode Sensing.

Fonctionnalités des blocs Merge et Diverge

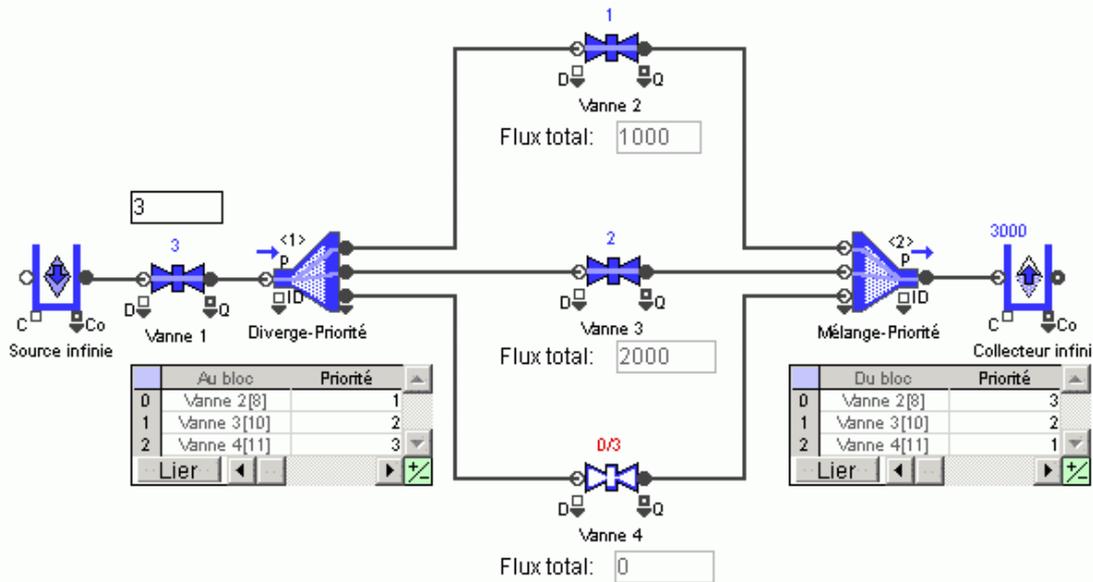
D'autres fonctionnalités des blocs Merge et Diverge présentent un intérêt :

- Paramètres de biais
- Paramètres modifiés dynamiquement
- Envoi / Réception internes.

Paramètres de biais – requêtes en concurrence sur le flux

Les modèles devenant plus complexes, il devient courant que les priorités ou les proportions définies dans un bloc Merge/Diverge entrent en conflit avec les priorités ou les proportions définies dans un autre bloc Merge/Diverge. Ce problème est résolu en affectant un biais aux blocs en compétition. Pour cela, remplissez l'onglet *Model Settings* de chaque bloc, ou l'onglet *Discrete Rate* du bloc Executive. L'exemple suivant montre des cas de concurrence et la manière de les résoudre.

- Comme certains modes permettent une grande souplesse dans la distribution des flux, les blocs Merge ou Diverge en modes Distributional, Priority ou Sensing doivent en plus spécifier un biais.



Modèle *Competing Requests for Flow*

Dans cet exemple:

- Une priorité en ordre descendant a été spécifiée sur les branches sortantes du bloc Diverge, tandis que les priorités sur les branches entrantes du bloc Merge sont en ordre ascendant.
- Le bloc Diverge cherche à satisfaire sa branche sortante du haut d'abord, mais le bloc Merge cherche à satisfaire sa branche entrante du bas. Les priorités du bloc Diverge ont été modifiées en choisissant *Each block defines its own* dans l'onglet *Discrete Rate* du bloc Executive, puis en sélectionnant le bloc Diverge dans la table du bloc Executive et en entrant un rang 1.

Si vous cochez l'option "Show bias order on icon" dans l'onglet *Discrete Rate* du bloc Executive vous verrez la valeur de rang affichée près de l'icône sous forme d'un "<x>". Dans ce modèle, le rang indique <1> pour le bloc Diverge et <2> pour le bloc Merge, indiquant que le bloc Diverge a la préséance sur les requêtes du bloc Merge.

- Le bloc Bias peut s'employer aussi pour résoudre des conflits de priorité.

Envoi/réception intégré

La bibliothèque Rate comporte deux blocs, Throw Flow et Catch Flow, conçus pour faire circuler le flux sans besoin de liens. Mais les blocs Diverge et Merge ont leur propre capacités d'envoi/réception, examinées page suivante.

Modifier dynamiquement les règles de décision

À l'exception des modes Batch/Unbatch et Neutral, les règles pour les modes des blocs Merge/Diverge peuvent changer dynamiquement durant la simulation, et cela de deux façons :

- Les blocs sont contrôlés par d'autres blocs. Pour cela, cochez la case appropriée (par exemple *Priority defined using value connectors*.) dans l'onglet Options. L'icône affiche alors autant de connecteurs continus en entrée qu'il existe de branches.
- Le tableau de paramètres est lié à une base de données ExtendSim ou à une matrice globale. Tout changement dans une table ou une matrice changera les valeurs.

Modify priorities during simulation

Priorities defined using value connectors

Update only when a True value is received at the GO connector

Poll new priorities only each:

Limiter le nombre de recalculs

Cette fonctionnalité permet de modifier dynamiquement le calcul des débits réels, mais invite à la prudence. Les calculs sur chaque branche sont effectués en séquence. Idéalement, les blocs Merge/Diverge ne calculent les nouveaux débits réels pour chaque branche que lorsque tous les paramètres ont été mis à jour. Avec certaines modifications dynamiques, les débits réels sont recalculés chaque fois qu'un paramètre est modifié. La moindre conséquence est serait des temps d'exécution allongés. La pire conséquence serait que des calculs redondants et répétés introduisent une erreur lorsque les débits réels seraient parfois calculés avec des paramètres qui ne sont plus à jour.

Pour éviter ce risque, trois possibilités :

- 1) Les entrées sur un bloc Merge/Diverge sont contrôlées par les sorties d'un bloc incluant une équation, par exemple le bloc Equation (bib. Value). L'équation met à jour toutes ses entrées avant de faire son calcul et d'envoyer des valeurs en sortie vers le bloc Merge/Diverge.
- 2) La case Mise à jour quand reçoit VRAI sur le connecteur GO de l'onglet Options est cochée. Le calcul des nouveaux débits réels est contrôlé explicitement. Dans ce cas, les changements sur les paramètres sont ignorés tant qu'un message n'est pas reçu sur le connecteur GO.
- 3) La case *Interroger les paramètres chaque*: x unités de temps de l'onglet Options du bloc est cochée. Cela fait qu'un ensemble de paramètres nouveaux est mis à jour à intervalles fixes. Dans ce cas, les changements sur les paramètres sont ignorés jusqu'à ce que le nouvel intervalle survienne.

☞ Les options *Mise à jour quand reçoit VRAI sur le connecteur GO* et *Interroger les paramètres chaque*: x unités de temps peuvent être utilisées ensemble.

Envoi et réception de flux à distance

Le flux progresse de manière simple d'un connecteur du flux sortant à un connecteur du flux entrant. Le lien entre connecteur peut devenir pénible dans un gros modèle ou avec des blocs hiérarchiques.

Dans la bibliothèque Rate :

- Le flux peut être envoyé par les blocs Throw Flow et Diverge.
- Le flux peut être reçu par les blocs Catch Flow et Merges.
- Tout bloc d'envoi peut utiliser n'importe quel bloc de réception, quel que soit leur emplacement.

Les mêmes règles s'appliquent qu'avec l'usage des connecteurs :

- Le flux est à sens unique – d'un envoi à une réception
- Un envoi ne peut aller que vers une et une seule réception
- Une réception ne peut puiser que d'un et un seul envoi.

☞ L'avantage à utiliser les blocs Diverge et Merge pour l'envoi/réception est que chaque branche (sortante ou entrante respectivement) peut envoyer/recevoir son propre flux à distance.

Les blocs Throw Flow et Catch Flow, par ailleurs, sont limités à une source ou destination chacun.

Créer un mécanisme envoi/réception

La création d'un lien par envoi/réception se fait soit par l'émetteur (Throw Flow ou Diverge) soit par le récepteur (Catch Flow ou Merge). Vous sélectionnez dans un menu l'autre bloc de la paire, parmi une liste de candidats comportant le label du bloc et son numéro global. Une fois la paire définie, les informations s'affichent dans le dialogue des deux blocs.

Choisir la position du connecteur pour les blocs Merge et Diverge

Si un bloc Diverge ou Merge fait partie d'une paire envoi/réception, vous devez aussi choisir sur quel connecteur se fait l'envoi/réception. Toutes leurs branches ne font pas forcément partie d'un mécanisme envoi/réception.

Select Catch Flow or Merge block

Block:

Position:

Show connection on icon.

Catch B [Catch Flow 14]

Le nombre qui indique le connecteur concerné est affiché dans la colonne de gauche du tableau dans l'onglet Throw/Catch du bloc Diverge/Merge; le numéro de la branche du haut étant zéro (0). Vous sélectionnez le connecteur dans un menu à la droite du champ Position dans le bloc correspondant. Le menu liste tous les connecteurs disponibles pour le bloc concerné.

- ☞ Un astérisque à la droite d'une position de connecteur dans le menu indique que le connecteur est déjà utilisé par un autre bloc envoi/réception.

Options de filtre pour faciliter les liens envoi/réception

Dans un grand modèle, les candidats pour un envoi/réception peuvent être très nombreux. Pour simplifier le repérage, trois types de filtres peuvent s'employer :

- Filtre par groupe
- Filtre par type de bloc
- Filtre par blocs encore libres

Ces trois filtres peuvent se combiner.

Filtre par groupe

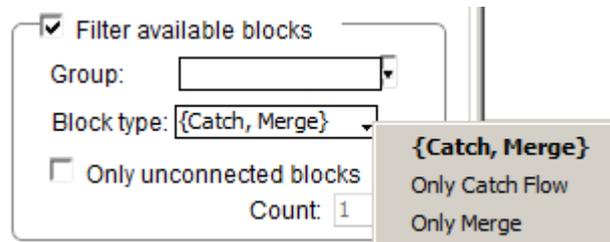
Tout bloc ayant des capacités d'envoi/réception peut faire partie d'un groupe. Les groupes sont créés et sélectionnés par le menu groupe figurant dans le dialogue des blocs. Lorsque le menu est vide, le bloc n'appartient à aucun groupe. Dès qu'un bloc fait partie d'un groupe, ses options d'envoi/réception sont limitées aux blocs de ce groupe.



Filtre par type de bloc

Par défaut, un bloc d'envoi ne peut s'apparier qu'aux blocs Catch Flow et Merge, et un bloc de réception qu'aux blocs Throw Flow et Diverge.

- un bloc d'envoi peut ne s'apparier qu'à des blocs Catch Flow ou qu'à des blocs Merge.
- de même un bloc de réception peut ne s'apparier qu'à des blocs Throw Flow ou qu'à des blocs Diverge.



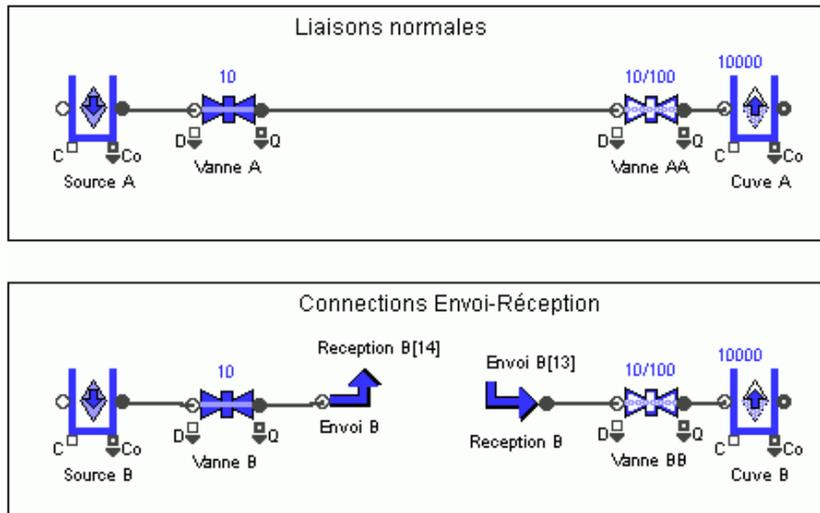
Only unconnected blocks (Uniquement les blocs libres)

Ce filtre réduit les blocs candidats à ceux qui n'ont pas encore de connexion envoi/réception établi.

Modèle Catch Flow and Throw Flow

Le modèle *Catch Flow and Throw Flow* présente deux lignes. Elles produisent les mêmes résultats, mais avec des connexions différentes:

- La ligne du haut relie de manière classique une Vanne A à une Vanne AA.
- La ligne du bas utilise une connexion envoi/réception pour lier la Vanne B à la Vanne BB.

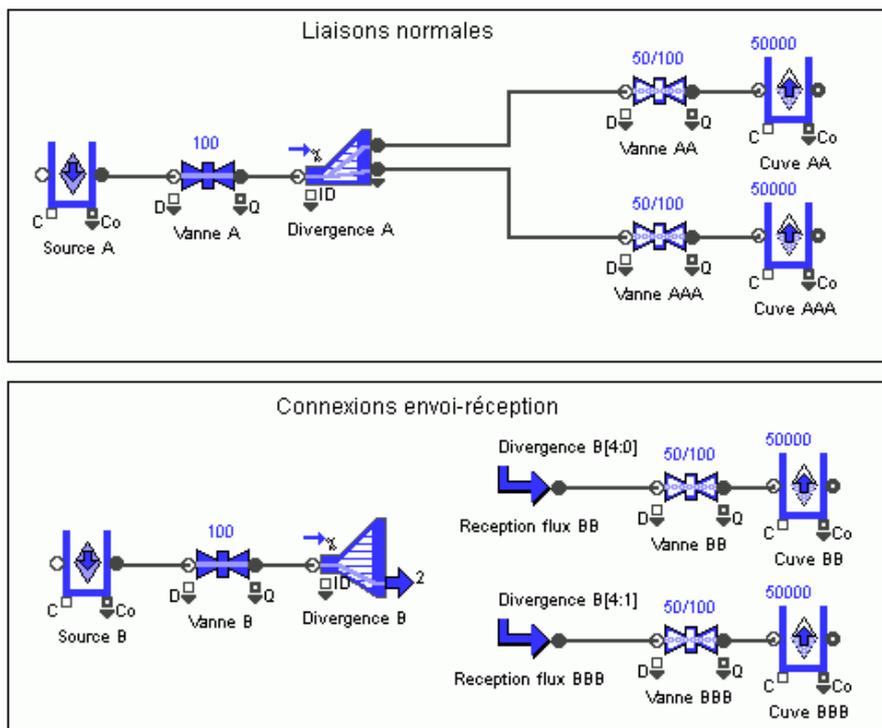


Dans le dialogue du bloc Throw Flow vous pouvez sélectionner le bloc Catch Flow et voir la connexion envoi/réception ; de même dans le dialogue du bloc Catch Flow.

Modèle Catch Flow and Diverge

Comme le modèle précédent, les deux lignes produisent les mêmes résultats, mais avec des connexions différentes :

- La ligne du haut relie de manière classique les branches sortantes d'un bloc Diverge aux blocs Vanne AA et Vanne AAA.
- La ligne du bas utilise des connexions envoi/réception.



Dans la ligne du bas, les deux blocs Catch Flow indiquent dans leur dialogue que le bloc de label Diverge B leur envoie le flux. Comme le flux est émis par un bloc Diverge pouvant avoir plusieurs branches et comme chaque Catch Flow doit recevoir son flux d'une source séparée, les blocs Catch Flow doivent aussi spécifier quel connecteur du bloc Diverge est leur pourvoyeur.

L'onglet Envoi du bloc Catch Flow BB indique que le flux provient du bloc Diverge de label Diverge B, par sa branche du haut (connecteur du flux sortant position 0).

☞ Si le lien envoi/réception est correctement défini pour une branche d'un bloc Merge ou Diverge, un bouton "Open" apparaît pour cette branche dans la dernière colonne de la table.

Retarder le flux

Maintenir le flux à une certaine vitesse ou le stopper

Ce chapitre décrit comment retarder le flux – pendant une période de temps ou jusqu'à ce que certaines conditions soient remplies– soit en bloquant le mouvement du flux, soit en réduisant sa vitesse.

- Contrôler comment et quand une Valve observe ou ignore ses paramètres de débit maximum:
- Etablir un objectif pour une quantité de flux
- Etablir un objectif pour une durée
- Utiliser une hystérésis pour dire combien de temps le débit maximum sera respecté
- Utiliser un bloc Shift (bib. Item) avec un bloc Convey Flow, Interchange, Tank ou Valve pour bloquer le mouvement du flux durant une période
- Transporter le flux à un certain rythme par le bloc Convey Flow

☞ La plupart des modèles de ce chapitre figurent dans le répertoire \Exemples\Item Rate\Delaying Flow. Certaines captures font référence à des modèles francisés : n'existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension.

Contrôler le débit maximum d'une Vanne

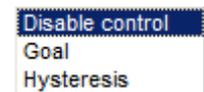
Nous avons appris comment définir le débit maximum d'une Valve. L'onglet Flow Control propose des méthodes avancées sur quand et comment la Valve applique son débit maximum.

Les options de contrôle sont notamment :

- Définir un objectif d'une certaine quantité de flux à faire passer par Valve.
- Définir un objectif de durée d'ouverture de la Valve, ou de durée de fermeture.
- Déterminer ce qui se passe lorsque l'objectif est terminé.
- Utiliser une hystérésis pour retarder la réponse de la Valve aux besoins du système.

Utiliser l'onglet Flow Control

Par défaut, l'onglet Flow Control de la Valve est désactivé. Choisissez Goal (Objectif) ou Hysteresis dans le menu. Si l'option Goal est sélectionnée, un second menu permet de préciser l'objectif en indiquant s'il est de quantité ('as a quantity') ou de durée ('as a duration').



Respecter le débit maximum pour un objectif

Que vous sélectionniez un objectif de quantité ou de durée, le débit maximum de la Valve est respecté tant qu'un objectif est *On*. Si l'objectif est *off*, vous pouvez choisir que le débit maximum soit respecté ou non, ou même le flux stoppé. Votre but en utilisant un objectif peut ainsi être de bloquer le flux pendant une certaine durée, et de le laisser avancer à nouveau ensuite. En fait, vous pouvez respecter le débit maximum même si l'objectif est *off*. Cela peut servir dans le cas d'un objectif de quantité, où vous souhaitez juste que la Valve indique lorsqu'un objectif est terminé.

- Si son débit maximum est de 0 (zéro), la Valve bloquera le flux tant que l'objectif est *On*.
- Si son débit maximum est de >0, blank ou infini, la Valve est passante à ce débit tant que l'objectif est *On*. (Si le débit maximum est blank, la Valve utilise un débit infini.)

☞ Si le débit maximum Valve est de zéro et qu'un objectif de quantité est *On*, aucun flux ne passe par le bloc et l'objectif ne se termine jamais.

Options quand l'objectif est Off

Des options permettent de choisir ce qui se passe lorsque l'objectif passe à *Off*:

- Stop flow (Stopper le flux)
- Ignore maximum rate (Ignorer le débit maximum)

- Observe maximum rate (Respecter le débit maximum)

Définir un objectif de quantité

Par le menu de l'onglet Flow Control, vous spécifiez que le bloc a un objectif de faire passer une certaine quantité. D'autres options permettent de spécifier ce que doit faire la vanne quand la quantité de flux est passée.

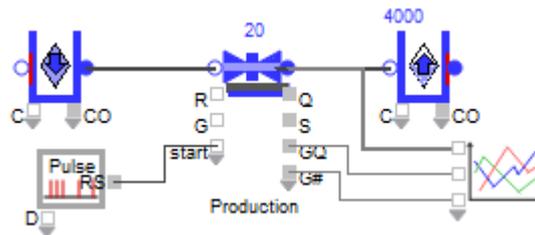
L'objectif de quantité module la contrainte critique de la Valve (son débit maximum) en alternant entre les états On et Off.

- Tant que l'objectif de quantité est On, la valeur du champ Maximum rate: de l'onglet Valve, ou celle sur son connecteur R (débit maximum), est respectée. L'objectif reste On jusqu'à ce que, soit la quantité de flux passé atteigne l'objectif, soit l'objectif est interrompu. L'objectif passe alors à Off.
- Ce qui se passe lorsque l'objectif passe à Off state dépend des options Off choisies dans le dialogue : stopper le flux, ignorer le débit maximum, ou respecter le débit maximum.

☞ Pour interrompre un objectif, envoyer une valeur sur le connecteur d'entrée stop de la Valve.

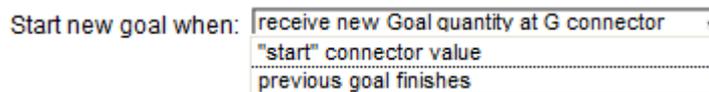
Modèle Quantity Goal

Ce modèle utilise un bloc Pulse (bib. Value) pour lancer périodiquement un nouvel objectif de quantité. De début de simulation puis toutes les 60 minutes, le bloc Pulse envoie une valeur VRAI (un nombre ≥ 0.5) au connecteur start de la Valve.



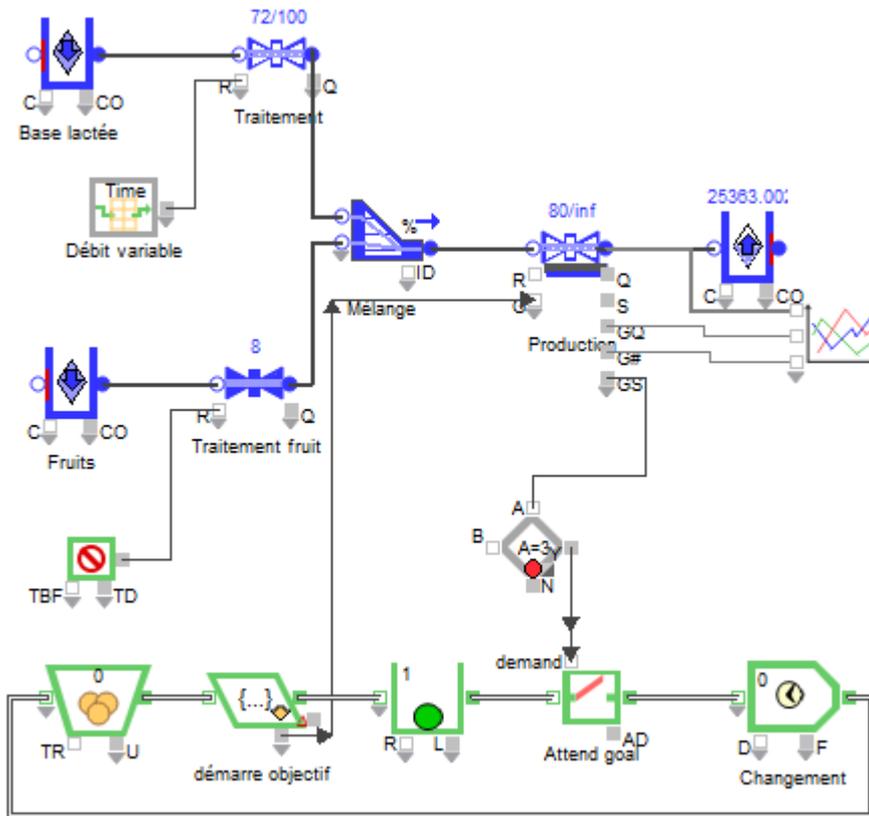
Le débit maximum de la Valve est de 20 litres/minute et la simulation dure 480 minutes. L'onglet Flow Control a défini qu'une quantité de 500 litres est l'objectif. Tant que l'objectif est On, le flux passe par la Valve au débit maximum de 20 litres/minute. Après 500 litres, l'objectif passe à Off, le flux est stoppé et le débit réel passe à 0. Un graphique affiche le débit réel de la vanne, la quantité de l'objectif avec l'heure où il a été atteint, et le nombre de nouveaux objectifs.

Les paramètres de l'onglet Flow Control font que la Valve commence un nouvel objectif si le connecteur "start" reçoit une valeur = 1.



Modèle Changeover Quantity Goal

Le modèle *Changeover Quantity Goal* est un modèle plus complexe que le précédent. Il s'appuie sur le modèle Yogurt Production et présente un objectif de quantité sur une Valve dans le but de contrôler des changements d'outils dans une séquence de production. Les blocs de la partie haute proviennent de la bib. Rate, ceux de la partie basse de la bib. Item.



Options pour démarrer un nouvel objectif

Un bloc Decision (bib. Value) transmet des valeurs d'une Valve à un bloc Gate (bib. Item).

Comme l'objectif initial est "none", l'objectif est à Off et rien ne passe par la vanne Production de début de simulation. Mais lorsqu'une entité passe par le bloc Get (bib. Item) de label Start Goal, un message est envoyé au connecteur G de la vanne Production. L'objectif passe alors de Off à On et la quantité est prise sur la valeur de l'attribut Quantity de l'entité (5000 litres).

Dans la portion du bas, l'entité quittant le bloc Get va dans une Queue, où elle reste bloquée avant de passer par le bloc Gate. La Gate reste fermée jusqu'à ce que la vanne Production atteigne le nouvel objectif de 5000 litres. Alors l'objectif passe de On à Off, le flux s'arrête (c'est l'option choisie dans l'onglet Flow Control) et la valeur sur GS (statut objectif ou 'goal status') prend la valeur 3 (indiquant que l'objectif est terminé).

Ensuite, la Gate s'ouvre et l'entité va dans l'activité Changement où elle est retenue pendant la durée d'un changement d'outils. L'entité retourne ensuite commencer le prochain cycle de production.

☞ Par défaut, le connecteur de sortie GS (statut objectif) d'une Valve indique les valeurs suivantes :

- 0 quand il n'y a pas d'objectif
- 1 quand un objectif commence
- 2 quand un objectif est en cours
- 3 quand un objectif est terminé
- 4 quand un objectif est interrompu

Observez comment est rempli l'onglet Flow Control :

Goal

Goal quantity at G: gallons

Goal impact on flow when On and Off:

On:

Off:

Start run with goal:

Start new goal when:

Interrupt goal when "stop" connector value

If a new goal arrives before the previous one is finished:

- La quantité pour l'objectif est reçue sur le connecteur G (goal). Dans ce modèle, elle est passée par une valeur d'attribut.
- Il n'y avait pas d'objectif initial. C'est pourquoi l'objectif était Off state en début de simulation.
- La Valve devait Stopper le flux lorsque l'objectif était Off.
- Un nouvel objectif devait démarrer à chaque message reçu sur le connecteur G. Une valeur sur G a deux conséquences: 1) l'objectif passe de Off à On ; 2) l'objectif a une nouvelle quantité définie par la valeur reçue sur G.
- Dans ce modèle, à cause de la régulation par entité, de type Grafcet, il ne peut pas arriver qu'un nouvel objectif arrive alors que l'ancien n'est pas terminé.

Notez que la vanne Production a un débit maximum infini. Elle n'est donc jamais limitante en tant que telle. C'est que son débit réel est en fait déterminé par les deux vannes en amont. Une de ces Valves est liée à une Lookup Table (bib. Value) qui modifie le débit maximum en fonction de l'heure du jour.

L'autre Valve est liée à un bloc Shutdown (bib. Item) qui arrête périodiquement le flux pour une certaine durée.

☞ Cet exemple utilise le connecteur G pour indiquer à la fois "quand" et "combien" pour l'objectif.

Définir un objectif de durée

Comme l'objectif de quantité, l'objectif de durée sert à moduler le débit maximum de la Vanne. L'objectif de durée passé des états On à Off en fonction du temps, et non par des critères de volume.

Un objectif de durée reste à On pendant un certain temps avant de passer à Off :

- Tant qu'un objectif de durée est On, la valeur du champ Maximum rate: de l'onglet Valve, ou celle sur son connecteur R (débit maximum), est respectée. L'objectif reste On jusqu'à ce que soit la durée de l'objectif soit passée, soit que l'objectif est interrompu. L'objectif passe alors à Off.
- Ce qui se passe lorsque l'objectif passe à Off state dépend des options Off choisies dans le dialogue : stopper le flux, ignorer le débit maximum, ou respecter le débit maximum.

☞ Pour interrompre un objectif, envoyer une valeur sur le connecteur d'entrée stop de la Valve.

Modèle Duration Goal

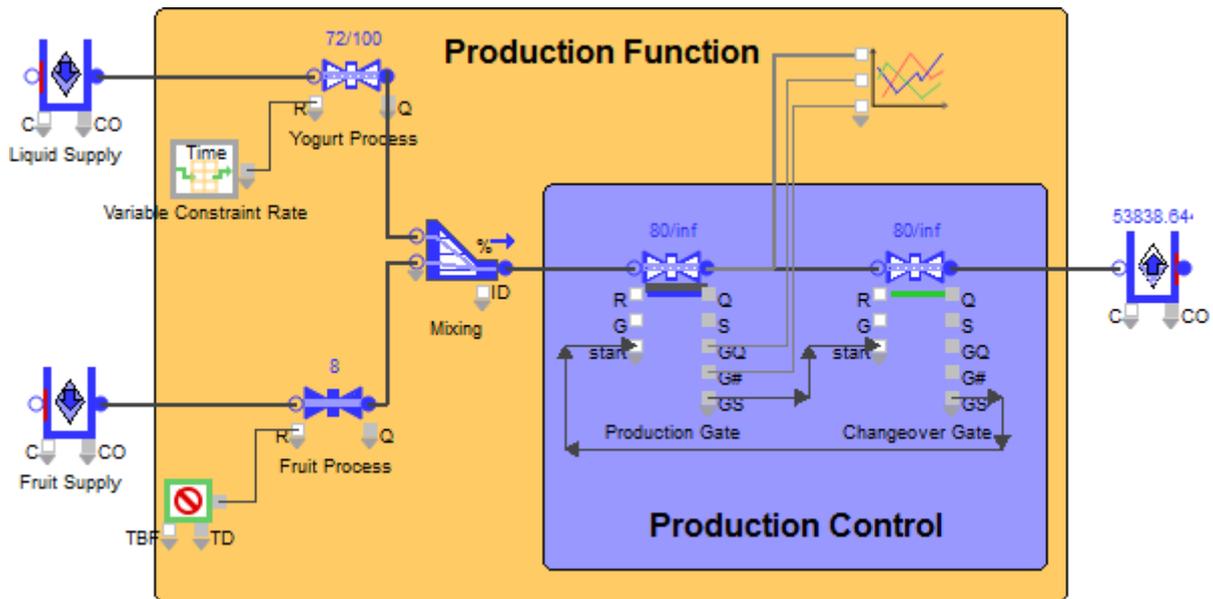
Le modèle *Duration Goal* est similaire au modèle *Quantity Goal*, sauf que l'objectif agit sur la durée.

Le débit maximum de la Valve est de 20 litres/minute et la simulation dure 45 minutes. Quand l'objectif est On, le flux passe par la Valve au débit maximum de 20 litres/minute. Après 45 minutes, l'objectif passe à Off et le flux est stoppé. Le bloc Pulse (bib. Value) relance l'objectif toutes les 60 minutes. Durant la simulation, le débit maximum du bloc passe de 20 litres/minute à 0, selon que l'objectif est On ou Off. Le débit réel reste à 0 pendant la période (15 minutes) entre la fin de l'objectif précédent et l'arrivée d'un nouvel objectif.

Modèle Changeover With Only Goals

Le modèle *Changeover With Only Goals* est équivalent du modèle *Changeover Quantity Goal* en termes de comportement. La vanne Production conserve un objectif de quantité, mais le changement d'outils est contrôlé par une seconde Valve avec un objectif de durée, plutôt que par une entité. Comme l'onglet

Flow Control spécifie un objectif de quantité initial de 5000 litres, la vanne Production commence la simulation en respectant son débit maximum– infini. Son débit réel est lié à celui des deux vannes en amont.



L'état de l'objectif de la vanne Production communique avec la vanne Changeover via le connecteur de sortie *GS* et de connecteur d'entrée *start*. Par ces liens et par le paramétrage de l'objectif de durée dans la vanne Changeover, l'objectif de durée démarre en état Off et passe à On seulement lorsque l'objectif de quantité de la vanne Production. Après 20 minutes, l'objectif de durée est terminé et la vanne Changeover envoie un message à la vanne Production pour démarrer un nouvel objectif de quantité.

Goal as a duration

Goal duration is minute*

Goal impact on flow when On and Off:

On: Observe maximum rate

Off:

Start run with goal:

Start new goal when: =

Interrupt goal when "stop" connector value

If a new goal arrives before the previous one is finished:

Ignore new goal when BLANK or negative value received.

Observez comment est rempli l'onglet Flow Control de la vanne Changeover :

- L'objectif de durée est une constante de 20 minutes. La durée aurait pu être variable, en choisissant de relier de connecteur O.
- Le blocage du flux durant 20 minutes permet le changement d'outils. Le débit maximum de la vanne Changeover est de 0. Cela causerait un blocage du flux en l'absence de tout objectif pour ce bloc. Mais si le bloc a un objectif de durée et qu'il est On, de débit maximum de 0 sera respecté et le flux sera bloqué. Lorsque l'objectif passe à Off après 20 minutes, la vanne Changeover n'applique aucune contrainte sur le débit parce qu'elle est paramétrée à Off: ignorer le débit maximum.
- Un nouvel objectif de durée ne démarre que lorsque le connecteur *start* reçoit une valeur de 3. Par conséquent, le nouvel objectif de durée ne commence que quand l'objectif de quantité de la vanne Production amont est terminé.

- Lorsque le changement d'outils est terminé et que l'objectif de durée passe à Off, un signal est envoyé de la vanne Changeover à la vanne Production. Il en résulte un nouvel objectif de quantité pour la vanne Production.

Définir une hystérésis

L'hystérésis est une propriété de systèmes permettant une réaction différée à un changement. Lorsqu'on ajoute une hystérésis à une portion de modèle, c'est pour introduire un délai dans le passage d'un état à un autre.

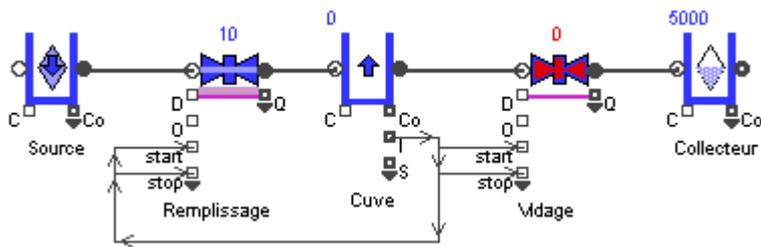
L'hystérésis permet d'introduire ce décalage dans la réponse d'une Vanne aux demandes du système. Elle permet d'éviter des oscillations et de mieux réguler le flux.

À la différence des objectifs examinés plus haut, où les conditions d'application du débit maximum de la Valve étaient saisies dans le dialogue, l'hystérésis est toujours pilotée par des informations externes au bloc. Elle s'appuie toujours sur le connecteur d'entrée *start* de la Valve pour savoir quand respecter le débit maximum de la Valve, et sur le connecteur d'entrée *stop* pour contrôler quand le débit maximum sera ignoré. Quand le débit maximum est ignoré, un menu du dialogue de la Valve permet de choisir si le flux sera stoppé, ou si la Valve sera passante sans contrainte.

Modèle Hysteresis

Dans ce modèle, la vanne Remplissage s'ouvre lorsque la cuve du milieu est vide et se ferme lorsqu'elle est pleine. Inversement, la vanne Vidage s'ouvre lorsque la cuve est pleine et se ferme lorsqu'elle est vide. Le modèle passe en boucle par les états suivants :

- La vanne Vidage se ferme et la vanne Remplissage s'ouvre
- La Cuve commence à se remplir
- La Cuve atteint l'état plein
- La vanne Vidage s'ouvre et la vanne Remplissage se ferme
- La Cuve commence à se vider
- La Cuve atteint l'état vide



En fonction de paramètres dans l'onglet Indicators, le connecteur de sortie I (indicateur) du Tank émet une valeur 0 pour vide et 2 pour plein.

Hysteresis

Observe Maximum rate when "start" connector = 0

Ignore Maximum rate when "stop" connector = 2

When ignoring: Stop flow

Start run with hysteresis: ignored

Lorsque le connecteur *start* de la vanne Remplissage reçoit un 0 par le connecteur I de la cuve, la vanne respecte son débit maximum. Lorsque le connecteur *stop* de la vanne Remplissage reçoit un 2, le bloc s'arrête.

La vanne Vidage est paramétrée à l'inverse. Lorsque le connecteur *start* reçoit un 2 de la cuve, elle respecte son débit maximum. Lorsque le connecteur *stop* reçoit un 0, le bloc s'arrête.

Retarder le flux par le bloc Shift

Le bloc Shift (bib. Item) est examiné dans la section sur les ressources. Il permet d'appliquer des horaires (de travail ou de capacité) à certains blocs de la bibliothèque Rate.

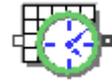
La bibliothèque Rate ne fonctionne qu'avec les horaires de type On/Off.

- Convey Flow. Lorsque l'horaire est Off, le débit entrant réel est de 0, bloquant l'entrée du flux. En fonction de l'option choisie dans le dialogue, la vitesse passera aussi à 0 ou restera inchangée. Si la case *Empty when shift is off* (Vider si horaire off) est cochée, le flux présent sur le convoyeur au moment du passage à Off continuera sa progression vers la sortie du bloc.
- Tank. Les débits entrant et sortant réels passent à 0 lorsque l'horaire est Off, ce qui bloque le flux.
- Interchange. Même logique que le Tank.

- Valve. Même logique que le Tank.

Ajouter un horaire à un modèle

Le plus simple est de cliquer sur le bouton *Add Shift* (Ajouter un horaire) de l'onglet Options des blocs Convey Flow, Tank, Interchange ou Valve, et automatiquement :



- Un bloc Shift est placé dans le modèle sous le bloc d'origine,
- Le nom de l'horaire est entré dans le champ *Use Shift* : du bloc d'origine,
- Le dialogue du bloc Shift est ouvert pour permettre le paramétrage.

Pour utiliser l'horaire, entrez des informations dans son dialogue (le bloc agit à distance et n'a pas besoin d'être relié). Chaque Horaire a un nom par défaut. Si vous modifiez ce nom, les blocs utilisant l'horaire répercuteront la modification.

Bloc Convey Flow

Le flux qui entre dans le bloc Convey Flow ne peut le quitter qu'après le délai spécifié et qui est fonction de la longueur et de la vitesse. Le Convey Flow est donc un bloc stockeur avec des éléments de flux distribués sur sa longueur à des densités variables. Par densité l'on entend la quantité de flux accumulée à un point du convoyeur.

Le bloc peut représenter un convoyeur, mais aussi un four industriel, un système de réfrigération ou tout autre équipement où une composante de longueur et de position intervient.

Le Convey Flow réalise d'intensifs calculs, et il est donc préférable de l'utiliser dans des systèmes où vous devez vraiment suivre le mouvement et la position du flux.

Déterminer la vitesse et la distance

Le bloc Convey Flow propose deux options principales : Temps de trajet basé sur la vitesse ou sur le délai ("Travel time based on...").

- **Vitesse (speed).** Spécifie la vitesse maximum potentielle à laquelle le convoyeur peut transporter le flux. Si la fourniture amont dépasse la demande aval, la vitesse réelle du flux peut être inférieure au paramètre de vitesse.
- **Délai (delay).** Représente le temps de séjour du flux dans le bloc sans blocage aval. Si le Temps de trajet est basé sur la vitesse, le séjour est calculé en divisant la Longueur par la Vitesse. Si le Temps de trajet est basé sur le délai, le flux utilisera le Délai indiqué pour parcourir la Longueur indiquée.
- **Longueur (length).** Représente la distance que doit parcourir le flux avant d'atteindre le point de sortie du bloc. La longueur d'un Convoyeur flux doit être supérieure à 0.
- **Densité maximum (maximum density).** La densité est la quantité de flux qui s'accumule en chaque point du convoyeur. Cette densité est fonction du débit de la fourniture amont, de la vitesse du convoyeur, du débit de la demande aval et des paramètres du dialogue. Le champ Maximum density limite l'accumulation possible en un point du convoyeur. Par exemple, si la fourniture amont est supérieure ou égale au débit entrant maximum (vitesse*densité maximum), la quantité de flux qui entre dans le convoyeur sera égale à la densité maximum. Dans ce cas, c'est la capacité du convoyeur à accepter le flux qui limite son débit entrant réel.

Select behavior

Mode:

Travel time based on:

Speed: length unit / minute*

Delay: minute*

Length: length unit

Maximum density: containers / length unit

Maximized capacity: containers

- ☞ Les paramètres de vitesse et de séjour peuvent changer dynamiquement durant la simulation; mais la longueur et la densité maximum restent fixes.

Comportement du Convey Flow

Le bloc Convey Flow est divisé en segments, les limites de chaque segment étant définies par un changement de la densité du flux. En fonction des options choisies dans le dialogue, le flux peut s'accumuler ou s'entasser sur du convoyeur dès que la quantité de flux prête à sortir dépasse la demande aval.

Le bloc Convey Flow propose trois options de mode, contrôlant l'accumulation du flux :

- **Accumulate-Maximum density** (Accumule-densité maximum). Permet au flux de s'accumuler jusqu'à la densité maximum. Si le flux prêt à sortir excède la demande aval, le flux en attente s'entasse en sortie du convoyeur, jusqu'à atteindre le niveau de densité maximum.
- **Accumulate-Fill empty segments** (Accumule-remplit les vides). Permet au flux de s'accumuler lorsque le flux prêt à sortir excède la demande aval. Un segment vide est une portion du convoyeur ayant une densité de 0. Cela diffère de la première option en ce qu'une section du flux ne peut s'entasser sur une autre.
- **Non-accumulating** (Non-accumulateur). Cette option n'autorise aucune accumulation. La vitesse du bloc se réduit lorsque sa fourniture dépasse la demande aval.

- ☞ Le modèle *Compare Convey Flow* compare trois blocs Convey Flow, chacun illustrant l'un de ces comportements.

Débits de contrainte

Nous avons vu que le Convey Flow calcule séparément les contraintes critiques ses connecteurs de flux entrant et sortant. Les contraintes critiques sont dérivées des conditions du modèle et de paramètres du dialogue.

- La contrainte critique pour le flux entrant du Convey Flow est calculée en multipliant la vitesse réelle par la densité maximum.
- ☞ La vitesse réelle peut être inférieure ou égale à la vitesse entrée dans le dialogue. Si le bloc est non accumulateur ou s'il est accumulateur mais ne peut accumuler davantage alors qu'il a du flux prêt à sortir qui excède la demande aval, elle sera inférieure.
- La contrainte critique pour le flux sortant est le résultat de la multiplication de la vitesse saisie par la densité de flux présente en sortie du bloc.

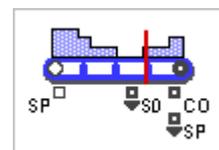
Informations dans le Convey Flow

Le bloc Convoyeur fournit plusieurs types d'informations sur la distribution du flux sur sa longueur. Certaines sont proposées par défaut, d'autres doivent être personnalisées dans le dialogue.

Distribution du flux

Si vous sélectionnez Simulation > Animation 2D avant de lancer la simulation, la distribution du flux sur le bloc Convey Flow s'affichera au-dessus de son icône.

Dans l'onglet Animation du bloc, le choix *Voir la distribution du flux dans la table durant la simulation* fait s'afficher une table. Elle se remplira des informations sur la distribution du flux dans les segments :



<input checked="" type="checkbox"/> Voir la distribution du flux dans la table durant la simulation:				
	Limite haute	Limite basse	Densité	Quantité
0	10 metres	9 metres	4 tonnes/metres	4 tonnes
1	9 metres	7 metres	6 tonnes/metres	12 tonnes
2	7 metres	6 metres	4 tonnes/metres	4 tonnes
3	6 metres	5,16666666667 m	4,8 tonnes/metre	4 tonnes
4	5,16666666667 m	4 metres	0 tonnes/metres	0 tonnes
5	4 metres	2 metres	2 tonnes/metres	4 tonnes
6	2 metres	1 metres	4 tonnes/metres	4 tonnes
7	1 metres	0 metres	6 tonnes/metres	6 tonnes

- ☞ La longueur d'un bloc Convey Flow se divise en segments, les limites de chaque segment étant définies par un changement de la densité du flux. La table ci-dessus indique un Convey Flow comportant 8 segments.

Point d'accumulation

Si le bloc permet l'accumulation, toute accumulation commence au point de sortie du flux, et se propage vers l'entrée. Le point au-delà duquel aucune accumulation n'est possible est appelé point d'accumulation. Ce point se déplace en général entre l'entrée et la sortie en cours de simulation.

L'onglet Results donne des informations sur le flux accumulé : la distance entre le point d'accumulation et la sortie du flux, l'indicateur et la quantité de flux accumulée.

Si vous sélectionnez Simulation > Animation 2D et qu'une accumulation se produit, une barre verticale rouge se déplace au-dessus de l'icône durant la simulation ; elle indique l'emplacement du point d'accumulation.

Capteurs

Le bloc Convey Flow intègre un onglet **Sensors** (Capteurs) pour indiquer l'emplacement et le point de déclenchement de capteurs le long du convoyeur. Chaque capteur lit et communique la densité du flux dans le temps à un endroit particulier du convoyeur. Cette information est affichée dans la table de l'onglet correspondant et émise sur le connecteur de sortie S (capteur) – un connecteur S par capteur.

Emplacements = Capteur émet:

	Emplacement	Déclenchement	Résultat	<input type="button" value="Voir exemple"/>
0	10 metres	0 tonnes/metres	2 tonnes/metres	
1	7,5 metres	0 tonnes/metres	6 tonnes/metres	
2	2,5 metres	0 tonnes/metres	2 tonnes/metres	
3	0 metres	0 tonnes/metres	6 tonnes/metres	

Vous devez spécifier dans la table non seulement le nombre de capteurs désiré, mais aussi l'emplacement de chacun (Utilisez le bouton +/- dans le coin inférieur droit pour changer le nombre de capteurs).

- ☞ Placez des capteurs avec mesure, car des calculs supplémentaires sont nécessaires pour leur mise à jour et celle des connecteurs correspondants.

Indicateurs

Nous avons défini le point d'accumulation comme le point au-delà duquel aucune accumulation n'est possible. Vous pouvez placer un indicateur pour signaler que le point d'accumulation est dans tel segment particulier du Convey Flow.

L'onglet Indicateurs s'emploie pour définir des segments indiquant où se situe le point d'accumulation sur la longueur. Chaque segment reçoit un nom, un plage et un numéro d'ID. L'ID met à jour le connecteur de sortie I (indicateur) lorsque le point d'accumulation se déplace d'un segment au suivant.

- ☞ Placez des indicateurs avec mesure, car des calculs supplémentaires sont nécessaires pour leur mise à jour et celle des connecteurs correspondants.

Quand ne pas utiliser le Convey Flow ?

Le Convey Flow ne doit être utilisé que dans des systèmes où vous devez vraiment suivre le mouvement et la position du flux. Le bloc est très précis, mais génère beaucoup de calculs, et donc ralentit la simulation. Quelques conseils :

- Le temps de trajet doit être assez long pour justifier l'emploi du bloc. Si son impact sur le résultat de la simulation est faible, il est préférable de l'ignorer. Par exemple, le temps de trajet pour aller d'une cuve à un autre est souvent insignifiant, et devrait être alors ignoré.
- La distribution du flux le long du Convey Flow doit avoir un impact significatif dans le modèle. SI le débit entrant réel varie de manière significative durant la simulation et/ou si la vitesse change, le produit peut être distribué de manière inégale le long du convoyeur. Cette discontinuité influence le débit du flux. Plus grande est la variation du flux, plus grand souvent est l'impact, et l'emploi du bloc peut se justifier.

Une alternative est d'utiliser un Tank et une Valve pour simuler un Convoyeur. Cela demande beaucoup moins de calculs, et pourra souvent rendre les mêmes résultats qu'un bloc Convey Flow.

On le voit dans le modèle *Tank And Valve To Convey*. Dans cet exemple, la ligne du bas est presque identique à celle du haut, sans utiliser le bloc Convey Flow. Comme les Valves A1 et B1 varient dans leur débit, le bloc Convey Flow (Convoyeur A) reçoit et crée un grand nombre de messages et ralentit la simulation. Mais la cuve Convoyeur B1 ne crée aucun événement supplémentaire, et sera très efficace, même si un peu moins précis.

Flux et entités discrètes

Les modèles “hybrides” impliquent de bien connaître les blocs des deux bibliothèques. Il y a deux manières d’intégrer des blocs discrets de la bib. Item à des blocs la bibliothèque Rate:

- 1) L’envoi de signaux et de valeurs par des connecteurs continus des blocs, pour signaler des changements d’état. Par exemple:
 - Une entité est générée lorsqu’un certain niveau est atteint dans un Tank.
 - La valeur d’un attribut d’une entité qui passe détermine un changement de débit d’une Valve.
- 2) Le mélange des flux via le bloc Interchange, comme abordé dans le Tutorial.

Ce chapitre examine les techniques utilisées dans des modèles “hybrides”, notamment :

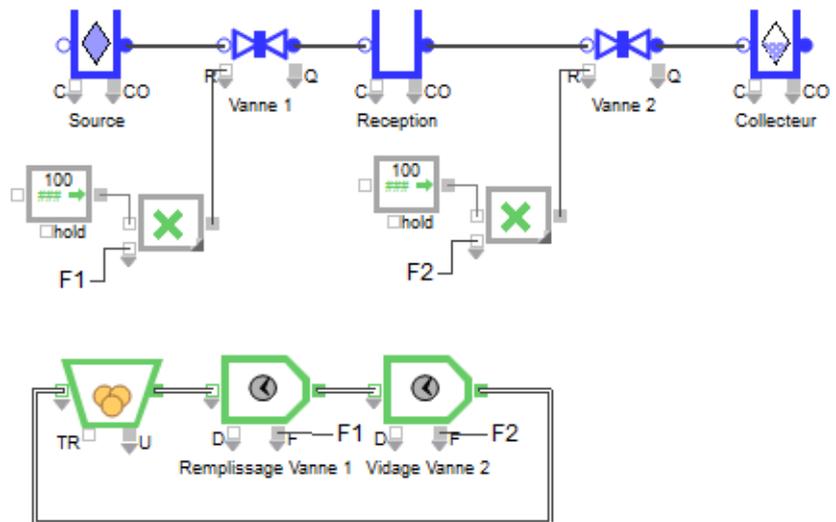
- Contrôler le flux par des blocs de la bib. Item
 - Contrôler des entités par des blocs de la bibliothèque Rate
 - Mélanger les deux types de flux par le bloc Interchange
- ☞ Les modèles de ce chapitre figurent dans le répertoire \Exemples\Item Rate\Flow and Items. Certaines captures font référence à des modèles francisés : n’existe désormais que la version anglaise des modèles, mais la capture commentée en français facilite la compréhension

Des entités qui contrôlent le flux

Le modèle *Item Controls Flow* montre comment une entité sert à ouvrir une vanne, permettant ainsi à une cuve de se remplir, puis fait appel à la même entité pour déclencher le vidage de la cuve.

Les Vannes 1 et 2 sont fermées tant que la présence d’une entité dans un bloc Activity ne les ouvre pas. La Vanne 1 ouverte remplit la cuve Réception. Lorsque l’entité change de bloc Activity, cela ferme la Vanne 1 et ouvre la Vanne 2. La cuve Réception se vide alors dans la cuve Collecteur.

L’ouverture et la fermeture des vannes se fait donc en fonction du bloc Activity contenant l’entité, et pour combien de temps.

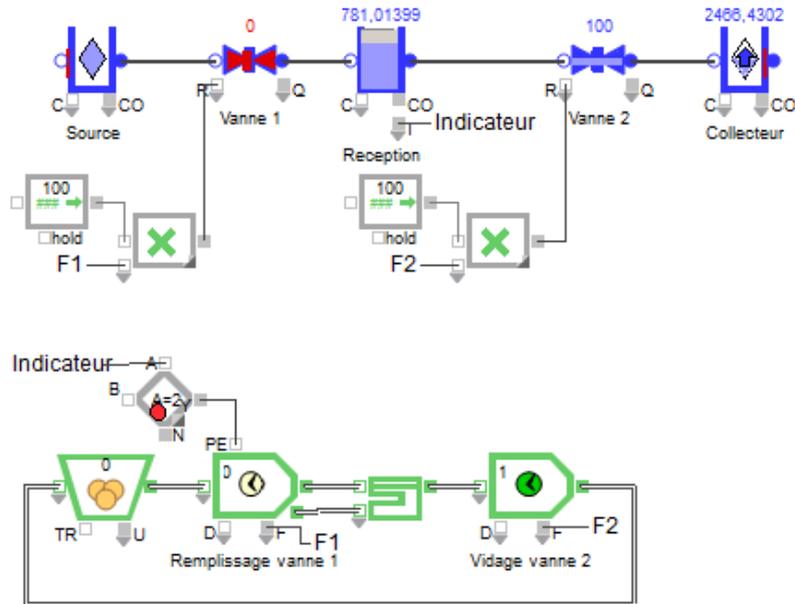


Dans ce modèle, des blocs Constant (bib. Value) envoient un débit potentiel de 100 aux vannes. La valeur est multipliée par la valeur émise par le connecteur F (Full) de l’activité, qui vaut soit 1 soit 0, résultant un débit de 100 ou de zéro, selon que l’entité est dans le bloc ou non.

Notez que la cuve Réception peut être pleine même si l’activité Remplissage Vanne 1 contient encore l’entité. Comme la cuve ne peut accepter davantage de flux si elle est pleine, il s’arrête mécaniquement.

Le flux qui contrôle des entités

Ce modèle est une variante du modèle ci-dessus, où l’état de la cuve Réception agit sur l’activité de remplissage. Dans ce modèle, quand la cuve Réception est pleine, elle le signale au bloc Decision (bib. Value). Ce bloc à son tour agit sur l’activité Remplissage Vanne 1, qui préempte l’entité dont la présence a ouvert la Vanne 1 et fait que la cuve s’est remplie. L’entité préemptée quitte le bloc et passe dans l’activité Vidage Vanne 2. Aucun temps n’est perdu dès que la cuve est remplie.



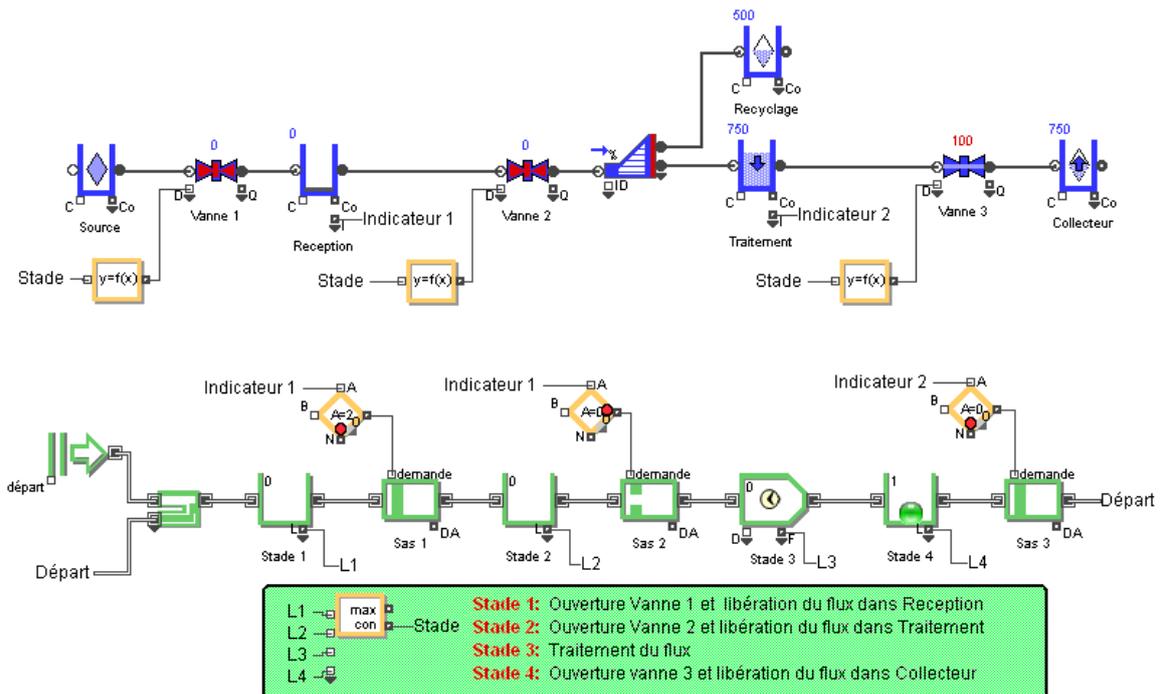
Contrôles croisés

Le modèle *Step The Flow Process* combine les deux logiques précédentes.

Dans ce modèle une entité “cyclique” déclenche l’ouverture et la fermeture des vannes, tandis que l’état de la cuve contrôle l’ouverture et la fermeture de portes pour l’entité. La logique de contrôle est circulaire : l’emplacement de l’entité définit une certaine étape ; l’état du modèle contrôle l’ouverture et la fermeture des vannes ; les vannes agissent sur le niveau du flux ; le niveau du flux contrôle l’emplacement de l’entité.

La partie haute du modèle contrôle l’entité. Les connecteurs I (Indicateur) des deux Tank (bib. Rate) contrôlent trois blocs Gate (bib. Item) utilisés dans le circuit de l’entité en partie basse du modèle. La Gate 1 ne s’ouvre que lorsque la cuve Réception est pleine. Inversement, la Gate 2 ne s’ouvre que lorsque la cuve Réception est vide. Enfin, la Gate 3 ne s’ouvre que lorsque la cuve Traitement est vide.

La partie basse du modèle contrôle le flux. Dans ce cas, les trois blocs Valve (bib. Rate) sont contrôlées par la longueur de trois blocs Queue (bib. Item). Il n’y a qu’une seule entité qui circule en boucle.



Par conséquent, la longueur de chaque file alterne entre 1 et 0. Par exemple, Vanne ne s’ouvre que lorsque la longueur de la file Stade 1 vaut 1. La même logique s’applique aux Vanne 2 et Vanne 3.

L'activité dans le modèle se divise en quatre étapes : ouvrir Vanne 1 pour remplir Réception ; ouvrir Vanne 2 pour remplir Traitement ; traiter le flux ; ouvrir Vanne 3 pour remplir Collecteur.

Utiliser bloc Interchange pour mélanger flux et entités

Le bloc Interchange est unique car il permet une interface directe entre flux et entités.

En de nombreuses occasions il est nécessaire que les entités puissent stocker et transporter le flux d'un endroit d'un modèle flux à un autre. Par exemple, l'emploi des attributs servira à distinguer le flux transporté par tel entité du flux transporté par telle autre, tandis que les unités de flux ne peuvent être distinguées les unes des autres sans cette transition par les blocs discrets.

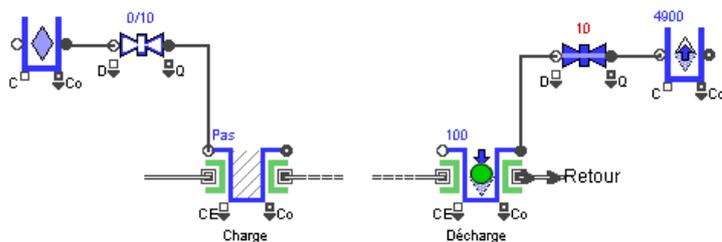
Règles de l'Interchange

Le bloc Interchange suit quelques règles, quel que soit son mode :

- Les connecteurs discrets du bloc Interchange doivent être tous les deux liés.
- Au moins l'un des connecteurs flux (entrée ou sortie) du bloc Interchange doit être lié (les deux peuvent l'être).
- La capacité à stocker des entités du bloc Interchange est de 1 uniquement.
- Le bloc Interchange prend ou libère son entité et son flux de manière instantanée.

Configuration du connecteur flux

Le comportement du bloc Interchange dépend fondamentalement de la manière dont son flux entrant et sortant sont liés. Un seul de ces connecteurs doit être lié. Si c'est le connecteur du flux entrant, les entités qui arrivent sont remplies par le flux ; si c'est le connecteur du flux sortant, les entités qui arrivent ne peuvent que décharger du flux.



Conditions de libération de l'entité

Les conditions qui déterminent quand une entité peut quitter le bloc Interchange sont les mêmes dans les deux modes. Le connecteur Preempt peut aussi déclencher une libération.

Il y a 5 options définissant quand libérer l'entité :

- **when contents >= Target (load process)** [quand contenu >= Cible (chargement)]. L'entité doit être remplie avec une certaine quantité de flux avant d'être libérée. La quantité cible est entrée dans le dialogue.
- **when contents <= Target (unload process)** [quand contenu <= Cible (déchargement)]. L'entité doit vider une certaine quantité de flux avant d'être libérée. La quantité cible est entrée dans le dialogue.
- **as soon as possible** [dès que possible]. Libère l'entité dès qu'elle peut être acceptée en amont, quel que soit le niveau du flux.
- **based on secondary release conditions** [en fonction de conditions secondaires] :
 1. **Release item when inflow attribute value changes** [quand la valeur de l'attribut en entrée change]. Libère l'entité quand un changement de valeur d'attribut est repéré (une valeur vide peut être acceptée).
 2. **Release item when "Preempt" connector value** [sur message du connecteur préemption]. Libère l'entité quand une valeur spécifique est reçue sur le connecteur PE (preempt).
- **when level reaches indicator** [quand le niveau atteint indicateur]. Le niveau du flux doit atteindre un certain point avant de permettre la libération. Des indicateurs (segments indiquant les niveaux) doivent être saisis dans l'onglet Indicators pour utiliser cette option.

Préemption

Les conditions qui déterminent quand une entité peut quitter le bloc peuvent être surpassées à tout moment par le connecteur préemption.

Release item when "Preempt" connector value \geq 3

Lorsque le connecteur préemption reçoit une valeur correspondant aux options de préemption sélectionnées dans l'onglet Item/Flow, cela déclenche une préemption.

- ☞ Pour se débarrasser de l'entité dès qu'elle sort, utilisez un bloc Exit (bib. Item) juste après de bloc Interchange. Pour avoir toujours une entité présente au contraire, utilisez un bloc Create qui crée des entités indéfiniment.

Les modes de l'Interchange

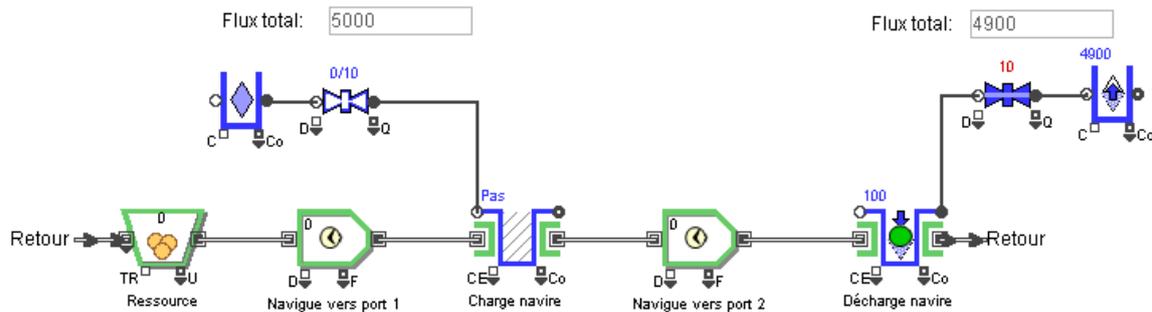
Tank only exists...

Dans ce mode, la capacité de l'Interchange à contenir du flux dépend entièrement de la présence d'une entité. L'entité est un "conteneur" d'une certaine capacité. Quand il entre dans un bloc Interchange, il peut charger ou décharger du flux. En l'absence de l'entité, le bloc Interchange n'a aucune capacité. Deux comportements importants résultent de la libération d'une entité lorsque le bloc est dans ce mode :

- Lorsque les conditions de libération sont remplies, l'entité qui part prend avec elle tout le flux qui se trouve dans le bloc à cet instant.
- Jusqu'à ce qu'une autre entité arrive, l'Interchange n'est plus passant du tout. Il n'est plus stockeur, et devient bloquant.

Modèle Shipping

Dans cet exemple, un bateau vide (une entité) arrive au quai de chargement (bloc Interchange) où il doit être rempli. Le bateau plein part pour un certain temps (représenté par une activité) avant d'atteindre un autre port (un autre bloc Interchange). Sa charge est alors déchargée.



Pour simuler le chargement, une quantité de flux passe par le connecteur du flux entrant pour charger le « bateau ». L'entité part avec cette quantité. À l'inverse, dans le second bloc Interchange, le flux est envoyé par le connecteur du flux sortant.

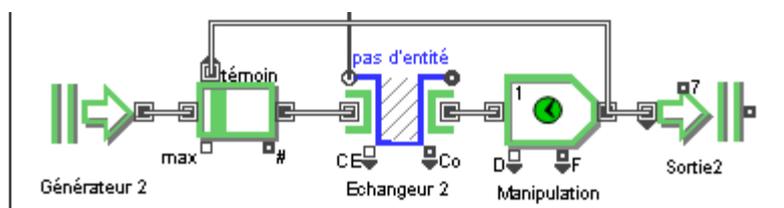
L'entité possède un attribut Quantity d'une valeur 1000 définissant sa capacité. Durant le chargement, l'Interchange libère l'entité lorsque 1000 unités de flux ont été prises. Le second bloc Interchange libère l'entité lorsque le bateau est vide, c'est à dire ayant produit 1000 unités de flux. En fin de simulation, l'entité n'est pas encore libérée car elle contient encore 100 unités de flux.

Modèle Yogurt Production

Dans ce modèle, des entités/palettes vides sont générées aléatoirement par un bloc Create (bib. Item). L'arrivée d'une palette fait que le bloc Interchange a une capacité (capacité maximum = 24 dans le dialogue). La palette pleine de 24 cartons quitte le bloc.

Modèle Yogurt Changeover

Similaire au précédent, ce modèle ajoute un temps de changement pour qu'un opérateur enlève la palette pleine et la remplace par une palette vide (toujours disponible).



Dans ce modèle, des blocs Activity (bib. Item) représentent les deux minutes de manipulation. La palette quittant le bloc Activity informe un bloc Gate (bib. Item) qui laisse entrer une nouvelle entité provenant du bloc Create. L'Interchange ainsi n'a pas de palette à remplir pendant la durée de la manipulation.

Tank is separate...

Dans ce mode, la capacité de l'Interchange à contenir du flux est semblable à celle d'un bloc Tank, qu'une entité soit présente ou non. L'Interchange peut avoir une quantité initiale, sa capacité est définie dans le dialogue ou par un connecteur, tout comme ses débits maximum.

Sa source de flux est double. Quand le bloc Interchange est en mode *cuve de stockage*, son flux peut provenir soit du connecteur du flux entrant soit de l'arrivée d'une entité qui décharge du produit, et il libère du flux soit par son connecteur du flux sortant soit « chargé » sur une entité qui part.

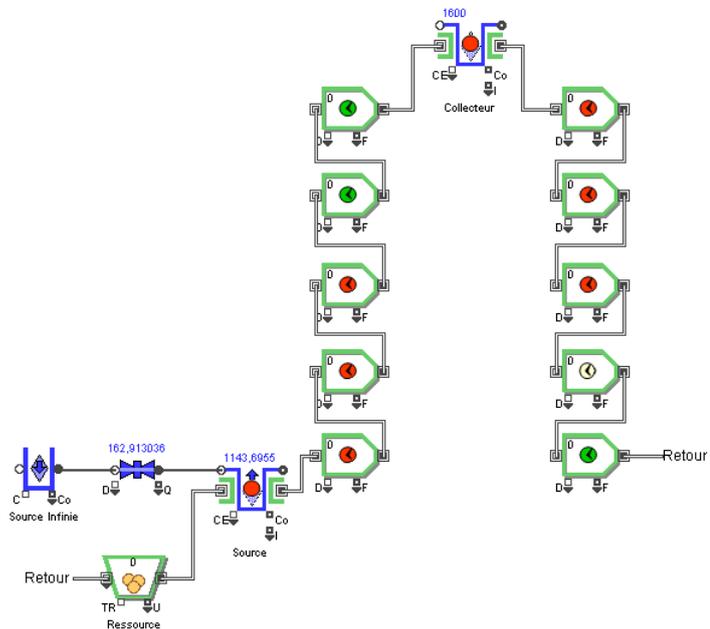
- ☞ Quand le bloc Interchange est dans ce mode, une entité chargée libère instantanément tout son contenu dès qu'elle arrive. En fonction d'autres paramètres du bloc, elle pourra aussi emporter du flux en partant. Dans ce cas, le niveau du bloc Interchange baisse subitement lorsque l'entité part. Une entité dont la charge dépasse la capacité du bloc ne pourra pas entrer dans l'Interchange.

Modèle Bucket Elevator 1

Ce modèle Bucket Elevator 1 simule un ensemble de godets (entités) puisant dans une source (Interchange) puis transportant l'eau dans un réservoir en hauteur (autre Interchange) et retournant vides jusqu'à la source. C'est le fonctionnement d'une noria.

Les deux blocs Interchange sont en mode *Tank is separate*. La source comme le réservoir ont la capacité de contenir de l'eau à tout instant.

Il y a dix godets pouvant contenir chacun 100 litres (bloc Resource Item et attribut Capacity). Il y a dix emplacements dans un "pseudo-convoyeur" – chaque emplacement représenté par un bloc Activity pouvant contenir une entité. Le délai de chaque activité est 1/10 de la valeur lue sur l'attribut Speed de l'entité.



Même si la source comporte moins de 100 litres, les godets avancent et prennent autant d'eau que possible. Lorsqu'un godet atteint le réservoir, il se vide instantanément et poursuit son circuit.

Modèle Bucket Elevator 2

Dans ce modèle, les blocs flux en haut et en bas du modèle représentent des cours d'eau. En bas, une certaine quantité d'eau est prélevée, mais le reste continue de couler. De même en haut.

Les deux blocs Interchange ont une capacité de 1000. En fonction de paramètres dans l'onglet Indicateurs, les niveaux d'eau affectent les débits sur la Valve qui suit chacun des blocs.

Les indicateurs permettent d'indiquer dans quelle plage tombe le niveau courant du flux (les plages sont définies dans l'onglet Indicateurs du bloc Source). Un bloc Equation (bib. Value) observe le connecteur I du bloc Interchange I pour déterminer le niveau de l'eau. Le bloc ajuste le débit maximum de la Valve en fonction de cette information et d'une équation.

Bibliothèque Rate : Notions diverses

Des concepts dont les autres chapitres n'ont su parler

Ce chapitre examine:

- La précision des calculs
- L'utilisation d'un biais pour donner préférence à certains composants ou portions d'un modèle, et comment cela affecte les débits réels
- Options liées au bloc Executive
- Abréviation et signification des connecteurs
- Types d'informations affichées par l'animation

Précision

Une zone PL est constituée d'un ou plusieurs sections de flux ; elle comprend toutes les sections de flux pour lesquelles le bloc Executive a été informé que les débits réels pouvaient changer. La zone PL est associée à un programme linéaire (PL) responsable du calcul du débit réel pour chaque section contenue dans la zone. La zone PL et les calculs PL sont examinés en fin de section.

La précision mathématique maximum pour une zone PL est de 12 chiffres. Comme un PL peut être responsable du calcul de plusieurs débits réels pour ses sections de flux et comme la précision PL est de 12 chiffres, certains problèmes peuvent survenir dans le calcul des débits réels. Par exemple, si une zone PL contient deux sections de flux où le débit réel de la section de flux était de 1,000,000 unités de flux par unité de temps (FPT), le débit réel de la seconde section ne peut être inférieure à 0.0001 FPT.

☞ Pour être sûr de la précision pour toutes les sections de flux, ne séparez pas deux débits réels d'une même zone PL de plus de 12 chiffres de précision.

Biaiser le flux

L'architecture flux inclut une fonction appelée biais – une méthode pour indiquer une préférence pour que le flux passe par une route plutôt que par une autre.

☞ Le biais n'est pertinent qu'en cas de mélange ou division du flux.

Les avantages de ce concept sont les suivants :

Il permet de définir des préférences sur la manière dont le flux circule dans une partie du modèle.

Il donne de la souplesse pour résoudre des conflits entre plusieurs branches candidates.

Le biais est présent dans un modèle à partir du moment où vous utilisez :

- Un bloc Bias dans un modèle contenant des blocs Merge ou Diverges qui sont en mode non fixes.
- Un bloc Merge ou un bloc Diverge (modes Distributional, Priority ou Sensing uniquement). Comme le biais influence la manière dont le flux est distribué, il est pris en compte par le calcul PL global et peut affecter les débits réels.

Rang

Si un bloc d'un modèle inclut un biais, il possède un numéro dans un classement qui comprend tous les autres blocs avec biais. Chaque bloc avec biais figure dans une liste allant du plus fort niveau de priorité (en haut) au plus faible. Les blocs en haut de la liste ont un classement supérieur à 0. Le classement descend au fur et à mesure que les chiffres montent ; un classement vide ou négatif est ignoré.

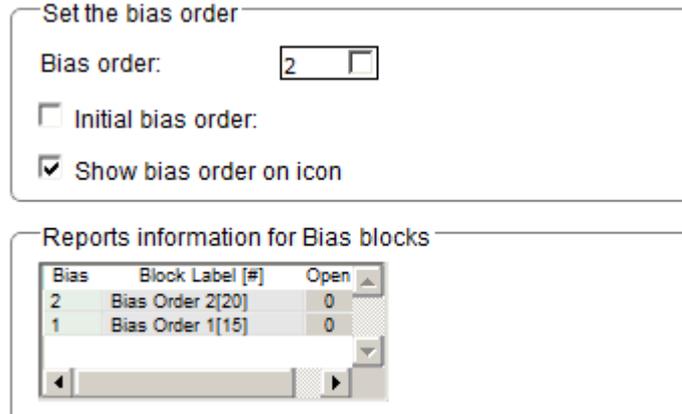
L'ordre de préférence étant pris en compte durant le calcul PL des débits réels, changer la préférence équivaut souvent à obtenir des débits réels différents.

☞ Le biais d'un bloc Bias est par définition plus fort que l'effet de biais d'un bloc Merge ou Diverge. Donc les blocs Bias seront toujours mieux classés que les blocs Merge et Diverge.

Le bloc Bias

Le bloc Bias permet d'indiquer une préférence pour que le flux suive un trajet plutôt qu'un autre. Le bloc Bias fait passer autant de flux que possible là où il se trouve. Si un modèle comporte plusieurs blocs Bias, chacun a son propre classement.

Si des blocs Merge et Diverge utilisent des règles non fixes pour choisir leur branche, le bloc Bias pourra être utile. Si des blocs Merge et Diverge utilisent des règles fixes, il n'est pas possible de biaiser le flux par le bloc Bias.



Paramètres de dialogue

Le rang de préférence du bloc peut être donné directement dans son dialogue ou modifié dynamiquement par le connecteur B (biais) ou encore en liant le paramètre de rang à une base de données ExtendSim. Vous pouvez entrer un rang initial qui a effet jusqu'à ce qu'une valeur soit émise dynamiquement.

Le dialogue Bias ci-dessus est l'un des deux blocs Bias d'un modèle. Le bloc a la plus faible préférence, comme indiqué par *Bias order: 2*.

La table indique pour chaque bloc Bias, son rang, son nom ou label et son numéro. Par la colonne *Bias* vous pouvez changer l'ordre de biais pour chacun des blocs.

Calcul du débit réel

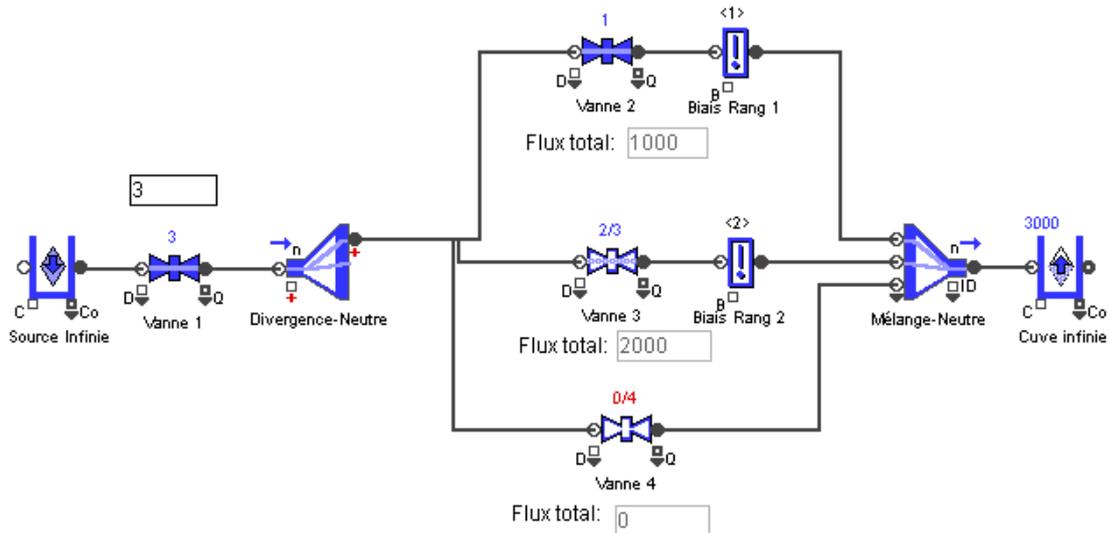
Les préférences données par les blocs Bias ont un effet sur le calcul du débit réel. Chaque bloc à tour de rôle calcule le débit réel maximum pouvant passer par lui, sans prendre en compte les préférences de blocs ayant un moindre rang.

- Le rang de préférence des blocs Bias peut changer dynamiquement durant la simulation
- Si plusieurs blocs Bias ont le même rang, on ne peut prédire comment sera distribué le flux entre les débits réels. Le modèle fera un choix.
- Si un bloc Bias a un rang vide ou est inférieur ou égal à 0, il n'exprime aucune préférence.

Modèle Prioritize With Bias Blocks

Ce modèle montre comment employer des blocs Bias. Il est similaire au modèle Competing Requests for Flow.

Ce modèle utilise des blocs Bias pour indiquer un trajet préféré, plutôt que de jouer sur le rang des blocs Merge et Diverge en mode Priority.



Dans le modèle *Prioritize With Bias Blocks*, les blocs Merge et Diverge sont en mode Neutral et le bloc Bias indique les préférences. Les résultats sont les mêmes.

☞ Ce modèle se trouve dans le dossier \Exemples\Item Rate\Merge and Diverge.

Les blocs Merge et Diverge

Nous avons vu que certains modes Merge/Diverge utilisent une règle fixe pour choisir une branche. Pour les autres modes Merge/Diverge, les règles de flux sont contextuelles.

Le concept de biais ne s'applique qu'aux blocs Merge ou Diverge en modes non fixes : Distributional, Priority ou Sensing. Pour éviter des conflits de requêtes, il est nécessaire dans ces modes de spécifier un ordre de biais.

Modes fixes

Les modes Batch/Unbatch, Proportional et Select utilisent une règle fixe pour choisir une branche. Par exemple en mode Proportional le flux respecte les proportions définies dans le dialogue, quoi qu'il arrive dans le reste du modèle. L'option de biais est inactive dans un tel cas.

☞ Si le modèle contient des blocs Merge et Diverge en modes Batch/Unbatch, Proportional ou Select, le biais n'a pas d'impact sur les débits réels.

Modes contextuels

Les modes Distributional, Neutral, Priority ou Sensing utilisent une règle contextuelle pour choisir une branche.

Un bloc Merge ou Diverge en mode Priority, peut par exemple influencer sur le flux comme suit:

- Prenant en compte la quantité de flux qu'il peut obtenir, le bloc essaie tant que possible de l'envoyer sur une branche prioritaire.
- Il exprime une "préférence" qui ne sera peut-être pas respectée.

☞ Si le modèle contient des blocs Merge ou Diverge en mode Distributional, Priority ou Sensing, la préférence doit être spécifiés (ce n'est pas requis en mode Neutral.)

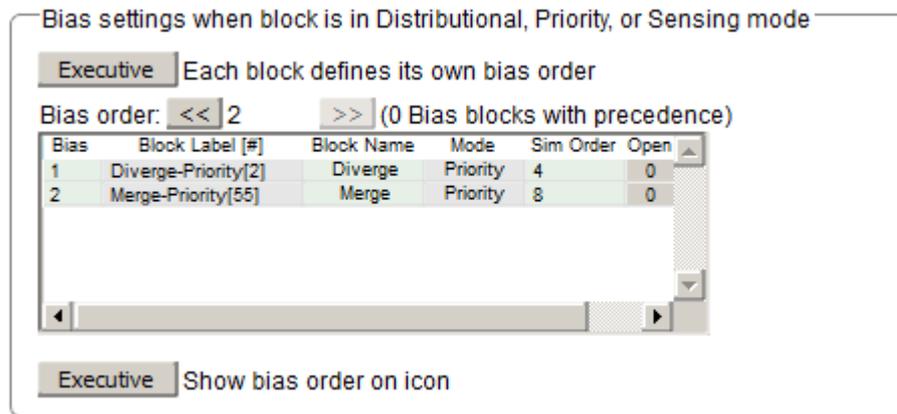
Définir le rang de préférence d'un bloc Merge ou Diverge

En mode Distributional, Priority ou Sensing, les blocs Merge et Diverge doivent exprimer un rang de préférence.

Par défaut, le bloc Executive indique *Bias order: defined by Simulation Order*. Ainsi le rang de chaque bloc Merge et Diverge est automatiquement déterminé par l'ordre de la simulation, lui-même défini par Simulation > Paramètres > onglet Continu ; Ordre du flux par défaut. Cet ordre du flux ne doit pas en général être abandonné.

Vous pouvez entrer directement un rang pour un bloc Merge ou Diverge. Pour cela, commencez par changer le paramètre par défaut du bloc Executive en *Bias order: each block defines its own*. Ensuite :

- Dans l'onglet Modèle, utilisez les boutons "<<" et ">>" pour changer le rang de préférence du bloc. Cela change la position du bloc dans la table.



- Ou bien, dans l'onglet *Discrete Rate* du bloc Executive, sélectionnez la ligne contenant le bloc désiré, puis utilisez les flèches << et >> pour changer sa position dans la table.

Table des préférences

La table de l'onglet Model Settings d'un bloc Merge ou Diverge affiche le rang du bloc, son nom ou label, son mode et son ordre dans la Simulation.

- Si le modèle est en *Bias order: defined by Simulation Order*, la table est désactivée car les préférences sont déterminées automatiquement.
- Si le modèle est en *Bias order: each block defines its own*, la table est activée pour les blocs Merge ou Diverge en mode Distributional, Priority et Sensing uniquement.

L'onglet Model Settings indique aussi le nombre de blocs Bias du modèle, car les blocs Bias ont toujours un rang supérieur aux blocs Merge ou Diverge.

Options avancées du bloc Executive

Le bloc Executive (bib. Item) pilote les modèles Rate. Il est responsable du calcul des débits réels, et centralise et coordonne les informations issues des blocs Rate.

L'onglet *Discrete Rate* du bloc Executive définit des options globales et avancées sur un modèle flux.

Options globales

Débit infini, débit nul

L'onglet Discrete Rate spécifie à partir de quel débit on considère qu'il s'agit d'un débit infini ; par défaut un débit $\geq 1e10$ est considéré comme infini. De même il permet d'indiquer en-deçà de quel débit on considère un débit nul. Ne changez ces valeurs que si vous avez une excellente raison de le faire !

Mise à jour du statut du flux

Dans la bibliothèque Rate, certaines valeurs changent en permanence dans le temps. La fréquence de mise à jour des valeurs peut être contrôlée par des options pour les blocs Convey Flow, Diverge, Interchange, Merge, Tank et Valve. Il définit combien souvent les informations suivantes sont mises à jour :

- Le niveau du flux dans les blocs stockeurs (Convey Flow, Interchange et Tank),
- La quantité de flux passant par les blocs Convey Flow, Diverge, Interchange, Merge, Tank et Valve.

Au minimum, le système met à jour le statut des flux lorsqu'il en a besoin. Deux autres options de calcul sont proposées. L'option sélectionnée dans le bloc Executive s'affiche dans chaque onglet Options avec les choix associés. Les choix du menu sont :

- **Only when necessary** (Uniquement quand nécessaire). Ce paramètre par défaut est le plus efficace en termes de rapidité des calculs. Il implique des recalculs:
 - Lorsqu'un bloc est défini comme faisant partie d'une zone PL
 - Lorsqu'un bloc crée un événement interne tel que d'obtenir un nouvel indicateur
 - Lorsqu'un bloc reçoit un message actif sur l'un de ses connecteurs de sortie continu

- **Each block defines how often** (Définie par chaque bloc). L'onglet Options de chaque bloc permet alors de cocher ou non *Update animation and results at each event*.
- **Ach block at each step** (Pour chaque bloc à chaque événement). À utiliser avec prudence car entraîne énormément de calculs.

Global options for discrete rate models

Any rate \geq is considered infinite

Any effective rate \leq is considered to be zero

Blocks update flow status:

- each block defines how often
- each block at each step

Valve animates and reports blocking and starving information

Valve animates and reports blocking and starving information

Cette option (La Valve indique et anime les états de blocage et pénurie) n'affecte que l'animation et les informations produites par les blocs Valve. Une Valve peut être limitante, non limitante, bloquée, en pénurie ou bloquée et pénurie. Par défaut, l'onglet Results de la Valve et son connecteur de sortie S (statut) ne rapportent que les états limitant (0) ou non limitant (1). L'onglet Results donne aussi le pourcentage de temps où le bloc a été limitant ou non limitant.

Lorsque l'option globale est cochée, l'onglet Results et le connecteur S émettent d'autres valeurs :

- limitant (0)
- pénurie (1)
- bloquée (2)
- pénurie et bloquée (3)

Un tableau complet de tous les états (et valeurs émises sur S)

0	Advanced Status	(Limit,Rate,Starve,Block)	Input	Val	Cumulative time
0	Limit only (r>0)	(Limit,rate>0, --, --)	0	0	
1	Starved only (r>0)	(--,rate>0,Starve, --)	1	0	
2	Blocked only (r>0)	(--,rate>0, --,Block)	2	0	
3	Starved and Blocked (r>0)	(--,rate>0,Starve,Block)	3	0	
4	Limit and Starved (r>0)	(Limit,rate>0,Starve, --)	4	0	
5	Limit and Blocked (r>0)	(Limit,rate>0, --,Block)	5	0	
6	Limit, starved, Blocked (r>0)	(Limit,rate>0,Starve,Block)	6	0	
7	Limit only (r=0)	(Limit,rate=0, --, --)	7	0	
8	Starved only (r=0)	(--,rate=0,Starve, --)	8	0	
9	Blocked only (r=0)	(--,rate=0, --,Block)	9	0	
10	Starved and Blocked (r=0)	(--,rate=0,Starve,Block)	10	0	
11	Limit and Starved (r=0)	(Limit,rate=0,Starve, --)	11	0	
12	Limit and Blocked (r=0)	(Limit,rate=0, --,Block)	12	0	
13	Limit, starved, Blocked (r=0)	(Limit,rate=0,Starve,Block)	13	0	

☞ Connaître précisément le statut d'une Valve est bien utile durant la construction d'un modèle et pour le débogage. Mais cela peut ralentir la simulation, donc ce n'est pas l'option par défaut.

Gérer les unités de flux : Manage flow units

Cette section de l'onglet *Discrete Rate* centralise la manière dont les unités de flux sont nommées, ajoutées ou supprimées dans un modèle. Choisissez une unité de flux dans la table et cliquez sur le bouton approprié.

Options avancées

Les options avancées (Advanced options) de l'onglet Discrete Rate ne s'appliquent qu'à des situations spécifiques :

- blocs Merge ou Diverge en mode Distributional, Priority ou Sensing
- blocs Merge en mode Proportional dans le cas d'une boucle vide

Le premier ensemble d'options détermine comment le rang de préférence est défini pour certains blocs Merge et Diverge et s'il est affiché sur l'icône du bloc.

Choix de l'ordre de biais

Pour un bloc Merge et Diverge en mode Distributional, Priority ou mode Sensing, le rang de préférence est défini par :

- L'ordre de la simulation. C'est le choix par défaut.
- Chaque bloc. L'utilisateur peut choisir le rang de préférence dans chaque bloc Merge et Diverge (mode Distributional, Priority ou Sensing uniquement).

Afficher l'ordre de biais

Pour les blocs Merge et Diverge en mode Distributional, Priority ou mode Sensing, un menu déroulant propose des choix pour afficher le rang de préférence :

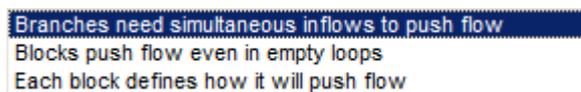
- Afficher l'ordre de préférence sur l'icône
- Ne pas montrer la préférence sur l'icône
- Chaque bloc choisit de montrer ou non son rang

Si le rang est affiché, c'est avec le format <#>, où # est le classement dans la préférence.

Blocs Merge en mode Proportional

Pour les blocs Merge en mode Proportional, il peut surgir un conflit si le bloc fait partie d'une boucle vide (ne pouvant obtenir de flux). Si les proportions sont de 1:2 par exemple, que se passe-t-il si la branche du haut fait partie d'une boucle vide et reste toujours en pénurie ?

Le bloc Executive propose trois options pour gérer ces situations :



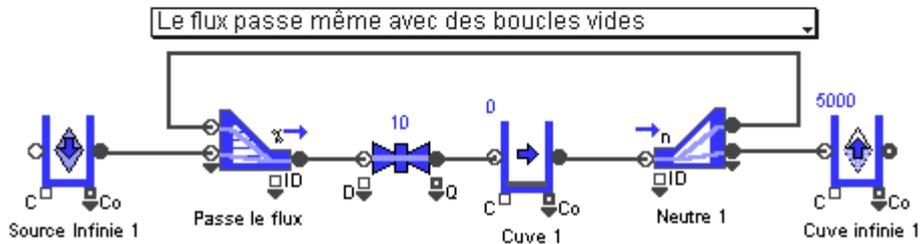
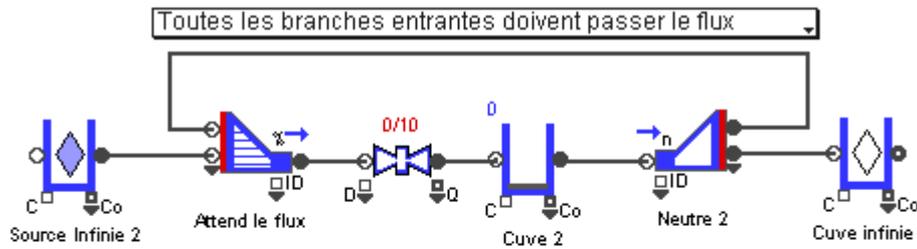
- **Branches need simultaneous inflows to push flow** (Toutes les branches entrantes doivent passer le flux). Avec ce paramètre par défaut, le flux est stoppé sur toutes les branches entrantes si l'une d'elles fait partie d'une boucle vide.
- **Blocks push flow even in empty loops** (Le flux passe même avec des boucles vides). Permet aux branches alimentées de faire passer le flux. Le résultat est que la branche qui fait partie d'une boucle vide obtiendra du flux.
- **Each block defines how it will push flow** (Chaque bloc définit comment il pousse le flux). Fait s'afficher un menu supplémentaire dans l'onglet Modèle des blocs Merge en mode Proportionne, proposant deux choix déjà connus :
 - **Blocks push flow even in empty loops** (Le flux passe même avec des boucles vides). C'est le choix par défaut ; il permet aux branches ayant de la matière de la laisser passer.
 - **Branches need simultaneous inflows to push flow** (Toutes les branches entrantes doivent passer le flux). Le flux est stoppé sur toutes les branches si au moins une d'entre elles n'obtient pas de matière.

Votre choix se fera au cas par cas suivant votre modèle.

Modèle Merge mode Proportionnel 2

Dans ce modèle deux lignes sont identiques sauf sur la manière de gérer les boucles vides des blocs Merge. Le bloc Executive laisse chaque bloc Merge définir son comportement.

Le bloc Merge du haut (Attend le flux) spécifie que *Branches need simultaneous inflows to push flow*, tandis que le bloc Merge du bas (Passe le flux) spécifie que *Blocks push flow even in empty loops*. Le flux est bloqué au niveau du bloc Merge en haut, mais il passe en bas.



Connecteurs courants des blocs Rate

Les blocs Rate ont des connecteurs de flux entrant et sortant ainsi que des connecteurs continus d'entrée et de sortie. Le bloc Interchange possède aussi des connecteurs discrets d'entrée et de sortie. Les connecteurs flux transmettent des informations de débit réel d'un bloc à un autre. Les connecteurs discrets passent des entités. Les connecteurs continus servent à manipuler des valeurs, et ont une lettre ou une abréviation pour les distinguer, et qui sont parfois contextuelles.

Voici les labels de connecteurs les plus utilisés:

Connecteur	Place	Signification
AL	Sortie	Longueur d'accumulation – pour un Convoyeur flux, la distance entre sa fin et le point d'accumulation.
AQ	Sortie	Quantité accumulée – la quantité de flux accumulée sur la longueur d'accumulation d'un Convoyeur flux.
B	Entrée	Classement dans l'ordre de biais
C	Entrée	Capacité
CO	Sortie	Contenu
D	Entrée	Délai
DR	Sortie	Débit potentiel de la demande en aval
factor	Entrée	Facteur de conversion du bloc Change Units
G	Entrée	Quantité de flux (objectif de quantité) ou durée en temps (objectif de durée) d'une Valve. Peut aussi servir à lancer un nouvel objectif, en fonction des paramètres de la Valve.
G#	Sortie	Numéro de l'objectif
GD	Sortie	Durée de l'objectif
GS	Sortie	Statut de l'objectif (goal status) 0 – pas d'objectif 1 – démarre 2 – en cours 3 – terminé 4 - interrompu
GQ	Sortie	Quantité de l'objectif
GO	Entrée	Déclenche une mise à jour du statut: 0 ou 1 (Sensor et Merge/Diverge)
I	Sortie	Indicateur (Convey Flow, Interchange ou Tank)

IC	Entrée	Capacité de l'entité
ICO	Entrée	Contenu de l'entité
ID	Entrée	ID de la branche entrante/sortante de Merge/Diverge en mode Select
IT	Entrée	Valeur cible pour libérer l'entité
L	Sortie	Longueur de la file d'entités : 0 ou 1
LE	Sortie	Niveau du contenu
NB	Sortie	Nombre bloqué : 0 ou 1
PE	Entrée	Prémption entité
PT	Sortie	Temps de traitement
Q	Sortie	Quantité cumulée
R	Entrée	Débit maximum: (Notez que le débit réel est indiqué directement par les connecteurs d'entrée et de sortie du flux.)
S (sur Convey Flow)	Sortie	Statut: 0 = vide 1 = intermédiaire 2 = plein
S (sur Interchange ou Tank)	Sortie	Tendance du statut (Interchange): -1 = diminution 0 = stable 1 = augmentation
S (sur Valve)	Sortie	Statut [si la Valve n'indique que limitant ou non limitant]: 0 = limitant 1 = non limitant
S (sur Sensor ou Valve)	Sortie	Statut [si la Valve indique son statut complet]: 0 = limitant (Valve) ou inutilisé (Sensor) 1 = pénurie 2 = bloquée 3 = pénurie et bloquée
S (0-n)	Sortie	Capteurs – numérotés de 0 à n (Convey Flow)
SP	Entrée	Paramètre de vitesse
SP	Sortie	Vitesse réelle
SR	Sortie	Débit potentiel de la fourniture en amont
start	Entrée	Démarrage hystérésis
stop	Entrée	Stoppe hystérésis
TL	Sortie	Temps cumulé où la Valve était limitante, débit réel > 0
TU	Sortie	Temps cumulé où la Valve était non limitante, débit réel > 0
TLO	Sortie	Temps cumulé où la Valve était limitante, débit réel = 0
TUO	Sortie	Temps cumulé où la Valve était non limitante, débit réel = 0

Animation

Les blocs Rate s'animent durant la simulation si Simulation > Animation 2D est sélectionné. La signification de l'animation est expliquée ci-dessous.

☞ Les modèles illustrant cette section figurent dans le répertoire \Exemples\Item Rate\Miscellaneous.

Tank

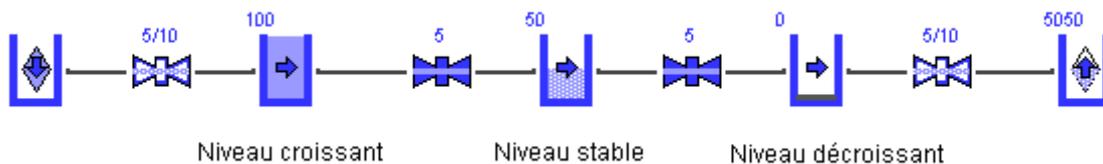
L'animation indique le niveau de la cuve ainsi que la tendance du niveau.

Le tableau ci-dessous montre l'animation du niveau du Tank dans différentes conditions.

Capacité finie	Capacité infinie	Comportement spécifique
 <p>Pleine (contenu 100, capacité 100)</p>	 <p>Pleine</p>	 <p>Dépassement (contenu 1,000, capacité 100)</p>
 <p>Non vide (contenu 50, capacité 100)</p>	 <p>Non vide (10 unités)</p>	 <p>Dépassement (contenu 1,000, pas de capacité)</p>
 <p>Vide (contenu 0, capacité 100)</p>	 <p>Vide</p>	 <p>Sans capacité et vide</p>

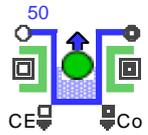
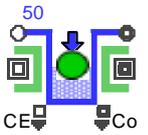
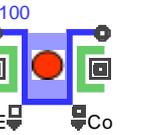
Tendance du niveau

Une flèche sur l'icône indique la direction du changement de niveau du Tank. S'il n'y a pas de flèche, cela signifie que le Tank ne reçoit ni ne fait partir du flux.



Interchange

Le bloc Interchange a la même animation qu'un Tank. De plus, une balle apparaît au milieu de l'icône lorsqu'une entité est présente dans le bloc. La balle est rouge si l'entité est prête pour quitter le bloc mais bloquée en aval. Sinon elle est verte. Si l'option *Tank only exists...* est choisie et qu'il n'y a pas d'entité dans le bloc, l'icône est hachurée en blanc.

 <p>Entité se remplit (contenu 50, capacité 100)</p>	 <p>Entité se vide (contenu 50, capacité 100)</p>	 <p>Entité bloquée en aval (contenu 99, capacité 100)</p>	 <p>Pas d'entité (Cuve Tank only exists...)</p>
---	--	---	--

Valve

Une Valve indique ses débits réels et maximums si l'animation est active. Si l'option *Valve animates and reports blocking and starving information* (La Vanne indique et anime les états de blocage et pénurie) n'est pas sélectionnée dans l'onglet *Discrete Rate* du bloc *Executive*, la Valve n'affiche que son statut limitant ou non limitant. Si cette option est sélectionnée, elle affiche aussi son statut pénurie et bloquée.

Affichage du statut limitant et non limitant

Par défaut, la Valve n'anime que l'information sur son statut limitant et non limitant.

Non-Limitante	Limitante
<p>5/10</p>  <p>Débit réel 5, débit maximum 10</p>	<p>5</p>  <p>Débit réel 5, débit maximum 5</p>
<p>0/10</p>  <p>Pas de flux, débit maximum 10</p>	<p>0</p>  <p>Pas de flux, débit maximum 0</p>

Limitante signifie que le débit réel égal le débit maximum. Dans ce cas, la Valve sert à limiter le flux. Lorsqu'une Valve est non limitante, le débit réel est inférieur au débit maximum de la Valve. La Valve n'a pas d'impact sur la progression du flux.

- Lorsque le débit réel n'est pas le débit de contrainte (maximum), les deux chiffres du débit réel/contraignant s'affichent au-dessus de l'icône. Si les deux sont identiques, un seul chiffre apparaît.
- Si la Valve est limitante et que débit réel est supérieur à 0, l'intérieur de l'icône est bleu. Si la Valve est limitante to 0, l'intérieur de l'icône est rouge. Si la Valve n'est pas limitante, l'intérieur de l'icône est blanc.
- Lorsque le débit réel est supérieur à 0, un rectangle bleu apparaît le long de l'icône et les débits sont écrits en bleu. Lorsqu'il n'y a pas de flux, le rectangle bleu n'apparaît pas le long de l'icône et les débits sont écrits en rouge.

Informations de statut bloqué ou pénurie

Si l'option *Valve animates and reports blocking and starving information* est sélectionnée dans l'onglet Discrete Rate du bloc Executive, la Valve affiche aussi des informations sur son statut bloqué ou pénurie.

Si une Valve est bloquée, c'est qu'un bloc en aval empêche le flux de partir. Si une Valve est pénurie, c'est qu'un bloc en amont empêche le flux d'arriver.

S'il n'y a pas de flux, il n'y a pas de ligne horizontale traversant la Valve. S'il y a du flux et que la Valve est non limitante, la ligne horizontale traversant la Valve est bleutée. La ligne horizontale est franchement bleue ou rouge s'il y a du flux.

Limitante uniquement	Bloquée	Pénurie	Bloquée et pénurie
<p>5</p>  <p>Limitante</p>	<p>5/10</p>  <p>Non-limitante, avec flux</p>	<p>5/10</p>  <p>Non-limitante, avec flux</p>	<p>5/10</p>  <p>Non-limitante, avec flux</p>
<p>0</p>  <p>Débit maximum: 0</p>	<p>0/10</p>  <p>Non-limitante, pas de flux</p>	<p>0/10</p>  <p>Non-limitante, pas de flux</p>	<p>0</p>  <p>Non-limitante, pas de flux</p>
	<p>5</p>  <p>Limitante, avec flux</p>	<p>5</p>  <p>Limitante, avec flux</p>	<p>5</p>  <p>Limitante, bloquée et pénurie avec flux</p>

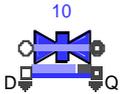
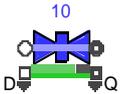
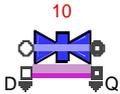
	 Limitante, pas de flux	 Limitante, pas de flux	 Limitante, bloquée et pénurie et pas de flux
--	---	---	---

- Si une Valve est limitante, la partie verticale et centrale de la Valve est bleue (si le débit maximum est >0) ou rouge (si le débit maximum = 0).
- Si la partie droite de l'icône de la Valve est bleue, le bloc est partiellement bloqué par des blocs en aval. S'il y a une ligne rouge sur la partie droite de l'icône, le bloc est complètement bloqué par des blocs en aval.
- Si la partie gauche de l'icône de la Valve est bleue, le bloc est partiellement en pénurie par des blocs en amont. S'il y a une ligne rouge sur la partie gauche de l'icône, le bloc est complètement en pénurie par des blocs en amont.

Animation de l'objectif et de l'hystérésis

Une Valve utilisant un objectif de quantité pour contrôler le flux a une ligne bleue sous son icône. Si c'est un objectif de durée, la ligne est verte. Tant que l'objectif est On, une barre de progression apparaît le long de cette ligne bleue ou verte.

Une Valve utilisant une hystérésis a une ligne violette sous son icône. Tant que l'hystérésis est active, la ligne violette est épaisse sur tout ou partie de sa longueur.

 Objectif de quantité, environ 75% terminé	 Objectif de durée, environ 75% terminé	 Hystérésis active
---	--	---

Sensor

La forme et la couleur du bloc Sensor indique si le flux est bloqué, pénurie ou les deux. Et si du flux est présent ou non.

Flux ?	Bloqué	Pénurie	Bloqué et pénurie
Oui >>>			
Non >>>			

Convey Flow

L'animation du Convey Flow montre dans quel mode est le bloc, la distribution du flux sur sa longueur, la position du point d'accumulation et d'autres informations.

Animation du mode

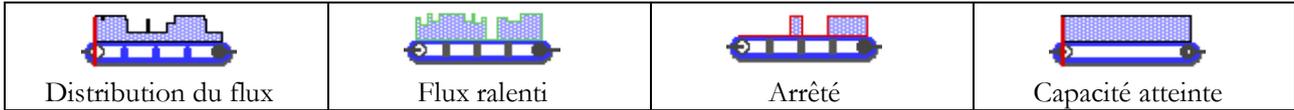
 Accumule-densité maximum	 Accumule-remplir les segments vides	 Non-accumulateur
--	---	---

Le modèle *Compare Convey Flow* compare le comportement de trois blocs Convey Flow, chacun dans un mode, subissant des débits différents. Le modèle figure dans le répertoire \Exemples\Item Rate\Delaying Flow.

Distribution du flux et autres informations

L'animation de la distribution du flux du Convey Flow est la suivante :

- Une forme bleue le long de la partie supérieure de l'icône indique la distribution du flux.
- Si le flux est ralenti, la forme bleue prend une bordure verte.
- Si le flux est arrêté, la forme bleue prend une bordure rouge.
- Si le bloc est plein, la forme est bleue dense et non en pointillés bleus.
- Si le bloc est vide, une ligne noire apparaît le long de la partie supérieure de l'icône.
- La position du point d'accumulation est indiquée par une ligne verticale rouge qui se déplace sur l'icône.



Modèles DED : Concepts avancés

Informations complémentaires pour ceux qui veulent tout comprendre

Les informations de ce chapitre ne sont pas nécessaires pour construire un modèle Rate, mais intéresseront les utilisateurs confirmés. Le chapitre examine :

- La technologie PL qui pilote globalement un modèle flux
- Comment est déterminée la zone PL
- La séquence d'événements d'un calcul PL
- Comment la préférence affecte les calculs
- Les types d'informations fournies au bloc Executive
- Comment est calculée une contrainte relationnelle
- Les débits potentiels "fourniture amont" et "demande aval"
- L'envoi de messages dans les modèles Rate

La technologie PL

Les cas traités par programmation linéaire impliquent l'optimisation d'un objectif soumis à un ensemble de contraintes. Certaines variables sont maximisées ou minimisées en fonction de l'objectif.

La technologie PL est la méthode adoptée par ExtendSim pour garantir globalement une progression maximum du flux dans le système. Cette architecture emploie une DLL intégrée Solver PL pour résoudre une série d'équations pour optimiser les débits à chaque instant de la simulation.

Généralités

Dans un modèle flux, les blocs de la bibliothèque Rate communiquent entre eux et avec le bloc Executive. À son tour, le bloc Executive communique avec un programme linéaire intégré (Solver PL). Les blocs Rate dépendent les uns des autres, s'influencent mutuellement et font partie d'un système PL global qui évalue le modèle pour calculer ses débits réels.

- Une section de flux est un ensemble de blocs liés entre eux et possédant tous le même débit réel. Établies en début de simulation, les sections de flux ne changent pas.
- Une zone PL est composée d'une ou plusieurs sections de flux ; leur configuration peut changer durant la simulation. Une nouvelle contrainte sur un bloc durant la simulation propage des messages dans toutes les sections de flux dont les débits réels pourraient changer. Cette propagation définit la zone PL à cet instant.
- Chaque section de flux dans la zone PL contribue à l'équation PL. L'objet du calcul est de déterminer les débits réels maximum pour le système, en fonction des contraintes des blocs. Le résultat est un ensemble de débits réels pour chaque section dans la zone PL à un instant de la simulation. Le système est optimisé pour que seules les sections de flux de la zone PL soient recalculées; les autres débits réels ne sont pas recalculés.
- Les calculs PL prennent en compte les contraintes critiques, les contraintes relationnelles, et le rang existant pour chaque bloc.

La zone PL

La zone PL est composée d'une ou plusieurs sections de flux. Le débit réel d'une section peut affecter le débit réel d'une autre section par des contraintes relationnelles. Lorsque survient un événement qui fait réévaluer les débits réels d'une section de flux, les blocs propagent des messages "flux" dans le modèle pour déterminer quelles autres sections de flux pourraient être impactées par le nouvel événement. Les sections de flux affectées constituent la zone PL et les sections de flux hors de cette zone PL ne sont pas incluses dans le calcul.

Les limites de la zone PL sont définies par la propagation des messages entre les blocs Rate. Tout changement des contraintes d'un bloc durant la simulation fait que le bloc informe le bloc Executive

et envoi des messages qui se propagent dans toutes les sections de flux dont les débits réels pourraient en être affectés.

Qu'un bloc soit à l'origine du calcul ou reçoive un message :

- Le bloc déclare laquelle des sections de flux qui lui est connectée fait partie de la zone PL. La propagation crée une liste des sections de flux à inclure dans la zone PL.
 - Si le bloc impose une contrainte relationnelle entre deux sections de flux ou plus, les débits réels associés au bloc sont interdépendants. Le bloc poursuit la propagation à tous ses blocs dépendants, et ainsi de suite.
- ☞ Les limites de la zone PL changeront dynamiquement durant la simulation en fonction des débits réels impliqués dans le recalcul et des contraintes relationnelles entre sections de flux.

La séquence d'événements

Tout changement des contraintes d'un bloc durant la simulation cause une réévaluation des débits réels. La séquence des actions est la suivante :

- 1) Une contrainte d'un bloc change.
 - Si le statut d'un bloc change, cela peut affecter ses débits réels entrant ou sortant. Par exemple, le Tank devenant plein crée un événement.
 - Lorsqu'un bloc réagit à de nouveaux paramètres, ses débits réels entrant ou sortant peuvent être réévalués. Par exemple, un bloc Merge en mode Select où la branche est initialement choisie reçoit un ordre de changement de branche, donc ses débits réels doivent être recalculés.
 - 2) Le bloc publie un événement immédiat pour faire réévaluer ses débits réels.
 - 3) La zone PL est déterminée par la propagation des messages de bloc.
 - A partir du bloc d'origine, les messages sont propagés à tous les blocs pouvant être affectés. Cette propagation définit les limites de la zone PL.
 - La zone PL comprend toutes les sections de flux ayant des débits réels pouvant changer durant le calcul.
 - 4) Les blocs de la zone PL se mettent à jour.
 - Chaque bloc de la zone PL met à jour la quantité de flux qu'il a fait passer depuis la dernière évaluation.
 - Si le bloc est un bloc stockeur (Convey Flow, Interchange ou Tank), il met aussi à jour la quantité de son contenu.
- Note : La fréquence de mise à jour indépendamment d'un calcul PL est définie dans le bloc Executive.
- 5) Les blocs indiquent au bloc Executive comment ils influent sur les débits réels.
 - Chaque bloc de la zone PL déclare les règles de flux (contraintes critiques et relationnelles) qu'il applique à sa section de flux.
 - Le rang de préférence du bloc est stocké dans une liste. Chaque bloc déclare son rang (s'il y a lieu) au bloc Executive. Son classement permet au bloc Executive de construire une fonction d'objectif prenant en compte l'effet de la préférence.
 - 6) Le bloc Executive détermine une fonction d'objectif.
 - L'objectif est de maximiser le débit de chaque section de flux soumise aux contraintes définies par les règles de flux du bloc.
 - Chaque variable de décision de la fonction d'objectif est le débit réel d'une section dans la zone PL. Par exemple, l'objectif pourrait être " Débit réel Maximum = 1*ER1+ 1*ER2 + 1*ER3...", où ERn est le débit réel d'une section de flux dans la zone PL.
 - S'il n'y a aucune préférence ou si tous les calculs intermédiaires relatifs au rang de préférence ont été effectués, le coefficient de chaque variable sera de 1. S'il y a une préférence, le coefficient sera autre que 1.
 - 7) Le bloc Executive communique avec le Solver PL.
 - Le bloc Executive indique au Solver PL la fonction d'objectif ainsi que les informations des blocs et des sections de flux dans la zone PL.

- Les contraintes critiques imposent une limite supérieure à certains débits réels (les variables de décision). Les contraintes relationnelles expriment un lien entre les débits réels de différentes sections de flux et doivent être prises en considération durant le calcul. Par exemple, si le débit réel X est inférieur ou égal au débit réel Y, la fonction d'objectif doit se conformer à cette information.
- 8) Le Solver effectue un calcul.
- Le Solver calcule des débits réels optimisés pour la zone PL.
 - Le résultat est un calcul PL intermédiaire (en cas de préférence) ou le calcul PL final (si pas de préférence).
- 9) Les étapes 6-8 sont répétées, si nécessaire.
- Si les blocs dans la zone PL ont une préférence, le bloc Executive doit déterminer des fonctions d'objectif intermédiaires et le Solver effectuer des calculs intermédiaires. Le nombre de recalculs est égal au nombre de blocs ayant une préférence explicite, plus 1.
 - A chaque recalcul, la liste des contraintes critiques et relationnelles est modifiée pour inclure les contraintes causées par les préférences actuellement prises en compte.
- 10) Les sections de flux sont informées.
- Le bloc Executive des messages "Executive Flux" au début des sections de flux pour mettre à jour les nouveaux débits réels.
 - Les blocs recevant l'information mettent à jour leur dialogue et leurs connecteurs continus. Ils publient de nouveaux événements au besoin.
- ☞ Certains débits réels changeront, d'autre pas.

Types d'informations fournies au bloc Executive

Lorsque les débits réels doivent être recalculés, les blocs flux donnent au bloc Executive des informations sur les contraintes critiques et relationnelles et le biais. Voyez le tableau ci-dessous.

Règles de flux

Les règles de flux sont formées de contraintes critiques et relationnelles.

- Si une section de flux a une contrainte critique, elle impose une limite supérieure à certains débits réels dans la section. Comme les débits réels sont des variables de décision dans le calcul PL de la fonction d'objectif, les contraintes critiques imposent une limite supérieure à certaines variables de l'équation.
- Les contraintes relationnelles décrivent les dépendances entre différentes sections de flux. Pour certains blocs, les contraintes relationnelles peuvent varier en fonction de l'état de sensibilité du bloc, tandis que dans d'autres elles sont actives en permanence.
 - Un exemple de contrainte relationnelle contextuelle se trouve dans le bloc Tank. Les cuves sont toujours aux limites de deux sections de flux. Tant qu'un Tank est vide, sa contrainte relationnelle se définit par "débit réel entrant est inférieur ou égal à débit réel sortant". Lorsque le Tank devient "non vide", la contrainte relationnelle ne s'applique pas.
 - Un exemple de contrainte relationnelle permanente se trouve dans un bloc Change Units où un facteur de conversion définit la relation entre le débit réel entrant et débit réel sortant. Si le facteur varie dans le temps, la contrainte relationnelle peut aussi varier. Mais la relation de dépendance est toujours active dès qu'un calcul implique ces deux sections de flux.

Information de préférence

Lorsqu'est créée la zone PL, le bloc Executive classe dans une liste tous les blocs Bias ainsi que les blocs Merge ou Diverge avec un ordre de préférence. L'Executive considère que la préférence en haut de sa liste doit faire partie de sa fonction d'objectif et demande au Solver d'effectuer un calcul PL intermédiaire utilisant cette fonction et les contraintes critiques et relationnelles courantes. Puis le bloc Executive prend en compte le rang suivant, et ainsi de suite.

Dans la mesure où la préférence affecte les contraintes critiques et relationnelles, chaque rang dans les préférences implique qu'une nouvelle fonction d'objectif et qu'un nouvel ensemble de contraintes critiques et relationnelles doivent être ajoutés aux précédents. Les résultats d'un calcul PL sont utilisés en entrée pour le suivant. On obtient un grand nombre de calculs PL intermédiaires – un pour chaque biais dans la liste – jusqu'au résultat final.

La plupart des blocs avec préférence indiquent :

- Des règles de flux (contraintes critiques et relationnelles) qui s'appliquent à tous les calculs PL. (Cette information n'est pas fournie par un bloc Bias.)
- Un ensemble de coefficients que le bloc Executive utilisera pour construire la fonction d'objectif correspondant à un rang. En fonction des informations de préférence reçues depuis les blocs de la zone PL, la fonction d'objectif intermédiaire peut inclure des coefficients autres que 1. Un coefficient de 0, par exemple, indique que tel débit réel n'a pas besoin d'être maximisé.
- Les règles de flux pour ce rang qui utilisent les résultats du calcul intermédiaire.

Il existe certains cas particuliers :

- Si un bloc Bias a un rang qui est vide ou inférieur ou égal à 0, le bloc n'exprime aucune préférence.
- Si plusieurs blocs Bias ont le même rang, les débits réels de ces blocs sont imprévisibles.
- Les blocs Merge et Diverge avec une préférence sont toujours moins bien classés que les blocs Bias.

Tableau résumé des contraintes et informations de biais

Le tableau résume les informations que les blocs Rate fournissent au bloc Executive pour le calcul des débits réels :

Bloc	Mode	Contrainte critique	Relationnelle: Permanente	Relationnelle: contextuelle	Rang préf.
Bias		Non	Oui	Non	Oui
Catch Flow		Non	Non	Non	Non
Change Units		Non	Oui	Non	Non
Convey Flow		Oui	Non	Oui	Non
Diverge	Neutral, Proportional, Select, Unbatch	Cela dépend	Oui	Non	Non
Diverge	Distributional, Priority, Sensing	Cela dépend	Oui	Non	Oui
Interchange		Oui	Non	Oui	Non
Merge	Batch, Neutral, Proportional, Select	Cela dépend	Oui	Non	Non
Merge	Distributional, Demande, Sensing, Priority	Cela dépend	Oui	Non	Oui
Sensor		Non	Non	Non	Non
Tank		Oui	Non	Oui	Non
Throw Flow		Non	Non	Non	Non
Valve		Oui	Non	Non	Non

☞ Les blocs Diverge et Merges peuvent impliquer une contrainte critique en fonction du paramètre de branche et du mode.

Le calcul de contrainte relationnelle

Le tableau suivant décrit comment la contrainte relationnelle est calculée pour les blocs ayant des contraintes relationnelles permanentes ou contextuelles.

Dans ce tableau :

- X_{in} est le débit réel entrant d'un bloc et X_{out} est le débit réel sortant.
- Index i décrit une des branches en entrée (Merge) ou en sortie (Diverge).
- Le nombre de branches est n .

Bloc	Calcul
Change Units	La limite permanente est $X_{in} = \text{facteur} * X_{out}$

Convey Flow	Si le bloc est en situation d'accumulation (débit sortant bloqué totalement ou partiellement) et n'a plus de capacité d'accumulation, une limite contextuelle s'applique : $X_{in} \leq X_{out}$
Diverge	Select - Une contrainte relationnelle permanente s'applique entre le débit réel entrant et le débit réel sortant choisi : $X_{in} = X_{out}$ choisi Proportional – Un ensemble de contraintes relationnelles permanentes s'applique pour garantir les proportions : $X_{out_i} = factor_i * X_{in}$ ($i: 0 \Rightarrow n-1$) Batch/Unbatch - . Contraintes relationnelles permanentes : $X_{out_i} = X_{in}$ ($i: 0 \Rightarrow n-1$) Neutral, Priority, Distributional et Sensing. Contrainte permanente relationnelle $X_{in} = X_{out_1} + \dots + X_{out_n}$ (Les autres calculs sortent du cadre de ce document.)
Interchange	Tant que la cuve est pleine, $X_{in} \leq X_{out}$. Tant que la cuve est vide, $X_{out} \leq X_{in}$ (Contrainte relationnelle contextuelle)
Merge	Comme Diverge avec X_{out} et X_{in} inversés
Tank	Voir Interchange

Calcul PL

S'il n'y a pas de bloc avec ordre de biais dans la zone PL, le bloc Executive maximise la somme de tous les débits réels dans la zone PL – dans ce cas, un seul calcul PL est nécessaire. Si les blocs dans la zone PL ont un biais, le bloc Executive détermine plusieurs fonctions d'objectif et le Solver lance plusieurs calculs intermédiaires. Le nombre de recalculs est égal au nombre de blocs ayant un biais, plus 1.

Les calculs intermédiaires sont faits avec une fonction d'objectif qui dépend du type de bloc évalué :

- Bloc Bias. La fonction à maximiser est la somme des débits réels associés aux blocs Bias ayant le même rang. Lors du calcul, la fonction est utilisée comme une nouvelle règle intervenant dans les calculs intermédiaires. Le calcul est : somme des débits réels à l'intérieur du rang \geq résultat de la fonction maximisée.
- Merge/Diverge en mode Priority. La fonction à maximiser contient les débits réels des débits entrant et/ou sortant des branches du bloc. Plus basse est la priorité de la branche, plus haut sera le coefficient associé à son débit réel. Le calcul est : somme $p * X_p$ (où $p: 1 \Rightarrow n$ avec 1 étant la meilleure priorité et n la moins bonne). Lorsque le calcul est terminé, la fonction d'objectif est utilisée avec une nouvelle règle pour le calcul intermédiaire suivant (somme $p * X_p \geq$ résultat de la fonction maximisée).
- Merge/Diverge en mode Batch/Unbatch, Distributional, Neutral, Proportional, Select ou Sensing. Ces méthodes de calcul sortent du cadre de ce document.

Lorsque tous les calculs PL intermédiaires sont terminés, le dernier calcul PL maximise la somme de tous les débits réels dans la zone PL.

Fourniture amont et demande aval

Le débit réel n'est pas le seul résultat que fournit un modèle Rate. La fourniture amont et la demande aval potentielles sont également utilisées dans certaines situations – ces débits déterminent les proportions pour chaque branche pour les blocs Merge et Diverge en mode Sensing, et le bloc Sensor peut être utilisé pour indiquer les débits potentiels avant des décisions du modèle.

- ☞ Il s'agit d'un concept avancé entraînant un fort potentiel d'erreur, comme expliqué ci-dessous. Il requiert de sérieuses vérifications et validations du modèle ainsi qu'une bonne connaissance de la technologie PL ExtendSim.

Définition

Le débit délivrable par l'amont est le débit théorique par lequel la source amont peut fournir du flux au début d'une section de flux s'il n'y a aucune limitation aval sur la progression du flux. Par exemple, pour un Tank non plein au début d'une section de flux, le débit de fourniture amont serait égal au débit réel entrant du Tank. Une cuve non pleine ne limite pas le débit qu'elle reçoit en entrée. Dans ce cas, le débit réel entrant est aussi le débit délivrable par l'amont.

Le débit absorbable par l'aval est le débit théorique auquel une section en aval du modèle peut recevoir du flux depuis la fin d'une section de flux s'il y avait une fourniture amont sans limite (source amont infinie). Par exemple, pour une cuve non pleine à la fin d'une section de flux, le débit absorbable par l'aval serait égal au débit réel sortant du Tank. Une Cuve non pleine ne limite pas le débit qu'elle fournit en sortie. Dans ce cas, le débit réel sortant est aussi le débit absorbable par l'aval.

La fourniture amont et la demande aval peuvent potentiellement être infinies. Par exemple, la fourniture amont juste après une cuve (si elle ne déclare aucune contrainte sur son débit sortant) est infinie.

☞ Lorsque le mouvement du flux va de gauche à droite, l'information pour calculer le débit de fourniture amont se propage de gauche à droite, tandis que l'information concernant la demande aval se propage de droite à gauche.

Calculs de la fourniture/demande

Pour que le bloc Executive puisse calculer la fourniture amont et la demande aval potentielles, au moins l'un des blocs suivants doit faire partie de la zone PL :

- Bloc Sensor. Le seul objet du bloc Sensor est d'afficher les débits potentiels là où il est placé. Le Sensor ne fait que fournir une information; il n' a pas d'impact sur le calcul des débits réels.
- Bloc Diverge en mode Sensing. Les proportions pour les branches sortantes sont calculées en fonction de la demande aval potentielle. Par exemple, la demande aval placée sur une branche sortante devient la proportion pour cette branche.
- Bloc Merge en mode Sensing. Le bloc utilise le débit de fourniture amont pour définir les proportions de chaque branche en entrée.

Précautions à prendre avec les débits potentiels

Nous conseillons une grande prudence dans l'emploi des blocs Merge/ Diverges en mode Sensing, ou en s'appuyant sur le débit potentiel indiqué par un bloc Sensor, parce que :

- Obtenir des informations à propos de ces débits ralentit la simulation. Le bloc Executive doit faire davantage de calculs PL pour extraire les informations sur les débits potentiels.
- Dans certains cas les débits potentiels indiqués par le bloc Sensor ne sont pas exacts. Suivant la manière dont le flux est mélangé ou divisé, la fourniture amont et la demande aval potentielles peuvent être cumulées alors qu'elles ne devraient pas.
- Les blocs Merge/ Diverge en mode Distributional, Neutral et Priority ne sont pas toujours compatibles avec les blocs Merge/ Diverge en mode Sensing. Par conséquent, les modèles avec des blocs qui mélangent les modes Sensing et modes Distributional, Neutral ou Priority peuvent causer des erreurs.
- Un bloc Merge ou Diverge en mode Sensing a un ordre de biais. Selon la place du bloc dans la liste de l'ordre de biais du bloc Executive :
 - Les débits réels pourraient être différents.
 - Le bloc pourrait ne pas respecter les proportions spécifiées en mode Sensing.

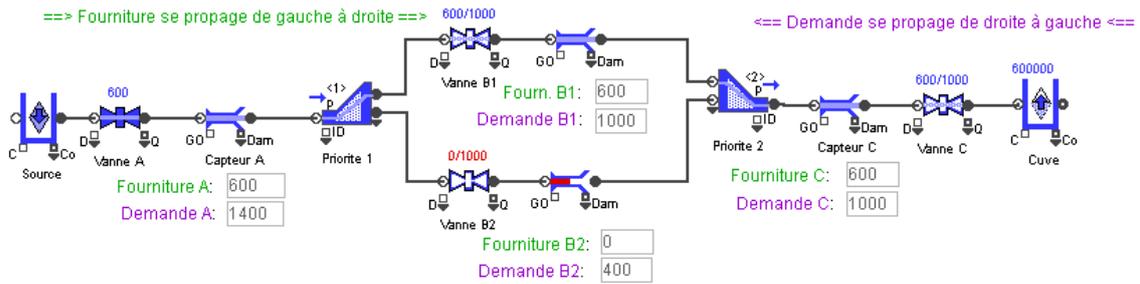
Par exemple, les débits réels sont différents dans "Combine Sensing Priority" et dans "Combine Priority Sensing" et aucun de ces modèles ne suit la règle Sensing.

Le bloc Sensor et les débits potentiels

Dans certaines configurations de modèle et suivant le mode choisi dans un bloc Merge ou Diverge, le bloc Sensor peut indiquer des informations erronées.

Dans le modèle *Supply & Demand Warning*, les blocs Sensor transmettent correctement la fourniture amont. Mais le Sensor A indique une demande aval incorrecte de 1400 litres/minute. La demande aval réelle est de 1000 litres/minute, indiquée par le Sensor C. On le détermine en vérifiant les résultats de

la demande, en la comparant à la valeur attendue, de gauche à droite.



Evaluation du modèle de droite à gauche (pour le débit de demande aval) :

- Le Sensor C indique une demande (Demande C) de 1000 litres/minute parce que la Vanne C limite le flux à 1000 litres/minute. Le réservoir à la fin de la ligne ne limite pas du tout le débit entrant.
 - La demande B1 est de 1000 litres/minute parce que le bloc Priority 2 a sa meilleure priorité sur la branche B1. (Vérification : si un Tank avec du flux était présent sur la branche B1, les blocs Priority 2 et Vanne C pourraient accepter 1000 litres/minute provenant de la branche B1.)
 - La demande B2 est de 400 litres/minute parce que le débit réel sur la branche B1 est de 600 et la Demande C de 1000 : $\text{Demande B2} = \text{Demande C} - \text{débit réel B1}$. (Vérification : si un Tank avec du flux était présent sur la branche B2, elle pourrait fournir 400 litres/minute parce que le total de ce que le bloc Priority 2 peut accepter est 1000 et que le bloc prend déjà 600 par la branche prioritaire.)
 - Le Sensor A indique une demande de 1400 litres/minute parce que Demande B1 est de 1000 et Demande B2 de 400. Mais c'est incorrect. (Vérification : si un Tank avec du flux était placé juste avant Priority 1, le débit sortant réel serait de 1000 litres/minute et non de 1400 litres/minutes. L'association de règles locales "correctes" ne garantit pas un résultat global "correct".)
- ☞ Deux solutions sont 1) que le bloc Priority 2 soit en mode Neutral et 2) placer une Vanne avec un débit maximum de 1000 entre le Sensor A et le Priority 1.

Mélanger les modes Merge/ Diverge

Il faut éviter les situations où des blocs Merge ou Diverge en mode Sensing sont mélangés avec des blocs Diverge ou Merges en mode Distributional, Neutral ou Priority dans une même zone PL.

Dans ces situations mixtes, la zone PL sera recalculée un grand nombre de fois pour établir les débits réels et les débits de fourniture amont et demande aval. Tous les blocs dans la zone PL doivent fournir un ensemble de contraintes au bloc Executive pour calculer les débits réels et un second ensemble de contraintes pour que le bloc Executive pour calculer les débits de fourniture amont et demande aval. Les calculs prennent du temps et peuvent donner des résultats faux.

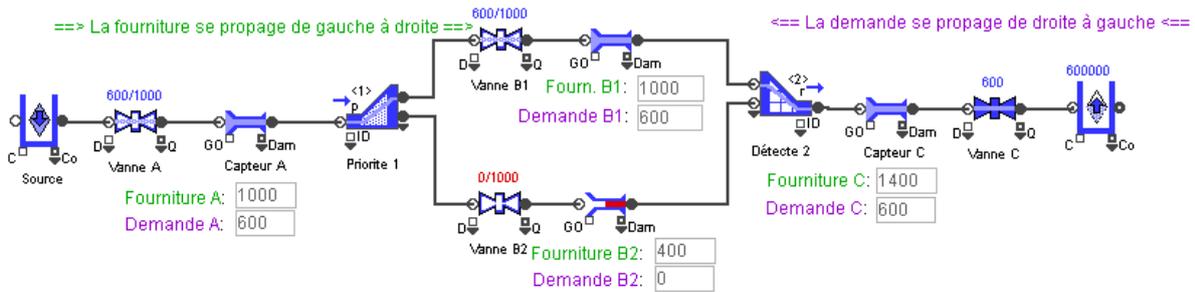
Effets inattendus des préférences

Nous l'avons mentionné, les modes Distributional, Priority et Sensing requièrent qu'on définisse des préférences. Cela affecte leurs débits réels. Mais le seul mode dont les proportions sont influencées par le rang de préférence est le mode Sensing.

Lorsque les débits réels dans une zone PL doivent être recalculés, le bloc Executive fait la liste des blocs dans l'ordre de préférence décroissant. Selon le classement d'un bloc Merge ou Diverge dans la liste, les débits réels seront différents. Si le bloc est en mode Sensing, les proportions seront aussi influencées. Deux modèles illustrent ces conditions.

Modèles Combine Sensing Priority

Dans ce modèle, un bloc Diverge (Priority 1) est en mode Priority et un bloc Merge (Sensing 2) en mode Sensing. L'onglet Model Settings de ces blocs indique *Chaque bloc définit son rang*. Le tableau montre que Priority 1 a un rang meilleur que Sensing 2.



La fourniture sur la branche B1 est de 1,000 tandis que la fourniture sur la branche B2 est de 400. Pourtant, le bloc Sensing 2 ne distribue pas le flux dans les proportions attendues.

- Comme le bloc Priority 1 a le meilleur ordre de biais, le bloc Executive donne priorité à ce bloc pour la distribution du flux entre les branches B1 et B2.
- Au résultat du premier calcul PL intermédiaire, la branche B1 reçoit un débit réel de 600 litres/minute et la branche B2 reçoit 0 litres/minute.
- Cette décision prise pour les branches B1 et B2, le bloc Sensing 2 avec le faible ordre de biais ne contrôle pas la proportion de flux qu'il obtient par les branches B1 et B2. Par exemple, si la fourniture depuis B1 est de 1000 litres/minute et la fourniture depuis B2 est de 400 litres/minute, la proportion entre les débits réels en entrée du bloc Sensing 2 n'est pas de 71% de B1 (1000/1400) et 29% de B2 (400/1400) mais plutôt 100% de B1 et 0% de B2.

Modèle Combine Priority Sensing

Comme dans le modèle précédent, le bloc Diverge (Priority 2) dans ce modèle est en mode Priority et le bloc Merge (Sensing 1) en mode Sensing, et l'onglet Modèle de ces blocs indique *Chaque bloc définit son rang*. Mais le tableau montre que Priority 2 a un rang moins bon que Sensing 1.

- Le bloc Sensing 1 a le meilleur rang, donc le bloc Executive donne préférence à ce bloc pour la distribution du flux entre les branches B1 et B2.
- Dans le calcul PL intermédiaire, le bloc Priority 2 reçoit une fourniture de 1000 litres/minute et ne sait comment distribuer le flux. Avec ces informations limitées, le débit potentiel amont est de 1000 litres/minute pour Fourniture B1 et de 1000 litres/minute pour Fourniture B2. Par conséquent, Sensing 1 décide d'obtenir 50% du flux par la branche entrante B1 et 50% par la branche B2.
- Cette décision étant prise pour les branches B1 et B2, le bloc Priority 2 avec le faible ordre de biais ne contrôle pas la distribution de flux sur les branches B1 et B2.

Les messages dans les modèles Rate

L'architecture ExtendSim s'appuie sur l'échange de message entre le moteur ExtendSim et des blocs, et entre les blocs eux-mêmes. Les modèles Rate utilisent les mêmes messages que les modèles continus ou discrets.

Messages de blocs

Lorsqu'un bloc est à l'origine d'un recalcul des débits réels, une succession de messages et de calculs commence. Il est important de limiter l'envoi de messages redondants, pour limiter les temps de calculs, et surtout pour éviter les résultats erronés ayant pris en compte des paramètres qui ont changé depuis le recalcul précédent.

Messages d'événements

Ces messages établissent une communication entre le bloc Executive et les blocs Rate. Le temps avance par sauts discrets, et à chaque événement, le bloc Executive envoie des messages d'événement aux blocs qui se sont associés à cet événement. Les événements sont futurs ou courants.

- Un message d'événement futur intervient lorsque l'horloge de la simulation atteint l'heure publiée par un bloc pour un de ses événements. Par exemple, lorsque le niveau d'un Tank augmente, il publie un événement pour le bloc Executive pour signaler quand il sera plein. Quand l'horloge de la simulation atteint ce moment, le bloc Executive envoie un message d'événement au Tank, pour lui signaler cet état plein.
- Un message d'événement courant intervient lorsqu'un bloc veut être activé avant que l'horloge de la simulation n'avance, mais après la fin d'autres messages. Par exemple, pour qu'une Vanne ne

recalcule pas son débit réel dès qu'un débit de contrainte change, elle publiera un événement courant pour le bloc Executive pour lui signaler qu'elle doit encore être recalculée. Les autres blocs du modèle ont le temps de se mettre à jour avant que ce calcul ait lieu.

- ☞ Dans les modèles Rate, les blocs de la bib. Value n'envoient généralement pas de message au bloc Executive ni n'en reçoivent de sa part. C'est important pour comprendre le comportement des blocs continus dans les modèles Rate.

Messages de connecteurs continus

Les blocs dans un modèle flux envoient des messages de connecteur continu soit parce qu'une nouvelle valeur est requise par un connecteur d'entrée, soit parce que la valeur d'un connecteur de sortie a changé. Par exemple, si le connecteur R d'un bloc Vanne reçoit une nouvelle valeur D, cela change la contrainte sur le flux et provoque un recalcul de paramètres dans le modèle.

Ces messages fonctionnent dans les modèles flux comme dans les modèles purement discrets.

Messages de connecteurs discrets

Les modèles flux comportent souvent des portions utilisant des blocs discrets de la bib. Item. Le bloc Interchange de la bibliothèque Rate permet l'interaction avec les blocs discrets dans un modèle flux.

Les messages de connecteurs discrets (essentiellement wants, needs et rejects) établissent une conversation permettant la progression des entités d'un bloc à l'autre. Les messages de connecteurs discrets fonctionnent dans les modèles flux comme dans les modèles purement discrets.

Messages de connecteurs flux

Les connecteurs flux indiquent la valeur des débits réels entrant ou sortant. Les messages de connecteurs flux provoquent la mise à jour de tous les blocs liés au moment où le calcul PL détermine que les débits réels ont changé.

Messages de connecteurs de débit

Dès qu'un bloc de la bibliothèque Rate reçoit un message, s'il détermine un possible changement des débits réel entrant ou sortant, cela lance la propagation de ces messages. Cette propagation de messages définit la zone PL – la zone du modèle affectée par le changement dans le bloc d'origine.

Messages flux du bloc Executive

Lorsque commence un calcul des débits réels pour une zone PL, le bloc Executive envoie des messages aux blocs pour propager les résultats. Ces messages mettent à jour tous les blocs de la zone affectée par le nouveau débit réel.

Menus

Cette rubrique décrit les commandes et les menus utilisés dans ExtendSim. Comme vous l'avez constaté, peu de commandes sont nécessaires pour créer ou exécuter des modèles. Vous trouverez ici toutes les commandes qui apparaissent dans les menus et dans quelles circonstances les utiliser.

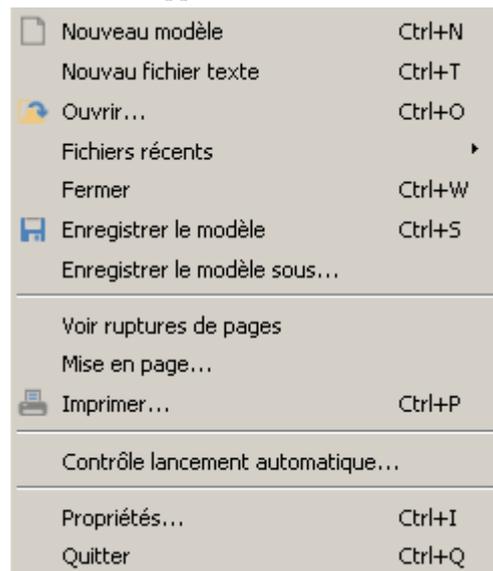
Certains des menus sont des menus hiérarchiques, c'est à dire comportant eux-mêmes des sous-menus. On repère un sous-menu par un triangle (◆) face à l'élément de base. Pour utiliser un menu hiérarchique, faire glisser la souris jusqu'à la commande désirée. Lorsque cette commande est noircie, un second menu apparaît à la droite du premier. En gardant toujours le bouton enfoncé, faire glisser le curseur sur la droite, jusqu'à l'élément souhaité du sous-menu, et relâcher la souris.

Au lieu d'utiliser la souris, on peut appuyer sur la touche CTRL et sur la lettre de l'option.

La fonction `ExecuteMenuCommand(numéro)` exécute la commande de menu indiquée **par le numéro** spécifié. Cela équivaut à sélectionner la commande dans les menus par la souris. Les programmeurs trouveront le numéro de la commande que l'on peut appeler par cette fonction entre parenthèse à côté de chaque commande.

Menu Fichier

Le menu Fichier permet d'ouvrir, d'enregistrer et d'imprimer des fichiers de modèles. Notez que les bibliothèques s'ouvrent à partir du menu Bibliothèque décrit plus loin. La plupart des options de ce menu sont classiques dans de nombreuses applications.



menu Fichier

Nouveau modèle (2)

Ouvre une fenêtre Sans titre pour un nouveau modèle.

Nouveau fichier texte (1601)

Ouvre une fenêtre Sans titre pour un nouveau fichier texte.

Ouvrir (3)

Ouvre un modèle existant. En ouvrant le modèle, ExtendSim ouvre aussi les bibliothèques qu'il utilise. Vous n'avez donc pas à ouvrir manuellement les bibliothèques si vous voulez seulement lancer la simulation. Bien sûr, si la bibliothèque est ouverte, ExtendSim ne l'ouvre pas à nouveau.

Parfois, si depuis la dernière utilisation vous avez déplacé le fichier du modèle ou une bibliothèque, ExtendSim peut ne pas retrouver une bibliothèque. Vous verrez alors le message "Recherche de la bibliothèque xxx...", tandis qu'ExtendSim cherche la bibliothèque dans le dossier/répertoire des bibliothèques ainsi que dans les dossiers/répertoires imbriqués, voire même dans tout le disque.

Fichiers récents (1555-1559)

Affiche le nom des dix fichiers (modèles ou fichiers texte) que vous avez ouvert le plus récemment. Cela permet de revenir rapidement sur l'un de ces fichiers.

Fermer (4)

Ferme le fichier du modèle actif. Vous pouvez aussi fermer le fichier en cliquant dans la case qui se trouve dans le coin supérieur droit de la fenêtre.

Enregistrer (5) et Enregistrer sous (6)

Enregistre le modèle sur disque. Enregistrer enregistre le modèle sous son nom courant, et Enregistrer sous permet de lui donner un autre nom.

Pour éviter d'endommager des fichiers, ExtendSim effectue une copie de la version déjà enregistrée du modèle et lui donne l'extension .BAK avant d'enregistrer le modèle. Si un incident survient lors de l'enregistrement, le fichier .BAK permettra de récupérer la version antérieure du modèle. Si l'enregistrement se passe normalement, vous pouvez choisir qu'ExtendSim supprime automatiquement le fichier .BAK (voir l'option Préférences du menu Edition).

Voir ruptures de page

Si cette option est choisie, ExtendSim montre les limites de page sur le modèle. Ces limites indiquent où auront lieu les ruptures de page en cas d'impression. Notez que la taille de la page dépend des paramètres de la commande *Mise en page*. Lorsque Voir ruptures de page est sélectionné, l'option apparaît cochée sur le menu. Pour que les ruptures de page ne soient plus visibles, sélectionnez l'option à nouveau. Les journaux de bord et les dialogues affichent toujours leur rupture de page.

Lorsque cette option est sélectionnée, le numéro de page apparaît dans le coin supérieur gauche de chaque page du modèle, pour vous permettre par exemple d'imprimer des pages spécifiques d'un modèle.

Mise en page

Sert à donner les paramètres d'impression pour votre imprimante. Choisissez cette commande après un changement d'imprimante ou quand vous souhaitez modifier des paramètres d'impression (orientation du papier, réduction ou agrandissement, etc.).

Imprimer (9)

Imprime le modèle, le journal de bord ou le dialogue ouvert. Si vous avez sélectionné Numéros de blocs ou Ordre de la simulation dans le menu Modèle avant d'imprimer, les blocs auront ces informations en plus.

L'option Imprimer permet aussi d'imprimer des fenêtres individuelles, telles que la fenêtre de structure ou de dialogue, les graphiques, etc. Lorsqu'une de ces fenêtres est la fenêtre active, Imprimer imprime le contenu de cette fenêtre. Les fenêtres de structure et de dialogue ont leur propre dialogue pour choisir quelle portion imprimer de la structure d'un bloc.

Contrôle lancement automatique (1410)

Si vous avez plusieurs versions d'ExtendSim installées (v9 et v10 par exemple) et que vous voulez que le lancement automatique par double-clic sur un modèle ou une bibliothèque fasse référence à une version en particulier, cette commande modifie le registre pour lancer par défaut la version courante.

Propriétés (2001)

Affiche des informations diverses sur les objets du modèle ou les blocs sélectionnés. En voici un exemple :

The screenshot shows a dialog box titled "[115] Rect" with the following fields and options:

- Object Name: Rect
- ID objet: 115
- ID Anim: -1
- X: 77, Largeur: 46
- Y: 21, Hauteur: 12
- Lignes: 0, Ordre simu: 0
- Colonnes: 0
- Parent: -1, None (dropdown)
- zOrder: 0
- Options:
 - Visible
 - Affichage seulement
 - Sur tous les onglets
 - Lissage
- ID groupe BR: 0
- Onglet n°: -1

*dialogue Propriétés***Quitter (1)**

Quitte ExtendSim. Si des modifications n'ont pas été enregistrées, on vous demande si vous souhaitez les enregistrer d'abord.

Menu Edition

Ce menu contient les commandes standard Couper, Copier et Coller, ainsi que les commandes relatives à l'analyse de sensibilité et aux préférences.

*menu Edition***Annuler**

Annule la plus récente action (option, édition, déplacement, etc.). Plusieurs actions sont mémorisées.

Refaire

Refait la plus récente action annulée (option, édition, déplacement, etc.). Plusieurs actions sont mémorisées.

Couper

Supprime l'élément sélectionné (un bloc, une portion de texte ou les données numériques d'une table de données) et le place dans le Presse-Papiers.

Copier

Copie l'élément sélectionné dans le Presse-Papiers. Cette option est notamment utile pour dupliquer des parties d'un modèle ou pour exporter vers d'autres applications. Vous pouvez copier un bloc, une portion de texte, des blocs et du texte, des objets graphiques ou des données numériques d'une table de données, etc. Vous pouvez aussi copier des sections d'un journal de bord ou d'un dialogue en tant qu'image. Cela dépend parfois du type de curseur sélectionné au moment de l'action.

Copier avec en-têtes...

Copie les en-têtes de tableaux en plus des données pour les structures tabulaires sélectionnées.

Coller

Copie le contenu du Presse-Papiers dans le modèle. Si le Presse-Papiers contient du texte, un bloc ou un élément graphique, l'élément copié est placé au point d'insertion. S'il n'y a pas de point d'insertion, l'élément est placé dans le coin supérieur gauche du modèle.

Dupliquer

Fait une copie des éléments sélectionnés et la place près de l'original. Souvent plus rapide que Copier+Coller.

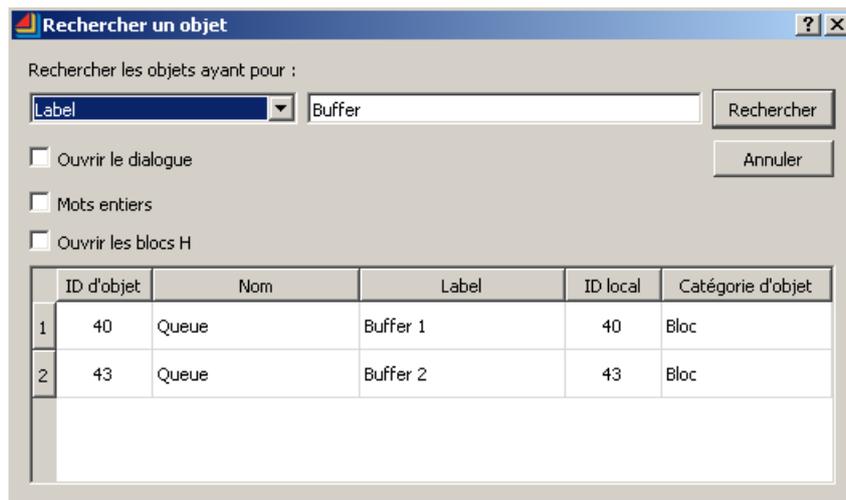
Effacer

Efface l'élément sélectionné.

Rechercher...

Cette commande affiche un dialogue différent en fonction de l'emplacement d'appel :

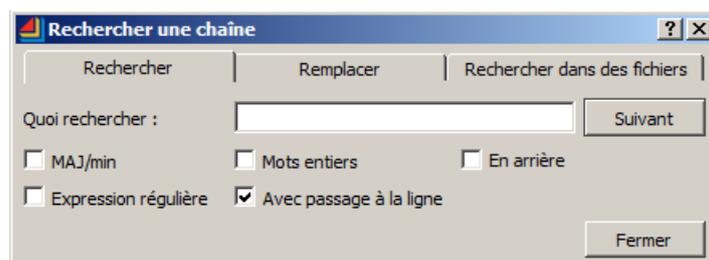
Si le curseur figure **dans la fenêtre du modèle**, recherche un objet ou un bloc dans le modèle grâce à des identificateurs. Cette option est utile pour repérer des blocs dans de très gros modèles. Le dialogue est alors celui-ci :



dialogue Rechercher un objet

L'ID d'objet est l'identifiant unique pour les blocs, texte, graphiques etc. La catégorie indique le type d'objet (bloc, texte, dessin etc.). Si un dialogue de bloc est la fenêtre au premier plan, la recherche s'effectue parmi les éléments du dialogue.

Si le curseur figure **dans du texte** (notamment un script de bloc), cherche une chaîne dans ce texte, en partant de la sélection courante et en allant jusqu'à la fin du texte. Si la chaîne spécifiée est trouvée, elle est sélectionnée. Si elle n'est pas trouvée, le point d'insertion n'est pas modifié. Voici le dialogue Rechercher une chaîne



dialogue Rechercher une chaîne

Entrez le texte à rechercher dans la zone Rechercher. L'option **Mots entiers** indique à ExtendSim de ne rechercher que des mots entiers (par exemple "animal" ne serait pas trouvé par une recherche sur "mal"). **Expression régulière** signifie la possibilité de rechercher par des expressions syntaxique de type 'regex' des chaînes de caractères.

La commande Rechercher effectue la recherche dans la portion de la fenêtre de structure où se trouve le curseur. Assurez-vous que le point d'insertion est dans la portion où vous souhaitez faire la recherche.

Dans l'onglet **Remplacer** une zone supplémentaire est utilisée pour les opérations de recherche/remplacement.

Rechercher dans des fichiers permet d'étendre la recherche en parcourant des fichiers include, des fichiers source externalisés, ou d'autres fichiers.

☞ Le bloc Find and Replace de la bib Utilities permet d'effectuer des remplacement automatiques dans des dialogues de blocs.

Suivant

Répète la plus récente opération de recherche, avec la même chaîne de recherche.

Tout sélectionner

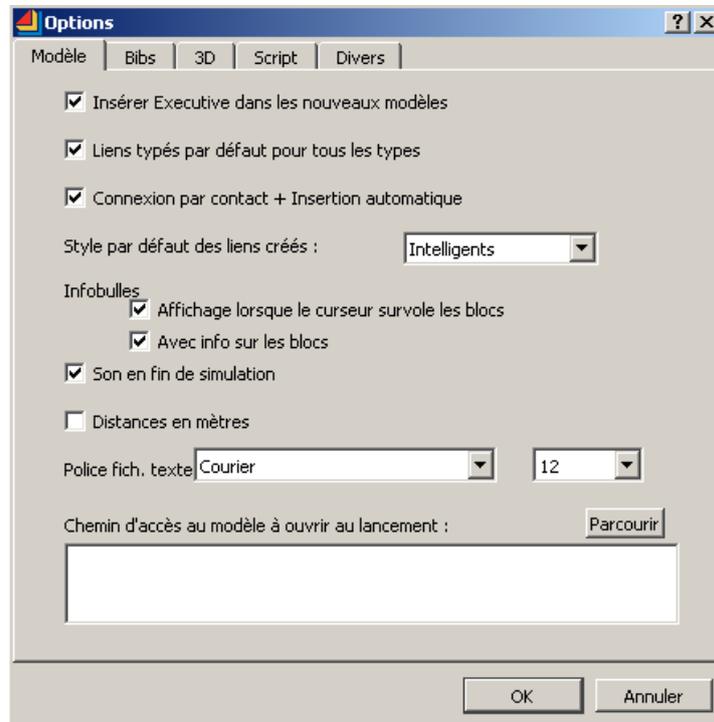
Sélectionne tous les éléments, par exemple tous les blocs d'un modèle ou tout le texte d'un champ. Les éléments sélectionnés sont fonction de l'outil de sélection actif.

Prendre sélection

Place le texte sélectionné dans la zone Rechercher. C'est utile pour trouver l'occurrence suivante d'un texte que vous avez sous les yeux dans l'aide ou dans le script.

Options

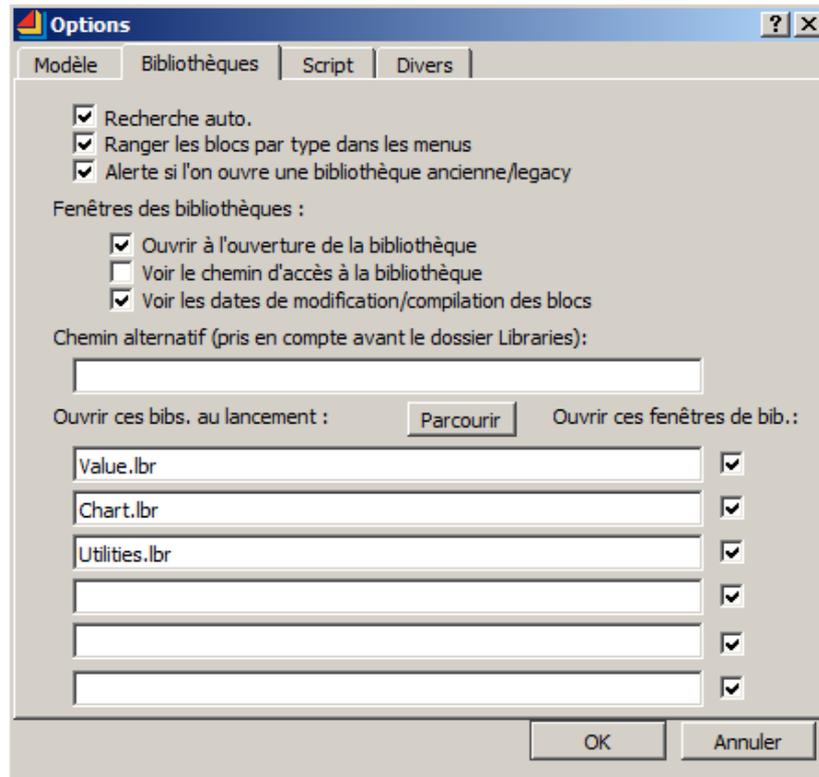
Permet de spécifier certains modes de fonctionnement d'ExtendSim. Contrairement à l'option Paramètres du menu Simulation, qui n'affecte qu'un modèle, cette option est valable pour tous les modèles. Le dialogue est divisé en cinq onglets : Modèle, Bibs, 3D, Script et Divers.



onglet Modèle du dialogue Options

Option	Description
Insérer Executive dans les nouveaux modèles	Si votre version d'ExtendSim installe la bibliothèque Item, il est considéré que vous voulez bâtir des modèles discrets (ou débits à événements discrets), qui requièrent le bloc Executive. Décochez cette case si

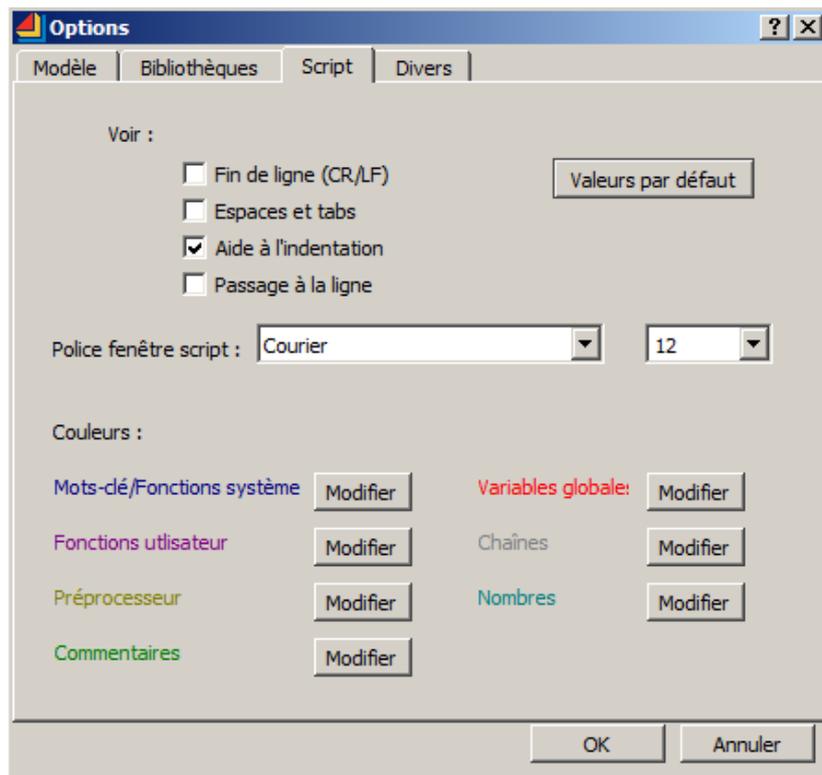
	vous faites en réalité des modèles continus.
Liens typés par défaut	Spécifie si ExtendSim utilise ou non les liens par défaut lorsqu'il crée de nouveaux liens (lignes doubles pour flux discret, simples pour flux continu).
Connexion par contact + Insertion automatique	<p>Utilisez <i>Connexion par contact</i> pour ajouter un bloc Discret ou Rate à un bloc du modèle. Cliquez sur le bloc souhaité dans la fenêtre de bibliothèque puis placez-le en faisant que son connecteur itemIn ou inFlow touche le connecteur de sortie correspondant d'un bloc déjà présent. Le nouveau bloc est ajouté et connecté au bloc existant.</p> <p>Avec <i>Insertion automatique</i> vous insérez un bloc Discret ou Rate entre deux blocs déjà reliés de la même bibliothèque. Cliquez sur le bloc souhaité dans la fenêtre de bibliothèque puis cliquez sur le lien entre les blocs, de sorte que les connecteurs itemIn ou inFlow chevauchent le lien. Le nouveau bloc est alors inséré et connecté entre les deux blocs existant.</p> <p>Ces options ne s'appliquent qu'aux connecteurs discrets ou flux.</p>
Style par défaut des liens créés	Spécifie par défaut un tracé à angle droit des liens (voir option Liens du menu Modèle).
Infobulles : Affichage lorsque le curseur survole les blocs	<p>Si coché ExtendSim affiche le nom, le numéro et la bib. des blocs que survole le curseur dans la fenêtre de modèle. Si le curseur survole un connecteur, affiche le nom et la valeur d connecteur.</p> <p>Ne concerne pas les infobulles de la barre d'outils.</p>
Infobulles : Avec info sur les blocs	En plus des informations ci-dessus, le bloc affichera aussi sa description.
Son en fin de simulation	ExtendSim émet un bip à la fin de chaque simulation.
Distances en mètres	Spécifie si ExtendSim utilise des mètres ou des pieds pour les calculs de distance dans l'interface des blocs de transport.
Police fichiers texte	<p>Permet de spécifier la police des caractères des fenêtres de fichiers texte.</p> <p>Cette option ne définit PAS la police et taille par défaut de la barre d'outils de l'application, ni celle des zones de texte ou du script d'un bloc (voir l'onglet Script pour ce dernier point).</p>
Chemin d'accès au modèle à ouvrir au lancement	Si un chemin d'accès est spécifié, ouvre le modèle en question. Si juste un nom de modèle est spécifié, il est recherché dans le répertoire utilisateur par défaut.



onglet Bibliothèques du dialogue Options

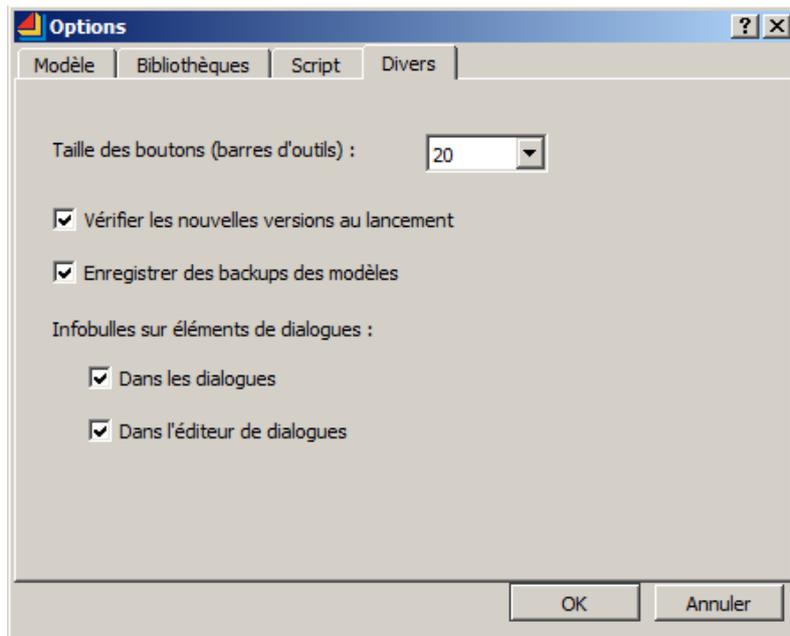
Option	Description
Recherche automatique	A l'ouverture d'un modèle, les bibliothèques utilisées sont recherchées automatiquement. Si non sélectionné, vous devez localiser chaque bibliothèque et l'ouvrir.
Ranger les blocs par type dans les menus	Les blocs des bibliothèques sont rangés en familles de blocs dans des sous-menus hiérarchiques. Sinon l'ordre est alphabétique.
Alerte à l'ouverture d'une bibliothèque ancienne/legacy	Cette option est sélectionnée par défaut et vous alerte en cas d'utilisation d'une ancienne bibliothèque dans un modèle récent. Vous pouvez désactiver l'alerte. NOTE: les bibliothèques « Legacy » sont incluses afin de pouvoir exécuter d'anciens modèles. Ne pas les utiliser pour créer de nouveaux modèles.
Fenêtres des bibliothèques : Ouvrir à l'ouverture de la bibliothèque	Lorsqu'une bibliothèque est ouverte, sa fenêtre spécifique s'ouvre aussi.
Fenêtres des bibliothèques : Voir le chemin d'accès à la bib.	Si sélectionné, la fenêtre de bibliothèque affiche le chemin d'accès à la bibliothèque.
Fenêtres des bibliothèques : Voir les dates	Si sélectionné, la date de modification et de compilation de chaque bloc figure dans la fenêtre de bibliothèque.
Chemin alternatif	Par défaut en ouvrant un modèle ExtendSim cherche le dossier contenant le modèle. S'il ne trouve pas il cherche dans le dossier Documents/ExtendSim/Libraries. Le chemin alternatif spécifie un emplacement où ExtendSim cherchera après le dossier du modèle, mais avant le dossier Libraries.
Ouvrir ces bibs. au lancement :	Entrez ici les noms des bibliothèques qui doivent être automatiquement ouvertes au lancement d'ExtendSim.

	<p>Vous pouvez taper le nom ou utiliser le bouton Parcourir pour localiser une bibliothèque. Si la bibliothèque de figure pas dans le répertoire par défaut des bibliothèques, l'emplacement de la bib. vous est demandé.</p> <p>NOTE : en général les bibliothèques principales d'ExtendSim sont proposées en chargement automatique.</p>
Ouvrir fenêtre bib.	Si coché, la bibliothèque spécifiée à la gauche aura aussi sa fenêtre automatiquement ouverte.
Parcourir	Permet de localiser et sélectionner une bibliothèque qui sera chargés automatiquement.



onglet Script du dialogue Options

Option	Description
Voir : Fin de ligne (CR/LF)	Montre les caractères de fin de ligne, qui sont normalement invisibles.
Voir : Espaces et tabs	Montre les caractères Espace et Tabulation, qui sont normalement invisibles (seulement montrés par du blanc).
Voir : Aide à l'indentation	Affiche des lignes verticales pour indiquer l'indentation du code là où vous avez utilisé les tabulations.
Voir : Passages à la ligne	Fait que l'éditeur de script effectue un passage à la ligne de sorte que toutes les lignes soient visibles dans la fenêtre de script.
Police fenêtre script :	Permet de spécifier la police et le corps des caractères de l'onglet Script de la fenêtre de structure.
Couleurs	Montre les couleurs par défaut pour signaler les mots-clés, commentaires etc. et permet de les modifier par le bouton Modifier.

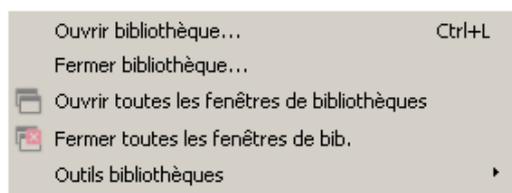


onglet Divers du dialogue Options

Option	Description
Taille des boutons (barres d'outils) :	Permet de changer la taille des boutons dans les barres d'outils.
Vérifier les nouvelles versions	Vérifie si une version plus récente d'ExtendSim n'est pas disponible (requiert un accès internet). Un message est affiché dans l'écran À propos...
Enregistrer des backups	Si cochée, lorsque vous enregistrez un modèle, la précédente version enregistrée est conservée avec l'extension .bak.
Infobulles sur éléments de dialogues : Dans les dialogues	Affiche le nom de la variable/élément de dialogue lorsque le curseur survole tel élément.
Infobulles sur éléments de dialogues : Dans l'éditeur de dialogues	Affiche le nom de la variable/élément de dialogue lorsque le curseur survole tel élément dans l'onglet Dialogue en mode Structure.

Menu Bibliothèque

ExtendSim ouvre automatiquement des bibliothèques associées lorsque vous ouvrez un modèle. En fonction des choix de l'onglet Bibliothèques du dialogue Options, certaines sont ouvertes dès le lancement d'ExtendSim. Pour ouvrir ou fermer une bibliothèque manuellement, utilisez le menu Bibliothèque.



menu Bibliothèque

Ouvrir

Ouvre un (ou plus) fichier bibliothèque. Son nom figure alors dans le menu Bib, dans l'ordre alphabétique. Vous ajoutez des blocs à vos modèles en les sélectionnant dans les bibliothèques ouvertes listées au bas du menu bibliothèques. Le premier choix de chaque liste est **Ouvrir fenêtre**, qui ouvre une fenêtre où figurent tous les blocs de la bibliothèque. Voyez plus haut l'emploi des bibliothèques.

Fermer

Ferme une bibliothèque ouverte. Cette commande liste toutes les bibliothèques qui ne sont pas en usage dans un modèle ouvert. Sélectionnez la bibliothèque à fermer dans la liste et cliquez sur Fermer. Les bibliothèques en usage n'apparaissent pas dans la liste et ne peuvent être fermées.

Ouvrir toutes les fenêtres de bibliothèques

Ouvre la fenêtre de bibliothèque pour toutes les bibliothèques ouvertes, et les ancre sur la droite. Notez les boutons au haut de zone d'ancrage, pour ouvrir et fermer les fenêtres de bibliothèques.

Fermer toutes les fenêtres de bibliothèques

Ferme toutes les fenêtres de bibliothèques actuellement ouvertes, sans fermer la bibliothèque. Notez les boutons au haut de zone d'ancrage, pour ouvrir et fermer les fenêtres de bibliothèques.

Outils

Affiche un menu hiérarchique relatif aux actions sur les bibliothèques. Sert surtout aux développeurs.

Nouvelle bibliothèque : Crée une nouvelle bibliothèque.

Compiler des bibliothèques : permet de choisir les bibliothèques à compiler parmi les bibliothèques ouvertes.

Compiler les blocs sélectionnés : permet de compiler plusieurs blocs sélectionnés. La fenêtre de bibliothèque doit être au premier plan et certains blocs sélectionnés pour activer la commande.

Ajouter code de débogage aux bibliothèques: permet de choisir les bibliothèques à compiler en incluant le code de débogage. Sélectionnez ensuite la commande *Compiler des bibliothèques*.

Effacer code de débogage des bibliothèques: permet de choisir les bibliothèques à compiler en retirant le code de débogage. Sélectionnez ensuite la commande *Compiler des bibliothèques*.

Externaliser le code pour les bibs.: permet de choisir les bibliothèques dont le code source va être copié dans un fichier texte indépendant, d'extension .cm (pour être éventuellement travaillé avec un autre éditeur de code).

Réintégrer le code externe aux bibs.: permet de choisir les bibliothèques le code source externe va être réintégré dans le code du bloc. Attention : réintégrer le code supprime les fichiers .cm externes Reportez-vous manuel *Technical Reference* pour plus d'information sur le code externalisé.

Protéger bibliothèque : après deux messages d'alerte, propose de protéger la bibliothèque, c'est-à-dire supprimer le script d'un bloc, de sorte qu'il ne soit plus accessible par l'utilisateur. N'utilisez cette commande qu'avec vos propres bibliothèques.

Définir la version : permet d'affecter un numéro de version (chaîne longue et chaîne courte) à une bibliothèque. Facilite la maintenance des bibliothèques.

Convertir bib. pour runtime: transforme une copie de la bibliothèque sélectionnée en format Runtime (pour exécution, et non modification). Cette opération supprime également le script des blocs. La bibliothèque ne peut alors être ouverte par les versions développeur d'ExtendSim, ce qui empêche que votre travail serve à d'autres pour construire leurs modèles.



Menu Modèle

Ce menu affecte la fenêtre du modèle et les objets qui s'y trouvent.

*Menu Modèle***Hiérarchiser la sélection**

Transforme les blocs sélectionnés en un bloc hiérarchique et les remplace par une seule icône dans la fenêtre.

Nouveau bloc hiérarchique

Crée un nouveau bloc hiérarchique. On vous demande le nom du bloc hiérarchique avant d'afficher la fenêtre de structure vide pour ce bloc.

Ouvrir structure de bloc H

Ouvre la fenêtre de structure du bloc hiérarchique sélectionné, permettant de modifier son icône, son texte d'aide et ses connecteurs.

Ouvrir le bloc H associé

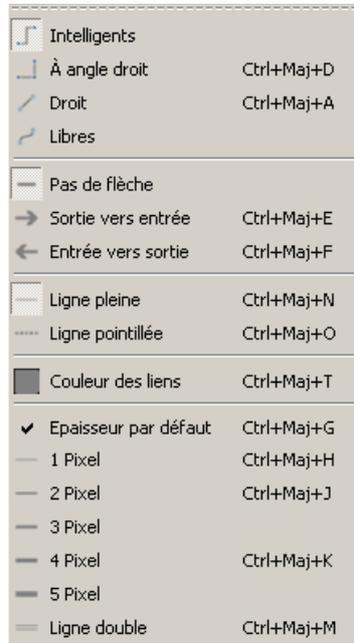
Trouve le bloc hiérarchique parent du bloc sélectionné. Si le bloc H parent est au niveau zéro du modèle, il sera mis en surbrillance. S'il est d'un niveau inférieur, la fenêtre du sous-modèle s'ouvrira.

Miniature sur l'icône H

Place une version miniature du sous-modèle dans l'icône du bloc hiérarchique sélectionné. L'icône est mise à jour avec les modifications du sous-modèle, et l'animation est visible durant la simulation

Style des liens

Détermine le format des liens sélectionnés. Les choix sont détaillés dans [Styles et manipulation des liens](#):

*choix de liens***Voir les liens nommés**

Affiche les liens réels entre des liens nommés. C'est utile pour voir ce qui se passe vraiment dans un modèle complexe qui comporte de nombreux liens nommés.

Masquer les liens

Cache les liens entre les blocs. Cette modification d'aspect est rarement utilisée, car le modèle devient incompréhensible, mais peut servir pour des présentations. S'applique à chaque niveau hiérarchique indépendamment des autres.

Masquer les connecteurs

Cache les connecteurs des blocs. Cette modification d'aspect est rarement utilisée, car le modèle devient incompréhensible, mais peut servir pour des présentations. S'applique à chaque niveau hiérarchique indépendamment des autres.

Voir labels de blocs

Montre les labels des blocs au-dessous de leur icône.

Voir l'ID des objets

Affiche les numéros d'identification des objets d'un modèle (notamment numéros des blocs) entre crochets sur les objets dans le modèle. Ces numéros sont des identificateurs uniques et permanents pour les blocs comme pour les textes et objets graphiques. Les blocs hiérarchiques et leur contenu affichent deux numéros : le premier numéro est le numéro global (ID propre à chaque bloc), le second est le numéro local (ayant sa logique à l'intérieur du sous-modèle).

Voir l'ordre de la simulation

Affiche sur chaque bloc un numéro qui est celui de l'ordre d'exécution des blocs durant une simulation. Les blocs internes aux blocs hiérarchiques ont leur propre ordre de simulation relatif au bloc parent. L'ordre de la simulation n'a de sens que pour les modèles continus.

Définir ordre de la simulation

Permet d'affecter aux blocs un numéro d'ordre d'exécution durant une simulation. Pour modèles continus, niveau avancé.

Créer/modifier un lien dynamique

Ouvre le dialogue Lien de sorte à lier l'élément de dialogue ou la table de données à une structure de données inter (BDD ExtendSim ou matrice globale). Cette action crée un lien bidirectionnel entre les deux structures de données. Les éléments de dialogue unique peuvent être liés à une cellule de matrice globale ou de table de BDD ; les tables seront liées à des matrices ou des tables.

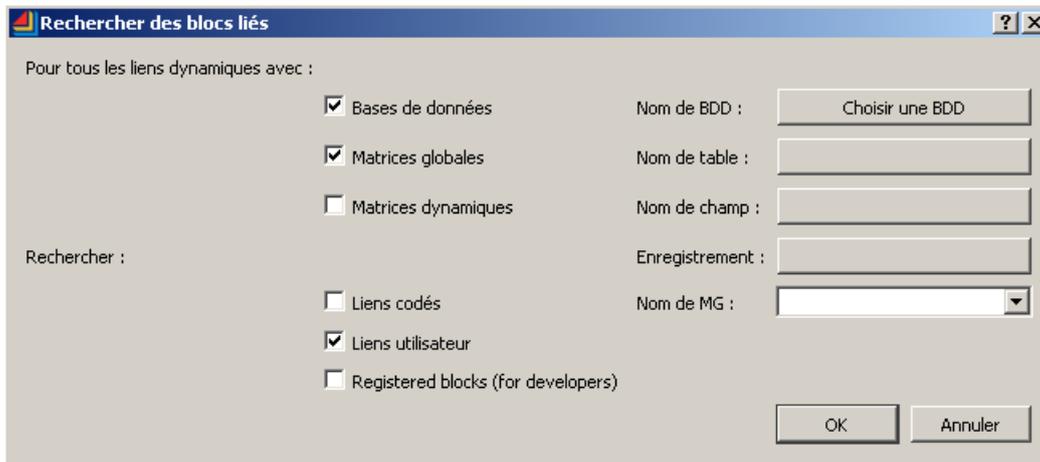
Une fenêtre apparaît pour vous permettre de choisir la structure de données, et les options s'adaptent à

celle-ci. Pour supprimer le lien, vous devez choisir cette même commande et sélectionner le bouton Effacer lien.

L'équivalent de cette commande est de cliquer du bouton droit sur un élément de dialogue et choisir Créer/modifier un lien dynamique dans le menu, ou de cliquer sur le bouton Lien dans une structure en table. Une table ayant un tel lien comporte les lettres DB ou GA dans le coin supérieur gauche.

Ouvrir les blocs avec liens dynamiques

Ouvre le dialogue *Rechercher les blocs liés*, présenté ci-dessous.



Ce dialogue permet de trouver les blocs avec un élément lié, mais aussi de retrouver les blocs « enregistrés » ('registered'). C'est un système qui permet aux développeurs que les blocs reçoivent un message spécifique lorsqu'est modifiée la source de données à laquelle le bloc est associé.

Les options du dialogue permettent de choisir la cible de la recherche (le contenu de bases de données, des matrices globales et/ou des matrices dynamiques), et l'objet de la recherche (des liens codés, des liens utilisateur ou des blocs enregistrés). Les *Liens utilisateurs* sont ceux créés par l'interface utilisateur du modèle (bouton lien), les *Liens codés* sont ceux définis par des fonctions dans le code ModL des blocs. Les *Blocs enregistrés* n'ont pas d'élément de dialogue spécifiquement associé à une source, c'est le bloc en entier qui est informé en cas de modification de la source de données.

Les menus sur la droite permettent de choisir la base de données ou la matrice à parcourir. Si aucun nom n'est choisi, la recherche s'appliquera à toutes les BDD et toutes les matrices globales.

Lorsque vous cliquez sur le bouton OK, ExtendSim ouvre le dialogue de tous les blocs correspondant aux liens spécifiés.

Paramètre de sensibilité...

Si un champ numérique de dialogue est sélectionné, cette commande ouvre le dialogue Paramètres de sensibilité, qui permet de paramétrer une [analyse de sensibilité](#). Un champ « sensible » a un cadre supplémentaire, qui est gris si l'analyse de sensibilité est activée, et rouge sinon.

Ouvrir les blocs sensibles

Ouvre le dialogue de tous les blocs ayant un élément affecté par l'analyse de sensibilité. Très utile pour repérer rapidement les blocs utilisés pour une analyse de sensibilité.

Insérer un objet...

Affiche la liste des objets connus (ActiveX/OLE) pouvant être insérés dans un modèle ExtendSim ou un dialogue de bloc.

Mode Design

Souvent l'application source d'un objet imbriqué s'ouvre si l'on double-clique sur celui-ci. Parfois le simple clic fait pareil : Mode Design fait alors que le double-clic s'y substitue, ce qui permet de sélectionner ou déplacer un objet sans activer l'application source.

Objet

Certains objets imbriqués ont des options de l'application externe qui sont prise en charge par ExtendSim. Ce menu s'active lorsqu'on sélectionne un objet imbriqué. Le contenu du menu dépend du type d'objet.

Verrouiller le modèle

Empêche toute modification du modèle sauf dans les dialogues des blocs. Cette commande sert aussi pour les blocs hiérarchiques, et l'utilisateur ne peut pas ajouter des liens ou des éléments graphiques, etc. C'est une commande utile lorsque vous devez confier des modèles à des personnes qui ne connaissent pas ExtendSim et pourraient les endommager. Pour déverrouiller le modèle, choisissez à nouveau la commande Verrouiller modèle. Attention : sauvegardez bien le mot de passe ou une version non verrouillée du modèle !

Menu Base de données

S'emploie pour gérer soit la structure d'une base de données, soit les données qu'elle contient. Pour ouvrir une fenêtre de BDD vous pouvez cliquer sur le nom de la base en bas du menu Base de données, soit cliquer sur le nom de la base dans la fenêtre qui s'ouvre lorsque vous sélectionnez la commande *Liste des BDD* dans le menu Fenêtres.



Nouvelle base de données...

Ouvre un dialogue pour créer et nommer une nouvelle base de données ExtendSim. Montre la liste des BDD du modèle actuel, pour éviter les doublons.

Propriétés de la BDD...

Ouvre un dialogue pour renommer une BDD.

Importer une BDD...

Ouvre un dialogue pour créer une nouvelle base par l'importation d'un fichier texte exporté d'ExtendSim ou de SDI Industry. Si vous cliquez sur un nom de BDD existant, les données importées remplaceront celles qui existent.

Ajouter BDD...

Alors qu'une fenêtre de BDD est active, ouvre un dialogue pour importer un fichier texte exporté d'ExtendSim ou de SDI Industry. Les tables importées s'ajoutent à celles qui existent.

Exporter toute la BDD...

Exporte l'intégralité de la BDD sélectionnée (ses tables, champs et enregistrements) vers un format texte spécial. Le fichier sera utilisé pour transférer la BDD dans un autre modèle, la partager avec un autre utilisateur d'ExtendSim ou la travailler avec l'Add-In pour Excel.

Exporter les tables sélectionnées...

Exporte les tables sélectionnées (champs et enregistrements) vers un fichier texte spécial. Comme la commande *Exporter toute les BDD*, mais pour agir sur un sous-ensemble de la BDD.

Nouvelle table...

Ouvre un dialogue pour créer et nommer une nouvelle table dans la BDD courante. Montre la liste des BDD du modèle actuel, pour éviter les doublons. Cliquer sur OK pour créer la table une fois nommée.

Propriétés de la table...

Ouvre un dialogue, décrit ci-dessous dans « Dialogue des Propriétés de la table », pour éditer les propriétés de la table. Cette commande équivaut à faire un clic droit dans une table de BDD et sélectionner *Editer les propriétés de la table*.

Cloner les tables sur l'onglet...

Clone la ou les tables sélectionnées sur un autre onglet de la structure de la BDD. Il ne s'agit pas de copier ou dupliquer les tables, mais de disposer d'un double qui est modifié comme l'original.

Importer fichier délimité...

Copie des données provenant d'un fichier texte simple dans la table sélectionnée. Un dialogue apparaît proposant de choisir le délimiteur de champs (les lignes qui deviendront les enregistrements étant forcément séparées par un retour-chariot).

Exporter fichier délimité...

Copie des données provenant de la table sélectionnée dans un fichier texte simple. Un dialogue apparaît proposant de choisir le délimiteur de champs (les enregistrements étant forcément séparés par un retour-chariot). Le fichier pourra être ouvert par un tableur ou un éditeur de texte.

Ajouter nouveau champ...

Crée un nouveau champ pour la table sélectionnée, sous les autres champs existant. Un dialogue apparaît proposant de choisir les propriétés du champ (nom, format, initialisation etc.).

Pour supprimer un champ, utiliser la touche Suppr, l'option *Effacer Champs sélectionné(e)s* dans le menu contextuel, ou la commande Edition> Effacer Champs sélectionné(e)s.

Insérer nouveau champ...

Comme *Ajouter nouveau champ*, sauf que le champ se place au-dessus du champ sélectionné.

Propriétés du champ...

Ouvre le dialogue de Propriétés du champ. Sert à éditer les informations, comme le nom et le type du champ, qui ont été entrés dans ce dialogue quand le champ a été créé. Équivaut à faire un clic droit dans une table de BDD alors que la fenêtre de la BDD est en mode Structure.

Ajouter des enregistrements...

Crée de nouveaux enregistrements à la fin de la table sélectionnée. Un dialogue apparaît demandant le nombre d'enregistrements à ajouter.

Insérer des enregistrements ...

Comme *Ajouter des enregistrements*, sauf que les enregistrements se placent au-dessus de l'enregistrement sélectionné.

Supprimer enr. sélectionnés...

Supprime les enregistrements sélectionnés de la table sélectionnée.

Rendre la cellule aléatoire...

Ouvre le dialogue Distribution aléatoire BDD pour rendre aléatoire le contenu des cellules sélectionnées.

Rendre la cellule constante...

Rend constantes les valeurs des cellules sélectionnées.

Vérification index lecture/écriture

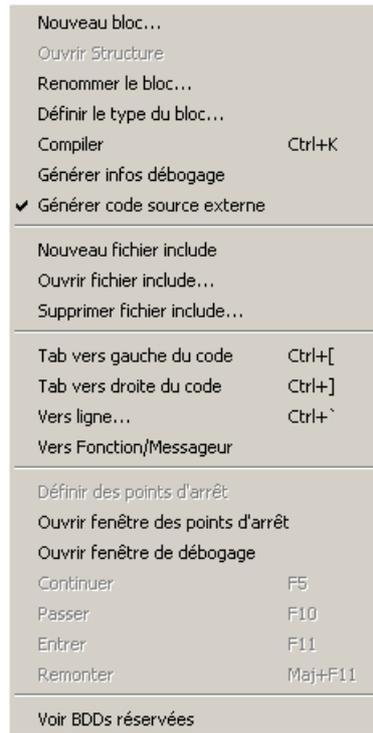
Si activée, ExtendSim émet un message d'erreur lorsque le modèle essaie de lire ou écrire des données via une adresse ou un index de BDD incorrect. Permet de déboguer le code de blocs interagissant avec la base de données. Coché par défaut pour les nouveaux modèles.

Liste des bases de données

Le bas du menu affiche la liste des bases de données ouvertes. Les bases de données réservées (voir menu Développeur) ont le caractère souligné en préfixe.

Menu Développeur

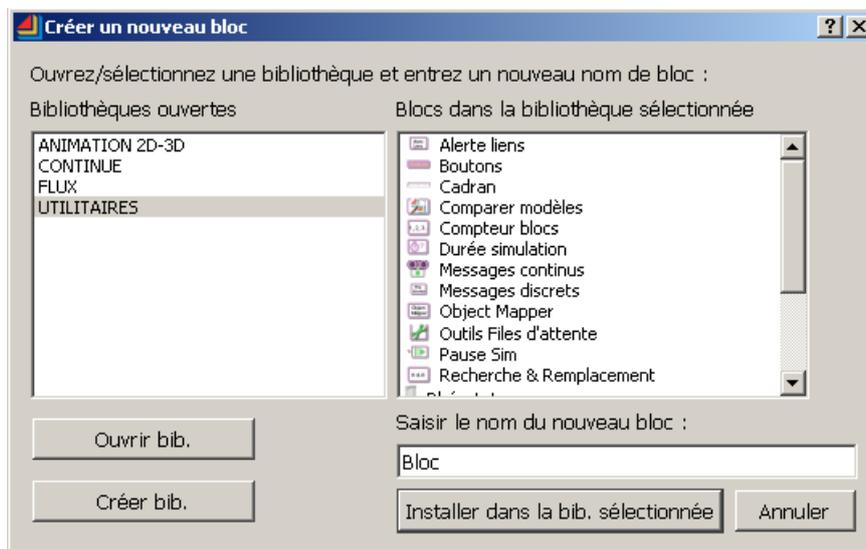
Le menu Développeur s'emploie pour créer ou modifier des blocs dans des bibliothèques.



menu Développeur

Nouveau bloc

Crée un nouveau bloc. Dans le dialogue initial, vous devez indiquer la bibliothèque dans laquelle vous voulez créer le bloc, ainsi que le nom du bloc. Utilisez les boutons Ouvrir bibliothèque ou Nouvelle bibliothèque pour ouvrir ou créer une bibliothèque pour le nouveau bloc.



dialogue Nouveau bloc

Une fois le bloc nommé est placé dans sa bibliothèque, la fenêtre de structure du nouveau bloc s'affiche avec les onglets pour modifier tous les aspects d'un bloc.

Ouvrir structure

Ouvre la fenêtre de structure des blocs sélectionnés. C'est l'équivalent du raccourci clavier Alt-double-clic (dans un modèle ou dans une fenêtre de bibliothèque).

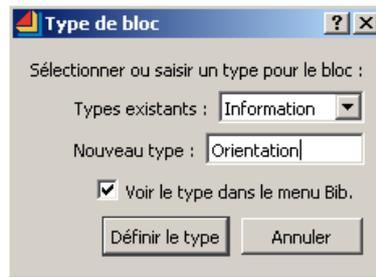
Renommer le bloc...

Permet de modifier un nom de bloc. Le bloc doit avoir sa fenêtre de structure ouverte pour la la

commande soit accessible.

Définir le type du bloc

Permet de définir le type du bloc, via le dialogue suivant :



dialogue Type de bloc

Le type d'un bloc peut avoir deux utilités :

- Permettre d'organiser les blocs par fonction dans une bibliothèque.
- Permettre d'organiser les blocs par fonction dans le rapport statistique

La commande *Définir le type du bloc* n'est accessible que lorsque la fenêtre de structure d'un bloc est active. Si l'option "Ranger les blocs par type dans les menus" n'est pas cochée, les noms figureront en ordre alphabétique lorsqu'on y accède par le menu Bibliothèque

Compiler

Compile le script du bloc. C'est utile pour vérifier la syntaxe du script en cours sans devoir fermer la fenêtre de structure. Cette option n'est accessible que lorsque la fenêtre de structure est la fenêtre active.

Générer infos débogage

Génère ces informations dans le code source lors de la compilation du bloc, si coché. Les blocs ayant du code de débogage sont plus lents et figurent en rouge dans la fenêtre de bibliothèque. Le menu contextuel du bloc permet d'ôter ce code supplémentaire.

Générer code source externe

Si cette commande est sélectionnée au moment de compiler un bloc, ExtendSim génère un fichier texte externe contenant le code source du bloc, ce qui peut servir pour contrôler les versions ou utiliser un autre éditeur. Si un bloc a été compilé de cette manière, cela sera indiqué dans la fenêtre de bibliothèque par les lettres CM (« code management ») à la droite de l'icône.

Nouveau fichier include

Crée une nouvelle de fenêtre pour un fichier Include.

Ouvrir fichier include

Ouvre une fenêtre existante pour un fichier Include.

Supprimer fichier include

Efface un fichier Include.

Tab à gauche

Déplace les lignes sélectionnées d'une tabulation vers la gauche. Accessible uniquement si l'onglet Script est au premier plan.

Tab à droite

Déplace les lignes sélectionnées d'une tabulation vers la droite.

Vers ligne

Amène le curseur sur la ligne dont vous spécifiez le numéro. Permet de retrouver facilement la ligne d'un script à l'origine d'un message d'erreur.

Vers Fonction/Messageur

Amène le curseur à l'emplacement de la définition d'une fonction ou d'un messageur. Sélectionnez auparavant le nom de la fonction. Equivalent à faire Alt-double clic sur le nom de la fonction dans le code.

Définir des points d'arrêt

Définissez un point d'arrêt dans la fenêtre de débogage d'un bloc en cliquant sur l'un des tirets dans la marge de gauche. En cliquant à nouveau vous annulez le point d'arrêt. Cette fenêtre montre le code source des blocs, avec une marge à gauche pour y placer des points d'arrêt. Si besoin est, le bloc est recompilé automatiquement pour y ajouter le code de débogage

Ouvrir fenêtre des points d'arrêt

Ouvre la fenêtre des Points d'arrêts (ou la met au premier plan) pour les blocs sélectionnés.

Ouvrir fenêtre de débogage

Ouvre la fenêtre de débogage du code source (ou la met au premier plan) pour les blocs sélectionnés

Continuer

Continue l'exécution jusqu'au point d'arrêt suivant. N'est accessible que lorsque la fenêtre de débogage est au premier plan, en cours d'exécution.

Passer

Passer par-dessus un appel de fonction.

Rentrer

Rentre dans un appel de fonction.

Remonter

Remonte jusqu'à l'appel de la fonction.

Montrer les BDD réservées

Liste les BDD réservées du modèle, s'il y en a, en bas du menu BDD et dans la Liste BDD. Les BDD réservées permettent aux programmeurs d'utiliser les capacités des BDD sans que l'utilisateur du modèle en soit conscient. Les BDD réservées ne peuvent être créées et modifiées que par l'API BDD ExtendSim.

Menu Simulation

Le menu Simulation permet de modifier la manière dont se déroule une simulation. Un sous-menu contient les commandes liées au débogage.



menu Simulation

Lancer la simulation

Démarre une simulation. La commande vérifie d'abord chaque bloc du modèle pour voir s'il est bien compilé.

Lancer Optimisation ou Scénarios

Lance une optimisation du modèle ou une analyse de scénarios (il faut que dans le modèle soit présent un bloc [Optimizer](#), ou un bloc [Scenario Manager](#) – mais pas les deux !). Une alerte avertie si la commande ne peut être exécutée.

Analyse de sensibilité

Fait qu'ExtendSim utilise les paramètres de sensibilité définis pour la simulation. L'option n'est disponible que si au moins un élément de dialogue est "sensible".

Paramètres...

Ouvre un dialogue pour modifier les paramètres temporels de la simulation.

Animation 2D

Permet de voir l'animation de blocs animés durant la simulation. Notez que certains blocs affichent une animation même si cette option n'est pas sélectionnée.

Animer les liens

Pour les modèles discrets, permet de voir le trajet des symboles d'animation sur les liens entre bloc et le suivant durant la simulation (si la commande Animation 2D est active). L'option est sélectionnée par défaut.

Animer les liens nommés

Comme l'option précédente, mais si le lien est un lien nommé (invisible sur le modèle), cette option fera que le symbole ira en ligne droite d'un label de texte à l'autre. Sinon, le symbole disparaît lorsqu'il atteint le label, réapparaît à l'emplacement du label associé et progresse jusqu'au bloc.

Animation 3D

Ouvre la fenêtre d'animation 3D, qui sera rouverte à chaque début de simulation. C'est l'équivalent de choisir *Voir l'animation 3D durant la simulation* dans l'onglet Animation 3D des paramètres de simulation.

Lancer Stat::Fit

Lance le module complémentaire de fitting statistique. Il faut que Stat::Fit soit installé (s'il fait partie de vos options).

Stop

Stoppe la simulation. Equivalent du bouton Stop dans la barre de statut.

Pause

Fait une pause temporaire dans la simulation. Equivalent du bouton Pause dans la barre de statut.

Continuer

Reprend une simulation en pause. Equivalent du bouton Continuer dans la barre de statut.

Débogage

Ce menu hiérarchique s'emploie pour mettre au point des modèles en suivant pas à pas leur déroulement, à l'écran ou dans un fichier. Les options de pas à pas ralentissent l'animation.

Générer la trace

Fait qu'ExtendSim génère un fichier trace en cours de simulation. Le nom du fichier Trace est demandé au début de la simulation.

Ajouter ces blocs à la trace

Les blocs sélectionnés sont inclus pour figurer dans la trace de la simulation.

Tout ajouter à la trace

Tous les blocs sont inclus pour figurer dans la trace de la simulation.

Retirer ces blocs de la trace

Les blocs sélectionnés sont retirés de la trace.

Tout retirer de la trace

Tous les blocs sont retirés de la trace.

Voir blocs de la trace

Les blocs figurant dans la trace sont signalés par le mot "trace" sur l'icône.

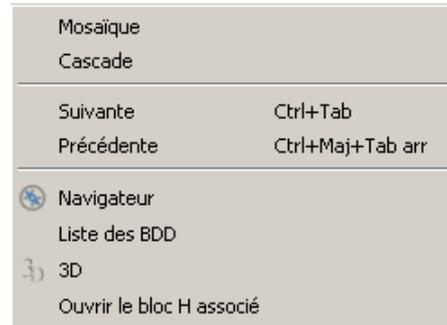
Générer la trace	
Ajouter ces blocs à la trace	Ctrl+8
Tout ajouter à la trace	
Retirer ces blocs de la trace	Ctrl+9
Tout retirer de la trace	
Voir blocs de la trace	
Profil des blocs	

Profil des blocs

Génère un fichier texte montrant le pourcentage de temps pris par chaque bloc durant l'exécution de la simulation.

Menu Fenêtres

Au bas liste toutes les fenêtres ExtendSim (modèles, dialogues et fenêtres de bibliothèques) qui sont ouvertes. Pour afficher une fenêtre en premier plan, sélectionnez-la dans le menu ou utilisez les touches de balayage.



Navigateur

Affiche la fenêtre de navigation pour un modèle (équivalent du bouton Navigateur dans la barre d'outils), afin des voir tous les éléments dans un modèle, y compris les niveaux hiérarchiques.

Liste des BDD

Affiche une fenêtre indiquant toutes les bases de données utilisées dans le modèle courant.

Animation 3D

Ouvre la fenêtre d'animation 3D. C'est l'équivalent du bouton 3D dans la barre d'outils.

Ouvrir le bloc H associé

Trouve le bloc hiérarchique qui contient le bloc sélectionné. Si ce bloc H parent est au niveau zéro, il sera sélectionné, s'il est d'un autre niveau, c'est la fenêtre du sous-modèle qui sera ouverte.

Menu Outils

Propose les noms des frises d'outils que vous pouvez activer ou désactiver selon que vous voulez les voir ou non.



Menu Aide

Aide ExtendSim

Ouvre une fenêtre qui affiche une liste de sujets d'aide disponibles.

Site de 1Point2

Lien vers le site de 1Point2, distributeur en France d'ExtendSim.

Vérifier mises à jour

Ouvre votre navigateur sur une page qui affiche la version la plus récente d'ExtendSim.

Site du concepteur

Ouvre votre navigateur sur le site de Imagine That, concepteur du logiciel.

Saisir la clé d'activation

Pour saisir la clé d'activation qui vous a été fournie lors de votre achat de la licence. Permet de transformer une version d'essai en licence durable.

Désactiver ExtendSim

Si vous changez d'ordinateur, vous devez désactiver ExtendSim sur l'ancien ordinateur afin de pouvoir le réactiver sur le nouveau.

Mise à jour clé d'activation

Cette commande est utile s'il y a eu un changement de votre licence, comme par exemple le renouvellement de votre maintenance (MSP), ou l'achat de cycles pour le module Reliability. La commande désactive la licence courante sans quitter ExtendSim, puis la réactive immédiatement sans changer de clé d'activation.

Licence réseau nomade

Cette commande n'est activée que si vous avez une licence flottante (réseau) d'ExtendSim faisant appel à un gestionnaire de licences.

Exporter une licence réseau

Permet de rendre nomade une des licences en la déconnectant du gestionnaire de licences. L'opération est effectuée en général pour un ordinateur portable, et réduit de un le nombre de licences utilisables sur le réseau.

Importer une licence réseau

Ramène au pool une licence nomade en la reconnectant au gestionnaire de licences. L'opération est effectuée en général au retour d'un ordinateur portable, et augmente de un le nombre de licences utilisables sur le réseau.

A propos d'ExtendSim

Le dialogue vous indique la version d'ExtendSim que vous utilisez. Cliquez sur la fenêtre pour continuer.

Nous contacter

N'hésitez pas à nous contacter pour toute information :

1Point2

5, rue de la Poste

38170 Seyssinet-Pariset

Tél.: (33) 04 76 27 77 85

Fax: (33) 04 76 27 24 67

email : infos@1point2.com

www.1Point2.com

ExtendSim est conçu par Imagine That, San Jose, USA.

1Point2 propose des formations standard à l'utilisation du logiciel ExtendSim et de ses modules. Nous proposons également une assistance à la modélisation, sous forme de journées isolées permettant d'optimiser la modélisation de votre système, ou de programmer des blocs « sur mesure ».

La simulation de flux est notre cœur de métier : 1Point2 peut bien entendu développer vos modèles après étude préalable : son expérience est importante sur un grand nombre de problématiques et de domaines d'application.

Index

1	
1Point2.....	296
A	
Abandon.....	149
Accélérer l'exécution.....	49
Activer un système.....	162
Activités.....	57, 157
Add-In ExtendSim-Excel.....	70
Aléatoire.....	47, 48
Analyse de sensibilité.....	87
Analyse multidimensionnelle.....	90
Animation.....	35, 79
Animation blocs Rate.....	259
Animation Tank.....	260
Animation Valve.....	260
Animer des blocs hiérarchiques.....	40
Arrêt des Machines.....	163
Arrivées aléatoires.....	135
Arrivées programmées.....	137
Attributs.....	56, 138
Attributs chaîne.....	141
Attributs correspondant aux coûts.....	191
Attributs DB Address.....	141
Attributs des composants.....	187
B	
Base aléatoire.....	47, 193
Bases de données.....	60
Biais.....	254
Bibliothèque Chart.....	107
Bibliothèque Item.....	53
Bibliothèque pour Runtime.....	26
Bibliothèque Rate.....	196
Bibliothèque Report.....	58
Bibliothèque Utilities.....	58
Bibliothèque Value.....	51
Bibliothèques.....	21
Bibliothèques ouvertes par défaut.....	279
Bloc Batch.....	186
Bloc Bias.....	253
Bloc Catch.....	170
Bloc Clear Statistics.....	49
Bloc Cost By Item.....	192
Bloc Diverge,Tableau des modes.....	228
Bloc Executive.....	58, 255
Bloc Executive et Rate.....	266
Bloc Gate.....	83
Bloc Holding Tank.....	78
Bloc Information.....	193, 194
Bloc Interchange.....	249
Bloc Merge,Tableau des modes.....	228
Bloc Notify.....	42
Bloc Optimizer.....	91
Bloc Query Equation.....	120
Bloc Queue Equation.....	118
Bloc Random Number.....	48
Bloc Resource Item.....	181
Bloc Select Item In.....	168
Bloc Select Item Out.....	172
bloc Shift.....	243
Bloc Shift.....	183
Bloc Throw.....	170
Bloc Unbatch.....	187
Blocage.....	148
Blocs Activity.....	157
Blocs d'animation.....	79
Blocs Rate.....	196
Blocs Statistiques.....	86
C	
Calcul des coûts.....	190
Capteurs du convoyeur flux.....	246
Changement de file.....	149
Chronométrer.....	195
Cible.....	101
Cible de facteur.....	101
Cloner des éléments.....	30
Clones.....	30
Coller.....	72
Combiner des ressources.....	191
Communication entre applications.....	77
Connecteur Arrêt.....	164
Connecteur Start.....	137
Connecteur Start.....	163
Connecteurs et liens.....	84
Connecteurs Flux.....	258
Connecteurs Rate.....	258
Connecteurs variables.....	25
Conseils généraux.....	78
Conserver l'individualité des entités.....	187
Construire un modèle.....	13
Continu et discret.....	21
Contrainte critique.....	220
Contrainte relationnelle.....	221
Contraintes du flux.....	220
Contrôle du flux.....	238
Contrôles.....	32
Convergence des flux.....	169
Convey Flow.....	244
Convoyeur.....	166
Couleurs.....	29
Coût d'attente.....	191
Coût fixe.....	190
Coût variable.....	190
Coûts.....	57, 84
Coûts dans Create.....	190
D	
Dates calendaires.....	45
DB Address.....	56
Débit infini.....	219
Débit maximum.....	218
Débit réel.....	218
Débogage.....	126

Débogueur	128
Debugger.....	292
Décisions d'orientation.....	171
Delta Time	46
Démarrer.....	13
Dérivation.....	148
Diagramme des sources et des cibles.....	101
Dialogue Paramètres.....	44
Disponibilité simultanée.....	189
Distributions aléatoires principales	194
DLL.....	75
Documenter.....	193
Durée des pannes	164

E

Editeur d'équations	117
Editeur d'équations	114
Emplacement des bibliothèques	22
Enregistrer un bloc hiérarchique	39
Entités chronométrées.....	195
Entités de passage.....	122
Entités discrètes.....	55
Entités Spawn.....	121, 122
Equation.....	114
Événements	55
Exporter une base de données.....	71

F

Facteurs	95
Fichier .BAK	274
Fichiers include	117
Fichiers texte	73
Figures géométriques	28
Files d'attente	147
Filtres Throw Catch.....	236
Flux continu.....	196
Formats graphiques.....	35
Fréquence d'arrivée	135
Frise d'outils.....	27

G

Génération d'entités	135
Gestionnaire de scénarios	95
Graphiques.....	107

H

Heure d'observation.....	193
Hiérarchie.....	36
Hiérarchie pure	36
Hot links.....	74
Hystérésis.....	243

I

Images.....	73
Indicateurs du convoyeur flux.....	246
Inter-process communication	77
Interrupteur	33
intervalle de confiance	87
IPC.....	77

J

Journal de bord	31
-----------------------	----

L

L'architecture Rate	196
Légendes de la souris.....	278
Licence d'exécution	12
Liens	284
Liens à angle droit.....	278
Liens nommés.....	35
Lire les attributs.....	142
Longueur d'une file d'attente.....	193
Lots	186
Lottage simple.....	188

M

Machines.....	158
Main d'oeuvre	181, 188
Matrices globales	74
Menu Aide.....	292
Menu Base de données.....	286
Menu Bibliothèque.....	282
Menu Débogage	291
Menu Développeur.....	288
Menu Edition.....	275
Menu Fenêtres.....	292
Menu Fichier.....	273
Menu Modèle	282
Menu Outils	292
Menu Simulation	290
Meter	33
Méthodes de plans d'expérience.....	105
Mise au point.....	126
Mode Batch/Unbatch	229
Mode Distributional	231
Mode Neutral.....	233
Mode Priority.....	230
Mode Proportional.....	230
Mode Select.....	229
Mode Sensing.....	232
Modèle "Car Wash"	13
Modèle "Reservoir".....	27
Modèle ouvert par défaut	278
Modélisations classiques	53
Modéliser des ressources	181
Multi-tâches.....	48
Multi-threading.....	48

N

Navigateur	31
Nombre d'entités.....	181
Nombre d'étapes	46
Nombres aléatoires.....	48, 193
NoValue.....	79
Numéro des menus.....	273
Numéro global.....	133
Numéro local	133

O

Objectif de durée.....	241
Objectif de quantité	239
Onglet Discrete Rate	255
Opérations.....	157
Optimisateur	91
Options avancées Discrete Rate.....	256
Outils de débogage	292

Outils pour les bibliothèques.....	282	Target.....	101
Ouvrir un modèle.....	13	Technologie PL.....	264
P			
Pannes aléatoires.....	164	Temps.....	48
Paramètres d'impression.....	274	Temps d'exécution.....	193
Période initiale.....	49	Temps partagé.....	159
Pièces assemblées.....	188	Texte libre.....	28
Plan factoriel complet.....	98	Throw Catch Flow.....	235
Planification.....	137	Timing d'ExtendSim.....	84
Planification des ressources.....	183	Trace d'un modèle.....	126
Police du code.....	280	Traitement cumulé.....	159
Précision des calculs.....	252	Traitement préparatoire.....	83
Préférence.....	254	Traitements en série.....	160
Priorité.....	56	Traitements parallèles.....	161, 171
Priorités.....	143	Tutorial.....	13
Profil d'un modèle.....	127	Types des blocs.....	289
Progression des entités.....	80	U	
Propriétés.....	56, 82	Unité métrique.....	278
Protéger une bibliothèque.....	26	Unités de flux.....	256
Q			
Quantité d'une entité.....	56	Unités de temps.....	45, 48
Quantités.....	144	V	
Queue.....	181	Variables d'entrée d'équations.....	114
Queue Equation.....	150	Variables de sortie d'équation.....	116
Queue Matching.....	153	Version française.....	11
R			
Rapport d'un modèle.....	106	Visualisation des coûts.....	192
Rapport du Gestionnaire de scénarios.....	100	Vues.....	42
Régime stable.....	49	Z	
Réglage.....	160	Zone d'animation.....	40
Relier des blocs.....	15	Zone PL.....	264
Répliquations.....	49, 89		
Réponses.....	97		
Report Manager.....	106		
Représentation des données.....	60		
Resource Pool.....	181		
Ressources.....	181		
Réunir des entités.....	186		
Runtime.....	12		
S			
Scénarios.....	99		
Section de flux.....	219		
Séparer des entités.....	186		
Série d'opérations.....	160		
Slider.....	32		
Source.....	100		
Source de facteurs.....	100		
Statistiques.....	53, 192		
Structure d'un bloc hiérarchique.....	38		
Style de liens.....	33		
Substituer un bloc.....	23		
Symboles d'animation.....	82		
Système fermé.....	182		
Système ouvert.....	182		
T			
Taille du lot.....	186		
Tank is separate.....	251		
Tank only exists.....	250		