Une démarche intégrée pour l'ingénierie de système complexe face aux risques

Vincent Chapurlat, Saber Aloui

Une démarche intégrée pour l'ingénierie de système complexe face aux risques

Vincent Chapurlat, Saber Aloui

Laboratoire de Génie Informatique et d'Ingénierie de Production - LGI2P - site EERIE de l'Ecole des Mines d'Alès, Parc Scientifique Georges Besse, F30035 Nîmes cedex 5, France - Tel : (+33) 466 387 066 - Fax : (+33) 466 387 074 - Email : Vincent.Chapurlat@ema.fr, Saber.Aloui@ema.fr

Résumé : *Cet article présente une approche d'ingénierie de système complexe intégrant à la fois des principes de modélisation système, de modélisation de risque et des mécanismes de preuve et de simulation. Elle permet de mettre en œuvre des langages de modélisation dédiés classiquement à la description des vues fonctionnelles, comportementales et structurelles du système. Elle intègre ensuite deux vues supplémentaires. La première vue permet de décrire, sous forme de propriétés, les exigences fonctionnelles et non fonctionnelles devant être couvertes par le système, les exigences devant être respectées par les différents modèles élaborés au moyen des langages de modélisation, et, enfin, de propriétés traduisant différents déficits systémiques cindynogènes (DSC) à l'origine de risques potentiels durant la vie du système. La deuxième vue est une vue ontologique permettant à plusieurs acteurs issus de domaines ou de métiers différents d'échanger et de parfaire leur compréhension de la représentation système / risque qui est alors possible. Les mécanismes de preuve permettent enfin d'établir la véracité de certaines propriétés et de faire ainsi apparaître soit des erreurs de modélisation, soit des causes potentielles de risques encore non maîtrisés. Enfin, la simulation permet aux acteurs du processus d'ingénierie de mieux percevoir l'évolution et le comportement éventuellement imprévisible du système, de percevoir de nouvelles situations encore objectivement ou subjectivement oubliées et de repenser ainsi tout ou partie de ce système.*

Mots clés *: Modélisation système, Risque, Vérification, Validation, Simulation*

1. Introduction

La maîtrise des risques dans un projet d'ingénierie de système complexe tel qu'un projet de (re)conception de produit, de procédé ou de processus, est un enjeu crucial. Dans l'état actuel, les acteurs de ce projet s'appuient sur des approches de modélisation et d'analyse du système qui intègrent encore très mal ces notions de risques. Les acteurs doivent alors faire appel à des approches dédiées plus spécifiquement à la description et à l'analyse des risques que l'on peut classer en approches *a posteriori* et approches *a priori* pour s'assurer d'une couverture suffisante des risques encourus.

Les approches *a posteriori* ont un objectif essentiellement explicatif et nécessitent de manipuler des modèles du système peu formalisés et de toute manière difficilement formalisables. Citons, par exemple, l'approche basée sur le retour d'expérience (REX). Il permet de comprendre un enchaînement de faits ayant conduit à un événement redouté et d'en dégager des recommandations d'amélioration ou des bonnes pratiques en termes de conception et de pilotage du système ou de la crise résultante du risque.

Dans les approches *a priori*, au contraire, les acteurs s'attachent généralement à modéliser le risque lui-même et s'appuient sur une description plus ou moins dédiée ou complète du système (son comportement, ses fonctions, ses composants et leurs nécessaires interactions). Les acteurs peuvent ainsi évaluer, rechercher des causes potentielles, quantifier (au travers de probabilités d'occurrence et de gravités) des causes en servant de référentiels pré établis par exemple. L'objectif poursuivi est un objectif de prévention et de résolution anticipative. Ces approches ont été développées et appliquées essentiellement dans et pour les industries

sensibles (nucléaire, spatial, aéronautique, agroalimentaire). Citons par exemple les arbres de défaillance (Lee *et al.*, 1985), l'AMDEC (l'Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité) (Scipioni *et al.*, 2002), HACCP (Hazar Analysis and Critical Control Point) (Wallace *et al.*, 2005) et d'autres méthodes référencées dans (Tixier, 2002). Choisir une de ces approches relève souvent de l'expérience que l'on en a et du niveau de maîtrise du système que l'utilisateur final ou les acteurs du projet définissent comme suffisant en tenant compte de contraintes de temps, financières, de ressources, etc. Le Tableau 1 inspiré de (Verdel, 2000) tente de caractériser quelques alternatives pour guider ce choix en fonction des éléments qui sont à l'origine et impactés par le risque.

Si l'on s'intéresse aux risques induit par l'impact de :	***On peut utiliser une méthode orientée :***
Installation → Installation	Sûreté de fonctionnement
Installation → Opérateur	Ergonomie (texte règlementaires poste de travail, nuisances)
Opérateur → Installation	Sécurité (fiabilité humaine, malveillance interne, sabotage)
Installation → Population	Hygiène, Santé publique
Population → Installation	Usage, Sécurité (malveillance externe, sabotage, terrorisme)
Installation → Ecosystème	Ecologie, normalisation, textes réglementaires
Ecosystème → Installation	Urbanisme, risques naturels

Tableau 1. Intérêts de l'acteur et type de méthode envisageable

Un acteur, soucieux de mieux comprendre, de se rassurer et de décider des barrières de prévention à mettre en place pour maîtriser un maximum de risques encourus devrait donc être apte à mettre en œuvre simultanément plusieurs approches. Cela suppose un effort de modélisation et d'analyse supplémentaire, des compétences différentes et des ressources certainement plus importantes que celles allouées dans le cadre du projet. Cet article présente donc un cadre conceptuel et une démarche d'ingénierie de systèmes en vue de la maîtrise des risques. Elle permet, non seulement de s'abstraire des problèmes liés aux aspects multi compétences et multi modèles, mais aussi de mettre à disposition de cet acteur des mécanismes d'analyse de modèles innovants dans le domaine de la gestion du risque.

2. Problématique et démarche proposée

La problématique abordée ici est triple :

- En termes de modélisation du système : l'acteur doit être apte à appréhender le système sous des points de vue différents :
  - téléologique : quels sont les objectifs pour lesquels le système a été construit ? quelles sont ses missions ?
  - fonctionnel : quelles sont les fonctions mises en œuvre pour atteindre cette ou ces missions ?
  - comportemental : quels sont les scénarios d'évolution envisageables ? quelles sont les configurations des ressources atteignables et que permettent-elles de faire ?
  - structurel : quelles sont ces ressources ? quelle est leur organisation en vue de soutenir l'atteinte des objectifs ? etc.

  Les connaissances issues de l'expression de ces points de vue sont généralement exprimées dans des formalismes et des langages spécifiques. Par exemple, (Zelm *et al.*, 1995; Chen *et al.*, 1997; Kosanke *et al.*, 1999; Vernadat, 2001; Berio, 2006) présentent

différents cadres de référence et langages développés pour l'industrie afin de répondre aux besoins d'amélioration de la production, d'urbanisation du système d'information ou encore de développement d'outil support à l'aide à la décision. Ces approches sont basées sur des concepts systémiques. Elles privilégient donc généralement une vision multi vues d'une part et hiérarchisée d'autre part en faisant apparaître plusieurs niveaux de détail afin d'appréhender le système dans sa globalité. Cependant, d'après (Vallespir *et al.*, 2003), l'hétérogénéité des langages (et des modèles qui en sont les instances), même si ils possèdent des bases conceptuelles proches voire communes, fait ressurgir un manque d'interopérabilité[1]. Il faut donc intégrer ces différents langages dans un cadre unique de modélisation système afin de préserver l'usage nécessaire de ces langages et de garder une cohérence entre les différentes vues du système.

- En termes de **modélisation des risques** : les approches de modélisation du risque citées plus haut sont complémentaires et couvrent les besoins de maîtrise dans différents types de systèmes et pour différents objectifs (sécurité, sûreté, etc.). Leur usage est donc tout à fait légitime. Cependant, elles utilisent des concepts de représentation hétérogènes et peuvent à leur tour être considérée comme non interopérables. Il reste difficilement concevable de devoir rebâtir d'autres modèles que celui d'ingénierie pour pouvoir appliquer chaque méthode à bon escient. Il faut donc intégrer un langage de modélisation du risque avec les langages vus ci-dessus.
- Enfin, en termes d'**analyse du modèle résultant système / risque** : Quelles garanties les acteurs pourront-ils avoir sur la qualité des modèles et leur pertinence ? Il faut donc disposer de mécanismes permettant de détecter des erreurs de modélisation, des incohérences, des inconsistances pour éviter de décrédibiliser ensuite des propositions d'amélioration difficilement validées et argumentées. De plus, un système complexe (Le Moigne, 1990) est le siège d'interactions nombreuses et variées difficiles à décrire et à percevoir complètement, quel que soit les langages de modélisation employés. Ce système évolue dans un environnement mouvant, incontrôlable et généralement décrit en terme d'entrées sorties et d'hypothèses plus ou moins bien formulées et complètes. Tout cela gêne la compréhension de la réelle dynamique d'évolution du système étudié. Des événements perturbants, à l'origine de risques potentiels, sont imprévisibles voire même difficilement imaginables. Les situations dans lesquelles le système est amené à évoluer sont impossibles à percevoir et à prévoir dans leur globalité. Cependant, le système se doit d'être plus robuste pour répondre à ces situations ou d'être insensible à ces événements. Il est donc nécessaire de disposer de mécanismes permettant de simuler le comportement du système pour provoquer et faciliter la compréhension de ces situations et comportements émergents.

L'approche proposée dans cet article est bâtie autour de langages de modélisation à la fois du système et du risque rendus interopérables par l'usage d'un méta modèle commun et unique suivant les principes de méta modélisation de l'approche MDA (Model Driven Architecture) (Bézivin et Gerbé, 2001) et plus généralement du MDE (Model Driven Environment) (Schmidt, 2006). Les langages choisis pour représenter chaque vue du système sont inspirés de l'ingénierie des systèmes et de la modélisation d'entreprise. Trois cadres conceptuels décrits plus loin ont ensuite été pris en compte et intégrés dans cette approche de modélisation mixte pour représenter les risques.

L'hypothèse essentielle de ce travail de recherche est qu'une analyse des risques encourus par un système est une démarche guidée de vérification[2] et de validation[3] (AIAA, 1998) et de

[1] L'interopérabilité se définit comme '*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*' (IEEE, 1990).

[2] Vérification : '*Process of determining that a model implementation and its associated data accurately represent the developer's conceptual description and specifications*'. En d'autres termes, '*avons-nous bien construit le modèle ?*'

simulation. En effet, la vérification cherche à prouver que l'on a bien construit le modèle du système. Un modèle vérifié sera donc exempt d'erreurs de construction et pourra être ainsi considéré comme le seul support de raisonnement fiable durant le projet d'ingénierie. La validation cherche ensuite à s'assurer que l'on a construit le bon modèle du système, qu'il est pertinent. Elle va consister ici, d'une part, à s'assurer de l'absence de causes potentielles de risques de diverses natures (normative, réglementaire ou encore liées à des déficits systémiques cindynogènes tels que présentés plus loin). Enfin, la simulation permet de faire émerger et d'anticiper certaines situations importantes mais omises.

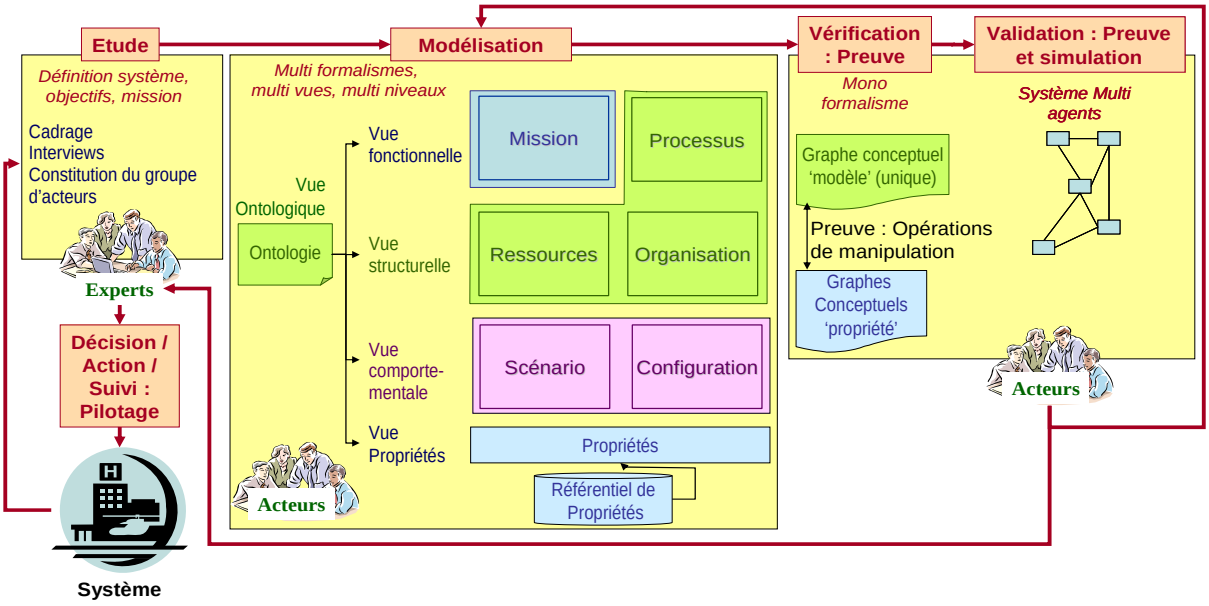

Figure 1. Synoptique de la démarche

Cette approche, schématisée dans la Figure 1, se scinde en trois étapes :

- **Etude** : Cette première étape vise à réduire la complexité de représentation du système et à l'inclure dans un *cadre de représentation* partagé (Penalva, 1997; Aloui *et al.*, 2006a). L'objectif est de fournir une vision globale du système restreinte aux objectifs de l'étude, c'est-à-dire au projet de l'équipe de modélisation. Elle rassemble les hypothèses de l'étude et explicite les grands principes de modélisation et d'analyse choisis.
- **Modélisation** : Le résultat de l'étape de modélisation est une représentation multi vues et multi modèles du système étudié (Aloui *et al.*, 2006a; Aloui *et al.*, 2006b). Cette représentation se traduit, d'une part, par plusieurs modèles interconnectés, soit par une relation verticale (hiérarchique) (granularité plus ou moins fine d'une partie du système), soit par une relation horizontale traduisant les liens entre les différentes vues. Elle se complète ensuite par un ensemble de propriétés (Lamine, 2001; Chapurlat et Aloui, 2006) qui décrivent les attentes et exigences à la fois normatives, règlementaires, fonctionnelles et non fonctionnelles auxquelles le système doit répondre.
- **Vérification** et **Validation**: La vérification a pour objectif de s'assurer de la cohérence des différents modèles dans chaque vue résultant de l'emploi de chaque langage de modélisation puis de la cohérence globale de la représentation entre ces vues. Elle permet de relever des erreurs de modélisation, de lever des ambiguïtés, de compléter les modèles dans chaque vue et d'améliorer leur richesse de détail si nécessaire. Les acteurs de l'étape de modélisation peuvent ainsi objectivement conférer aux modèles un niveau de confiance suffisant et reconnu. Cette phase repose sur une technique de preuve formelle de

[3] Validation : '*Process of determining the degree to which a model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model*. En d'autres termes, '*avons-nous construit le bon modèle ?*'

propriétés (Accelera, 2004) telle que proposées dans (Chapurlat *et al.*, 2003; Kamsu-Foguem, 2004) reposant sur l'emploi des Graphes Conceptuels (Sowa, 1984). Ensuite, la validation a pour objectifs, d'une part de s'assurer de la pertinence du modèle vis-à-vis du système réel, d'autre part, de rechercher et d'identifier les causes potentielles de risques encore ignorés et, enfin, de rechercher des comportements encore méconnus pouvant entraîner le système à subir de nouveaux risques. Deux techniques sont mises en œuvre. La technique de preuve déjà utilisée durant la phase de vérification est alors orientée vers la preuve de propriétés spécifiques. Une approche multi agents est ensuite employée pour simuler le comportement du système au moyen d'agents communicants possédant leur propre dynamique.

3. Modélisation : le système et les risques

La Figure 1 schématise les 5 vues proposées dans cette approche pour lesquelles les langages de modélisation choisis sont présentés dans le Tableau 2.

Vue	***Objectif***	***Langage***
Vue Ontologie	Construire de proche en proche une ontologie décrivant les concepts communs et/ou leurs équivalences sémantiques afin de supporte l'échange et la communication entres les différents acteurs impliqués dans le processus d'ingénierie. Cette ontologie est aussi nécessaire aussi lors de la spécification des propriétés de la vue propriété.	**Web Ontology Language** (OWL) (Allemang *et al.*, 2005): extension XML basée sur RDF
Vue Fonction	Décrire les objectifs, la finalité et les missions ou fonctions à remplir (à couvrir) pour atteindre ces objectifs	**KAOS** (Van Lamsweerde, 2000) est une méthodologie d'ingénierie des exigences qui permet de représenter et de structurer les différents niveaux d'exigences, d'attente et d'objectif du système. Chaque objet de modélisation est doté d'une telle vue
Vue Organisation	Décrire les différentes unités d'organisation et les ressources humaines, matérielles et logicielles devant être impliquées dans le système, leurs responsabilités, compétences et interactions vis-à-vis des objets processés par le système. Décrire enfin les processus (client, support et de pilotage) devant permettre au système d'atteindre ses objectifs en utilisant au mieux ces ressources.	**UEML** (Unified Enterprise Modeling Language) (Vernadat, 2002) et **Construct** (ISO/DIS 19440, 2004) **eFFBD** (Enhanced Functional Flow Block Diagram)(Long, 2002) pour la description des processus client, support et de pilotage
Vue Comportement	Décrire les différents scénarios d'exécution de chaque processus selon les configurations des unités d'organisation et des ressources, c'est-à-dire l'état dans lequel elles se trouvent à chaque instant pour accomplir leurs tâches.	**eFFBD** pour la description des scenarios instanciés à partir des modèles de processus de la vue Organisation **Statecharts** (Harel, 1987) Décrit les changements d'état d'un objet en réponse à des événements
Vue propriété	Décrire les propriétés Modèles, Axiomatiques et Système(Chapurlat, Kamsu-Foguem and Prunet, 2003; Kamsu-Foguem, 2004) permettant de vérifier, d'une part, le modèle du système, d'autre part, de rechercher les causes potentielles de risques en s'inspirant des Déficits Systémiques Cindynogènes (DSC). Un référentiel de propriétés est alors fournir pour supporter la phase de spécification des propriétés par les acteurs non experts du domaine.	**LUSP** (Langage Unifié de Spécification de Propriétés) (Chapurlat *et al.*, 2006)

Tableau 2. Les différentes vues de modélisation et les langages associés

Concernant la modélisation système, le choix s'est porté sur des approches et des langages issus de l'ingénierie des exigences (KAOS), de l'ingénierie des systèmes (eFFBDn Statecharts) et de la modélisation d'entreprise (UEML, ISO/DIS 19440).
Afin de permettre dans un premier temps de vérifier la cohérence des modèles obtenus, une vue *propriété* a été intégrée. Une propriété est *une connaissance que l'on a du système ou d'un modèle qui traduit une exigence, une finalité ou une caractéristique à satisfaire.* (Lamine, 2001) a proposé ainsi les bases d'un langage formel de représentation des propriétés baptisé LUSP (Langage Unifié de Spécification de Propriétés) qui définit une propriété comme une relation causale typée et contrainte. Un référentiel de propriétés définit dans (Chapurlat, Kamsu-Foguem and Prunet, 2003; Kamsu-Foguem, 2004) regroupe ensuite des propriétés génériques de système décrites en suivant cette formalisation. La spécification des propriétés des différents modèles repose sur l'usage de ce référentiel de propriétés.
Enfin, la vue Ontologie permet de définir un univers du discours commun à tous les acteurs du projet, permettant ainsi de faciliter la manipulation des langages de modélisation et des propriétés.
Concernant la modélisation du risque, le travail de recherche a mis en évidence trois cadres conceptuels et donc trois méta modèles possibles (non présentés ici) du risque qui sont répertoriés dans le Tableau 3. Ils sont respectivement issus de l'approche systémique et, en particulier, de la Méthodologie d'Analyse des Dysfonctionnements dans les Systèmes (MADS) (Lesbats *et al.*, 1999), de l'approche des Cindyniques (Kervern, 1995) et de l'approche par les situations de management (Penalva, 2004; Aloui *et al.*, 2005).

Cadre	***Type de risque***	***Méta modèle du risque***	***Mécanismes d'analyse***
Systémique	Risque déjà répertorié par exemple dans un référentiel, quantifié et/ou qualifié, explicable et quelquefois déjà maîtrisé	Méta modèle du risque inspiré de l'approche **MADS MOSAR**	Aspect assez mécaniste d'évaluation et de quantification du risque
Cindyniques	Risque potentiel liés à l'organisation et aux acteurs du système	Méta modèle du risque basé sur la notion de **Déficits Systémiques Cindynogènes** (DSC)	Raisonnement et expertise sur les ressources et leurs interactions
Situationniste	Risques liés aux situations rencontrées	Méta modèle du **risque situationnel** en phase de pilotage du système	Raisonnement sur les situations du système et les événements occurents

Tableau 3. Cadres conceptuels pour la modélisation et l'analyse des risques

Ce sont ici aussi les propriétés qui permettent alors d'intégrer ces différents méta modèles :

- Les risques déjà *identifiés* (donc modélisables au moyen du méta modèle systémique et souvent issus des entretiens avec les professionnels lors de la modélisation)
- Les risques liés aux e*xigences* normatives, réglementaires ou empiriques qui, si elles ne sont pas respectées, impactent le fonctionnement de l'organisation.
- Les risques liés aux Déficits Systémiques Cindynogènes (DSC) (Kervern, 1995) qui décrivent des carences classiques d'un groupe d'acteurs au sein d'une organisation pouvant effectivement exposer cette dernière à des risques (Chapurlat et Aloui, 2006).

Le cadre de modélisation résultant de ce travail de recherche a été implémenté au travers de l'outil GME (GME, 2006). Tous les concepts du langage sont ainsi issus d'un même et unique méta modèle et il n'existe ainsi pas de problèmes d'interopérabilité inhérents classiquement à l'emploi de plusieurs langages de modélisation définissant chacun leurs propres concepts et leur propre sémantique (Vallespir, Braesch, Chapurlat and Crestani, 2003). Un exemple de modélisation au moyen d'un eFFBD du processus de prise en charge d'un patient dans un système de santé est donné Figure 2.

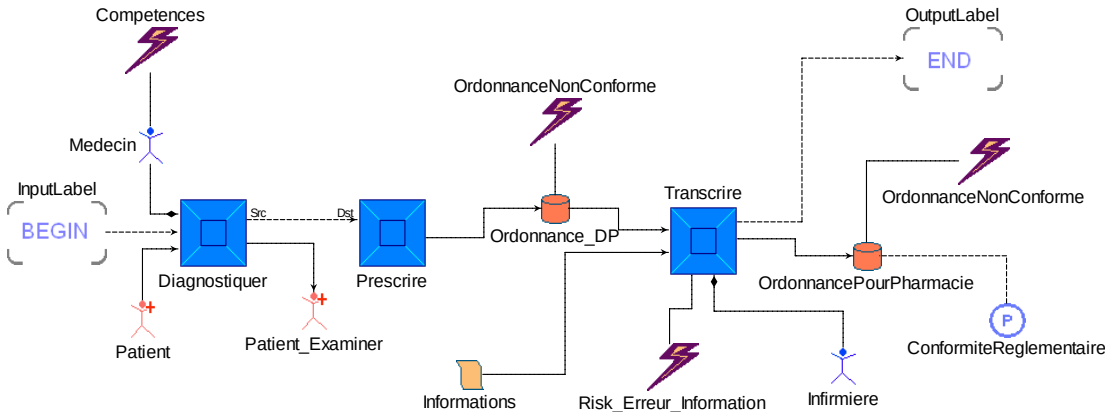

Figure 2. Exemple de modélisation d'un processus avec des eFFBD

4. Analyser : Prouver des propriétés et Simuler des situations

Une fois le modèle global du système bâti, lui-même étant composé de plusieurs modèles spécifiques à chaque vue, la première phase d'analyse va consister à vérifier sa cohérence globale.

4.1 Preuve pour vérifier

Cette première phase de l'analyse permet de lever des ambiguïtés ou erreurs de modélisation, de compléter les modèles dans chaque vue et dans chaque niveau de détail à l'intérieur d'une même vue.

Il existe de nombreuses approches et outils formels de preuve (Yahoda, 2003). Cependant, ils restent difficiles d'accès et d'usage sans un effort considérable de formation des acteurs et d'adaptation des modèles à vérifier. Il est donc nécessaire de trouver un compromis entre une vérification complètement formelle, et donc exhaustive, et une vérification ad hoc qui s'autorise une certaine latitude dans la rigueur de la preuve. La démarche de preuve proposée repose sur l'utilisation des Graphes Conceptuels (Kamsu-Foguem, 2004) proposés à l'origine par (Sowa, 1984). Un graphe conceptuel est un langage formel de représentation de connaissances qui possède plusieurs atouts. Il permet d'abord de représenter de manière graphique, aisée à manipuler, rigoureuse et lisible la connaissance sous la forme de graphes alternant des nœuds représentant les concepts et de liens représentant les relations utilisées dans les langages de modélisation visés. Il repose enfin sur des fondements mathématiques relativement plus aboutis bien que moins outillé d'un point de vue pratique que la majorité de ses concurrents. En effet, des mécanismes formels de projection, des principes de règles et de contraintes permettent de s'assurer de la véracité ou plus simplement de la présence d'une connaissance donnée dans un graphe. La démarche proposée ici repose donc sur :

- La traduction des modèles à vérifier dans un graphe conceptuel unique centralisant toute la connaissance contenue dans les différents modèles des différentes vues. Cela est rendu possible par le respect d'un méta modèle commun à tous les langages de modélisation utilisés.
- La traduction de chaque propriété dans un graphe indépendant dit graphe propriété. Ce graphe peut prendre différentes formes (simple, règle ou contrainte selon le type de propriété).
- L'utilisation des mécanismes essentiellement de projection et de recherche de contraintes pour s'assurer de la véracité de la propriété.

La Figure 3 présente un exemple simple de vérification d'une propriété P1 issue du référentiel de propriétés. Cet exemple est extrait de la modélisation du circuit du médicament pour lequel

un référentiel de propriété particulier a été développé à partir du référentiel des exigences de la Haute Autorité de Santé (HAS) (HAS, 2005). Cette propriété indique que seul un *médecin* est habilité à réaliser l'activité *Prescription*. A la fin de l'étape de modélisation, un des processus du système est décrit dans un diagramme eFFBD. Ce modèle et la propriété sont traduits dans deux graphes conceptuels indépendants. Le mécanisme de projection de graphe propriété sur le graphe modèle permet de rechercher et de prouver que le modèle respecte bien P1. Dans cet exemple, la projection a échouée. Il peut s'agir d'une erreur de modélisation auquel cas les acteurs peuvent modifier ou compléter le modèle. Il peut s'agir au contraire d'un fait avéré dans la réalité auquel cas une action doit être tentée pour respecter cette exigence.

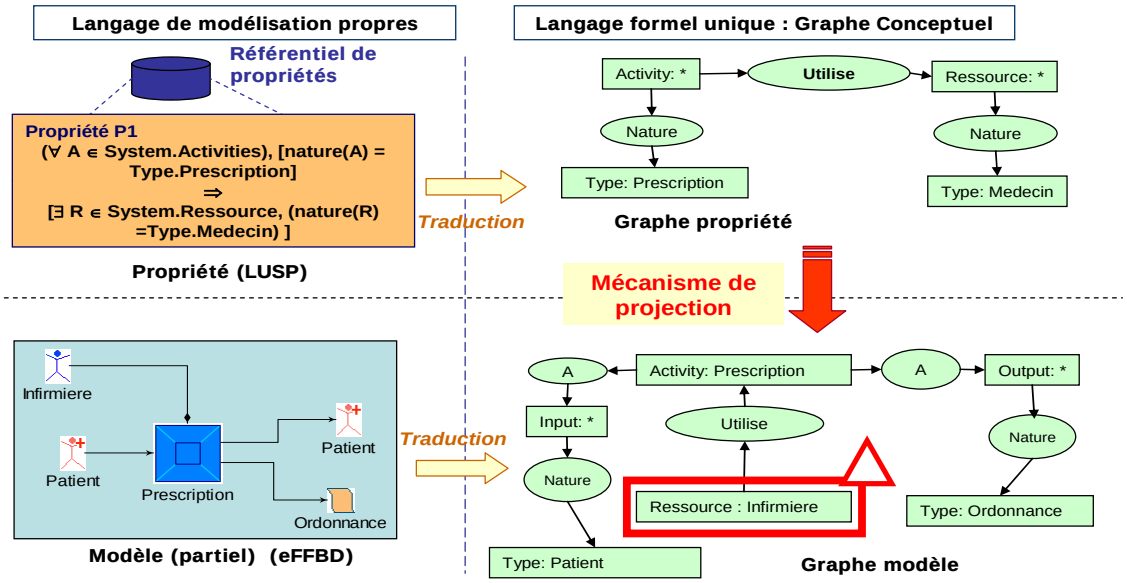

Figure 3 . Exemple de traduction Propriété / Modèle vers un graphe conceptuel et projection

L'outil support choisi ici est l'outil Cogitant de manipulation de graphes conceptuels (Cogitant, 2005).

4.2 Preuve pour rechercher des causes de risques

Une fois vérifié, le modèle doit alors être validé pour s'assurer de sa pertinence et pouvoir l'utiliser alors à des fins d'analyse des causes de risques. Le but poursuivi est la recherche et l'identification de risques liés aux DSC modélisés grâce aux propriétés. La technique utilisée est alors la même que celle présentée ci-dessus.

Par exemple, le concept de DSC de non communication se traduit par un cloisonnement au niveau des activités d'un même processus et engendre une difficulté pour les ressources de communiquer et d'échanger de l'information. La cause peut être technique, organisationnelle ou autre. Une propriété possible modélisant ce type de DSC permet de s'assurer que les activités qui constituent un processus échangent bien des objets de type « information » et qu'il existe donc bien un flux d'information et une traçabilité de cette information tout au long du processus. L'absence de ce type d'objet dans les échanges indique un risque potentiel lié à ce DSC.

4.3 Simuler pour valider et faire émerger des comportements imprévus a priori

La validation d'un modèle passe souvent par sa simulation, c'est-à-dire son exécution en tenant compte d'hypothèses liées au temps ou au niveau de détail. Pour cela, il est nécessaire de disposer d'une sémantique opérationnelle et de règles d'exécution formelles pour chacun des langages de modélisation employés dans la vue comportementale du système.

L'approche retenue se base ainsi sur une technique de simulation multi agents. L'approche multi agents est intéressante à plus d'un titre. Il existe en effet de nombreux travaux qui s'en inspirent (Parunak et Brueckner, 2001; Bernon *et al.*, 2006) pour tenter de modéliser l'émergence de comportements imprévus et les possibilités d'auto organisation entre des agents doués d'une certaine autonomie décisionnelle et comportementale. Ces travaux mettent aussi en avant la possibilité de disposer d'une simulation distribuée, plus rapide à mettre en œuvre et rendant compte de l'évolution parallèle de différentes entités complexes indépendamment les unes des autres.

L'objectif est double. Il faut d'abord animer le modèle afin de permettre aux acteurs du processus de valider les scénarios et les configurations décrits, c'est-à-dire de s'assurer de leur pertinence, et au besoin de les corriger, au regard du comportement perçu du système. Il faut ensuite focaliser l'attention des acteurs sur des comportements non prévus ou des situations émergentes de l'interaction des ressources entre elles (Beurier *et al.*, 2003) dans différents scenarii ou configuration décrits.

Chaque entité de modélisation comportementale, c'est-à-dire chaque scénario, configuration, activité et ressource du système, est ainsi traduite dans un agent dont le comportement et les interactions avec d'autres agents sont spécifiques. L'exemple illustré par la Figure 4 décrit le graphe d'état d'un agent modélisant une ressource matérielle. Les transitions indiquent les conditions d'évolution de l'état de l'agent. La Figure 5 schématise enfin les interactions entre les différents types d'agent retenus.

La plate forme JADE (Java Agent DEvelopement framework) (Bellifemine *et al.*, 2003) a été choisie pour mener à bien cette simulation.

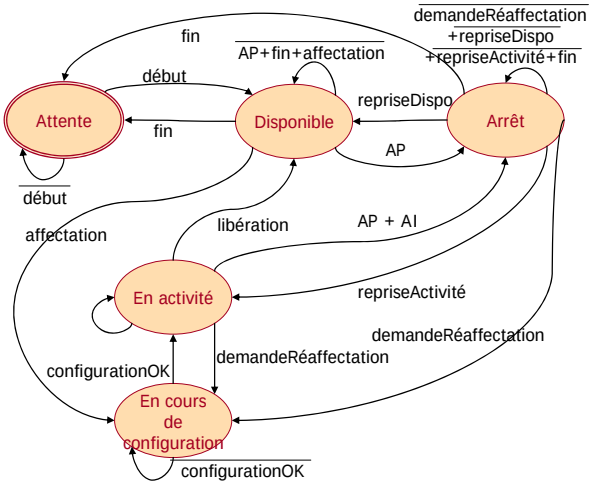

Figure 4. Graphe d'état d'un agent Ressource

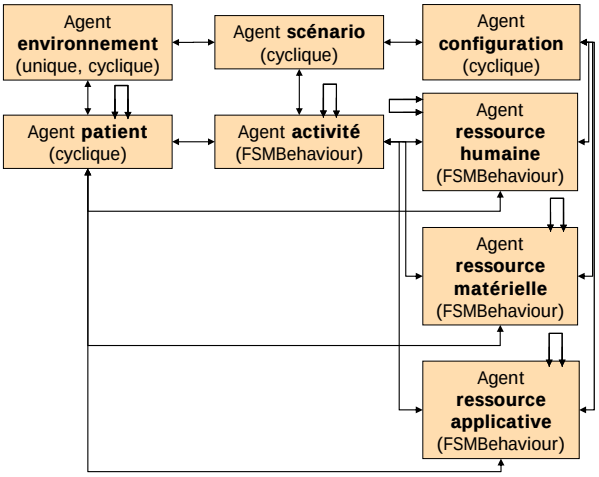

Figure 5. Architecture agents du simulateur

Les deux technique (preuve et simulation) nécessitent de traduire les modèles dans un langage tiers (Figure 6) en respectant des principes de traduction formalisés entre les différents outils GME, Cogitant et JADE.

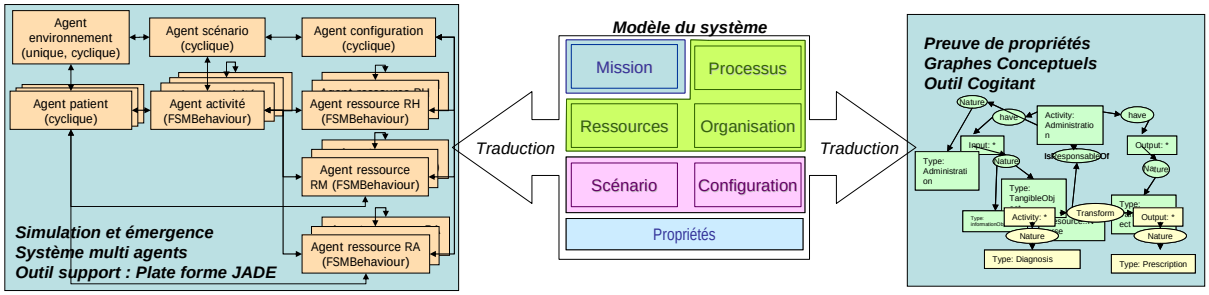

Figure 6. Preuve et Simulation : synoptique des techniques mises en oeuvre

5. Conclusion

L'approche proposée dans cet article est une approche mêlant des concepts issus de l'ingénierie des systèmes et de la maîtrise des risques avec des techniques de preuve formelle de propriétés et de simulation au moyen d'une plate forme multi agents. Elle met en œuvre des principes, des langages de modélisation et des mécanismes d'analyse intégrés dans un cadre unique formalisé.
Cette approche est actuellement appliquée dans le cadre d'un Centre Hospitalier Universitaire pour aider les acteurs de ce système de santé à mieux comprendre leur organisation, son fonctionnement, puis à le piloter en tenant compte des risques iatrogènes liés à chaque activité. La totalité de cette approche appliquée aux systèmes de santé est actuellement en cours de formalisation et un outil support est en cours de développement.

6. Bibliographie

Accelera (2004). PSL Property Specification Language Reference Manual. Version 1.1 Accelera Formal Verification Technical Committee (FVTC),.

AIAA (1998). Guide for the Verification and Validation of Computational Fluid Dynamics Simulations, American Institute of Aeronautics & Astronautics.

Allemang, D., I. Polikoff et R. Hodgson (2005). "Enterprise architecture reference modeling in OWL/RDF." Semantic Web - Iswc 2005, Proceedings 3729: 844-857.

Aloui, S., V. Chapurlat et J.-M. Penalva (2005). Réingénierie de système complexe pour le management du risque. JDMACS05 Journée doctorales et Nationale du GDR MACS Lyon, France.

Aloui, S., V. Chapurlat et J.-M. Penalva (2006a). How to improve socio-technical systeme interioperability ? A methodological approach. INCOM 2006 (12th IFAC Symposium on Information Control Problems in Manufacturing), St Etienne.

Aloui, S., R. Collomp, V. Chapurlat, J.-M. Penalva, A. Mousnier, P. Staccini et J.-F. Quaranta (2006b). Modélisation de système hospitalier pour le management du risque. GISEH06, Gestion et Ingénierie des SystÈmes Hospitaliers, Luxembourg.

Bellifemine, F., G. Caire, A. Poggi et G. Rimassa (2003). "JADE A White Paper." EXP in search of innovation 3(3): 6-19.

Berio, G. (2006). "UEML 1.0 and UEML 2.0: Benefits, problems and comparison." Business Process Management Workshops 3812: 245-256.

Bernon, C., M.-P. Gleizes et G. Picard (2006). Enhancing Self-Organising Emergent Systems Design with Simulation 7th International Workshop on Engineering Societies in the Agents World (ESAW'06), Dublin.

Beurier, G., O. Simonin et J. Ferber (2003). Un modèle de Système Multi-Agents pour l'Emergence Multi-Niveaux. JFSMA'03: Journées Francophones sur les Systèmes Multi-Agents Hammamet, Hermès.

Bézivin, J. et O. Gerbé (2001). Towards a Precise Definition of the OMG/MDA Framework. 16th IEEE International Conference on Automated Software Engineering (ASE'01), San Diego, USA.

Chapurlat, V. et S. Aloui (2006). How to detect risks with a formal approach? From property specification to risk emergence. MSVVEIS-2006, The 4th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems on ICEIS, 8th International Conference on Enterprise Information Systems, Paphos, Cyprus.

Chapurlat, V., B. Kamsu-Foguem et F. Prunet (2003). "Enterprise model verification and validation: an approach." Annual Review in Control 27(2): 185-197.

Chapurlat, V., B. Kamsu-Fogum et F. Prunet (2006). "A formal verification framework and associated tools for enterprise modeling: Application to UEML." Computers in Industry 57(2): 153-166.

Chen, D., B. Vallespir et G. Doumeingts (1997). "GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology." Computers in Industry 33(2-3): 387-394.

Cogitant (2005). CoGITaNT Version-5.1 – Reference Manual (voir http://cogitant.sourceforge.net).

GME (2006). The Generic Modeling Environment, (voir http://www.isis.vanderbilt.edu/projects/gme/).

Harel, D. (1987). "Statecharts: A visual formalism for complex systems." Science of Computer Programming, 8(3): 231-274.

HAS (2005). Organisation du circuit du médicament en établissement de santé. Haute Autorité de Santé, Organisation du circuit du médicament en établissement de santé, Fiche thématique, 2005.

ISO/DIS 19440 (2004). Enterprise integration: Constructs of enterprise modelling. Geneve, Draft Version.

Kamsu-Foguem, B. (2004). Modélisation et Vérification des propriétés de systèmes complexes : Application aux processus d'entreprise, Université de Montpellier II. Ph. D.

Kervern, G.-Y. (1995). Eléments fondamentaux des cindyniques, Economica.

Kosanke, K., F. Vernadat et M. Zelm (1999). "CIMOSA: enterprise engineering and integration." Computers in Industry 40(2-3): 83-97.

Lamine, E. (2001). Définition d'un modèle de propriété et proposition d'un langage de spécification associé : LUSP, Université Montpellier II. PhD.

Le Moigne, J.-L. (1990). La modélisation des systèmes complexes. Paris, Bordas, Dunot.

Lee, W. S., D. L. Grosh, F. A. Tillman et C. H. Lie (1985). "Fault tree analysis, methods, and applications – A review " IEEE Transactions on Reliability vol. 34(n°3): 194-203.

Lesbats, M., J. D. Santos et P. Périlhon (1999). Contribution à l'élaboration d'une science du danger. Ecole d'été "Gestion Scientifique du risque", Albi.

Long, J. (2002). Relationships between Common Graphical Representations in Systems Engineering Vitech Corporation

Parunak, V. D. H. et S. Brueckner (2001). Entropy and self-organization in multi-agent systems. International Conference on Autonomous Agents Agents, Montreal, Quebec, Canada, ACM Press New York,.

Penalva, J.-M. (1997). La modélisation par les systèmes en situations complexes. Paris, Université de Paris XI - Paris Sud. Ph.D.

Penalva, J.-M. (2004). Situations et systèmes complexes, Ecole des Mines d'Alès (EMA) Laboratoire de Génie Informatique et d'Ingénierie de la Production (LGI2P).

Schmidt, D. C. (2006). "Guest Editor's Introduction: Model-Driven Engineering." Computer vol. 39(2): 25-31.

Scipioni, A., G. Saccarola, A. Centazzo et F. Arena (2002). "FMEA methodology design, implementation and integration with HACCP system in a food company." Food Control volume 13(Issue 8): Pages 495-501.

Sowa, J. F. (1984). Conceptual structures: information processing in mind and machine. New York (U.S.A.), Addison-Wesley Longman Publishing Co., Inc.

Tixier, J. (2002). Méthodologie d'évaluation du niveau de risque d'un site industriel de type Seveso, basée sur la gravité des accidents majeurs et la vulnérabilité de l'environnement. Biosciences de l'environnement, chimie et santé, Aix Marseille PhD: 259.

Vallespir, B., C. Braesch, V. Chapurlat et D. Crestani (2003). L'intégration en modélisation d'entreprise : les chemins d'UEML. MOSIM03- 3ème Conférence Francophone de Modélisation et Simulation, Toulouse.

Van Lamsweerde, A. (2000). Formal specification: a roadmap. ICSE - Future of SE Track: 147-159.

Verdel, T. (2000). Méthodologies d'évaluation globale des risques. Application au Génie Civil. Colloque Risques et Génie Civil, Paris, Presses de l'Ecole Nationale des Ponts et Chaussées.

Vernadat, F. (2001). "Enterprise modelling." Production Planning & Control 12(2): 107-109.

Vernadat, F. (2002). "UEML: towards a unified enterprise modelling language." International Journal of Production Research 40(17): 4309-4321.

Wallace, C. A., S. C. Powell et L. Holyoak (2005). "Development of methods for standardised HACCP assessment." British Food Journal 107(10-11): 723-742.

Yahoda (2003). web site presenting an overview of formal verification tools (see http://anna.fi.muni.cz/yahoda/)

Zelm, M., F. B. Vernadat et K. Kosanke (1995). "The Cimosa Business Modeling Process." Computers in Industry 27(2): 123-142.