

UNE PLATE-FORME DE MODÉLISATION ET SIMULATION POUR LES SYSTÈMES HOSPITALIERS

V. Augusto, X. Xie

Centre Ingénierie et Santé
École Nationale Supérieure des Mines de Saint-Étienne
158 cours Fauriel 42023 Saint-Étienne cedex 2
augusto@emse.fr, xie@emse.fr

RÉSUMÉ : *Les méthodes de modélisation et de simulation actuelles ne sont pas adaptées aux systèmes hospitaliers. Pour remédier à cela, nous présentons dans cet article une nouvelle méthodologie de modélisation pour la résolution de problèmes d'organisation en milieu hospitalier. Il s'agit d'un méta-modèle centré sur le patient, décomposé en trois vues : (i) Organisation (relations entre ressources, spécialisations, structure du système), (ii) Ressource (activité de chaque ressource), et (iii) Processus (flux de patients et/ou de produits). Ces vues personnalisées s'inscrivent dans une architecture classique basée sur une décomposition systématique du système. Le formalisme de la modélisation et les algorithmes de vérification et de simulation sont décrits dans ce papier. Le tout forme un puissant outil de communication grâce à un formalisme graphique épuré basé sur UML. Tout méta-modèle peut être instancié pour une unité médicale spécifique et immédiatement simulé. Nous proposons ainsi un outil de prototypage rapide pour les systèmes hospitaliers.*

MOTS-CLÉS : *modélisation, simulation, hospitalier, outil, méta-modèle, vue, medPRO*

1. INTRODUCTION

La recherche dans le domaine du génie industriel hospitalier prend aujourd'hui une nouvelle direction, s'orientant vers la conception de modèles et d'outils dédiés spécifiquement à ces systèmes. Plusieurs états de l'art en la matière (Jun, Jacobson & Swisher 1999, Augusto & Xie 2006) révèle que les outils de modélisations utilisés en industrie ne sont pas adaptés aux systèmes hospitaliers. En effet, un grand nombre de particularités les distinguent des systèmes manufacturiers classiques :

- observation et modélisation de flux de patients, et non de produits (prise en compte du comportement des entités) ;
- intervention d'un grand nombre de ressources humaines très variées ;
- organisation spécifique par service de soins, reposant sur une coordination précise des ressources humaines et matérielles ;
- pilotage du système essentiel afin de réagir rapidement aux aléas.

La différence majeure entre les systèmes hospitaliers et manufacturiers réside dans la nature des flux qui les constituent. Les études menées en milieu hospitalier ont souvent pour objet le patient et son parcours au travers d'une ou plusieurs unité(s) de soins qui doit être analysé et simulé. Or le parcours patient dépend de nombreux facteurs : dossier médical et pathologie, disponibilité de ressources (lits, médecins), évolution de la pathologie, réaction du patient à la maladie... Ces facteurs influent grandement sur l'issue du parcours patient et doivent être pris en compte dans le cadre d'une modélisation précise. De tels éléments sont inexistant dans les systèmes manufacturiers.

Ainsi la plupart des études existantes reposent sur une modélisation spécifique au problème étudié : il n'existe pas de modèle générique dédié aux systèmes hospitaliers qui encapsule les caractéristiques listées ci-dessus. Pourtant, les problématiques sont souvent similaires, et portent essentiellement sur l'organisation ou la réorganisation de services de soins. La simulation est par ailleurs un outil très utilisé, permettant la modélisation de flux complexes et l'obtention rapide de résultats.

Nous proposons dans ce papier une nouvelle plateforme permettant la modélisation et la simulation de systèmes hospitaliers. Un bref état de l'art est présenté section 2. L'objet de cette étude est présenté section 3. La plate-forme elle-même est présentée dans les sections 4, 5 et 6. Les procédures de vérification et de simulation sont décrites section 7. Enfin, un exemple est présenté section 8.

2. ÉTAT DE L'ART

Cette brève revue de littérature est décomposée en plusieurs domaines. De nombreuses études de cas nécessitent une modélisation préalable du système, un **outil de modélisation d'entreprise** doit donc être utilisé. Cependant, afin d'éviter la répétition du processus de modélisation, plusieurs études s'orientent vers la notion de **modèle générique**, alors que d'autres études présentent la mise en œuvre de **plate-formes de modélisation/simulation dédiées aux systèmes hospitaliers** afin de répondre à une demande de plus en plus rigoureuse. Ces outils doivent permettre une modélisation facilitée et précise d'un service en particulier ou de l'hôpital en général. La méthodologie et la plate-forme présentées dans cet article appartiennent à cette dernière catégorie.

2.1. Outils de modélisation pour l'industrie

Les outils de modélisation dédiés à l'industrie sont nombreux (Vernadat 1999), les plus connus étant IDEF0/3, CIMOSA, GRAI, ARIS et UML. Ces outils présentent un certain nombre de caractéristiques communes : (i) modélisation organisée autour de vues et de perspectives, (ii) présence de concepts de modélisation, (iii) méthodologie structurée, permettant un accompagnement à la modélisation, et (iv) représentation de flux selon leur type (matériel, information, contrôle).

Fixer les objectifs et les enjeux de l'étude est primordial car ces paramètres influenceront sur la modélisation du système. L'identification des destinataires (demandeurs et personnel touché par l'étude) doit permettre l'établissement d'un modèle adapté aux besoins de tous. Les limites de ces outils ont été identifiées : la modélisation de comportements humains n'est pas prise en compte et la notion de système de pilotage reste vague. Enfin, la transition modélisation-simulation n'est pas claire dans ces méthodologies, laissant une marge de manœuvre importante et surtout risquée à l'utilisateur.

Trois outils de modélisation (SADT, GRAI et ARIS) ont été testés pour la modélisation de systèmes hospitaliers et comparés dans (Trilling, Besombes,

Chaabane & Guinet 2004) selon une sélection de critères jugés représentatifs. Les avantages et inconvénients de chacune de ces méthodes y sont discutés.

2.2. Modèles génériques

Un outil permettant la simulation de centres hospitaliers destiné à des utilisateurs non informaticiens a été présenté dans (Moreno, Aguilar, Martín, Pineiro, Estévez, Sigut, Sánchez & Jiménez 1999). Deux fonctions sont implémentées : une fonction d'observation (extraction de données concernant l'état de l'hôpital), et une fonction de simulation (prévisions sur les ressources à affecter dans les différents services de l'hôpital). La modélisation est centrée sur le patient et son évolution au cours de son hospitalisation. Chaque patient (entité) est unique, caractérisé par un certain nombre d'attributs. Dans l'optique d'une généralisation du modèle, une approche orientée objet a été adoptée, allant de pair avec le choix de l'environnement de simulation (ModSim). Peu de détails sont donnés au sujet de la modélisation UML réalisée et le détail inhérent à la modélisation d'un hôpital n'est pas encore assez poussé pour satisfaire l'acuité des résultats attendus.

Plus récemment, une méthodologie de modélisation pour la simulation basée sur des méta-modèles génériques (MDA-UML) a été proposée dans (Roux, Combes & Duvivier 2006). Ces méta-modèles sont définis de manière à modéliser une classe de système (ici, la classe du système « hôpital »). Le modèle obtenu est ensuite instancié, permettant la génération semi-automatique d'un modèle de simulation représenté sous forme de machines à états. Une partie de la méthodologie est illustrée sur un cas d'étude mettant en œuvre une unité de dialyse. Le système de décision est en cours de développement, ainsi qu'une interface graphique pour la simulation.

Enfin, un modèle de connaissance générique est présenté dans (Chauvet, Fénies & Gourgand 2007), permettant la modélisation et la simulation de plusieurs services hospitaliers, tel le service des urgences pédiatriques par exemple. La connaissance est structurée de manière unique suivant la méthodologie ASCI (Analyse, Spécification, Conception, Implantation) (Gourgand & Keller 1991). Les processus ont ainsi été formalisés dans un modèle de connaissance ARIS. Le modèle de simulation a été réalisé sous Witness.

2.3. Plate-formes de modélisation et simulation

Une plate-forme de modélisation appelée medBPM est décrite dans (Ramudhin, Chan & Mokadem 2006), permettant de prendre en compte la complexité des

processus hospitaliers. Dépendances et synchronisation entre ressources, informations et matériels au cours des processus sont pris en compte. Cette plateforme est centrée sur les ressources, et utilise un formalisme graphique simple permettant un échange facilité avec le personnel hospitalier. Un module de simulation est en cours de développement. medBPM a été développé sous Microsoft Visio et utilisé dans un hôpital aux États-Unis pour l'étude et la modélisation du système de distribution de médicaments (pharmacie).

Une plateforme de simulation multi-agents dédiée aux systèmes hospitaliers est introduite dans (Montreuil, Charfeddine, Labarthe & Côté 2007). En cours de développement, cette plateforme générique propose un guide de référence pour construire le modèle d'un système hospitalier. Trois éléments structurants principaux sont mis en avant : agents, objets, environnement et expérience. Ce projet doit constituer une aide permettant d'accélérer le processus de modélisation tout en minimisant les risques d'oubli d'éléments importants et d'interactions.

3. OBJET DE L'ÉTUDE

Nous proposons dans cet article une méthodologie de modélisation originale dédiée aux systèmes hospitaliers, présentée figure 1. Nous construisons à partir d'une analyse préliminaire du domaine hospitalier un cadre de modélisation appelé medPRO (medical Process-Resource-Organisation modeling) : il s'agit d'un ensemble de briques de base encapsulant les caractéristiques d'une unité médicale particulière. Une *plate-forme de modélisation* et un *guide méthodologique* permettent alors la modélisation de cette unité, afin de répondre à un problème donné. La simulation découle immédiatement de la modélisation, permettant d'apporter rapidement une réponse aux problèmes formulés par le demandeur. Le *guide méthodologique* regroupe toutes les consignes de modélisation relatives à un service de soin particulier : il s'agit par exemple de recommandations de bonnes pratiques dans le cas du service de pharmacie d'un hôpital, de consignes de prise en charge du patient, etc. Ce guide permet de poser les premiers repères pour aider l'utilisateur au cours de la modélisation, réalisée grâce à la *plate-forme* proposée.

L'outil présenté dans ce papier s'adresse aux chercheurs, ingénieurs ou chargés d'étude désireux de modéliser rapidement et efficacement un service médical particulier. Le processus de modélisation est facilité, permettant à l'utilisateur de minimiser les risques de conflits, d'erreurs ou d'oublis. Nous avons choisi de construire une plate-forme à partir de zéro afin de proposer un outil parfaitement adaptés aux besoins

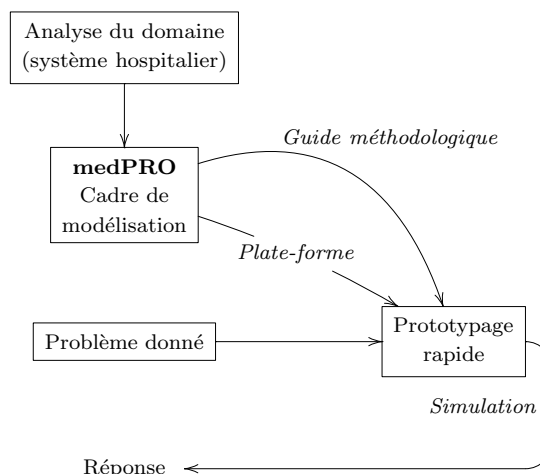


Figure 1: Méthodologie globale

de la modélisation de flux en milieux hospitaliers, encapsulant les caractéristiques listées en introduction. Cette plate-forme a été conçue de manière à être aussi flexible que possible selon l'unité médicale étudiée.

L'architecture de medPRO est la suivante : comme pour les systèmes manufacturiers, une décomposition systémique nous permet de diviser un système hospitalier en trois sous-systèmes : opérationnel (physique), informationnel et décisionnel (pilote) (Gourgand & Keller 1991). Le sous-système de pilotage est un ensemble de centres de décision conçus pour piloter le sous-système opérationnel. Le sous-système informationnel fait le lien entre le sous-système physique et le sous-système de pilotage ; il regroupe et ordonne toutes les données techniques (coûts, variables d'état, dossiers, etc.) qui caractérisent le système modélisé, et permet au sous-système décisionnel de gérer et de piloter le sous-système physique. Une telle décomposition s'applique parfaitement aux systèmes hospitaliers où l'opérationnel, l'informationnel et le décisionnel sont trois aspects fondamentalement distincts. De nombreuses données de natures très différentes caractérisent un système médical. Ces données sont généralement éparpillées sur des supports très différents. De plus, une partie de ces données concernent le patient et sont confidentielles (dossier patient, prescriptions), donc difficile d'accès. Ainsi la récolte, le classement et la protection de ces données sont autant d'étapes indispensables à la modélisation du système. De la même façon, le pilotage du système est un élément indispensable à son bon fonctionnement : dans un environnement hautement stochastique tel que l'hôpital, il est nécessaire de prévoir et de s'adapter à la demande. Le traitement d'événements aléatoires en fait également partie. Le système physique qui regroupe la logistique, le par-

cours patient et l'organisation du système interagit continuellement avec les deux autres parties.

Nous superposons à cette architecture un habillage dépendant du système étudié, permettant à l'utilisateur de retrouver lors de la modélisation un certain nombre d'éléments qui lui seront familiers. Dans le cas du service de pharmacie par exemple, un certain nombre d'organisations pourront être proposées en amont de la modélisation, permettant de poser les bases du projet : mode d'approvisionnement des armoires, mode de distribution des médicaments nominaux, etc.

4. SOUS-SYSTÈME PHYSIQUE

Le sous-système physique est décomposé en trois vues distinctes : organisation, ressource et processus. Une *vue* est une représentation du système sous un certain angle, permettant la mise en valeur de certains aspects ou de certaines particularités. Ainsi la vue organisation offre une représentation globale du système, regroupant les déclarations des entités et des ressources, leurs relations, leurs effectifs, etc. La vue ressource permet de décrire pour chaque ressource (humaine ou non) son mode opératoire, i.e. toutes les activités auxquelles elle pourra participer et leur enchaînement. Enfin, la vue processus regroupe les flux parcourus par les entités déclarées dans la vue organisation et leurs interactions avec les ressources du système.

4.1. Vue Organisation

La vue organisation propose une représentation globale du système, et comporte plusieurs sous-vues permettant d'ordonner les différents diagrammes qui la compose. Nous définissons la notion d'organisation de la manière suivante : il s'agit d'une représentation des interactions entre les acteurs du systèmes, les ressources matérielles et les patients, permettant de comprendre le fonctionnement global du système modélisé.

4.1.1 Déclarations

Tous les objets qui interagissent dans le système représenté doivent être déclarés dans la vue organisation. Bien que regroupés sous l'étiquette *agent* durant la simulation, les déclarations sont distinctes et concernent les *entités* et les *ressources*. Ces déclarations sont représentées sous la forme de diagrammes de classes distincts. Les objets déclarés ne sont pas liés entre eux dans cette représentation.

Les entités circulent dans les flux représentés dans la vue processus. Une entité possède un nom unique et est associée à une table dans le sous-système infor-

mationnel. Cette table possède un certain nombre de champs permettant de décrire les caractéristiques de cette entité. Dans le cas d'un patient, il s'agirait par exemple de son nom, prénom, etc. Les ressources circulent dans les flux représentés dans la vue ressource. Pour chaque ressource déclarée, un diagramme d'état doit être créé dans la vue ressource, permettant de modéliser le comportement de cette dernière. Une ressource possède un nom unique et un effectif. De plus, une ressource est associée à deux tables dans le sous-système informationnel : la première permet à l'instar de l'entité de décrire les caractéristique de la ressource ; la deuxième est une table de type « emploi du temps », permettant de décrire les horaires et la fréquence de travail de la ressource.

4.1.2 Spécialisations

Les notions de spécialisations et de savoir faire sont courantes dans le milieu médical. Ces notions permettent de classer les acteurs du système en fonction de leurs compétences. La notion d'héritage, propre à UML, nous permet de modéliser cet aspect caractéristique des systèmes hospitaliers et est définie de la manière suivante. D'un point de vue pratique, l'héritage offre à l'utilisateur une plus grande souplesse dans la modélisation et l'affectation de ressources. Brièvement, la réalisation d'une action par une entité nécessite au moins une ressource (ce mécanisme sera décrit dans la section 4.3). Si cette ressource possède un ou plusieurs héritiers, l'action peut implicitement être réalisée par cette ressource ou par l'un de ses héritiers, direct ou indirect. Le choix de la ressource dépend alors des directives données au sous-système de pilotage. Par exemple, si l'entité de type « Patient » doit subir une opération de type « Ophtalmologie », le choix de la salle opératoire se portera directement sur celle équipée du matériel spécifique à cette spécialité. Les consignes de ce type font partie du *guide méthodologique*, car elles sont communes à tous les blocs opératoires.

4.1.3 Associations

Les diagrammes d'associations permettent de lier les ressources entre elles afin de les regrouper. Il est ainsi possible de former des ensembles de ressources (ou équipes, dans le cas de ressources humaines) afin de reproduire l'organisation du système étudié et modéliser les relations privilégiées entre certains acteurs. Nous introduisons par la même occasion la notion de visibilité, permettant de définir le champ d'action d'une ressource.

Le regroupement de ressources permet notamment de simplifier la modélisation d'actions communes à plusieurs ressources. En outre l'association entre

plusieurs acteurs du système permet la modélisation de réseaux de confiance. Ces aspects relationnels très utilisés en simulation multi-agents n'ont pas encore été exploités dans cette étude.

4.1.4 Appartenance et dépendance

Ce lien permet de modéliser l'appartenance (ou la dépendance, selon le type de ressource) d'une ressource à une autre. Nous utilisons la relation de composition pour représenter ce lien. En pratique, cette notion permet d'introduire des contraintes de compatibilité entre ressources lors de la modélisation. Il sera alors possible d'interdire l'utilisation simultanée de certaines ressources pour une même tâche. De la même manière, l'appartenance permet la modélisation de l'organisation physique du système : lits, salles de consultation, unités de soins, etc.

4.2. Vue Ressource

La vue ressource regroupe les diagrammes d'états associés aux ressources déclarées dans la vue Organisation. Il existe exactement un diagramme d'état pour chaque ressource. Ces diagrammes d'états sont construits selon le formalisme d'UML ; plusieurs modifications ont été apportées afin de simplifier l'utilisation et la vérification de ces diagrammes. Les diagrammes d'états utilisés dans ce modèle comportent un nombre limité de blocs différents : blocs d'entrée et de sortie, bloc d'état et bloc d'attente.

Un diagramme d'états comporte un unique **bloc d'entrée**, représenté par un disque. Ce bloc dispose d'un seul point de sortie, et permet de générer les agents symbolisant la ressource considérée. Ce bloc ne nécessite donc aucun paramétrage étant donné que la table décrivant l'emploi du temps de la ressource est précisée lors de sa déclaration dans la vue Organisation. Ainsi, un bloc d'entrée permettra de générer un nombre d'agents égal à l'effectif de la ressource considérée aux dates de début de service. Symétriquement, un diagramme d'état comporte un unique **bloc de sortie**, représenté par un disque entouré. Ce bloc dispose d'un seul point d'entrée, et permet de détruire les agents qui ne sont plus disponibles. Ce bloc ne nécessite aucun paramétrage.

Les **blocs d'état** permettent de décrire l'activité de la ressource modélisée. Un bloc d'état comporte un point d'entrée et un point de sortie, et n'est pas décomposable. Tout bloc d'état peut être lié à un **bloc action** appartenant à la vue processus. Dans ce cas, on dit que les deux blocs sont synchronisés, et le bloc d'état prend alors le même nom que le bloc d'action. S'il n'est pas lié, l'utilisateur doit paramétrer la durée de cet état. Les **blocs d'attente** comportent un point d'entrée et un ou plusieurs points de sortie, correspondant à des conditions définies par l'utilisateur.

Lorsqu'un agent arrive dans un bloc d'attente, il est mis en attente jusqu'à ce que l'une des conditions de sortie devienne vraie. Si plusieurs conditions sont vraies au même moment, c'est la première sortie valide qui sera choisie, selon l'ordre défini par l'utilisateur.

Grâce aux diagrammes d'états, la vue ressource possède un formalisme simple : elle offre ainsi une représentation claire de l'activité de chaque acteur impliqué dans le processus de prise en charge du patient et un échange facilité. Il est ainsi possible de corriger ces diagrammes en fonction de l'apport de chacun et d'observer immédiatement les conséquences des modifications sur les autres vues et par la simulation.

4.3. Vue Processus

La vue Processus regroupe les diagrammes de flux représentant le parcours d'une ou plusieurs entités à travers un service médical. Les diagrammes d'activité UML sont utilisés pour représenter ces flux. Un diagramme d'activité est une succession d'actions. Une action est une étape élémentaire de l'activité. Afin d'être réalisée, une action nécessite éventuellement une ou plusieurs ressources.

Une activité commence avec un **bloc d'entrée**. Tout comme le diagramme d'états, ce bloc d'entrée dispose d'un seul point de sortie et permet de générer les agents représentant l'entité considérée. Un même bloc d'entrée ne peut générer qu'un seul type d'entités. De plus, l'utilisateur doit préciser le schéma de génération des entités : loi de probabilité ou planning pré-défini dans le sous-système informationnel. Un certain nombre de paramètres supplémentaires peuvent être renseignés, telle la possibilité de créer les entités par groupes ou de modifier la date de première création. Tout comme dans la vue Ressource, Le **bloc de sortie** permet de détruire les agents qui arrivent en fin de parcours. Ce bloc dispose d'un seul point d'entrée et ne nécessite pas de paramétrage.

Les **blocs action** permettent de modéliser une tâche réalisée par l'entité ou sur l'entité. Un bloc action comporte un point d'entrée et un point de sortie, et n'est pas décomposable. Une action nécessite un temps d'exécution et éventuellement une ou plusieurs ressources. Pour chaque ressource nécessaire à l'exécution de l'action, un **bloc d'état** lié à cette action doit exister dans le diagramme d'états correspondant. Ce lien traduit le fait que la ressource doit être présente pour réaliser la tâche modélisée. Le processus de synchronisation sera décrit dans la section 7. Enfin, nous introduisons le concept de **bloc d'attente**. Son principe de fonctionnement est identique à celui décrit dans la vue ressource.

Le but de la vue processus est double : (i) elle permet d'offrir au patient une représentation simple de son parcours au sein de l'hôpital avant son admission et vise à diminuer son stress (seules les activités directement liées au patient sont représentées) ; (ii) cette vue est également utile aux différentes catégories de personnel hospitalier afin de visualiser l'enchaînement des actions qui seront réalisées sur le patient et leurs contributions tout au long de cette chaîne.

5. SOUS-SYSTÈME INFORMATIONNEL

Le sous-système informationnel est une base de donnée regroupant toutes les informations nécessaires à la modélisation du système. Son organisation et son contenu sont à la discrétion de l'utilisateur, qui est libre de créer lui-même son modèle conceptuel de données. Cette base contient un certain nombre de tables, pouvant être réparties en deux catégories : (i) l'ensemble des tables nécessaires au fonctionnement de la simulation (plannings, emploi du temps, suivi d'entités, etc.), et (ii) l'ensemble des tables décrivant les particularités du système modélisé (base de données existantes).

Les données présentes dans le sous-système informationnel sont disponibles à tout moment durant la modélisation et la simulation. Leur accessibilité peut cependant être restreinte selon le système étudié. L'utilisation d'un système de gestion de bases de données « standard » permet en outre une importation/exportation aisée avec le système utilisé par l'hôpital au centre de l'étude.

6. SOUS-SYSTÈME DE PILOTAGE

Le sous-système de pilotage englobe un module de simulation multi-agents, permettant l'implantation d'une logique de contrôle et de décision. Ce modèle de contrôle est indépendant du modèle décrit dans le sous-système physique, il est donc théoriquement interchangeable. Le module de pilotage est invoqué en amont et en aval de chaque événement présent dans l'ordonnanceur de la simulation à événements discrets. Au moment de la synchronisation, la simulation est interrompue et l'état du système est transmis au modèle de contrôle. Selon la situation, un processus de prise de décision est mis en œuvre afin d'orienter le déroulement de la simulation.

La simulation multi-agents a été choisie pour modéliser le processus de prise de décision, permettant d'offrir une certaine liberté dans la stratégie de négociation entre agents. Nous désignons par « agent » indifféremment entités et ressources qualifiés pour participer à la prise de décision. L'objet de la négociation peut porter sur des objectifs de niveaux

différents. Nous ne nous étendons pas sur la description de la simulation multi-agents dans ce papier, mais décrirons brièvement plusieurs exemples de situations où l'intervention du modèle de contrôle est nécessaire.

6.1. Choix d'un ensemble de ressources

Le modèle de contrôle est invoqué lorsque le choix d'une ou plusieurs ressource(s) pour une certaine activité n'est pas explicite, notamment dans le cas de ressources spécialisées. La modélisation d'une activité nécessite parfois l'affectation conditionnelle de ressources en fonction du type d'entité. Dans ce cas, au lieu de lister les ressources nécessaires à l'accomplissement de l'activité, une expression conditionnelle permet de connaître quelles sont les ressources nécessaires en fonction d'un ou plusieurs paramètres. Un paramètre pourrait être la valeur de l'attribut de l'entité. De la même manière, lorsqu'une ressource est affectée à une activité, cette même ressource peut-être remplacée par l'une de ses héritières pour effectuer l'activité en question.

6.2. Ré-ordonnement d'entités

La simulation à événements discrets est souvent mise en œuvre pour tester la robustesse de la solution d'un problème d'optimisation. Il s'agit par exemple du planning des patients à opérer dans le bloc opératoire sur une semaine. La simulation permet de faire intervenir un certain nombre d'événements aléatoires susceptible de perturber le fonctionnement du système. Dans ce cas, un ré-ordonnement du planning en entrée est parfois nécessaire en cours de simulation afin de remédier à une situation de blocage ou d'attente prolongée. L'appel à un module d'optimisation est possible.

6.3. Modification de l'organisation du système

Dans le cas où le planning d'entrée ne peut être modifié (arrivées aléatoires), il est parfois nécessaire de modifier l'organisation du système au cours de la simulation. Nous appelons « modification » tout changement effectué automatiquement au sein de la vue organisation décrite section 4.1, par exemple : augmentation de l'effectif d'une ressource (impliquant éventuellement un coût supplémentaire), changement d'affectation d'une ressource, etc.

7. EXÉCUTION

La modélisation d'une unité médicale est réalisée selon le formalisme décrit dans les sections précédentes. Ce modèle peut directement être réalisé dans l'outil de modélisation/simulation proposé au moyen d'une interface graphique. Une fois terminé,

le modèle est simulé après vérification et compilation. L'outil est en cours de développement et est codé en Java. Le sous-système informationnel est une base de données réalisée grâce au SGBD MySQL. Cette section décrit la procédure automatique de conversion du modèle UML en modèle de simulation.

7.1. Vérification

Une fois le modèle terminé, celui-ci doit être vérifié afin de pouvoir être exécuté. Un certain nombre de vérifications sont réalisées durant la construction du modèle : validité de valeurs numériques, d'expressions logiques, etc. La vérification proprement dite se décompose en une succession d'étapes simples, visant à valider la cohérence du modèle.

1. Vérification de la complétion des flux : tout flux doit commencer par un bloc d'entrée et se terminer par un bloc de sortie. Tous les points d'entrée et/ou de sortie doivent être liés à un bloc. Un seul flux est défini pour chaque ressource. En revanche, plusieurs flux peuvent être définis dans la vue processus.
2. Un seul type d'entité doit être associé à chaque flux de la vue processus.
3. Vérification des liens entre vue processus et vue ressource : pour chaque action nécessitant l'intervention d'une ou plusieurs ressources, un état correspondant doit être présent dans le diagramme associé de chaque ressource (ou de l'une de ses ressources filles).
4. Toutes les références au sous-système informationnel doivent être valides (existence des tables et des données), en particulier pour les plannings et les emplois du temps nécessaires à la génération d'agents durant la simulation.
5. Vérification des paramètres de la simulation : durée, nombre de répliques, unités temporelles, etc.

La présence de cycles au sein des flux ne génère pas d'erreur, mais sera notifiée.

7.2. Compilation

La compilation du modèle physique consiste à construire un modèle logique tout en synthétisant les informations entrées par l'utilisateur. En pratique, il s'agit de parcourir récursivement l'ensemble des flux du modèle à partir des blocs d'entrée afin de construire le modèle logique grâce à des objets chaînés qui seront lus par le simulateur. Nous définissons une liste d'événements dans laquelle figure l'ensemble des événements à traiter par le simulateur. Les

événements sont classés dans l'ordre chronologique de leur apparition, et sont traités un par un par le simulateur. Nous définissons également une liste de conditions, qui sont autant de drapeaux (*flags* ou *switches* en anglais) permettant de connaître l'état d'une variable, la position d'un agent, etc.

7.3. Simulation

Un algorithme de simulation adapté à ce cadre de modélisation a été développé. Il s'agit d'un simulateur à événements discrets, identique sur le principe à celui décrit par (Law & Kelton 2004). Nous ne décrivons pas en détail les composants et l'organisation d'un modèle de simulation à événements discrets dans ce papier ; en revanche, les algorithmes de traitement d'événements ont été adaptés en fonction des besoins du modèle.

L'événement « Création » crée un nouvel agent selon le modèle défini dans le bloc associé. S'il est planifié, ce nouvel agent est initialisé selon les données qui figurent dans le sous-système informationnel. Dans le cas contraire, ses données sont générées aléatoirement selon un schéma fixé par l'utilisateur. La prochaine création est ensuite insérée dans la liste d'événements. On insère également l'événement associé au bloc qui suit la création dans le modèle logique : (i) un événement « Arrivée » pour un bloc action ou état, (ii) un événement « Décision » pour un bloc d'attente, ou (iii) un événement « Destruction » pour un bloc de sortie.

L'événement « Arrivée » modélise le début d'une activité ou d'un état pour un agent. Le traitement de cet événement nécessite un algorithme de synchronisation entre l'entité et les ressources nécessaires, décrit figure 2.

1	Si l'état/l'action n'est pas lié:
1.1	Programmer l'evt. suivant;
2	Sinon:
3.1	Si l'entité et toutes les ressources sont disponibles:
3.1.1	Libérer les autres agents;
3.1.2	Programmer l'evt. suivant;
3.2	Sinon:
3.2.1	Mettre l'agent en attente;

Figure 2: Algorithme de synchronisation

Ainsi la programmation de l'événement suivant ne peut être réalisée que lorsque toutes les ressources nécessaires sont prêtes et que l'entité est présente. La durée de l'action (ou de l'état) détermine la date du prochain événement. De plus la liste de conditions est vérifiée : toute condition se rapportant à l'événement en cours de traitement devient vraie et un événement

adéquat est immédiatement inséré.

L'événement « Décision » modélise le processus de prise de décision entre plusieurs alternatives, symbolisées par des conditions à vérifier. Ces conditions sont vérifiées dans l'ordre donné par l'utilisateur ; l'agent empruntera la sortie correspondant à la première condition reconnue comme étant vraie. Si aucune condition n'est valide, l'agent est mis en attente : il sera libéré lorsqu'une condition changera d'état. Dans le cas d'un bloc d'attente purement aléatoire, l'agent est libéré immédiatement selon le résultat d'un tirage aléatoire.

L'événement « Destruction » détruit l'agent. Aucune autre action n'est entreprise.

La simulation s'arrête lorsque la liste d'événements est vide ou lorsqu'une condition d'arrêt donnée par l'utilisateur devient vraie. La génération de rapports statistiques est alors lancée, permettant d'analyser le comportement du système après simulation. Le guide méthodologique associé à l'unité de soin étudiée joue ici un rôle important car il permet de sélectionner automatiquement les indicateurs de performance pertinents pour le système modélisé. Les états successifs du système sont enregistrés tout au long de la simulation, permettant de « photographier » un état particulier de l'unité de soins.

8. EXEMPLE D'APPLICATION : LE BLOC OPÉRAIRE

8.1. Modélisation

Le méta-modèle présenté dans les sections précédentes ne peut être utilisé tel quel. Nous devons l'instancier pour une unité médicale spécifique. Le bloc opératoire a été choisi dans cette section pour plusieurs raisons : (i) le bloc opératoire a fait l'objet de nombreuses études visant à optimiser son fonctionnement, ses différents modes opératoires sont aujourd'hui bien connus ; de plus (ii) nous avons eu l'occasion d'observer le fonctionnement du bloc opératoire de l'hôpital Bellevue (CHU de Saint-Étienne) et avons proposé une méthode d'optimisation pour la planification des patients basée sur la relaxation Lagrangienne (Perdomo, Augusto & Xie 2006).

Nous nous intéressons dans ce cas d'étude à un bloc opératoire de petite taille : moins de six salles opératoires et moins de dix lits de réveil. Le processus de prise en charge du patient se décompose en plusieurs phases distinctes. L'intervention est programmée à l'avance pour tous les patients qui ont pris rendez-vous. Le jour de l'opération, une équipe de brancardiers va chercher le patient dans sa cham-

bre et le transporte jusqu'au bloc opératoire. Après une éventuelle attente, celui-ci est pris en charge par une équipe chirurgicale et est admis dans une salle opératoire sélectionnée en fonction de la nature de l'intervention qu'il doit subir. L'opération débute ensuite. Lorsque l'intervention est terminée, le réveil du patient commence. Si un lit de réveil est libre, le patient est transféré immédiatement dans ce lit. Sinon, la salle opératoire est bloquée jusqu'à ce que le réveil se termine ou qu'un lit en salle de réveil devienne disponible. Une fois la salle opératoire libérée, celle-ci est nettoyée. Le patient complètement réveillé est transporté vers sa chambre par l'équipe de brancardiers.

Le processus patient est modélisé figure 3. Nous avons représenté les différentes étapes de son intervention par plusieurs blocs action. Le réveil du patient constitue un cas particulier, car la durée et le lieu de l'action dépend de la disponibilité des lits de réveil et n'est pas connue à l'avance. Nous décomposons donc le réveil du patient en deux blocs distincts : réveil dans la salle opératoire et réveil dans un lit de réveil. Dans ce cas, la durée de l'action est symbolisée par le temps d'attente de l'entité dans ces deux blocs. Nous définissons les interrupteurs *lit disponible* et *fin du réveil*, permettant de connaître quand ces conditions deviennent vraies. Enfin, les ressources nécessaires sont indiquées entre parenthèses pour chaque action. Nous avons planifié à l'avance les patients qui doivent être opérés durant la simulation dans le bloc de création.

La vue ressource permet de visualiser les tâches de chaque intervenant dans le processus patient. Une ressource passe par une succession d'états permettant la réalisation des actions décrites dans la vue processus. Certains états propres aux ressources apparaissent dans cette vue, tel l'état « Aller dans la chambre » pour les brancardiers, ou l'état « Nettoyage » pour la salle opératoire. Les états portant le même nom que les actions de la vue processus sont synchronisés. Comme cela a été défini plus haut, chaque ressource dispose d'un planning de travail, permettant de connaître les dates de début et de fin de service pour chacune d'entre elles.

La déclaration des entités et des ressources impliquées dans le système est réalisée dans la vue organisation, et est illustrée figure 5. Une table d'attributs permettant la caractérisation de ces agents doit être associée à toutes les entités et ressources du système. Les ressources possèdent également une table qui symbolise leur emploi du temps au sein du système. Une fois déclarées, entités et ressources (i.e. agents) sont agencées de manière à former une organisation, présentée figure 6. Ce diagramme reflète la



Figure 3: Bloc opératoire : sous-système physique, vue processus

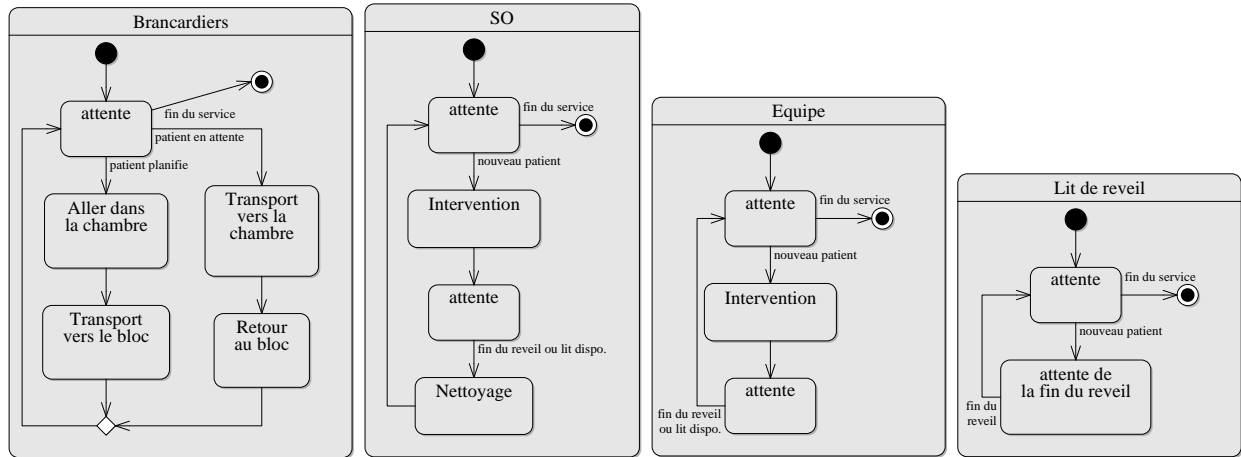


Figure 4: Bloc opératoire : sous-système physique, vue ressource

description du bloc opératoire.

Patient tableAttributs	Brancardiers tableAttributs emploiDuTemps	SO tableAttributs emploiDuTemps
----------------------------------	--	--

Figure 5: Bloc opératoire : sous-système physique, vue organisation, exemples de déclarations

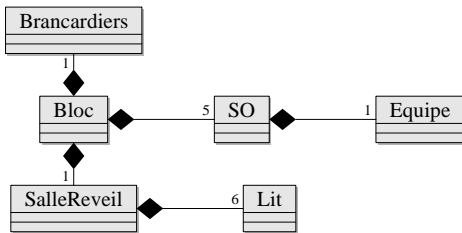


Figure 6: Bloc opératoire : sous-système physique, vue organisation, dépendances

Le sous-système informationnel est une base de données contenant un ensemble de tables organisées selon un modèle conceptuel de données. Parmi ces tables se trouvent les tables d'attributs des agents définis dans la vue organisation, ainsi que les tables d'emploi du temps des ressources et le planning de prise en charge des patients référencé dans le bloc création de la vue processus. En particulier, la table Patient contient l'ensemble des durées opératoires référencées dans le modèle (transport, intervention, réveil), car ces informations dépendent de son dossier

médical personnel. Enfin, le sous-système de contrôle permet de piloter la simulation. Nous ne détaillerons pas plus avant ce module encore en phase de test.

8.2. Commentaires

Nous avons pu modéliser précisément le processus de prise en charge du patient dans le bloc opératoire grâce au cadre de modélisation présenté dans ce papier. Plusieurs particularités du système, tel le blocage de la salle d'opération par le patient lorsque la salle de réveil est saturée, ont été modélisées. Ces phénomènes sont souvent négligés dans les modèles présentés dans la littérature. Le formalisme épuré du modèle permet en outre une compréhension rapide de la logique représentée ; la séparation des activités de chaque ressource permet d'établir un dialogue privilégié avec un intervenant en particulier. Les diagrammes d'états offrent la possibilité d'intégrer un grand nombre d'états en marge de la vue processus. En effet, il est possible d'aller plus loin et de modéliser le processus complet de prise en charge du patient dans le bloc opératoire en détaillant les différentes phases de l'anesthésie et de l'intervention.

La modification du modèle est facilitée par le concept de vues. La stratégie de prise en charge du patient après l'intervention par exemple peut être modifiée rapidement en corrigeant le diagramme de la vue processus. De la même manière, plusieurs configurations différentes peuvent être testées en rectifiant la vue organisation. Enfin, la conversion directe du modèle

vers la simulation permet le test de plusieurs scénarii très rapidement, pourquoi pas au cours d'une réunion avec le personnel hospitalier.

Le développement de l'outil n'étant pas terminé, nous ne pouvons présenter de résultats dans ce papier. Plusieurs tests de scénarii avec des jeux de données fictifs sont en cours afin de tester la fiabilité du modèle. Nous avons également entrepris la modélisation des services de pharmacie et de neurologie du même hôpital afin de tester la robustesse du cadre de modélisation.

9. CONCLUSION

Nous avons présenté dans cet article une méthodologie de modélisation ainsi qu'une plate-forme de simulation dédiées aux systèmes hospitaliers. Cet outil s'adresse à des utilisateurs ayant certaines connaissances en matière de modélisation/simulation et doit être vu comme une plate-forme de travail adaptée au secteur hospitalier. Une architecture basée sur une décomposition systémique en trois sous-systèmes (physique, informationnel et contrôle) a été adoptée, permettant un partitionnement clair entre flux, données et contrôle. Une telle décomposition apporte plusieurs avantages, favorisant notamment une communication plus aisée avec le personnel médical grâce au système de vue personnalisées. La simulation immédiate du modèle permet l'obtention rapide de résultats sélectionnés et interprétés en fonction du service médical étudié.

Le système de pilotage sera l'objet de toutes nos attentions dans un futur proche. L'intégration d'un module de simulation multi-agents permettra la mise en place de stratégies de négociation pour la prise de décision visant à améliorer le fonctionnement du système. Les systèmes holoniques (Valckenaers, Brussel, Bongaerts & Wyns 1997) en particulier représentent une alternative intéressante. Selon les souhaits de l'utilisateur, le système pourra ainsi être amené à évoluer vers un état différent de son état initial. Plusieurs études à venir en collaboration avec le CHU de Saint-Étienne permettront l'enrichissement de notre cadre de modélisation, fournissant à l'utilisateur une aide à la modélisation personnalisée en fonction du service de soin étudié.

REFERENCES

- Augusto, V. & Xie, X. (2006). Modélisation et analyse de flux par la simulation en milieu hospitalier : état de l'art, *Actes de la conférence GISEH 2006*.
- Chauvet, J., Féliès, P. & Gourgand, M. (2007). A contribution to emergency medical systems modelling and simulation, *Proceedings of the 2007 IC-IESM*.
- Gourgand, M. & Keller, P. (1991). Conception d'un environnement de modélisation de systèmes de production, *Actes du troisième congrès international de génie industriel, Tours*, pp. 191–203.
- Jun, J. B., Jacobson, S. H. & Swisher, J. R. (1999). Application of discrete-event simulation in health care clinics: A survey, *Journal of the Operational Research Society* **50**(2): 109–123.
- Law, A. M. & Kelton, W. D. (2004). *Simulation Modeling and Analysis*, McGraw-Hill.
- Montreuil, B., Charfeddine, M., Labarthe, O. & Côté, A. (2007). A generic agent-based framework for simulation in distributed healthcare systems, *Proceedings of the 2007 IC-IESM*.
- Moreno, L., Aguilar, R., Martín, C., Pineiro, J., Estévez, J., Sigut, J., Sánchez, J. & Jiménez, V. (1999). Patient-centered simulation tool for aiding in hospital management, *Simulation Practice and Theory* **7**: 373–393.
- Perdomo, V., Augusto, V. & Xie, X. (2006). Operating theatre planning using lagrangian relaxation, *Proceedings of the 2006 IC-SSSM*, pp. 1234–1239.
- Ramudhin, A., Chan, E. & Mokadem, A. (2006). A framework for the modelling, analysis and optimization of pathways in healthcare, *Proceedings of the 2006 IC-SSSM*, pp. 698–702.
- Roux, O., Combes, C. & Duvivier, D. (2006). A modeling methodology dedicated to simulation and based on generic meta-models using mda-uml: an application to a chronic renal dialysis unit, *Proceedings of the 2006 IC-SSSM*, pp. 692–697.
- Trilling, L., Besombes, B., Chaabane, S. & Guinet, A. (2004). Investigation et comparaison des méthodes et outils d'analyse pour l'étude des systèmes hospitaliers, *Technical report*, Rapport de recherche sur le projet HRP2.
- Valckenaers, P., Brussel, H. V., Bongaerts, L. & Wyns, J. (1997). Holonic manufacturing systems, *Integrated Computers Aided Engineering* **4**: 191–201.
- Vernadat, F. (1999). *Techniques de Modélisation en Entreprise : Application aux Processus Opérationnels*, Economica.