



**HAL**  
open science

# SYSTEM-ANALYST -UN OUTIL MBSA POUR L'ANALYSE DES RISQUES, LIBRE DE DIFFUSION ET COMPATIBLE AVEC ARBRE-ANALYSTE ET OPEN- ALTARICA

Sylvain Breton, Pierre Le-Com, Thierry Thomas, Emmanuel Clement

## ► To cite this version:

Sylvain Breton, Pierre Le-Com, Thierry Thomas, Emmanuel Clement. SYSTEM-ANALYST -UN OUTIL MBSA POUR L'ANALYSE DES RISQUES, LIBRE DE DIFFUSION ET COMPATIBLE AVEC ARBRE-ANALYSTE ET OPEN- ALTARICA. Congrès Lambda Mu 21, “ Maîtrise des risques et transformation numérique : opportunités et menaces ”, Oct 2018, Reims, France. hal-02063690

**HAL Id: hal-02063690**

**<https://hal.science/hal-02063690>**

Submitted on 29 Mar 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SYSTEM-ANALYST – UN OUTIL MBSA POUR L'ANALYSE DES RISQUES, LIBRE DE DIFFUSION ET COMPATIBLE AVEC ARBRE-ANALYSTE ET OPEN-ALTARICA

## SYSTEM-ANALYST – A NEW MBSA TOOL DEDICATED TO RAMS ANALYSES, FREE OF USE AND COMPATIBLE WITH ARBRE-ANALYSTE AND OPEN-ALTARICA

Sylvain BRETON, Pierre Le-COM et Thierry THOMAS.  
SOM, Ligeron  
23 Rue Nicéphore Niepce  
ZA de Loscoat 29200 Brest

Emmanuel CLEMENT  
Thales Defense Mission Systems  
10 Avenue 1ère Dfl  
29200 Brest

### Résumé

Nous présentons un nouvel outil, libre de diffusion et libre d'utilisation, dédié à l'analyse des risques. Cet outil, basé sur le formalisme MBSA, permet de modéliser un système complexe puis de mener un ensemble d'analyses de performances de Sûreté de fonctionnement tout en assurant une parfaite interopérabilité.

L'outil permet, à partir d'un modèle unique, de réaliser l'analyse fonctionnelle, l'analyse des modes de défaillances de leurs effets (AMDEC), l'analyse par arbres de défaillances (IEC 61025 "Fault tree analysis), l'analyse de testabilité ainsi que diverses analyses par simulations stochastiques. Chaque modification du modèle est ainsi automatiquement propagée dans toutes les analyses entraînant donc une très forte cohérence.

### Summary

We are presenting a new tool, free of use and community driven, dedicated to RAMS analyses. System-Analyst is a Model-Based Safety Assessment tool which aims to assist RAMS analysts.

System-Analyst aims to carry out, from an unique model, these following analyses: functional analysis, failure mode and effects analysis (FMEA), Fault tree analysis, testability analysis and various stochastic simulations. Each modification of the model will be automatically spread in every analysis, ensuring a great coherence.

## 1 - Introduction

### 1.1 Contexte

La complexité croissante des systèmes entraîne, notamment en Sûreté de fonctionnement, la nécessité de progresser dans les méthodes d'analyses en développant l'ingénierie dirigée par les modèles (MBSA) (Riera et al.,2012), (Prosvirnova et al.,2013). Le MBSA (Model-Based Safety Assessment) permet, par la réalisation de modèles de plus haut niveau, c'est-à-dire proches des architectures fonctionnelles et physiques des systèmes, d'assurer le partage des hypothèses, des données d'entrée et des résultats des analyses entre les différents acteurs du projet. Cela permet une plus grande cohérence, une meilleure validation et une capitalisation des modèles et des différentes analyses. C'est dans ce cadre qu'intervient cette communication.

### 1.2 Objectif

L'objectif de cette publication est de présenter un nouvel outil informatique dont le but est de standardiser et de pérenniser les modélisations de systèmes complexes en s'appuyant sur deux piliers : un format d'enregistrement des projets libre et documenté d'une part, et une interopérabilité forte avec les formats documentés Open-PSA, Open-Altarica et les moteurs de calculs associés.

System-Analyst propose à la communauté de profiter d'un nouvel outil libre de diffusion et libre d'utilisation couvrant les aspects MBSA (Prosvirnova et al.,2013).

### 1.3 Problématique

Les systèmes industriels, faisant l'objet d'analyses de sûreté de fonctionnement, font intervenir des acteurs de plus en plus nombreux et leur conception s'étale sur plusieurs dizaines d'années. Ce constat montre l'importance

du choix des outils informatiques. Ceux-ci doivent permettre une abstraction forte du système à analyser tout en assurant une parfaite interopérabilité tant avec les différents acteurs que dans la durée.

### 1.4 Approche

Nous présentons, d'un point de vue industriel, un nouvel outil graphique de modélisation de systèmes complexes permettant la réalisation d'analyses de Sûreté de fonctionnement. Les analyses menées sont les suivantes

- Analyse fonctionnelle;
- Analyse des modes de défaillances, de leurs effets et de leur criticité (AMDEC), (IEC 60812);
- Analyse des taux de couverture de tests (identification des modes de défaillances dangereux et non détectés);
- Analyse par arbres de défaillances d'événements redoutés (IEC 61025), (Prosvirnova et Rauzy.,2013) ;
- Simulations pas-à-pas et simulations stochastiques.

Les analyses par arbres de défaillances se font très simplement par une compatibilité forte avec l'outil Arbre-Analyste (Clement et al.,2014). Les simulations pas-à-pas et stochastiques, du modèle saisi dans System-Analyst, sont réalisées par l'intermédiaire de l'outil OpenAltarica (Prosvirnova et al.,2013).

Nous présentons, d'un point de vue industriel, un nouvel outil de modélisation MBSA entièrement graphique dont l'objectif est de soumettre à la communauté une nouvelle méthode de modélisation des systèmes dans le cadre de la réalisation d'études de sûreté de fonctionnement.

Notre publication est articulée de la façon suivante :

1. Présentation de System-Analyst ;
2. Principe de modélisation avec System-Analyst ;
3. Exportation de l'AMDE ;
4. Exportation de l'arbre de défaillances ;
5. Exportation du modèle Altarica 3.0.

Nous terminerons la publication par une conclusion qui présentera, de façon synthétique, les apports de ce nouvel outil ainsi que les prochaines étapes de son développement.

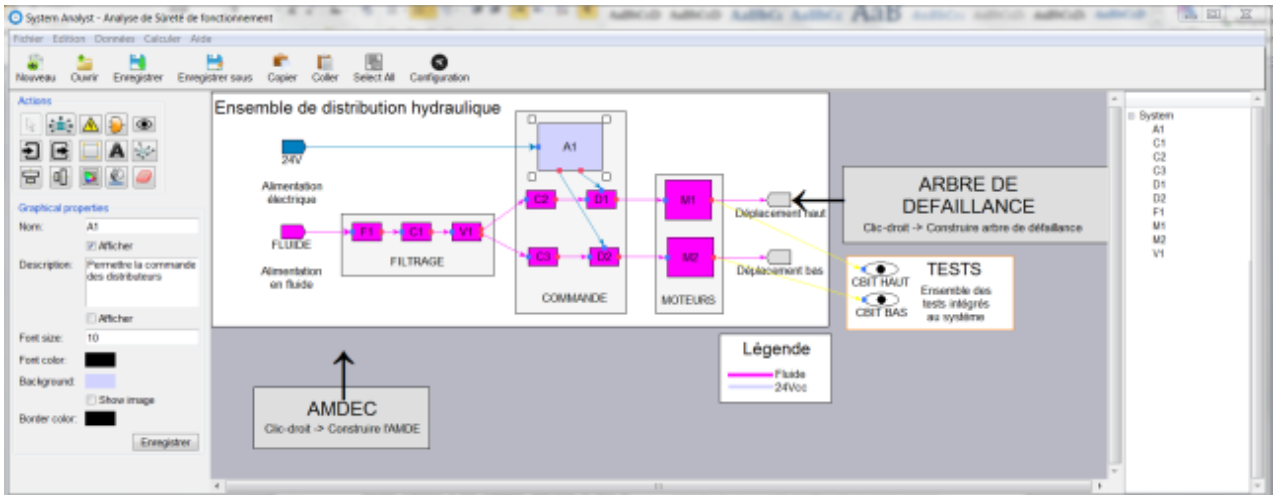


Figure 1 -Interface de System-Analyst

## 2 - System-Analyst

### 2.1 Présentation de System-Analyst

System-Analyst se base sur quatre principes.

Premièrement, la modélisation du système dans l'outil est très proche des architectures fonctionnelles et physiques, ce qui améliore les échanges avec les ingénieries systèmes et matérielles et ainsi facilite la validation du modèle.

Deuxièmement, le principe de modélisation est très fortement orienté « dépendance ». De ce fait, un ensemble d'algorithmes permettent de réaliser automatiquement des analyses : taux de couverture de tests, propagation des pannes au niveau système, génération d'arbres de défaillances, contrôles de cohérence du modèle etc...

Troisièmement, l'interopérabilité est au cœur du logiciel. System-Analyst est un outil libre d'utilisation et libre de diffusion et tous les formats de fichiers utilisés sont ouverts et documentés. Ces deux aspects sont primordiaux pour la réalisation d'analyses sur des systèmes industriels dont le nombre d'acteurs à être impliqués augmente et dont la durée des programmes court sur plus d'une décennie.

Quatrièmement, System-Analyst complète une suite d'outils libres d'utilisation déjà existants : XFTA, Arbre-Analyste et OpenAltarica.

### 2.2 Principe de modélisation avec System-Analyst

#### 2.2.1 Interface

L'interface de System-Analyst permet de réaliser toutes les étapes de construction des modèles graphiquement. La saisie du modèle est menée par l'intermédiaire de fenêtres et de champs à renseigner guidant ainsi l'utilisateur.

#### 2.2.2 Concepts

Le système est décrit grâce à l'utilisation de blocs (représentant une fonction ou bien un élément matériel) des entrées/sorties, des flux reliant les entrées/sorties, des portes logiques (ET, OU, K/N), ainsi que des événements représentant les aspects comportementaux et dysfonctionnels.

De plus, l'architecture du système est représentée par l'intermédiaire d'une structure hiérarchisée par niveau permettant une clarification et une décomposition analytique de la complexité du système.

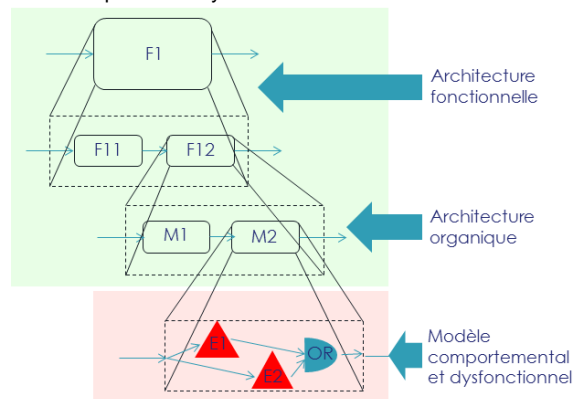


Figure 2 - Structure hiérarchisée par niveau

### 2.2.3 Gestion des dépendances

Les dépendances fonctionnelles et matérielles sont représentées par l'intermédiaire de ports (entrées/sorties) et par des flux reliant les ports entre eux.

Pour le moment, System-Analyst supporte les flux unidirectionnels et les ports de type « entrée » et « sortie ». Le support des flux bidirectionnels est en cours d'implémentation.

La gestion des dépendances sous System-Analyst est transparente et fonctionne sur le même principe que les diagrammes blocs de fiabilité.

La figure 3 représente une redondance entre les blocs fonctionnels « C1 » et « C2 ».

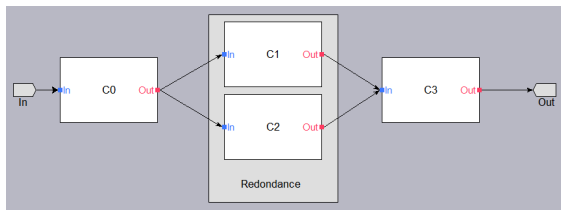


Figure 3 - C1 et C2 sont redondants

### 2.2.4 Description comportementale et dysfonctionnelle

La description comportementale et dysfonctionnelle est réalisée à l'intérieur des blocs. La figure 4 représente une modélisation d'un distributeur hydraulique.

Les hexagones rouges représentent les modes de défaillances. Plusieurs types sont proposés : perte de la fonction, fonctionnement intempestif, perte d'une sécurité, fonctionnement dégradée etc... Les modes de défaillances interviennent sur le flux auquel ils sont raccordés.

L'ovale bleu représente une porte logique. Plusieurs types de portes logiques sont disponibles : « ou », « et » et « k/n ».

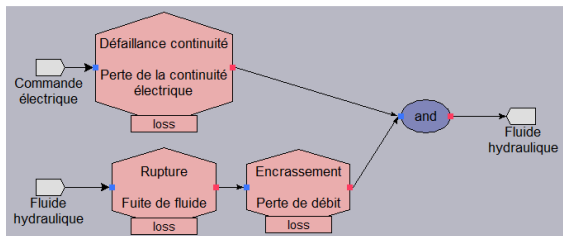


Figure 5 - Modélisation d'un distributeur hydraulique

### 2.2.5 Testabilité

La testabilité est l'aptitude d'un système à rendre compte de son état fonctionnel. Deux performances sont habituellement employées pour mesurer le niveau de testabilité :

1. Le taux de couverture  
Le taux de couverture correspond au ratio entre les pannes couvertes par au moins un test et l'ensemble des pannes du système ayant un impact opérationnel.
2. Le taux de localisation  
Le taux de localisation correspond à la faculté d'identifier l'élément interchangeable, source de la panne.

System-Analyst permet de renseigner les tests graphiquement dans le modèle, en sélectionnant le « flux » surveillé.

La figure 5 montre un test de présence de tension.

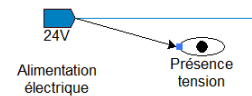


Figure 6 - Description d'un test

Un algorithme permet, grâce au modèle de dépendances, d'identifier tous les modes de défaillances couverts par le test.

Il est prévu de compléter la description des tests en ajoutant :

1. Une probabilité de défaillance du test ;
2. Une probabilité de détection intempestive (fausse alarme).

### 2.2.6 Simulation pas-à-pas

Un simulateur « pas-à-pas » est disponible directement depuis l'interface graphique. Ce simulateur permet de jouer manuellement des événements sur le modèle afin d'observer les évolutions et l'impact de ces événements sur l'état du modèle. Pour chacun des actions, le modèle est mis à jour en temps réel afin de propager les effets de l'action et ainsi modifier l'état des observateurs impactés.

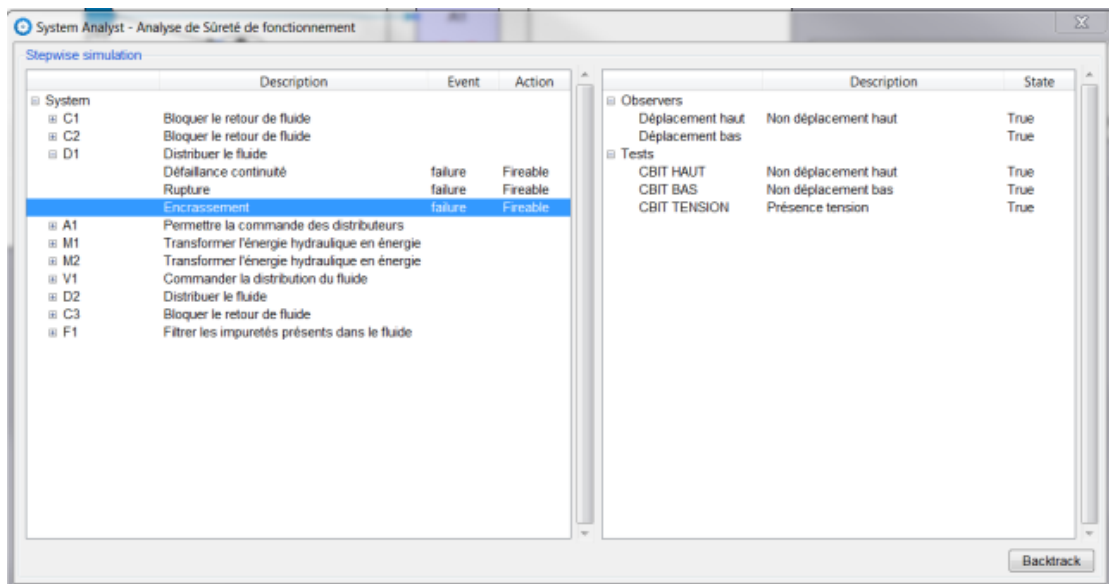


Figure 4 - Simulateur "pas-à-pas" de System-Analyst

### 3 - Exportations

#### 3.1 Exportation de l'AMDE

L'analyse des modes de défaillances et de leurs effets (IEC 60812), est une analyse exhaustive et systématique de chaque mode de défaillance de chaque constituant du système. Elle prend la forme d'un grand tableau.

Elle permet, pour chaque mode de défaillance, de faire apparaître :

- ses causes ;
- sa fréquence ou bien sa probabilité ;
- ses effets ;
- les mesures mises en place pour détecter la défaillance.

System-Analyst comprend des algorithmes permettant de générer directement l'analyse des modes de défaillances et de leurs effets directement à partir du modèle saisi. Ainsi, les modes de défaillance sont propagés dans le modèle de dépendances afin d'identifier tous les effets (locaux et finaux) ainsi que leur capacité à être détectés par un ou plusieurs tests.

L'AMDE ainsi générée est directement accessible dans un logiciel bureautique de type tableur.

La figure 7 est l'AMDE correspondant au modèle « Système hydraulique » présenté précédemment.

#	Niveau	Nom	Fonction	Quantité	Taux de défaillance	Mode de défaillance	Cause de défaillance	Effet local	Effet final	Testabilité
2	1	System		1						
3	2	A1	Permettre la commande des distributeurs	1	1,00E-05	Défaillance électrique	Choc mécanique Choc électrique Défaillance intrinsèque	Perte de l'armoire de commande	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT
4	2	C1	Bloquer le retour de fluide	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT
5					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT
6	2	C2	Bloquer le retour de fluide	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement haut de System	CBIT HAUT
7					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement haut de System	CBIT HAUT
8	2	C3	Bloquer le retour de fluide	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement bas de System	CBIT BAS
9					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement bas de System	CBIT BAS
10	2	D1	Distribuer le fluide	1	1,00E-05	Défaillance continuité	Choc mécanique Rupture Corrosion	Perte de la continuité électrique	Perte Déplacement haut de System	CBIT HAUT
11					2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement haut de System	CBIT HAUT
12					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement haut de System	CBIT HAUT
13	2	D2	Distribuer le fluide	1	1,00E-05	Défaillance continuité	Choc mécanique Rupture Corrosion	Perte de la continuité électrique	Perte Déplacement bas de System	CBIT BAS
14					2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement bas de System	CBIT BAS
15					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement bas de System	CBIT BAS
16	2	F1	Filtrer les impuretés présents dans le fluide	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT
17					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT
18	2	M1	Transformer l'énergie hydraulique en énergie mécanique	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement haut de System	CBIT HAUT
19					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement haut de System	CBIT HAUT
20	2	M2	Transformer l'énergie hydraulique en énergie mécanique	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement bas de System	CBIT BAS
21					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement bas de System	CBIT BAS
22	2	V1	Commander la distribution du fluide	1	2,00E-05	Encrassement	Présence d'impuretés	Perte de débit	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT
23					1,00E-05	Rupture	Choc mécanique	Fuite de fluide	Perte Déplacement bas de System Perte Déplacement haut de System	CBIT BAS CBIT HAUT

Figure 7 - AMDE du système hydraulique générée par System Analyst

Il est prévu d'améliorer le support des AMDE dans System-Analyst en ajoutant les aspects « criticité ». Pour cela, les éléments suivants seront proposés pour compléter l'AMDE :

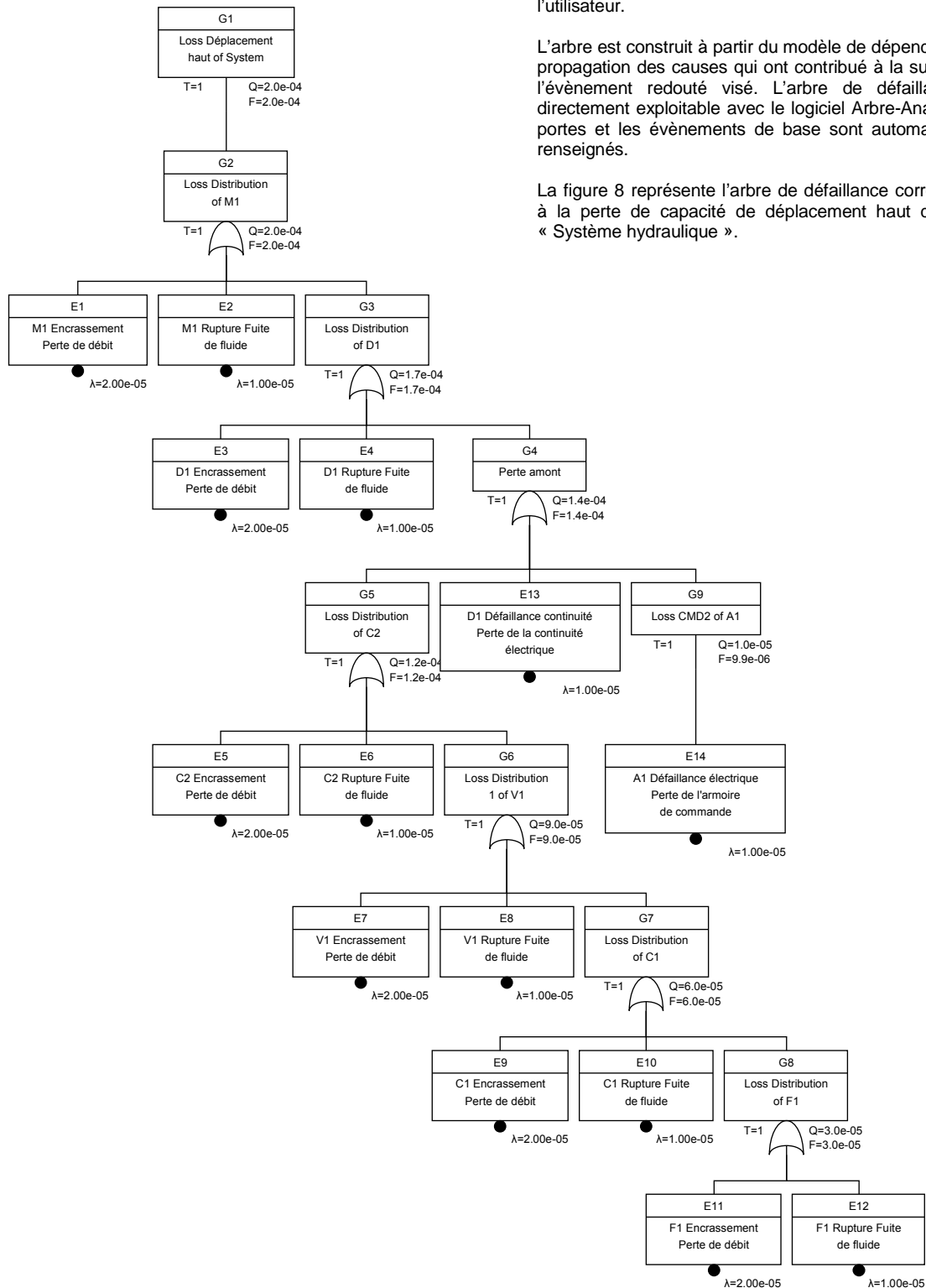
- des classes de gravité ;
- des classes de probabilité ;
- des matrices de criticité ;
- une gestion des compensations pour les éléments critiques.

Par défaut, les classes de gravité et de probabilité ainsi que les matrices seront issues des normes. Il sera aussi possible de saisir ses propres définitions des classes et des matrices.

### 3.2 Exportation de l'arbre de défaillances

#### 3.2.1 Généralité

Un arbre de défaillances (IEC 61025), (aussi appelé arbre de pannes ou arbre de fautes) consiste à représenter graphiquement les combinaisons possibles d'événements qui permettent la réalisation d'un événement indésirable prédéfini. Une telle représentation graphique met donc en évidence les relations de cause à effet. Cette technique est complétée par un traitement mathématique qui permet d'identifier la combinaison de défaillances simples ainsi que de calculer leur probabilité d'apparition. Elle permet ainsi de quantifier la probabilité d'occurrence d'un événement indésirable, également appelé « événement redouté ».



#### 3.2.2 Arbre-Analyste

System-Analyst est compatible avec le logiciel Arbre-Analyste]. Arbre-Analyste est un outil d'arbre de défaillances libre de diffusion et libre d'utilisation (Clement et al.,2014) : [www.arbre-analyste.fr/](http://www.arbre-analyste.fr/)

#### 3.2.3 Exportation depuis System-Analyst

System-Analyst comprend un algorithme permettant de générer directement l'arbre de défaillances correspondant à la survenue d'un événement redouté souhaité sur action de l'utilisateur.

L'arbre est construit à partir du modèle de dépendances par propagation des causes qui ont contribué à la survenue de l'événement redouté visé. L'arbre de défaillances est directement exploitable avec le logiciel Arbre-Analyste. Les portes et les événements de base sont automatiquement renseignés.

La figure 8 représente l'arbre de défaillance correspondant à la perte de capacité de déplacement haut du modèle « Système hydraulique ».

Figure 8 - Arbre de défaillances du système hydraulique généré depuis le modèle

### 3.3 Exportation du modèle en Altarica 3.0

#### 3.3.1 AltaRica 3.0 et OpenAltaRica

AltaRica 3.0 (Prosvirnova *et al.*, 2013) est, comme son nom l'indique, la troisième version du langage de modélisation AltaRica. Il améliore la version précédente, AltaRica Data-Flow (Boiteau *et al.*, 2006) suivant deux points fondamentaux. La sémantique d'exécution basée les les Systèmes de Transitions Gradées (GTS) et le paradigme de structuration basé sur S2ML.

En premier lieu sa sémantique d'exécution est basée sur les Systèmes de Transitions Gardées, GTS pour Guarded Transitions Systems (Rauzy, 2008). Les GTS généralisent les formalismes classiques d'analyse du risque (arbres de défaillance, chaînes de Markov, réseaux de Pétri stochastiques, etc.), sans augmenter la complexité des calculs d'indicateurs du risque. L'exécution des modèles AltaRica 3.0 est similaire aux autres formalismes à événements discrets dans le sens où chaque fois qu'une transition est tirable, elle est alors planifiée et sera alors potentiellement tirée après un certain délai (Cassandras *et al.*, 2008), (Zimmerman, 1976). Ces délais peuvent être déterministes ou stochastiques ; et dans ce dernier cas, AltaRica 3.0 fournit les délais usuels (du type exponentiel, Weibull, etc.) et des délais empiriques.

De plus, le paradigme de structuration du langage AltaRica 3.0 est basé sur S2ML, pour System Structure Modeling Language (Batteux *et al.* 2015). S2ML unifie les deux paradigmes (de structuration) dominants des langages de modélisation, i.e. l'orienté objet et l'orienté prototype. Il permet de modéliser suivant deux approches combinables. L'approche dite 'top-down', avec une vision au niveau système permettant la réutilisation de schémas de modélisation, et donc utilisant le paradigme orienté prototype. L'approche dite 'bottom-up', avec une vision au niveau composants permettant la réutilisation de composants, définis en bibliothèques, et donc utilisant le paradigme orienté objet.

Le pouvoir d'expression du langage AltaRica 3.0 est seulement une partie de la solution d'analyse du risque de systèmes complexes. L'autre partie concerne les outils d'évaluation associés, notamment l'efficacité pratique des algorithmes d'évaluation. En effet, l'évaluation pour la sûreté ou la fiabilité est un problème dit de complexité #P-hard (Valiant 1979). Cela signifie que cette barrière de complexité n'est pas un problème de technologie, mais un problème théorique. Il est donc nécessaire de concevoir des outils d'évaluation implémentant efficacement les algorithmes de traitement. Le langage AltaRica 3.0 vient de ce fait avec un ensemble d'outils permettant de calculer différents indicateurs. Différents outils permettent d'évaluer les modèles AltaRica 3.0. Le simulateur pas-à-pas est l'outil permettant de réaliser des expériences virtuelles sur les modèles (vérification manuelle de modèles par exemple). Les deux outils de compilation : vers les arbres de défaillance et vers les chaînes de Markov (pas encore disponible) permettent de générer, à partir de modèles AltaRica 3.0, des modèles vers ces formalismes cibles. L'utilisation d'outils d'évaluation associés à ces formalismes est ensuite possible. Le générateur de séquences permet d'obtenir l'ensemble des séquences d'événements amenant à un événement redouté considéré. Enfin, le simulateur stochastique permet d'obtenir des statistiques sur des exécutions aléatoires de modèles AltaRica 3.0. L'ensemble de ces outils est

distribué au sein de la plateforme OpenAltaRica issu du projet du même nom.

Le projet OpenAltaRica est un projet réalisé au sein de l'Institut de Recherche Technologique (IRT) SystemX et en partenariat avec Thales, Apsys, Safran et AltaRica Association. D'une durée de 5 ans et ayant démarré fin 2014, ce projet vise l'implémentation de la plateforme de référence pour le langage AltaRica 3.0. Cette plateforme d'expérimentation, disponible sur le site du projet ([www.openAltaRica.fr](http://www.openAltaRica.fr)), intègre déjà l'ensemble des outils développés. Elle sera complétée par les outils en cours de développement.

#### 3.3.2 Objectifs de l'exportation

L'exportation du modèle en direction de la plateforme OpenAltaRica permet l'emploi de l'ensemble des outils de simulations dont :

- le simulateur pas-à-pas ;
- le simulateur stochastique ;
- le générateur de séquences (en cours d'implémentation).

#### 3.3.3 Exportation depuis System-Analyst

L'exportation du modèle en Altarica 3.0, sur demande de l'utilisateur, est transparente. System-Analyst créé automatiquement un nouveau projet dans l'interface OpenAltaRica puis l'ouvre.

## 4 - Conclusion

Cette communication permet de présenter à la communauté un nouvel outil libre, dédié à l'analyse des risques. Ce nouvel outil complète, aux côtés d'OpenAltarica, d'XFTA (Rauzy, 2012) et d'Arbre-Analyste (Clement et al., 2014), l'offre de logiciels libres disponibles pour l'ensemble de la communauté de la maîtrise des risques.

System-Analyst apporte la possibilité de modéliser graphiquement un système à l'aide d'un formalisme de haut niveau et proche des architectures fonctionnelles et matérielles. A partir de cette modélisation, il est alors possible de réaliser automatiquement un ensemble d'analyses afin de répondre à différents besoins de la Sûreté de fonctionnement. Réaliser les analyses à partir d'un modèle commun entraîne intrinsèquement une cohérence forte entre les analyses. Par ailleurs, baser les analyses de Sûreté de fonctionnement sur les modèles permet d'apporter la capacité de:

- se rapprocher des formalismes des ingénieries systèmes et matérielles ;
- valider plus facilement les analyses ;
- capitaliser les modèles.

System-Analyst est encore au stade de développement actif. Cependant, une version « bêta » est en libre diffusion sur le site Internet du projet : <http://www.system-analyst.fr/>

## 8 Références

IEC 61025 "Fault tree analysis"

IEC 60812 "Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)"

Clement, E.; Rauzy, A.; Thomas, T (2014)- Arbre Analyste : un outil d'arbres de défaillances respectant le standard Open-PSA et utilisant le moteur XFTA

Riera, D., Milcent, F., J.Parisot, & Clement, E. (2012). Modélisation dynamique en sûreté de fonctionnement : une avancée pour la modélisation des systèmes complexes. Dans J. Barbet (Éd.), Actes du Congrès Lambda-Mu 18.

Rauzy, A. (2012). XFTA : pour que cent arbres de défaillance fleurissent au printemps. Actes du Congrès Lambda-Mu 18.

Prosvirnova, T., Batteux, M., Brameret, P.-A., Cherfi, A., Friedlhuber, T., Roussel, J.-M., et al. (2013). The AltaRica 3.0 project for Model-Based Safety Assessment. 4th IFAC Workshop on Dependable Control of Discrete Systems, DCDS 2013.

Prosvirnova, T., & Rauzy, A. (2013). AltaRica 3.0 project: compile Guarded Transition Systems into Fault Trees. European Safety and Reliability Conference, ESREL 2013.

Boiteau M., Dutuit Y., Rauzy A., Signoret J-P. (2006). The altarica dataflow language in use: modeling of production availability of a multistate system

Rauzy A. (2008). Guarded transition systems: a new states/events formalism for reliability studies

Batteux, M.; Prosvirnova, T.; Rauzy, A. (2015). System Structure Modeling Language (S2ML).

Valiant, L. (1979). The Complexity of Computing the Permanent

Cassandras, C.; Lafortune, S. (2008), Introduction to Discrete Event Systems, 2nd Edition, Springer - Verlag.

Zimmermann, K. (1976). Extremal Algebra. Ekonomický ústav ČSAV