

EDF R&D

MANAGEMENT DES RISQUES INDUSTRIELS

ETUDES PROBABILISTES DE SURETE ET DE DISPONIBILITE DES SYSTEMES

7 boulevard Gaspard Monge 91120 PALAISEAU - +33 (1) 78 19 32 00

27/10/2016

Syntaxe du langage de modélisation probabiliste Figaro

BOUISSOU Marc

BUFFONI Lena

HOUDEBINE Jean-Christophe

EDF R&D - MRI

Linköping University

Société Aristè

6125-1612-2016-15549-FR

1.0

Type d'information : Note technique

Ce document fournit la description détaillée de la syntaxe du langage Figaro. Il est destiné à accompagner le manuel de référence du langage qui explique les principes de modélisation, l'existence de deux niveaux appelés ordre 1 et ordre 0, la sémantique du langage et la façon de l'utiliser pour construire des bases de connaissances permettant la modélisation fiabiliste des systèmes discrets. Le présent document est conçu pour être facile à utiliser par une personne connaissant les concepts du langage Figaro, mais pas la façon détaillée de les mettre en œuvre ; il présente successivement les aspects suivants : 1) le lexique du langage en Anglais et en Français, 2) la grammaire du langage, 3) les contrôles grammaticaux effectués sur la base de connaissances, 4) les fonctions prédéfinies dans le langage.

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

Circuit de validation

Auteur	BOUISSOU Marc	01/09/2016	
Vérificateur	CHAUDONNERET Thomas	06/09/2016	
Approbateur	PESME Helene	27/10/2016	

Code affaire	P11DD
---------------------	-------

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

Liste de diffusion

Groupe destinataire
1612-EPSDS

Pré-diff	Diff	Destinataire	Structure	E-mail
	X	BOUISSOU Marc	EDF R&D - MRI	marc.bouissou@edf.fr
	X	BOUSKELA Daniel	EDF R&D - STEP	daniel.bouskela@edf.fr
X	X	BUFFONI Lena	Linköping University	lena.buffoni@liu.se
	X	CHAUDONNERET Thomas	EDF R&D - MIRE	thomas.chaudonneret@edf.fr
X	X	HOUEBINE Jean-Christophe	Société Aristè	ariste@wanadoo.fr
	X	LAAROUCI Youssef	EDF R&D - SINETICS	youssef.laarouchi@edf.fr
	X	PESME Helene	EDF R&D - MRI	helene.pesme@edf.fr
	X	THUY N	EDF R&D - STEP	n.thuy@edf.fr

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

AVERTISSEMENT / CAUTION

L'accès à ce document, ainsi que son utilisation, sont strictement limités aux personnes expressément habilitées par EDF.

EDF ne pourra être tenu responsable, au titre d'une action en responsabilité contractuelle, en responsabilité délictuelle ou de toute autre action, de tout dommage direct ou indirect, ou de quelque nature qu'il soit, ou de tout préjudice, notamment, de nature financière ou commerciale, résultant de l'utilisation d'une quelconque information contenue dans ce document.

Les données et informations contenues dans ce document sont fournies "en l'état" sans aucune garantie expresse ou tacite de quelque nature que ce soit.

Toute modification, reproduction, extraction d'éléments, réutilisation de tout ou partie de ce document sans autorisation préalable écrite d'EDF ainsi que toute diffusion externe à EDF du présent document ou des informations qu'il contient est strictement interdite sous peine de sanctions.

The access to this document and its use are strictly limited to the persons expressly authorized to do so by EDF.

EDF shall not be deemed liable as a consequence of any action, for any direct or indirect damage, including, among others, commercial or financial loss arising from the use of any information contained in this document.

This document and the information contained therein are provided "as are" without any warranty of any kind, either expressed or implied.

Any total or partial modification, reproduction, new use, distribution or extraction of elements of this document or its content, without the express and prior written consent of EDF is strictly forbidden. Failure to comply to the above provisions will expose to sanctions.

Sommaire

AVERTISSEMENT / CAUTION	1
SOMMAIRE	2
1. GÉNÉRALITÉS	3
2. COMMENTAIRES	3
3. LEXIQUE DU LANGAGE FIGARO.....	4
3.1. CONSTRUCTION DES IDENTIFICATEURS, NOMBRES ET DÉLIMITEURS	4
3.2. MOTS CLÉS DU LANGAGE FIGARO	4
3.2.1. <i>Principales entités Figaro</i>	4
3.2.2. <i>Opérateurs élémentaires</i>	5
3.2.3. <i>Interfaces</i>	5
3.2.4. <i>Termes propres aux règles d'occurrence</i>	6
3.2.5. <i>Termes propres aux règles d'interaction</i>	6
3.2.6. <i>Termes communs aux règles d'occurrence et d'interaction</i>	7
3.2.7. <i>VARIABLES, EDITION, DOMAINE ET VALEURS</i>	8
3.2.8. <i>Systèmes d'équations linéaires</i>	10
4. SYNTAXE DU LANGAGE FIGARO	11
4.1. DÉCLARATION DE LA LISTE DES ÉTAPES	11
4.2. DÉCLARATION DE LA LISTE DES GROUPES	11
4.3. DÉCLARATION DE LA LISTE DES SYSTÈMES D'ÉQUATIONS	12
4.4. DÉFINITION D'UN TYPE	12
4.5. DÉFINITION D'UN OBJET	12
4.6. DÉFINITION DES CARACTÉRISTIQUES D'UN TYPE OU D'UN OBJET	13
4.6.1. <i>Définition des Variables</i>	13
4.6.2. <i>Définition des Interfaces</i>	14
4.6.3. <i>Définition des Règles</i>	15
4.6.4. <i>Définition des équations</i>	16
4.6.5. <i>Actions</i>	17
4.6.6. <i>Définition des termes des expressions</i>	17
4.6.7. <i>Définition d'un modèle fiabiliste et des données associées</i>	18
5. CONTRAINTES GRAMMATICALES DU LANGAGE FIGARO 1	19
5.1. CONTRAINTES GRAMMATICALES LIÉES À L'HÉRITAGE	19
5.2. CONTRAINTES GRAMMATICALES LIÉES AUX INTERFACES	20
5.3. CONTRAINTES GRAMMATICALES LIÉES AUX EXPRESSIONS	20
6. ANNEXE : LISTE DES FONCTIONS.....	24
6.1. FONCTIONS MATHÉMATIQUES	24
6.2. FONCTIONS RETOURNANT UN NOMBRE ALÉATOIRE SUIVANT UNE LOI DONNÉE	24
6.3. FONCTIONS SPÉCIALES	27

1. Généralités

Ce document fournit la description détaillée de la syntaxe du langage Figaro. **Il est destiné à accompagner le manuel de référence du langage** qui explique les principes de modélisation, l'existence de deux niveaux appelés ordre 1 et ordre 0, la sémantique du langage et la façon de l'utiliser pour construire des bases de connaissances permettant la modélisation fiable des systèmes discrets.

Le présent document est conçu pour être facile à utiliser par une personne connaissant les concepts du langage Figaro, mais pas la façon détaillée de les mettre en œuvre ; il présente successivement :

- Le lexique du langage en Anglais et en Français
- La grammaire du langage
- Les contrôles grammaticaux effectués sur la base de connaissances.
- Les fonctions prédéfinies dans le langage

Le langage Figaro à l'ordre 1 permet de décrire des connaissances génériques, alors qu'à l'ordre 0 il ne peut être employé que pour décrire des systèmes particuliers et sert d'entrée à tous les algorithmes de traitement des modèles. En Figaro 1, on a deux types de fichiers : les fichiers de bases de connaissances et les fichiers de description de systèmes. Les bases de connaissances ne peuvent contenir que des types (TYPE) et des objets permanents et visibles de tous les autres objets (OBJET_SYSTEME), alors que les descriptions de systèmes ne peuvent contenir que des objets (OBJET). A l'ordre 0, il existe un seul type de fichier, qui ne peut contenir que des objets (OBJET). Le *parser* FIGARO a deux modes différents faits pour le contrôle de modèles à l'ordre 1 et à l'ordre 0.

A quelques détails près, qui seront signalés explicitement, le langage d'ordre 0 est un sous-langage de l'ordre 1.

Les conventions d'écriture sont les suivantes dans la définition de la syntaxe :

- | : indique un choix exclusif entre différentes possibilités,
- [c] : indique que la présence de la construction syntaxique c est facultative,
- *Caractères italiques* : indique un mot clé ou une construction disponible *seulement à l'ordre 1*,
- Les parenthèses servent à grouper des éléments : (a|b) c signifie (a ou b) suivi de c alors que a | (bc) signifie a ou (b suivi de c). Les parenthèses faisant partie du texte Figaro seront notées LPAR et RPAR.

La distinction entre majuscules et minuscules est importante. Les termes TERM et term représentent donc deux entités différentes. Tous les mots-clés du langage Figaro sont en majuscules.

2. Commentaires

Les commentaires sont autorisés partout dans le code Figaro. Tout texte compris entre « (*) » et « *) » est pris comme commentaire. Les commentaires imbriqués sont autorisés.

Exemple :

```
(* commentaire  
    (* commentaire imbriqué *)  
*)
```

3. Lexique du langage Figaro

Le lexique est l'ensemble des éléments atomiques de la syntaxe.

3.1. Construction des identificateurs, nombres et délimiteurs

contenu	suite de caractères attendue
lettre	A B ... Z a b... z _
chiffre	0 1 ... 9
suitecar (suite de caractères)	(lettre chiffre) (suitecar)
identificateur (appelé ID dans les diagrammes syntaxiques)	lettre suitecar
chaîne de caractères	suite de caractères <i>quelconques</i> sauf " et
LITERAL_STRING (chaîne de caractères entre doubles apostrophes)	"chaîne de caractères"
identificateur entre barres verticales (ID)	" " chaîne de caractères " "
identificateur entre apostrophes (QUOTID)	' suitecar '
entier (LITERAL_INTEGER)	chiffre (entier)
réel (LITERAL_REAL)	entier[entier] [(E e)[- +] entier]
fin d'ordre Figaro	;
séparateur d'identificateurs, d'actions	,
parenthèses ouvrante et fermante (LPAR et RPAR dans les diagrammes syntaxiques)	()
Préfixe de désignation d'une variable d'un système d'équations	?

3.2. Mots clés du langage Figaro

3.2.1. Principales entités Figaro

Ces mots-clés sont ceux que l'on va trouver au plus haut niveau dans l'arbre syntaxique d'un texte Figaro.

sens Figaro	terme Français	terme Anglais
annonce de la liste d'étapes	ORDRE_DES_ETAPES	STEPS_ORDER
annonce d'une condition d'étape	CONDITION	CONDITION
annonce de la liste des groupes	NOMS_DES_GROUPES	GROUP_NAMES

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

annonce de la liste des systèmes d'équations	NOMS_DES_SYSTEMES	SYSTEM_NAMES
annonce d'un type	TYPE	CLASS
lien d'héritage d'un type	SORTE_DE	KIND_OF
ensemble de tous les objets (Figaro 1)	OBJET	OBJECT
annonce d'un objet (Figaro 1 et Figaro 0)		
annonce d'un objet « permanent » et global	OBJET_SYSTEME	SYSTEM_OBJECT
lien d'héritage d'un objet	EST_UN EST_UNE	IS_A IS_AN

3.2.2. Opérateurs élémentaires

sens Figaro	terme Français	terme Anglais
opérateur d'assignation	<--	<--
opérateur booléen égalité assignation en Figaro 0	=	=
opérateur booléen différent	<>	<>
opérateur booléen inférieur strictement	<	<
opérateur booléen inférieur ou égal	<=	<=
opérateur booléen supérieur strictement	>	>
opérateur booléen supérieur ou égal	>=	>=
opérateur de multiplication	*	*
opérateur de division	/	/
opérateur d'addition	+	+
opérateur de soustraction	-	-
modulo (a % b désigne a modulo b)	%	%
puissance (a**b désigne a puissance b)	**	**
négation booléenne	NON	NOT
opérateur booléen ET	ET	AND
opérateur booléen OU	OU	OR

3.2.3. Interfaces

sens Figaro	terme Français	terme Anglais
annonce des INTERFACES Figaro	INTERFACE	INTERFACE
annonce du type des objets autorisés dans une interface	GENRE	KIND
annonce des limites du nombre d'objets admis dans une interface	CARDINAL	CARDINAL
annonce de la limite maximale du nombre d'objets admis dans une interface	JUSQUA	TO
limite infinie pour le nombre d'objets admis	INFINI	INFINITY

dans une interface		
--------------------	--	--

3.2.4. Termes propres aux règles d'occurrence

sens Figaro	terme Français	terme Anglais
annonce des règles d'occurrence	OCCURRENCE	OCCURRENCE
annonce d'une condition de règle s'ajoutant à la condition implicite liée à l'utilisation des mots-clés DEFAILLANCE ou REPARE suivis d'un nom de PANNE	SI	IF
annonce d'une transition	IL_PEUT_SE_PRODUIRE	MAY_OCCUR
annonce d'une transition alternative	OU_BIEN	OR_ELSE
transition standard	TRANSITION	TRANSITION
transition de type défaillance	DEFAILLANCE	FAULT
transition de type indisponibilité	INDISPONIBILITE	UNAVAILABILITY
transition de type réparation	REPARATION	REPAIR
annonce de la liste des pannes réparées par la transition	REPARE	REPAIRS
annonce du type de loi de probabilité liée à la transition	LOI	DIST
annonce des conséquences de la transition	PROVOQUE	INDUCING
loi de probabilité de type exponentielle	EXP EXPONENTIELLE	EXP EXPONENTIAL
loi de probabilité de type instantanée	INS INSTANTANEE	INS INSTANTANEOUS
loi de "probabilité" de type temps constant	T_C TEMPS_CONSTANT	C_T CONSTANT_TIME
autres lois, utilisables seulement dans des modèles destinés à être traités par simulation de Monte Carlo : cf. tableau au §6.	CYCLE, POINT, UNI (Uniform), TRIANG, ERL (Erlang), WEI WEIBULL, GUMBEL, FRECHET, PARETO, NORMAL, LGN, GAMMA, BETA	Mêmes noms qu'en français
Version « avec mémoire » des lois précédentes. On retrouve toutes les lois définissant des temps sauf la loi exponentielle, qui par définition est sans mémoire.	T_C_M, CYCLE_M, POINT_M, UNI_M, TRIANG_M, ERL_M, WEI_M, GUMBEL_M, FRECHET_M, PARETO_M, NORMAL_M, LGN_M, GAMMA_M, BETA_M	Mêmes noms qu'en français

3.2.5. Termes propres aux règles d'interaction

sens Figaro	terme Français	terme Anglais
annonce des règles d'interaction	INTERACTION	INTERACTION
annonce des étapes auxquelles la règle appartient	ETAPE	STEP

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

annonce des groupes auxquels la règle appartient	GROUPE	GROUP
annonce de la condition de la règle	SI	IF
annonce de l'action exécutée lors de la réalisation de la condition	ALORS	THEN
annonce de l'action exécutée lors de la réfutation de la condition	SINON	ELSE

3.2.6. Termes communs aux règles d'occurrence et d'interaction

sens Figaro	terme Français	terme Anglais
permet à une règle de faire référence à l'objet qui contient cette règle	<i>MOI_MEME</i>	<i>MYSELF</i>
quantificateur	<i>SOIT</i>	<i>GIVEN</i>
quantificateur	<i>QQSOIT</i>	<i>FOR_ANY</i>
quantificateur	<i>IL_EXISTE</i>	<i>IT_EXISTS</i>
annonce d'un ensemble d'objets (interface ou OBJET)	<i>UN UNE</i>	<i>A AN</i>
annonce d'un nom d'objet ou d'interface de cardinal 1	<i>DE</i>	<i>OF</i>
annonce d'un nom de type	<i>DE_TYPE</i>	<i>OF_TYPE</i>
annonce une condition constante	<i>VERIFIANT</i>	<i>VERIFYING</i>
annonce de la condition associée à un quantificateur QQSOIT	<i>ON_A</i>	<i>WE_HAVE</i>
annonce de la condition associée à un quantificateur IL_EXISTE	<i>TEL_QUE, TELLE_QUE</i>	<i>SUCH_THAT</i>
opérateur booléen de test de l'ensemble des pannes d'un objet	<i>MARCHE</i>	<i>WORKING</i>
opérateur booléen de test de l'ensemble des pannes d'un objet	<i>PANNE</i>	<i>FAILURE</i>
début de l'expression au moins ... parmi ou au moins x ...	<i>AU_MOINS</i>	<i>AT_LEAST</i>
annonce d'un ensemble d'expressions booléennes à tester suivant le mode au moins ...parmi	<i>PARMI</i>	<i>WITHIN</i>
opérateur de test de l'appartenance d'un objet à un ensemble (interface)	<i>INCLUS_DANS</i>	<i>INCLUDED_IN</i>
annonce un ensemble d'objets sur lesquels une action est définie	<i>POUR_TOUT</i>	<i>FOR_ALL</i>
annonce des actions associées au quantificateur POUR_TOUT	<i>FAIRE</i>	<i>DO</i>
opérateur somme	<i>SOMME</i>	<i>SUM</i>
opérateur produit	<i>PRODUIT</i>	<i>PRODUCT</i>

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

opérateur maximum	<i>MAXIMUM</i>	<i>MAXIMUM</i>
opérateur minimum	<i>MINIMUM</i>	<i>MINIMUM</i>
annonce des termes des opérateurs somme, produit, maximum et minimum	<i>DES_TERMES</i>	<i>OF_TERMS</i>

3.2.7. VARIABLES, EDITION, DOMAINE ET VALEURS

sens Figaro	terme Français	terme Anglais
annonce d'une chaîne de caractères de commentaire d'une facette d'un type	<i>LIBELLE</i>	<i>LABEL</i>
annonce des constantes Figaro	<i>CONSTANTE</i>	<i>CONSTANT</i>
annonce des paramètres de loi probabiliste (considérés comme des constantes)	<i>PARAMETRE_LOI</i>	<i>DIST_PARAMETER</i>
annonce des attributs Figaro	<i>ATTRIBUT</i>	<i>ATTRIBUTE</i>
annonce des effets Figaro	<i>EFFET</i>	<i>EFFECT</i>
annonce des pannes Figaro	<i>PANNE</i>	<i>FAILURE</i>
annonce d'une propriété d'édition d'une variable (constante, attribut, effet, panne)	<i>EDITION</i>	<i>EDITION</i>
valeur de la propriété d'édition d'une variable	<i>VISIBLE</i>	<i>VISIBLE</i>
valeur de la propriété d'édition d'une variable	<i>MODIFIABLE</i>	<i>MODIFIABLE</i>
valeur de la propriété d'édition d'une variable	<i>OBLIGATOIRE</i>	<i>MANDATORY</i>
annonce de contraintes sur les utilisations d'une caractéristique. Ces contraintes sont liées aux algorithmes de construction d'objets déduits. Cf. les différents rôles possibles dans les points suivants :	<i>ROLE</i>	<i>ROLE</i>
<ul style="list-style-type: none"> aucune contrainte sur l'utilisation d'une caractéristique. Sert à supprimer des contraintes héritées. 	<i>STANDARD</i>	<i>STANDARD</i>
<ul style="list-style-type: none"> contrainte sur l'utilisation d'une <i>CONSTANTE</i> : ne peut pas changer de valeur dans une variante. Ainsi on peut s'appuyer sur cette constante dans les algorithmes d'objets déduits. Par extension, c'est une contrainte que l'on doit ajouter aux paramètres intervenant dans les modèles de fiabilité, car le générateur d'arbres ne saurait pas gérer des variantes portant sur les valeurs de ces paramètres. 	<i>CONCEPTION</i>	<i>DESIGN</i>
<ul style="list-style-type: none"> désigne les interfaces permettant de savoir si un objet déduit peut être conservé (au cas 	<i>REFERENCE</i>	<i>REFERENCE</i>

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

où on en a modifié des caractéristiques) au lieu de le reconstruire.		
<ul style="list-style-type: none"> signale une caractéristique qui ne sert que lors de la construction des objets déduits, et n'a pas d'utilité dans le modèle Figaro final. 	<i>PROVISOIRE</i>	<i>TEMPORARY</i>
annonce d'une chaîne de caractères associée à un objet	DESCRIPTION	DESCRIPTION
annonce de la description du domaine d'une variable	<i>DOMAINE</i>	<i>DOMAIN</i>
annonce de l'expression de définition de la valeur initiale par défaut d'une variable	<i>PAR_DEFAULT</i>	<i>DEFAULT</i>
annonce de l'expression de description de la valeur de réinitialisation d'une variable	REINITIALISATION	REINITIALISATION
	VRAI	TRUE
	FAUX	FALSE
	ENTIER	INTEGER
	REEL	REAL
	BOOLEEN	BOOLEAN
variable donnant le temps courant (utilisable seulement en simulation de Monte Carlo)	DATE_COURANTE	CURRENT_DATE
annonce de modèles de fiabilité associés à une PANNE	<i>DONNEES_FIABILISTES</i>	<i>RELIABILITY_DATA</i>
annonce du modèle de fiabilité par défaut	<i>MODELE_PAR_DEFAULT</i>	<i>DEFAULT_MODEL</i>
annonce du modèle de fiabilité remplacé	<i>MODELE_REMPLACE</i>	<i>MODEL_REPLACED</i>
modèle de fiabilité : fréquence	<i>MODELE_F</i>	<i>MODEL_F</i>
modèle de fiabilité : valeur constante VRAI ou FAUX	<i>MODELE_FIGE</i>	<i>MODEL_FROZEN</i>
modèle de fiabilité : probabilité indépendante du temps	<i>MODELE_G</i>	<i>MODEL_G</i>
modèle de fiabilité : réparable, défaillances à la sollicitation et en fonctionnement	<i>MODELE_GLM</i>	<i>MODEL_GLM</i>
modèle de fiabilité : défaillance à la sollicitation et en fonctionnement, temps de mission fixé	<i>MODELE_GLTM</i>	<i>MODEL_GLTM</i>
modèle de fiabilité : testé aléatoirement	<i>MODELE_TA</i>	<i>MODEL_RT</i>
modèle de fiabilité : testé selon calendrier fixé	<i>MODELE_TPE</i>	<i>MODEL_PET</i>
modèle de fiabilité : non réparable, loi de Weibull	<i>MODELE_WB</i>	<i>MODEL_WB</i>
nom de paramètre de modèle de fiabilité associé à une PANNE	ALPHA	ALPHA
idem	<i>FREQUENCE</i>	<i>FREQUENCE</i>

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

idem	<i>GAMMA</i>	<i>GAMMA</i>
idem	<i>GAMMA_NON_DETECTI ON</i>	<i>GAMMA_NO_DETECTIO N</i>
idem	<i>GAMMA_RECONFIG</i>	<i>GAMMA_RECONFIG</i>
idem	<i>GAMMA_TEST</i>	<i>GAMMA_TEST</i>
idem	<i>LAMBDA</i>	<i>LAMBDA</i>
idem	<i>LAMBDA_TEST</i>	<i>LAMBDA_TEST</i>
idem	<i>MU</i>	<i>MU</i>
idem	<i>T_C_M</i>	<i>C_T_M</i>
idem	<i>T_INIT_TEST</i>	<i>T_INIT_TEST</i>
idem	<i>T_INTER_TEST</i>	<i>T_INTER_TEST</i>
idem	<i>T_TEST</i>	<i>T_TEST</i>
idem	<i>T0</i>	<i>T0</i>
idem	<i>TM</i>	<i>MT</i>

3.2.8. Systèmes d'équations linéaires

sens Figaro	terme Français	terme Anglais
annonce d'une liste d'équations	EQUATION	EQUATION
type des équations	LINEAIRE	LINEAR
annonce des systèmes auxquels est liée une équation	SYSTEME_EQUATIONS	EQUATION_SYSTEM
demande de résolution d'un système d'équations	RESOUDRE_SYSTEME	SOLVE_SYSTEM
annonce d'une équation	FORMULE	FORMULA

4. Syntaxe du langage Figaro

Ce chapitre détaille la syntaxe du langage, autrement dit la décomposition de l'arbre syntaxique en structures de plus en plus élémentaires jusqu'à arriver aux éléments lexicaux décrits dans le chapitre précédent. Les diagrammes syntaxiques utilisent les conventions données au début du document.

Remarque importante : depuis la dernière version de Figaro 1 l'utilisation de variables GLOBALES a été supprimée car elle était une source de bugs ; elle est remplacée par les objets système.

NB : afin de faciliter la maintenance de ce document, les « diagrammes syntaxiques » de ce chapitre sont donnés avec le vocabulaire et les mots clés anglais. Les mots clés Figaro sont écrits en gras pour faciliter leur repérage et les distinguer des éléments terminaux de la syntaxe qui sont aussi notés en majuscules (Ex : `LITERAL_STRING`). *Ce qui est en italiques est réservé à l'ordre 1.*

La racine de l'arbre syntaxique d'un texte Figaro est `Figaro_model`.

`Figaro_model` :

```
(KB_DESCRIPTION LITERAL_STRING ; | steps | groups | systems | class |
object) [Figaro_model]
```

Contraintes :

- Il peut y avoir au plus une description introduite par `KB_DESCRIPTION` dans une base de connaissances.
- Les structures `steps`, `groups`, `systems` doivent être déclarées au plus une fois.

4.1. Déclaration de la liste des étapes

```
steps : STEPS_ORDER liststeps
liststeps : list_ids [CONDITION term [label_string]]
            | LABEL label_string CONDITION term
            | [LABEL label_string] [liststeps] ;
            | STEP LITERAL_INTEGER list_ids [liststeps] ;
list_ids : ID [, list_ids]
label_string : LITERAL_STRING
```

Contraintes :

- Une étape ne peut apparaître qu'une fois dans la liste.

Remarque :

- Une règle d'interaction non associée explicitement à une étape est mise dans l'étape (toujours existante) appelée `etape_par_defaut` (`default_step` en anglais).

4.2. Déclaration de la liste des groupes

```
groups : GROUP_NAMES group_list
```

```
group_list: ID [LABEL label_string] ; [group_list]
```

Contraintes :

- Un groupe ne peut apparaître qu'une fois dans la liste.

4.3. Déclaration de la liste des systèmes d'équations

```
systems : SYSTEM_NAMES system_list
system_list : ID [LABEL label_string] ; [system_list]
```

Contraintes :

- Un système ne peut apparaître qu'une fois dans la liste.

4.4. Définition d'un type

```
class :
  CLASS ID [GR_LINK|GR_NODE] [KIND_OF enum_ids] ; fields

fields :
  (const | attribute | effect | failure | dist_param | interface |
  occurrence | interaction | equation | description) [fields]
```

Contraintes :

- Les mots-clés GR_LINK et GR_NODE servent uniquement pour indiquer à un outil de visualisation graphique (pendant son paramétrage) que les objets de ce type doivent être représentés sous la forme d'une icône ou d'un lien. Ils ne doivent normalement pas rester présents dans une base de connaissances finalisée.
- Un type ne doit être défini qu'une seule fois.
- Chaque type père ne doit apparaître qu'une seule fois dans la liste des pères annoncée par KIND_OF.

4.5. Définition d'un objet

```
objet : (OBJECT | SYSTEM_OBJECT)
      ID IS_A ID ; objfields
objfields : (description | const | attribute | effect
            | failure | dist_param | interface | occurrence
            | interaction | equation)
description : DESCRIPTION LITERAL_STRING ;
```

Contraintes :

- Un objet ne peut être défini qu'une seule fois.

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

- Un objet système est un objet spécial, créé d'emblée lors de l'initialisation et visible de tous les autres objets du système. Il peut donc être utilisé pour partager des valeurs. Un tel objet ne peut être déclaré que dans un fichier base de connaissances.

4.6. Définition des caractéristiques d'un type ou d'un objet

Ci-après, quelques structures communes à plusieurs caractéristiques.

`label : LABEL LITERAL_STRING`

`domain : DOMAIN (INTEGER | REAL | BOOLEAN | enum_quote_ids)`

`enum_quote_ids : QUOTID [enum_quote_ids]`

`default_term : DEFAULT term`

`affect_term : = term (réservé aux objets)`

`reinit_term : REINITIALISATION term`

`edition : EDITION terms_ed`

`terms_ed : (term_ed | NON term_ed`

`| terms_ed , term_ed | terms_ed , NON term_ed)`

`term_ed : (VISIBLE | MODIFIABLE | MANDATORY)`

`role : ROLE (STANDARD | CONCEPTION | REFERENCE | PROVISoire)`

Un rôle sert à définir des contraintes sur l'utilisation d'une caractéristique. Cf. tableau du 3.2.7 pour plus de détails.

Contraintes :

- Le domaine ne peut pas être défini plusieurs fois pour une même variable.
- Un domaine énuméré ne peut pas contenir deux fois la même valeur.
- La valeur par défaut et la valeur de réinitialisation (si elle existe) d'une variable sont identiques et ne peuvent être définies qu'une seule fois pour chaque variable.
- Le libelle ne peut être défini qu'une seule fois par caractéristique.

4.6.1. Définition des Variables

Contraintes :

- Une variable (CONSTANTE, EFFET, PANNE, ATTRIBUT, PARAMETRE_LOI) ne peut être définie qu'une seule fois dans chaque description de type.
- Une variable ne peut pas avoir le même nom qu'une interface.
- Pour chaque variable, un certain nombre de « facettes » (*slots* en anglais) ont un nom dans les diagrammes syntaxique commençant par f_. Chacune de ces facettes doit être définie au plus une fois.

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

- Dans un type, on peut seulement lui donner une valeur par défaut (`default_term`) ou de réinitialisation (`reinit_term`), alors que dans un objet, on peut seulement lui donner une valeur initiale (`affect_term`) ou de réinitialisation (`reinit_term`).

4.6.1.1. Attributs

```
attribute : ATTRIBUTE attributes
attributes : (ID f_attribute | error) ; [attributes]
f_attribute : (edition | role | domain | default_term
              | reinit_term | affect_term | label) [f_attribute]
```

NB : la “structure” error (qu’on verra en de multiples endroits) peut être ignorée, car elle sert au *parser* à pouvoir continuer l’analyse syntaxique même si une erreur est rencontrée dans la structure en cours d’analyse, délimitée par un « ; ». Ainsi l’utilisateur peut avoir plusieurs diagnostics d’erreur en un seul passage du vérificateur de syntaxe.

4.6.1.2. Constantes

```
const : CONSTANT constants
constants : ID f_constant ; [constants]
f_constant : (edition | role | domain | default_term | affect_term | label)
[f_constant]
```

4.6.1.3. Paramètres de loi

Il n’y a pas de différence de fond entre la déclaration d’un PARAMETRE_LOI (`DIST_PARAMETER`) et d’une CONSTANTE *réelle*. Il y a deux mots-clés différents seulement pour faciliter la compréhension d’une base de connaissances. Un PARAMETRE_LOI doit être utilisé uniquement, comme son nom l’indique, dans la définition des lois de probabilité, et ne peut être employé en Figaro 0.

```
dist_param : DIST_PARAMETER dist_params
dist_params : (ID f_dist_param ; | error) [dist_params]
f_dist_param : (edition | affect_term | role | default_term | label)
[f_dist_param]
```

4.6.1.4. Pannes

```
failure : FAILURE failures
failures : ID f_failure ; [failures]
f_failure : (edition | affect_term | role | default_term | label |
rel_data) [f_failure]
```

4.6.1.5. Effets

```
effect : EFFECT effects
effects : (ID f_effect | error) ; [effects]
f_effect : (edition | role | label) [f_effect]
```

4.6.2. Définition des Interfaces

```
interface : INTERFACE interfaces
```

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

```

interfaces : (ID f_interface | error) ; [interfaces]
f_interface : (edition | kind | cardinal | affect_ids | label | role)
[f_interface]
kind : KIND ID
cardinal : CARDINAL LITERAL_INTEGER
          [ TO (LITERAL_INTEGER | INFINITY) ]
affect_ids : = enum_ids
enum_ids : ID [enum_ids]

```

Contraintes :

- Une variable ne peut pas avoir le même nom qu'une interface.
- Le minimum du cardinal doit être inférieur ou égal au maximum.

4.6.3. Définition des Règles

Remarque : pour les deux types de règles (occurrence et interaction), on trouve les structures suivantes :

```

condition : IF term
group : GROUP enum_ids

```

Contrainte : ces deux structures ne doivent apparaître qu'une fois dans une règle.

4.6.3.1. Règles d'occurrence

```

occurrence : OCCURRENCE occurrences
occurrences : ([ID] f_occurrence MAY_OCCUR
               alternatives | error) ; [occurrences]
f_occurrence : (group | condition | label) [f_occurrence]
alternatives : [alternatives OR_ELSE] alternative
alternative : (TRANSITION ID f_transition
              | FAILURE ID f_failure
              | REPAIR ID f_reparation
              | UNAVAILABILITY ID f_unavail)

f_transition : (label | dist | induces) [f_failure]
f_failure : (label | dist | induces) [f_failure]
f_unavail : (label | dist | induces) [f_unavail]
f_repair : (label | dist | repairs | induces) [f_repair]
dist : DIST ID LPAR term RPAR
induces : INDUCING action
repairs : REPAIRS enum_ids

```

Contraintes :

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

- Une règle d'occurrence ne peut être définie qu'une seule fois par type.
- Une règle d'occurrence doit avoir au moins une transition.
- Une règle d'occurrence contenant plus d'une alternative (donc contenant le mot-clé OU_BIEN (OR_ELSE)) ne doit contenir que des lois de type INSTANTANE.
- Toutes les transitions d'une même règle doivent avoir un nom différent.
- La facette LOI (DIST) et la facette PROVOQUE (INDUCES) ne doivent pas être définies plus d'une seule fois pour chaque transition.
- La facette REPARE (REPAIRS) ne doit pas contenir deux fois la même réparation.
- La facette GROUPE (GROUP) doit contenir au moins un groupe défini dans la liste des groupes.

4.6.3.2. Règles d'interaction

```

interaction : INTERACTION interactions
interactions : ([ID] f_interaction | error) ; [interactions]
f_interaction : (step | group | condition | label | given | then | else)
[f_interaction]
step : STEP enum_ids
given : (GIVEN anobject | given AND anobject | given , GIVEN anobject)
then : THEN action
else : ELSE action
anobject : ID (A|AN) obj  [OF_TYPE ID] [VERIFYING term]

```

Contraintes :

- Une règle d'interaction ne peut être définie qu'une seule fois par type.
- Une règle d'interaction doit avoir au moins la facette ALORS (THEN).
- La facette ETAPE doit contenir au moins une étape définie dans la liste des étapes.
- La facette GROUPE doit contenir au moins un groupe défini dans la liste des groupes.

4.6.4. Définition des équations

```

equation : EQUATION equations
equations : [ID f_equation | error] ; [equations]
f_equation : (kind_eq | system_eq | condition
              | formula_eq | label) [f_equation]
kind_eq : KIND LINEAR
systeme_eq : EQUATION_SYSTEM ID
formula_eq : FORMULA term = term

```

Contraintes :

- Une équation ne peut être définie qu'une seule fois par type.
- La facette SYSTEME_EQUATIONS (EQUATION_SYSTEM) doit utiliser le nom d'un système défini dans la liste des systèmes.

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

4.6.5. Actions

```

action : (action_elem
  | action , action | LPAR action RPAR)
action_elem : assign
  | FOR_ALL anobject DO action
  | SOLVE_SYSTEM LPAR ID RPAR)
assign : (var <-- term | var | NOT var)

```

Contraintes :

- Le système annoncé dans la facette RESOUDRE_SYSTEME (SOLVE_SYSTEM) doit être présent dans la liste de systèmes.

4.6.6. Définition des termes des expressions

```

term : constante | var
  | term (> | >= | = | <> | < | <=) term
  | isworking | failed
  | term (AND | OR) term | NOT term
  | FOR_ANY anobject WE_HAVE term
  | IT_EXISTS anobject SUCH_THAT term
  | IT_EXISTS AT_LEAST term anobjectin SUCH_THAT term
  | AT_LEAST term WITHIN LPAR list_sep_terms RPAR
  | obj INCLUDED_IN obj
  | CARDINAL OF obj
  | CARDINAL LPAR obj RPAR
  | term( + | - | * | / | ** | % ) term
  | term ?= LPAR free_terms RPAR
  | LPAR error RPAR
  | LPAR term RPAR
  | - term
  | ?var (attention : pas de blanc après le "?")
  | SUM FOR_ALL anobject OF_TERMS term
  | PRODUCT FOR_ALL anobject OF_TERMS term
  | MAXIMUM FOR_ALL anobject OF_TERMS term
  | MINIMUM FOR_ALL anobject OF_TERMS term
  | RAND
  | CURRENT_DATE
  | INTEGRAL LPAR term RPAR
  | ALREADY_REALIZED LPAR term RPAR
  | STATE_TIME LPAR term RPAR

```

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

```

anobjectin : ID INCLUDED_IN obj [OF_TYPE ID] [VERIFYING term]
isworking : WORKING [LPAR obj RPAR | OF obj]
failed : FAILURE [LPAR obj RPAR | OF obj]
var : ID [LPAR var RPAR | LPAR list_terms RPAR | OF var]
list_terms : term [, list_terms]
list_sep_terms : term [, list_sep_terms]
constant : (LITERAL_INTEGER | LITERAL_REAL | TRUE | FALSE | QUOTID)
obj : ID (LPAR obj RPAR | OF obj) | MYSELF

```

Contraintes :

- Les mots clés RAND, CURRENT_DATE, INTEGRAL, ALREADY_REALIZED et STATE_TIME ne peuvent être utilisés que dans des modèles destinés à être exploités par simulation de Monte Carlo.

4.6.7. Définition d'un modèle fiabiliste et des données associées

Cette structure sert à définir un ou plusieurs modèles de feuilles associés à une PANNE en vue de générer des arbres de défaillances.

On ne peut déclarer des modèles fiabilistes qu'en Figaro d'ordre 1.

Contraintes

- Plusieurs modèles peuvent être définis, mais un seul peut être déclaré comme modèle par défaut.

```
rel_data : RELIABILITY_DATA rel_elements
```

```

rel_elements :
    | rel_elements condition
    | rel_elements group
    | rel_elements DEFAULT_MODEL model_type
    | rel_elements replaces_model
    | rel_elements rel_model
    | rel_elements error

```

```
replaces_model : MODEL_REPLACED | REPLACES term
```

```
rel_model : model_type rel_parameters
```

```
rel_parameters : | rel_parameter [rel_parameters]
```

```
rel_parameter : ID = term
```

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

model_type :

MODEL_GLM | MODEL_FROZEN | MODEL_GLTM | MODEL_G | MODEL_F | MODEL_RT |
MODEL_PET | MODEL_WB

Rappel des noms en français :

MODELE_GLM | MODELE_FIGE | MODELE_GLTM | MODELE_G | MODELE_F |
MODELE_TA | MODELE_TPE | MODELE_WB

5. Contraintes grammaticales du langage Figaro 1

Dans les chapitres précédents, des contraintes simples car locales aux structures syntaxiques décrites ont déjà été énoncées. Le respect de ces contraintes ne suffit pas à assurer la cohérence d'ensemble d'un modèle Figaro.

Le présent chapitre a pour but de décrire les contraintes plus globales qui existent aussi. Le respect de ces contraintes est vérifié par les outils Figaro. Leur non-respect donne lieu, selon le cas, à des avertissements ou à des erreurs.

5.1. Contraintes grammaticales liées à l'héritage

Les contraintes grammaticales liées à l'héritage exigent en premier lieu l'existence d'au moins un type dans la base de connaissances.

Un type est décrit par ses caractéristiques statiques (variables et interfaces) et dynamiques (règles d'occurrence et d'interaction, équations). Chaque caractéristique des types pères d'un type est copiée au niveau même du type afin de réaliser la surcharge des facettes.

Le mécanisme de surcharge est mis en œuvre par l'intermédiaire des deux principes suivants :

- L'héritage des caractéristiques se fait dans l'ordre `SORTE_DE`, par noms (si deux variables ont le même nom, il y a surcharge) et par facettes (seules les facettes surchargées sont modifiées, les autres sont conservées).
- “*Le dernier qui parle a raison*” : la surcharge peut s'effectuer entre types hérités, dans ce cas, c'est l'ordre des noms précisés dans la facette `SORTE_DE` qui détermine l'ordre de prise en compte des caractéristiques et, par-là, celui de leurs facettes. Ainsi, dans l'exemple suivant :

```
TYPE A ;
ATTRIBUT poids DOMAINE reel PAR_DEFAULT 14 ;
TYPE B ;
ATTRIBUT poids DOMAINE reel PAR_DEFAULT 18 ;
TYPE C SORTE_DE B A ;
```

La valeur par défaut de poids dans C est 14.

La résolution de l'héritage exige les contraintes grammaticales suivantes sur les surcharges par noms et par facettes et sur le graphe d'héritage entre types :

- Les surcharges par noms :

- ◇ Un même nom ne peut être utilisé pour une variable dans un type père et pour une variable de type de champ différent dans un type héritant.
- ◇ Un type ne peut hériter de deux variables, de types de champ différents et de même nom, provenant de deux types différents.
- ◇ Il est recommandé d'utiliser des noms différents pour une règle d'interaction définie dans un type père et une règle d'occurrence définie dans le type héritant du type père et inversement.
- ◇ Il est recommandé de ne pas utiliser un nom de transition héritée pour définir une nouvelle transition dans un type héritant.
- Les surcharges par facettes :
 - ◇ Les domaines des variables héritées ne doivent pas être redéfinis dans le type héritant à l'exception des domaines énumérés.
 - ◇ Le genre déclaré pour une INTERFACE doit être un sous-type du type déclaré comme GENRE dans l'interface de même nom du type père.
- Il est interdit de créer une boucle dans les liens d'héritage.

5.2. Contraintes grammaticales liées aux interfaces

Les contraintes grammaticales liées aux interfaces sont les suivantes :

- Le type référencé par la facette GENRE d'une interface doit être défini dans la base de connaissances.
- Il est recommandé de ne pas utiliser le même nom pour une interface et un type.

5.3. Contraintes grammaticales liées aux expressions

Les contraintes grammaticales liées aux expressions sont les suivantes :

- Pour une variable :
 - ◇ Les valeurs par défaut des constantes CONSTANTE et PARAMETRE_LOI doivent être des expressions constantes, c'est-à-dire des valeurs numériques ou littérales faisant intervenir uniquement d'autres constantes. Il ne doit pas y avoir de définition circulaire entre constantes.
 - ◇ Les expressions des valeurs par défaut doivent être compatibles avec les domaines des variables. Le tableau ci-après donne les combinaisons autorisées.
 - ◇ Les valeurs de réinitialisation des variables ATTRIBUT doivent être des expressions constantes, c'est-à-dire des valeurs numériques ou littérales faisant intervenir uniquement des constantes.
 - ◇ Les expressions des valeurs de réinitialisation doivent être compatibles avec les domaines des variables. Le tableau ci-après donne les combinaisons autorisées.
 - ◇ Les opérateurs PANNE et MARCHE doivent être appliqués sur des objets possédant des variables PANNE.
 - ◇ Le tableau suivant donne les règles de « promotion » possibles lors de la définition de valeurs par défaut ou de valeurs de réinitialisation.

Domaine de la variable	Domaine de l'expression	Autorisé
BOOLEEN	BOOLEEN	Oui
BOOLEEN	ENTIER	Non
BOOLEEN	REEL	Non
BOOLEEN	Enuméré	Non
ENTIER	BOOLEEN	Oui (pris égal à 0 ou 1)
ENTIER	ENTIER	Oui
ENTIER	REEL	Non
ENTIER	Enuméré	Non
REEL	BOOLEEN	Oui (pris égal à 0 ou 1)
REEL	ENTIER	Oui
REEL	REEL	Oui
REEL	Enuméré	Non
Enuméré	BOOLEEN	Non
Enuméré	ENTIER	Non
Enuméré	REEL	Non
Enuméré	Enuméré	<i>Oui, si et seulement si une valeur est commune aux 2 domaines</i>

- Plus généralement, au sein d'une expression, un booléen est promu en entier (0 ou 1) lorsqu'il est l'un des opérandes d'une opération numérique dont l'autre opérande est un nombre. Le résultat d'une opération entre un entier et un réel est un réel.
- Pour une équation :
 - ◇ Les inconnues d'une équation doivent être des attributs réels.
 - ◇ Une équation a un genre qui ne peut être que LINEAIRE :
 - * L'opérateur * ne doit avoir qu'un seul de ses opérandes contenant une inconnue, l'autre ne doit pas en contenir.
 - * L'opérateur / ne doit pas contenir d'inconnue dans son opérande droit.
 - * L'opérateur PRODUIT dispose de deux opérandes, le premier déclare une variable liée et le second est une expression. Cette dernière expression ne doit pas contenir d'inconnue sauf si le cardinal de l'ensemble sur lequel la variable liée est définie est 0 ou 1.
 - * L'opérateur SOMME dispose de deux opérandes, le premier déclare une variable liée et le second est une expression. Cette dernière expression doit respecter les règles annoncées sur les opérateurs * et /.
 - ◇ La facette SI utilisée pour décrire la condition d'application d'une équation doit porter sur une expression booléenne.
- Pour une règle en général :
 - ◇ Si au moins une règle de la base de connaissances appartient à un groupe, il est recommandé que toutes les règles de la base appartiennent à au moins un groupe .

- Pour une règle d'interaction :
 - ◇ Si au moins une règle d'interaction appartient à une étape, il est recommandé que toutes les règles d'interaction appartiennent à une étape.
- Pour une règle d'occurrence :
 - ◇ Seule la dernière transition d'une règle instantanée peut être dispensée de la facette LOI. Dans tous les autres cas, la facette LOI est obligatoire.
 - ◇ Des transitions correspondant à des types de lois différents ne peuvent appartenir à une même règle d'occurrence.
 - ◇ Le nombre de transitions définies dans une règle d'occurrence doit être au moins égal à un pour les lois instantanées et égal à un pour toutes les autres lois.
 - ◇ Pour les actions en conclusion de règles (une transition sans action explicite ni action implicite est une erreur) :
 - * Si la transition n'est ni une défaillance, ni une réparation, ni une indisponibilité, la facette PROVOQUE est obligatoire.
 - * Si la transition est une défaillance et si son nom ne correspond pas à une variable PANNE, la facette PROVOQUE est obligatoire.
 - * Si la transition est une réparation :
 - ÷ Si la facette REPARE existe, celle-ci doit être appliquée à une variable PANNE.
 - ÷ Si la facette REPARE n'est pas précisée, la facette PROVOQUE est obligatoire.
 - * Si la transition est une indisponibilité,
 - ÷ Si le nom de la transition ne correspond pas à une variable PANNE, la facette PROVOQUE est obligatoire.
 - ÷ La transition doit être instantanée.
- Pour une expression de façon générale :
 - ◇ La facette VERIFIANT doit être une expression ne faisant intervenir que des constantes de CONCEPTION.
 - ◇ Les deux opérandes d'un opérateur binaire doivent avoir des domaines compatibles. Le tableau suivant donne pour divers opérateurs le domaine de l'expression, le nombre et le type des opérandes attendus.

Opérateur	Domaine expression	Premier opérande	Deuxième opérande
DE_TYPE	Objet	Expression Sélection	Nom type
VERIFIANT	Objet	Expression Sélection	Expression booléenne
ET	Booléen	Expression booléenne	Expression booléenne
OU	Booléen	Expression booléenne	Expression booléenne
NON	Booléen	Expression booléenne	

EDF R&D	Syntaxe du langage de modélisation probabiliste Figaro	6125-1612-2016-15549-FR Version 1.0
---------	--	--

Opérateur	Domaine expression	Premier opérande	Deuxième opérande
INCLUS_DANS	Booléen	Expression Accès à ensemble de cardinal 1	Expression Accès à ensemble
CARDINAL	Entier	Expression Accès à ensemble	
PANNE	Booléen	Expression Accès à ensemble de cardinal 1	
MARCHE	Booléen	Expression Accès à ensemble de cardinal 1	
QQSOIT... ON_A...	Booléen	Expression Création alias	Expression booléenne
IL_EXISTE... TEL_QUE...	Booléen	Expression Création alias	Expression booléenne

6. Annexe : Liste des Fonctions

L'application d'une fonction de nom ID à un ensemble d'arguments est représentée dans la grammaire par ID puis " (" liste de termes séparés par des virgules ")". Ceci est accepté de manière générale de façon à éviter de changer les fichiers Lex et Yacc à chaque ajout d'une fonction. Lorsqu'il rencontre ce type de construction, le *parser* Figaro cherche dans une liste d'opérateurs si la fonction existe.

Les fonctions ont le même nom en français et en anglais, sauf les fonctions spéciales listées au § 6.3.

6.1. Fonctions mathématiques

Dans le tableau ci-dessous, a désigne un entier, r désigne un nombre réel et x, y et z peuvent être entiers ou réels.

Nom de la fonction	Syntaxe
Exponentielle	EXP(r)
Sinus	SIN(r)
Cosinus	COS(r)
Tangente	TAN(r)
Racine carrée	SQRT(r)
Logarithme népérien	LN(r)
Logarithme en base 10	LOG(r)
Valeur absolue	ABS(r)
Entier supérieur	CEIL(r)
Entier inférieur	FLOOR(r)
Nombre entier le plus proche	NINT(r)
Factorielle (n!)	FACT(a)
Logarithme en base 10 de la factorielle (log(n!))	LFACT(a)
Maximum	MAX(x, y, z,...)
Minimum	MIN(x, y, z,...)

6.2. Fonctions retournant un nombre aléatoire suivant une loi donnée

Toutes les fonctions citées dans cette section ne peuvent être utilisées que dans le cadre d'une simulation de Monte Carlo.

RAND_EXP, RAND_POINT, RAND_UNI, RAND_TRIANG, RAND_ERL, RAND_WEI,
RAND_GUMBEL, RAND_FRECHET, RAND_PARETO, RAND_NORMAL, RAND_LGN,
RAND_GAMMA, RAND_BETA.

La définition des lois et leurs paramètres sont les mêmes que ce que l'on trouve dans la définition des délais aléatoires de tir pour les transitions temporisées. Le tableau ci-dessous donne ces définitions.

Remarque : on ne trouve pas RAND_CYCLE dans la liste ci-dessus, car CYCLE a une signification bien spécifique. Ce mot clé est utilisé comme « loi » associée à une transition pour modéliser que cette transition se déclenche à intervalles réguliers, de manière déterministe.

Nom	Nombre d'arguments	Type des arguments	Commentaire
EXP(λ)	1	réel	Loi exponentielle λ est le taux d'occurrence
CYCLE(T,T0)	2	réels	Loi pour un événement cyclique T est la période du cycle T0 est l'instant du premier événement
POINT(T1,F1,...,Tn,Fn)	2*n	réels	Loi définie point par point par sa fonction de répartition $0 \leq T_n \leq T_{n+1}$ $0 \leq F_n \leq F_{n+1} \leq 1$
UNI(A,B)	2	réels	Loi uniforme sur [A,B] ($A \leq B$)
TRIANG(A,M,B)	3	réels	Loi triangulaire sur [A,B] de sommet d'abscisse M ($A \leq M \leq B$)
ERL($\lambda_1, K_1, \dots, \lambda_n, K_n$)	2*n	arguments impairs réels, arguments pairs entiers	Loi d'Erlang généralisée
WEI(S,T,A, β)	4	premier argument entier, les autres arguments réels	Loi de Weibull S = 1 : A moyenne, S = 2 : A médiane, S = 3 : A échelle, T décalage ($T \geq 0$), β paramètre de forme ($\beta > 0$)
GUMBEL(m, σ)	2	réels	Loi de Gumbel m : moyenne ($m > 0$) σ : écart-type ($\sigma > 0$)

Nom	Nombre d'arguments	Type des arguments	Commentaire
FRECHET(T,α,β)	3	réels	Loi de Fréchet T : décalage (T≥0) α : échelle (α>0) β : paramètre de forme (β>0)
PARETO(T,σ,β,δ)	4	réels	Loi de Pareto T : décalage (T≥0) σ : échelle (σ>0) β : paramètre de forme (β>0) δ : paramètre de forme inverse (δ>0)
NORMAL(m,σ)	2	réels	Loi normale m : moyenne σ : écart-type (σ>0)
LGN(S,A,B)	3	premier argument entier, les autres arguments réels	Loi log normale S = 1 : A moyenne, B écart-type S = 2 : A médiane, B facteur d'erreur S = 3 : A moyenne, B facteur d'erreur
GAMMA(S,T,A,B)	4	premier argument entier, les autres arguments réels	Loi gamma S = 1 : A moyenne, B écart-type S = 2 : A échelle, B paramètre de forme
BETA(p,q,a,b)	4	réels	Loi beta de type I de paramètres p,q a : minimum b : maximum a≤b

6.3. Fonctions spéciales

Toutes les fonctions citées dans cette section ne peuvent être utilisées que dans le cadre d'une simulation de Monte Carlo.

Dans le tableau ci-dessous, b désigne une expression booléenne et x une expression réelle.

sens Figaro	terme Français	terme Anglais
Variable booléenne indicatrice du fait que l'argument b est passé à VRAI à DATE_COURANTE ou plus tôt	DEJA_REALISE(b)	ALREADY_REALIZED(b)
Calcul de l'intégrale de 0 à DATE_COURANTE de l'expression x en argument.	INTEGRALE(x)	INTEGRAL(x)
Cumul des temps passés dans l'état défini par l'expression indicatrice b, entre 0 et DATE_COURANTE.	TEMPS_SEJOUR(b)	STATE_TIME(b)