



L'Atelier Syscience

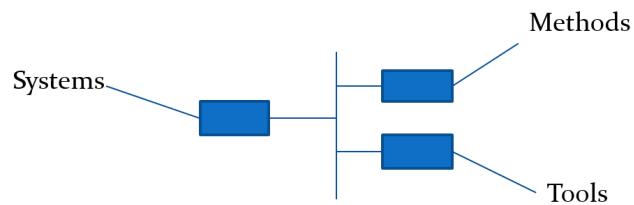
Manuel utilisateur

System: Manual

Pascal Krapf, Sébastien Berthier

Syscience

01/03/2023



Reference: Syscience UM V2



Table of contents

1	Introduction	4
1.1	Objet du document	4
1.2	Présentation	4
1.3	Documents de référence	4
1.4	Terminologie	4
1.4.1	Abréviations	4
1.4.2	Glossaire	5
2	Consultation	5
2.1	Présentation générale	5
2.2	Gestion des droits	5
2.3	Navigation	5
2.4	Dictionnaire de données	5
3	Actions sur la base de données	6
3.1	DICO	6
3.1.1	STRUCTURE	6
3.2	Mises à jour	7
3.2.1	PROCESS	7
3.2.2	UPLOAD	7
3.2.3	DOWNLOAD	7
3.3	Gestion du cycle de vie	7
3.3.1	DRAFT	7
3.3.2	FREEZE	7
3.3.3	OBSOLETE	8
3.3.4	DELETE	8
4	Diagrammes	8
4.1	Règles générales	8
4.2	Diagrammes de contexte	9
4.3	State	9
4.4	Sequence	11
4.5	Architecture	12
4.6	Diagramme de bloc central	13
4.7	DataTable	14
4.8	Tracé de courbes	15
4.9	Diagramme de bloc en arborescence	16
4.10	Diagramme de cas d'utilisation	17
4.11	Diagramme de Blocs multiples	18

4.12	Diagramme de Bloc unique	19
5	Exigences	20
5.1	Exigences générées automatiquement	20
5.2	Liens vers les exigences dans les diagrammes	20
5.3	Liens entre les exigences	21
6	Documents	21
6.1	Présentation	21
6.2	Document	21
7	Génération automatique de diagrammes	23
7.1	USECASE	23
7.2	STATE	23
7.3	ARCHITECTURE	23
7.4	CONTEXT	24

1 Introduction

1.1 Objet du document

Ce document est destiné à servir de référence aux utilisateurs de l'Atelier Syscience. Il décrit les fonctionnalités de l'atelier et donne des instructions pour faciliter son utilisation.

1.2 Présentation

L'Atelier Syscience est un logiciel d'ingénierie fonctionnant en mode SaaS. Il a pour objectif de rendre visuelles les données de conception pour des systèmes complexes et de faciliter un mode de conception collaboratif. Il est plus particulièrement destiné au déploiement de l'ingénierie système, mais peut être utilisé pour générer des documents quel que soit le domaine technique et la méthode utilisée.

1.3 Documents de référence

Les documents suivants constituent des références pour le déploiement de l'ingénierie système. Ils peuvent être utilisés indépendamment de l'Atelier Syscience.

- IEEE1220 : Standard for Application and Management of the Systems Engineering Process
- IEEE15288 : Systems Engineering - System Life Cycle Processes
- IEEE1471 : Recommended Practice for Architectural Description of Software-Intensive Systems

1.4 Terminologie

1.4.1 Abréviations

- MBSE : Model Based System Engineering
- SaaS : Software as a Service
- SysML : System Modeling Language
- IBD : Internal Block Diagram
- BDD : Block Definition Diagram
- SM : State Machine
- Seq: Sequence diagram

1.4.2 Glossaire

- Diagramme de séquence : diagramme représentant les activités de différents acteurs et les messages échangés.
- Diagramme de contexte : diagramme interne de blocs (IBD) représentant les éléments en relation avec un élément central.

2 Consultation

2.1 Présentation générale

Chaque utilisateur est associé à une base de données dite "client". Chaque base de données est associée à une adresse web. A cette adresse, les utilisateurs peuvent consulter l'ensemble des éléments de la conception qui sont gérés dans la base de données client qui leur est associée. Chaque utilisateur peut se connecter au site par son identifiant qui est son adresse mail et un mot de passe.

2.2 Gestion des droits

Pour utiliser l'Atelier Syscience, un utilisateur doit être enregistré. Chaque utilisateur enregistré possède un profil auquel sont attachés des droits d'accès. Ces droits d'accès sont gérés par l'équipe Syscience selon les besoins de chaque utilisateur.

Il existe deux types de profils utilisateur : le profil "standard" et le profil "administrateur". Certaines fonctions sont accessibles uniquement aux profils "administrateur".

2.3 Navigation

Le site web est organisé par systèmes. La page d'accueil du site internet donne la liste des systèmes présents en base de données. Chaque système peut contenir des documents et des diagrammes. Ceux-ci sont accessible directement par la navigation sur la page WEB.

On affiche le contenu d'un système en cliquant sur la zone active située devant son nom. On accède ainsi à la liste des documents et des diagrammes contenus dans ce système. Pour visualiser le contenu d'un document ou d'un diagramme, il suffit de cliquer sur l'icone .PDF associée. Chaque document contient des diagrammes qui décrivent différentes vues du système.

Un exemple peut être trouvé à l'adresse suivante: https://syscience.org/account_CLIENT/Exemple/ind

2.4 Dictionnaire de données

Un dictionnaire permet d'associer un alias alphanumérique à un objet, puis d'appeler cet objet à l'aide de son alias.

Le dictionnaire est accessible en lecture en cliquant sur le bouton DICO en haut de la page principale du site WEB de l'outil.

- La première colonne du tableau donne le type de l'objet : (System, StakeHolder, Actor, Function, Component, Interface).
- La seconde colonne donne son nom.
- La troisième donne son alias

Pour appeler un objet dans un diagramme par son alias, il suffit de remplacer le nom de l'objet par son alias précédé du symbole "#" et suivi de ":" (exemple: "#alias:"). Lors de la remontée en base de données, l'alias sera automatiquement associé à l'objet qu'il désigne dans le dictionnaire de données.

Le dictionnaire de données peut être édité dans un fichier tableau nommé "Table_Dico".

3 Actions sur la base de données

Les documents et ces diagrammes sont stockés dans une base de données. Le présent paragraphe décrit les interactions de l'utilisateur avec la base de données. Les actions sont accessibles dans l'onglet qui s'ouvre lorsque l'on clique sur l'icône en forme de crayon associée à un document ou un diagramme.

3.1 DICO

La fonction DICO permet d'interagir avec le dictionnaire de données de la base de données : si un fichier nommé Table_Dico est fourni en PJ, les nouveaux éléments du dictionnaire sont remontés en BDD.

Le retour de mail renvoie l'ensemble du fichier dico (qu'il y ait eu une pj au départ ou pas) avec les éléments préexistants auxquels sont ajoutés les nouveaux objets. Un objet existant peut être modifié.

C'est notamment par le DICO que peuvent être créés ou modifiés les systèmes.

3.1.1 STRUCTURE

Cette fonction permet d'obtenir le contenu de la base de données sous forme de liste (noms des systèmes, noms des documents par système et noms des diagrammes par système). Cette action est déclenchée par le bouton STRUCTURE accessible dans l'onglet UPLOAD

3.2 Mises à jour

3.2.1 PROCESS

Cette fonction permet de compiler un document ou un diagramme à partir de la base de données et de le mettre à jour sur le site WEB. Elle est déclenchée par le bouton PROCESS.

3.2.2 UPLOAD

Pour créer ou mettre à jour documents et diagrammes, l'utilisateur utilise le bouton UPLOAD. Il peut alors télécharger des données sont collectées dans des tableaux Excel selon un format type.

Cette fonction permet de remonter les documents, diagrammes et exigences dans la base de données.

Le système considéré est indiqué par une ligne « System » dans le document et dans chaque diagramme.

3.2.3 DOWNLOAD

Cette fonction permet de récupérer sous format Excel le documents numéro n avec ses diagrammes et exigences dans la base de données.

3.3 Gestion du cycle de vie

A chaque diagramme et à chaque document sont associé un cycle de vie constitué de 3 phases : DRAFT, FROZEN, OBSOLETE

3.3.1 DRAFT

Cette fonction permet de passer un document ou un diagramme à l'état DRAFT. Celui-ci est dès lors ouvert en modification.

3.3.2 FREEZE

Cette fonction permet de passer un document ou un diagramme à l'état FROZEN. Celui-ci ne sera plus modifiable et ne peut pas être supprimé.

3.3.3 OBSOLETE

Cette fonction permet de passer un document ou un diagramme à l'état OBSOLETE. Celui-ci n'est pas modifiable mais pourra être supprimé.

3.3.4 DELETE

Fonction utilisateur

Cette fonction permet de supprimer un document ou un diagramme de la base de données. Un document ou un diagramme pouvant avoir plusieurs versions, toutes ses versions seront supprimées par cette commande. Seuls les documents et diagrammes à l'état OBSOLETE peuvent être supprimés.

4 Diagrammes

4.1 Règles générales

Les données concernant des diagrammes sont envoyées à l'Atelier Syscience dans un fichier .xls ou .xlsx). Le nom de ce fichier doit commencer par "Multiple_". Les données doivent être dans le premier onglet qui sera le seul utilisé.

Lorsqu'elles sont traitées par l'Atelier Syscience, les données sont représentées par des diagrammes visuels au format jpeg ou Pdf. Chaque diagramme est décrit par un ensemble de lignes dans ce fichier. Le présent chapitre décrit le formalisme des fichiers tableurs correspondant aux différents types de diagrammes.

Quelques règles sont transversales pour tous les diagrammes :

- Les fichiers Multiples_ sont structurés en diagrammes. Un diagramme est un groupe de lignes dont la première case de la première ligne définit le type de diagramme (State, MultiBlock, etc).
- Dans un fichier Multiple_, la deuxième case de la ligne qui contient le type de diagramme donne le nom du diagramme. Le nom d'un diagramme doit comporter uniquement des caractères alphanumériques : a...z, A...Z, 0...9, ou un tiret bas (underscore) "_".
- Dans un diagramme, la première case de chaque ligne détermine l'utilisation qui sera faite des données de la ligne. L'utilisation des données est détaillée pour chaque type de diagramme dans les paragraphes suivants.
- La seconde case de la ligne contient un texte qui sera affiché différemment selon le contenu de la première case.
- La troisième case de la ligne contient une couleur, sélectionnée dans la liste des couleurs disponibles qui est accessible dans le fichier Color (cf §2).

4.2 Diagrammes de contexte

Un diagramme de contexte est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "Context".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- **System** : La case en colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD
- **central** : la case en colonne 2 contient le texte affiché sur la boîte centrale
- **environnement** : La case en colonne 2 contient le texte affiché dans une boîte périphérique et la colonne 6 le texte affiché sur le lien avec la boîte centrale
- **Template** : La case en colonne 2 donne la règle de génération des exigences



Figure 1: `Context_Example`

4.3 State

Un diagramme d'état est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "State".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- **Name** : A partir de la colonne 5, cette ligne contient les noms des états

- Color : Cette ligne contient la couleur pour chaque état
- Logo : Cette ligne contient le logo associé à chaque état
- Transition : ce type génère une flèche qui part de l'état désigné par "S" (source) et qui va vers l'état désigné par "T" (target). Le texte de la colonne 2 est positionné au dessus de la flèche.
- Activity : Le texte écrit dans la colonne 2 est écrit dans les états sélectionnés par un "X"
- Initial : Ce type permet de générer une transition initiale (flèche en provenance d'un petit disque) sur l'état sélectionné par un "X"
- Final : Ce type permet de générer une transition finale (flèche vers un petit disque entouré d'un cercle) sur l'état sélectionné par un "X"
- Template : Donne la règle de génération des exigences, [S] état de départ, [T] état d'arrivée, [Condition] condition de transition
- System : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD



Figure 2: **State_Modes**

4.4 Sequence

Un diagramme de sequence est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "Sequence".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- actor name : A partir de la colonne 5, cette ligne contient les noms des acteurs
- actor color : Cette ligne contient la couleur pour chaque acteur
- actor logo : Cette ligne contient le logo associé à chaque acteur
- activity : la deuxième case contient un texte qui sera écrit dans une boîte sur la gauche de l'image et relié à une barre verticale représentant une activité. Les colonnes de droite contiennent un "X" si l'activité est réalisée par l'acteur mentionné en tête de colonne. Si la colonne contient un "C" dans les cases situées en dessous, alors la barre d'activité est prolongée vers le bas tant qu'un "C" est présent. Si la colonne contient un "A" dans les cases situées au dessus, alors la barre d'activité est prolongée vers le haut tant qu'un "A" est présent.
- message : la deuxième case contient un texte qui sera écrit au dessus d'une flèche représentant un message. "S" indique l'acteur correspondant à l'expéditeur du message (départ de la flèche) et "R" indique le destinataire (arrivée de la flèche).
- St: la seconde colonne contient un texte qui sera placé dans un bloc aux coins arrondis placé sur la ligne de vie d'un acteur. L'acteur concerné sera indiqué par un X dans la colonne correspondante.
- blank : les données ne sont pas prises en compte et un espace séparera les activités qui précèdent des activités qui suivent.
- Template_Activity : nommage : un acteur est désigné par "[X]", l'activité est désignée par "[Activity]", le dernier message reçu est désigné par "[message]"
- Template_Message : Nommage : le texte du message est désigné par "[message]", la dernière activité est désignée par "[Activity]", l'acteur envoyant le message est désigné par "[S]", l'acteur recevant le message est désigné par "[R]";
- Case : La colonne 2 contient le nom du cas d'utilisation dans lequel le scénario se réalise. Il n'est pas affiché directement dans le diagramme de séquence, mais dans une diagramme de cas d'utilisation généré automatiquement.
- System : La colonne 2 contient le nom du système affiché dans la cartouche et utilisé pour la remontée en BDD



Figure 3: **Sequence_Collision Avoidance**

4.5 Architecture

Un diagramme d'architecture est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "Architecture".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- Name : A partir de la colonne 5, cette ligne contient les noms des blocs.
- Color : Cette ligne contient la couleur pour chaque bloc.
- Logo : Cette ligne contient le logo associé à chaque bloc.
- Flow : la colonne 2 contient un texte qui sera écrit au dessus d'une flèche partant du bloc marqué par "S" vers le bloc marqué par "T".
- Activity : activité insérée dans le compartiment de la boîte marquée par X.
- Outflow : même définition que "Flow" mais en plus le flux est associé à un port de sortie externe.
- System : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD.

Figure 4: **Architecture_Example**

4.6 Diagramme de bloc central

Un diagramme de bloc central est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "BlackBox".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- **Name** : A partir de la colonne 5, cette ligne contient les noms des blocs. Le premier bloc est le bloc central.
- **Color** : Cette ligne contient la couleur pour chaque bloc
- **Logo** : Cette ligne contient le logo associé à chaque bloc
- **Compartment** : La colonne 2 contient le texte à insérer dans un compartiment du bloc repéré par "X". Le même texte peut être utilisé dans plusieurs compartiments. Plusieurs lignes **Compartment** peuvent s'appliquer à un bloc donné.
- **Flow** : la colonne 2 contient un texte qui sera écrit au dessus d'une flèche partant du bloc marqué par "S" vers le bloc marqué par "T", à condition que l'un des deux soit le bloc central.
- **Sample** : la colonne 2 contient le texte qui sera présenté dans une boîte en dessous du diagramme pour servir de légende.
- **System** : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD



Figure 5: **BlackBox Example**

4.7 DataTable

Un tableau est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "DataTable".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- **Name** : A partir de la colonne 5, cette ligne contient les noms des colonnes
- **Color** : Cette ligne contient la couleur pour chaque colonne
- **Size** : Cette ligne contient la largeur de chaque colonne
- **Line** : ligne contenant des données
- **System** : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD

Figure 6: **DataTable_Example**

4.8 Tracé de courbes

Un tracé de courbes est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "Plot".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- Title : titre de la courbe
- Xaxis : nom de l'axe X
- Yaxis : nom de l'axe Y
- Xdata : les données au dela de la colonne 5 déterminent les abscisses des points à tracer
- Ydata : les données au dela de la colonne 5 déterminent les ordonnées des points à tracer
- Xline : Liste des abscisses
- Yline : Liste des ordonnées
- System : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD
- Trace : ce type délimite la fin d'une série de données à tracer. Les données suivantes seront positionnées sur un nouveau tracé.



Figure 7:

CallDiag: Figure(s) not found

4.9 Diagramme de bloc en arborescence

Un diagramme de bloc en arborescence est défini dans un fichier Multiple_ par un tableau qui commence par une ligne dont la première case est "Tree".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- System : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD
- L0 : désigne une boîte de niveau 0 dans l'arbre. la colonne B contient le nom de la boîte correspondante. La colonne C la couleur de la boîte et la colonne F le texte du compartiment.
- L1 : désigne une boîte de niveau 1 dans l'arbre.
- L2 : désigne une boîte de niveau 2 dans l'arbre.
- et ainsi de suite jusqu'à L9
- MuteL0 : les boîtes de niveau 0 ne sont pas tracées. Cette opération n'est valide que pour le niveau L0.

Figure 8: **Tree_Example**

4.10 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est défini dans un fichier `Multiple_` par un tableau qui commence par une ligne dont la première case est "UseCase".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- Name : A partir de la colonne 5, cette ligne contient les noms des acteurs
- Color : Cette ligne contient la couleur pour chaque acteur
- Logo : Cette ligne contient le logo associé à chaque acteur
- Central : nom de la boîte rectangulaire qui contient les use cases
- Case : cas d'utilisation principal
- SCase : cas d'utilisation spécialisé (sous cas d'utilisation)
- ECase : Extended UseCase
- ICase : Included Use Case
- Template : Donne la règle de génération des exigences
- System : La colonne 2 contient le nom du système affiché dans le cartouche et utilisé pour la remontée en BDD



Figure 9: UseCase_Example

4.11 Diagramme de Blocs multiples

Un diagramme de blocs multiples (MultiBlock) est défini dans un fichier Multiple_ par un tableau qui commence par une ligne dont la première case est "MultiBlock".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- Central : nom de la boîte rectangulaire centrale
- Lateral : nom d'une boîte rectangulaire latérale
- Logo : nom du logo associé à la dernière boîte (latérale ou centrale) mentionnée dans les lignes précédentes.
- Compartment : Texte de compartiment associé à la dernière boîte (latérale ou centrale) mentionnée dans les lignes précédentes.
- Input : texte associé à une ligne munie d'une flèche entrante de la dernière boîte (latérale ou centrale) mentionnée dans les lignes précédentes.
- Output : texte associé à une ligne munie d'une flèche sortante de la dernière boîte (latérale ou centrale) mentionnée dans les lignes précédentes.
- Line : texte associé à une ligne sans flèche liant la boîte centrale à la dernière boîte latérale mentionnée dans les lignes précédentes.

Figure 10: **MultiBlock_Example**

4.12 Diagramme de Bloc unique

Un diagramme de bloc unique (SingleBlock) est défini dans un fichier Multiple_ par un tableau qui commence par une ligne dont la première case est "SingleBlock".

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- Central : nom de la boîte rectangulaire centrale
- Logo : nom du logo associé à la boîte centrale.
- Compartment : Titre d'un compartiment associé à la boîte centrale.
- Compartment_Texte : Texte inclus dans un compartiment associé à la boîte centrale.
- Compartment_Item : Élément d'une liste à puces dans un compartiment associé à la boîte centrale.
- Input : texte associé à une ligne munie d'une flèche entrante de la boîte centrale.
- Output : texte associé à une ligne munie d'une flèche sortante de la boîte centrale.

Figure 11: **SingleBlock_Example**

5 Exigences

5.1 Exigences générées automatiquement

Les diagrammes donnent une description visuelle du système à produire. Cette description peut être rendue textuelle en générant automatiquement des exigences à partir des diagrammes. La présence d'une chaîne de caractères dans la colonne D déclenche la génération d'exigence pour la ligne concernée. La référence de l'exigence sera le nom du diagramme suivi de la chaîne de caractères de la colonne D. Le texte de l'exigence est généré à partir d'un schéma standard défini pour chaque type de diagramme. Il est possible d'utiliser un schéma de génération d'exigence personnalisé en le définissant par une ligne commençant par "Template".

Une ligne commençant par "Attribute" permet d'ajouter un attribut à l'exigence générée. Le texte (contenu) de l'attribut est donné dans la colonne B. Le nom de l'attribut est donné colonne D (même colonne que l'identifiant d'exigence).

5.2 Liens vers les exigences dans les diagrammes

Il est possible d'insérer dans un diagramme un lien vers une exigence.

- Pour une exigence générée, il faut indiquer entre crochets le nom du diagramme suivi d'un espace suivi de l'identifiant de l'exigence (Id) entre crochets.

Exemple : Sequence.Test 01 pour une exigence définie dans le diagramme Sequence.Test dont l'Id est 01 .

- Pour une exigence ajoutée manuellement, il faut indiquer entre crochets "REQ" suivi d'un espace suivi de l'identifiant de l'exigence (Id).

Exemple : [REQ 01] pour une exigence dont l'Id est 01.

5.3 Liens entre les exigences

Il est possible de définir des liens d'une exigence vers d'autres exigences. Pour cela, il suffit d'associer à une exigence un attribut nommé Link qui contient entre crochets la référence des exigences vers lesquelles on veut faire un lien. Si l'exigence vers laquelle on veut pointer le lien se situe dans un autre système, alors on précède l'exigence de l'alias de ce système suivi de ":", sans espace ni avant ni après les ":". Par exemple : [S1:MultiBlock_{productionNeedsR003}]pouruneexigencedanslesystèmeS1.

6 Documents

6.1 Présentation

Les diagrammes générés par l'atelier Syscience peuvent être intégrés à des documents générés au format PDF. Ces documents possèdent des liens hypertextes qui permettent de naviguer entre les diagrammes et les exigences.

6.2 Document

Le document est généré à partir de données structurées en tableau. Le tableau se lit ligne par ligne, de haut en bas. Le document est généré de façon séquentielle, ligne par ligne.

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- DocTitle: la case en colonne 2 définit le titre du document qui sera écrit en page de titre et repris en bas de chaque page du document.
- DocSubTitle: la case en colonne 2 définit un sous-titre du document, écrit en page de titre
- System: la case en colonne 2 définit le système auquel correspond le document. Il est écrit en page de titre et sert aussi pour remonter le document en base de données.
- Author: la case en colonne 2 définit le nom des auteurs du document, écrits en page de titre

- Organization: la case en colonne 2 définit le nom de l'organisateur qui génère le document, écrit en page de titre
- Date: la case en colonne 2 définit la date du document, écrite en page de titre
- Reference: la case en colonne 2 définit une référence pour le document, écrite en page de titre
- Disclaimer: la case en colonne 2 définit un avertissement, écrit en page de titre
- DocType: la case en colonne 2 permet de créer un document sans en-tête si la seconde colonne contient "NOCOVER".
- T1: la case en colonne 2 contient un titre de niveau 1
- T2: la case en colonne 2 contient un titre de niveau 2
- et ainsi de suite jusqu'au titre de niveau 6, T6.
- Text: la case en colonne 2 contient un paragraphe de texte
- Item: la case en colonne 2 contient un élément d'une liste à puces. Toutes les puces sont du même niveau.
- Picture: la case en colonne 2 permet d'appeler une image au format JPG
- Diagram: la case en colonne 2 contient un texte de caption. La case en colonne 3 contient le nom du diagramme à intégrer.
- CallDiag: la case en colonne 2 permet d'appeler un ou plusieurs diagrammes selon le début du nom : "Context_N" pour tous les diagrammes de contexte commençant par N. "State_T" pour tous les diagrammes d'état commençant par T etc. Le début du nom des diagrammes appelés est indiqué en colonne 3. Si la case est vide alors tous les diagrammes sont appelés.
- CallReq: la case en colonne 2 permet d'appeler des exigences à partir de la valeur d'un de leur attribut. La valeur est indiquée en colonne 2 et le nom de l'attribut en colonne 3. Si le texte en colonne 2 se termine par le symbole * alors sont appelées toutes les exigences dont le texte de l'attribut commence par ce texte. Si il n'y a pas de symbole * alors seules sont appelées les exigences où la correspondance est exacte. Les exigences générées à partir d'un diagramme ont un attribut nommé "Diagram" dont la valeur est le nom du diagramme.
- BreakPage: permet de passer à la page suivante.
- Language: la case en colonne 2 permet de définir la langue des textes standard (sommaire, légendes etc). Les valeurs possibles en colonne 2 sont : "french", "english", "german", "spanish".
- Activate: la case en colonne 2 permet d'activer la génération automatique de certains tableaux. La colonne 2 peut prendre les valeurs : StateDictionary, ActivityDictionary, ObjectDictionary. Selon cette valeur des tableaux de synthèse sont intégrés en fin de document

7 Génération automatique de diagrammes

Certains diagrammes peuvent être générés automatiquement à partir de diagrammes existants. Pour générer automatiquement un diagramme, on insère dans le fichier Multiple le type de diagramme attendu et comme nom de diagramme "Generic"

7.1 USECASE

Un diagramme de cas d'utilisation peut être généré à partir d'un batch de diagrammes de séquence. Dans le diagramme généré, les scénarios apparaissent rattachés aux cas d'utilisation. Chaque diagramme de séquence peut être associé à un cas d'utilisation en insérant une ligne "Case" qui précise le nom du cas d'utilisation. Dans le fichier Use-Case-Generic une ligne "Diagram" précise le nom du diagramme généré. Puis les lignes "Case" permettent de sélectionner les cas d'utilisation visualisés. Les acteurs sont insérés automatiquement à partir des diagrammes de séquence.

7.2 STATE

Un diagramme d'état peut être généré à partir de diagrammes de séquence dans lesquels apparaissent des états. La génération de diagramme est déclenchée si le fichier Multiple_ comporte une ligne contenant en première colonne State et en seconde Generic.

7.3 ARCHITECTURE

Un diagramme d'architecture peut être généré à partir de diagrammes de séquence ou de diagrammes SingleBlock. Les acteurs des diagrammes de séquence et les éléments centraux des SingleBlock donnent les blocs d'architecture. Les messages des diagrammes de séquence et les Inputs / Outputs des SingleBlock donnent les flèches du diagramme d'architecture. Les doublons sont éliminés automatiquement.

La génération de diagramme est déclenchée si le fichier Multiple_ comporte une ligne contenant en première colonne Architecture et en seconde un nom terminé par _Generic

Les données de chaque ligne sont utilisées de façon suivante en fonction du contenu de la première case de la ligne.

- **Select** : permet de sélectionner les diagrammes utilisés pour la génération automatique des diagrammes. La deuxième colonne contient la section initiale des nom des diagrammes à utiliser. Par exemple Sequence_USN utilisera tous les diagrammes de séquence débutant par Sequence_USN. Le fichier peut contenir autant de lignes "Select" que nécessaire pour sélectionner les diagrammes à utiliser.
- **Activate**: La case en colonne 2 permet d'activer des fonctionnalités complémentaires. Activate | Compartment permet de créer dans le diagrammes d'architecture des

compartiments qui contiennent toutes les activités réalisées par l'acteur correspondant dans les diagrammes de séquence. Activate | BlackBox permet de générer l'activation de diagrammes BlackBox pour chaque bloc du diagramme d'architecture.

7.4 CONTEXT

Un diagramme de contexte peut être généré à partir d'autres diagrammes de contexte (en tant que synthèse de parties prenantes). La génération de diagramme est déclenchée si le fichier Multiple_ comporte une ligne contenant en première colonne Context et en seconde Generic