

ALIX : UNE METHODE DE MODELISATION DES FLUX D'UN SYSTEME DE PRODUCTION

Alix Dontaine, Denis Pichel, Sergei Iassinovski, Abdelhakim Artiba* & Christophe Fagnart**

Résumé. - Ce papier présente un formalisme graphique de modélisation en entreprise destiné à la simulation des flux de systèmes de production de biens et de services. Parmi les contextes statique, décisionnel et dynamique de la modélisation, seul le contexte dynamique est abordé dans cet article ; les deux autres aspects étant en phase de développement. Ce formalisme est inspiré de deux méthodes de référence : SADT et UML. Il est destiné à s'intégrer dans une plate-forme complète d'analyse et d'optimisation. Celle-ci constitue un projet en cours de développement au sein de l'équipe de recherches du CREGI. Cette plate-forme d'analyse a pour objectifs la simulation, l'optimisation et l'analyse de la performance, axes de recherches privilégiés et depuis longtemps utilisés au CREGI dans bon nombre de projets industriels.

Mots-clés : Ingénierie d'entreprise, modélisation, simulation, réorganisation.

1. Introduction

Au cours des trente dernières années, le rôle des modèles d'entreprise a pris une réelle importance. Ils sont devenus indispensables au sein des organisations d'une certaine taille [FOX 98]. En effet, il ne s'agit plus seulement de posséder les matières premières, l'énergie et les ressources pour progresser et y bâtir sa croissance à long terme, tel que les économies nationales le font depuis des décennies. Il faut également maîtriser des ressources devenues stratégiques : l'information et la connaissance, disponibles notamment via la modélisation en entreprise [KAL

* Facultés Universitaires Catholiques de Mons - Centre de Recherches et d'Etudes en Gestion Industrielle, Mons, Belgique.

** Simpartners, Parc Scientifique INITIALIS, Rue Descartes 2, B-7000 Mons, Belgique.

01]. De nombreuses méthodes ont ainsi vu le jour, apportant de nouveaux concepts, de nouvelles approches, pour augmenter la diversité et la difficulté dans le choix d'une méthode de modélisation. Vernadat [VER 01] a mis en évidence les difficultés rencontrées par les utilisateurs qui ne peuvent fournir assez de temps ni d'argent pour produire des modèles d'entreprise sur base des méthodes et outils de modélisation actuels. C'est dans ce contexte que le CREGI a entamé la réalisation d'une plate-forme d'analyse et d'optimisation de processus [IAS 01]. Ce projet, qui vise notamment à coupler simulation et optimisation, est brièvement présenté sous son aspect « analyse de flux » dans le point suivant. La méthode de modélisation graphique est destinée à être utilisée directement par les entreprises et les chercheurs. Nous souhaitons qu'elle soit à la portée des PME, mais aussi de toute organisation désireuse de comprendre, d'améliorer, de communiquer aisément par rapport à son organisation.

Cet article a pour objectif de présenter l'avancement de nos travaux en modélisation. Dans un premier temps, nous situons notre approche dans le cadre de la démarche de réalisation d'une plate-forme d'analyse et d'optimisation actuellement en cours de développement au sein du CREGI. Nous présentons ensuite quelques outils et méthodes de modélisation et justifions le développement de notre approche.

2. Modélisation d'un système productif

La modélisation en entreprise tournée vers la simulation met en présence trois éléments : le domaine d'étude du monde réel, le modèle et le modèle de simulation. Un modèle est une représentation abstraite d'une partie du monde réel (le domaine d'étude), exprimée dans un langage de représentation. La simulation, via le modèle de simulation, imite le système réel en reproduisant pas à pas son comportement [LAN 97].

Ce passage du monde réel au modèle d'entreprise n'est pas évident tant la complexité (quantité de flux, de ressources, d'interactions, de prises de décisions) des systèmes est importante. L'un des axes de travail actuels de notre centre de recherches est de développer une plate-forme pour faciliter ce passage et permettre l'analyse et l'optimisation de systèmes réels. Le projet vise à traduire le domaine d'étude en modèles qui pourront être compris rapidement par toute personne impliquée et traduits en langage textuel afin de leur appliquer ensuite la méthode de résolution adaptée (simulation, systèmes experts, outils d'optimisation). La figure 1 présente la philosophie de ce projet en distinguant l'aspect modélisation et l'aspect résolution. La plate-forme a pour objectif d'unifier les 3 niveaux de modélisation (statique, décisionnel et dynamique) ainsi que des méthodes de résolution et d'aide à la prise de décisions via une modélisation graphique.

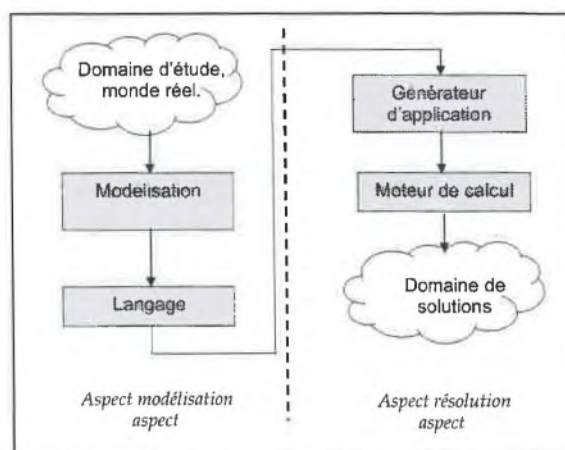


Figure 1 : Approche unifiée de résolution de problèmes de prises de décisions.

Dans le cadre de ce projet, le formalisme graphique sert de lien entre le domaine étudié et le langage. En amont, la modélisation est destinée à représenter tout système socio-économique : entreprise de production, de services, institution hospitalière, etc...¹. En aval, les méthodes de résolution ne doivent pas se limiter à la simulation mais également aux outils d'optimisation. Dans ce document, nous présentons la gestion des flux et la traduction en un langage de simulation : RAO [ART 98].

3. Méthodes de modélisation : nos inspirations

Il existe de nombreuses architectures de référence relatives à diverses approches de modélisation. Parmi les plus connues, citons – par ordre chronologique d'apparition – SADT, CIMOSA, GIM, ARIS, l'ENV40003, AIOSCOP, PERA et GERAM [VER 99]. Lors du congrès MOSIM'01, F. Vernadat en a présenté un historique complet. [VER 01]. Il en découle de nombreux langages de modélisation en entreprise parmi lesquels CIMOSA process language, ARIS, IDEF, IDEF*, IEM, WfMC, ODP, PSL/NIST,...

Malgré la diversité des approches en modélisation, il est difficile de trouver des outils de spécification et de modélisation accessibles aux PME. Les rares outils existants sont, soit, destinés aux grandes entreprises, soit, ne sont pas dotés de génération de modèles de simulation.

Après avoir analysé quelques méthodes de modélisation et au vu des acquis et expériences de notre équipe, nous avons analysé nos propres besoins en matière de modélisation en

¹ Dans l'expression « modèle d'entreprise » souvent utilisée, « entreprise » représente tout ou partie d'un système socio-économique donné [VER 99].

entreprise. Nos projets de recherche nous menant souvent à la rencontre des industriels, et plus particulièrement, des PME, mais également d'organisations de divers secteurs, nous souhaitons une méthode de modélisation flexible, simple et facilement compréhensible par tous. Des points de vue conceptuel et pratique, nous nous sentons ainsi proches des méthodes SADT et UML. Les paragraphes suivants mettent brièvement en évidence les notions de base de ces deux méthodes.

3.1 Description de SADT et UML

SADT (Structured Analysis Design Technique) [ROS 97][LIS 90] est une méthode de modélisation développée par Softech (USA) et IGL Technologie (France) en 1977 et introduite en Europe à partir de 1982 pour analyser des systèmes complexes. Bien que SADT ne nécessite aucun outil spécifique, quelques langages informatiques ont été créés à partir de ses concepts, et entre autres, les approches IDEF (Integration Definition for Function Modeling).

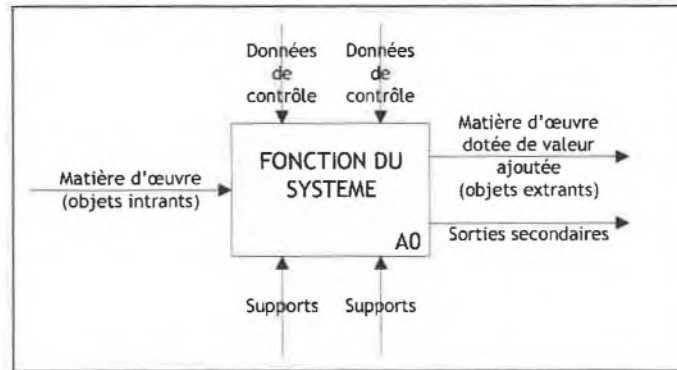


Figure 2 : Bloc fonctionnel selon le formalisme SADT.

SADT étudie le système dans son aspect fonctionnel (ou fonction d'usage) et dans son aspect structurel (ou fonction globale). SADT possède un formalisme graphique qui procède par analyse descendante (présence d'une hiérarchie par décomposition fonctionnelle). Un système est modélisé graphiquement via un bloc fonctionnel représenté par un rectangle à l'intérieur duquel est mentionnée la fonction globale (ou d'usage) (cf. figure 2). Sont représentés sur ce bloc :

- les entrées de matières d'œuvre transformées par la fonction ;
- les données de contrôle qui provoquent ou modifient la mise en œuvre de la fonction ;
- la sortie de matière d'œuvre dotée de la valeur ajoutée ;
- les sorties secondaires : flux d'informations associés au processus, sous-produits ou déchets ;
- les supports de la fonction (éléments matériels).

UML : L'utilisation grandissante de l'approche orientée objet dans tous les domaines a bouleversé les méthodes de conception des logiciels et des systèmes d'information. De nouvelles méthodes et de nouveaux langages de modélisation sont apparus. Parmi ces derniers, UML se veut un langage fédérateur de tous ces courants. Ainsi, UML apparaît naturellement dans les ouvrages consacrés à la programmation orientée objets [BEN 98][CAR 01] ou aux côtés de Merise [TAR 00] dans les ouvrages dédiés à l'analyse des systèmes d'information [HAI 00][MOR 00]. UML est l'union de trois méthodologies, utilisées pour le développement « Orienté Objets » [BOO 99]. Mis en place par Rational Software, UML a été accepté comme un standard en 1997 par l'Object Management Group. UML couvre tous les aspects du développement d'un logiciel – spécifications, design, développement, documentation, tests, déploiement – sous forme de neuf types de diagrammes.

En UML, selon Kim [KIM 02], les trois points de vue principaux de modélisation sont : les diagrammes des use cases, les modèles statiques et les modèles dynamiques. Les diagrammes use cases décrivent les exigences du système du point de vue de l'utilisateur. Les modèles statiques sont essentiellement des diagrammes de classe qui décrivent les éléments du système et leurs interrelations. Les modèles dynamiques décrivent le comportement du système au cours du temps.

3.2 Apports et limites des méthodes SADT et UML

Individuellement, les notations graphiques de SADT ont été conçues pour modéliser une entreprise d'un point de vue fonctionnel ou informationnel. Cette caractéristique est perçue par Kim [KIM 02] comme étant à la fois une force et une faiblesse. En outre, les notations SADT sont généralement plus appropriées pour décrire, de plusieurs points de vue, ce qu'un système très complexe (comme une entreprise manufacturière) est sensé produire. SADT n'est pas approprié pour les modèles plus détaillés décrivant les comportements des objets de l'entreprise (personnes, logiciels et machines) de manière à ce que les responsabilités des activités puissent être attribuées efficacement et que les structures, composants et logiciels puissent être créés et réutilisés rapidement et efficacement. Les modèles SADT/IDEF0 assurent la consistance du système modélisé à chaque niveau de décomposition. Ils sont néanmoins statiques : ils représentent uniquement les activités du système et leurs interrelations sans faire apparaître directement les dépendances logiques et de durées entre eux [SAN 98]. Enfin, les diagrammes SADT ne précisent pas la façon dont la matière d'œuvre est transformée par la fonction (les changements d'état). UML étant un langage de modélisation et non pas une démarche de modélisation, il n'y a pas de procédure à suivre pour le développement de modèles de systèmes. Cela apporte une plus grande flexibilité mais signifie également que la modélisation par UML est du ressort d'un expert car peu d'information sémantique est inhérente au langage de modélisation. [PAI 00] a également étudié le langage UML par rapport aux principes clés de la modélisation. Il en ressort qu'UML respecte peu certains principes clés de modélisation.

Premièrement, UML ne respecte pas le principe d'unicité car les diagrammes de séquence et de collaboration sont, selon ces auteurs, sémantiquement équivalents. Ensuite, le principe de consistance pose problème car il faut toujours vérifier à l'implémentation que l'un des diagrammes UML ne contredise pas un autre diagramme. En outre, il n'est jamais possible d'avoir une vue complète représentant à la fois les activités, les objets qui participent à celles-ci ainsi que les supports aux activités. Cette vue est pourtant indispensable pour la simulation. Finalement, UML n'est pas un langage simple à utiliser. L'utilisation d'UML permet de mieux contrôler les aspects du développement d'un système et d'anticiper ses fluctuations. Il permet une meilleure communication entre les équipes d'un projet (analystes, architectes, programmeurs ...), particulièrement dans le cas de projets complexes. Cependant, UML est un langage objet graphique, il ne propose aucune démarche de modélisation.

Les deux méthodes évoquées précédemment ont chacune leurs forces et faiblesses. Nous nous en sommes inspirés pour développer les concepts sous-jacents de notre méthode de modélisation. Parallèlement, nous avons défini les objectifs de notre méthode en fonction de nos besoins et de ceux de nos partenaires. Le tableau 1 présente notre synthèse de critères (entre autres, inspirés de [MON 01] et de [TAL 02]) sélectionnés et appliqués à SADT et à UML, en comparaison avec les objectifs du CREGI en matière de modélisation.

	UML	SADT	ALIX
Apprentissage de la méthode	Difficile	Facile	Facile
Représentation graphique	Oui	Oui	Oui
Lisibilité	Faible	Bonne	Bonne
Représentation des ressources	A la demande	Simpliste	Détaillée
Gestion dynamique des ressources (pré conditions, états)	Oui	Oui	Oui
Représentation des flux	Oui	Mal adapté	Oui
Gestion dynamique des flux	Oui	Non	Oui
Représentation des règles de gestion	Oui	Par activité	Oui
Gestion temporelle	Oui	Simpliste	Oui
Orientation processus	Non	Oui	Oui
Orientation objets	Oui	Non	Oui
Adaptation à la simulation	Faible	Moyenne	Forte
Méthodologie de modélisation	Non	Oui	Oui
Domaine d'applications	Vaste	Limité	Vaste

Tableau 1 : Parallélisme avec les méthodes de référence.

Dans le cadre de nos besoins de modélisation en entreprise et en tenant compte du fait que nous souhaitons une méthode accessible aux PME au moins en terme compréhension, il nous paraît difficile d'utiliser UML pour des raisons évidentes de maîtrise pour un public non-initié. SADT par contre offre l'avantage d'une simplicité de représentation mais ne prend pas en compte l'aspect dynamique des systèmes et est dès lors difficilement exploitable pour générer

des modèles de simulation. Aussi, notre méthode tente d'allier les avantages d'une représentation orientée objets telle que UML et une simplicité de représentation à l'image de SADT.

4. Simulation : antagonisme entre flexibilité et convivialité

Notre méthode de modélisation a été conçue pour se coupler au simulateur avec lequel nous travaillons actuellement dans le cadre de nombreux projets industriels. Le choix de travailler avec RAO, notre simulateur, peut se justifier pour diverses raisons.

Il n'est pas fréquent de trouver une comparaison entre simulateurs sur les critères de flexibilité et convivialité. Cela peut se justifier par le fait que cette analyse requiert la mise en application de chaque outil sur un ou des systèmes plus ou moins complexes. Cette tâche est laborieuse et prendrait énormément de temps à finaliser. Sur base de notre connaissance en simulation, nous avons tenté de créer quatre catégories d'outils de simulation. La liste suivante les décrit par ordre croissant de flexibilité.

- Les **outils dédiés**. Ces outils, tels que les outils de gestion de centres d'appels, de systèmes de manutention, de chaîne logistique... sont destinés à être utilisés pour simuler des situations bien définies. Les objets manipulés ne sont pas des objets abstraits mais des objets appartenant au système. Ils ne sont donc applicables qu'à ce système.
- Les **outils de gestion**. Ces outils, comme FirstStep, Adonis, Corporate Modeler, sont des outils de modélisation qui permettent de simuler des systèmes à partir d'objets prédéfinis, dans les limites conceptuelles de l'outil.
- Les **outils génériques graphiques**. Ce sont des outils, tels que ARENA et AWESIM, qui proposent un certain nombre de nœuds abstraits fonctionnels à partir desquels l'utilisateur construit son modèle. Pour modéliser de situations plus complexes ou spécifiques, l'utilisateur doit soit utiliser un très grand nombre de nœuds et le schéma en perd en lisibilité, soit l'utilisateur doit être capable d'utiliser un langage de programmation pour modifier les caractéristiques des nœuds.
- Les **bibliothèques de classes**. Avec ce type d'outils (citons entre autres Simple++, C++Sim), l'utilisateur doit programmer lui-même sa classe d'objets via les bibliothèques d'objets standards. Leur utilisation nécessite la connaissance d'un langage de programmation.

Ce qui nous intéresse est de savoir ce qu'il est possible de simuler, jusqu'à quelle complexité il est possible de travailler, quelle est la flexibilité offerte par chaque outil. En effet, la flexibilité et la convivialité des outils de simulation sont des caractéristiques souvent incompatibles. La flexibilité d'un outil est sa souplesse, la qualité à simuler toute situation, qu'elle soit élémentaire ou très complexe. La convivialité décrit la qualité d'un outil à être facile

à utiliser. Il est souvent difficile de trouver un outil alliant ces deux caractéristiques. Les outils complètement dédiés sont très simples à utiliser et conviviaux, tant que l'on reste dans les limites dans lesquelles ces outils sont destinés à être utilisés. Pour dépasser ces limites, il faut se lancer dans une programmation beaucoup plus ardue. Par contre, bien souvent, les outils qui offrent plus de souplesse et permettent de simuler une plus grande variété de systèmes sont moins graphiques et dès lors moins faciles à utiliser.

La première raison de bâtir notre graphisme de modélisation ALIX sur le simulateur RAO est que celui-ci est depuis longtemps utilisé au sein de notre équipe, maîtrisé et validé dans les projets industriels que nous avons traités. La seconde est sa flexibilité. La logique de RAO rend possible la simulation de tout système à événements discrets. RAO n'est pas dédié, il n'a pas de contraintes spécifiques à un système donné. Il est basé sur les systèmes experts et sur un moteur d'inférence. Son principe se fonde sur des règles de production modifiées de type « Si... , alors.... » (cf. figure 3).

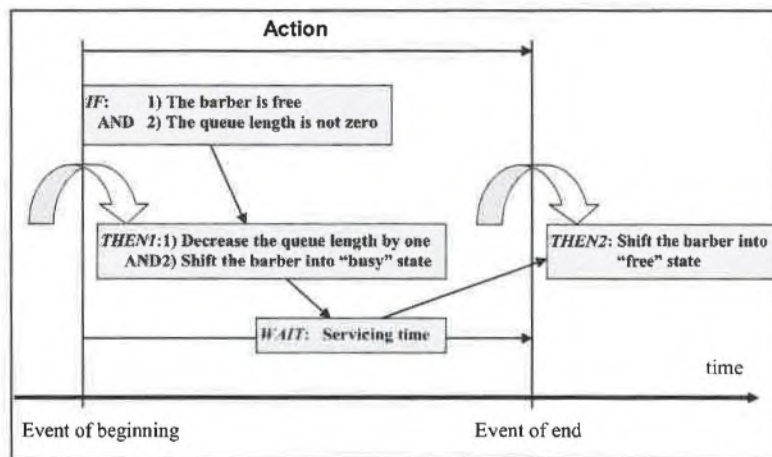


Figure 3 : Règle de production du simulateur RAO.

Celles-ci engendrent les concepts d'états et de préconditions, associés aux objets et aux activités. Cela permet de représenter le flux de façon dynamique, en faisant évoluer les objets tout au long de leur progression à travers les activités qui les transforment.

Enfin, RAO intègre également un moteur d'optimisation. Le point faible du simulateur est son manque de convivialité. C'est la raison d'être de notre projet de création d'une plateforme intégrant la modélisation graphique ALIX, la simulation et l'optimisation.

ALIX mariée à RAO est aussi flexible que UML et SADT car rien n'est prédéfini dans les objets et les activités. Cela enrichit l'ensemble des systèmes qu'il est possible de modéliser. Par contre, ils permettent, ensemble, de simuler le modèle construit, ce que ne permet pas SADT. Et

enfin, ALIX est une méthode de modélisation graphique simple à l'utilisation et à la lecture, plus facile à utiliser en tant qu'outil de communication que UML.

5. Description de ALIX, *Advanced Language for Integration and Exchanges*

Nous avons développé le formalisme de modélisation ALIX et la démarche de modélisation associée en fonction de nos objectifs ainsi que de notre simulateur : RAO. Ce formalisme graphique s'adresse à toute organisation, tout système socio-économique donné et, étant assorti d'une démarche de modélisation, il s'adresse également à tout utilisateur. Il est intuitif et simple d'interprétation. Enfin, ce formalisme convient particulièrement pour des études ayant recours à la simulation.

Notre approche dynamique de la modélisation s'articule autour de deux éléments : les activités et les objets.

Les activités. Une activité est un processus élémentaire indivisible composé de un ou de deux événements. Son exécution mobilise des ressources, du temps et des compétences et a pour effet de transformer un état d'entrée (ou objets d'entrée) en un état de sortie (ou objets de sortie). L'activité est caractérisée par un nom, un ou des objet(s) entrant(s) et sortant(s) et un ou des objet(s) de support (en quantité illimitée) ainsi qu'un numéro hiérarchique. Nous avons défini trois types d'activités (cf. figure 4) et une méta-activité.

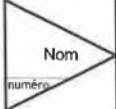
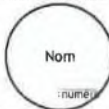
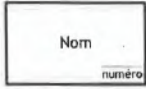

<p><u>événement irrégulier</u> :</p> <p>activité qui s'exécute à intervalles de temps définis, aléatoires ou pas, indépendamment de l'état du système (un seul événement).</p>	
<p><u>règle</u> :</p> <p>activité d'ordre décisionnel ; elle modifie l'état des objets mais n'a pas de durée (un événement).</p>	
<p><u>action</u> :</p> <p>activité qui transforme des objets en modifiant leur(s) état(s) ; elle est caractérisée par sa (ses) précondition(s), une certaine durée et est bornée par deux événements.</p>	
<p><u>méta activité</u> :</p> <p>formalisme graphique permettant d'englober des sous processus particuliers et, de cette manière, permet de décomposer hiérarchiquement le schéma global.</p>	

Figure 4 : Les différents types d'activité en ALIX.

L'**événement irrégulier** permet de modéliser et de simuler, par exemple, une horloge, l'arrivée de commandes, l'arrivée de clients...

La **règle** est utilisée, de manière simple, pour une prise de décision, par exemple, en tant que règle d'affectation d'un ordre de fabrication à un atelier en particulier, un type de produit à un type de machine défini. Le moteur d'inférence du simulateur permet d'aller aussi plus loin dans l'utilisation de la règle : on peut aller jusqu'à décrire un système expert hiérarchisé.

L'**action** est illustrée dans la figure 5. L'action sert à décrire des activités qui ont une certaine durée, par exemple, une opération de production. Il y a donc deux éléments importants à décrire : ce qu'il se passe au début de l'action et à la fin de l'action.

La figure suivante représente l'action A n° 1 qui modifie l'état d'un objet entrant A et le fait passer du Stade 1 au Stade 2. A la fin de cette action, un objet B, dont l'état est « créé » est produit. On peut également représenter les ressources nécessaires à l'exécution de cette action, les ressources supports, support Z et support Y.

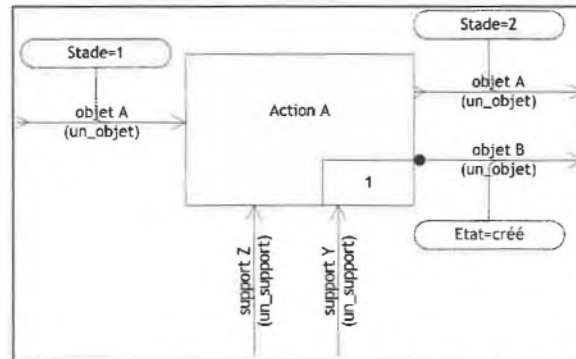


Figure 5 : Présentation générale d'une activité.

Les **objets**. Un objet est caractérisé par des attributs et des méthodes. Dans le cadre de notre approche orientée processus et simulation, l'objet matérialise les ressources d'un système de production. Deux types d'objets ont été mis en évidence :

- les objets « transformés » qui traversent et qui, de par leur état, font évoluer le système ;
- les objets « utilisés », « consommés » pour supporter les activités (« objets supports »).

Un objet est matérialisé par une flèche accompagnée des informations suivantes : son nom, le nom de son type ou de l'instance. Dans le cas d'un objet transformé, il faut également y trouver son état (précondition) et un changement d'état (optionnel). Un objet « support » est identifié par une flèche verticale dirigée vers une activité et se trouvant sous celle-ci.

Les états et préconditions représentés sur les liens sont essentiels pour la modélisation. Ce sont ces éléments qui apportent la dynamique du modèle. Via ce composant, on identifie la dynamique du flux. Il n'est pour autant pas nécessaire de surcharger les schémas des attributs

de l'objet qui n'apportent rien à la compréhension du processus. Sont donc uniquement représentés dans le modèle graphique, les états et préconditions essentiels à la compréhension du modèle.

Lorsqu'un système est très complexe et constitué d'activités secondaires (transport, opérations de set-up,...) récurrentes, il est possible de simplifier le schéma pour ne représenter que les activités génératrices de valeur ajoutée. Dans ce cas, le modèle consiste en un flux « allégé » ainsi que des activités indépendants, secondaires, représentées alors une seule fois dans le modèle. Ceci n'est possible que via l'utilisation des concepts d'états et de préconditions.

Les liens. Le lien de base entre deux activités représente le parcours d'un objet transformé par une activité. Ce principe est illustré en figure 6. On peut y voir que l'objet principal quitte l'activité n° 1 dans un certain état (Etat = A), ce qui déclenche la réalisation de l'activité n° 2, pour autant que toutes les conditions soient remplies.

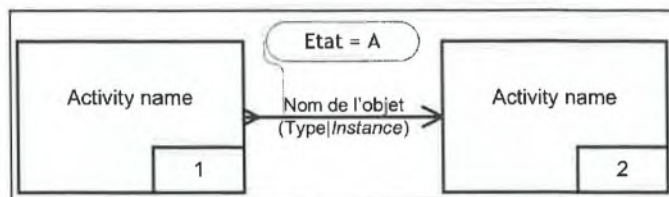


Figure 6 : Lien simple entre deux activités.

C'est sur le lien que nous représentons la création et la destruction des objets comme le montre la figure 7.

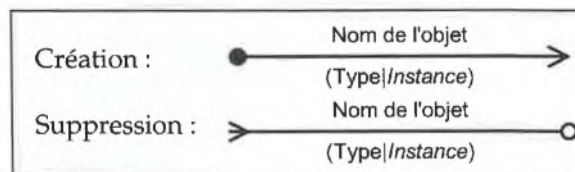


Figure 7 : Création et suppression d'un objet.

Le lien simple peut être adapté afin de réaliser des branchements et des fusions entre les activités. Une autre possibilité qu'offre ALIX est le système de bouclage. Une ressource peut passer plusieurs fois par la même activité tant que son état remplit la précondition de l'activité. Cependant, pour éviter de boucler à l'infini, il est nécessaire qu'une propriété de la ressource change et ne satisfasse plus la précondition de réalisation de l'activité. Enfin, afin d'alléger les schémas, nous proposons aussi un composant de rupture permettant de rompre une flèche et de la reprendre plus loin dans le schéma à la condition d'utiliser le symbole de rupture et la numérotation adéquate.

La méta activité. La méta activité consiste en un regroupement d'un certain nombre d'activités par souci de clarté et de facilité de modélisation. Elle est représentée en figure 8. Un zoom peut être réalisé afin de décrire plus précisément le contenu de la méta activité.

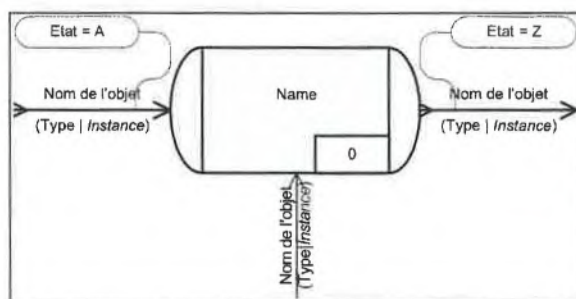


Figure 8 : Une méta activité en ALIX.

L'interface de modélisation. Pour tester et valider le développement de cette méthode de modélisation, nous avons décidé d'utiliser le logiciel Microsoft VISIO, notamment pour sa facilité d'utilisation. Premièrement, il nous permet de définir des blocs, des liens et de les réutiliser autant de fois que souhaité en paramétrant certaines informations spécifiques. Deuxièmement, il est possible de récupérer toutes ces informations et de les transcrire dans une base de données. Enfin, il possède un langage de programmation Visual Basic pour la génération automatique du langage de simulation.

6. Du modèle d'entreprise au modèle de simulation

Le modèle d'entreprise créé via ALIX, dont la première utilisation peut se faire en tant qu'outil de communication, offre par la suite la possibilité d'être exploité en tant que langage de simulation. Pour illustrer nos propos, reprenons un cas standard de pilotage qui vise à attribuer une pièce à un opérateur selon des contraintes variées (compétences, caractéristiques techniques, contraintes organisationnelles). La modélisation de cette règle d'affectation en ALIX est présentée dans la partie gauche de la figure 9.

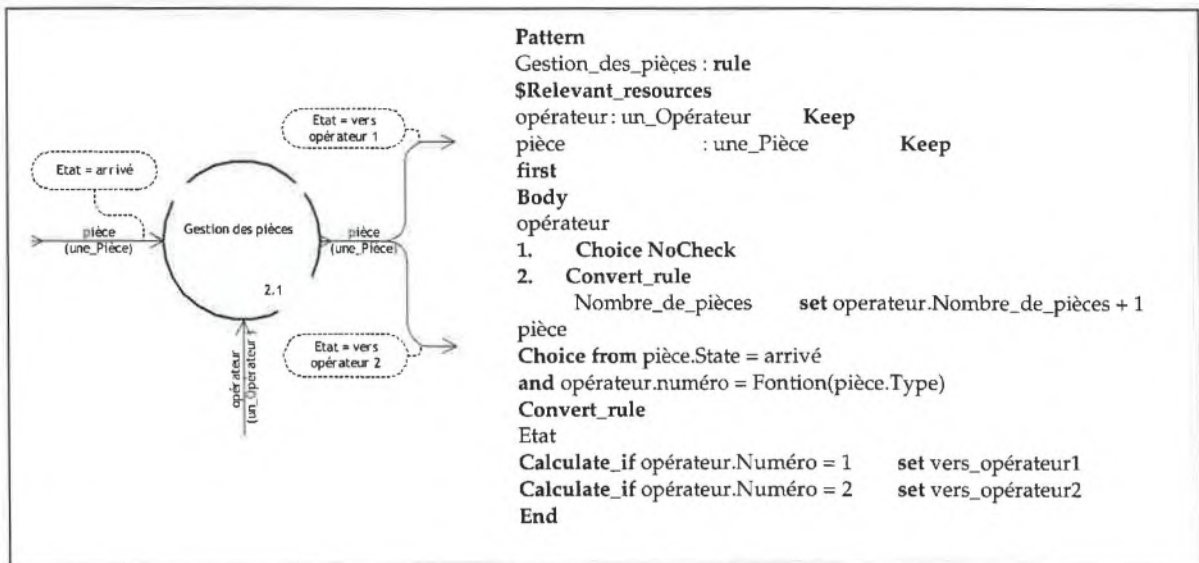


Figure 9 : Règle d'affectation de pièces à un opérateur en ALIX.

Lorsqu'une pièce arrive à l'activité Gestion des pièces (règle), elle est affectée à un opérateur en tenant compte des corrélations entre les deux types d'objets qui sont précisés dans une fonction appelée Fonction (pièce.Type). Un paramètre inhérent aux opérateurs est incrémenté afin de compter les pièces traitées et l'état des pièces est modifié. Ces paramètres sont directement accessibles par l'interface VISIO, en cliquant sur l'objet concerné. De plus, le stockage de toutes les informations du schéma dans une base de données permet de générer de façon automatique le langage de simulation RAO que nous utilisons [ART 98].

Le résultat de cette génération est présenté dans la partie droite de la figure 9. La règle en ALIX devient un « pattern » de type « rule » en RAO. Deux types de ressources sont impliqués dans ce pattern : un opérateur et une pièce. Ces types ont été générés dans un autre fichier spécifique qui n'est pas présenté ici. Pour les ressources de type pièce, on précise des pré conditions (arrivée de la pièce et corrélation avec l'opérateur via la fonction Fonction (pièce.Type)).

7. Applications

Notre méthode de modélisation est en cours de validation auprès de plusieurs entreprises industrielles dans le cadre d'analyses et de réorganisations des flux de production.

A ce jour, tous les industriels auxquels nous avons présenté la méthode et les modèles correspondants ont très vite compris le principe et l'ont adopté comme outil de communication.

De plus, la phase de validation de ce modèle par l'entreprise est grandement facilitée par la représentation graphique ALIX.

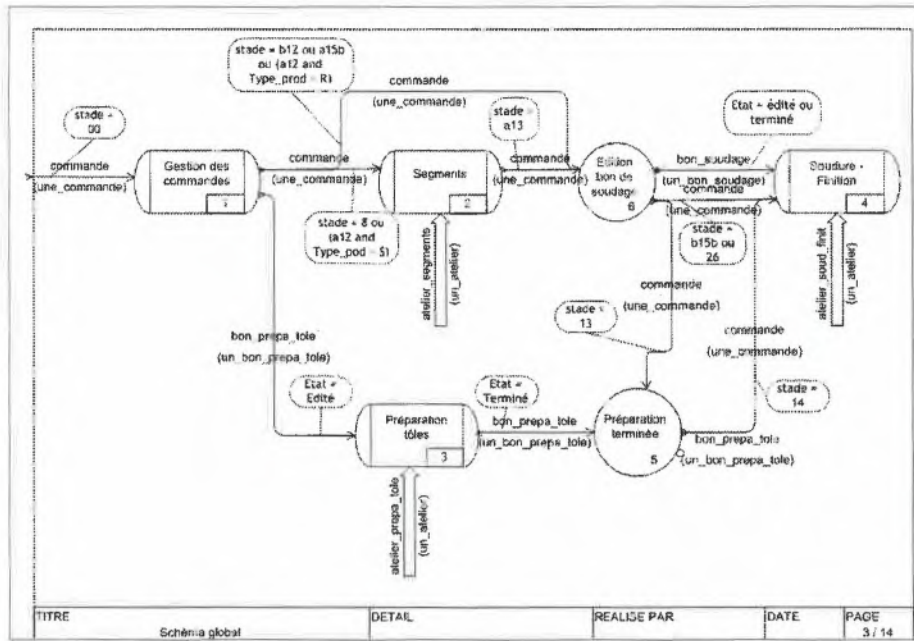


Figure 10 : Schéma global d'un cas d'application en ALIX.

La figure 10 présente un schéma global d'un des cas industriels. Cette modélisation a pu être réalisée en quelques jours, ce qui est très peu par rapport à la complexité du système étudié. Une dizaine de pages décrivent en détails les processus de fabrication et les flux qui traversent le système. A partir de ces schémas, un modèle de simulation a été généré dans le but d'analyser ces flux.

8. Conclusions et perspectives

Les applications ont mis certaines lacunes d'ALIX en évidence. Nous pouvons citer par exemple, le manque de facilité de représentation des processus continus. Cependant, il a été très rassurant de voir à quel point les industriels qui ont été confrontés à notre méthode y ont rapidement adhéré.

ALIX a ainsi été très utile en interne pour la formalisation et la représentation des processus au sein de l'entreprise. Cela a conforté notre vision de la modélisation graphique en tant qu'outil de communication entre les mondes de la recherche et de l'entreprise et a satisfait aux objectifs que nous nous étions fixés en matière de facilité d'utilisation et d'interprétation. En outre, ce formalisme a permis de décrire toutes les situations rencontrées dans le cadre de nos projets industriels.

Des pistes de réflexion et des perspectives d'amélioration sont à l'étude. Citons notamment la modélisation des processus semi continus, les contextes statique et décisionnel et l'intégration des méthodes d'optimisation. Les processus semi continus peuvent être représentés de manière discrète via une action dont la durée ainsi que les valeurs des variables continues seraient calculées de manière algébrique via des équations différentielles. En ce qui concerne les contextes statique et décisionnel, le formalisme graphique doit être complété afin de pallier à ce manque. Au sujet des méthodes d'optimisation, d'autres membres de l'équipe travaillent actuellement sur les méthodes d'optimisation qui seront intégrées à notre plateforme.

9. Références

- [ART 98] A. ARTIBA, V.V. EMELIANOV, S.I. IASSINOVSKI, *Introduction to Intelligent Simulation : the RAO language*, Kluwer Academic Publishers, Boston, 1998.
- [BEN 98] A. BEN HAMADOU ET F. GARGOURI, *Développement de logiciels - L'approche objet*, Centre de Publication Universitaire, ISBN: 9-973-93741-4, Tunis, 1998.
- [BOO 99] G. BOOCH, J. RUMBAUGH ET I. JACOBSON, *The Unified Modeling Language - User Guide*, Addison Wesley, ISBN: 0-201-57168-4, Massachusetts, USA, 1999.
- [CAR 01] A. CARDON ET C. DABANCOURT, *Initiation à l'algorithmique objet - Modélisation avec UML et exemples de code en Java et C++*, Eyrolles, Paris, ISBN: 2-212-09258-X, 2001.
- [FOX 98] M. S. FOX, M. GRUNINGER, *Enterprise Modelling*, The American Association for Artificial Intelligence, AI Magazine, 19(3), pp. 109-121, 1998.
- [HAI 00] J.-L. HAINAUT, *Bases de données et modèles de calcul*, Dunod, ISBN: 2-100-04553-9, Paris, 2000.
- [IAS 01] S. IASSINOVSKI, A. ARTIBA, V. BACHELET, F. RIANE, "Integration of Simulation and Optimization for Solving Complex Decision Making Problems", Proceedings of the *International conference on industrial engineering and production management (IEPM'2001)*, pp. 1186-1195, Quebec City, Canada, August 20-23, 2000.
- [KAL 01] B. KALPIC, P. BERNUS, "Business process modelling in industry-the powerful tool in enterprise management", Elsevier, *Computers in Industry*, Volume 47, Issue 3, Pages 299-318, March 2002.
- [KIM 02] C.-H. KIM, R. H. WESTON, A. HODGSON, K.-H. LEE, "The complementary use of IDEF and UML modelling approaches", *Computers in Industry*, Elsevier, Volume 50, Issue 1, Pages 35-56, January 2003.
- [LAN 97] R. LANDRY, M. SANTERRE, "Méthodes de simulation en science politique." Disponible sur le site <http://cours.fss.ulaval.ca/pol-15429/materiel/reference/index.html>.
- [LIS 01] M. LISSANDRE, *Maîtriser SADT*, Éditions Armand Colin, 2001.
- [MON 01] T. MONTEIRO, "Conduite distribuée d'une coopération entre entreprises : le cas de la relation donneurs d'ordres-fournisseurs", Thèse du Laboratoire d'Automatique (LAG) de l'Institut National Polytechnique de Grenoble, 2001.
- [MOR 00] C. MORLEY, J. HUGUES ET B. LEBLANC, *UML pour l'analyse d'un système d'information*, Dunod, Paris, ISBN: 2-100-04826-0, 2000.

- [PAI 00] R. F. PAIGE, J. S. OSTROFF, P. J. BROOKE, "Principles for modelling language design. Information and Software Technology", Elsevier, Volume 42, Issue 10, pp. 665-675, July 2000.
- [ROS 77] D. T. ROSS, "Structured Analysis (SA) : a language for communication ideas". *IEEE Transactions on Software Engineering*, vol. 3, pp. 16-34, 1977.
- [SAN 98] K. SANTAREK, I. M. BUSEIF, "Modelling and design of flexible manufacturing systems using SADT and Petri net tools". *Journal of Materials Processing Technology* 76, pp. 212-218, 1998.
- [TAL 02] A. TALBI « Contribution à l'intégration des fonctions de l'entreprise. Application aux fonctions production et maintenance. », Thèse de l'Université Mohammed V à Rabat (Maroc), 2002.
- [TAR 00] H. TARDIEU, A. ROCHFELD ET R. COLLETTI, *La méthode Merise - Principes et Outils*, Editions d'Organisation, Paris, ISBN: 2-708-12473-0, 2000.
- [VER 99] F. VERNADAT, *Techniques de modélisation en entreprise : applications aux processus opérationnels*, Collection Gestion, Editions Economica, Paris, 1999.
- [VER 01] F. VERNADAT, "UEML : Towards a Unified Enterprise Modeling Language." *Actes de la Conférence MOSIM'01*, pp. 3-11, avril 2001, Troyes (France).