Modélisation, analyse et pilotage de flux en milieu hospitalier à l'aide d'UML et des réseaux de Petri

Vincent Augusto

Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

N° d'ordre : 496 GI

THESE
présentée par

Vincent Augusto

Pour obtenir le grade de Docteur
de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Génie Industriel

Modélisation, analyse et pilotage de flux en milieu hospitalier à l'aide d'UML et des réseaux de Petri

Soutenue à Saint-Étienne le 7 novembre 2008

Membres du jury

M. Benoit MONTREUIL	Professeur, Université Laval, Québec, Canada	Président
M. Michel GOURGAND	Professeur, Université Blaise Pascal, Clermont-Ferrand	Rapporteur
M. Christian TAHON	Professeur, Université de Valenciennes	Rapporteur
M. Éric MARCON	Professeur, Université Jean Monnet, Saint-Étienne	Examinateur
M. Patrick MARTINEAU	Professeur, Université François Rabelais, Tours	Examinateur
M. Frédéric GRIMAUD	Maître-assistant, École des Mines de Saint-Étienne	Examinateur
M. Xiaolan XIE	Professeur, École les Mines de Saint-Étienne	Directeur de thèse
Mme Françoise LORCA	Directrice des affaires médicales, CHU de Saint-Étienne	Membre invitée
M. Luc MERCHIER	Directeur des travaux et des équipements, CHU de Saint-Étienne	Membre invité

Spécialités doctorales :	**Responsables :**
SCIENCES ET GENIE DES MATERIAUX	J. DRIVER Directeur de recherche – Centre SMS
MECANIQUE ET INGENIERIE	A. VAUTRIN Professeur – Centre SMS
GENIE DES PROCEDES	G. THOMAS Professeur – Centre SPIN
SCIENCES DE LA TERRE	B. GUY Maître de recherche – Centre SPIN
SCIENCES ET GENIE DE L'ENVIRONNEMENT	J. BOURGOIS Professeur – Centre SITE
MATHEMATIQUES APPLIQUEES	E. TOUBOUL Ingénieur – Centre G2I
INFORMATIQUE	O. BOISSIER Professeur – Centre G2I
IMAGE, VISION, SIGNAL	JC. PINOLI Professeur – Centre CIS
GENIE INDUSTRIEL	P. BURLAT Professeur – Centre G2I
MICROELECTRONIQUE	Ph. COLLOT Professeur – Centre CMP

Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	MA	Mécanique & Ingénierie	CIS
BATTON-HUBERT	Mireille	MA	Sciences & Génie de l'Environnement	SITE
BENABEN	Patrick	PR 2	Sciences & Génie des Matériaux	CMP
BERNACHE-ASSOLANT	Didier	PR 1	Génie des Procédés	CIS
BIGOT	Jean-Pierre	MR	Génie des Procédés	SPIN
BILAL	Essaïd	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR 2	Informatique	G2I
BOUCHER	Xavier	MA	Génie Industriel	G2I
BOUDAREL	Marie-Reine	MA	Sciences de l'inform. & com.	DF
BOURGOIS	Jacques	PR 1	Sciences & Génie de l'Environnement	SITE
BRODHAG	Christian	MR	Sciences & Génie de l'Environnement	SITE
BURLAT	Patrick	PR 2	Génie industriel	G2I
CARRARO	Laurent	PR 1	Mathématiques Appliquées	G2I
COLLOT	Philippe	PR 1	Microélectronique	CMP
COURNIL	Michel	PR 1	Génie des Procédés	SPIN
DAUZERE-PERES	Stéphane	PR 1	Génie industriel	CMP
DARRIEULAT	Michel	ICM	Sciences & Génie des Matériaux	SMS
DECHOMETS	Roland	PR 1	Sciences & Génie de l'Environnement	SITE
DESRAYAUD	Christophe	MA	Mécanique & Ingénierie	SMS
DELAFOSSE	David	PR 1	Sciences & Génie des Matériaux	SMS
DOLGUI	Alexandre	PR 1	Génie Industriel	G2I
DRAPIER	Sylvain	PR 2	Mécanique & Ingénierie	CIS
DRIVER	Julian	DR	Sciences & Génie des Matériaux	SMS
FOREST	Bernard	PR 1	Sciences & Génie des Matériaux	CIS
FORMISYN	Pascal	PR 1	Sciences & Génie de l'Environnement	SITE
FORTUNIER	Roland	PR 1	Sciences & Génie des Matériaux	CMP
FRACZKIEWICZ	Anna	MR	Sciences & Génie des Matériaux	SMS
GARCIA	Daniel	CR	Génie des Procédés	SPIN
GIRARDOT	Jean-Jacques	MR	Informatique	G2I
GOEURIOT	Dominique	MR	Sciences & Génie des Matériaux	SMS
GOEURIOT	Patrice	MR	Sciences & Génie des Matériaux	SMS
GRAILLOT	Didier	DR	Sciences & Génie de l'Environnement	SITE
GROSSEAU	Philippe	MR	Génie des Procédés	SPIN
GRUY	Frédéric	MR	Génie des Procédés	SPIN
GUILHOT	Bernard	DR	Génie des Procédés	CIS
GUY	Bernard	MR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HERRI	Jean-Michel	PR 2	Génie des Procédés	SPIN
KLÖCKER	Helmut	MR	Sciences & Génie des Matériaux	SMS
LAFOREST	Valérie	CR	Sciences & Génie de l'Environnement	SITE
LI	Jean-Michel	EC (CCI MP)	Microélectronique	CMP
LONDICHE	Henry	MR	Sciences & Génie de l'Environnement	SITE
MOLIMARD	Jérôme	MA	Sciences & Génie des Matériaux	SMS
MONTHEILLET	Frank	DR 1 CNRS	Sciences & Génie des Matériaux	SMS
PERIER-CAMBY	Laurent	PR1	Génie des Procédés	SPIN
PIJOLAT	Christophe	PR 1	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR 1	Génie des Procédés	SPIN
PINOLI	Jean-Charles	PR 1	Image, Vision, Signal	CIS
STOLARZ	Jacques	CR	Sciences & Génie des Matériaux	SMS
SZAFNICKI	Konrad	CR	Sciences de la Terre	SITE
THOMAS	Gérard	PR 1	Génie des Procédés	SPIN
VALDIVIESO	François	MA	Sciences & Génie des Matériaux	SMS
VAUTRIN	Alain	PR 1	Mécanique & Ingénierie	SMS
VIRICELLE	Jean-Paul	MR	Génie des procédés	SPIN
WOLSKI	Krzysztof	CR	Sciences & Génie des Matériaux	SMS
XIE	Xiaolan	PR 1	Génie industriel	CIS

Glossaire :

PR 1	Professeur 1ère catégorie
PR 2	Professeur 2ème catégorie
MA(MDC)	Maître assistant
DR (DR1)	Directeur de recherche
Ing.	Ingénieur
MR(DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
ICM	Ingénieur en chef des mines

Centres :

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
SITE	Sciences Information et Technologies pour l'Environnement
G2I	Génie Industriel et Informatique
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

Remerciements

Je remercie vivement M. Benoit MONTREUIL de m'avoir fait l'honneur de présider le jury, ainsi que M. Michel GOURGAND et M. Christian TAHON d'avoir accepté de rapporter cette thèse et d'y avoir consacré tout le temps et l'énergie nécessaire. Je tiens également à remercier l'ensemble des examinateurs pour l'intérêt qu'ils ont porté à mes travaux et la pertinence de leurs questions. Leurs conseils avisés et leurs sincères encouragements ont ouvert la voie à de nombreuses perspectives de recherches.

Je remercie chaleureusement M. Xiaolan XIE, dont l'aide sur le plan technique et les grandes qualités humaines m'ont permis de mener à bout cette thèse. Je mesure la grande confiance qui m'a été accordée et je suis heureux d'avoir pu développer mes intuitions ainsi que mon propre style de recherche pendant ces trois années.

Cette thèse n'aurais pas pu être menée à bien sans la précieuse collaboration du CHU de Saint-Étienne et l'aide de Mme Françoise LORCA et de M. Luc MERCHIER, qui m'ont permis d'entrer dans le « sanctuaire » hospitalier. Je souhaite ainsi remercier l'ensemble du personnel médical et administratif qui m'ont gentillement accueilli lors de mes nombreuses incursions. Tout ce travail n'aurait pas été possible sans eux.

Je veux tout particulièrement exprimer ma reconnaissance à MM. Christophe LENTÉ et Jean-Louis BOUQUARD, qui ont su, avec l'ensemble de l'équipe Ordonnancement et Conduite du Laboratoire d'Informatique de Polytech'Tours, me convertir aux joies de la recherche lors de mon master.

Je tiens également à remercier l'ensemble du Centre Ingénierie et Santé ainsi que les membres du départements MPE pour les innombrables bons moments partagés durant cette thèse. Je n'oublie pas mes amis, thésards ou non, qui m'ont aidé et soutenu pendant ces trois années. Je garde un souvenir ému de ces pauses mémorables (ouiii), de ces bavardages de nerdz pour savoir qui a tiré en premier, de ces ~~soirées~~ nuits passée sur Tenkaichi (et autres), de ces séances de bloc, de ces cours de japonais et de salsa, (...) de tous ces moments qui m'ont donné la Force d'aller au bout de cette thèse...

J'ai une pensée toute particulière pour Viviana, qui a largement contribué à une partie de mes travaux, et qui m'a apporté beaucoup de joie durant ma première année de thèse. Te extraño mi amiga...

Finalement j'adresse un grand merci à toute ma famille qui a toujours été présente lorque j'en ai eu besoin. Je remercie en particulier mes parents et mon frère pour leur précieuse influence et tout ce qu'ils m'ont apporté. Mon ultime merci revient à Magalie, qui m'a suivi dans ces contrées stéphanoises, qui a toujours été présente lorsque j'en ai eu besoin et qui a su me supporter pendant les moments les plus difficiles, et toujours avec le sourire... Mi aim a ou !

Table des matières

Table des figures

Liste des tableaux

Introduction générale

L'ingénierie des systèmes hospitaliers au XXI^{e} siècle

L'information est au centre de toute chose à l'aube du XXI^{e} siècle : le développement foudroyant de multiples technologies de communication permet de traiter la bonne information au bon moment pour la bonne audience. Le domaine industriel bénéficia largement de ce formidable progrès par l'intermédiaire de méthodes scientifiques issues de la recherche opérationnelle ou de l'ingénierie informatique, constituant ainsi le vaste domaine du génie industriel.

Cependant ces progrès ne profitèrent que sporadiquement aux organismes du domaine public, généralement moins ouverts aux nouvelles technologies. Les systèmes hospitaliers en particulier n'échappent pas à ce constat. Le domaine de la santé publique est une priorité dans la plupart des pays industrialisés : l'accès aux soins est un droit qui nous concerne tous et qui représente sans surprise une part importante du PIB de la plupart de ces pays : 11,1 % pour la France, 8,3 % pour le Royaume-Uni et jusqu'à 15,3 % pour les États-Unis (INSEE, 2005).

Durant ces vingt dernières années, les établissements hospitaliers ont connu une évolution importante : confrontés à un contexte socio-économique de plus en plus rude, ceux-ci doivent se plier à de nouvelles règles de gestion afin de minimiser les coûts engendrés tout en conservant une certaine qualité de service. Entre 1986 et 2006, le nombre de lits installés en hospitalisation complète, qui constitue un bon indicateur des ressources hospitalières globales d'un pays, a baissé de plus de 23 %, passant de 573.000 à 440.000 lits ; pourtant la consommation de soins et de biens médicaux a augmenté durant la même période de plus de 175 %, passant de 56,9 à 156,6 milliards d'euro en France.

L'objet des études scientifiques de ces vingt dernières années (Jun *et al.*, 1999; Fone *et al.*, 2003) est d'apporter aux systèmes hospitaliers des gains significatifs en terme d'efficacité et de productivité par la mise en place d'organisations plus efficientes, tout en veillant à l'amélioration de la qualité des soins. L'application de méthodes scientifiques issues du domaine du génie industriel constitue une excellente approche pour atteindre cet objectif. La mise en œuvre du programme de recherche HRP3 (*Hôpitaux en Réseaux : Prévoir, Partager et Piloter*), soutenu par la région Rhône-Alpes, constitue un excellent exemple de mutualisation de travaux de recherche dans le but d'élaborer des approches méthodologiques nouvelles pour la conception et le pilotage de systèmes de production de soins complexes. Cette démarche est cependant semée d'embûches, car si les domaines industriel et hospitalier sont similaires sur de nombreux points, ces derniers diffèrent sur bon nombre d'éléments cruciaux : nous ne parlons pas de *produits* et de *machines*, mais de *patients* et de *médecins*.

Les difficultés liées à l'application de méthodes issues du domaine du génie industriel au milieu hospitalier se situent à plusieurs niveaux :

- L'analyse d'un système hospitalier est fortement liée à l'observation et la modélisation de flux de patients, et non de produits : il est difficile de prédire le parcours d'un patient au sein d'un système hospitalier car il dépend de multiples facteurs tel que sa pathologie ou son mode de prise en charge.
- L'activité du personnel hospitalier est très diversifiée et fait preuve d'une haute capacité d'adap-

tation à la demande : l'urgence est une notion récurrente à l'origine de la plupart des problèmes d'organisation. Enfin, l'environnement hospitalier est hautement stochastique, rendant difficile la planification de ressources.

– Un hôpital est constitué d'une multitude de sous-systèmes généralement cloisonnés et fonctionnant de manière concurrentielle. Chacun de ses services possède une organisation unique reposant sur des liens affectifs forts, mettant en exergue la notion de réseau social.

La méthodologie permettant de mener une étude au sein d'un hôpital est également cruciale : les différences culturelles entre les domaines médical et industriel impliquent la mise en œuvre de méthodes de travail adaptées pour l'analyse, la modélisation et la présentation de résultats.

À partir (i) d'un état de l'art mettant en valeur la diversité des approches analytiques menées dans le domaine hospitalier, et (ii) d'études de cas réalisées au sein du Centre Hospitalier Universitaire de Saint-Étienne dans plusieurs unités médicales aux spécialités différentes, nous sommes capable de formuler les objectifs scientifiques de cette thèse qui reposent sur l'élaboration de méthodes pour la modélisation et l'analyse de flux en milieu hospitalier.

Objectifs scientifiques

L'objectif principal de cette thèse repose sur la spécification et le développement d'une plate-forme de modélisation, d'analyse et de simulation de flux dédiée au milieu hospitalier. Nous nous plaçons dans une optique de prototypage rapide : le but est de proposer, via la simulation, des solutions rapides et pertinentes aux problèmes organisationnels formulés par le personnel hospitalier. Cette plate-forme devra proposer un environnement de modélisation adapté aux système hospitalier avec conversion immédiate vers la simulation à événements discrets. Un certain nombre de fonctionnalités reposant sur la planification, l'ordonnancement et le pilotage de systèmes devront également être intégrées pour couvrir l'éventail des problèmes classiques liées à l'organisation de systèmes de soins. De manière plus précise, l'objectif principal se décompose en quatre sous-objectifs :

1. **Développer un outil de modélisation flexible et orienté vers la représentation de flux de patients :** nous proposons dans cette optique un ensemble de spécifications du langage UML permettant la modélisation de processus de prise en charge dédiés aux patients d'une unité de soins.
2. **Proposer une méthode de conversion automatique du modèle théorique vers un modèle de simulation :** une classe particulière de réseaux de Petri, accompagnée de plusieurs algorithmes de conversion, a été développée dans ce but ; une application de l'algorithme général de simulation à événements discrets aux réseaux de Petri est également proposée.
3. **Résoudre les problèmes liés à la planification et à l'affectation de ressources :** les propriétés intrinsèques des réseaux de Petri sont exploitées et deux méthodes dédiées à la planification à court terme et à l'ordonnancement tirées d'un contexte industriel sont proposées.
4. **Proposer une architecture de pilotage performante :** une méthode hybride combinant pilotage hiérarchique et hétérarchique est proposée pour contrôler la simulation de flux. Plusieurs modules sont développés, permettant le traitement de problèmes d'affectations génériques ainsi que la modélisation de structures décisionnelles propres aux systèmes hospitaliers.

Plan de la thèse

L'ensemble de ce mémoire s'articule autour de huit chapitres répartis en deux parties. La première partie regroupe (i) la revue de littérature, (ii) la description des objectifs de la thèse, et (iii) les bases

théoriques de la méthodologie proposées pour la modélisation et la simulation de flux en milieu hospitalier ainsi que la plate-forme medPRO supportant cette méthodologie.

Le chapitre 1 constitue une large revue de littérature autour de la modélisation et de la simulation de flux hospitaliers de ces dix dernières années. Cet état de l'art est crucial dans la mesure où il nous permet de dégager les problématiques liées au contexte médical auxquelles nous nous efforçons de répondre au travers de cette thèse.

Le chapitre 2 est consacré à la description de la méthodologie proposée et de l'architecture adoptée pour la plate-forme de modélisation et de simulation medPRO. Nous nous sommes efforcés de délimiter clairement le champ d'application de cette étude et de définir à quel public s'adresse cet outil. L'architecture est également détaillée sur plusieurs plans : une approche systémique est adoptée pour la structuration logique de la plate-forme en trois sous-systèmes (physique, informationnel et décisionnel) ; une décomposition en plusieurs couches logicielles est également proposée pour accéder à une flexibilité d'utilisation maximale, allant du plus spécifique au plus générique.

Le chapitre 3 vise à spécifier l'outil de modélisation intégré dans cette plate-forme. UML est choisi pour la représentation du système sous trois angles différents : la vue *processus* permet la visualisation des flux liés au patient et uniquement au patient ; la vue *ressource* offre une représentation précise de l'activité de chaque intervenant dans le processus de prise en charge du patient ; la vue *organisation* regroupe les caractéristiques des intervenants et leurs relations. La synchronisation entre ces trois vues et les avantages d'une telle représentation sont détaillés.

Le chapitre 4 décrit le comportement dynamique du modèle UML spécifié dans le chapitre précédent. La classe des réseaux de Petri de santé est définie. Un algorithme de conversion général est proposé, permettant de passer automatiquement du modèle UML au modèle de simulation. Enfin, l'algorithme de simulation à événements discrets est adapté pour notre classe de réseaux de Petri.

Le chapitre 5 regroupe les spécifications du système de décision utilisé dans medPRO. Nous présentons dans un premier temps l'application de méthodes tirant parti des réseaux de Petri pour la planification et l'ordonnancement d'unités de soins. La mise en œuvre de ces méthodes vise à automatiser le test de plannings par la simulation dans un environnement stochastique. Puis nous décrivons dans un second temps une approche hybride hiérarchique/hétérarchique pour la mise en œuvre d'un système de pilotage en temps réel. Le principe du pilotage repose sur un échange de données entre le modèle d'émulation et le modèle de contrôle durant la simulation. Outils génériques et spécifiques sont proposés pour offrir un contrôle précis du modèle en fonction de ses caractéristiques.

La deuxième partie de ce mémoire regroupe trois études de cas liées à trois services de soins du Centre Hospitalier Universitaire de Saint-Étienne.

Le chapitre 6 décrit l'étude des durées de séjour des patients victimes d'accidents vasculaires cérébraux (AVC) au sein de l'unité neuro-vasculaire. Un modèle de simulation réalisé sous medPRO a été mis en œuvre afin de valoriser les modules de contrôle intégrés à la plate-forme.

Le chapitre 7 est consacré à l'étude du problème de transport de médicaments. Une méthode de programmation linéaire en deux étapes est proposée et intégrée dans la plate-forme medPRO. Les plannings créés sont simulés sous Arena et medPRO pour tester leur robustesse dans des conditions stochastiques.

Le chapitre 8 concerne la planification des interventions chirurgicales dans un bloc opératoire spécialisé. Les flux de ce système ont été modélisés avec medPRO et SADT, permettant de mettre en valeur les avantages de notre approche. Nous proposons également un outil de planification dédié au bloc opératoire et intégré dans la plate-forme medPRO.

Enfin, nous concluons ce mémoire avec un bilan final du travail réalisé. Nous ouvrirons également quelques perspectives de recherche.

Première partie

Une méthodologie pour la modélisation, la simulation et le pilotage de flux en milieu hospitalier

Chapitre 1

Revue de littérature : modélisation et simulation de flux en milieu hospitalier

Nous présentons dans ce chapitre une large revue de littérature portant sur le thème de la modélisation et de la simulation de flux en milieu hospitalier. Après une description des techniques de modélisation en entreprise et des outils de simulations classiques, un état de l'art organisé autour de quatre grands thèmes est détaillé : l'organisation et la gestion du service d'urgence, du bloc opératoire, de centres médicaux ambulatoires, et de systèmes multiservices. Les problématiques liées à la réutilisabilité de modèles et à l'émergence de plate-formes de modélisation et de simulation dédiées aux systèmes hospitaliers sont également abordées.

1.1 Introduction

L'importance des travaux de recherche dans le milieu hospitalier a pris une nouvelle dimension au cours des dix dernières années. En effet, confrontés à un contexte socio-économique difficile, la majorité des établissements hospitaliers du monde entier doivent se plier à de nouvelles règles de gestion afin de minimiser les coûts engendrés et de maximiser le confort et les soins des patients. De nombreux chercheurs se sont ainsi penchés sur ce problème, tentant d'apporter de nouvelles stratégies d'organisation et de planification dédiées au milieu hospitalier.

Avant de décrire le panorama des études scientifiques traitant de l'application des techniques de simulation aux problèmes organisationnels du milieu hospitalier, nous rappelons les bases de la modélisation en entreprise : ces techniques se sont révélées performantes dans un contexte industriel et sont très souvent utilisées comme base de travail et de compréhension dans la plupart des études recensées dans cet état de l'art. Nous nous efforçons par ailleurs de marquer la différence entre la modélisation d'un système hospitalier et la méthodologie adoptée pour cette même modélisation.

La simulation est un outil souvent utilisé dans les études concernant l'organisation de systèmes hospitaliers : Jun *et al.* (1999) proposent dans leur revue de littérature sur l'application de la simulation aux systèmes de soins une taxonomie articulée autour de deux axes majeurs : les *flux de patients* (planification et admission de patients, orientation des patients, planification et disponibilité des ressources humaines) et l'*allocation de ressources* (dimensionnement et allocation en nombre de lits, en salles, en ressources humaines). Le choix de cette taxonomie n'est pas anodin : elle permet d'identifier rapidement les problématiques phares du domaine hospitalier jusqu'à la fin du siècle dernier. Les auteurs identifient

par ailleurs les tendances concernant les travaux à venir telle l'apparition de modèles complexes, impliquant plusieurs services et un plus grand nombre d'entités. L'état de l'art de la simulation d'unités médicales que nous présentons ici se veut représentatif des problématiques récentes avec un accent sur les nouveaux besoins, les nouvelles organisations, les nouveaux systèmes de financement. Au-delà des spécificités relatives à chaque pays, il est possible de mettre en avant l'émergence de thèmes de recherche privilégiés.

Après un bref rappel des techniques de modélisation et de simulation présentées sections 1.2, 1.3 et 1.4, nous examinons successivement quatre domaines hospitalier importants dans la section 1.5 : l'organisation et la gestion du service d'urgence, du bloc opératoire, de centres médicaux ambulatoires, et de systèmes multiservices. Nous aborderons également section 1.6 les nouvelles problématiques liées à la réutilisabilité de modèles et à l'émergence de plate-formes de modélisation et de simulation dédiées aux systèmes hospitaliers. Enfin, une synthèse du panorama scientifique présenté dans ce chapitre sera proposée section 1.7, accompagnée d'un récapitulatif général.

1.2 Techniques de modélisation en entreprise appliquées au milieu hospitalier

La modélisation est la première étape de toute étude centrée sur l'ingénierie d'un système, hospitalier ou non. Cette étape est très importante car le modèle théorique permet (i) de comprendre le fonctionnement du système, et (ii) de déterminer une méthode de résolution en fonction du problème à résoudre. Dans la plupart des études présentées dans cet état de l'art, l'utilisation de simples logigrammes au formalisme plus ou moins rigoureux est monnaie courante. Pourtant, il existe de nombreuses méthodes de modélisation dédiées à l'origine aux entreprises ou aux systèmes industriels ; le terme « entreprise » doit ici être compris au sens large, désignant un large éventail de systèmes de production, dont les hôpitaux.

La majeure partie des méthodes de conception et d'analyse utilisées dans le domaine de la réingénierie de systèmes repose sur la même procédure : modélisation du système actuel (appelé *as-is system*) permettant la réalisation d'un diagnostic, puis modélisation du système cible (appelé *to-be system*) afin de mettre en œuvre les objectifs de la réorganisation et de construire un plan d'action. La modélisation est une phase essentielle de la procédure d'analyse, et l'outil de modélisation approprié doit être judicieusement sélectionné.

La modélisation du système permet la représentation de la structure et des processus du système étudié. Cette activité est essentielle pour bien comprendre le fonctionnement du système, faciliter la communication avec les acteurs, isoler les indicateurs de performance pertinents et procéder à la simulation du système.

Les principaux outils de modélisation et d'analyse en entreprise sont présentés dans cette section (Vernadat, 1999). Nous nous limitons à la présentation de cinq outils qui ont déjà été appliqués au milieu hospitalier : SADT, GRAI, ARIS, UML et les réseaux de Petri. Les trois premières méthodes ont été utilisées dans (Trilling *et al.*, 2004) pour modéliser trois systèmes hospitaliers différents : une évaluation de leurs atouts et de leurs inconvénients en matière de modélisation hospitalière est proposée, permettant leur évaluation et leur comparaison. UML est un outil de modélisation orienté objet de plus en plus utilisé dans le domaine de la modélisation hospitalière ; son formalisme hautement personnalisable a été appliqué dans de nombreux domaines. Enfin, les réseaux de Petri offrent un formalisme de modélisation rigoureux permettant une représentation dynamique très utile dans le domaine hospitalier.

Nous nous attarderons également sur l'application de démarches d'analyse et de modélisation dans le milieu hospitalier au travers de méthodologies appropriées tirées de la littérature.

1.2.1 Outils de modélisation

Un rapport de recherche (Trilling *et al.*, 2004) entrant dans le cadre du projet HRP2 (*Hôpitaux : Regroupement, Partage, Pilotage*) présente un état de l'art exhaustif des techniques de modélisation pour l'entreprise et l'application de trois d'entre elles – SADT, GRAI et ARIS – jugées pertinentes pour l'analyse de trois processus opératoires (de l'admission à la sortie de l'hôpital). Ce rapport entre dans le cadre d'un projet régional visant à traiter le problème du regroupement de plateaux médico-techniques.

SADT

La méthode SADT (*Structured Analysis and Design Technics*) (Ross, 1977, 1985), connue aussi sous le label IDEF0 (*Integration DEfinition for Function modeling*) offre une analyse fonctionnelle descendante, permettant une modélisation modulaire et progressive de systèmes complexes. SADT est une démarche systémique de modélisation d'un système complexe ou d'un processus opératoire. Cette méthode offre également un formalisme graphique clair, pouvant être utilisé comme outil de communication synthétique et universel. Les confusions d'interprétation sont ainsi minimisées grâce à l'utilisation d'une syntaxe simple.

L'analyse du système est réalisée grâce à une collection de diagrammes organisés hiérarchiquement et composés d'un nombre fini d'éléments. Cette modélisation permet la représentation conjointe d'actions (actigrammes) et d'informations (datagrammes), organisées au sein d'une structure arborescente. Une fonction est représentée par une « boîte » ou un « module » selon la vue considérée (datagramme ou actigramme). Quatre types de flèches sont utilisées pour définir les contraintes de liaisons entre boîtes. La figure 1.1 présente un exemple de boîte : les *entrées* sont les entités entrantes à traiter, les *sorties* sont les entités sortantes modifiées par le processus, les *contrôles* sont les contraintes d'exécution, et les *mécanismes* sont les ressources requises pour cette activité. Une fonction A_i peut ensuite être décomposée au niveau inférieur en un nombre fini m de sous-fonctions constituantes notées $A_{i1}, \ldots, A_{im}$.

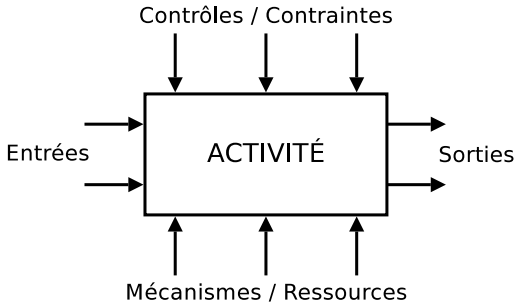


FIG. 1.1 – Représentation d'une fonction SADT

Le processus chirurgical de l'Hôpital de la Croix Rousse (Lyon, France) a été modélisé et analysé grâce à la méthode SADT afin de déterminer les points faibles de l'organisation. Ce modèle est présenté dans (Chaabane, 2004) et dans l'annexe C. Onze diagrammes ont été obtenus, permettant la description des processus pré-opératoire (de la consultation à l'admission dans l'hôpital), per-opératoire (de l'admission à la fin de l'intervention chirurgicale) et post-opératoire (de la sortie du bloc opératoire à la sortie de l'hôpital). SADT offre une représentation claire des différentes activités, mais le point faible de cette méthode réside dans la difficulté liée à la différenciation des types de flux et des types de ressources : les activités liées au patient sont mêlées aux activités préparatoires réalisées par les ressources. De plus, SADT ne comporte aucun formalisme pour l'aspect dynamique de la modélisation : sa conversion en un modèle de simulation est difficile. En résumé :

+ SADT propose une structure hiérarchisée par niveau permettant une représentation claire d'un système, aussi complexe soit-il.
+ Les diagrammes sont intemporels.
– Absence d'opérations de logique booléenne (ET, OU, etc).
– Les flux ne peuvent être différenciés en fonction de leur nature.
– Aucune représentation dynamique n'est proposée.

GRAI

GRAI (*Graphes et Réseaux d'Activités Inter-reliés*) est une méthodologie d'analyse et de conception des systèmes de décision et de gestion de production (Doumeingts, 1984). La force de la méthode GRAI réside dans sa capacité à fournir aux utilisateurs la possibilité de modéliser efficacement le système décisionnel de l'entreprise, à savoir l'organisation des processus qui génèrent les décisions. La méthodologie GRAI fournit un modèle de référence basé sur les concepts de système et de processus, et utilisant deux outils principaux : la grille GRAI pour le modèle du système décisionnel et les réseaux GRAI pour le modèle détaillé de chaque centre de décision (Roboam, 1993).

La grille GRAI apporte une vision macroscopique globale du système et permet de représenter le système décisionnel de l'entreprise selon deux axes : vertical (axe du temps) et horizontal (type de décision). Cette matrice permet de coordonner la vue fonctionnelle de l'entreprise et la vue processus par niveau de décision. À l'intersection entre les fonctions et les périodes temporelles se trouvent les centres de décision qui s'échangent des flux. Un centre de décision est un ensemble d'activités de décision appartenant à un même couple horizon-période et remplissant une même fonction. Le réseau GRAI permet la représentation des activités de décision. Le but de ces diagrammes est de connecter entre eux activités et résultats de ces activités. La figure 1.2 offre une représentation des outils GRAI.

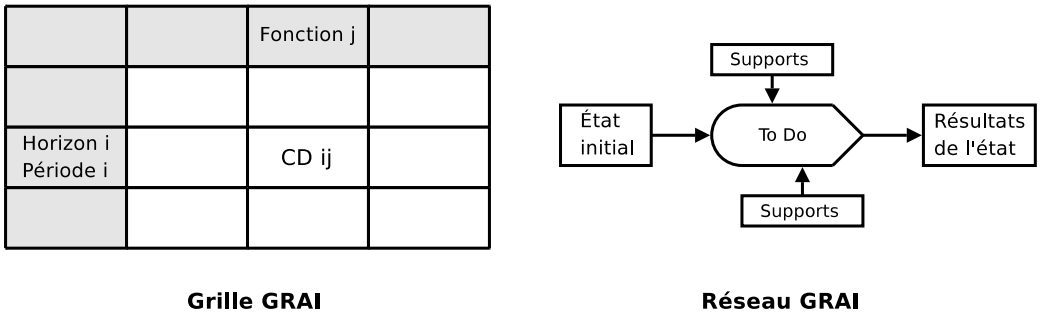


FIG. 1.2 – Représentation simplifiée des outils GRAI

Afin d'anticiper le regroupement de l'ensemble des blocs opératoires du CHU de Saint-Étienne (France), la méthode GRAI a été mise en œuvre pour modéliser le processus chirurgical orthopédique (Besombes et Merchier, 2004). Trois modèles ont été construits pour l'analyse et la comparaison de différents processus : (i) le *modèle physique* permettant de définir la typologie des flux du processus de prise en charge principal, (ii) le *modèle de processus* permettant la représentation des différentes pratiques organisationnelles, et (iii) le *modèle informationnel* permettant l'identification des flux d'informations afin d'en faciliter l'accès et l'utilisation. La grille GRAI permet de synchroniser les activités de décision et de coordination des différents niveaux de décision. Selon Besombes et Merchier (2004), les avantages de la méthode GRAI résident dans l'approche créative et l'importance du système de décision qui offre une représentation claire des processus dynamiques. Cependant comprendre une grille GRAI n'est pas chose aisée ; de plus, la représentation de flux n'est pas toujours claire graphiquement. En résumé :

+ Une méthodologie complète est proposée, offrant une approche originale du système.

+ GRAI propose une excellente représentation du système de décision.
– Une grille GRAI devient rapidement difficile à lire si toutes les décisions sont modélisées.
– La correspondance entre vue décisionnelle et vue processus est difficile à réaliser.

ARIS

ARIS est un cadre de modélisation représenté figure 1.4, bâti sur une approche multi-niveaux (conceptuel, technique, implémentation) et multi-vues (fonction, information, organisation, contrôle). Un modèle ARIS est constitué de diagrammes de processus permettant une représentation du système selon plusieurs vues : chaque niveau de chaque vue est représenté par un ensemble de diagrammes. ARIS met l'accent sur l'analyse et la définition des besoins durant la phase de conception de systèmes d'information et de gestion. Une plate-forme méthodologique générique bien documentée est proposée. ARIS est souvent associé à SAP car la migration de modèles pour l'incorporation dans SAP est possible. Enfin, ARIS est un langage sans méthodologie.

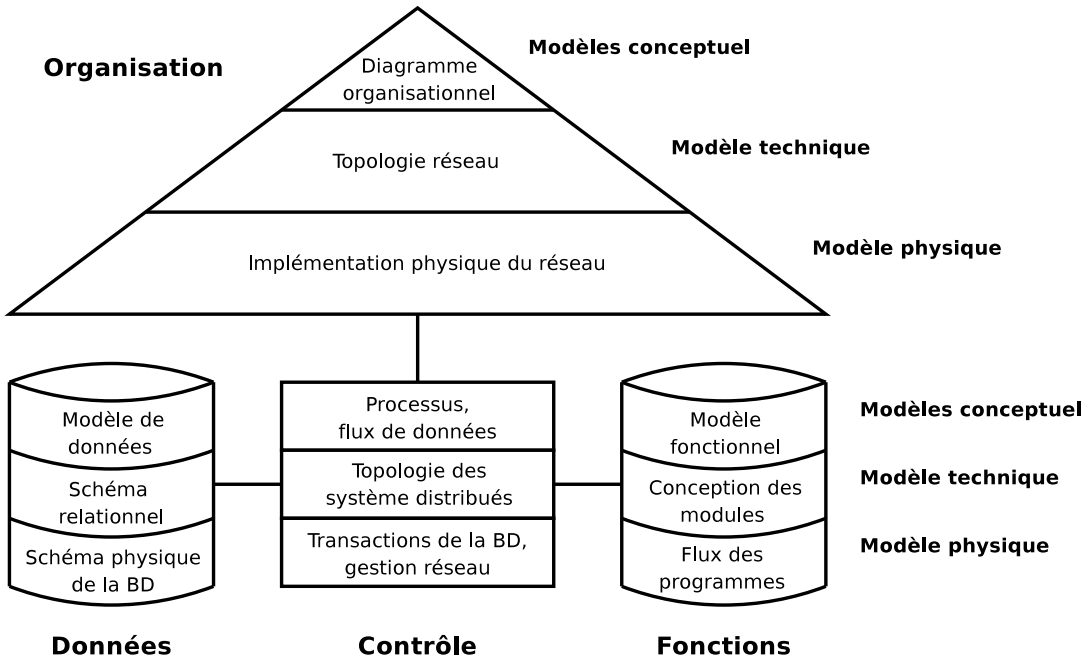


Fig. 1.3 – L'architecture ARIS

Le cadre de modélisation ARIS a été utilisé pour la modélisation du bloc opératoire du CH Saint-Joseph Saint-Luc (Lyon, France). Plusieurs modèles ARIS sont proposés : organisation (diagramme de flux), information, processus d'intervention chirurgical et système de décision. Le modèle ARIS ne peut pas être immédiatement converti en modèle de simulation, mais peut cependant être exécuté avec un outil issu de la suite logicielle d'ARIS. Les points forts et les points faibles de la méthode ARIS sont les suivants (Trilling *et al.*, 2004) :

+ ARIS offre une représentation graphique attrayante et intuitive.
+ La vue organisation structure le modèle et assure la cohérence entre les vues.
+ La représentation distincte d'activités, de données et de ressources dans un même modèle est possible.
+ L'intégration d'informations dynamiques est possible (durées opératoires par exemple).
– ARIS est un cadre de modélisation, aucune méthodologie n'est fournie.
– Les fonctionnalités de simulation sont restreintes.

UML

UML (*Unified Modelling Language*) est un langage de modélisation et de spécification non-propriétaire orienté objet principalement utilisé dans le domaine du développement informatique. UML propose un ensemble de notations graphiques standardisées regroupées en treize types de diagrammes. UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun. UML se décompose en plusieurs sous-ensembles : (i) les vues, permettant de décrire le système d'un point de vue donné (organisationnel, dynamique, temporel, architectural, etc.), (ii) les diagrammes, permettant de décrire graphiquement le contenu des vues, qui sont des notions abstraites, et (iii) les modèles d'élément, briques de base d'UML utilisées dans plusieurs types de diagramme. Kil *et al.* (2003) présente UML comme un support de communication universel, permettant la représentation de plusieurs vues complémentaires d'un système avec plusieurs niveaux d'abstraction.

UML est également un support de communication permettant d'exprimer visuellement grâce à son formalisme graphisme épuré une solution objet en limitant les ambiguïtés et les incompréhensions, facilitant la comparaison et l'évaluation de différentes solutions. Il s'agit de plus d'un langage universel compatible avec un grand nombre de supports et facile à assimiler.

Staccini *et al.* (2001) utilisent UML pour la création d'un modèle de données dédié à plusieurs processus hospitaliers. Une méthodologie est proposée pour structurer les besoins des usagers en utilisant une analyse orientée processus. Cette méthodologie est appliquée au processus de transfusion sanguine. Dans (Vasilakis et Kuramoto, 2005), plusieurs diagrammes d'états sont utilisés pour représenter les activités de trois chirurgiens, travaillant en parallèle sur trois activités différentes. Le comportement de chaque processus modélisé est défini grâce à l'ensemble admissible de séquences d'évènements, de conditions et d'actions. Une logique temporelle est associée aux évènements. Un modèle de simulation a été construit par la suite.

Les avantages et les inconvénients d'UML pour la modélisation de systèmes de soins peuvent être résumés de la manière suivante :

+ UML est un langage formel et normalisé, offrant précision et stabilité.
+ UML est un support de communication performant, permettant de cadrer l'analyse ; la compréhension de représentations abstraites complexes est facilitée et son caractère polyvalent et sa souplesse en font un langage universel.
– La mise en pratique d'UML est dangereuse car la modélisation est totalement libre et les outils sont variés.
– UML doit être spécifié pour être utilisé dans un cadre précis sans risque d'erreur.

Réseaux de Petri

Un réseau de Petri est un modèle mathématique permettant la représentation de systèmes distribués discrets (informatiques, industriels, etc.) introduit par Petri (1962). Un réseau de Petri est également un langage de modélisation représenté sous forme d'un graphe biparti orienté composé de nœuds appelés *places* et *transitions*, connectés grâce à des arcs orientés pondérés. Une présentation mathématique exhaustive des réseaux de Petri est donnée dans l'annexe A.

Les réseaux de Petri n'ont pas pénétré le milieu industriel faute de normes et d'une utilisation plutôt orientée recherche qu'industrie. Sa forme voisine, le Graphcet, s'est en revanche répandu dans le milieu industriel. Pourtant les réseaux de Petri sont le seul ensemble d'outils qui permet à la fois la spécification fonctionnelle, la modélisation et l'évaluation des systèmes de production. En outre, les réseaux de Petri les plus simples offrent un support graphique naturel pour les concepteurs.

Dans (Celano *et al.*, 2006) les réseaux de Petri sont utilisés afin de simuler le fonctionnement du département de radiologie d'un hôpital. Un tel outil permet l'étude des relations de dépendances entre

les ressources impliquées ainsi que la représentation de processus concurrents régis par des contraintes de précédence. Les auteurs insistent par ailleurs sur le fait que les réseaux de Petri permettent l'analyse d'un grand nombre de propriétés, favorisant la validation du modèle. De plus, ce dernier n'est pas plus lourd qu'un simple modèle de simulation étant donné que le programme de simulation associé procède de la même manière que n'importe quel outil de simulation à évènements discrets. En revanche, les réseaux de Petri ont deux inconvénients majeurs : la modélisation des performances temporelles, ainsi que la taille et la complexité du réseau. Une utilisation similaire des réseaux de Petri est à noter dans (Sampath *et al.*, 2006), où leur utilisation permet d'aider et d'assister l'organisation de la planification hospitalière et du système de réponse aux urgences.

Les avantages et les inconvénients des réseaux de Petri pour la modélisation de systèmes de soins peuvent être résumés de la manière suivante :

+ Les réseaux de Petri offrent un langage de modélisation précis et rigoureux.
+ Les propriétés des réseaux de Petri permettent la mise en évidence de certains types de comportements généraux.
+ Les similitudes de comportement entre réseaux de Petri élémentaires et systèmes de production discrets permettent la mise en œuvre de techniques originales pour la planification et l'ordonnancement.
– La représentation de systèmes complexes donne lieu à des réseaux illisibles ; une approche modulaire ou l'utilisation de réseaux de Petri colorés peut résoudre ce problème.
– Le formalisme graphique des réseaux de Petri n'est pas intuitif et difficile à comprendre pour une personne non-initiée.

1.2.2 Méthodologies de modélisation de systèmes de soins

Peu de travaux portent sur la conception d'une méthodologie de modélisation de systèmes de soins. Eldabi et Paul (2001) proposent une approche différente des méthodologies traditionnelles, théoriquement adaptée au milieu hospitalier. Selon les auteurs, la plupart des approches de modélisation existantes ne tiennent pas compte de l'étape de formulation et de structuration du problème. Il s'agit pourtant d'un point primordial, très spécifique au domaine hospitalier. Les auteurs ajoutent que la collecte de données se révèle souvent très approximative dans ce même milieu, et constitue donc une étape risquée.

Une approche alternative est donc proposée, où l'utilisation de la simulation est assujettie à la compréhension et la communication avec les acteurs du système étudié. En effet, d'après un panorama de l'utilisation de la simulation dans les milieux hospitaliers réalisé par Sanchez *et al.* (2000), trois problèmes principaux apparaissent : (i) manque de données, (ii) incompréhension du personnel demandeur, et (iii) nombreux conflits entre les demandes de ces derniers. Le modèle devrait ainsi être construit pour aider à la compréhension du problème. Pour cela, la communication avec les acteurs est essentielle ; ces derniers doivent pouvoir comprendre eux-même le problème et participer à sa résolution. Cette approche est appelée MAPIU (*Modeling Approach that is Participatory Iterative for Understanding*), décomposée en deux étapes distinctes :

1. L'initialisation, qui consiste en la classification des acteurs concernés par le problème (détenteurs du problème, experts, utilisateurs actuels). Cette étape est un point de départ permettant de faciliter le processus de collecte des informations auprès des acteurs.
2. Le processus de modélisation, qui consiste à identifier les besoins pour chaque acteur du système selon leur fonction. Trois sous-étapes sont présentées (c.f. table 1.1).

Finalement MAPIU permet la résolution de deux problème majeurs : la compréhension des processus propres aux systèmes de soins par les personnes concernées, et la résolution des conflits entre les objectifs souvent différents de ces mêmes acteurs. L'approche proposée est essentiellement basée sur la participation

Modélisation	**Activités liées au modèle : développement, manipulation de données, processus de sortie**
Spécifications	Besoins et notes de validation (informations données par les détenteurs du problème et les experts)
Incorporation	Construction ou modification du modèle en fonction des besoins et des notes de validation
Expérimentation	Changement de la structure du modèle et de ses paramètres (analyse de sensibilité et besoin en données)
Communication	**Lien entre les participants du processus**
Acteur ↔ Acteur	Communications entre acteurs du système (intercommunications)
Acteur ↔ Modèle	Communications entre un acteur et le modèle
Information	**Regroupe tous les feedbacks des acteurs**
Tangible	Résultats quantifiables ou indicateurs découlant du modèle (généralement après exécution du modèle basé sur le temps)
Intangible	Informations inquantifiables provenant du modèle (pendant le développement ou l'utilisation)

Tab. 1.1 – Le processus MAPIU en détail (Eldabi et Paul, 2001)

de ces derniers, en suivant un processus itératif. Elle permet ainsi une meilleure compréhension du modèle pour tous, favorisant son enrichissement et une meilleure inter-communicabilité entre les différents acteurs du système.

Morrison et Bird (2003) traitent de méthodes de conception et d'implantation en matière de simulation dans des centres médicaux ambulatoires à travers deux objectifs : (i) améliorer le fonctionnement général de ces centres en identifiant les ressources goulots, et (ii) tester l'opportunité de standardiser le modèle du centre de soin afin de définir une méthodologie générale. La définition du cahier des charges est une première étape cruciale. La récolte des données constitue une deuxième étape gourmande en temps. Les tests ont pu être réalisés de manière très souple par le personnel médical sans occasionner une charge de travail supplémentaire (mise en place d'outils graphiques). La communication fut un élément primordial pour l'achèvement de ces travaux, permettant de comprendre les besoins et d'expliquer les possibilités offertes par la simulation.

Dans un contexte similaire, Anderson (2002) réalise une évaluation de l'impact de l'implantation d'un nouveau logiciel au sein d'un centre hospitalier. L'auteur décrit trois exemples d'application : la mise en place (i) d'un réseau à grande échelle entre centres de soins, (ii) d'un système de transmission de consignes des médecins au sein d'un hôpital, et (iii) d'un système d'information conçu pour éviter les erreurs médicales. L'auteur démontre ici combien la simulation peut être utilisée pour l'évaluation des coûts et des performances de systèmes alternatifs.

Seuls les travaux d'Eldabi et Paul (2001) font état de la mise en œuvre d'une méthodologie de modélisation, de communication et d'accompagnement au changement dédiée aux systèmes hospitaliers. Dans la plupart des études présentées dans cette revue de littérature, cette phase est souvent négligée, ou à peine abordée. Son importance est pourtant capitale dans le milieu hospitalier, où le personnel soignant doit être rassuré quant à l'intérêt et la portée de ces outils utilisés avant tout dans le milieu industriel. De plus la mise en place d'une méthodologie de modélisation est également profitable au chargé d'étude, qui pourra ainsi appréhender le milieu hospitalier sans pré-requis médicaux : un tel guide permet de minimiser erreurs de compréhension et de communication sur les aspects sensibles du fonctionnement hospitalier.

1.3 Simulation à évènements discrets

Law et Kelton (2004a) décrivent la simulation à événements discrets comme la modélisation d'un système dont l'état évolue au cours du temps selon une représentation dans laquelle les variables d'état changent à certaines dates précises. Les *événements* se produisent à ces dates, un événement étant une occurrence instantanée suceptible de faire évoluer l'état du système.

La simulation à événements discrets a été très largement utilisée dans la littérature pour la résolution de problèmes d'organisation hospitalière depuis une vingtaine d'années (Jun *et al.*, 1999; Sanchez *et al.*, 2000; Augusto et Xie, 2006). Il s'agit de la méthode la plus facile d'accès et la plus rapide pour l'obtention de solutions précises et adéquates (sous réserve d'une bonne utilisation). La simulation à évènements discrets s'applique aussi bien dans une optique de diagnostic ou de réingénierie. Cependant, la simulation à événements discrets est mal adaptée au milieu hospitalier : le processus de prise en charge d'un patient est différent du processus de transformation d'un produit, et chaque acteur du système possède un comportement qui lui est propre. Plusieurs éditeurs de logiciels de simulation à événements discrets proposent ainsi des solutions adaptées aux systèmes hospitaliers.

Nous passerons rapidement sur les outils de simulation « généralistes », tel que Witness, Arena, Simul8, ProModel, etc. En marge de ces logiciels, plusieurs modules dédiés au domaine hospitalier ont fait surface. Nous choisissons de détailler dans cet état de l'art MedModel, extension de ProModel, qui constitue selon nous le module le plus abouti.

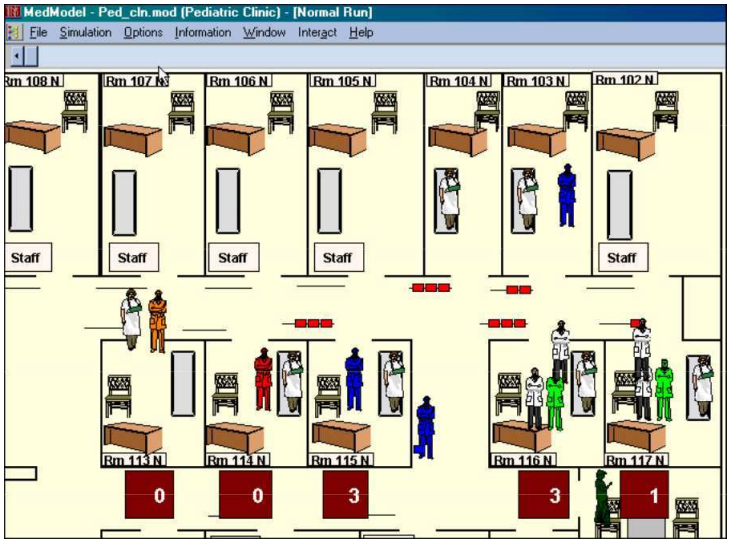


FIG. 1.4 – Simulation d'une clinique pédiatrique sous MedModel (Harrell et Lange, 2001)

Heflin et Harrell (1998) donnent une description détaillée de MedModel apparu en 1993 et reconnu comme le premier outil de simulation de sa catégorie. Cet outil permet notamment la modélisation de cinq systèmes génériques : service d'urgence, bloc opératoire, unité d'infirmiers, centre de soins ambulatoires et certains services « auxiliaires ». Chaque patient est identifié de manière unique par un identifiant ; il est ainsi possible d'associer à chaque patient un dossier médical qui lui est propre. La préemption de tâches est autorisée ; il est également possible de créer des routines et des macros permettant la modélisation rapide et efficace d'activités similaires et répétitives. Enfin, la planification de shifts récurrents est également proposée. Le module SimRunner Optimization permet l'ajout de fonctionnalités d'optimisation au modèle de simulation. Avantages et inconvénients peuvent être résumés de la manière suivante :

+ MedModel est simple à utiliser. Peu d'éléments de modélisation sont manipulés, ce qui permet d'optimiser le temps passé par l'utilisateur sur la phase de modélisation. De plus, l'outil de pro-

grammation intégré possède une aide facile à utiliser.

+ Les modèles peuvent être entièrement écrits sous Excel. En spécifiant les éléments du modèle et les paramètres correspondants, celui-ci peut être construit automatiquement.
+ MedModel possède de grandes possibilités graphiques permettant l'affichage d'un grand nombre de graphes pour des paramètres standard comme pour ceux définis manuellement.
+ MedModel possède un outil permettant la création de scénarios facilement.
– MedModel ne possède aucun comportement orienté objet, il n'est pas possible de définir des objets réutilisables. Les blocs de modélisation sont fixés.
– MedModel ne suit pas une approche de modélisation hiérarchique.
– Les éléments de modélisation et les opérations sont définies dans des tables. Cela rend la modélisation difficile si la table devient trop grande.
– MedModel manque de connexions vers des applications externes, comme MS Visio par exemple.

1.4 Simulation multi-agents

Un système multi-agent (SMA) est un ensemble d'agents situés dans un certain environnement et interagissant selon une certaine organisation. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome. Un agent peut modéliser un processus, un robot, un être humain, etc. Objet de longue date de recherches en intelligence artificielle distribuée, les systèmes multi-agents forment un type intéressant de modélisation de sociétés, et ont à ce titre des champs d'application larges, allant jusqu'aux sciences humaines. Les SMA sont de plus en plus prisés dans le domaine de la modélisation de systèmes de production de soins ; nous avons donc jugé utile de les introduire dans cette revue de littérature.

MedPAge

Bartelt *et al.* (2002) présentent une spécification orientée multi-agents concernant la planification de patients au sens large du terme, afin de contribuer au développement d'un système dédié aux services de soins. Les auteurs décrivent dans un premier temps une structure générique pour l'hôpital, puis discutent des différents problèmes d'ordonnancement et de planification qui se présentent. Les auteurs décrivent ensuite l'architecture du système multi-agents ainsi défini. Une méthodologie de construction de système d'ingénierie multi-agents a été choisie, en l'occurrence MESSAGE/UML, basée sur UML.

Trois concepts existent : *Entités concrètes* (agents, organisations), *Activités* (tâches, interactions) et *Entités d'états mentaux* (éléments de savoir des agents). Plusieurs représentations graphiques sont associées à ces éléments. Cinq vues sont proposées : organisation, but/tâche, agent/rôle, interaction et domaine. En ce qui concerne l'architecture proprement dite, plusieurs types d'agents ont été définis, à savoir des agents-patients (P-Agents), des agents-ressources (R-Agents) ainsi que des agents-savoir (K-Agents), symbolisant le savoir des P- et R-Agents. Plusieurs diagrammes ont ainsi été créés, à savoir les relations structurelles ainsi que les flux de données. Par la suite, les auteurs définissent plusieurs fonctions utiles afin de modéliser l'état de santé des agents ainsi que leurs durées de séjour au sein de l'hôpital. Un protocole d'interactions - (E)AUML - entre agents a été défini pour que ces derniers puissent coordonner leurs plans afin d'atteindre leurs objectifs. La figure 1.5 présente un exemple de modélisation orientée objet utilisée dans MESSAGE/UML.

Le problème spécifique de la planification des patients est décrit par Paulussen *et al.* (2003). Afin de modéliser l'état de santé des patient, représentés par des agents autonomes, les auteurs introduisent la notion de fonction de coût. Les agents sont alors amenés à optimiser ces fonctions afin de satisfaire leurs propres objectifs, à savoir l'amélioration de leur état de santé et la minimisation de leur durée de séjour. L'affectation de créneaux horaire se fait au terme d'une négociation entre deux agents patients modérée

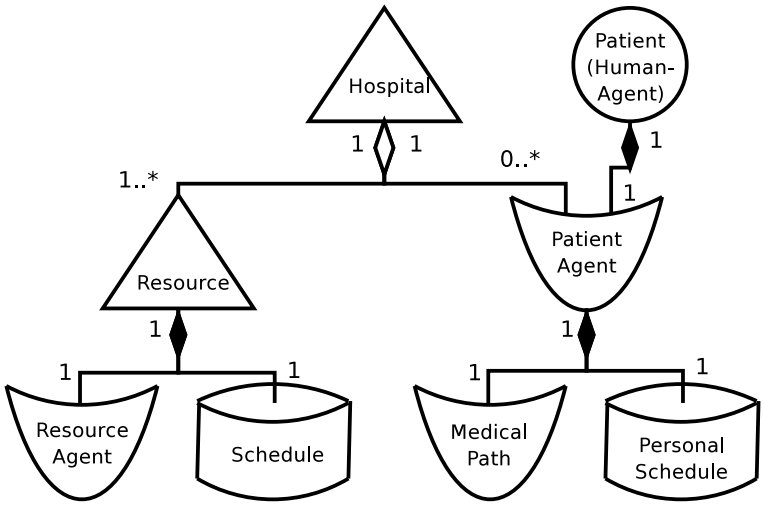


FIG. 1.5 – Modélisation orientée objet MESSAGE/UML

par un agent de type ressource (le médecin, par exemple) selon un mécanisme de gestion de marché. Ce mécanisme de coordination permet l'affectation ou la réaffectation de créneaux en fonction de l'état de santé des agents patients, chaque agent cherchant à maximiser sa propre fonction objectif. Une telle négociation intervient lorsque le créneau désiré par un agent patient est déjà occupé ; on entend ici par « créneau désiré » le créneau le plus proche en fonction de l'attente maximale autorisée par l'état de santé du patient. Une telle approche permet l'introduction de durées opératoires stochastiques, concept proche de la réalité hospitalière. Les agents du personnel médical cherchent à maximiser leur temps d'utilisation et à minimiser leur inactivité.

Agent.Hospital

Agent.Hospital est une plate-forme multi-agents dédiée au domaine hospitalier présentée dans (Kirn *et al.*, 2003). Ce projet rassemble les pré-requis ainsi que plusieurs solutions afin de mettre en œuvre le développement et la connexion de systèmes multi-agents autonomes. Le modèle Agent.Hospital se veut généralisable et empirique : il s'agit d'une plate-forme ouverte où interagissent un certain nombre d'acteurs faisant partie du domaine de la santé. Un exemple de scénario intitulé « Clinical Trials » est présenté : il s'agit de tests exécutés sur des échantillons aléatoires de patients. La procédure commence avec plusieurs tâches de diagnostics et de traitements qui doivent être coordonnées ; plusieurs ressources doivent également être planifiées et éventuellement informées. Les patients sont sélectionnés en fonction de leur dossier et présentés devant un médecin. Un rendez-vous pour l'examen est ensuite donné au patient dans les quatre prochaines semaines à partir de la consultation. Plusieurs jours avant la date programmée, les ressources nécessaires sont rassemblées. Un rendez-vous peut éventuellement être déprogrammé en fonction des urgences. Le processus se termine avec le départ du patient.

Les auteurs concluent sur l'importance de s'orienter vers le développement de nouvelles théories et méthodologies pour la modélisation et la simulation de systèmes décentralisés. Le projet Agent.Hospital constitue les balbutiements d'un projet présentant une valeur ajoutée importante en matière de technologie informatique.

1.5 Étude de systèmes hospitaliers à l'aide de la simulation

Nous proposons dans cette section un panorama des études de modélisation et de simulation des systèmes hospitalier. Cet état de l'art est organisé selon quatre axes de recherches importants correspondant à des secteurs hospitaliers caractéristiques. Les travaux présentés dans chaque axe permettent de situer la nature des études réalisées et les attentes des hospitaliers par rapport aux problèmes organisationnels détectés. La taxonomie adoptée est présentée figure 1.6.

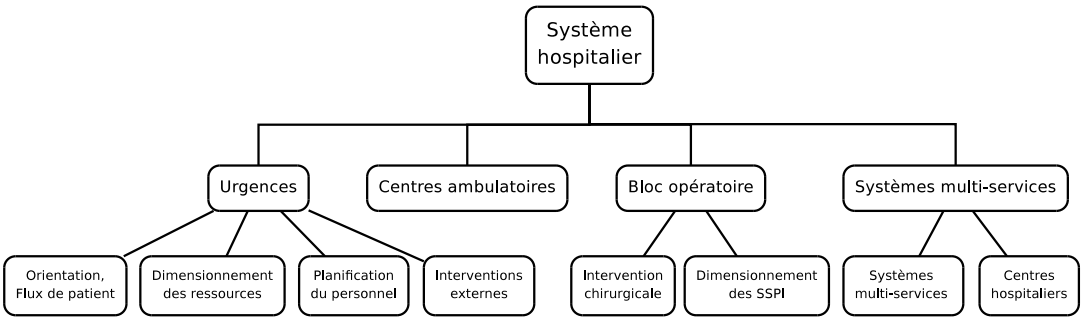


FIG. 1.6 – Taxonomie adoptée

1.5.1 Organisation et gestion du service d'urgence

Orientation, flux de patient

La simulation trouve tout son intérêt lorsqu'elle est utilisée pour modéliser certains flux de patients complexes à travers l'hôpital et pour tester plusieurs scénarios en changeant les règles de pilotage. Le cas se présente typiquement dans les services d'urgence, où les patients arrivent de manière stochastique et nécessitent un large éventail de traitements (allant du soin léger à la prise en charge de blessures graves), couvrant moult services et de spécialités. Étant donné que le schéma d'arrivée est inconnu, le personnel médical doit prendre en charge les patients les uns après les autres et les diriger au mieux à travers le service des urgences, et, si besoin est, au sein de l'hôpital. Une mauvaise orientation peut ainsi provoquer une augmentation significative des temps d'attentes et des coûts de prise en charge.

Centeno *et al.* (2001) examinent quatre domaines précis d'un service de maternité afin d'optimiser (i) le processus de flux de patientes et (ii) l'utilisation des ressources. Un diagramme de flux a été établi, modélisant le parcours des patientes en urgence (accouchements) et en rendez-vous pour des examens ou des interventions. Le système étudié mêle processus d'arrivée stochastique et déterministe, ce qui renforce l'intérêt de cette étude. Trois processus principaux sont modélisés : le suivi en zone de préparation, le suivi dans les salles de travail et le suivi dans la salle de réveil. Le modèle de simulation ainsi réalisé a permis de déceler certains problèmes d'organisation et d'émettre plusieurs recommandations.

Miller *et al.* (2003) se sont penchés sur l'amélioration du fonctionnement des services d'urgences afin de minimiser le temps de séjour des patients. L'équipe de recherche a ainsi tenté d'identifier et d'analyser les différents processus discrets centrés sur les patients. Plusieurs idées d'améliorations furent proposées, transformées par la suite en scénarios de simulation, afin de déterminer (i) les causes de l'attente pour l'obtention d'un lit dans le service des urgences, et (ii) la durée excessivement longue du séjour des patients dans le service. Il apparait que l'investissement de temps en phase de modélisation peut se révéler payant dans l'optique de création d'un outil de simulation adaptatif et puissant. Enfin, l'apport de cet outil a permis de favoriser une meilleure compréhension de l'organisation du service des urgences pour le personnel soignant.

Samaha *et al.* (2003) montrent qu'un problème d'organisation n'est pas forcément lié aux ressources mises en jeu ; l'opération consistait à réduire le temps de séjour des patients dans le service des urgences en identifiant les ressources goulot et en visualisant les différents processus de ce service. Après une description détaillée du parcours du patient, le test de différents scénarios sous Arena a finalement permis d'affirmer que le problème d'organisation était relié aux processus et non aux ressources.

Un plan d'expérience est utilisé par Baesler *et al.* (2003) afin d'estimer la demande supplémentaire pouvant être supportée par un service de soins d'urgence, tout en maintenant un temps d'attente raisonnable pour les patients et en considérant les ressources matérielles et humaines actuelles. Cela revient donc à estimer la capacité maximum du service en question. Plusieurs augmentations de la demande ont été simulées et une interpolation de la courbe de réponse a permis estimer le temps d'attente correspondant. L'effectif minimum nécessaire a ensuite été déterminé afin de satisfaire cette demande sans détériorer les performances du système.

Glaa *et al.* (2006a) proposent une modélisation du parcours patient au sein d'un service d'urgence au moyen de la méthodologie GRAI, l'objectif étant d'optimiser ce parcours et d'améliorer la qualité de la prise en charge du patient. Les modèles sont caractérisés par une grande diversité dans les activité et par un nombre important d'intervenants. La simulation a permis d'identifier les problèmes organisationnels du système et d'évaluer plusieurs organisations alternatives portant sur la disposition physique, les effectifs, etc.

Dimensionnement des ressources

Kim *et al.* (2000) proposent une méthode permettant de minimiser l'annulation d'interventions programmées en raison de l'indisponibilité de lits dans l'unité de soins intensifs. La simulation est utilisée afin de déterminer l'impact de cinq stratégies d'allocation flexible de lits dans les unités de soins intensifs. Malgré la difficulté inhérente à ce problème (les nouvelles demandes pour les soins intensifs apparaissent de manière stochastique), il apparait qu'une réservation exclusive d'un nombre de lits pour les interventions programmées réduit significativement le nombre d'interventions annulées. L'allocation flexible des ressources notamment est très utile, et n'engendre pas de coûts supplémentaires.

Miller *et al.* (2004) se sont également penchés sur le problème de l'organisation matérielle d'un service d'urgence (nombre de lits dans le service lui-même, nombre de chambres, amélioration du processus de prise en charge) ; ceux-ci décrivent ici une démarche standard utilisant le logiciel de simulation EDsim afin d'arriver à une configuration du service d'urgence optimale. L'expérimentation a permis de fixer les contraintes concernant successivement : les patients (nombre de lits requis, et durée de séjour dans le service) et le service lui-même (effectifs du personnel, nombre de lits d'urgence). Plusieurs améliorations du service peuvent alors être proposées dans différents secteurs, tels que l'affectation de lits, l'enregistrement administratif, le temps opératoire de procédures classiques, les dates de sortie des patients, l'élimination des temps morts.

Wiinamaki et Dronzek (2003) ont traité un problème très similaire afin de déterminer les équipements requis en nombre de lits dans l'optique d'une extension d'un service d'urgence constitué de deux centre de soins, l'un pour les patients gravement atteints, l'autre pour les soins bénins. Le flux des patients a été déterminé et le modèle a été construit. Plusieurs scénarios clés ont été testés (augmentation du volume de patients, heures de fermeture, équipements requis en salle de radiologie, etc.) et ont permis de prédire la capacité future requise pour le service de soins d'urgence. De plus, les auteurs insistent sur le fait que le modèle est parfaitement réutilisable, et pourra servir au personnel d'outil d'aide à la décision.

Devant la constante augmentation de l'âge de la population, les services de gériatrie doivent faire face à une demande de plus en plus forte. El-Darzi *et al.* (1998) ont utilisé un modèle de simulation basé sur des files d'attente afin de démontrer que les lits constituent la ressource goulot d'un tel service. Une mauvaise affectation des lits disponibles peut entraîner un blocage, par exemple lorsqu'un patient

nécessitant des soins de réhabilitation est bloqué dans le service de soins intensifs suite à une erreur. Les patients arrivent avec ou sans rendez-vous, le processus d'arrivée suit ainsi une loi de Poisson. La règle choisie pour chaque file d'attente est FIFO malgré le fait qu'elle n'est pas toujours adaptée à la réalité (facteur de gravité). Les files d'attente sont utilisées afin de mesurer le taux d'étranglement au niveau des lits entre les services de soins intensifs et de réhabilitation, et entre les services de réhabilitation et de séjour prolongé.

Planification du personnel des services d'urgence

Les travaux concernant la planification de personnel (en particulier des infirmières) font principalement appel à des techniques de recherche opérationnelle classique, en particulier la programmation linéaire ; dans ce cas, la simulation est rarement utilisée, si ce n'est pour tester le modèle proposé. Cependant, il existe plusieurs travaux faisant uniquement appel à la simulation, notamment dans le cas où l'arrivée des patients est entièrement stochastique. Le test de scénarios permet alors de prévoir les effectifs dans le futur. L'étude proposée par Griffiths *et al.* (2005) illustre parfaitement ce concept avec la résolution du problème de planification des shifts des infirmières d'une unité de soins intensifs en prenant également en compte les ressources matérielles (lits). Le modèle comprend plusieurs schémas d'arrivée dépendant du temps ainsi que les distributions appropriées pour la durée de séjour des patients. Le nombre d'infirmières requis est calculé shift par shift. La principale particularité de cette étude réside dans la différenciation entre infirmières régulières et infirmières supplémentaires (coût horaire plus élevés, engagement en renfort). Les auteurs pensent que le modèle de simulation est représentatif du fonctionnement du service de soins intensifs étudié, malgré certaines limitations et approximations qu'il serait bon de corriger : différenciation des infirmières, validation sur la base de données plus récentes, prise en compte des périodes de vacances, etc.

Tan *et al.* (2002) ont étudié l'impact d'une augmentation des effectifs en médecins dans un centre de soins d'urgence. Le système étudié se révèle inadapté à la demande quotidienne en soins, une étude préliminaire ayant déterminé que les médecins représentaient la ressource goulot. Deux modèles de simulation ont été développés afin d'évaluer l'impact de la réorganisation des ressources humaines prévue. Après expérimentation, il apparait que les améliorations prévues sont confirmées : le patient passe en moyenne 18% de temps en moins dans le système.

La simulation peut également être couplée avec la programmation linéaire pour résoudre certains problèmes, notamment les problèmes de planification de personnel : Centeno *et al.* (2003) utilisent la programmation linéaire en nombre entiers (PLNE) et la simulation afin de prévoir les besoins en personnel dans un service d'urgence. Le modèle de simulation qui permet de prévoir les besoins en personnel pour chacune des périodes prédéfinies, et le PLNE permet de générer un planning précis. Le modèle de PLNE repose sur une fonction objectif visant à minimiser le coût relatif aux infirmières tout en respectant les contraintes imposées d'une part par la législation et d'autre part par les résultats obtenus après l'exécution de la simulation. Les résultats indiquent que l'heuristique proposée est meilleure que l'approche empirique actuellement en vigueur. La génération automatique des plannings représente un gain de temps et de fiabilité non négligeable pour le personnel hospitalier. Glaa *et al.* (2006b) proposent une méthode pour l'optimisation de la gestion des infirmiers dans un service d'urgences en tenant compte des compétences. La modélisation du service a été réalisée et le problème d'affectation a été modélisé sous forme d'un programme linéaire.

Interventions externes

En marge de ces études, de nouvelles problématiques ont émergé durant les dernières années ; celles-ci concernant le routage sur un plan géographique. Cette problématique concerne essentiellement les services d'urgence mobiles.

Plusieurs travaux portent sur le routage des ambulances (interventions d'urgence et rapatriement des patients). Stevenson *et al.* (2001) utilisent la simulation pour déterminer quelle est la meilleure stratégie en matière d'affectation d'une destination pour des ambulances (urgences ou centre neurologique) pour des personnes accidentées, en fonction de la sévérité de leurs blessures. Il s'agit alors : (i) d'évaluer les stratégies sur une courte période, dans des coûts raisonnables, (ii) de s'assurer de la validité de la comparaison, et (iii) de réaliser des analyses précises sur l'influence de chacune des variables d'entrée. Plusieurs niveaux de gravité ont été définis pour quatre issues possibles. Le modèle de simulation a été soumis à onze stratégies différentes afin d'estimer la probabilité de survie des patients.

Su et Shih (2003) proposent quant à eux la modélisation du processus d'intervention d'urgence d'ambulances (localisation de l'accident, examen du malade sur place et rapatriement). Il s'agit de construire un modèle pour évaluer le système actuel et éventuellement l'améliorer. Les auteurs proposent une construction du modèle en cinq étapes : (i) entretiens (synthèse de la marche à suivre), (ii) examen de données d'interventions archivées, (iii) construction d'un modèle basique, (iv) validation et vérification du modèle de simulation (coopération d'experts du domaine médical), et (v) analyse des données en sortie pour les stratégies choisies. Plusieurs résultats amènent les auteurs à repenser le mode de fonctionnement actuel de l'étude de cas présentée. Le critère à minimiser est le nombre de pertes humaines.

1.5.2 Organisation et gestion du bloc opératoire

Processus d'intervention chirurgicale

Le bloc opératoire représente aujourd'hui encore la ressource la plus coûteuse dans un hôpital : l'ouverture de salles d'opération, la mobilisation de chirurgiens et d'équipements de haute technologie contribue à l'augmentation du coût horaire de fonctionnement de ce service. Il s'agit donc de maximiser son utilisation dans la journée, et ce sans pour autant occasionner des heures supplémentaires.

Tyler *et al.* (2003) cherchent à ordonnancer des interventions du même type, sans urgence. Afin d'étudier l'impact de plusieurs scénarios, plusieurs variables ont été modifiées (le délai entre deux interventions par exemple), l'objectif étant de minimiser le temps d'attente des patients et de maximiser le temps d'utilisation du bloc opératoire sans heures supplémentaires. L'ouverture d'une salle de préparation pour les patients en avance (alors qu'aucune salle n'est libre) permet d'obtenir un taux d'utilisation de 85 %. Cependant les auteurs précisent que le modèle présenté ne correspond pas à la réalité, où les durées des interventions varient énormément et où des urgences peuvent survenir.

La programmation élective des patients en chirurgie est un autre sujet largement traité grâce à la recherche opérationnelle. Vasilakis et Kuramoto (2005) ont testé deux méthodes de planification de rendez-vous en chirurgie. En pratique, un patient lambda doit prendre rendez-vous pour une consultation avec le chirurgien, puis prendre à nouveau rendez-vous pour l'intervention. Les deux politiques testées sont l'utilisation de plusieurs listes de rendez-vous (une par chirurgien), ou l'utilisation d'une liste globale pour les trois chirurgiens. Le mode de fonctionnement a été représenté sous forme d'un système réactif conduit par des évènements impliquant de multiples processus concurrents. La simulation montre que l'utilisation d'une unique liste d'attente pour les consultations permet de réduire le temps d'attente pour le premier rendez-vous, mais augmente le délai pour l'intervention s'il ne s'agit pas d'une urgence.

Dimensionnement des SSPI

Une grande partie des études ayant pour objet le bloc opératoire traitent de l'allocation en lits en salle de soins post opératoire (SSPI). Marcon *et al.* (2003) ont développé un modèle de simulation afin de déterminer une stratégie de planification adéquate. L'une d'elle est basée sur une planification ouverte (open scheduling). Dans ce cadre, un bloc opératoire est ouvert pour n'importe quelle spécialité. Chaque chirurgien organise ses interventions pour la journée, et chaque semaine une planification est

mise en place par spécialité, selon les demandes des différents chirurgiens. Un modèle mathématique est proposé afin de minimiser la date de fermeture des salles d'opération. Le modèle de simulation proposé permet alors de modéliser le parcours du patient, centré sur l'intervention chirurgicale (transport aller, anesthésie, préparation du chirurgien, intervention, réveil et transport retour). Les résultats montrent que les brancardiers sont la ressource goulot du problème.

Dans (Dussauchoy *et al.*, 2003), la simulation est utilisée pour étudier le fonctionnement du bloc opératoire, et plus particulièrement son dimensionnement en nombre de salles opératoires et de lits de réveil. On notera que cet article apporte une nouvelle méthode d'approximation des temps de passage au bloc opératoire, et que la règle d'ordonnancement appliquée ici est de type LPT. Plusieurs problématiques sont ainsi traitées : dimensionnement, réactivité du système aux aléas, planification des opérations.

Plus généralement, la modélisation de l'usage des lits dans le service de chirurgie a été abordée dans (Millard *et al.*, 2000), où des patients urgents ou non sont indifféremment admis. Le modèle montre que l'augmentation du nombre de lits dans l'unité de soins intensifs n'entraînera pas forcément la réduction du temps de séjour. En revanche, les méthodes visant à améliorer la condition du patient apportent des gains à plus long terme et réduisent les erreurs d'orientation.

1.5.3 Organisation et gestion de centres médicaux ambulatoires

Les études de cas portant sur les centres de soins ambulatoires se sont récemment multipliées. Nous attirons l'attention du lecteur sur la différence entre un centre médical ambulatoire et un service hospitalier ambulatoire. Dans le premier, il s'agit de petites unités de soins comportant un personnel soignant d'au plus une dizaine de personnes recevant les patients généralement sans rendez-vous. Dans le second, nous parlons d'un service appartenant à un hôpital où les patients sont reçus et libérés dans la journée pour une consultation, une intervention chirurgicale bénigne, etc. Dans les deux cas, la durée de séjour du patient doit rester courte afin de faire face aux éventuelles augmentations de fréquentation de l'unité. La simulation se révèle être un outil particulièrement bien adapté à l'étude d'un tel organisme, car il est très facile de construire un modèle simple, fidèle à la réalité et exploitable. Les critères généralement évalués sont les horaires d'ouverture, l'effectif du personnel, ou encore la taille des locaux.

Ferrin *et al.* (2004) se sont intéressés à l'application de la simulation dans le cadre d'un processus de prise en charge de patients avec une étape de chirurgie et l'utilisation d'unités post-anesthésie et d'unités de chirurgie ambulatoire, ce en plusieurs phases : développement du modèle conceptuel, codage de la simulation, expérimentation avec plusieurs scénarios de routine, notification des résultats de la simulation aux personnes concernées et formation des futur utilisateurs. Les résultats révèlent qu'une augmentation du régime de la salle d'opération ou de la procédure de pré admission n'est pas significative ; en revanche, une reconfiguration des unités de chirurgie ambulatoire s'avérerait utile, essentiellement sur le plan du dimensionnement en nombre de lits.

Ramis *et al.* (2001) se sont penchés sur la construction d'un modèle de simulation pour évaluer les différentes alternatives concernant la création d'un centre de chirurgie ambulatoire. Un tel centre doit permettre aux patients de rentrer en consultation et de ressortir opérés dans la même journée (sauf cas grave détecté). L'optimisation des conditions opérationnelles du système est alors une priorité, tout en réduisant les coûts et en améliorant la qualité d'accueil et de soin des patients. Il s'agit alors de modéliser les futurs locaux pour étudier les différentes conditions opératoires afin de maximiser le flux quotidien des patients. Les auteurs concluent sur le fait que la règle d'ordonnancement LPT est la meilleure dans l'optique de diminuer le temps d'attente des patients d'une part, et la date de fermeture du centre d'autre part.

Osidach et Fu (2003) ont construit un modèle de simulation afin de déterminer la configuration des locaux et la planification du personnel optimale dans l'optique de construire un centre d'examen mobile. Arena a été choisi pour simuler le flux de patients à travers les différents examens proposés.

Plusieurs distributions normales sont utilisées pour décrire le processus d'arrivée des patients, ainsi que les différents temps opératoires. Les auteurs ont pu, en conclusion, déterminer l'effectif médical idéal et prodiguer quelques recommandations quant à l'organisation du centre de soins.

Dans (Alexopoulos *et al.*, 2001), les auteurs se sont intéressés à l'application de la simulation dans le cadre de cliniques de proximité pour les plus démunis à travers un accord à but non lucratif. En effet les techniques de simulation s'appliquent particulièrement bien à ces petites structures et peuvent donner lieu à des optimisations de coûts de fonctionnement souvent cruciales. Le modèle créé tend à être le plus général possible afin d'être implanté dans le maximum de centres de soins possible ; il est à noter que le processus par lequel passe le patient lambda est toujours plus ou moins identique, à quelques variations près. Afin de permettre au personnel soignant de personnaliser l'application, une interface de paramétrage a été créée. Le choix d'Arena est motivé par son interface très intuitive. Une fois implémenté, les utilisateurs doivent être capable de lancer des expériences et d'interpréter les résultats facilement.

Enfin, la simulation est utilisée dans (Wijewickrama et Takakuwa, 2005) afin de déterminer la meilleure stratégie de planification de rendez-vous de patients dans une unité de consultation externe dans un environnement réaliste, l'accent étant mis sur la prise en compte des relations entre les sous unités (laboratoire, imagerie, etc.) constituant le centre de soin. Une analyse poussée de vingt règles de planification de rendez-vous est également réalisée, tout en tenant compte de leur degré de sensibilité sous certains facteurs environnementaux réalistes, telle la probabilité de non présence d'un patient ou de variations dans les durées des consultations. Aucune règle de pilotage ne se démarque, mais il est possible d'associer plusieurs règles à un environnement d'étude précis.

1.5.4 Organisation et gestion de systèmes multi-services

La littérature de la dernière décade comporte un nombre croissant de travaux traitant de la modélisation partielle ou intégrale d'établissements hospitaliers. Cela est essentiellement dû au fait que la dissémination de petits hôpitaux est très coûteuse, et qu'il s'agit aujourd'hui de les regrouper. Dans cette optique, il est nécessaire de rassembler une collection de méthodes et de connaissances pertinentes afin de la réutiliser dans un cadre qui se veut le plus général et le plus abstrait possible.

Modélisation de systèmes multi-services

La problématique concernant la modélisation de systèmes multi-services est abordée de plusieurs manières. Kao et Tung (1981) se sont penchés sur le problème d'allocation de lits pour les patients dans différents services d'un hôpital. Après avoir exposé la méthode utilisée pour l'étude analytique (utilisation de la théorie des files d'attentes), les auteurs testent le modèle ainsi construit grâce à la simulation, en comparant le fonctionnement de deux hôpitaux. Il s'agit alors de réallouer le nombre de lits dans chaque service de chacun des hôpitaux. Cette méthode est innovante dans la mesure où elle permet la réallocation en nombre de lit de l'ensemble du centre hospitalier. Les différents tests réalisés permettent de valider cette méthode, et ainsi de minimiser les sur-/sous-occupations de services.

Takakuwa et Shiozaki (2004) se sont penchés sur la simulation d'un service d'urgence en tenant compte des services immédiatement en relation. Cette étude a été réalisée en prévision de la construction d'un nouveau service d'urgence plus important dans le but d'accroître la capacité d'accueil, afin d'examiner le flux de patients et plus particulièrement leurs temps d'attente. Une classification des patients à l'admission a été réalisée, en fonction du mode d'arrivée (véhicule personnel, ambulance, etc.) et de la gravité. Les différents processus opératoires associés au flux de patients ont pu être déterminés et chronométrés, permettant ainsi de retracer avec précision le mouvement d'un patient au sein du service. L'expérimentation a permis de fixer le nombre de lits en soins intensifs, afin de diminuer significativement le temps d'attente des patients.

Plus récemment, Ashton *et al.* (2005) se sont concentrés sur la conception d'un modèle de simulation d'un complexe médical multi-services susceptible d'être agrandi. La première étape consiste donc à comprendre les activités associées aux patients, au service de traitement infirmier et à la clinique au moyen de diagrammes de flux. Cela donnera lieu à un modèle de simulation décomposé en trois parties (génération des arrivées des patients, traitements infirmiers, et processus propres à la clinique). Les patients arrivent avec ou sans rendez-vous, et sont traités par des infirmiers ou dans la clinique. Une fois les données collectées et le modèle validé, plusieurs scénarios ont été testés afin d'étudier le fonctionnement du centre à certains niveaux (nombre total de patients en attente dans le système, réorganisation des aires de réception, étude des différents systèmes de triage et planification du travail dans le service de traitement infirmier). Plusieurs effets de bords intéressants ont été remarqués, notamment l'intérêt des discussions entre dirigeants et chercheurs, très riches et bénéfiques pour la compréhension du système étudié. La simulation apporte un support visuel didactique non négligeable.

Modélisation de centres hospitaliers

La littérature récente fait état de quelques tentatives de modélisation intégrale d'établissements hospitaliers, indépendamment des services et ou de la spécificité du-dit hôpital. Moreno *et al.* (1999) ont créé un outil permettant la simulation de sociétés virtuelles complexes telles que les hôpitaux, destiné à des utilisateurs non informaticiens. Deux fonctions sont implémentées : observation de l'état du système à un instant donné, et simulation (prévisions sur les ressources à affecter). La modélisation utilisée ici repose sur une approche basée sur l'étude des différents flux existants dans un centre hospitalier. Celle-ci 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). Cependant, peu de détails sont donnés au sujet de la modélisation UML réalisée. En effet, la complexité d'un tel système implique une grande rigueur quant à sa description en terme d'objets. De plus, le détail inhérent à la modélisation de services particuliers n'est pas encore assez poussé pour satisfaire l'acuité des résultats attendus.

Sampath *et al.* (2006) se sont penchés sur l'utilisation de réseaux de Petri dans le but de modéliser un centre hospitalier complet en se basant sur une étude de cas. À ce jour il n'existe pas de système d'information suffisamment performant permettant d'obtenir une « photographie » de l'état de l'hôpital à n'importe quel instant. Le système actuel se révèle également incapable d'offrir une fonction de planification à moyen ou long terme des ressources du centre hospitalier. Les auteurs introduisent ainsi l'utilisation de réseaux de Petri comme un outil de modélisation, comportant des informations à propos des flux de patients, des unités hospitalières, des dépendances entre unités, des ressources, etc. Un tel modèle se prête tout naturellement à la simulation. Plusieurs stratégies sont étudiées concernant la construction d'un tel modèle en matière de suivi des ressources humaines ou matérielles du centre hospitalier :

- suivi de tous les patients et de toutes les ressources ;
- suivi de tous les patients et de toutes les classes de ressources ;
- suivi de toutes les classes de patients et de toutes les ressources ;
- suivi de toutes les classes de patients et de toutes les classes de ressources.

La première stratégie décrit un système d'information entièrement déterministe ; cela pose plusieurs problèmes, notamment le fait qu'il est impossible de connaître l'état de chaque ressource, et surtout qu'un tel modèle se révélerait trop gourmand en ressources. D'autre part, le suivi de toutes les classes de patients pose problème dans la mesure où le processus stochastique de chaque patient pris individuellement peut varier. Les auteurs s'orientent donc par élimination vers la deuxième stratégie. La construction du modèle est réalisée à partir de plusieurs entités de base, telles les ressources hospitalières (humaines ou non), les fonctions liées aux ressources, le flux des patients. Les auteurs décrivent ensuite une stratégie d'intégration

du système ainsi défini au sein de l'hôpital considéré en conjonction avec les outils informatiques déjà présents. Enfin, une étude illustrative décrit le parcours de patients dans le cas de douleurs abdominales inférieures droites (risques d'appendicites) grâce au modèle présenté. Sampath *et al.* (2006) concluent sur le succès concernant l'utilisation d'un tel outil dans l'optique de modélisation d'un système hospitalier important. De plus, un tel modèle se conjugue aisément à l'apport de la simulation, d'une interface graphique, d'une analyse mathématique. Enfin, l'utilisation de réseaux de Petri permet de nous orienter vers une approche de gestion d'un centre de soins en temps-réel.

1.6 Réutilisabilité des modèles

La réutilisabilité de modèles créés pour une étude de cas précise est un thème très discuté. Carter et Blake (2004) démontrent que la création d'un modèle de simulation pour un centre hospitalier particulier n'est pas implantable dans l'état dans n'importe quel hôpital, et illustrent ce fait par une étude de cas portant sur l'évaluation de l'impact de la planification chirurgicale élective sur l'allocation de ressources. Le projet se résumait à l'optimisation de la planification des horaires de travail des infirmières pour un certain nombre d'hôpitaux différents, tout en minimisant les horaires contraignants (week-end supplémentaires, plages de travail isolées, etc.). Dans cette optique, un logiciel destiné au personnel soignant a été développé, permettant le lancement de simulations et la modification de paramètres afin d'évaluer l'impact de changements sur les emploi du temps. L'outil ainsi développé devait être implantable dans tous les hôpitaux demandeurs, mais le modèle n'a pu être installé dans l'état dans d'autres centres hospitaliers, au prix de modifications au cas par cas. Carter et Blake (2004) concluent sur le fait qu'un centre de soin n'est pas un environnement comme les autres, et que son analyse et sa modélisation requièrent de grandes précautions. Les techniques usuelles en recherche opérationnelle peuvent parfaitement être appliquées dans un tel environnement, à condition de comprendre sa nature unique.

Sinreich et Marmor (2004) se sont concentrés sur le développement d'un outil de simulation simple et intuitif, mais conçu avec l'idée de réutilisabilité, en conservant un niveau d'abstraction suffisamment élevé. Dans cette étude, les flux de patients sont différenciés selon le service qui les recevra. Il est ainsi possible de caractériser chaque processus au sein du service des urgences indépendamment de l'hôpital. Un modèle général du processus d'arrivée des patients à trois niveaux a pu être créé : modélisation du processus d'arrivée dans le service d'urgence, dans le service d'imagerie, et modélisation des déplacements du personnel. Cette étude a le principal avantage de présenter les fondations d'un logiciel qui se veut indépendant de l'hôpital dans lequel il sera implanté.

D'une manière générale, de nombreuses études de cas nécessitant une modélisation préalable du système implique le choix d'un outil de modélisation (d'entreprise). Cependant, afin d'éviter la réitération 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 ce mémoire appartiennent à cette dernière catégorie.

1.6.1 Modèles génériques

Un outil permettant la simulation de centres hospitaliers destiné à des utilisateurs non informaticiens a été présenté dans (Moreno *et al.*, 1999). Plus récemment, une méthodologie de modélisation pour la simulation basée sur des méta-modèles génériques (MDA-UML) a été proposée dans (Roux *et al.*, 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.

Un modèle de connaissance générique est présenté dans (Chabrol *et al.*, 2006; Chauvet *et al.*, 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 et Kellert, 1991; Chabrol et Sarramia, 2000). Les processus ont ainsi été formalisés dans un modèle de connaissance ARIS. Le modèle de simulation a été réalisé sous Witness.

1.6.2 Plates-formes de modélisation et simulation

Pitt (1997) présente un outil de simulation dédié au personnel soignant exclusivement, basé sur un logiciel de simulation (WITNESS) et sur une interface graphique développée sous MS Visual Basic, appelé PRISM. Plusieurs caractéristiques décrivent cet outil : (i) simplicité d'utilisation (le personnel soignant doit être capable de l'utiliser), (ii) transparence (simplicité de l'interface graphique), (iii) interactivité (l'utilisateur doit pouvoir paramétrer entièrement les variables clefs afin d'examiner l'impact sur le résultat de la simulation), (iv) flexibilité (support d'un grand nombre de modèles et de scénarios), et (v) validation par le personnel médical. Le modèle de simulation utilisé est basé sur un réseau d'états/transitions (State Transition Network, STN) afin de satisfaire le besoin d'une modélisation centrée sur le patient. Les diagrammes ainsi produits sont facilement compréhensibles par le personnel médical, offrant une représentation proche de ce qu'ils utilisent déjà. Les principaux éléments de ces diagrammes sont les *états des patients*, les *évènements* conduisant à des *transitions*, et les *éléments de branchement conditionnel* qui déterminent le flux basé sur les attributs d'un patient donné. L'auteur se penche ensuite un cas d'étude, la modélisation du flux des patients dans les différents services d'un centre hospitalier. Des données réelles issues des archives de l'hôpital en question sont utilisées afin de générer les charges quotidiennes en nombre de patients. Le parcours de chaque patient individuellement modélisé est alors retracé à travers les différentes spécialités de soins et de traitements. Plusieurs variables peuvent être configurées (démographiques, admissions, configuration de l'hôpital, durée de séjour, simulation) afin d'obtenir un panel de résultats aussi large que possible. Les sorties, affichées grâce à l'interface graphique de PRISM, sont validées par le personnel médical : en effet, la majeure partie des résultats se révèle en accord avec la politique actuelle d'organisation de l'hôpital étudié (affectation en lits par exemple). Des projections ont également été réalisées. Plusieurs axes de recherches futures concernent la modélisation de services plus spécifiques et plus complexes.

Une plate-forme de modélisation appelée medBPM est décrite dans (Ramudhin *et al.*, 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 plate-forme 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 plate-forme de simulation multi-agents dédiée aux systèmes hospitaliers est introduite dans (Montreuil *et al.*, 2007). En cours de développement, cette plate-forme 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'oublis d'éléments importants et d'interactions. Charfeddine et Montreuil (2008) proposent une approche de mappage stratégique de réseaux de santé qui s'intègre dans le développement de cette plate-forme. Cette approche prend en considération les aspects organisationnels et les comportements spécifiques liés au domaine de la santé,

tout en conservant une représentation simple et compréhensible, notamment en intégrant plusieurs vues complémentaires du réseau.

1.7 Synthèse : apport de la simulation dans le domaine de la santé

La simulation à événements discrets est une technique de recherche opérationnelle particulièrement bien adaptée aux systèmes hospitaliers, permettant aux utilisateurs finaux (cadres médicaux, personnel hospitalier, etc.) de tester l'efficacité du système modélisé. Il est notamment possible de proposer de nouveaux scénarios, de construire de nouveaux systèmes, cela sans posséder une connaissance approfondie du programme de simulation. Le principal attrait de cette technique réside dans l'opportunité de modéliser des systèmes complexes, là où les méthodes analytiques deviennent trop compliquées.

Devant la prolifération d'articles utilisant cette méthode de résolution, il est raisonnable de s'interroger sur la valeur et l'impact de ces études : la simplicité d'usage est-elle synonyme d'apport scientifique faible ? Fone *et al.* (2003) se sont intéressés à l'utilité et la valeur de la simulation informatique appliquée dans les milieux hospitaliers. Les auteurs se concentrent essentiellement sur le fait que les modèles, bien que de plus en plus nombreux, ne sont pas forcément toujours adéquats et rigoureusement testés. De plus, aucune revue des méthodologies de modélisation dans le domaine de la santé n'a jamais été publiée.

Après une sélection et une revue de 990 articles publiés entre 1980 et 1999, 182 furent sélectionnés comme étant « pertinents » selon plusieurs critères d'évaluation définis par les auteurs. Le thème de l'organisation et de la gestion des hôpitaux apparaît dans 52 % d'entre eux (78 % sont originaires des États-Unis). Le thème le plus populaire relève de la planification et l'ordonnancement des admissions des patients pour la modélisation, pour les rendez-vous en clinique (ponctuels et courts séjours) et pour l'ordonnancement de blocs opératoires, afin d'optimiser l'utilisation des ressources et de minimiser les temps d'attente ou d'inactivité. Un grand nombre de systèmes ont été modélisés, afin d'étudier la productivité globale, les temps d'attente des patients, l'utilisation des ressources (humaines et/ou matérielles) ainsi que les coûts associés. La modélisation et le dimensionnement en nombre de lits apparaît également comme un facteur essentiel pour étudier différentes configurations possibles avant l'implantation. La localisation des services ambulanciers dépend d'un grand nombre de facteur corrélés tels que la densité de population, la densité d'accidents, le tracé routier, etc. Le nombre de vies sauvées peut être estimé au moyen de scénarios. Finalement, un grand nombre d'articles apportent des améliorations significatives à l'organisation de centres de soins d'une part, et d'autre part permettent l'identification de goulots d'étranglement au sein de ces systèmes.

En conclusion, les remarques suivantes peuvent être avancées au sujet de l'étendue et de la qualité de la simulation appliquée au domaine de la santé :

- le nombre de publications augmente, les domaines traités sont plus variés ;
- la qualité des articles est variable, mais augmente globalement avec le temps ;
- la qualité des papiers traitant du suivi de maladies est meilleure par rapport au thème de l'organisation et de la gestion ;
- il n'existe pas de revue « attitrée » pour les articles évalués ici.

Les auteurs remarquent également qu'il est difficile d'évaluer la qualité de l'implémentation des modèles présentés ainsi que leur usage car les informations à ce sujet manquent systématiquement. Finalement, il ressort de cette étude que la simulation constitue un outil puissant, offrant de nombreuses possibilités. Cependant, il est important de ne pas négliger les phases de validation et d'interprétation du modèle, afin de favoriser son exploitation.

Nous venons de présenter un état de l'art mettant en valeur les principales problématiques de la dernière décade propres au domaine hospitalier et les techniques de simulation utilisées visant à apporter

des solutions. Les études de simulation tendent, après s'être largement répandues dans les années 90, à se révéler de plus en plus pertinentes : les modèles se doivent d'être proches de la réalité, compréhensibles et si possible utilisables par les acteurs du système. Le panorama de thèmes en matière d'organisation et de gestion d'hôpitaux est maintenant très large, et les prochains axes de recherche comportent une volonté forte de généralisation : les méthodologies d'approche et de modélisation se rationalisent, la combinaison simulation/optimisation devient systématique, allant jusqu'à être intégrée dans les outils de simulation eux-mêmes. On notera par ailleurs que l'évolution de ces logiciels continue, proposant aujourd'hui de multiples possibilités en faveur de la communicabilité (création de modèle en trois dimensions) et du confort d'utilisation. Ainsi nous parlons aujourd'hui de simulation distribuée ou orientée objet, afin de tendre vers la création de normes en matière de modélisation hospitalière, telles que nous pouvons en trouver concernant la modélisation d'entreprise. Enfin, le domaine de l'ingénierie hospitalière a pris une telle importance qu'il constitue à lui seul un pôle de recherche et de développement très fertile, laissant présager un futur prometteur.

Chapitre 2

Méthodologie proposée et architecture de la plate-forme medPRO

Ce chapitre est divisé en deux grandes parties. La méthodologie proposée ainsi que les objectifs de la thèse sont détaillés dans un premier temps ; nous décrivons ainsi la problématique traitée dans ce mémoire et le positionnement de ce travail par rapport à la littérature existante. L'objectif consiste à poser les bases d'un outil de modélisation et de simulation dédié aux systèmes hospitaliers dans une optique de prototypage rapide. La méthodologie de travail est présentée, ainsi que les champs d'application et les destinataires d'un tel outil. Dans un second temps, l'architecture de la plate-forme medPRO supportant la méthodologie proposée est présentée : une approche systémique est utilisée pour l'architecture logique, tandis qu'une approche par couche de spécialisation est adoptée pour l'architecture logicielle. Enfin, les choix d'UML comme outil de modélisation et des réseaux de Petri pour la description du comportement dynamique sont justifiés.

2.1 Introduction

Nous avons choisi de placer la revue de littérature concernant l'analyse, la modélisation et la simulation de flux en milieu hospitalier avant la présentation des objectifs de la thèse afin de permettre au lecteur de s'imprégner du contexte scientifique du domaine de l'ingénierie des systèmes de soins. La plate-forme de modélisation et de simulation décrite dans ce mémoire a été pensée, définie et implémentée de manière à prendre en compte les enseignements des études présentées dans la revue de littérature et jugées significatives, suivant le processus de réflexion adopté tout au long de la thèse. La plate-forme présentée dans cet ouvrage n'est pas uniquement le fruit de recherches théoriques, mais bénéficie également des enseignements reçus lors de plusieurs études de cas menées sur le terrain.

Nous nous sommes efforcés de définir des objectifs simples mais innovants pour la spécification et la formalisation de cette plate-forme. Un processus de réflexion inscrit dans une démarche génie logiciel a été adopté afin de définir caractéristiques, limites, domaine d'application et utilisateurs ciblés. Nous en avons déduit l'approche méthodologique d'utilisation et les outils de base mis en œuvre pour les phases de modélisation et de simulation. La formalisation mathématique a permis la description d'un cadre d'utilisation adapté à chacune des classes de destinataires auxquelles s'adresse cet outil.

Dans un premier temps, nous détaillons l'objectif de la thèse ainsi que les moyens et les travaux mis en œuvre pour atteindre cet objectif section 2.2. Nous justifions par la même occasion les orientations que

nous avons choisies et mettons en avant la finalité de ces travaux de thèse au travers de la présentation de la plate-forme medPRO. Nous basculons dans un second temps sur la description de l'architecture de cet outil section 2.3 et mettons l'accent sur les différentes fonctionnalités proposées. Les choix réalisés pour la mise en œuvre de cet outil sont présentés section 2.4. Enfin, nous tentons de mettre en exergue la valeur ajoutée d'une telle plate-forme et de classifier ce travail à la lumière de la revue de littérature présentée dans le chapitre précédent dans la section 2.5.

2.2 Objectif de la thèse

2.2.1 Problématique

L'objectif de la thèse consiste à poser les bases d'un outil de modélisation et de simulation dédié aux systèmes hospitaliers. L'état de l'art présenté dans le chapitre précédent révèle que les outils de modélisation et/ou de simulation utilisés ne sont pas adaptés aux systèmes hospitaliers. En effet, un grand nombre de particularités les distinguent des systèmes manufacturiers classiques. La liste ci-après n'est pas exhaustive ni classée, mais décrit les principales différences que nous avons pu relever dans la littérature ou sur le terrain.

1. L'analyse d'un système hospitalier consiste en l'observation et la modélisation de flux de patients, et non de produits (impliquant la prise en compte du comportement des entités).
2. Le dossier médical influe sur l'ensemble du parcours patient au sein de l'unité médicale (administration de soins en fonction de la pathologie).
3. La prise en charge du patient implique l'intervention d'un grand nombre de ressources humaines très variées, allant du manutentionnaire au chirurgien.
4. Chaque service de soin possède une organisation spécifique, reposant sur une coordination précise des ressources humaines et matérielles.
5. Les services de soins d'un même hôpital sont hautement cloisonnés.
6. La coordination d'unités de soins différentes au sein d'un même hôpital est nécessaire afin d'assurer la prise en charge du patient selon sa pathologie.
7. Le pilotage du système est essentiel afin de réagir rapidement aux aléas (prise en charge des urgences notamment).

Le parcours d'un patient au sein de l'hôpital dépend de sa maladie et des traitements qu'il devra suivre : il est généralement impossible de savoir à l'avance quel sera son cheminement à travers les différents services médicaux d'un centre hospitalier. Certaines tâches sont séquentielles (le patient doit passer une radio, faire une prise de sang), d'autres peuvent être simultanées (examen des radiographies et/ou des prises de sang). Les durées opératoires des différentes tâches sont hautement stochastiques, tout comme les processus d'arrivée de ces mêmes patients. Les complications et urgences représentent un réel problème d'organisation, car il est nécessaire de traiter ces cas particuliers le plus rapidement possible, au risque de perturber le fonctionnement normal du système.

Ainsi la plupart des études existantes reposent sur une modélisation spécifique au problème étudié : il s'agit de l'approche généralement adoptée dans la littérature. Le modèle générique dédié aux systèmes hospitaliers qui encapsule les caractéristiques listées ci-dessus n'existe pas encore. Pourtant, les problématiques sont souvent similaires, et portent essentiellement sur l'organisation ou la réorganisation de services de soins. La simulation est par ailleurs (et à juste titre) un outil très utilisé, permettant la modélisation de flux complexes et l'obtention rapide de résultats. Enfin, nous avons remarqué au travers du chapitre précédent que les études de simulation dans le milieu hospitalier se sont multipliées durant les dix dernières

années, mais qu'elles se sont aussi fortement spécialisées : nous avons pu dégager deux grandes tendances si l'on analyse les travaux les plus récents.

La *généricité* est de mise dans la plupart des cas d'études : un effort est fourni pour gommer les spécificités des systèmes de soins étudiés, afin de proposer au final un outil « générique », pouvant être réutilisé dans un autre hôpital (c.f. section 1.6.1). Cette réduction est pourtant difficile à réaliser, parfois même impossible dans certains cas où l'organisation du système est trop spécifique (Carter et Blake, 2004). Nous avons effectivement constaté que chaque service de soin au sein d'un même hôpital adoptera une organisation particulière, parfois inédite. De tels choix sont souvent dus aux préférences des médecins (et/ou du personnel soignant d'une manière général), qui adapteront l'organisation du service à leurs habitudes et non l'inverse.

En marge de ces études, nous avons constaté l'apparition de *plate-formes* ou *modules* de modélisation et de simulation dédiés aux systèmes hospitaliers (c.f. section 1.6.2). Développées dans un premier temps par des grands de la simulation industrielle (Arena, MedModel, Witness, etc.), nous constatons un intérêt croissant pour ces outils permettant une analyse rapide d'un système quelconque. Aucun modèle générique n'est proposé, il s'agit généralement d'utiliser une « boîte à outils » dédiée aux systèmes de soins. Parallèlement à cela, plusieurs études font mention de méthodologies permettant l'accompagnement du personnel soignant durant l'étude et le changement d'organisation. Un nouveau regard est porté sur la communication et la compréhension entre les milieux du génie industriel et de la médecine au sens large.

2.2.2 Positionnement de la thèse

Les travaux présentés dans ce mémoire visent à la conception et à l'implémentation d'une plate-forme de modélisation et de simulation dédiée aux systèmes hospitaliers. Cet outil doit prendre en compte les spécificités des systèmes de production de soins énoncées dans la section précédente, pour offrir un environnement de modélisation parfaitement adapté à ces caractéristiques tout en évitant les écueils mis en exergue dans le chapitre précédent. Nous nous plaçons dans une optique de prototypage rapide, permettant de dégager rapidement les problèmes organisationnels du système étudié. La taille du système ciblé correspond à celle d'un service de soins : ce choix nous permet de proposer un ensemble d'outils et de recommandations pertinentes pour le service étudié sous la forme d'un guide méthodologique destiné à l'utilisateur.

Nous insistons sur le fait qu'il ne s'agit pas de modèles génériques, mais d'environnements de développement permettant la modélisation de systèmes dont l'organisation varie d'un hôpital à l'autre. Par exemple il est possible de proposer un certain nombre de primitives de base dédiées aux blocs opératoires sans imposer une organisation (prise en charge du patient, politique de réservation de salles opératoires, structure physique) qui variera sans doute d'un hôpital à l'autre.

Si nous laissons la généricité de côté, nous nous attachons à proposer un outil destiné à plusieurs catégories de public (hospitaliers, chargés d'étude et développeurs) avec plusieurs interfaces adaptées en fonction de l'utilisateur. L'outil doit être implantable dans le service de soins, et sera doté de fonctionnalités adaptées à l'organisation du système considéré et utiles au personnel de ce système.

Les méthodes de résolution scientifiques abordées sont centrées sur la simulation. Cependant un certain nombre de fonctionnalités mettant en œuvre des méthodes exactes pour la planification et l'ordonnancement sont également proposées.

2.2.3 Méthodologie de travail

La plate-forme de modélisation et de simulation décrite dans ce mémoire doit permettre le prototypage rapide d'un système médical au sens large (service de soins, pôle hospitalier, bloc opératoire, etc.) tout en

évitant les écueils classiques inhérents à la spécificité du domaine. Une méthodologie de travail présentée figure 2.1 a été mise au point (Augusto *et al.*, 2007).

Pour un **problème** donné, lié à un système hospitalier, nous proposons une solution de **prototypage rapide** dans le but de fournir une **réponse** appropriée grâce à la *simulation*. Le mot « problème » doit être considéré au sens large : il s'agira par exemple de la restructuration d'un service impliquant la mise en place d'une nouvelle organisation à tester, de la conception d'une nouvelle unité de soin, ou encore de l'analyse du temps de séjour des patients atteints d'une certaine pathologie. Le dénominateur commun de ces problèmes réside dans la phase de modélisation, étape essentielle pour la validation de notre compréhension du système en accord avec le personnel médical initiateur de l'étude.

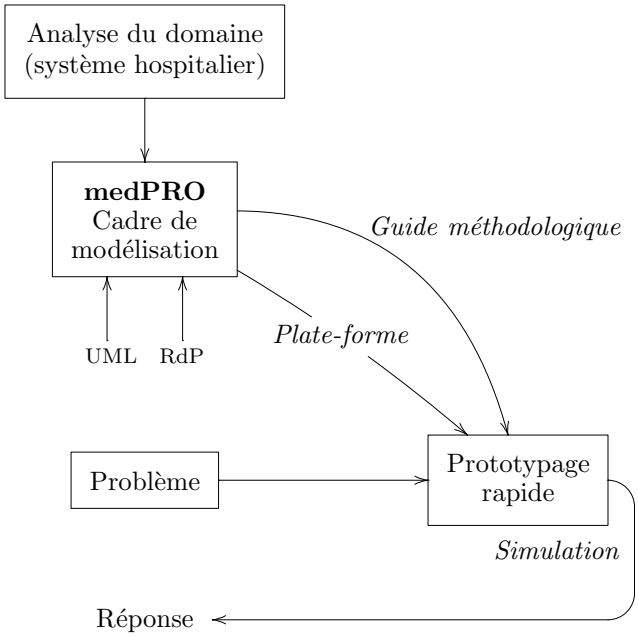


FIG. 2.1 – Méthodologie de travail globale

La mise en œuvre de cette stratégie de prototypage rapide est possible au travers de l'utilisation d'un **cadre de modélisation** appelé **medPRO** (medical Process-Resource-Organisation modeling) : il s'agit d'un environnement de travail encapsulant les caractéristiques d'une unité médicale particulière, basé sur les outils UML pour la représentation/modélisation et les réseaux de Petri pour la formalisation dynamique et la simulation. Cet outil se doit d'être fortement intuitif afin de faciliter les phase de modélisation (côté concepteur) et de communication (côté demandeur). Un tel cadre de modélisation est créé d'après une **analyse préliminaire du domaine hospitalier**. Nous décrirons plus avant dans la suite de chapitre les différents niveaux d'analyse requis à cette étape.

Une *plate-forme de modélisation* et un *guide méthodologique* permettent la modélisation et la simulation de cette unité, afin de répondre au 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 soins particulier : recommandations de bonnes pratiques dans le cas du service de pharmacie d'un hôpital, consignes de prise en charge du patient, coûts d'utilisation de ressources critiques, etc. Ce guide permet de poser les premiers repères pour aider l'utilisateur au cours de la modélisation réalisée grâce à la plate-forme medPRO. Il existe un guide différent pour chaque unité de soins. Trois exemples de guides méthodologiques sont présentés dans les chapitres 6, 7 et 8 pour l'unité neuro-vasculaire, la pharmacie et le bloc opératoire respectivement.

2.2.4 Champ d'application

Le champ d'application de la plate-forme medPRO a été précisément délimité afin de proposer un outil centré sur le prototypage rapide de systèmes hospitaliers. Ainsi la plate-forme medPRO peut être utilisée pour la modélisation et la simulation d'un système hospitalier quelconque en utilisant les outils proposés dans le cadre de modélisation. Son usage est comparable dans ce cas à l'usage que l'on ferait d'un logiciel de simulation classique, à la différence près qu'il existe un certain nombre de fonctionnalités uniques liées au milieu médical. En revanche, nous nous sommes efforcés d'analyser plusieurs systèmes de soins particuliers afin d'en capturer les particularités et d'offrir un certain nombre de modules spécifiques. Pour résumer, une analyse à deux niveaux est possible :

Analyse opérationnelle à partir de zéro : le système est modélisé avec les outils de bases proposés dans la plate-forme. L'utilisateur doit fournir un effort particulier pour la représentation des mécanismes uniques qui font la particularité du système. L'éventail des outils proposés doit être suffisamment large pour couvrir les primitives de base permettant la modélisation intégrale du système considéré. Ce type d'approche est comparable à l'utilisation d'outils de modélisation et/ou de simulation généralistes.

Analyse cadrée par un guide méthodologique : les caractéristiques du système sont incluses dans un guide méthodologique et un certain nombre de modules spécifiques permettent une analyse poussée de ce système. Ces modules permettent la modélisation de flux particuliers ou d'un processus de décision nécessitant la mise en œuvre d'une logique algorithmique poussée. Nous nous plaçons ici dans un cadre proche des modules spécialisés proposés par MedModel ou Witness par exemple.

Les deux analyses proposées ne sont pas disjointes : la modélisation d'un système à partir de primitives de bases n'empêche pas l'utilisation de modules de pilotages avancés, et réciproquement. De plus la mise en œuvre d'une modélisation à plusieurs niveaux et selon plusieurs vues (c.f. section 2.3) permet si nécessaire d'affiner la modélisation selon les besoins.

2.2.5 Destinataires

La plate-forme medPRO est un outil de modélisation et de simulation destiné à plusieurs classes d'utilisateurs différents. Ce logiciel est constitué de plusieurs couches d'abstraction distinctes (décrites dans la section suivante) permettant un accès aisé aux différents utilisateurs selon leur statut professionnel :

Personnel hospitalier : les membres du personnel hospitalier (médecins, infirmières, etc.) sont les premiers concernés par les changements organisationnels qui résulteront de l'étude de cas concernant leur unité de soins ou leur hôpital. Ces personnes doivent être capables de comprendre, de corriger et de valider la représentation abstraite de leur système à travers un modèle théorique. Il en va de même concernant la simulation, l'extraction et la présentation de résultats. Enfin, les membres du personnel soignant doivent être capable d'utiliser l'outil dans un cadre éventuellement plus restreint et adapté à leurs besoins quotidiens.

Chargé d'étude : nous entendons par « chargé d'étude » les chercheurs, ingénieurs ou consultants désireux de modéliser rapidement et efficacement un système hospitalier au sens large. La plate-forme doit offrir une liberté assez large pour permettre la modélisation et la simulation de systèmes complexes en capturant leurs particularités fondamentales, mais aussi assez restreinte pour éviter de perdre du temps dans la représentation de mécanismes anecdotiques nécessitant un « bricolage ».

Développeurs : le mécanisme de formalisation permettant la simulation des modèles réalisés doit rester accessible dans le cas où certaines particularités liées au système à modéliser nécessiteraient un travail de codage tout particulier.

Dans les trois cas 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 de la modélisation de flux en milieu hospitalier, encapsulant les caractéristiques listées en introduction.

2.3 Architecture

La plate-forme medPRO est bâtie sur le modèle d'un logiciel d'analyse, de modélisation et de simulation de flux. Son architecture est comparable sur de nombreux points à celle de logiciels commerciaux tels que ProModel, Arena ou encore Witness. Nous proposons une philosophie d'utilisation similaire avec une interface de modélisation intuitive. Cependant, le concept de modélisation ainsi que la formalisation du modèle et sa simulation reposent sur des mécanismes fondamentalement différents. De plus nous proposons une interface différente selon l'utilisateur par l'intermédiaire de couches logicielles.

2.3.1 Couches logicielles

La plate-forme medPRO est composée de trois couches logicielles distinctes, présentées figure 2.2. Chaque couche de la plate-forme dépend de la couche inférieure. La première couche est l'**interface spécifique au service de soin étudié**. Il s'agit d'un formulaire simplifié permettant le renseignement de paramètres connus par le personnel hospitalier. Cette interface est spécialement conçue pour les acteurs du système étudié (i.e. le personnel soignant). Ces derniers ont ainsi accès au prototypage rapide de leur système, sans avoir à entrer dans le détail du modèle de simulation. Cette couche logicielle est interchangeable en fonction du système modélisé et permet ainsi un confort d'utilisation supplémentaire pour l'utilisateur. Plusieurs exemples d'interfaces personnalisées sont présentés dans les chapitres 7 et 8 où un outil d'optimisation spécifique est intégré à ce niveau.

La couche immédiatement inférieure regroupe l'**interface de modélisation**, destinée au chargé d'étude et au personnel hospitalier. Il s'agit d'un ensemble d'outils basés sur UML permettant la modélisation du parcours patient, de l'activité des ressources et de l'organisation du système. Ainsi cette couche logicielle contient une représentation semi-formelle du système étudié, permettant de favoriser un échange autour du modèle lors de réunions avec le personnel soignant et de provoquer des débats d'idées dans un but de validation. L'architecture de cette couche sera décrite plus avant dans le chapitre 3.

Enfin, la couche la plus basse permet la génération et la modification du modèle de simulation sous forme de réseaux de Petri. La conversion permet de formaliser le modèle et de le convertir en modèle de simulation à évènements discrets. La rigueur des réseaux de Petri offre une représentation claire des mécanismes dynamiques de simulation et de prise de décision. Ces mécanismes seront décrits plus avant dans le chapitre 4.

2.3.2 Plate-forme de modélisation

Une approche systémique (von Bertalanffy, 1968) a été adoptée au sein de la plate-forme de modélisation medPRO. À la différence de la décomposition analytique, on ne cherche pas à descendre au niveau des composants élémentaires mais à identifier les sous-systèmes (modules, organes, sous-ensembles, etc.) qui jouent un rôle fondamental dans le fonctionnement du système. Cela suppose de définir clairement les frontières de ces sous-systèmes (pour faire ensuite apparaître les relations qu'ils entretiennent entre eux ainsi que leur finalité par rapport à l'ensemble). De la même manière que pour les systèmes manufacturiers, une décomposition systémique nous permet de diviser un système hospitalier en trois sous-systèmes (Gourgand et Kellert, 1991) :

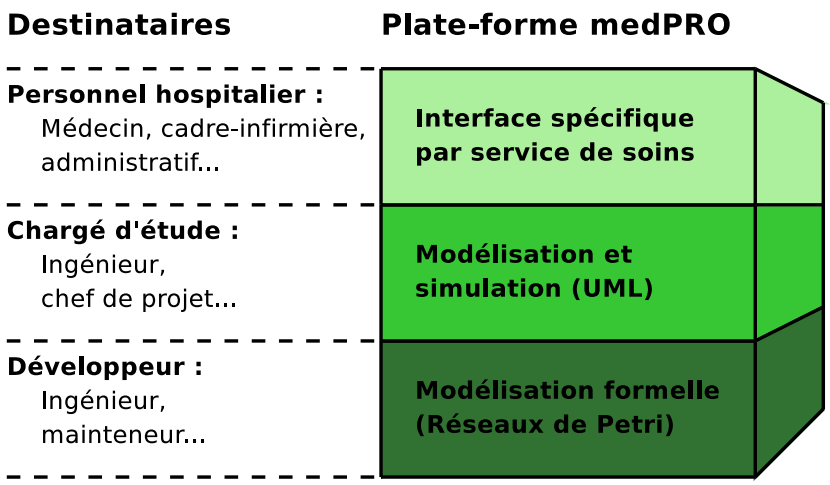


FIG. 2.2 – Les trois couches logicielles de medPRO

Opérationnel : également appelé sous-système physique, ce module offre une modélisation du système réel et des flux qui le composent. Cette modélisation peut être structurée en plusieurs vues, comme nous allons le détailler dans le chapitre 3. Schématiquement, le sous-système physique contient des informations relatives au parcours patient, à l'activité du personnel hospitalier, à l'utilisation des ressources matérielles et à l'organisation physique et logique du système.

Décisionnel : également appelé sous-système de pilotage, il s'agit d'un ensemble de centres de décision conçus pour piloter le sous-système opérationnel : création d'ordres, de plannings, allocation de ressources. Le sous-système décisionnel doit être vu comme un outil d'aide à la décision dont les degrés de liberté sont fixés par l'utilisateur lors de la phase de modélisation.

Informationnel : le sous-système informationnel fait le lien entre le sous-système physique et le sous-système de décision ; 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. Le sous-système informationnel doit être vu comme une base de données dont le modèle conceptuel correspond au modèle opérationnel du système. Ce sous-système permet également de créer un pont avec le système d'information de l'unité étudiée.

Une telle décomposition s'applique parfaitement aux systèmes hospitaliers où l'opérationnel, l'informationnel et le décisionnel sont trois aspects fondamentalement distincts. 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 existante). 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 par le biais d'outils d'interrogation tel SQL.

De nombreuses données de natures très différentes caractérisent un système médical, très souvent 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 parcours patient et l'organisation du système interagit continuellement avec les deux autres parties. La figure 2.3 offre une représentation épurée de l'architecture de la plate-forme de modélisation medPRO.

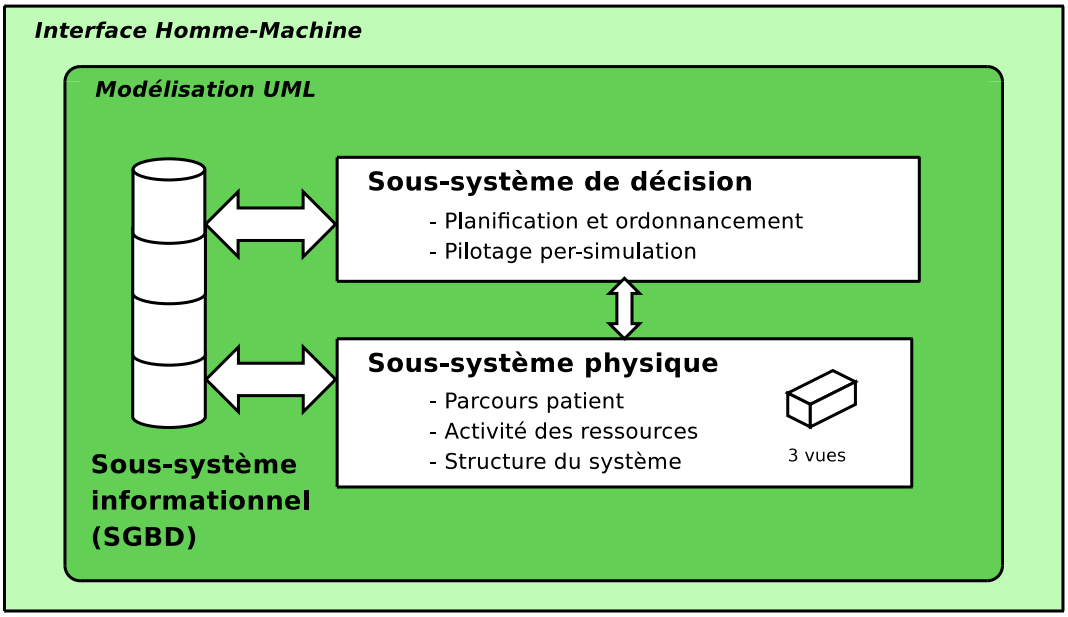


FIG. 2.3 – Architecture générale de la plate-forme medPRO

La couche spécifique à certains systèmes hospitaliers présentée figure 2.2 se superpose à cette architecture grâce à 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. Prenons l'exemple de la pharmacie : un certain nombre d'organisations décrites dans le guide méthodologique associé pourront être proposées en amont de la modélisation, permettant de poser les bases du projet (mode d'approvisionnement et/ou de distribution des médicaments, classement de solutés ou encore organisation des transports).

2.4 Justification des choix de spécification

L'objectif de ce travail est la conception d'un outil de prototypage rapide dédié aux systèmes hospitalier. Une fois l'objectif fixé, il a fallu réaliser un certain nombre de choix quant aux orientations à adopter. Nous justifions ces choix dans cette section.

2.4.1 Vers une approche orientée simulation

La simulation à événements discrets est un outil flexible permettant de diagnostiquer rapidement les problèmes organisationnels de n'importe quel système de production, aussi complexe soit-il. La revue de littérature présentée dans le chapitre précédent (c.f. section 1.5) atteste ce fait avec la multiplication des études de ce type : il ressort que la simulation est l'outil idéal pour la détection rapide de problèmes organisationnels. Carter (2007) affirme en particulier que la simulation est un outil puissant pour l'analyse des services d'urgence (archétype de l'unité de soins « complexe »), malgré le fait que les études de simulation peuvent être difficiles et semées d'embûches. L'outil proposé dans cette thèse se place ici comme un intermédiaire, permettant d'utiliser la simulation dans un cadre hospitalier en minimisant les

risques d'erreurs et d'incompréhensions récurrentes lors de l'analyse de systèmes de production de soins. Il s'agit de tirer les enseignements prodigués dans certaines études de cas, en particulier au niveau de la méthodologie de modélisation et de communication (Eldabi et Paul, 2001).

La simulation n'est pas sans défauts. Les lacunes de cette technique au niveau du système de décision ont été abordées dans des domaines différents (Valckenaers *et al.*, 1997; Klein et Thomas, 2006; Blanc *et al.*, 2006) et des solutions ont été proposées, permettant l'amélioration du contrôle de la simulation. La simulation n'est pas adaptée à la planification de ressource ou à l'ordonnancement ; ces thèmes sont pourtant au cœur de problématiques importantes telle la planification de bloc opératoire. Plusieurs avancées scientifiques ont permis de pallier à ce problème en combinant la simulation avec des techniques d'optimisation (Baesler et Sepúlveda, 2001; Albert et Marcon, 2005; Centeno *et al.*, 2003; Proth et Xie, 1995a) ; la simulation est alors utilisée comme un outil de validation car elle permet d'introduire processus et durées stochastiques dans le problème sans pour autant le complexifier.

2.4.2 Un outil de modélisation pour la communication : UML

Nous avons remarqué dans le panorama présenté section 1.5 que la communication et l'accompagnement au changement du personnel hospitalier a compté pour beaucoup dans le succès de l'étude de cas (Alexopoulos *et al.*, 2001; Su et Shih, 2003) ; Anderson (2002) considère que l'évaluation de l'impact de l'implantation d'un nouveau logiciel (quel que soit son type) au sein d'un centre hospitalier mérite une attention toute particulière. Eldabi et Paul (2001) rappellent que la communication constitue une part importante de la modélisation. Enfin, nous avons pu constater dans (Trilling *et al.*, 2004) que le choix d'un outil pour la modélisation d'un centre hospitalier n'est pas anodin : les critères importants sont la clarté de la modélisation, la modularité, l'exhaustivité et son impact sur la communication avec le personnel hospitalier.

Un sous-ensemble de diagrammes UML ont été choisis pour la modélisation opérationnelle et organisationnelle au sein de la plate-forme medPRO. Outre les enseignements tirés de la littérature, plusieurs raisons permettent de justifier ce choix :

1. UML est basé sur une représentation graphique simple à comprendre et à manipuler : l'utilisation de diagrammes d'état permet la modélisation précise de flux en tenant compte des aspects dynamiques du système. Ces diagrammes peuvent être présentés en réunion, discutés et modifiés par les membres du personnel hospitalier eux-même.
2. UML peut (doit) être spécifié en fonction des besoins : souvent citée comme un défaut, la liberté d'action de l'utilisateur d'UML peut être délimitée de manière à restreindre l'utilisation de cet outil à un cadre précis, en l'occurrence le domaine hospitalier.
3. UML est un langage de modélisation orienté objet : cette norme permet de mettre en regard le modèle réalisé avec n'importe quelle interface de communication, avec une base de données, un langage de programmation, un autre outil de modélisation ou de simulation.
4. UML peut être étendu. Cela a déjà été mis en pratique pour la spécification de la plate-forme medPRO, et peut se poursuivre pour d'éventuelles extensions.

Nous avons choisi UML en premier lieu pour ses atouts en matière de communication. Testé à plusieurs reprises sur le terrain (c.f. chapitres 6 et 7), cet outil s'est révélé très simple à assimiler, à comprendre et à modifier lors de réunions avec le personnel hospitalier. Des impressions similaires ressortent de (Vasilakis et Kuramoto, 2005; Ramudhin *et al.*, 2006) où le formalisme de modélisation utilisé est proche voire identique. Enfin, les principaux inconvénients d'UML peuvent être contournés de manière à obtenir un langage de modélisation à la fois simple et rigoureux. La spécification d'UML et les règles d'utilisation sont décrites dans le chapitre 3.

2.4.3 Un outil de modélisation pour la simulation : les réseaux de Petri

Si UML constitue un excellent langage pour la modélisation et la communication, il n'existe aucune formalisation ou méthodologie permettant de décrire le comportement dynamique de modèles réalisés au moyen de diagrammes d'état. La formalisation dynamique du modèle proposé est une étape incontournable, permettant de déboucher automatiquement sur la simulation. Au lieu de définir un ensemble d'algorithmes d'exécution dédiés à la simulation des modèles réalisés en UML, nous avons choisi d'utiliser les réseaux de Petri pour décrire cette dynamique d'exécution.

Les réseaux de Petri ont rarement été utilisés dans le domaine de la modélisation/simulation de flux en milieu hospitalier, mais les rares études qui en font état ont été très fructueuses (Celano *et al.*, 2006; Sampath *et al.*, 2006) : les réseaux de Petri constituent un excellent outil pour s'orienter rapidement sur la simulation :

1. Les propriétés mathématiques des réseaux de Petri permettent l'application de méthodes et de résultats intéressants publiés dans la littérature.
2. Les réseaux de Petri se prêtent particulièrement bien à la planification et l'ordonnancement de systèmes de production (Proth et Xie, 1995a) : ces méthodes peuvent être transposées au milieu hospitalier de manière transparente.
3. Le formalisme des réseaux de Petri permet une description du comportement dynamique d'un système de manière précise en faisant évoluer le marquage du réseau.
4. Les réseaux de Petri peuvent être interfacés avec un système de pilotage.

L'intégration des réseaux de Petri est réalisée de manière à décrire le comportement des modèles UML proposés. La complexité inhérente aux réseaux de Petri est donc invisible aux yeux du personnel hospitalier. En revanche, l'enrichissement du réseau de Petri pour la description de comportements particuliers reste possible du point de vue du développeur ou du chargé d'étude. La conversion des modèles UML ainsi que la spécification des réseaux de Petri et leur simulation sont décrites dans le chapitre 4.

2.4.4 Environnement de développement

L'environnement et les bibliothèques de développement de medPRO ont été choisis en fonction de trois critères : (i) cohérence avec le langage de modélisation (UML), (ii) liberté d'utilisation, et (iii) portabilité du code. Ces deux derniers critères sont primordiaux lorsque les études concernent des hôpitaux à but non-lucratif : les équipements informatiques de ceux-ci sont généralement anciens et pauvres en logiciels. De plus l'investissement dans des outils coûteux tels que CPLEX ou Arena (entre autres) n'est généralement pas justifié pour la plupart des études d'après la revue de littérature présentée dans le chapitre précédent.

Nous avons choisi de développer la plate-forme medPRO à partir de zéro en utilisant le langage Java. Grâce au principe de la machine virtuelle, le code Java est théoriquement exécutable sur n'importe quelle machine, quel que soit son type ou son système d'exploitation. L'utilisation de bibliothèques de développement libres telle que GLPK pour l'optimisation est également préconisée afin de garantir un coût minimal d'exploitation pour l'hôpital concerné. De plus un certain nombre de plates-formes de simulation existantes sont développées en Java, permettant la mise en place d'extensions éventuelles vers la simulation multi-agents (Bellifemine *et al.*, 2007). Enfin le développement intégral d'une plate-forme est certes coûteux en temps mais permet la mise en œuvre d'un outil fidèle aux spécifications scientifiques énoncées dans ce mémoire.

Les aspects génie logiciel seront à peine effleurés dans ce recueil afin de ne pas dénaturer le contenu scientifique proposé. Deux guides pour l'utilisateur et pour le développeur seront réalisés en marge de cet ouvrage pour faciliter l'utilisation de l'outil ainsi que sa maintenance.

2.5 Originalité de l'approche proposée

Nous proposons dans ce mémoire la spécification d'une plate-forme d'analyse, de modélisation et de simulation appelée medPRO (medical Process-Resource-Organisation modelling). UML a été choisi pour la modélisation et la présentation tandis que les réseaux de Petri ont permis la formalisation d'un point de vue dynamique. Cet outil s'adresse à plusieurs types d'utilisateurs différents par l'intermédiaire d'interfaces adaptées. 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 pilotage. La simulation appliquée au réseaux de Petri généré permet l'obtention rapide de résultats, sélectionnés et interprétés en fonction du service médical étudié.

La spécification et l'implémentation de la plate-forme medPRO ont été réalisées en suivant un processus de type génie logiciel. Le cahier des charges de l'outil a été établi à partir : (i) des enseignements tirés de la revue de littérature sur les études de modélisation et de simulation appliquées au milieu hospitalier (c.f. chapitre 1), et (ii) des études de cas réalisées sur le terrain durant la thèse dans trois services différents (c.f. chapitres 6, 7 et 8). L'ensemble des caractéristiques originales de medPRO que nous avons jugé primordiales pour un outil dédié à l'analyse de flux en milieu hospitalier sont synthétisées comme suit :

1. **Formalisme de modélisation :** la différenciation entre le formalisme de modélisation pour la communication et pour l'exécution permet de faire participer le personnel hospitalier lors des réunions (Ramudhin *et al.*, 2006). L'intégration du processus de modélisation dans une méthodologie d'accompagnement (Eldabi et Paul, 2001) est envisageable.
2. **Prise en compte des spécificités liées au milieu hospitalier :** la spécification d'UML permet l'intégration de modules personnalisés selon les processus observés dans un environnement médical.
3. **Partitionnement des données :** l'organisation des données selon l'architecture proposée permet de manipuler plusieurs types d'informations de manière transparente et automatique. La distinction opérationnel/informationnel/décisionnel est une caractéristique absente des plus grands outils informatiques de simulation disponibles.
4. **Conversion automatique modélisation → simulation :** ce processus de conversion est difficile à mettre en œuvre car il requiert la mise en place d'algorithmes permettant de prendre en compte à la fois les spécificités du modèle et du logiciel de simulation. Spécifier le modèle grâce à UML nous permet d'établir un outil de modélisation sur-mesure et prêt à être converti en réseau de Petri pour son analyse et sa simulation.

La suite de cet ouvrage est consacrée à la description détaillée de la méthodologie de modélisation proposée et à la plate-forme medPRO qui s'appuie sur cette méthodologie.

Chapitre 3

Modélisation du système opérationnel

Ce chapitre présente l'approche adoptée pour la modélisation du système opérationnel et regroupe l'ensemble des spécifications de l'outil de modélisation intégré à la plate-forme medPRO. Basé sur UML, les modalités d'utilisation de cet outil sont détaillées en respectant l'architecture du sous-système physique. Une série de définitions préliminaires permettent d'identifier clairement les termes et notations utilisés dans la suite de ce mémoire. Le processus de construction du modèle dans chacune des vues proposées (processus, ressource et organisation) est décrit ; les détails des interactions entre ces vues, ainsi que les avantages et les limites d'une telle architecture sont discutés en fin de chapitre.

3.1 Introduction

L'approche de modélisation proposée dans la plate-forme medPRO est basée sur UML. Ce langage de modélisation unifié constitue un puissant outil de communication visuel grâce à une palette d'éléments graphiques facile à apprendre et à comprendre. Cependant les spécifications d'utilisation de ce langage doivent être définies en fonction du domaine d'application : UML a été conçu pour être ouvert à toute interprétation. Cette liberté implique un travail important au niveau de l'établissement des modalités d'utilisation des caractéristiques du langage. Ce chapitre permet de délimiter le cadre d'utilisation d'UML en tant qu'outil de modélisation dédié aux systèmes de santé. Selon le contexte, les outils et formalismes graphiques utilisés pour la modélisation de systèmes de soins sont précisément redéfinis.

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 processus (ou vue patient) regroupe l'ensemble des flux de patients à travers le système de soin représenté ainsi que leurs interactions avec les ressources du système. La vue ressource (ou vue comportementale) permet de décrire pour chaque ressource (humaine ou matérielle) son mode opératoire par le biais de missions ponctuelles. Enfin, la vue organisation offre une représentation globale du système, et synthétise la déclaration des entités en interaction dans le système ainsi que leurs caractéristiques et leurs relations. La description d'un système en adoptant différents points de vue permet une identification plus facile et plus rapide des différents types d'informations modélisés.

Après une série de définitions préliminaires présentées sections 3.2 et 3.3, les spécifications du sous-système physique sont décrites de manière séquentielle en présentant successivement les trois vues permettant sa représentation : processus (section 3.4), ressource (section 3.5), et organisation (section 3.6). L'intérêt d'une telle représentation ainsi que les interactions qui existent entre ces trois vues sont présentés section 3.7. Un certains nombres d'extensions seront données section 3.8.

3.2 Préliminaires

Nous nous attarderons dans cette section à la définition et à la description de notions utilisés dans la suite de ce mémoire. Nous définissons ainsi un vocabulaire précis qui nous permettra de décrire le cadre de modélisation medPRO sans ambiguïté par la suite. Les définitions données dans cette section sont informelles ; elles permettent de délimiter les contours de termes généraux souvent utilisés abusivement dans la littérature.

3.2.1 Système hospitalier

Un système hospitalier est divers et complexe, composé d'établissements de tailles et de statuts variables (public ou privé, lucratif, etc.), distinctions auxquelles s'ajoutent aujourd'hui des critères d'activité et d'environnement. D'un point de vue systémique, un système hospitalier est qualifié comme étant un système socio-technique dont la mission principale est de prodiguer les meilleurs soins aux patients (Boumane *et al.*, 2006).

Nous appelons « système hospitalier » tout système physique comportant un ou plusieurs flux de patients sans tenir compte de la taille du système.

3.2.2 Unité de soins

Une unité de soins accueille les patients hospitalisés, chaque unité de soins étant spécialisée selon les disciplines médicales ou la qualité des patients. Dans le langage courant, une unité de soin désigne une structure de prise en charge médicale beaucoup plus petite qu'un système hospitalier où les patients sont amenés à séjourner pendant plusieurs jours. Nous distinguons les unités de long, moyen et court séjour :

Unité de long séjour : établissement ou partie d'établissement qui a pour mission d'assurer l'hébergement de longue durée des personnes ayant perdu leur autonomie de vie et dont l'état nécessite une surveillance médicale constante et des traitements d'entretien ;

Unité de moyen séjour : service hospitalier permettant d'assurer, après la phase aiguë de la maladie, le prolongement temporaire de soins actifs ainsi que les traitements nécessaires à la réadaptation en vue du retour à une existence autonome ;

Unité de court séjour : service visant à prendre en charge le patient sur une durée très courte, généralement à l'occasion d'une intervention chirurgicale bénigne ou d'une traitement médical ponctuel.

3.2.3 Prise en charge ambulatoire

Le terme ambulatoire désigne un traitement qui ne nécessite pas l'hospitalisation du malade. Une prise en charge ambulatoire, à l'opposé d'une hospitalisation traditionnelle, permet de diagnostiquer, de traiter, de suivre un patient sans que celui ci ne soit hospitalisé. Une intervention chirurgicale ambulatoire est un geste chirurgical qui se fait dans la journée, et qui ne nécessite pas que le patient séjourne à l'hôpital dans une unité de soins.

3.3 Concepts de base

Ce chapitre regroupe un certain nombre de définitions formelles, de règles d'utilisation et d'exemples permettant la modélisation précise d'un système hospitalier dans les meilleures conditions. Seuls les diagrammes de classes et les diagrammes d'état seront utilisés. Les définitions suivantes décrivent les briques de base du cadre de modélisation medPRO.

3.3.1 Classe, objet, attribut et entité

Une classe représente un groupe d'objets possédant des états et un comportement communs. Une classe peut être considérée comme une sorte de plan permettant de construire des objets dans un système orienté objet. Une classe possède un nom ainsi que plusieurs attributs qui la caractérisent. La plupart des concepts présentés dans ce chapitre seront modélisés par des classes. Par exemple, l'ensemble des infirmières diplômées d'état (IDE) d'un hôpital peuvent être représentées au moyen d'une classe nommée `IDE`. Chacune des infirmières qui travaillent dans l'hôpital est une *instance* de cette classe, également appelée *objet*.

Par exemple, il est possible de définir plusieurs instances de la classe `IDE` en fonction de la spécialité médicale : neurologie, cardio-vasculaire, etc. Chaque instance de la classe `IDE` est un objet à qui l'on peut attribuer un nom propre. Un objet peut également être anonyme. Tous les objets qui interagissent dans le système doivent être déclarés préalablement sous forme de classe.

Un attribut est une donnée primitive simple (entier, réel, chaîne de caractères) permettant de caractériser une classe. Dans la plate-forme medPRO, définir une classe revient à créer une table dans le système d'information : les champs de cette table correspondent aux attributs de la classe, tandis que les enregistrements correspondent aux valeurs de ces attributs pour les instances de cette classe.

Une entité est une classe permettant de représenter patients, médicaments, ou tout autre acteur ou objet en mouvement dans le système. Le statut des entités change au cours du temps selon les états par lesquels elles passent. En général, plusieurs flux d'entités sont considérés dans un même système.

Définition 3.3.1 (Entité) *Une entité est un singleton $u = (A)$ où $A = \{a_1, \ldots, a_n\}$, $n \in \mathbb{N}$, est une liste d'attributs d'une même table caractérisant la classe.*

Définition 3.3.2 (Instance d'entité) *Une instance d'entité est un objet circulant dans une machine d'état de la vue processus.*

Les instances d'entités découlent d'une même classe et diffèrent par leurs attributs. Cependant leur comportement est similaire. Les valeurs des attributs qui caractérisent une instance d'entité sont définies dans une table du sous-système informationnel qui possède un certain nombre de champs permettant de décrire les caractéristiques de cette entité. Dans la suite de ce mémoire, nous utiliserons abusivement la dénomination entité pour désigner indifféremment sa classe ou l'une de ses instances.

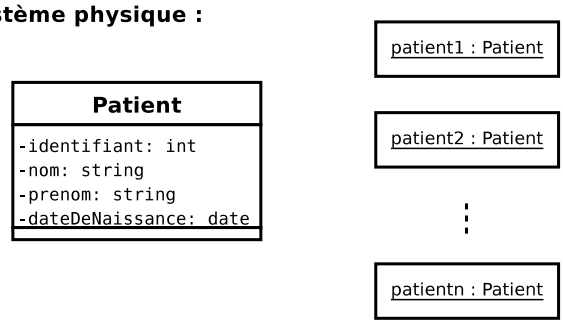
Exemple 3.3.1 *L'exemple présenté figure 3.1 décrit une classe* `Patient` *associée à une table comportant plusieurs attributs :* `identifiant`, `nom`, `prenom` *et* `dateDeNaissance`. *Chaque instance de la classe* `Patient` *(*`patient1`, `patient2`, *etc.) correspond à un enregistrement de la table associée.*

3.3.2 Machine d'état

Les diagrammes de machine d'état permettent de décrire le comportement d'un système au sens large. Les machines d'état fournissent une excellente façon de modéliser le parcours d'une entité, l'activité d'une ressource ou encore la coordination entre plusieurs partis. Les diagrammes produits sont simples à comprendre et intuitifs, favorisant l'échange sur la validité des modèles proposés. UML propose deux types de machines d'état (Pilone et Pitman, 2006) : les machines d'état *comportementales*, qui permettent la description de comportements d'éléments d'un système, et les machines d'état à *protocole*, qui décrivent le fonctionnement d'un protocole. Nous ferons exclusivement référence aux machines d'état comportementales dans la suite.

La modélisation du comportement d'entités et de ressources est réalisée au moyen d'*états* et de *transitions*. Des transitions entre les états se produisent lorsque des *événements* sont envoyés. Lorsqu'une

Côté sous-système physique :



Côté sous-système informationnel :

Attributs A			
identifiant	nom	prenom	dateDeNaissance
1	Duss	Jean-Claude	16/04/1952
2	Leblanc	Juste	08/12/1947
3	Perrin	François	16/08/1934
...			

FIG. 3.1 – Exemple d'une classe Patient

machine d'état s'exécute, les activités sont déclenchées sur la base de transitions, d'entrées dans un état, etc. Enfin, une *entrée* et une *sortie* sont nécessaire pour alimenter une machine d'état.

Définition 3.3.3 (Machine d'état) *Une machine d'état (MdE) est un triplet $M = (E, V, \zeta)$ tel que :*
- *$E = \{e_1, e_2, \ldots, e_n\}$ est un ensemble fini d'états ;*
- *$V \subseteq (E \times E)$ est l'ensemble fini des transitions ;*
- *$\zeta : V \to \mathbb{E}$ est l'application qui associe à une transition de V sa condition de garde.*

$\mathbb{E}$ est l'ensemble des conditions de garde associées aux transitions d'une machine d'état. Cet ensemble sera défini dans la section 3.3.4. Nous définissions également le concept de sous-machine d'état.

Définition 3.3.4 (Sous-machine d'état) *Soit $M = (E, V, \zeta)$ une machine d'état. Considérons deux sous-ensembles $E' \subset E$ et $V' \subset V$ ainsi que la fonction $\zeta' : V' \to \mathbb{E}$. $M' = (E', V', \zeta')$ est une sous-machine d'état de M (SMdE) si pour toute transition de V' ses extrémités sont dans E'.*

Définition 3.3.5 (Sous-machine d'état ouverte) *Soit $M = (E, V, \zeta)$ une machine d'état. Considérons deux sous-ensembles $E' \subset E$ et $V' \subset V$ ainsi que la fonction $\zeta' : V' \to \mathbb{E}$. $M' = (E', A', \zeta')$ est une sous-machine d'état ouverte de M si pour toute transition V' au moins une de ses extrémités est dans E'. Toute transition (e, f) telle que $e \in E'$ et $f \notin E'$ (resp. $e \notin E'$ et $f \in E'$) est appelée sortie (resp. entrée) de la sous-machine d'état ouverte E'.*

Afin de définir le concept de graphe associé à une machine d'état, nous rappelons brièvement la définition d'un graphe.

Définition 3.3.6 (Graphe) *Un graphe orienté est un couple $G = (X, U)$ caractérisé par :*
- *un ensemble de sommets $X = \{1, 2, \ldots, n\}$, $n \in \mathbb{N}^*$; G est dit d'ordre n ;*
- *un ensemble de couples de sommets ordonnés (arcs) noté U ; $(i, j) \in U$ est un arc ($i \in X$, $j \in X$), i est l'extrémité initiale et j l'extrémité terminale.*

Définition 3.3.7 (Graphe associé à une machine d'état) *Soit $M = (E, V, \zeta)$ une machine d'état. Le graphe $G = (X, U)$ associé à la machine d'état M est défini de la manière suivante :*

- *chaque état de E correspond à un unique sommet de X : $\forall e \in E$, $\exists! x \in X$ et $|X| = |E|$;*
- *chaque transition de V correspond à un arc de U : $\forall (e, f) \in V$, $\exists (i, j) \in U$ tel que e correspond à i et f correspond à j, et $|U| = |V|$.*

Considérer le graphe associé à une machine d'état ou à une sous-machine d'état permet d'extraire uniquement les informations structurelles d'un diagramme d'état et de s'affranchir des caractéristiques des états et des transitions conditionnelles.

3.3.3 Ressource

Une ressource est une classe permettant de représenter médecins, infirmiers, salles opératoires, ou tout autre acteur ou objet symbolisant une ressource humaine ou matérielle. Tout comme les entités, le statut des ressources change au cours du temps selon les tâches qui leurs sont assignées. Chaque ressource se voit attribuer un certain nombre de missions.

Définition 3.3.8 (Ressource) *Une ressource est un quintuplet $r = (A, EDT, MI, \zeta, eff)$ où :*

- *$A = \{a_1, \ldots, a_n\}$, $n \in \mathbb{N}^*$ est une liste d'attributs d'une même table caractérisant la ressource sous forme de classe ;*
- *$\text{EDT} = \{(s_1, c_1), \ldots, (s_m, c_m)\}$, $m \in \mathbb{N}$ est une liste de plages de disponibilité de la ressource (ou emploi du temps), avec $s_j < c_j$ $\forall j \in \{1, \ldots, m\}$;*
- *$MI = \{mi_1, \ldots, mi_p\}$, $p \in \mathbb{N}^*$ est un ensemble de missions affectées à la ressource ;*
- *$\zeta : MI \to \mathbb{E}$ est l'application qui associe à une mission de MI sa condition d'exécution ;*
- *$eff \in \mathbb{N}^*$ est l'effectif de la ressource, i.e. le nombre d'instances de cette ressource.*

Définition 3.3.9 (Mission) *Une mission est un quadruplet $mi = (M, e_{in}, e_{out}, eff)$ tel que :*

- *$M = (E, V, \zeta)$ est la machine d'état modélisant le déroulement de la mission ;*
- *$e_{in} \in E$ (resp. $e_{out} \in E$) est l'état d'entrée (resp. de sortie) de la mission ;*
- *$eff \in \mathbb{N}^*$ est l'effectif requis pour effectuer cette mission.*

Définition 3.3.10 (Instance de ressource) *Une instance de ressource est un objet circulant dans une machine d'état de la vue ressource.*

À la différence des classes d'entité, chaque ressource est associée à un état de base et à un ensemble de missions qui peuvent être accomplies par cette ressource. Ces missions sont décrites dans la vue ressource grâce à des machines d'état distinctes. Toutes les instances d'une ressource circulent dans ces machines d'état.

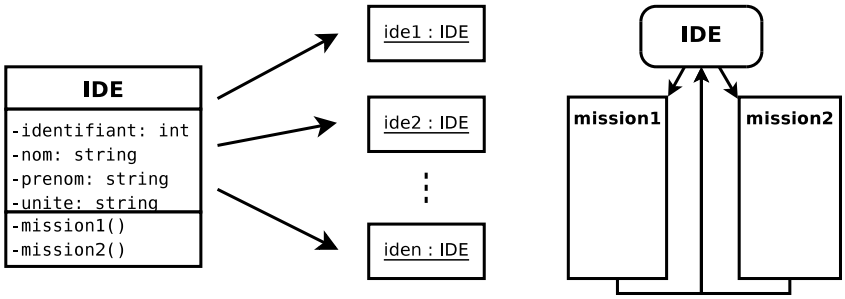
Une ressource ne se limite pas à un acteur du système ou à un objet unique. Plusieurs types de ressources distincts peuvent également être définis, permettant la représentation d'équipes par exemple. Les différentes classes de ressources sont décrites en détails dans la section 3.6.

Exemple 3.3.2 *La figure 3.2 présente un exemple de classe de ressource* `IDE` *associée à :*

- *une table comportant plusieurs attributs (*`identifiant`, `nom`, `prenom`, `unite`*) ;*
- *un emploi du temps ;*
- *un ensemble de machines d'état décrivant ses missions (son comportement).*

Les instances `ide1`, `ide2`, *etc. de la ressource* `IDE` *circulent dans les machines d'état associées à la classe de la ressource instanciatrice. Ces instances de ressources sont générées et détruites selon l'emploi du temps de la ressource également présenté figure 3.2.*

Côté sous-système physique :



Côté sous-système informationnel :

Attributs A			
identifiant	nom	prenom	unite
0	Morin	Nathalie	Neurologie
1	Rocourt	Christine	Neurologie
...			

Emploi du temps EDT	
s_i	d_i
8h00	10h20
10h30	14h00
...	

FIG. 3.2 – Exemple d'une classe IDE

3.3.4 Définitions complémentaires

Variable

Une variable associe un nom à une valeur ou à un objet. Le contenu d'une variable est commun et accessible à tous les objets en interaction dans le système. Une variable définie dans medPRO n'est pas typée et peut contenir un entier, un réel, un objet ou encore une chaîne de caractère. Tout comme dans la plupart des langages de programmation, il est également possible de créer des vecteurs (tableaux à une dimension) ou des matrices (tableaux à deux dimensions).

Étiquettes

Les étiquettes permettent d'identifier de manière unique les états dans l'ensemble du modèle UML. Une étiquette est un nom choisi par l'utilisateur. Nous appelons $\mathbb{L}$ l'ensemble des étiquettes.

Transition conditionnelle

L'ensemble $\mathbb{E}$ regroupe les expressions admissible pour définir la condition de garde d'une transition. La table 3.1 présente les différents types de conditions admissibles pour les transitions conditionnelle d'une machine d'état quelconque.

Transition aléatoire

Une probabilité peut être affectée à une transition conditionnelle. Ces transitions aléatoires sont toujours évaluées en dernier. La probabilité est exprimée sous la forme d'un réel compris entre 0 et 1 qui sera comparée avec la variable aléatoire $X \sim U(0,1)$.

Test sur la valeur d'un attribut ou d'une variable	
Description	La valeur de l'attribut d'une instance de classe est accessible grâce à l'opérateur point, permettant ainsi le test sur sa valeur. La valeur d'une variable est immédiatement accessible. Les opérateurs admissibles sont $=$, $>$, $<$, $\geq$, $\leq$ et $\neq$.
Syntaxe	`entite.attribut`<operateur>`valeur`
Exemples	• `patient.identifiant`=`123` • `patient.age`>`50` `patient` désigne une classe d'entité tandis que `identifiant` et `age` sont des attributs de cette classe. • `chirurgien.specialite`=`"neuro-vasculaire"` `chirurgien` désigne une classe de ressource et `specialite` est un attribut de cette classe.

Test sur la valeur de l'horloge globale	
Description	La valeur de l'horloge de simulation est accessible grâce à la variable t_{now} : il est donc possible de comparer la valeur de l'horloge avec une constante, un attribut ou une variable. Nous definissons en outre les fonctions `minute`, `heure`, `jour`, etc. permettant de convertir la valeur de t_{now} en une unité de temps classique.
Syntaxe	t_{now} <operateur>`valeur`
Exemples	• `jour`(t_{now})=`"jeudi"` La condition est vraie lorsque le jour correspondant à l'horloge de simulation est jeudi. • `patient.dateEntree`> t_{now} L'attribut `dateEntree` de l'entité `patient` est comparée à la valeur de l'horloge de simulation.

Test sur l'état de base d'une ressource	
Description	Il est possible de scruter l'état de base d'une ressource pour déterminer sa disponibilité par l'intermédiaire de la fonction `estLibre`.
Syntaxe	`estLibre`(`classe`)
Exemples	• `estLibre`(`brancardier`) Cette condition devient vraie au moment où au moins un `brancardier` est libre (i.e. se trouve dans son état de base).

Test spécifique lié à un module médical	
Description	Un certain nombre de tests spécifiques lié au milieu médical sont implémentés sous forme de module : il s'agit de fonctions permettant de connaître le délai de réception d'un examen ou encore la durée indicative d'une intervention chirurgicale particulière. L'ensemble de ces modules et fonctions seront décrites plus avant dans la section 5.4.

TAB. 3.1 – Expressions admissibles pour une transition conditionnelle d'une machine d'état

3.4 Vue processus

La vue processus regroupe les diagrammes de flux représentant le parcours d'une ou plusieurs entités (généralement des patients) à travers un système hospitalier. Ces processus sont représentés sous la forme de machines d'état.

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

La modélisation du parcours d'un patient (ou plus généralement d'une entité) est réalisée au moyen d'*états* et de *transitions*. La combinaison de ces éléments permet la création de diagrammes d'état, de manière identique à n'importe quelle ressource. Une *entrée* et une *sortie* sont nécessaires pour modéliser les flux entrant et sortant d'entités dans une machine d'état. Nous associons une machine d'état par classe d'entité considérée. Nous détaillons dans cette section les éléments spécifiquement sélectionnés pour cette vue. Soit $M = (E, V, \zeta)$ une machine d'état appartenant à la vue processus.

3.4.1 Entrée/Sortie

Les entrées et sorties sont des points de connexion permettant de définir l'interface externe de la machine d'état. Une machine d'état modélisant le processus d'une entité possède au moins une entrée et une sortie.

Définition 3.4.1 (Entrée) *Une entrée d'une machine d'état appartenant à la vue processus est un couple $e_{in} = (u, S)$ tel que $e_{in} \in E$ et $\forall (e, f) \in V \ e_{in} \neq f$. u est le type (la classe) d'entités générées et S le schéma de génération d'entités.*

Selon le formalisme UML, une entrée est représentée par un disque. Une entrée ne possède pas de point d'entrée et possède un unique point de sortie. Une entrée permet de générer les instances d'une classe d'entité définie dans la vue organisation. Le schéma de génération S des entités doit être connu à l'avance : loi de probabilité ou planning pré-défini dans le sous-système informationnel. Durant la simulation, un compteur associé à chaque entrée permet de connaître le nombre d'entités générées à tout moment.

Définition 3.4.2 (Schéma de génération) *Un schéma de génération $S : \mathbb{N} \rightarrow \mathbb{N}$ est l'application permettant de connaître la date de création d'une entité à partir de la précédente. Plusieurs schémas de générations peuvent être adoptés selon les besoins. Soit u_1 la première entité générée à la date s_1. La table 3.2 regroupe l'ensemble des schémas de génération admissibles pour une entrée, permettant de calculer la date de création s_i de l'entité u_i, $i \in \mathbb{N}^*$.*

Définition 3.4.3 (Sortie) *Une sortie d'une machine d'état appartenent à la vue processus est un singleton $e_{out} = \{u\}$ tel que $e_{out} \in E$ et $\forall (e, f) \in V \ e_{out} \neq e$. u est le type (la classe) d'entités détruites par cet état.*

Une sortie permet la destruction de toutes les entités passant par ce point. Représentée par un disque entouré, une sortie ne possède pas de point de sortie et possède au moins un point d'entrée. Aucun paramétrage n'est requis.

3.4.2 États

Définition

Un état de la vue processus (P-état) modélise une opération réalisée par ou sur le type d'entité considéré dans la machine d'état. Afin d'être réalisée, cette opération nécessite éventuellement une ou

Schéma	Description	Obtention de s_i
Planifié	Les dates d'arrivée sont connues pour l'ensemble des entités générées dans cette entrée. Soit $\pi : U \to \mathbb{N}$ l'application qui associe à toute entité de U sa date de création.	$s_i = \pi(e_i)$.
Régulier	Les entités sont générées à intervalle régulier $t_{inter} \in \mathbb{N}^*$.	$s_{i+1} = s_i + t_{inter}$ avec s_0 fixé.
Aléatoire	Les temps inter-arrivées des entités sont générés suivant une loi exponentielle de fonction de distribution : $F(x) = \begin{cases} 1 - e^{-\frac{x}{\beta}} & si\ x \geq 0 \\ 0 & sinon \end{cases}$ La loi exponentielle a pour paramètre $\beta > 0$.	$s_{i+1} = s_i - \beta \ln(U)$ avec $U \sim U(0,1)$.

TAB. 3.2 – Schémas de génération admissibles pour une entrée

plusieurs ressources. Un état peut représenter une situation statique, telle que « attendre le brancardier », ou une situation dynamique, au cours de laquelle l'état est actif, telle que « déjeuner ». Dans ce dernier cas, l'état possède une *durée*. Un état est soit *actif*, soit *inactif*. Un état est considéré comme actif dès lors qu'il est déclenché suite à une transition. De la même manière, un état est considéré comme inactif dès que l'activité est terminée ou dès que l'on en sort.

Définition 3.4.4 (P-état) *Un P-état est un triplet $pe = (l, \tau, R)$ tel que :*

- *$l \in \mathbb{L}$ est l'étiquette associée à l'état ;*
- *τ modélise la durée d'exécution de la tâche au moyen d'une constante, d'une variable aléatoire ou d'une fonction définie par l'utilisateur ;*
- *$R = \{(r_1, eff_1), \ldots, (r_n, eff_n)\}$ est une liste conjonctive de ressources requise pour l'exécution de la tâche associées à l'effectif requis eff_i, $i \in \{1, \ldots, n\}$.*

Exemple 3.4.1

La figure 3.3 présente un exemple d'état appartenant à la vue processus. Dans cet exemple, l'intervention chirurgicale modélisée par un état `interventionChirurgicale`. La tâche associée à cet état requiert une `salleOperatoire`, un `chirurgien` et deux `ide`. Cette tâche possède une durée fixe de 50 minutes. Lorsqu'un patient entre dans cet état, les ressources nécessaires sont réunies ; si l'une des ressources vient à manquer, le patient est mis en attente jusqu'à ce que toutes les ressources deviennent disponibles.

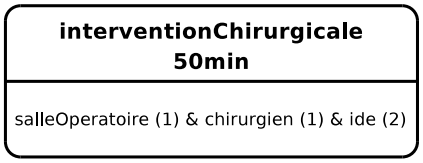


FIG. 3.3 – Un état modélisant une intervention chirurgicale

Formellement, cet état est défini par le triplet $pe = (l, \tau, R)$ avec $l =$`interventionChirurgicale`, $\tau = 3000$ et $R = (\{(r_1, 1), (r_2, 1), (r_3, 2)\})$, r_1, r_2 et r_3 représentant les ressources `salleOperatoire`, `chirurgien` et `ide` respectivement. Notez que les durées opératoires sont systématiquement converties en secondes.

Spécification des ressources d'un P-état

Lors de la création d'un nouvel état impliquant un certain nombre de ressources, l'utilisateur a le choix entre plusieurs alternatives pour la spécification de l'ensemble R :

Sélection nominative : l'utilisateur choisit les ressources nécessaires en précisant explicitement leur type (leur classe). C'est le cas dans l'exemple 3.4.1, où les classes de ressources (`salle_operatoire`, `chirurgien`, etc.) sont précisés.

Sélection par compétence : au lieu de préciser une ressource particulière, il est possible d'indiquer simplement une compétence requise partagée par plusieurs ressources pour l'exécution de la tâche liée à l'état ; dans ce cas le sous-système de pilotage est invoqué lors de la simulation pour sélectionner la ressource adéquate.

Sélection par équipe : l'utilisateur choisit un ensemble de ressources regroupées en équipe sous la forme d'une classe ; ici encore, le sous-système de pilotage est chargé de regrouper les membres de l'équipe lors de la simulation et de gérer les aléas éventuels dûs aux absences.

Le processus de décision mis en œuvre pour décider de la meilleure combinaison de ressources à retenir sera décrit plus avant dans le chapitre 5. L'organisation de ressources en équipes et/ou par compétence sera détaillée dans la section 3.6.

Exécution de la tâche liée à un P-état

L'exécution de l'action liée à un P-état se décompose en plusieurs phases distinctes gérées par le sous-système de pilotage :

1. L'entité entre dans la file d'attente du P-état.
2. Le sous-système de pilotage tente de sélectionner les ressources nécessaires.
3. Si une combinaison de ressources disponibles a pu être trouvée, la tâche du P-état est exécutée ; sinon l'entité est mise en attente jusqu'à ce que les ressources nécessaires deviennent libres.

Lorsque la tâche liée au P-état est terminée, l'entité est dirigée vers le prochain état au travers d'une transition conditionnelle.

3.4.3 Transitions conditionnelles

Une transition conditionnelle représente une relation, ou chemin, entre deux états. Elle symbolise le changement effectif dans la configuration d'une machine d'état, lorsque celle-ci bascule d'un état à l'autre. Chaque transition doit posséder une condition de garde précisant si cette transition peut être effectuée (*activée*). Les transitions sont indiquées sous la forme d'une ligne de texte située entre deux états, un flèche pointant de l'état initial vers l'état final.

Si aucune transition ne peut être activée à la sortie d'un état, l'entité est mise en attente jusqu'à ce que l'une des conditions de sortie devienne vraie ; on dit dans ce cas que l'état devient inactif. Si plusieurs conditions sont vraies au même moment, c'est la première sortie valide qui sera choisie, selon un ordre de priorité défini par l'utilisateur.

Les transitions sont définies et utilisées dans les vues processus et ressource de manière identique. Se reporter à la section 3.3.4 pour un descriptif complet des conditions de garde admissibles.

3.5 Vue ressource

La vue ressource regroupe les diagrammes d'état associés aux missions définies pour chacune des ressources déclarées dans la vue organisation. La modélisation de missions sous forme de machines d'état

permet une représentation précise du comportement de chaque ressource et offre une meilleure visualisation de l'activité du personnel hospitalier et/ou de l'utilisation de matériels médicaux. Une telle représentation permet en outre la modélisation d'activités invisibles aux yeux du patient. Enfin, une meilleure gestion de la disponibilité des ressources est possible grâce à cette représentation : l'utilisateur peut détailler le comportement d'une ressource et préciser explicitement où et comment celle-ci intervient au sein du parcours patient.

Ces machines d'état sont construites selon le formalisme UML ; plusieurs modifications ont été apportées afin de simplifier l'utilisation et la vérification de ces machines. De la même manière que pour la vue processus, nous détaillons dans cette section l'ensemble des briques de bases utilisées pour la modélisation de l'activité de ressources. Nous associons à toute ressource r : (i) un *état de base*, permettant de gérer la disponibilité des instances de la ressource r, et (ii) un ensemble de machines d'état distinctes modélisant les différentes *missions* de la ressource r.

L'utilisateur n'est pas tenu de définir des missions pour chacune des ressources déclarées. Une ressource sans mission possède uniquement un état de base. Les dates de disponibilités ainsi que les attributs définis sont cependant pris en compte et restent accessibles.

Dans la suite de cette section, nous détaillerons les deux principales composantes du diagramme d'état d'une classe de ressource, à savoir l'état de base et les missions.

3.5.1 État de base

Comme cela a été dit dans l'introduction, l'état de base permet la modélisation d'un état d'attente par défaut (*standby* en anglais) ou de tâches annexes interruptibles à tout moment.

Définition 3.5.1 (État de base) *L'état de base de la classe de ressource r est un état possédant autant de transitions d'entrée et de sortie que la ressource possède de missions.*

Remarque 3.5.1 *Une ressource sans mission ne possède qu'un état de base permettant uniquement de connaître la disponibilité de cette ressource.*

3.5.2 Missions

Définition

Une mission $mi = (M, e_{in}, e_{out}, eff)$ est une machine d'état dont les états d'entrée et de sortie e_{in} et e_{out} sont connectés à l'état de base de la ressource considérée. Une mission permet de modéliser l'activité d'une ressource sous forme d'état. Certains états d'une mission peuvent être synchronisés avec certains états de la vue processus, permettant de représenter l'accompagnement du patient pour une certaine durée et pour un certain nombre de tâches par une ressource. Une ressource peut se voir affecter une ou plusieurs missions reflétant son implication dans le processus de prise en charge du patient. Cependant, le nombre d'instances de ressources nécessaires pour chacune de ces missions (eff) est constant.

La machine d'état modélisant une mission est ainsi composée d'états *synchronisés* avec le parcours patient et d'états *transitoires* représentant les tâches particulières réalisées par la ressource et qui n'apparaissent pas du point de vue patient.

États d'une mission

Les états de la vue ressource (R-état) modélisent un moment spécifique du comportement de la ressource. Plus précisément, ces états modélisent une situation pour laquelle une *condition invariante* est satisfaite par la ressource dont on décrit le comportement. Les statuts statiques et dynamiques ainsi que les états actif et inactif décrits dans la vue processus sont également valables dans cette vue.

Définition 3.5.2 (R-état) *Un R-état est un couple $re = (l, \tau)$ tel que $l \in \mathbb{L}$ est l'étiquette associée à l'état et τ modélise la durée d'exécution de la tâche au moyen d'une constante, d'une variable aléatoire ou d'une fonction définie par l'utilisateur.*

Exemple 3.5.1

La figure 3.4 présente un R-état simple dont l'unique activité `anesthesier` dure 15 minutes, défini par $re = (l, \tau)$ avec $l =$`anesthesier` et $\tau = 900$.

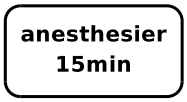


FIG. 3.4 – Un état simple

Synchronisation avec la vue processus

Considérons une mission $mi = (M, e_{in}, e_{out}, eff)$. Un état ou un sous-ensemble d'état de cette mission peut-être *synchronisé* avec la vue processus afin de modéliser l'allocation et la libération de la ressource durant le parcours patient. Il s'agit d'un « rendez-vous » pris entre le patient et la ressource pour la réalisation d'une ou plusieurs tâche(s) en commun. Les états synchronisés possède la même étiquette.

Exemple 3.5.2

La figure 3.5 présente un exemple de machine d'état pour une infirmière. L'état `Infirmiere` est l'état de base de la ressource : les instances de la ressource se trouvent par défaut dans l'état de base. Nous avons représenté dans cet exemple deux missions : la première est estampillée *Prise de sang* et modélise l'action consistant à réaliser une prise de sang sur un patient ; la deuxième est appelée *Accueil* et modélise l'accueil et l'installation d'un nouveau patient dans sa chambre. Les états grisés forment une sous-machine d'état synchronisée avec la vue processus. Plusieurs états annexes apparaissent également en blanc, permettant la modélisation de tâches de transitions : préparation, nettoyage, etc.

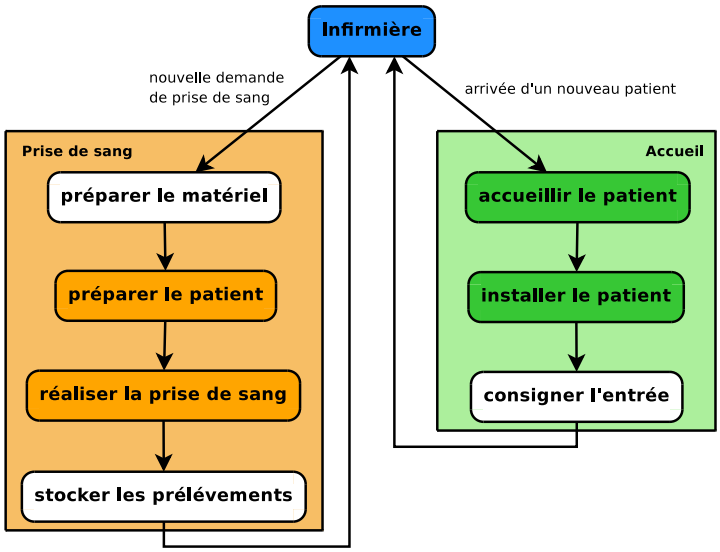


FIG. 3.5 – Un exemple de machine d'état pour une ressource de type infirmière

Ainsi, une sous-machine d'état de la vue ressource peut être liée à une sous-machine d'état de la vue processus. Cette synchronisation permet de modéliser le fait qu'une entité et une ressource prennent un

rendez-vous pour la réalisation d'une ou plusieurs tâches regroupées sous forme de missions, telle la prise de sang ou l'accueil dans l'exemple présenté figure 3.5. Les définitions suivantes permettent de formaliser le concept de mission et de synchronisation.

Définition 3.5.3 (Synchronisation d'état) *Soient $e = (l, \tau, R)$ et $f = (l', \tau')$ deux états des vues processus et ressource respectivement. e et f sont dits synchronisés si $l = l'$.*

Définition 3.5.4 (Synchronisation de sous-machines d'état) *Soient $M = (E, V, \zeta)$ une sous-machine d'état ouverte de la vue processus et $M' = (E', V', \zeta')$ une sous machine d'état ouverte de la vue ressource. M et M' sont dits synchronisés si :*

- *les graphes associés à M et M' sont identiques ;*
- *les états de E sont synchronisés deux à deux avec les états de E' ;*
- *le même nombre d'unités de la ressource est requis dans tous les états de E.*

L'exécution des états de sous-machines d'état synchronisées est assujettie à la présence de l'entité côté vue processus et à la présence de la ressource côté vue ressource en entrée de la sous-machine d'état. Nous traduisons cette contrainte par le fait qu'une ressource nécessaire à l'accomplissement d'une mission doit être présente au même moment que l'entité. L'exécution est alors réalisée de manière simultanée dans les deux vues, entité et ressource étant synchronisées avant le début de chacun des états des machines d'état synchronisées. Une fois la synchronisation réalisée, entité et ressource restent liées jusqu'à la fin du parcours de la sous-machine d'état (i.e. jusqu'à ce que l'une des sortie de la sous-machine soit atteinte).

Exemple 3.5.3

La figure 3.6 présente deux exemples de synchronisation entre la vue processus et la vue ressource. Dans l'exemple de gauche, trois états sont synchronisés au sein d'une sous-machine d'état qui comporte une entrée et deux sorties ; une sous-machine d'état identique apparaît dans la vue ressource de `R1`. Dans l'exemple de droite, seuls deux états sont synchronisés, la sous-machine d'état n'est plus la même mais comporte toujours une entrée et **deux** sorties ; la sous-machine d'état de la vue ressource fait bien apparaître les deux états synchronisés et comporte une entrée vers `Etat1`, une sortie depuis `Etat1` et une sortie depuis `Etat2`.

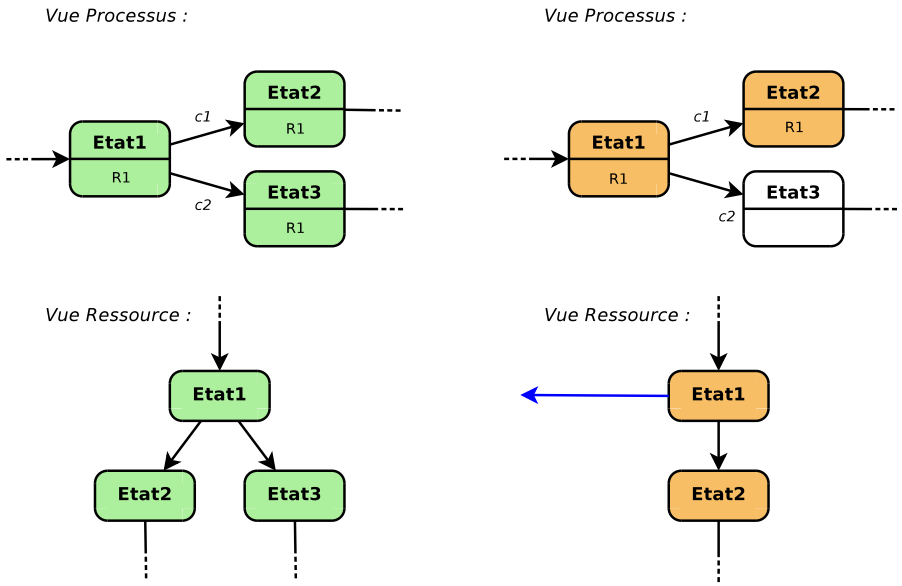


FIG. 3.6 – Deux exemples de synchronisation

Cet exemple permet d'insister sur les formes des sous-machines d'état qui doivent être rigoureusement identiquent d'un point de vue structurel afin d'éviter tout blocage lors de la simulation. En particulier les sorties conditionnelles au cours d'une mission de doivent pas être omises.

Exemple 3.5.4

La figure 3.7 présente un exemple de synchronisation de sous-machine d'état. Dans cet exemple, afin de conserver une cohérence de lieu, la sous-machine d'état liée à l'utilisation de la salle opératoire (`SO`) est reportée dans les vues processus et ressource : les états `anesthesier`, `interventionA` et `interventionB` constitue cette sous-machine.

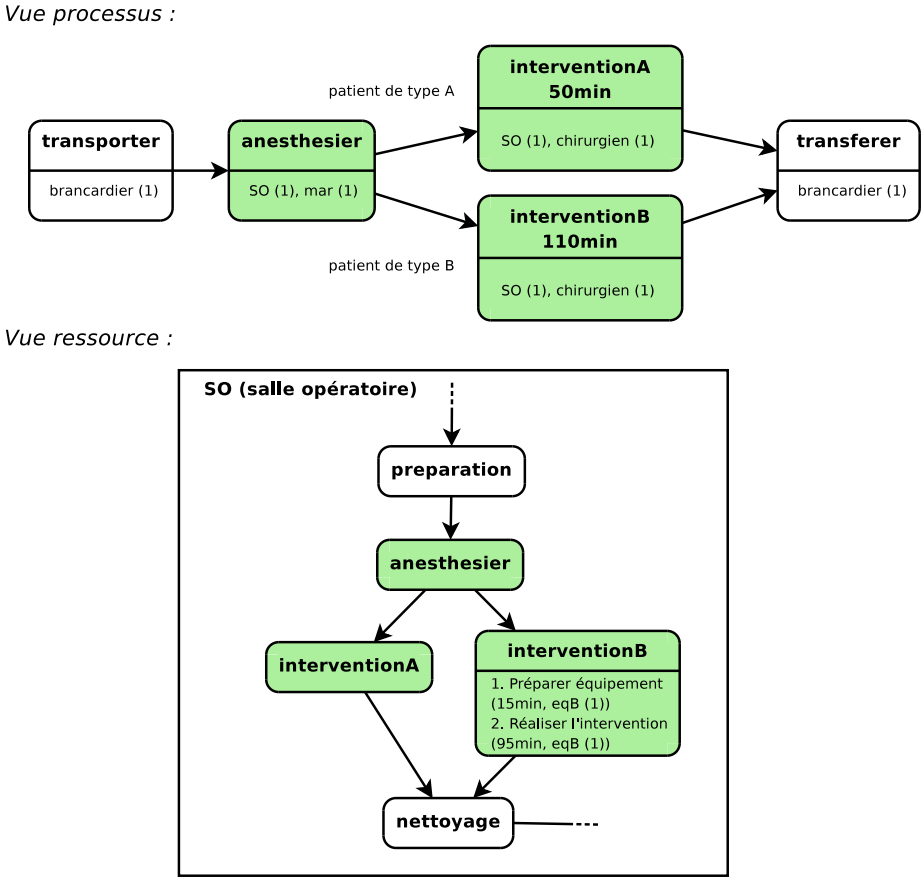


FIG. 3.7 – Un exemple de synchronisation de sous-machines d'état

Le processus présenté dans l'exemple 3.7 peut être décrit comme suit : après le transport du patient par un `brancardier`, l'anesthésie commence lorsqu'une salle opératoire (`SO`) et un médecin anesthésiste réanimateurs (`MAR`) sont libres. Le MAR est considéré comme une ressource sans MdE à deux états : son allocation est réalisée dès qu'un jeton est libre. En revanche, la SO sera considérée comme libre lorsque sa préparation sera terminée dans la vue ressource. L'anesthésie peut alors commencer simultanément dans les vues processus et ressource. Selon le type de patient, une intervention de type A ou B est réalisée par un `chirurgien` dans la même SO. S'il s'agit d'une intervention de type B, une ressource simple `eqB` est requise et une tâche de préparation préliminaire doit être réalisée. La durée d'exécution de la tâche associée à l'état `interventionB` est donc fonction des activités internes décrites dans la vue ressource. Enfin, le transfert du patient est réalisé à l'issu de l'intervention par un `brancardier`. La SO est libérée à ce moment et nettoyée.

3.5.3 Transitions conditionnelles

De la même manière que pour la vue processus, les transitions conditionnelles permettent de relier des états deux à deux dans la vue ressource. Se reporter à la section 3.3.4 pour un descriptif complet des conditions de garde admissibles.

3.6 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 intéractions entre les acteurs du systèmes, les ressources matérielles et les patients, permettant de comprendre le fonctionnement global du système modélisé.

Le terme « ressource » regroupe les ressources simples (individualité, matériel unique), les équipes et les compétences déclarées. Les diagrammes de classes sont utilisés pour la vue organisation. Ces derniers constituent les diagrammes les plus fondamentaux d'UML, et sont utilisés pour définir les relations statiques au sein d'un système ainsi que sa structure physique.

3.6.1 Déclarations de classes

Tous les objets qui interagissent dans le système représenté doivent être déclarés dans la vue organisation sous forme de classes. Comme cela a été dit en début de chapitre, une *classe* représente un groupe d'objets possédant des états et un comportement communs, et un *objet* est une *instance* d'une classe.

Une classe est représentée sous la forme d'un cadre rectangulaire divisé en trois compartiments qui contiennent : (i) le nom de la classe, (ii) ses attributs, et (iii) ses opérations. Le troisième compartiment réservé aux missions sera utilisé pour lister les missions que cette classe sera amenée à accomplir durant la simulation. L'insertion de commentaires durant la déclaration des classes est encouragée : ceux-ci permettrons une meilleure compréhension de la modélisation des classes mises en jeu ainsi qu'un allègement des vues processus et ressource.

Plusieurs classes isolées les unes des autres ne permettraient en rien de modéliser un système et les différentes interactions entre ses composants. UML propose un certain nombre de moyens pour représenter les relations entre classes. Chaque relation représente un type spécifique de lien entre les classes ; leurs subtilités respectives ne sont pas totalement décrites dans les spécifications d'UML. Par conséquent, le sens des relations décrites dans ce mémoire sera systématiquement précisé.

3.6.2 Spécialisation et remplacement

Les notions de spécialisation 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 relation de généralisation permet de modéliser cet aspect caractéristique des systèmes hospitaliers.

Une relation de généralisation (ou héritage) permet d'indiquer qu'une classe constitue un cas plus général, moins spécifique, d'une autre classe. Les relations de généralisation sont souvent utilisées pour extraire des propriétés communes à plusieurs classes différentes, afin de les regrouper au sein d'une entité plus générale. Par exemple, nous disposons de deux classes nommées `IDE` (Infirmière Diplômée d'État) et `AS` (Aide-Soignante) ; il est possible de créer une généralisation de ces deux classes, nommée `PersonnelSoignant`. Une relation de généralisation peut s'interpréter comme une relation de type « est un », le premier membre de celle-ci étant une classe spécifique, le second étant une classe plus générale : une `IDE` est un `PersonnelSoignant`. La classe spécifique est aussi appelée classe fille alors que la classe

plus générale est appelée classe mère. Une relation de généralisation est représentée au moyen d'une ligne continue, terminée par une tête de flèche pointant vers la classe plus générale.

Définition 3.6.1 (Relation de spécialisation) *Soient deux classes de ressources r_1 et r_2 telles que $r_1 = (A_1, EDT_1, MI_1, \zeta_1, eff_1)$ et $r_2 = (A_2, EDT_2, MI_2, \zeta_2, eff_2)$. On dit que r_1 hérite de r_2 si $A_2 \subseteq A_1$ et $MI_2 \subseteq MI_1$. On note $r_1 \triangleright r_2$.*

Exemple 3.6.1

La figure 3.8 montre un exemple de relation entre `IDE` et `AS`. On remarque que les attributs de la classe mère figurent implicitement dans la classe `IDE`. En revanche, l'attribut `qualification` n'existe que pour la classe `IDE`. Il en est de même pour les missions de chaque classe.

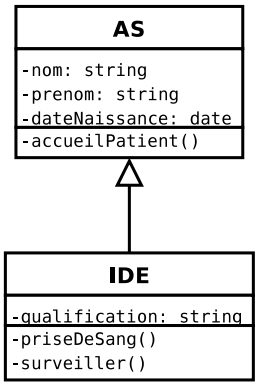


FIG. 3.8 – La classe *AS* est une généralisation de la classe *IDE*.

La relation représentée figure 3.8 s'écrit $r_1 \triangleright r_2$ où :

- $r_2 = (A_2, EDT_2, MI_2, \zeta_2, eff_2)$ modélise l'`AS` avec $A_2 = \{a_1, a_2, a_3\}$ ses attributs (`nom`, `prenom` et `dateNaissance`) et $MI_2 = \{mi_1\}$ sa mission (`accueilPatient`);
- $r_1 = (A_1, EDT_1, MI_1, \zeta_1, eff_1)$ modélise l'`IDE` avec $A_1 = A_2 \cup \{a_4\}$ ses attributs (`nom`, `prenom`, `dateNaissance` et `qualification`) et $MI_1 = MI_2 \cup \{mi_2, mi_3\}$ ses missions (`accueilPatient`, `priseDeSang`, `surveiller`).

D'un point de vue pratique, la généralisation procure à l'utilisateur plusieurs avantages en matière de modélisation, notamment en clarté et souplesse : la généralisation offre un gain de temps important lors de la déclaration des classes qui modélisent les entités et les agents du système. La déclaration d'une classe fille permet la récupération instantanée des attributs et opérations définis pour la classe mère. L'usage de relations de généralisation permet également de gagner en clarté.

Les répercussions d'un point de vue opérationnel d'une relation d'héritage sont visibles à plusieurs niveaux :

Comportement : une ressource spécialisée possède par défaut le même diagramme d'état (i.e. les même missions) que sa classe mère. Des missions supplémentaires peuvent par la suite être ajoutées pour adhérer à la spécialisation de la ressource considérée.

Exécution d'une tâche : une ressource requise pour l'exécution d'une tâche peut être remplacée par l'un de ses héritiers au moment de la réservation, si les conditions de simulation le permettent et si l'utilisateur l'autorise. Le choix de la ressource dépend alors des directives données au sous-système de pilotage. Par exemple, si l'entité `Patient` doit subir une intervention qui requiert l'intervention d'une `IDE`, le choix de la ressource se portera directement sur l'`IDE`. En revanche, si l'intervention ne requiert pas explicitement l'intervention d'une `IDE` mais d'une `AS`, le choix de la ressource se portera sur l'`AS` ou sur l'`IDE`.

Remarque 3.6.1 *Le mécanisme de généralisation est très utile pour limiter ou étendre les compétences des ressources humaines ou matérielles. Nous insistons sur la rapidité qu'offre cette pratique, car la modélisation du personnel médical d'un service de soins peut être déclinée sous forme de spécialisations : par exemple la classe* `Professeur` *hérite de la classe* `Docteur` *qui hérite de la classe* `Interne` *qui hérite de la classe « racine »* `Personnel_soignant`*.*

D'un point de vue allocation/libération de ressource, le sous-système de pilotage est implicitement autorisé à choisir toute classe de ressource fille de la classe de ressource spécifiée pour la réalisation d'une tâche liée à un P-état. Ainsi toute classe de ressource fille d'une classe de ressource requise pour une tâche est potentiellement réquisitionnable par le sous-système de pilotage, sauf si l'utilisateur précise le contraire.

3.6.3 Association en équipes

Le travail en équipe est une notion fondamentale en milieu hospitalier qui doit être modélisée avec soin afin de retranscrire le plus fidèlement possible ce mode de travail. Le bloc opératoire est l'exemple classique permettant d'illustrer le travail en équipe pour l'intervention sur un patient, ou pour la gestion générale du bloc (nettoyage, stérilisation, logistique). Il est donc nécessaire de définir une relation permettent de modéliser des liens entre plusieurs ressources afin de former des groupes.

La relation d'association constitue un type de dépendance entre classe plus fort que la simple dépendance ; elle indique qu'une classe est en relation avec une autre pendant un certain laps de temps. Dans ce cadre, une association peut s'interpréter comme une relation de type « fait partie de » et est représentée par une ligne simple.

Définition 3.6.2 (Équipe) *Une équipe est un triplet* $eq = (A, R, MI)$ *tel que :*

- $A = \{a_1, \ldots, a_n\}$ *est une liste d'attributs définie par l'utilisateur ;*
- $R = \{(r_1, eff_1), \ldots, (r_m, eff_m)\}$ *est l'ensemble des membres de l'équipe, désignés par un couple* (r_i, eff_i), $i \in \{1, \ldots, m\}$ *où* r_i *est une ressource et* $eff_i \in \mathbb{N}^*$ *l'effectif requis ;*
- $MI = \{mi_1, \ldots, mi_p\}$ *est l'ensemble des missions spécifiques à l'équipe déclarée.*

Définition 3.6.3 (Relation d'association par équipe) *Soit* $r = (A, EDT, MI, \zeta, eff)$ *une ressource et* $eq = (A', R, MI')$ *une classe d'équipe. On dit que* r *appartient à l'équipe* e *si* $\exists (r_i, eff_i) \in R$ *tel que* $r_i = r$ *et si* $MI' \subseteq MI$*. On note* $r \mapsto eq$*.*

Exemple 3.6.2

La figure 3.9 montre un exemple de relation entre les classes de ressource `Infirmiere`, `AideSoignante` et `Chirurgien`. La modélisation d'une équipe implique la création d'une classe virtuelle nommée `EquipeChir`. La relation entre les deux classes de chirurgien indique que la ressource `Chirurgien` peut être remplacée si elle est indisponible par la ressource appelée `ChirurgienSuppleant` (héritage/remplacement).

Soient r_1, r_2, r_3 et r_4 les ressources associées respectivement à l'infirmière, à l'aide-soignante, au chirurgien et au chirurgien suppléant. La classe d'équipe $eq = (A, R, f, MI)$ peut être définie ainsi :

- $A = \{a_1\}$ où a_1 correspondent à l'attribut `specialite` de l'équipe eq ;
- $R = \{(r_1, 1), (r_2, 1), (r_3, 1)\}$;
- $MI = \{mi\}$, mi étant la mission `intervention`.

La multiplicité permet d'indiquer l'effectif nécessaire pour une ressource membre de l'équipe. Si aucune valeur n'est spécifiée, la multiplicité a pour valeur implicite 1. Pour indiquer une valeur différente, il suffit de la mentionner à proximité de la classe concernée. La figure 3.10 montre l'exemple d'une `Équipe` constituée de 3 instances d'`Infirmiere`.

Tous les membres d'une équipe sont indispensables et doivent être présents pour la réalisation d'une activité par l'équipe.

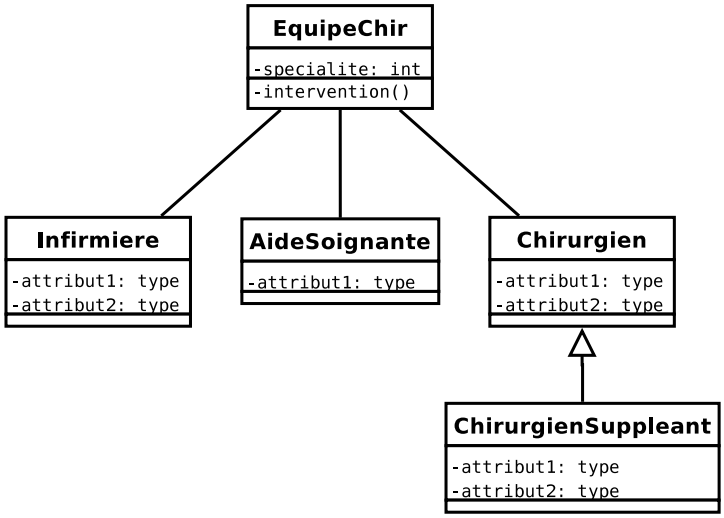


FIG. 3.9 – L'*Infirmière*, l'*AideSoignante* et le *Chirurgien1* forment une *EquipeChirurgicale*.

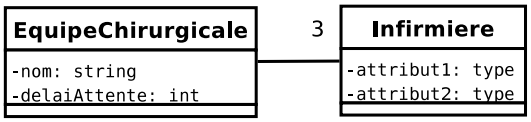


FIG. 3.10 – Équipe constituée de trois instances de la classe *Infirmiere*.

Modélisation de l'entente entre travailleurs : les membres d'une équipes habitués à travailler ensemble depuis plusieurs années auront tendance à accomplir des tâches plus rapidement et plus efficacement. Cette caractéristique est modélisée par l'intermédiaire des attributs de la classe virtuelle du groupe. Au contraire, si l'équipe doit être divisée une même tâche demandera plus de temps pour être exécutée.

Sélection de ressources : la constitution d'associations en groupes ou en équipes permet de simplifier la modélisation d'actions communes à plusieurs ressources ; il est possible d'associer une ou plusieurs mission(s) à la classe virtuelle correspondant à une équipe. Les membres héritent automatiquement des missions déclarées dans la classe de leur équipe.

3.6.4 Association par compétence

Outre son titre, chaque acteur dans le domaine hospitalier possède un certain nombre de compétences qui lui permettent d'exercer son métier dans un domaine ou un service particulier. Ces compétences peuvent être identifiées de manière objective (un préparateur en pharmacie doit être capable de réaliser une dotation pour un service de soins de la même manière que le dosage d'une chimiothérapie ou d'une poche) ou subjective (une infirmière exerçant depuis vingt ans dans un service ne pourra accomplir toutes les tâches relatives à un autre service, dans un premier temps du moins).

Cette dernière notion sera modélisée sous la forme d'une association entre classes. Ce type de relation indique que plusieurs classes (plusieurs ressources humaines dans ce cas) possèdent une compétence commune qui n'est pas forcément décrite par leurs spécialités respectives. Une telle association peut s'interpréter comme une relation de type « possède la compétence ». La représentation de l'association de plusieurs ressources par compétences est identique à la représentation en équipe.

Définition 3.6.4 (Classe de compétence) *Une classe de compétence est un triplet* $co = (A, R, MI)$ *tel que :*

- $A = \{a_1, \ldots, a_n\}$ *est une liste d'attributs définie par l'utilisateur ;*
- $R = \{r_1, \ldots, r_m\}$ *est l'ensemble des classes de ressources qui possède la compétence déclarée ;*
- $MI = \{mi_1, \ldots, mi_p\}$ *est l'ensemble des missions spécifiques à la compétence déclarée, représentées sous forme de sous-machines d'état.*

Définition 3.6.5 (Relation d'association par compétence) *Soit* $r = (A, EDT, MI, \zeta, eff)$ *une classe de ressource et* $q = (A', R, MI')$ *une classe de compétence. On dit que* r *possède la compétence* q *si* $r \in R$ *et si* $MI' \subseteq MI$. *On note* $r \succ q$.

Exemple 3.6.3

La figure 3.11 présente un exemple de classe de compétence appelée `InstallerHolter`, décrivant la pratique de l'installation d'un holter sur un patient. Cette classe possède une seule mission appelée `installerHolter`. Deux classes de ressources possèdent cette compétence : `Infirmiere` et `AideSoignante`.

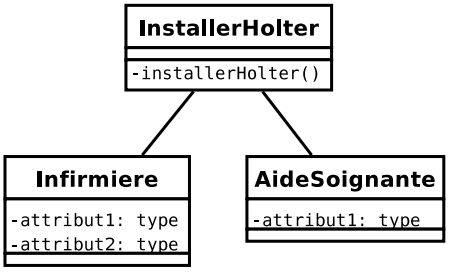


FIG. 3.11 – Les classes *Infirmiere* et *AideSoignante* possède la compétence *InstallerHolter*.

Si r_1 et r_2 désignent respectivement les classes de ressource de l'infirmière et de l'aide-soignante, la compétence $c = (A, R, MI)$ peut être définie comme suit :

- $A = \oslash$;
- $R = \{r_1, r_2\}$;
- $MI = \{mi\}$, mi étant la mission `installerHolter`.

Les ressource possédant une compétence héritent automatiquement des missions définies pour cette classe de compétence. La déclaration de compétences sous forme de classe permet leur utilisation dans les autres vues du sous-système physique. Une compétence déclarée dans la vue organisation peut être requise pour la réalisation d'une activité dans la vue processus. Dans ce cas, le choix de la ressource sera réalisé par le sous-système de pilotage en fonction de paramètres donnés par l'utilisateur : toute ressource possédant la compétence déclarée dans la vue processus peut être réquisitionnée, sauf si l'utilisateur précise le contraire.

3.7 Interactions entre vues processus, ressource et organisation

Décomposer la modélisation d'un système hospitalier en trois vues distinctes offre un certain nombre d'avantages en terme de d'intuitivité et de rapidité. Nous nous sommes efforcés d'argumenter l'existence des vues présentées dans les sections précédentes ; en effet, les avantages procurés par l'utilisation d'un modèle scindé en plusieurs vues sont multiples :

- favorisation de la communication ;
- représentation détaillée de l'activité des ressources humaines et matérielles ;
- séparation des concepts organisation et logistique ;
- spécification si nécessaire de la nature, du comportement et du relationnel entre ressources.

Nous allons tenter de détailler dans cette section ces différents points afin de mettre en valeur les concepts avancés dans ce chapitre et d'éclaircir certains points concernant les liens entre les vues, et à un niveau supérieur, entre les sous-systèmes.

3.7.1 Communication et impact visuel

L'approche systémique permet d'isoler des informations de natures très différentes et de les analyser séparément. Nous avons constaté que la collecte et l'organisation de données constitue à elle seule une étape souvent longue et fastidieuse : l'intégration de ces données au sein d'un système d'information dédié et générique permet une grande souplesse de travail tant au niveau de l'organisation que de la validation, facilitant par la même occasion la génération de rapports statistiques. De la même manière, la mise en œuvre d'un système de pilotage et de contrôle permet de séparer les algorithmes de traitement associés à une organisation particulière.

Le découpage du sous-système physique en trois vue favorise la communication avec les hommes et les femmes directement concernés par la modélisation du système dans lequel ils évoluent ou évolueront. La vue processus offre une représentation épurée du parcours patient uniquement, permettant la visualisation de son entrée dans le système, des étapes de son hospitalisation, de sa durée de séjour et de la mobilisation des ressources nécessaire pour son accueil. La vue ressource permet à chacun de visualiser ses activités et son implication dans le processus de prise en charge du patient. Enfin, la vue organisation permet une représentation précise de l'ensemble des acteurs en interaction, de leurs caractéristiques et de leurs relations. Cette structure a été choisie car elle met l'accent sur le dialogue personnalisé tout en offrant une représentation globale de l'activité du système, offrant la possibilité aux uns et aux autres de réagir sur la modélisation adoptée.

L'utilisation de diagrammes d'état simplifiés offre une représentation compréhensible par tous au même titre que de simples logigrammes. La formalisation mathématique détaillée dans ce chapitre permet en revanche une description détaillée du modèle réalisé grâce à un ensemble de contraintes simples.

3.7.2 Allocation et libération de ressources

Dans la plupart des outils de modélisation et/ou de simulation existant, les ressources sont communément représentées sous la forme de jetons. Un tel jeton possède deux états : libre ou occupé. Lorsqu'une ressource est nécessaire pour l'exécution d'une tâche, une requête est réalisée. Si la ressource est libre, elle est allouée et devient occupée. Si la ressource est occupée au moment de cette requête, l'entité à l'origine de la demande entre dans une file d'attente. Cette dernière sera libérée lorsque son tour viendra et que la ressource sera libérée. La ressource reste occupée jusqu'à la fin de la tâche, moment auquel celle-ci est libérée.

Ce mécanisme très simple permet de modéliser le phénomène d'allocation et de libération de ressources. Plusieurs améliorations existent, telle la prise en compte de temps de montage et/ou de démontage, de pannes, de périodes d'indisponibilité, etc. Cependant, un tel mécanisme possède plusieurs désavantages :

- Une mauvaise gestion des allocations peut mener à un verrou mortel, entraînant un blocage complet du système : dans ce cas, il est nécessaire d'énumérer les cas critiques et de les résoudre séparément.
- Les activités propres aux ressources sont difficilement modélisables : outre la prise en compte de temps de montage et/ou de démontage, les tâches liées aux ressources ne peuvent être modélisées. Dans ce cas, le recours à un modèle « dual » est préconisé, permettant la représentation du modèle du point de vue de la ressource considérée.
- Les notions de période de disponibilité restent limitées et ne permettent pas la prise en compte d'emploi du temps dynamique.

– La notion de panne est limitée et ne permet pas la modélisation d'un parcours hautement stochastique tel que le serait celui d'un médecin au sein d'un service d'urgence. Ici encore, le recours à une vue « dual » est la seule solution pour pallier à ce problème.

Nous proposons ainsi un mécanisme d'allocation/libération différent : une ressource n'est plus un simple jeton à deux états mais devient un objet avec un comportement prédéfini qui lui est propre, représenté sous la forme d'une machine d'état. L'allocation naît de la *synchronisation* entre deux états appartenant respectivement à la vue processus et à la vue ressource. L'application d'une telle méthode possède un grand nombre d'avantages, mais également quelques inconvénients :

+ Étant donné que chaque classe de ressource possède son propre schéma d'activité, la modélisation d'activités en marge du parcours des entités est maintenant possible sans ambigüité.
+ Attributs et disponibilités de ressources sont gérés par l'intermédiaire de tables modifiables dynamiquement, offrant une liberté d'action supplémentaire durant la simulation : une ressource n'est plus astreinte au même enchaînement d'activités au cours du temps.
+ En complément du point précédent, le comportement d'une ressource est entièrement configurable pour tenir compte d'évènements stochastiques dépendant du système modélisé.
+ L'utilisateur est toujours libre de revenir à un système d'allocation/libération classique en s'affranchissant de la représentation duale du comportement de la ressource considérée. Les remarques citées précédemment s'appliquent alors. Une telle décision peut s'avérer judicieuse pour la modélisation de systèmes impliquant un grand nombre de ressources élémentaires (matériel médical, outils).

– La probabilité de verrou mortel n'est pas nulle, et l'utilisateur doit toujours s'assurer de pallier aux situations de blocage, en proposant par exemple une sélection alternative de ressources pour l'accomplissement d'une tâche.

Lorsque deux états sont synchronisés dans deux vues différentes, un même nom (étiquette) leur est attribué. L'accomplissement de la tâche associée à cet état implique que l'entité et que toutes les ressources soient présentes. Il existe ainsi une probabilité pour que l'entité ou l'une des ressources soit mise en attente. Tous les états synchronisés dans le sous-système physique seront formalisés sous la forme d'une transition unique dans le réseau de Petri sous-jacent. Les détails de la conversion seront donnés dans le chapitre suivant.

Les règles d'allocation/libération de ressources peuvent être résumées comme suit. Nous considérons un P-état quelconque :

1. Sans synchronisation, une ressource est allouée au début d'une tâche et est libérée immédiatement lorsque celle-ci se termine, avant même le branchement vers l'état suivant.
2. Une ressource requise pour une tâche de la vue processus doit posséder un état ou une sous-machine d'état synchronisé(e). Dans ce cas, la tâche commence lorsque l'entité et la ressource requise sont prêtes. La synchronisation de sous-machines d'état permet de modéliser la conservation d'une ressource durant plusieurs tâches : les lignes de vie de l'entité et de la ressource sont alors liées pendant toute la durée de la synchronisation. Les machines d'état synchronisées sont rigoureusement identiques et ne possèdent exactement un état d'entrée et un état de sortie.

3.7.3 Organisation de ressources

Habituellement l'ensemble des ressources définies dans un modèle ne possède pas de liens et/ou ne forment pas d'organisation particulière. Pourtant la structuration de l'organisation du système offre un gain de temps non négligeable, offrant une plus grande souplesse en matière de modélisation :

– du point de vue processus, le patient est amené à passer par plusieurs états lors de son parcours qui requièrent l'intervention de ressources possédant certaines compétences. Au lieu de préciser explicitement le nom de chaque membre de l'équipe médical, il est plus commode d'indiquer uniquement les

compétences nécessaires ou les équipes mises en jeu, définies au préalable dans la vue organisation. Selon l'organisation du système, la sélection individuelle de ressources est déléguée au sous-système de pilotage. De la même manière, certaines contraintes organisationnelles impliquent l'utilisation conjointe d'un ensemble de ressources. L'utilisateur peut laisser le sous-système de pilotage gérer ces dépendances pour se concentrer sur la modélisation proprement dite.

– du point de vue organisation, les modifications apportées à ce niveau se répercutent de manière transparente sur le parcours patient : le sous-système de pilotage est alors requis pour adapter son comportement à la nouvelle organisation. Par exemple, l'ajout d'une ressource possédant une compétence requise au niveau processus sera automatiquement pris en compte lors de la simulation, cette ressource apparaissant comme un candidat pour l'exécution des tâches correspondantes.

Le sous-système de pilotage joue le rôle d'interface entre les vues processus et organisation, et offre à l'utilisateur une gestion optimale des ressources à sa disposition en fonction de contraintes définies lors de la modélisation du système. Ces mécanismes encapsulés sous forme de *modules de décision* seront décrits plus avant dans le chapitre 5.

Structurer les classes de ressource grâce à des associations offre un large éventail de subtilités pour la modélisation de comportements particuliers. La déclaration d'entités et de ressources sous forme de classes permet d'interagir dynamiquement avec les caractéristiques de chacune, jusqu'à la modification en temps réel d'attributs en fonction de l'état du système.

3.8 Synthèse et perspectives d'extensions

Nous avons décrit dans ce chapitre les spécifications de l'outil de modélisation UML utilisé au sein de la plate-forme medPRO. Les machines d'état permettent la représentation des flux de patients d'une part et la représentation de l'activité des ressources d'autre part. Les diagrammes de classes permette de définir l'organisation du système et de déclarer l'ensemble des objets en interaction. Nous avons opté pour l'utilisation d'un formalisme de modélisation simple, essentiellement basé sur l'utilisation de boîtes et de flèches. Une telle représentation permet de valider le modèle en accord avec le personnel hospitalier, favorisant ainsi l'échange et la communication autour de la modélisation.

Un certain nombre de perspectives d'extensions au modèle UML proposé dans ce chapitre ont été développées : (i) les états composites (section 3.8.1) permettent de modéliser une sous-machine d'état sous forme d'un état unique ; (ii) les modes d'exécution (section 3.8.2) permettent de définir rapidement plusieurs ensembles de ressources qualifiées pour la réalisation d'une tâche ; (iii) les activités internes d'une ressource (section 3.8.3) permettent de détailler un état particulier de la vue ressource.

3.8.1 États composites

Un état composite est un état possédant un compartiment supplémentaire, appelé compartiment de décomposition. Un compartiment de décomposition est une vue détaillée d'un état composite, dans laquelle on peut représenter les sous-états et les transitions d'un état composite. La figure 3.12 présente un état composite.

Le compartiment de décomposition peut être masqué afin d'augmenter la lisibilité du diagramme. Dans ce cas, une icône composite est utilisée pour indiquer que la décomposition de l'état n'est pas représentée sur ce diagramme. La figure 3.13 présente un état composite dont le compartiment de décomposition a été caché.

Un état composite est considéré actif lorsque la machine d'état est dans l'un des sous-états de cet état composite. Lorsqu'un état composite est actif, l'arborescence des états actifs allant de l'état composite lui-même jusqu'au sous-état courant est appelée *configuration d'état*. Par exemple, une configuration d'état pour la figure 3.12 pourrait être : `Intervention chirurgicale ->Extuber le patient`.

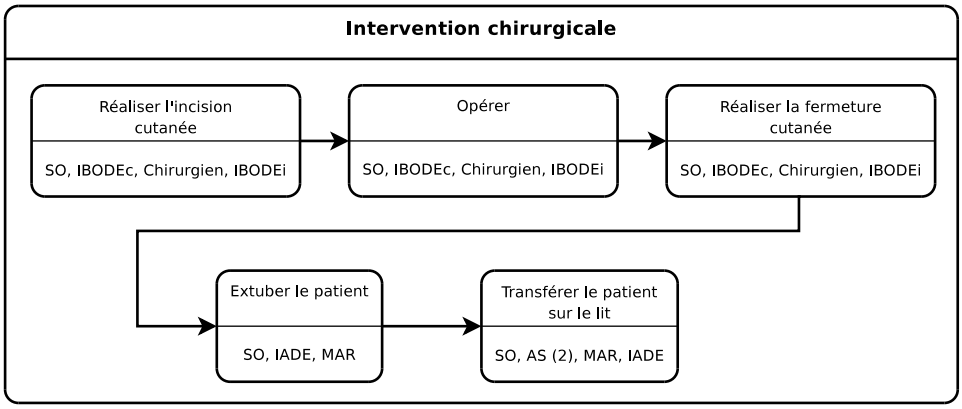


FIG. 3.12 – État composite associé à l'activité *Intervention chirurgicale*

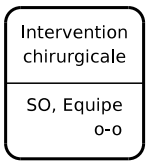


FIG. 3.13 – État composite possédant une icône composite

Les états composites sont très utiles pour modéliser un système selon plusieurs niveaux de détails. Cette approche est utilisée pour le bloc opératoire dont le modèle est présenté section 8.3. Une telle représentation, comparable à la décomposition adoptée dans SADT, offre une visualisation claire des activités majeures d'un système, tout en laissant la possibilité de détailler certaines d'entre elles.

3.8.2 Modes d'exécution

Plusieurs modes de fonctionnement peuvent être définis pour chaque activité. Chaque mode est décrit par une liste conjonctive de ressources, compétences ou équipes définies dans la vue organisation. Le sous-système de pilotage choisit l'une ou l'autre des combinaisons de ressources disponible en fonction de leur disponibilité pour éviter le blocage du système. L'effectif de chaque ressource doit être précisé.

Définition 3.8.1 (Mode d'exécution) *Un mode d'exécution est un couple $M = (\tau, R)$ tel que :*
- *τ modélise la durée d'exécution de la tâche au moyen d'une constante, d'une variable aléatoire ou d'une fonction définie par l'utilisateur ;*
- *$R = \{(r_1, eff_1), \ldots, (r_n, eff_n)\}$ est une liste conjonctive de ressources requise pour l'exécution de la tâche associées à l'effectif requis eff_i, $i \in \{1, \ldots, n\}$.*

Définition 3.8.2 (P-état) *Un P-état est un n-uple $pe = (l, m_1, \ldots, m_{n-1})$, $n \in \mathbb{N}^*$ où l est l'étiquette du P-état et m_i est un mode d'exécution défini pour l'état pe, $i \in \{1, \ldots, n\}$.*

Pour simplifier, nous considérons uniquement deux modes d'exécution pour chaque P-état. Le mode normal est adopté tant que les ressources spécifiées dans ce mode sont disponibles. En revanche, le mode dégradé est adopté dès que le mode normal n'est plus utilisable.

La figure 3.14 présente un exemple d'état appartenant à la vue processus. Dans cet exemple, l'intervention chirurgicale modélisée par un état requiert une `salle_operatoire` (classe c_1) ; si aucune salle n'est libre au moment de l'intervention, une salle opératoire d'urgence est alors allouée (`salle_op_urgence`, classe c_2). Un `chirurgien` (classe c_3) et deux `ide` (classe c_4) sont également nécessaires. Si l'une des ressources manque encore dans le dernier cas, le patient est mis en attente et ne peut être opéré.

Formellement, l'état présenté dans cet exemple est défini par le couple $PE = (M_1, M_2)$ avec $M_1 = (50, \{(c_1, 1), (c_3, 1), (c_4, 2)\})$ et $M_2 = (50, \{(c_2, 1), (c_3, 1), (c_4, 2)\})$. M_1 et M_2 représentent les deux modes d'exécutions possibles pour la réalisation de cette activité.

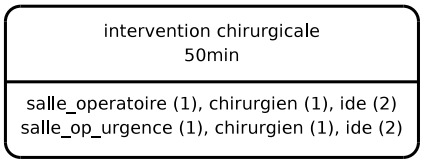


FIG. 3.14 – Exemple d'allocation conditionnelle de ressources

Les modes d'exécution offrent une solution simple et rapide à mettre en œuvre pour remédier aux éventuels blocages lors de la simulation. La conversion des modes d'exécution au sein des réseaux de Petri est réalisée en utilisant les couleurs : pour chaque transition associée à un P-état, une couleur différente est définie pour chaque mode d'exécution.

3.8.3 Activités internes d'une ressource

Chaque état de la vue ressource peut être détaillé afin de faire apparaître des tâches ou des activités internes supplémentaires.

Définition 3.8.3 (Activité interne) *Une activité interne est un couple $a = (\tau, R)$ tel que :*
- *τ modélise la durée d'exécution de l'activité au moyen d'une constante ou d'une variable aléatoire ;*
- *$R = \{(r_1, eff_1), \ldots, (r_n, eff_n)\}$, $n \in \mathbb{N}^*$ est une liste conjonctive de ressources accompagnée de l'effectif requis eff_i, $i \in \{1, \ldots, n\}$.*

Définition 3.8.4 (R-état) *Un R-état est un n-uple $re = (l, a_1, \ldots, a_{n-1})$, $n \in \mathbb{N}^*$ où l est l'étiquette du R-état et a_i est une activité interne appartenant à l'état re, $i \in \{1, \ldots, n\}$.*

Définition 3.8.5 (R-état simple) *Un R-état simple est un R-état qui ne possède qu'une seule activité interne.*

La figure 3.15 présente un état décrit avec plusieurs compartiments modélisant le transport d'un patient. Les activités internes décrites dans le deuxième compartiment permettent de connaître le détail des tâches que la ressource accomplit lorsqu'elle se trouve dans cet état. Trois tâches sont présentées dans cet exemple, décrivant (i) l'installation du patient sur un brancard, (ii) le transport de ce patient et (iii) l'installation du patient sur son lit. La troisième tâche requiert l'intervention d'une aide-soignante (`as`) dont la ressource sera désignée par r. L'état présenté dans cet exemple est défini par $re = (a_1, a_2, a_3)$ avec $a_1 = (5min, \oslash)$, $a_2 = (15min, \oslash)$ et $a_3 = (5min, \{(r, 1)\})$.

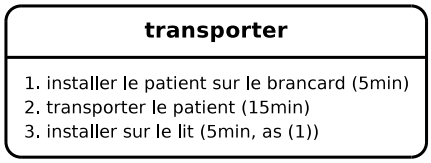


FIG. 3.15 – Un état possédant plusieurs activités internes

Chapitre 4

Comportement dynamique

Ce chapitre est consacré à la description du comportement dynamique du modèle UML spécifié dans le chapitre précédent. Dans cette optique, nous avons opté pour une conversion du modèle en réseau de Petri ; cet outil offre en effet un formalisme mathématique rigoureux permettant la description formelle du comportement dynamique du modèle ainsi que des applications en matière de planification et d'ordonnancement transposables au milieu hospitalier. Une définition formelle d'une nouvelle classe de réseaux de Petri est détaillée, à savoir les réseaux de Petri de santé. Une transposition de l'algorithme général de simulation à événements discret est également définie. Nous disposons ainsi d'une méthode complète permettant l'exécution et la simulation du modèle UML/medPRO au travers d'une classe particulière de réseaux de Petri.

4.1 Introduction

La plate-forme medPRO offre un formalisme de modélisation basé sur UML simple et intuitif. Les modalités d'utilisation du langage permettant à tout utilisateur de comprendre l'utilisation d'UML au sein de la plate-forme medPRO ont été spécifiées dans le chapitre précédent. Cependant les modèles réalisés sous UML sont appelés à être simulés : afin de décrire précisément le comportement dynamique de ces modèles, nous avons opté pour une conversion automatique en réseau de Petri, dont la dynamique d'exécution est parfaitement connue. Nous nous affranchissons ainsi de la spécification formelle de l'exécution dynamique d'un modèle UML, sujette à imprécisions, et définissons uniquement des règles de conversion en réseau de Petri au moyen d'algorithmes implémentés au sein de la plate-forme medPRO.

L'ensemble des diagrammes d'état et de classes qui constituent les différentes vue d'un modèle medPRO peuvent être représentés sous forme d'une classe particulière de réseaux de Petri, appelés réseaux de Petri de santé. Ce choix n'est pas anodin :

- les réseaux de Petri offrent une description formelle du comportement dynamique du modèle permettant de simplifier les algorithmes de simulation ;
- les propriétés des réseaux de Petri peuvent être transposées aux réseaux générés à partir de modèles UML, permettant la vérification de propriétés structurelles ;
- les applications des réseaux de Petri en matière de planification et d'ordonnancement peuvent être appliquées aux systèmes hospitaliers (c.f. chapitre 5) ;
- l'automatisation de la conversion et de la simulation du modèle réalisé grâce à la plate-forme medPRO est possible, offrant un gain de temps supplémentaire à l'utilisateur.

Après une description détaillée des différents algorithmes permettant la conversion d'un modèle UML en réseau de Petri section 4.2, nous proposons une définition formelle de la classe de réseaux de Petri de santé section 4.3. Les mécanismes de simulation mis en œuvre ainsi que les indicateurs de performance

sélectionnés sont décrits section 4.4. Enfin, une synthèse est proposée section 4.5. Le lecteur pourra se référer à l'annexe A où figurent quelques rappels de notions de bases sur les réseaux de Petri utilisées dans ce chapitre.

4.2 Algorithme de conversion

Un algorithme de conversion automatique permettant de passer du modèle UML du système physique à un réseau de Petri a été mis au point. Nous avons choisi de décrire la procédure de conversion avant de définir la classe de réseau de Petri de santé étant donné que les propriétés de ce réseau découlent de sa construction. La conversion est réalisée de manière séquentielle : la vue processus est traitée en premier avec la conversion du parcours patient. Les sous-réseaux correspondant aux différents types de ressources et à leurs interactions sont ensuite ajoutés. Enfin, les machines d'état associées aux missions de la vue ressource sont modélisées. La première étape du processus de conversion consiste à initialiser les ensembles et fonctions de référence.

4.2.1 Initialisation

La phase d'initialisation permet de définir les ensembles et les fonctions qui permettront la construction du réseau de Petri au fûr et à mesure de la conversion.

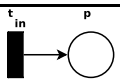
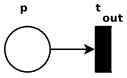
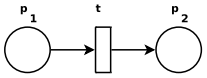
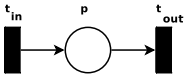
Ensembles de référence

Les ensembles suivants sont construits à partir du modèle UML :
- E, ensemble des états des vues processus et ressource ;
- V, ensemble des arcs des vues processus et ressource ;
- R, ensemble des ressources simples déclarées dans la vue organisation ;
- CO, ensemble des compétences déclarées dans la vue organisation ;
- EQ, ensemble des équipes déclarées dans la vue organisation ;

Correspondance medPRO-RdP

Soit $\mathcal{RP} = (P, T, F, W, \theta, M_0)$ un réseau de Petri temporisé classique. Avant de commencer la conversion, nous établissons une table de correspondance (table 4.1) entre les éléments du modèle UML initial et les éléments du réseau à construire. Ainsi nous choisissons de faire correspondre à tout état une transition possédant une seule place d'entrée et une seule place de sortie. Une entrée correspond à une transition source suivie d'une place tandis qu'une sortie correspond à une transition puits précédée d'une place. L'état de base de toute ressource devient un triplet transition source, place de base, transition puits et une transition conditionnelle UML est associée à une transition. Les ensembles associés à ces places et transitions particulières sont également définis :
- $T_{in} \subset T$ (resp. $T_{out} \subset T$) est l'ensemble des transitions source (resp. puits) du réseau de Petri ;
- $TE \subset T$ est l'ensemble des transitions correspondant à des états ;
- $TC \subset T$ est l'ensemble des transitions correspondant aux transitions conditionnelles UML entre états ;
- $PB \subset P$ est l'ensemble des places correspondant aux états de base des ressources.

Les éléments ajoutés dans ces ensembles lors de la conversion sont également ajoutés dans les ensembles P et T. Les transitions sources et les transitions puits sont représentées par un rectangle plein. Les transitions correspondant à des états sont représentées par un rectangle vide. Les transitions correspondant aux transitions conditionnelles UML sont représentées par un trait. Ce formalisme permet de mieux visualiser les différents éléments des réseaux de Petri générés.

medPRO	Réseau de Petri	Ensembles
Entrée		$t_{in} \in T_{in}$; $p \in P$
Sortie		$t_{out} \in T_{out}$; $p \in P$
P-état ou R-état		$p_1, p_2 \in P$; $t \in TE$
État de base		$t_{in} \in T_{in}$; $p \in PB$; $t_{out} \in T_{out}$
Transition cond.		$t \in TC$

TAB. 4.1 – Correspondance medPRO/Réseau de Petri

Fonctions usuelles

Les fonctions suivantes sont initialisées pendant la conversion et permettrons d'établir une correspondance claire entre de modèle UML d'origine et le réseau de Petri en construction :

- $\gamma : E \rightarrow TE$, fonction injective qui associe à un état $e \in E$ quelconque sa transition équivalente $t \in TE$; cette fonction permet de retrouver la transition correspondant à un état déjà converti ;
- $\eta : V \rightarrow TC$, fonction injective qui associe à un arc entre deux états sa transition équivalente $t \in TC$;
- $\Gamma : R \cup CO \cup EQ \rightarrow PB$, fonction injective qui associe à une classe de ressource, de compétence ou d'équipe $r \in R \cup CO \cup EQ$ sa place de base $p \in PB$.

Enfin nous définissons la fonction $\xi : TC \rightarrow \mathbb{E}$ permettant de vérifier la condition de garde liée à une transition conditionnelle.

4.2.2 Conversion de la vue processus

La première phase de la conversion consiste à créer le réseau de Petri associé à la vue processus sans tenir compte des allocations/libérations de ressources. Nous nous intéressons uniquement au parcours patient et aux différents branchements de ce parcours. Pour chaque machine d'état $PM = (E, V, \zeta)$ appartenant à la vue processus, l'algorithme de conversion suivant est appliqué :

1. Convertir l'entrée en un couple transition source $t_{in} \in T_{in}$ – place $p \in P$; ajouter l'arc (t_{in}, p) à l'ensemble F.
2. Convertir la sortie en un couple place $p \in P$ – transition puits $t_{out} \in T_{out}$; ajouter l'arc (p, t_{out}) à l'ensemble F.
3. Convertir chaque P-état $e \in E$ de durée τ en un triplet place d'entrée $p_1 \in P$ – transition $t \in TE$ – place de sortie $p_2 \in P$; ajouter les arcs (p_1, t) et (t, p_2) à l'ensemble F. La transition t est associée à l'état $e : \gamma(e) = t$. La durée de franchissement de t est initialisée : $\theta(t) = \tau$.
4. Pour chaque transition conditionnelle $(e, f) \in V$:
   - créer une transition $t \in TC$;
   - ajouter $(\gamma(e)\bullet, t)$, $(t, \bullet\gamma(f))$ à l'ensemble F ;
   - associer la transition t à (e, f) : $\eta(e, f) = t$;
   - associer la condition à t : $\xi(t) = \zeta(e, f)$.

Les poids associés aux arcs du réseau résultant sont tous égaux à 1 : $W(x,y) = 1 \ \forall \ (x,y) \in F$. Chaque machine d'état de la vue processus correspond à un sous-réseau de Petri indépendant noir et blanc. L'algorithme de conversion est détaillé dans l'annexe B.1.

Exemple 4.2.1 *La figure 4.1 présente un exemple de modélisation de P-état sans allocation de ressource. Les états* `E1`*,* `E2`*,* `E3` *et* `E4` *correspondent respectivement aux transitions* t_3*,* t_5*,* t_8 *et* t_{10} *du réseau de Petri. Pour ce premier exemple nous ne précisons pas les durées des états ni les conditions des transitions en sortie de* `E1`*. Selon l'algorithme de conversion, le réseau de Petri correspondant s'écrit de la manière suivante :*

(i) $P = \{p_1, \ldots, p_{10}\}$.

(ii) $T = \{t_1, \ldots, t_{12}\}$, $TE = \{t_3, t_5, t_8, t_{10}\}$ *et* $TC = \{t_2, t_4, t_6, t_7, t_9, t_{11}\}$.

(iii) $F = \{(t_1, p_1), (p_1, t_2), \ldots\}$.

(iv) $W(x,y) = 1 \ \forall (x,y) \in F$ *;*

(v) Le marquage initial est nul.

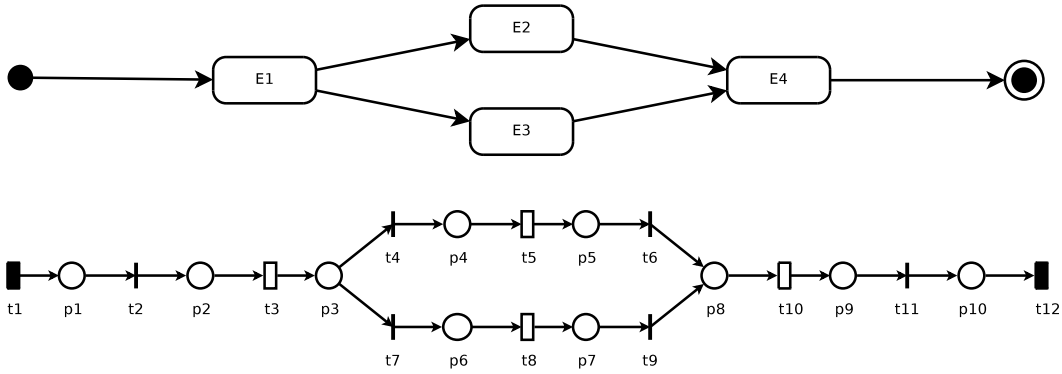


FIG. 4.1 – Conversion d'une machine d'état sans allocation de ressources en réseau de Petri

4.2.3 Conversion des ressources déclarées dans la vue organisation

Une fois la vue processus convertie en réseau de Petri, nous ajoutons les états de base correspondant aux ressources simples, équipes et compétences déclarées dans la vue organisation :

1. Pour chaque ressource simple $r \in R$ d'effectif eff, créer un triplet transition source $t_{in} \in T_{in}$ – place de base $p \in PB$ – transition puits $t_{out} \in T_{out}$; ajouter les arcs (t_{in}, p) et (p, t_{out}) à l'ensemble F. La place de base p est associée à la ressource simple $r : \Gamma(r) = p$. De plus nous avons $W(t_{in}, p) = W(p, t_{out}) = eff$.
2. Pour chaque équipe et compétence $r \in EQ \cup CO$, créer une place de base $p \in PB$. Cette place est associée à la ressource $r : \Gamma(r) = p$.

Cet algorithme de conversion est détaillé dans l'annexe B.2. À l'issu de son exécution, nous disposons d'un réseau de Petri contenant l'ensemble des places de base des ressources déclarées dans la vue organisation ainsi que les sous-réseaux correspondant à chaque machine d'état de la vue processus.

4.2.4 Conversion de la vue ressource

La prochaine étape consiste à convertir les missions définies dans la vue ressource du système physique. Nous décomposons la conversion de la vue ressource en deux étapes : (i) la conversion des machines d'état liées aux missions des ressources, et (ii) l'allocation ponctuelle d'une ressource à une tâche.

Missions

La première phase de la conversion consiste à intégrer les machines d'état relatives aux différentes missions pour ressources simples, compétences ou équipes déclarées dans la vue ressource . Il s'agit ici de formaliser le mécanisme de synchronisation décrit dans le chapitre précédent. Soit $mi = (M, e_{in}, e_{out}, eff_{mi})$ avec $M = (E, V, \zeta)$ une mission définie pour la ressource $r \in R \cup CO \cup EQ$.

Nous convertissons tout d'abord la machine d'état correspondant à la mission mi de la même manière qu'une machine d'état de la vue processus :

1. Créer un couple transition $t_{in} \in T$ – place $p \in P$; ajouter l'arc (t_{in}, p) à l'ensemble F. Ce sous-réseau constitue l'entrée de la mission.
2. Créer un couple place $p \in P$ – transition $t_{out} \in T$; ajouter l'arc (p, t_{out}) à l'ensemble F. Ce sous-réseau constitue la sortie de la mission.
3. Convertir chaque R-état $e \in E$ de durée τ en un triplet place d'entrée $p_1 \in P$ – transition $t \in TE$ – place de sortie $p_2 \in P$; ajouter les arcs (p_1, t) et (t, p_2) à l'ensemble F. La transition t est associée à l'état e : $\gamma(e) = t$. La durée de franchissement de t est initialisée : $\theta(t) = \tau$.
4. Pour chaque transition conditionnelle $(e, f) \in V$:
   - créer une transition $t \in TC$;
   - ajouter $(\gamma(e)\bullet, t)$, $(t, \bullet\gamma(f))$ à l'ensemble F ;
   - associer la transition t à (e, f) : $\eta(e, f) = t$;
   - associer la condition à t : $\xi(t) = \zeta(e, f)$.
5. Ajouter les arcs $(\Gamma(r), t_{in})$ et $(t_{out}, \Gamma(r))$ à l'ensemble F. La condition de démarrage de la mission est initialisée : $\xi(t_{in}) = \zeta(r)$. $W(\Gamma(r), t_{in}) = W(t_{out}, \Gamma(r)) = eff_{mi}$.

Les poids associés aux arcs du réseau résultant sont tous égaux à 1 : $W(x, y) = 1 \ \forall \ (x, y) \in F$. Nous obtenons ainsi le réseau de Petri associé à la mission mi, dont l'entrée et la sortie sont modélisées par les transitions t_{in} et t_{out} respectivement. La synchronisation entre ce réseau de Petri et le réseau associé à la vue processus créé précédemment est réalisée en fusionnant les transitions communes et les sous-réseaux de transitions communs de ces machines d'état. L'algorithme détaillé pour cette conversion est donné dans l'annexe B.3.

Exemple 4.2.2 *La figure 4.2 présente un exemple de conversion avec synchronisation entre la vue processus et la vue ressource. Le sous-réseau de Petri G_1 est la conversion de la vue processus : les états* `Etat1`, `Etat2`, `Etat3`, `Etat4` *et* `Etat5` *correspondent respectivement aux transitions t_1, t_3, t_5, t_8 et t_{10} ; la place correspondant à l'état de base de la ressource* `R1` *est p_9. Le sous-réseau de Petri G_2 est la conversion de la mission représentée dans la vue ressource : les états* `Etat2`, `Etat3`, `Etat4` *et* `Etat6` *correspondent respectivement aux transitions t'_3, t'_5, t'_8 et t'_{10} ; l'entrée et la sortie de la mission sont modélisées par les transitions t'_1 et t'_{12} respectivement. Le troisième réseau est le résultat de la fusion des transitions communes liées aux états synchronisés ; l'entrée et la sortie de la mission sont reliées à la place de base de la ressource considérée.*

Allocation ponctuelle

Reproduire le mécanisme d'allocation ponctuelle (la ressource est allouée pour une seule tâche et libérée immédiatement) consiste à relier une place de base à une transition modélisant un état. Pour chaque état $e = (l, \tau, R) \in E$, l'ensemble R des ressources requises est examiné. Pour chaque ressource $(r, eff) \in R$, les arcs $(\gamma(e), \Gamma(r))$ et $(\Gamma(r), \gamma(e))$ sont ajoutés à F. De plus nous avons $W(\gamma(e), \Gamma(r)) = W(\Gamma(r), \gamma(e)) = eff$. Cet algorithme de conversion est détaillé dans l'annexe B.5.

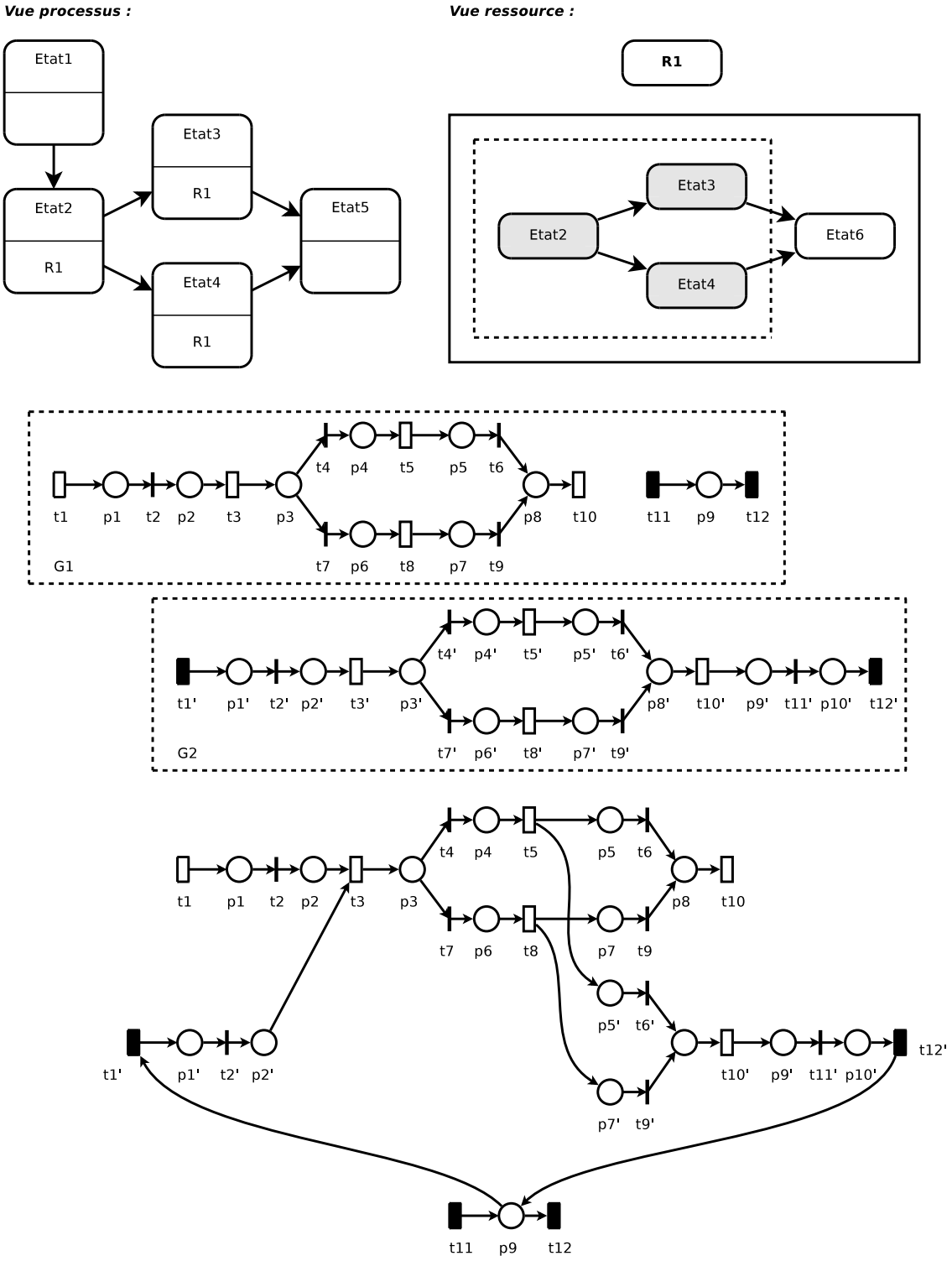


FIG. 4.2 – Conversion d'une machine d'état avec synchronisation

Exemple 4.2.3 *La figure 4.3 présente un exemple de modélisation de P-état avec allocation ponctuelle de ressources. Les états* `E1` *et* `E2` *de la machine d'état correspondent aux transitions* t_1 *et* t_2 *du réseau de Petri. L'exécution de la tâche correspondant à* `E2` *nécessite une instance de la ressource* `R1`*, une instance de la ressource* `R2` *et deux instances de la ressource* `R3`*. Les trois ressources requises sont des ressources simples et sont modélisées par les places* pr_1*,* pr_2 *et* pr_3*. Pour plus de clarté, les transitions*

source et les transitions puits associées à ces ressources ne sont pas représentées et les arcs aller-retour sont représentés par des flèches doubles.

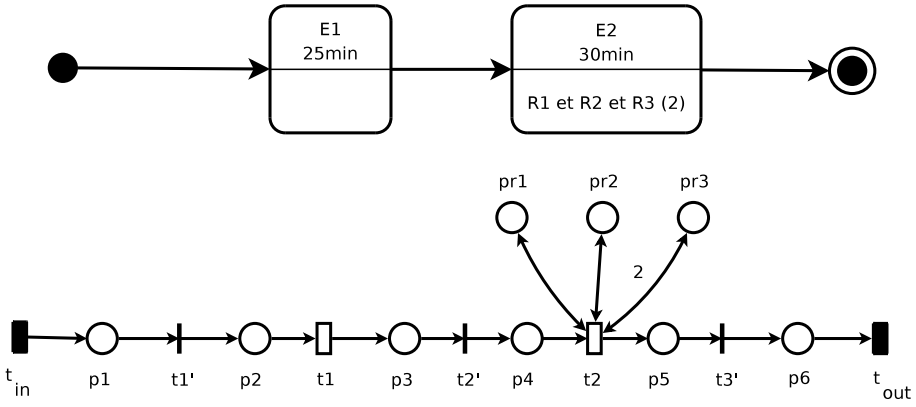


FIG. 4.3 – Conversion d'une machine d'état avec allocation ponctuelle de ressources

Selon l'algorithme de conversion, le réseau de Petri présenté figure 4.3 s'écrit de la manière suivante :

(i) $P = \{p_1, \ldots, p_6, pr_1, pr_2, pr_3\}$ *et* $P_{base} = \{pr_1, pr_2, pr_3\}$.

(ii) $T = \{t_{in}, t_{out}, t_1, t_2, t'_1, t'_2, t'_3\}$, $TE = \{t_1, t_2\}$ *et* $TC = \{t'_1, t'_2, t'_3\}$.

(iii) $F = \{(t_{in}, p_1), (p_1, t'_1), (t'_1, p_2), (p_2, t_1), (t_1, p_3), (p_3, t'_2), \ldots\}$.

(iv) $W(pr_3, t_2) = W(t_2, pr_3) = 2$ *et* $W(x, y) = 1\ \forall (x, y) \in F \setminus \{(pr_3, t_2), (t_2, pr_3)\}$.

(v) $\theta(t_1) = 25\ min$, $\theta(t_2) = 30\ min$;

(vi) $\xi(t) = 1\ \forall t \in TP_{cond}$;

(vii) Le marquage initial est nul.

4.2.5 Conversion des relations entre ressources

La dernière étape de la conversion consiste à modéliser les relations entre ressources permettant de décrire héritages, compétences et équipes.

Équipes

Soit $EQ = \{eq_1, \ldots, eq_n\}$ l'ensemble des classes d'équipe. L'algorithme 4.1 décrit la mise en place des états de base pour les classes d'équipe déclarées dans la vue organisation. Nous définissons deux transitions t_{aller} et t_{retour} permettant de réunir ou de libérer tous les membres de l'équipe (lignes 2–3). Nous définissons d'autre part n places et transitions intermédiaires permettant de pré-sélectionner les ressources durant la constitution de l'équipe (lignes 5–8). La figure 4.4 illustre le sous-réseau associé à une équipe.

D'un point de vue dynamique, la constitution ou la dissolution d'une équipe ne peuvent avoir lieu que dans certaines conditions telle l'allocation de ressource pour une tâche ou pour le début d'une mission. Le tir des transitions de ce sous-réseau est déterminé par le système de pilotage.

Héritage et compétences

Pour modéliser les relations liées à l'héritage et à l'association par compétence, nous avons besoin d'introduire des couleurs dans le réseau de Petri considéré jusqu'à maintenant. Soit Ω l'ensemble des couleurs considérées. Une couleur spécifique $c(r)$ est associée à chaque ressource **simple**. Nous avons

Algorithme 4.1 Mise en place du sous-réseau associé à une équipe

```
1   pour chaque équipe eq = (A, R, f, MI) ∈ EQ faire
2       T ← T ∪ {t_aller, t_retour}
3       F ← F ∪ {(t_aller, Γ(eq)), (Γ(eq), t_retour)}
4       pour chaque ressource (r_i, eff_i) ∈ R, i ∈ {1, ..., n} faire
5           T ← T ∪ {t_i}
6           P ← P ∪ {pr'_i}
7           F ← F ∪ {(Γ(r_i), t_i1), (t_i1, pr'_i), (t_retour, Γ(r_i))}
8           W(pr_i, t_i) = W(t_i, pr'_i) = W(pr'_i, t_aller) = W(t_retour, pr_i) = eff_i
9       fin pour
10      W(t_aller, Γ(eq)) = W(Γ(eq), t_retour) = 1
11  fin pour
```

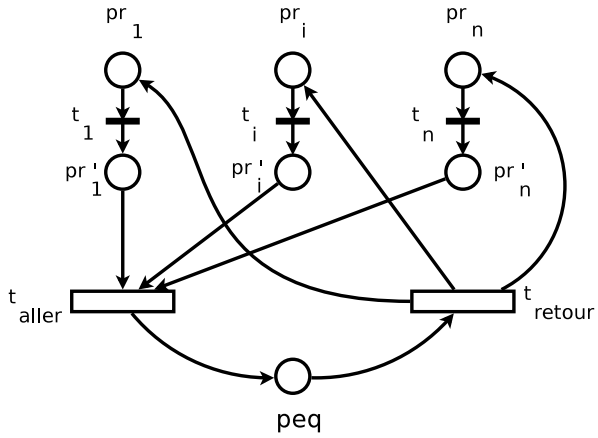


FIG. 4.4 – Sous-réseau associé à une équipe

donc $\Omega = \{c(r_1), \ldots, c(r_n)\}$ avec $r_1, \ldots, r_n \in R$. $c : R \to \Omega$ est donc une fonction bijective qui associe à une ressource sa couleur. Soient C, W^-, W^+ trois fonctions définies comme suit :

(i) $C : (P \cup T) \to \Omega$ est la fonction qui associe à une place ou à une transition un ensemble de couleurs. $C(p)$, $p \in P$, est l'ensemble des couleurs associées à la place p (i.e. l'ensemble des couleurs que peut contenir la place p) et $C(t)$, $t \in T$ est l'ensemble des couleurs associées à la transition t (i.e. l'ensemble des manières de franchir t). Une place de base de ressource simple ne peut contenir que des jetons de sa propre couleur : $C(\Gamma(r)) = c(r)\ \forall r \in R$.

(ii) $W^-_{p,t} : C(t) \to \mathbb{N}^{|C(P)|}$ est la fonction attachée aux arcs indiquant la pré-condition d'une transition par rapport à une couleur. $W^-_{p,t}(x)$, $x \in \mathbb{N}^*$, fait correspondre à chaque manière de franchir t (i.e. à chaque couleur associée à t) un vecteur qui a autant d'éléments que de couleurs dans les places du RdP. La valeur du ième élément de ce vecteur est un entier qui indique le nombre de jetons de cette couleur nécessaires pour franchir la transition t.

(iii) $W^+_{t,p} : C(t) \to \mathbb{N}^{|C(P)|}$ est la fonction attachée aux arcs indiquant la post-condition d'une transition par rapport à une couleur. $W^+_{t,p}(x)$, $x \in \mathbb{N}^*$, fait correspondre à chaque manière de franchir t (i.e. à chaque couleur associée à t) un vecteur dont les composantes sont des entiers qui indiquent le nombre de jetons de chaque couleur générés après le franchissement de la transition.

Soit $CH = \{(r_1, q_1), \ldots, (r_n, q_n)\}$ un ensemble de couples de ressources simples (r_i, q_i) telles que $r_i \triangleright q_i$. La relation d'héritage permet de modéliser la capacité de r_i à remplacer q_i pour la réalisation d'une tâche. L'algorithme 4.2 décrit la mise en place du réseau permettant cet échange. Pour chaque couple de ressources (r, q), une transition t_{aller} et une transition t_{retour} sont mises en place (lignes 2–3). Il n'existe qu'une manière de tirer t_{aller} et t_{retour} (ligne 4). La place $\Gamma(q)$ peut contenir des jetons

de couleur $c(q)$ ou de couleur $c(r)$ (ligne 5). Les pré-conditions et post-conditions de ces transitions permettent de fixer le comportement de ce réseau en autorisant le remplacement dans un seul sens (ligne 6). La figure 4.5 illustre le sous-réseau associé à une relation d'héritage.

D'un point de vue dynamique, la relation d'héritage permet de déterminer si une ressource peut être remplacée par une autre. Un remplacement ne peut avoir lieu que dans certaines conditions précises telle l'allocation de ressource pour une tâche ou pour le début d'une mission. Le tir des transitions t_{aller} et t_{retour} est déterminé par le système de pilotage.

Algorithme 4.2 Mise en place du sous-réseau associé à une relation d'héritage

1 **pour** chaque couple $(r, q) \in CH$ **faire**
2 $\quad T \leftarrow T \cup \{t_{aller},\ t_{retour}\}$
3 $\quad F \leftarrow F \cup \{(\Gamma(r), t_{aller}),\ (t_{aller}, \Gamma(q),\ (\Gamma(q), t_{retour}),\ (t_{retour}, \Gamma(r))\}$
4 $\quad C(t_{aller}) = C(t_{retour}) = \{1\}$
5 $\quad C(\Gamma(q)) \leftarrow C(\Gamma(q)) \cup \{c(r)\}$
6 $\quad W^-_{\Gamma(r),t_{aller}} = W^+_{t_{aller},\Gamma(q)} = W^-_{\Gamma(q),t_{retour}} = W^+_{t_{retour},\Gamma(r)} = 1$
pour la composante $c(r)$, 0 sinon
7 **fin pour**

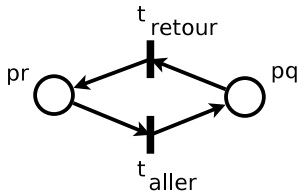


FIG. 4.5 – Sous-réseau coloré associé à une relation d'héritage

L'algorithme 4.3 décrit la mise en place du sous-réseau permettant de lier une compétence aux ressources simples qui la possède. Pour chaque ressource simple r_i possédant la compétence co, une transition aller $t_{1,i}$ et une transitions retour $t_{2,i}$ sont créées (lignes 3–5). Les conditions de tir associées aux transitions $t_{1,i}, \ldots, t_{2,n}$ sont fixées (ligne 8). La figure 4.6 illustre le sous-réseau associé à une compétence.

Plusieurs facteurs déterminent la sélection d'une ressource par compétence : allocation au début d'une tâche ou début d'une mission associée à cette compétence. Dans tous les cas, le système de pilotage est chargé de sélectionner la ressource appropriée selon les conditions et de la restituer à la fin de la tâche ou de la mission selon sa couleur.

Algorithme 4.3 Mise en place du sous-réseau associé à une compétence

1 **pour** chaque compétence $co = (A, R, MI) \in CO$ **faire**
2 $\quad$ **pour** chaque ressource $r_i \in R$, $i \in \{1, \ldots, n\}$ **faire**
3 $\quad\quad T \leftarrow T \cup \{t_{1,i},\ t_{2,i}\}$
4 $\quad\quad F \leftarrow F \cup \{(\Gamma(r_i), t_{1,i}),\ (t_{2,i}, \Gamma(r_i))\}$
5 $\quad\quad F \leftarrow F \cup \{(t_{1,i}, \Gamma(co)),\ (\Gamma(co), t_{2,i})\}$
6 $\quad\quad C(\Gamma(eq)) \leftarrow C(\Gamma(eq)) \cup \{c(r_i)\}$
7 $\quad\quad C(t_{1,i}) = C(t_{2,i}) = \{1\}$
8 $\quad\quad W^-_{\Gamma(r_i),t_{1,i}}(1) = W^+_{t_{1,i},\Gamma(co)}(1) = W^-_{\Gamma(co),t_{2,i}}(1) = W^+_{t_{2,i},\Gamma(r_i)}(1) = 1$
pour la composante $c(r_i)$, 0 sinon
9 $\quad$ **fin pour**
10 **fin pour**

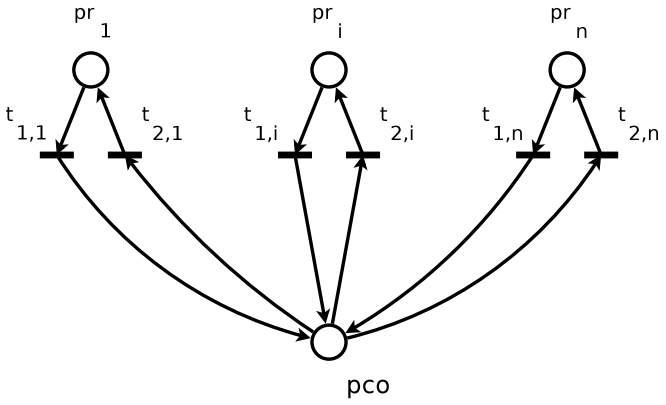


FIG. 4.6 – Sous-réseau coloré associée à une compétence

Exemple 4.2.4 *Considérons l'exemple décrit figure 4.7. Le modèle UML est présenté dans la partie supérieure (vues processus, ressource et organisation). Le réseau de Petri équivalent est présenté dans la partie inférieure. La vue processus comporte une machine d'état partielle composée de cinq états* `Etat1` *à* `Etat5`*. Cinq classes de ressources* `R1`*,...,*`R5` *et une classe de compétence* `C1` *sont considérées. Enfin, une unique mission est définie pour la classe de ressource* `R1` *dont la machine d'état comporte trois états* `Etat2`*,* `Etat3` *et* `Etat6` *;* `Etat2` *et* `Etat3` *constituent une sous-machine d'état synchronisée.*

Le réseau de Petri partiel équivalent fait apparaitre les transitions t_1, t_3, t_5, t_8 et t_{10} correspondant aux états de la vue processus et la transition t_{16} appartenant à la mission définie pour `R1` *et correspondant à l'état* `Etat6`*. Cinq places $pr_1, \ldots, pr_5$ correspondent aux classes de ressources et une place pc_1 correspond à la classe de compétence, toutes déclarées dans la vue organisation du modèle. Pour une meilleure lisibilité, les arcs aller et retour entre la place pr_3 et la transition t_5 sont représentés par une double flèche, et les transitions sources/puits ne sont pas représentées. Le RdPS de la figure 4.7 s'écrit de la manière suivante :*

(i) P *est l'ensembles des places du réseau avec* $PB = \{pr_1, pr_2, pr_3, pr_4, pr_5, pc_1\}$.

(ii) T *est l'ensembles des transitions du réseau, avec :*
- $TE = \{t_1, t_3, t_5, t_8, t_{10}\}$;
- $TC = \{t_2, t_4, t_6, t_7, t_9, t_{14}, t_{15}, t_{17}\}$.

(iii) $F = \{(t_1, p_1), \ldots\}$.

(iv) L'ensemble des couleurs est $\Omega = \{c(r_1), c(r_2), c(r_3), c(r_4), c(r_5)\}$.
$C(pr_1) = \{c(r_1), c(r_5)\}$, $C(pr_2) = \{c(r_2)\}$, $C(pr_3) = \{c(r_3)\}$, $C(pr_4) = \{c(r_4)\}$,
$C(pr_5) = \{c(r_5)\}$, $C(pc_1) = \{c(r_3), c(r_4)\}$.

(v) Seuls les sous-réseaux correspondants aux relations entre ressources sont colorés (encadrés en pointillés fins sur la figure) :
- $W^-_{pr_3,t_{20}}(1) = W^+_{t_{20},pc_1}(1) = W^-_{pc_1,t_{19}}(1) = W^+_{t_{19},pr_3}(1) = [0,0,1,0,0]$;
- $W^-_{pr_4,t_{22}}(1) = W^+_{t_{22},pc_1}(1) = W^-_{pc_1,t_{21}}(1) = W^+_{t_{21},pr_4}(1) = [0,0,0,1,0]$;
- $W^-_{pr_5,t_{12}}(1) = W^+_{t_{12},pr_1}(1) = W^-_{pr_1,t_{11}}(1) = W^+_{t_{11},pr_5}(1) = [0,0,0,0,1]$.

Les autres fonctions de poids sont définies de manière classique :
$W(pr_2, t_1) = W(t_1, pr_2) = 2$, *et* $W(x,y) = 1\ \forall (x,y) \in F \setminus \{(r_2,t_1),(t_1,r_2)\}$.

(vi) $\theta(t_1) = 10\text{min},\ \theta(t_3) = 15\text{min},\ \ldots,\ \theta(t_{10}) = UNIF(10,20)\text{min}$.

(vii) $\xi(t_4) = \text{cond1}$, $\xi(t_7) = \text{cond2}$.

(viii) Le marquage initial est nul.

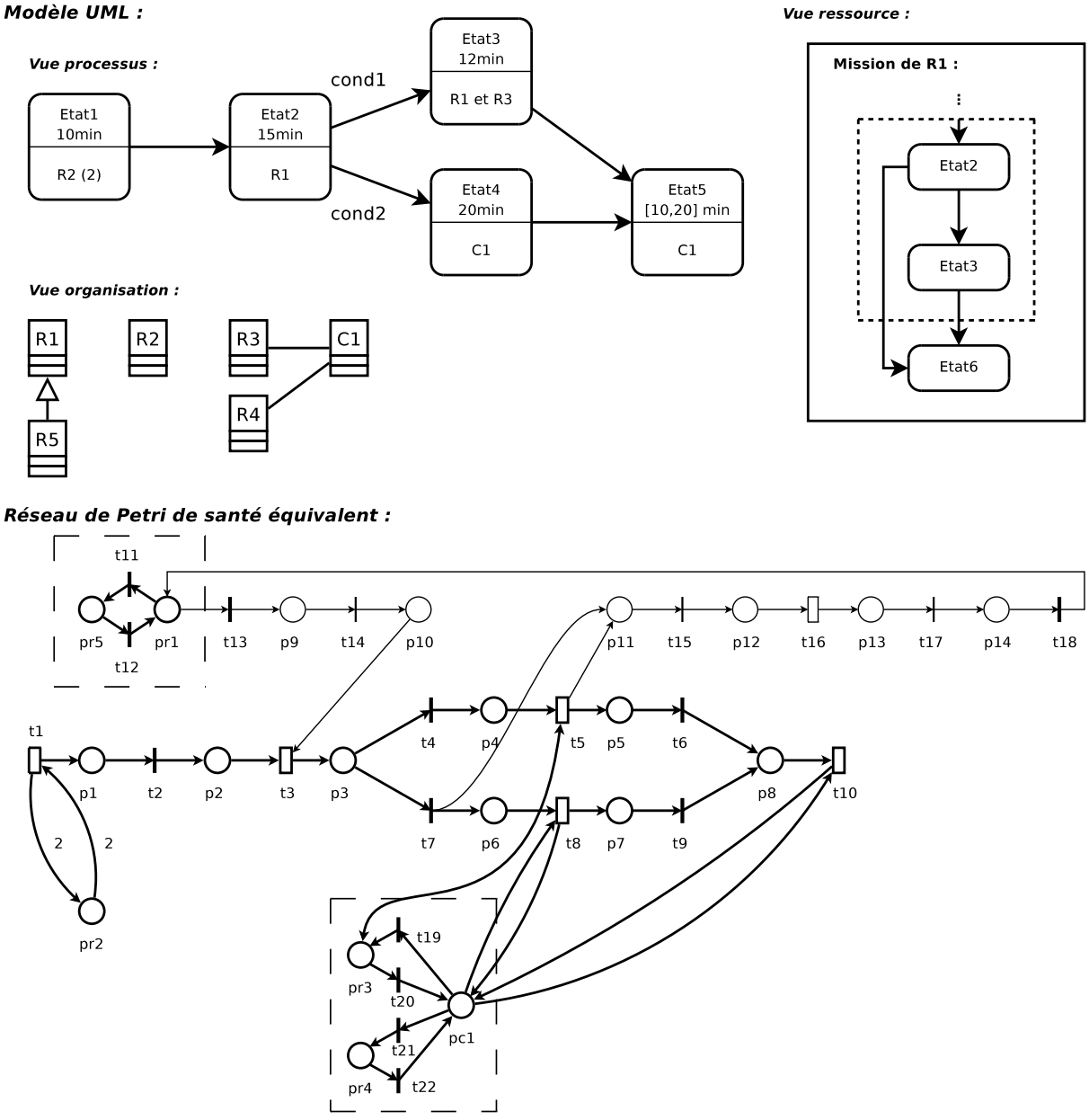


FIG. 4.7 – Conversion d'une machine d'état avec synchronisation

Les couleurs servent principalement à identifier les ressources sur le réseau : la mission définie pour `R1` *peut également être accomplie par* `R5` *car* `R5` *hérite de* `R1`*. Cependant la couleur du jeton qui circule dans ce réseau n'est pas considérée pendant la mission car elle n'influence pas la dynamique modélisée ; la couleur est examinée uniquement en début et en fin de mission. Il en est de même pour la compétence* `C1` *: les transitions* t_8 *et* t_{10} *sont tirées de la même façon quelle que soit la ressource sélectionnée.*

4.3 Une classe de réseaux de Petri : un réseau de Petri de santé

Nous nous proposons de définir une classe particulière de réseaux de Petri adaptées à nos besoins. Nous appelons cette classe réseau de Petri de santé. La définition de ce réseau et de ses propriétés sont décrites dans cette section. Tout modèle réalisé selon le formalisme proposé dans la plate-forme medPRO peut être converti en réseau de Petri de santé.

4.3.1 Préliminaires

Définition 4.3.1 (Machine d'état) *Une machine d'état est un sous-réseau de Petri $M = (P, T, F, M_0)$ tel que :*

(i) P *est l'ensemble des places du réseau.*

(ii) T *est l'ensemble des transitions du réseau. Toute transition possède une unique place d'entrée et une unique place de sortie :* $\forall t \in TE, |\bullet t| = |t \bullet| = 1$. *De plus nous définissons les ensembles* TE *et* TC *tels que :*
- $TE \subset T$ *est l'ensemble des transitions associées à des états.*
- $TC \subset T$ *est l'ensemble des transitions associées à des transitions conditionnelles entre états.* $TE \cap TC = \oslash$.

(iii) F *est l'ensemble des arcs du réseau,* $F \subseteq (P \times T) \cup (T \times P)$.

(iv) $\forall (t, p) \in F$ *tel que* $t \in TE$ *(resp.* $t \in TC$*) et* $p \in P$ *alors* $|\bullet p| = 1$ *et* $p\bullet \in TC$ *(resp.* $|\bullet p| \geq 1$ *et* $p\bullet \in TE$*).*

(v) $\forall (p, t) \in F$ *tel que* $p \in P$ *et* $t \in TE$ *(resp.* $t \in TC$*) alors* $|p \bullet| = 1$ *et* $\bullet p \in TC$ *(resp.* $|p \bullet| \geq 1$ *et* $\bullet p \in TE$*).*

(vi) M *possède au moins une transition source et au moins une transition puits appelées entrée et sortie :* $\exists t_{in}, t_{out} \in T$ *tel que* $\bullet t_{in} = t_{out}\bullet = \oslash$.

(vii) M_0 *est le marquage initial du réseau.*

Les hypothèses (iv) et (v) de la définition précédente permettent d'imposer l'alternance entre transitions associées à des états et transitions conditionnelles entre états dans le sous-réseau de Petri. Une machine d'état permet la représentation d'un flux de la vue processus ou d'une mission de la vue ressource.

Exemple 4.3.1 *La figure 4.8 présente un exemple de machine d'état. Les transitions* t_1, t_3, t_5, t_8, t_{10}, t_{12} *et* t_{14} *(représentées par des rectangles) appartiennent à l'ensemble* TE, *tandis que les transitions* t_2, t_4, t_6, t_7, t_9, t_{11}, t_{13} *et* t_{15} *(représentées par des barres) appartiennent à l'ensemble* TC. t_1 *et* t_{12} *sont respectivement les transitions source et puits du réseau. On notera l'alternance entre transitions état et transitions conditionnelles.*

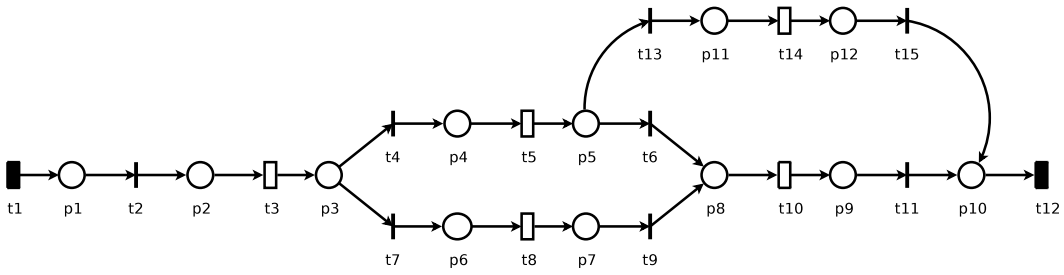


FIG. 4.8 – Exemple de machine d'état

Définition 4.3.2 (Ressource simple) *Une ressource simple est un sous-réseau de Petri $R = (P, T, F, M_0)$ tel que :*

(i) $P = \{p_{base}\}$, p_{base} *est appelée place de base de la ressource.*

(ii) $T = \{t_{in}, t_{out}\}$, t_{in} *est une transition source et* t_{out} *est une transition puits.*

(iii) $F = \{(t_{in}, p_{base}), (p_{base}, t_{out})\}$.

(iv) M_0 *est le marquage initial du réseau.*

Définition 4.3.3 (Compétence et équipe) *Une compétence ou une équipe est un sous-réseau de Petri* $R = (P, T, F, M_0)$ *tel que* $P = \{p_{base}\}$ *et* $T = F = \oslash$. M_0 *est le marquage initial du réseau et* p_{base} *est appelée place de base de la compétence ou de l'équipe.*

Chaque ressource, équipe et compétence est représentée par le sous-réseau de Petri correspondant. Les jetons dans les places de base représentent la disponibilité d'un certain type de ressource. Chaque jeton désigne une instance de cette ressource.

La synchronisation de machines d'état est réalisée en fusionnant les transitions communes et les sous-réseaux de transitions communs de ces machines d'état (Jeng et DiCesare, 1995; Xie et Jeng, 1999).

Définition 4.3.4 (Sous-réseau de transitions) *Un sous-réseau de transitions* $G_\alpha = (P_\alpha, T_\alpha, F_\alpha, M_{\alpha,0})$ *d'un réseau de Petri* G *est un sous-réseau de* G *tel que* $\forall p \in P_\alpha$, $(\bullet p \cup p\bullet) \in T_\alpha$. *En d'autres termes, les places d'un sous-réseau de transitions sont locales.*

Définition 4.3.5 (Machines d'état synchronisées) *Soit* $C = \{G_i | G_i = (P_i, T_i, F_i, M_{i,0}),\ i = 1 \ldots n\}$ *un ensemble de machines d'état partageant des sous-réseaux de transitions. Un réseau de machines d'état synchronisées* $G = (P, T, F, M_0)$ *de* C *est un réseau tel que* $P = \bigcup_{i=1}^{n} P_i$, $T = \bigcup_{i=1}^{n} T_i$, $F = \bigcup_{i=1}^{n} F_i$, *et* $M_0(p) = M_{i,0}(p)$ *avec* $p \in P_i$.

Exemple 4.3.2 *La figure 4.9 présente la machine d'état de l'exemple précédant synchronisée avec une deuxième machine d'état. Le sous-réseau de transitions encadré en pointillés est commun aux deux machines et permet la synchronisation.*

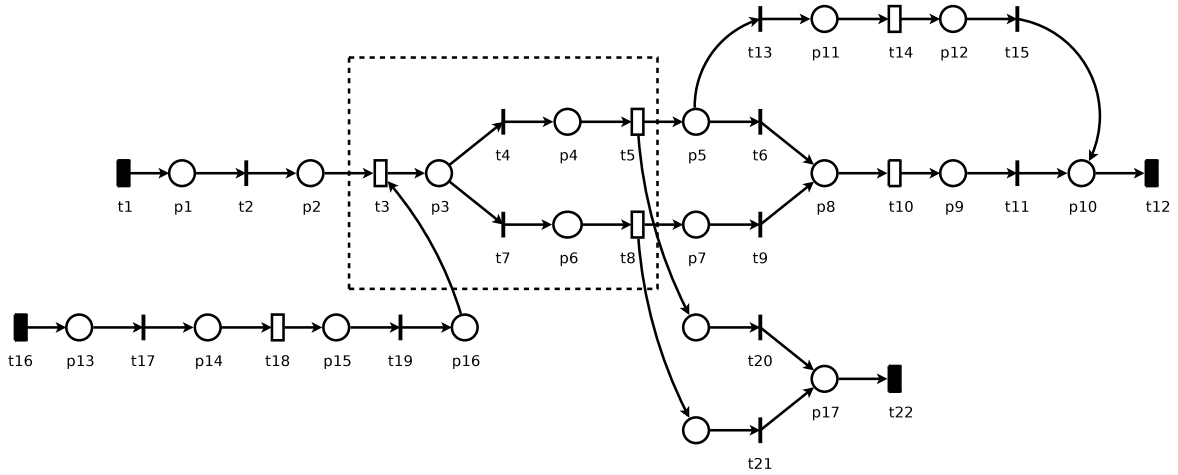


FIG. 4.9 – Exemple de deux machines d'état synchronisées

Définition 4.3.6 (Allocation de ressources) *Soit* G *un réseau de machines d'état synchronisées. L'allocation ponctuelle d'une ressource simple est réalisée en reliant la place de base de cette ressource à une transition de l'ensemble* TE *par un arc aller et un arc retour. L'allocation ponctuelle d'une ressource quelconque est réalisée de la même manière, mais uniquement pour les transitions de l'ensemble* TE *modélisant des états de la vue processus. Enfin, chaque machine d'état correspondant à une mission est liée à la ressource propriétaire de cette mission.*

Exemple 4.3.3 *La figure 4.10 présente la machine d'état de l'exemple précédent à laquelle nous avons ajouté les ressources identifiées par les places de base* $p_{18}, \ldots, p_{23}$. *Dans cet exemple, la ressource* p_{18}

est forcément une ressource simple (pas une équipe ni une compétence) car elle est requise pour un état appartenant à une mission. La mission qui débute par la transition t_{16} et se termine par la transition t_{22} appartient à la ressource p_{23}. Enfin les ressources p_{19}, p_{20} et p_{21} sont des ressources simples et la ressource p_{22} est une compétence.

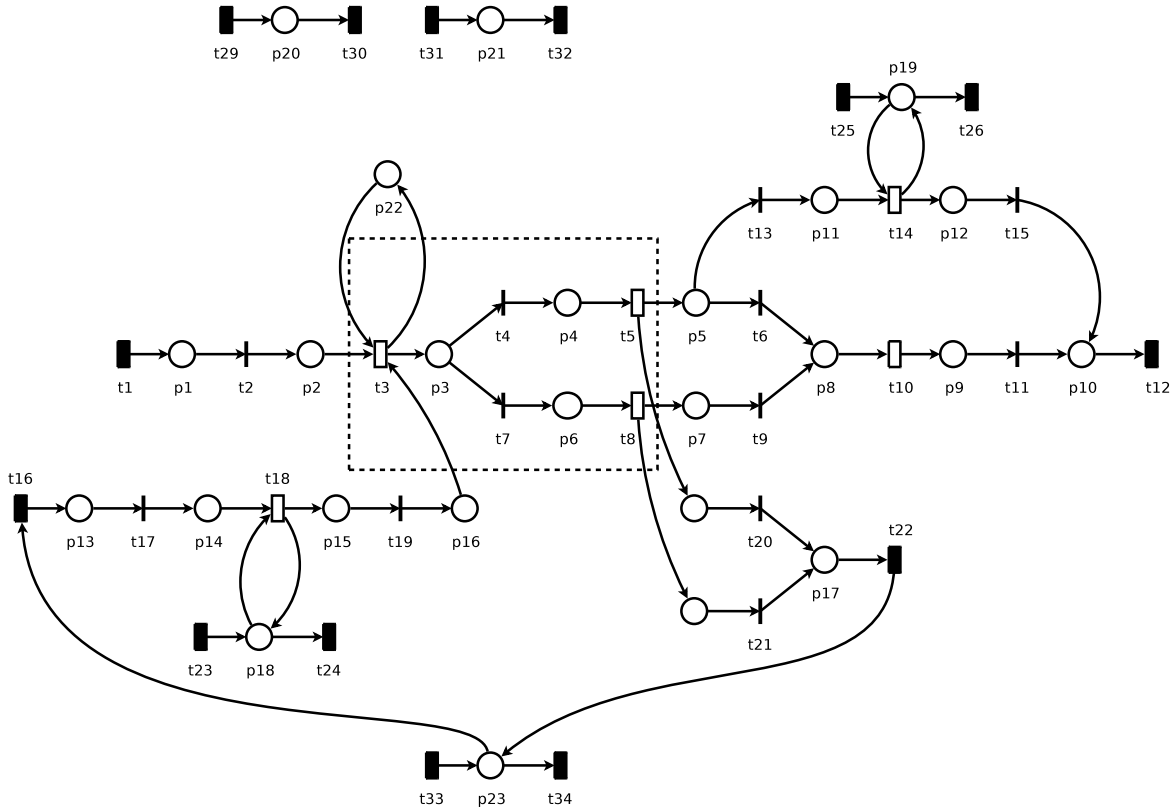


FIG. 4.10 – Allocation de ressources

Définition 4.3.7 (Équipe) *Soit $\{R_i | R_i = (P_i = \{p_{i,base}\}, T_i, F_i, M_{i,0}),\ i = 1, \ldots, n\}$ un ensemble de RdP modélisant des ressources simples. Soit $E = (P' = \{p_{base}\}, T', F', M'_0)$ un RdP modélisant une équipe. Le réseau $G = (P, T, F, M_0)$ modélise l'équipe E constituée des ressources simples $R_1, \ldots, R_n$ si :*

(i) $P = \bigcup_{i=1}^{n} P_i$, $T = \bigcup_{i=1}^{n} T_i$, $F = \bigcup_{i=1}^{n} F_i$, *et* $M_0(p) = M_{i,0}(p)$ *avec* $p \in P_i$.

(ii) $\exists t_1, \ldots, t_n \in T$ *telles que* $(p_{i,base}, t_i) \in F\ \forall i \in \{1, \ldots, n\}$.

(iii) $\exists p_1, \ldots, p_n \in P$ *telles que* $(t_i, p_i) \in F\ \forall i \in \{1, \ldots, n\}$.

(iv) $\exists t_{aller}, t_{retour} \in T$ *telles que* $(p_i, t_{aller}) \in F$ *et* $(t_{retour}, p_{i,base}) \in F\ \forall i \in \{1, \ldots, n\}$. *De plus* $(t_{aller}, p_{base}) \in F$ *et* $(p_{base}, t_{retour}) \in F$.

(v) M_0 *est le marquage initial du réseau.*

Le réseau de Petri ainsi défini est similaire à celui présenté dans la figure 4.4.

Afin de modéliser les relations de compétence et de remplacement/héritage, nous avons besoin d'introduire la notion de couleur. Soit Ω l'ensemble des couleurs considérées. Une couleur spécifique $c(r)$ est associée à chaque ressource simple. Nous avons donc $\Omega = \{c(r_1), \ldots, c(r_n)\}$ avec $r_1, \ldots, r_n$ ressources simples. $c : R \rightarrow \Omega$ est une fonction bijective qui associe à une ressource simple sa couleur. Une définition complète d'un réseau de Petri coloré est donnée dans l'annexe A.

Définition 4.3.8 (Héritage) *Soient R_1 et R_2 deux RdP modélisant des ressources simples, avec $R_1 = (P_1 = \{p_{1,base}\}, T_1, F_1, M_{1,0})$ et $R_2 = (P_2 = \{p_{2,base}\}, T_2, F_2, M_{2,0})$. $G = (P, T, F, C, W^-, W^+, M_0)$ modélise le fait que la ressource simple R_1 hérite de R_2 si :*

(i) $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, $F = F_1 \cup F_2$, *et* $M_0(p) = M_{i,0}(p)$ *avec* $p \in \{P_1, P_2\}$.

(ii) $\exists t_1 \in T$ *telle que* $(p_{1,base}, t_1) \in F$ *et* $(t_1, p_{2,base}) \in F$.

(iii) $\exists t_2 \in T$ *telle que* $(p_{2,base}, t_2) \in F$ *et* $(t_2, p_{1,base}) \in F$.

(iv) $C : (P \cup T) \to \Omega$ *est la fonction qui associe à une place ou à une transition un ensemble de couleurs :*
- $C(p_{1,base}) = c(R_1)$ *et* $C(p_{2,base}) = c(R_2)$.
- $C(t_1) = C(t_2) = \{1\}$.

(v) $W^-_{p,t} : C(t) \to \mathbb{N}^{|C(P)|}$ *est la fonction attachée aux arcs indiquant la pré-condition d'une transition par rapport à une couleur :*
- $W^-_{p_{1,base},t_1}(1) = 1$ *pour la composante* $c(R_1)$, *0 sinon.*
- $W^-_{p_{2,base},t_2}(1) = 1$ *pour la composante* $c(R_1)$, *0 sinon.*

(vi) $W^+_{t,p} : C(t) \to \mathbb{N}^{|C(P)|}$ *est la fonction attachée aux arcs indiquant la post-condition d'une transition par rapport à une couleur :*
- $W^+_{t_1,p_{2,base}}(1) = 1$ *pour la composante* $c(R_1)$, *0 sinon.*
- $W^+_{t_2,p_{1,base}}(1) = 1$ *pour la composante* $c(R_1)$, *0 sinon.*

Le réseau de Petri coloré ainsi défini est similaire à celui présenté dans la figure 4.5.

Définition 4.3.9 (Compétence) *Soit $\{R_i | R_i = (P_i = \{p_{i,base}\}, T_i, F_i, M_{i,0}),\ i = 1, \ldots, n\}$ un ensemble de RdP modélisant des ressources simples. Soit $C = (P' = \{p_{base}\}, T', F', M'_0)$ un RdP modélisant une compétence. Le réseau $G = (P, T, F, C, W^-, W^+, M_0)$ modélise le fait que les ressources simples $R_1, \ldots, R_n$ possède la compétence C si :*

(i) $P = \bigcup_{i=1}^{n} P_i$, $T = \bigcup_{i=1}^{n} T_i$, $F = \bigcup_{i=1}^{n} F_i$, *et* $M_0(p) = M_{i,0}(p)$ *avec* $p \in P_i$.

(ii) $\exists t_{1,1}, \ldots, t_{1,n} \in T$ *telles que* $(p_{i,base}, t_{1,i}) \in F$ *et* $(t_{1,i}, p_{base}) \in F\ \forall i \in \{1, \ldots, n\}$.

(iii) $\exists t_{2,1}, \ldots, t_{2,n} \in T$ *telles que* $(p_{base}, t_{2,i}) \in F$ *et* $(t_{2,i}, p_{i,base}) \in F\ \forall i \in \{1, \ldots, n\}$.

(iv) $C : (P \cup T) \to \Omega$ *est la fonction qui associe à une place ou à une transition un ensemble de couleurs :*
- $C(p_{i,base}) = c(R_i)\ \forall i \in \{1, \ldots, n\}$.
- $C(p_{base}) = \{c(R_1), \ldots, c(R_n)\}$.
- $C(t_{1,i}) = C(t_{2,i}) = \{1\}\ \forall i \in \{1, \ldots, n\}$.

(v) $W^-_{p,t} : C(t) \to \mathbb{N}^{|C(P)|}$ *est la fonction attachée aux arcs indiquant la pré-condition d'une transition par rapport à une couleur :*
- $W^-_{p_{i,base},t_{1,i}}(1) = 1$ *pour la composante* $c(R_i)$, *0 sinon,* $\forall i \in \{1, \ldots, n\}$.
- $W^-_{p_{base},t_{2,i}}(1) = 1$ *pour la composante* $c(R_i)$, *0 sinon,* $\forall i \in \{1, \ldots, n\}$.

(vi) $W^+_{t,p} : C(t) \to \mathbb{N}^{|C(P)|}$ *est la fonction attachée aux arcs indiquant la post-condition d'une transition par rapport à une couleur :*
- $W^+_{t_{1,i},p_{base}}(1) = 1$ *pour la composante* $c(R_i)$, *0 sinon,* $\forall i \in \{1, \ldots, n\}$.
- $W^+_{t_{2,i},p_{i,base}}(1) = 1$ *pour la composante* $c(R_i)$, *0 sinon,* $\forall i \in \{1, \ldots, n\}$.

Le réseau de Petri coloré ainsi défini est similaire à celui présenté dans la figure 4.6.

Exemple 4.3.4 *La figure 4.11 présente la machine d'état de l'exemple précédent à laquelle nous avons ajouté une relation de type compétence et une relation de type héritage. Les ressources simples p_{20} et p_{21} possèdent la compétence p_{22}, tandis que la ressource p_{21} hérite de la ressource p_{19} (i.e. p_{19} peut être remplacée par p_{21}). Les couleurs associées ne sont pas représentées sur la figure.*

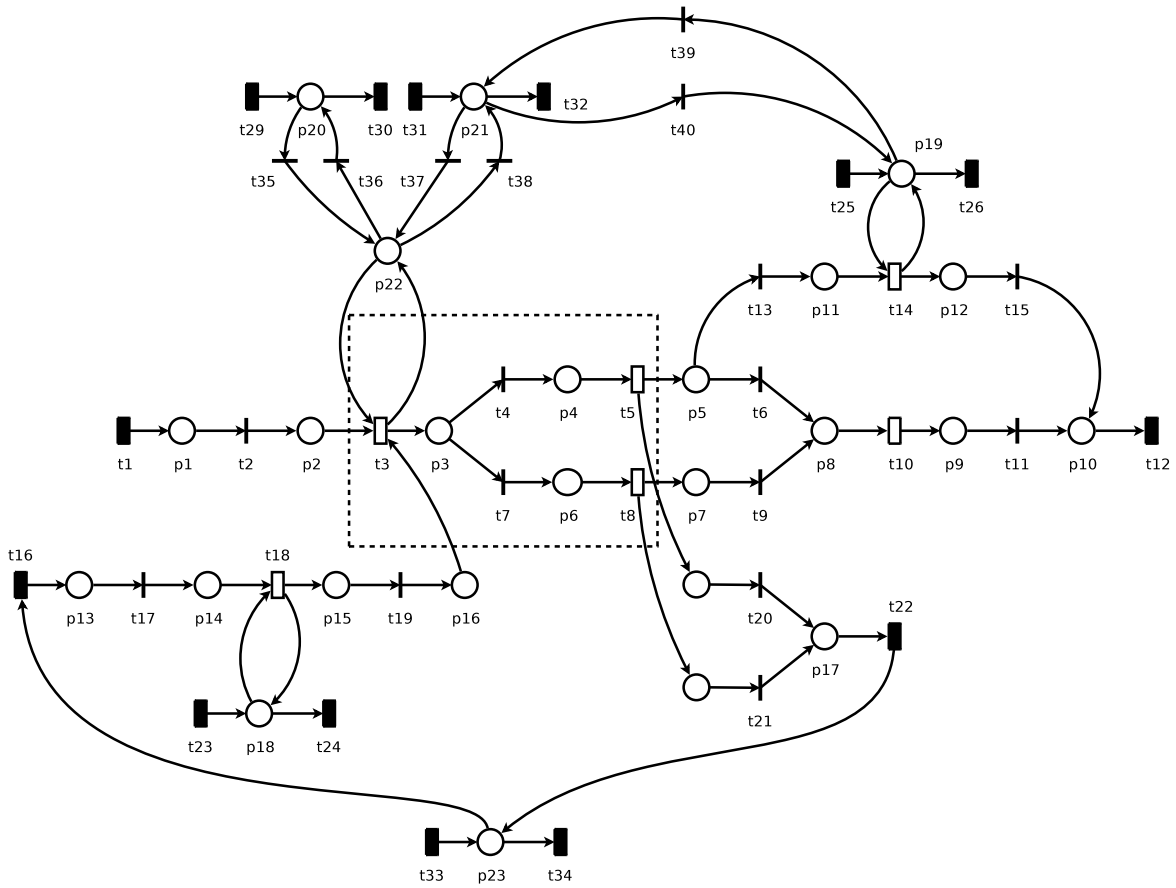


FIG. 4.11 – Relations entre ressources

4.3.2 Réseau de Petri de santé

Nous pouvons ainsi définir un réseau de Petri de santé comme un ensemble de machines d'état synchronisées avec allocation de ressources ; ces ressources sont éventuellement liées par plusieurs types de relations permettant la constitution d'équipes, la définition de compétences partagées et de remplacements. Les réseaux de Petri colorés sont utilisés pour modéliser ces relations. Nous supposons que ces couleurs sont préservées dans les machines d'état considérées.

Définition 4.3.10 (Réseau de Petri de santé) *Un réseau de Petri de santé (RdPS) est une sous-classe d'un réseau de Petri coloré défini comme un 9-uple* $H = (P, T, F, C, W, W^-, W^+, \theta, \xi, M_0)$ *où les hypothèses suivantes sont considérées :*

(i) H *est un réseau de Petri coloré respectant les définitions (4.3.1–4.3.7) données précédemment.*

(ii) $W : F \to \mathbb{N}^*$ *est la fonction poids attachée aux arcs.*

(iii) $\theta : T \to \mathbb{N}$ *est la fonction de temporisation permettant de connaître la durée de franchissement d'une transition.*

(iv) $\xi : T \to \mathbb{E}$ *est la fonction permettant de vérifier la condition de garde liée à une transition.*

(v) M_0 *est le marquage initial du réseau.*

Nous supposons que l'ensemble des jetons qui circulent dans le RdPS possède une liste d'attributs défini dans la vue organisation. Nous supposons également que les ressources synchronisées avec les entités pendant la simulation sont connues à tout moment. Ces attributs sont uniquement manipulés par le système de pilotage.

4.3.3 Programmation de tirs dans un RdPS

La programmation de tirs de transitions est utile pour modéliser le processus de création d'entités ou la prise en compte du planning de ressources.

Définition 4.3.11 (Génération de jetons) *Les jetons sont générés dans les transitions source du RdPS. Soit* $\tau : T_{in} \rightarrow \mathbb{N}$ *la fonction qui associe à une transition source* $t \in T_{in}$ *sa prochaine date de tir* $\tau(t)$.

Remarque 4.3.1 *La fonction* τ *est déduite du schéma de génération défini par l'utilisateur pour chaque entrée de chaque machine d'état du modèle medPRO.*

Définition 4.3.12 (Planning de tir) *Un planning de tir pour un RdPS correspond à la programmation de tirs. Soit* $\tau : T \setminus T_{in} \rightarrow \mathbb{N}$ *la fonction qui associe à toute transition* $t \in T \setminus T_{in}$ *sa date de tirage* $\tau(t)$.

Ce planning de tir est scruté durant la simulation ; toutes les transitions $t \in T \setminus T_{in}$ sont tirées à la date $\tau(t)$ si le marquage courant l'autorise. Sinon, les transitions doivent être tirées dès que possible.

4.4 Exécution d'un réseau de Petri de santé : simulation

4.4.1 Simulation à événements discrets

Selon Law et Kelton (2004a), la simulation à événements discrets se rapporte à la modélisation d'un système dont l'état évolue au cours du temps selon une représentation dans laquelle les variables d'état changent instantanément à certaines dates précises. Les *événements* se produisent à ces dates, un événement étant une occurence instantané susceptible de faire évoluer l'état du système. De par la nature dynamique de la simulation à événements discrets, il est primordial de mettre en place une horloge de simulation permettant de mesurer le temps simulé ainsi qu'un mécanisme capable de faire évoluer ce temps simulé d'une valeur à une autre. En général il n'existe aucun lien entre le temps simulé et le temps nécessaire à l'exécution d'une simulation sur ordinateur. Comme dans la plupart des langages de simulation, nous adoptons l'approche *next-event time-advance* dans laquelle l'horloge est initialisée à 0 et où les dates des événements futurs sont déterminées. L'horloge de simulation est avancée à la date de l'événement le plus urgent de ces futurs événements, date à laquelle l'état du système est mis à jour ; la liste des événements futurs est également mise à jour, et ainsi de suite. Ce processus est itéré jusqu'à ce qu'une condition d'arrêt soit rencontrée ou jusqu'à ce que la liste d'événements en attente soit vide. Cette méthode permet de sauter les périodes d'inactivité du système, les sauts étant généralement inégaux en taille.

Un algorithme de simulation à événements discrets (Law et Kelton, 2004a) a été adapté pour le réseau de Petri de santé défini précédemment. Bien que la simulation trouve des applications dans un grand nombre de systèmes, les modèles de simulation a événements discrets partagent un certain nombre de composants commun organisés de manière logique. Cette organisation offre une grande souplesse pour la programmation, le debuggage et l'évolution du modèle. Les composants suivants en particuliers ont un sens dans le contexte de la plate-forme medPRO :

État du système : l'état du système est modélisé par un ensemble de variables d'état permettant de décrire le système à n'importe quel moment. Dans notre cas il s'agit du marquage du réseau.

Horloge de simulation : variable t_{now} permettant de connaître la valeur du temps simulé.

Liste d'événements : liste contenant les prochains événements classés par ordre chronologique. Dans notre cas, un événement est le début ou la fin du tir d'une transition.

Compteurs statistiques : ensemble de variables utilisées pour enregistrer des informations statistiques sur les performances du système.

Procédure d'initialisation : fonction chargée d'initialiser le modèle de simulation à la date $t_{now} = 0$.

Procédure de timing : fonction permettant de déterminer le prochain événement à traiter dans la liste d'événements. L'horloge de simulation est ensuite mise à jour.

Procédure de traitement : fonction chargée de mettre à jour l'état du système lorsqu'un événement particulier a lieu.

Procédure de génération de variables aléatoires : fonction permettant la génération de variables aléatoires à partir de distributions de probabilité définies dans le modèle de simulation.

Générateur de rapport : fonction permettant le calcul, le regroupement et la mise en forme de rapports lorsque la simulation est terminée.

Programme principal : le programme principal invoque la procédure de timing pour déterminer le prochain événement et laisse la main à la procédure de traitement pour mettre à jour correctement l'état du système. Le programme principal vérifie également la condition de terminaison et appelle le générateur de rapport lorsque la simulation est terminée.

Les relations logiques entre ces composants sont présentées figure 4.12. L'algorithme présenté dans cette figure a été adapté pour la simulation d'un réseau de Petri de santé et sera détaillé comme tel.

La simulation commence à la date 0 lorsque le programme principal appelle la procédure d'initialisation : l'horloge de simulation t_{now} est initialisée à 0, l'état du système et les compteurs statistiques sont initialisés. La procédure de timing est ensuite appelée pour déterminer le prochain événement à traiter correspondant au début ou à la fin du franchissement d'une transition. L'horloge de simulation est avancée à cette date et le programme principal appelle la procédure de traitement correspondant à l'événement courant ; l'état du système et les compteurs statistiques sont alors mis à jour et le nouveau marquage du réseau est calculé. La génération de variables aléatoires assurée par une procédure dédiée est parfois requise durant ce traitement. Lorsque le traitement de l'événement est terminé, une vérification est réalisée pour savoir si la simulation est terminée. Dans l'affirmative le générateur de rapport est appelé afin de réunir les indicateurs de performances dans un rapport. Dans le cas contraire, une nouvelle itération est entreprise avec l'appel de la procédure de timing.

Nous détaillerons dans cette section les différentes étapes de cet algorithme de simulation dans le cadre de la plate-forme de modélisation et de simulation medPRO.

4.4.2 Initialisation

La phase d'initialisation de la simulation est composée de trois étapes distinctes :

1. Initialisation de l'horloge de simulation ;
2. Initialisation de l'état du système et des compteurs statistiques ;
3. Initialisation de la liste d'événements.

Définition 4.4.1 (Horloge de simulation) *L'horloge de simulation est un entier $t_{now} \in \mathbb{N}$ correspondant au temps simulé.*

Le temps simulé n'a pas de relation définie avec une quelconque unité de temps connue. Cependant, pour une meilleure compréhension nous fixons la valeur d'une unité de temps simulé à une seconde. Par exemple $t_{now} = 3600$ dans le système simulé correspond à une heure dans le système réel.

Définition 4.4.2 (État du système) *L'état du système à la date t_{now} correspond à un certain marquage $M^{t_{now}}$ du RdPS.*

Le marquage initial M_0 du système est généralement nul étant donné que l'ensemble des objets circulant dans le réseau (entités et ressources) sont générées au travers de transitions sources activées selon un planning de génération prédéfini.

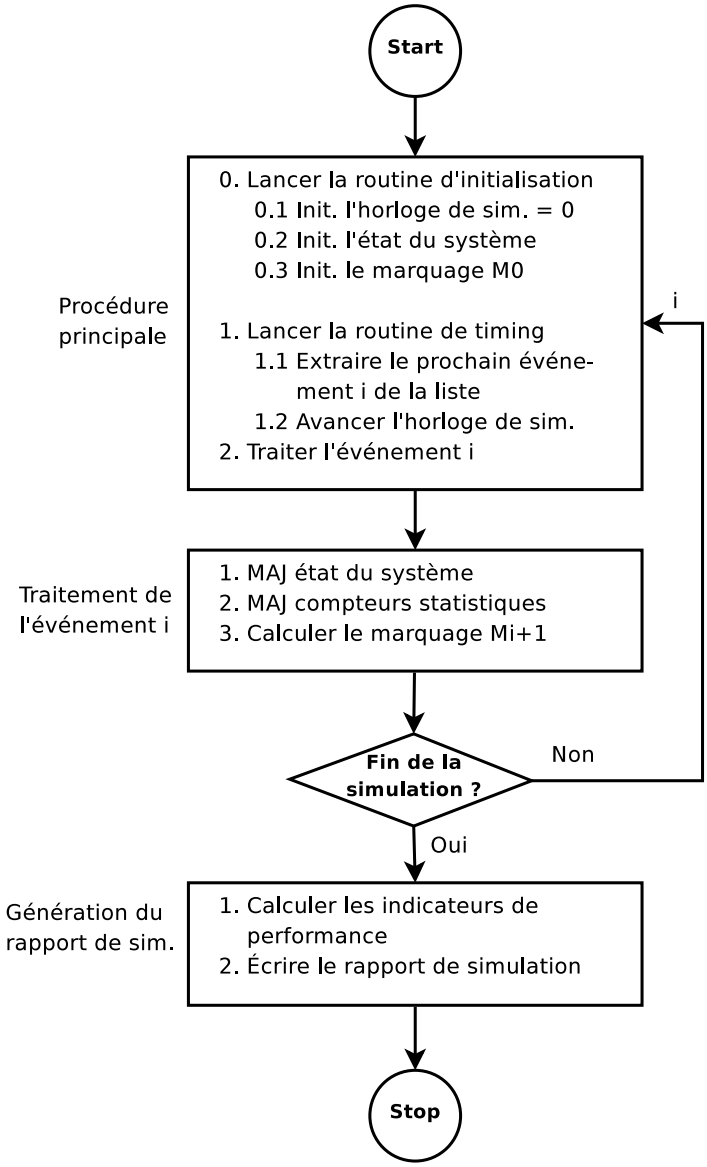


FIG. 4.12 – Algorithme de simulation

Définition 4.4.3 (Événement) *Un événement est un triplet $(d, t, type)$ tel que :*

- *$d \in \mathbb{N}$ est la date de l'évènement ;*
- *$t \in T$ est la transition dont le franchissement débute ou se termine ;*
- *type est le type d'événement,* $type = \begin{cases} 1 & \text{si l'événement est une fin de tir,} \\ 0 & \text{si l'événement est un début de tir.} \end{cases}$

Définition 4.4.4 (Liste d'événements) *La liste d'événements LE regroupe les transitions dont les dates de début ou de fin de franchissement sont connue(s). Les évènements sont classés par ordre de priorité.*

La liste d'évènement doit être vue comme un planning de tir généralisé, permettant de connaître les dates auxquelles les jetons sont retirées et ajoutés aux places du RdPS. L'algorithme 4.4 présente la procédure utilisée pour initialiser la liste d'événements avant la simulation. Chaque transition source du RdPS est examinée et sa date de premier tir est ajoutée à la liste d'évènements.

Algorithme 4.4 Algorithme d'initialisation de la liste d'événements

```
1   pour chaque transition source t_in ∈ T_in faire
2       Ajouter (τ(t_in), t_in, 1) à la liste d'événements
3   fin pour
```

Définition 4.4.5 (Liste d'attente) *La liste d'attente LA regroupe les transitions qui doivent être tirées dès que possible, classées par ordre de priorité.*

Les événements de la liste d'attente sont toujours du type $(d, t, 0)$. La faisabilité de l'ensemble des événements de la liste LA doit être testée avant chaque traitement d'un nouvel événement.

4.4.3 Timing

La procédure de timing permet de déterminer la date du prochain événement à traiter et d'avancer l'horloge de simulation à cette date. Nous avons défini deux types d'événements dans la section précédente, à savoir le début ou la fin du tir d'une transition. La détermination du prochain événement dépend de la liste d'événements d'une part, et d'autre part du marquage du réseau (i.e. de l'état du système). L'algorithme 4.5 décrit la procédure de timing utilisée dans medPRO pour déterminer le prochain évènement à traiter. La valeur de l'horloge de simulation est t_{now}. Nous supposons qu'il n'existe aucune transition tirable à la date t_{now}.

Algorithme 4.5 Algorithme de la procédure de timing

```
1   extraire l'événement (d, t, type) en tête de la liste d'événements
2   t_now ← d
```

La procédure de timing est très simple dans notre cas car nous avons seulement défini deux types d'événements, à savoir le début ou la fin d'un franchissement. Le test de faisabilité et la mise en œuvre du franchissement seront traités dans la procédure de traitement d'événement.

4.4.4 Traitement d'un événement

La procédure de traitement permet de mettre à jour le marquage du réseau en appliquant les effets d'un événement. Ce dernier a été extrait de la liste d'événements grâce à la procédure de timing, permettant par la même occasion d'avancer l'horloge de simulation. La procédure de traitement se décompose en plusieurs phases distinctes :

1. Test de la faisabilité et détermination des conditions d'exécution de l'événement par le sous-système de pilotage ;
2. Mise à jour du réseau ;
3. Mise à jour des compteurs statistiques ;
4. Prise de décision concernant les jetons dans les centres de décision par le sous-système de pilotage ;
5. Ajout des nouveaux événements à la liste d'événements générés par le traitement de l'événement courant.

La première phase traitée par le sous-système de pilotage consiste à tester la faisabilité d'un franchissement et à déterminer le type de ce franchissement. Ces opérations seront décrites en détail dans la chapitre suivant. Le réseau est ensuite mis à jour en procédant au retrait et/ou à l'ajout de jetons dans les places appropriées. Les compteurs statistiques sont également mis à jour. Le sous-système de pilotage

est à nouveau invoqué pour la prise de décision concernant les jetons qui se trouvent dans un centre de décision du réseau. Enfin, les événements résultant du traitement de l'événement courant sont ajouté à la liste globale.

Procédure de traitement d'un événement

L'algorithme 4.6 décrit plus formellement la procédure de traitement en fonction du type d'événement. La valeur de l'horloge de simulation est t_{now}. Nous considérons que l'événement à traiter s'écrit $(d, t, type)$. Le marquage avant traitement est M_i et le marquage après traitement est M_{i+1}.

Algorithme 4.6 Algorithme de traitement d'un événement

```
1   si type = 0
2       tester la faisabilité et déterminer les conditions du franchissement de t
        (ssp)
3       si t est tirable
4           pour tout p ∈ •t faire
5               M(p) ← M(p) − W(p, t)
6           fin pour
7           ajouter((t_now + θ(t), t, 1),LE)
8       sinon
9           ajouter(t,LA)
10      fin si
11  sinon
12      pour tout p ∈ t• faire
13          M(p) ← M(p) + W(t, p)
14          déterminer la prochaine transition t' ∈ p° à tirer ainsi que sa date de
            tir d' (ssp)
15          ajouter((d', t', 0),LE)
16      fin pour
17      si t ∈ T_in
18          ajouter((τ(t_in), t_in, 1,LE)
19      fin si
20  fin si
```

L'algorithme de traitement d'un événement se décompose en deux parties distinctes : le traitement du début d'un franchissement (lignes 1–10) et la traitement de la fin d'un franchissement (11–20). Dans le premier cas ($type = 0$), la faisabilité du tir est testée grâce à un appel au sous-système de pilotage (ligne 2). Si la transition t est tirable, le marquage de chaque place précédant t est mis à jour (lignes 4–6) et l'événement correspondant à la fin du franchissement de t est ajouté à la liste d'événéments (ligne 7). Si t n'est pas tirable, elle devra être tirée dès que possible (lignes 8–9) Dans le deuxième cas ($type \neq 0$), le marquage de chaque place p succédant à t est mis à jour (ligne 13) ; le sous-système de pilotage est ensuite invoqué pour déterminer quelle sera la prochaine transition à tirer parmi les successeurs de p (ligne 14), et l'événement correspondant au tir de la transition choisie est ajouté dans la liste d'événements (ligne 15). La dernière étape de l'algorithme consiste à générer les événements correspondant aux prochains tirs des transitions source du réseau (lignes 17–19).

Pré-franchissement et post-franchissement

D'après l'algorithme 4.6, un appel au sous-système de pilotage est réalisé avant et après le franchissement de toute transition du RdPS simulé, constituant deux traitements spécifiques. Le sous-système de pilotage, également appelé modèle de contrôle, est synchronisé avec le sous-système physique (ou modèle d'émulation) en amont et en aval de chaque transition. Cette technique décrite plus avant dans le chapitre suivant permet de contrôler l'exécution de la simulation par le biais de modules de pilotage ou d'optimisation. Les tâches liées à une synchronisation pré- ou post-franchissement sont diverses :

Pré-franchissement : la faisabilité du tirage d'une transition doit être vérifiée ; si la transition est tirable, les ressources nécessaires sont réunies et le franchissement peut commencer.

Post-franchissement : à l'issu d'un franchissement, il est nécessaire de déterminer quelles seront les prochaines transitions à tirer, entraînant l'évaluation de conditions liées à des transitions, le tirage de variables aléatoires, etc.

La simulation est considérée terminée si la liste d'événements est vide à l'issu de la procédure de traitement ou si la date de fin de simulation fixée par l'utilisateur est atteinte. Dans le cas contraire la procédure de timing est à nouveau appelée est le processus est ré-itéré pour le prochain événement à traiter.

Mise à jour des compteurs statistiques

Outre les variables définies par l'utilisateur permettant de mesurer certains indicateurs de performances spécifiques au système étudié, plusieurs types de compteurs statistiques sont automatiquement mis à jour lors de la simulation :

Compteurs de tir : le nombre de franchissement de chaque transition du réseau est mesuré, permettant de déterminer les états goulots du système ainsi que les branchements conditionnels les plus fréquentés. Le nombre d'entités et de ressources générées et détruites est également comptabilisé.

Taux de fréquentation : le temps passé dans chaque état (i.e. dans chaque place et transition du réseau) pour chaque type d'entité ou de ressource est mesuré, permettant la déduction de taux d'utilisation. Nous pouvons en déduire le taux d'activité de chaque ressource durant ses horaires de travail.

Durée de séjour : la durée de séjour totale du patient (de sa création à sa destruction) est systématiquement mesurée.

Selon les cas d'études, d'autres indicateurs de performance pourront être mesuré et automatiquement reportés grâce aux modules prédéfinis dans la plate-forme medPRO. Nous définissons par exemple dans le cas de l'étude des délais d'obtention de rendez-vous pour des examens médicaux deux indicateurs de performance afin de mesurer pour chaque examen le délai et l'urgence du cas.

4.4.5 Génération de variables aléatoires

La « génération d'une variable aléatoire » consiste à obtenir une observation d'une variable aléatoire à partir de la distribution désirée. Nous ne nous étendrons pas sur le sujet dans ce mémoire étant donné que ce problème a largement été traité dans (Law et Kelton, 2004b). Le lecteur pourra s'y rapporter pour consulter l'ensemble des méthodes permettant la génération de variables aléatoires, par ailleurs largement utilisées dans la plupart des outils de simulation du marché.

4.4.6 Génération du rapport de simulation

La génération du rapport de simulation est réalisée de la même manière que dans le logiciel Arena : l'ensemble des indicateurs de performance spécifiés par l'utilisateur ainsi que les indicateurs par défauts

sont regroupés et présentés sous forme de tables dans un rapport formaté selon un modèle prédéfini. Nous ne décrirons pas plus avant les algorithmes permettant le recueil des indicateurs ainsi que les algorithmes de mise en forme car il relève plus de l'ingénierie de développement logiciel que de la formalisation scientifique de la simulation décrite dans ce chapitre.

4.5 Synthèse

La conversion en réseau de Petri du modèle UML ouvre de nouveaux horizons : l'analyse des réseaux de Petri permet notamment la vérification de propriétés qualitatives du point de vue des systèmes hospitaliers indépendamment de leurs propriétés quantitatives. De nombreuses applications d'origine industrielle sont présentées dans la littérature et peuvent être transposées aux systèmes hospitaliers. Ces propriétés se décomposent en deux catégories : les *propriétés comportementales* et les *propriétés structurelles*. Les premières dépendent à la fois de la structure du RdP et du marquage initial. Les secondes dépendent uniquement de la structure du RdP. Nous sommes ainsi en mesure de proposer une structure basée sur les réseaux de Petri permettant de détecter rapidement et automatiquement des problèmes organisationnels d'origine structurelle.

Nous avons également vu dans ce chapitre que la conversion en réseau de Petri permet une formalisation mathématique de l'exécution du modèle via la simulation à événements discrets. L'algorithme proposé permet de faire évoluer le réseau en interaction avec le système de pilotage avec une grande souplesse (l'état du système est modélisé grâce à aux marquages successifs du réseau). Cette approche permet de conserver une trace du déroulement de la simulation sous forme d'une séquence de franchissements : les choix réalisés au cours de la simulation peuvent être analysés et éventuellement corrigé. Le relevé et la mise en forme d'indicateurs de performance sont réalisés très simplement au terme de la simulation.

Les propriétés des réseaux de Petri sont également utiles pour proposer une approche cohérente de la planification à court terme et de l'ordonnancement. Nous verrons dans le chapitre suivant que les systèmes hospitaliers modélisés au sein de la plate-forme medPRO peuvent être assimilés à des systèmes acycliques appelés à répondre à des demandes exogènes qui évoluent dans le temps. Nous proposerons en particulier plusieurs méthodes de planification et d'ordonnancement adaptées aux systèmes hospitalier.

Les deux principaux défauts de la méthode d'analyse proposée dans ce mémoire résident dans la complexité de la conversion et dans la taille des réseaux obtenus en sortie : un modèle UML entraîne la création d'un réseau contenant approximativement trois fois plus de places et de transitions que d'états ; la mise en place des règles d'affectation, de libération et de synchronisation de ressources complexifie un peu plus le réseau. L'exécution des algorithmes de conversion reste cependant rapide en supposant que la taille des modèles UML reste raisonnable. Le problème de la taille et de la complexité des réseaux générés n'est pas un réel problème non plus, car ces derniers restent invisibles du point de vue de l'utilisateur. Le réseau de Petri généré n'existe que pour servir de support à l'analyse de propriétés qualitatives, à la simulation et au pilotage. Enfin, l'application de méthodes de réduction ou de modélisation modulaire (Proth *et al.*, 1997) peuvent être appliquées en réponse à ces critiques.

Chapitre 5

Système de décision

Ce chapitre est consacré à la description de la structure décisionnelle utilisée au sein de la plate-forme medPRO. Nous nous intéressons dans un premier temps à l'application de techniques de planification à court terme et d'ordonnancement issues du milieu industriel. Ces techniques ont été adaptées aux réseaux de Petri de santé définis dans le chapitre précédent afin de proposer des outils d'optimisation génériques exécutables en amont de la simulation. Nous décrivons ensuite la structure de pilotage en temps réel : une approche hybride hiérarchique/hétérarchique est mise en œuvre au travers d'un module de contrôle en interaction avec le réseau de Petri modélisant le système. Le principe du pilotage repose sur un échange de données entre le modèle d'émulation (flux représentés dans le système physique) et le modèle de contrôle durant la simulation. Plusieurs tâches sont affectées au système de pilotage : sélection de ressources, prise de décision, branchement conditionnel. Deux modules spécifiques permettant la modélisation de processus de prise de décision propres au milieu médical sont également présentés.

5.1 Introduction

Le système de pilotage et de décision intervient pendant le fonctionnement du système (simulé ou réél) et permet de contrôler son évolution. Une telle politique est fréquemment utilisée dans les milieux hospitaliers pour réagir aux aléas. La mise en œuvre d'un système de pilotage permet également de séparer les domaines opérationnels et décisionnels afin de gagner en lisibilité et d'accélérer le processus de test de scénarios de simulation.

Le système de pilotage est structuré sur la base d'un ensemble de modules permettant la modélisation de processus de décision particuliers, certains d'entre eux étant spécifiques aux systèmes hospitaliers. L'ajout et/ou la modification de modules permet la programmation de comportements spécifiques à un système.

Afin de présenter la structure décisionnelle adoptée au sein de la plate-forme medPRO, nous proposons dans un premier temps une application des approches de la planification à court terme et de l'ordonnancement adaptés aux systèmes de production (Proth et Xie, 1995b) à certains types de systèmes hospitaliers. La planification permettra l'anticipation des différentes actions liées à un projet (les interventions programmées dans le bloc opératoire par exemple) ; l'ordonnancement permettra ensuite d'organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, précédence) et de contraintes portant sur la disponibilité des ressources requises. Ces deux approches sont indépendantes et peuvent s'appliquer dans des conditions différentes. Soulignons tout de même que le modèle utilisé pour l'ordonnancement s'obtient en complétant celui de la planification à court terme, ce qui confirme que les réseaux de Petri autorisent une approche cohérente du couple planification-ordonnancement.

Dans un second temps, une approche permettant de cumuler les bénéfices des architectures hiérarchique

et hétérarchique pour le pilotage en temps réel de systèmes de soins est décrite. Les modules de contrôle se répartissent en trois catégories distinctes : les traitements pré-franchissement, les traitements post-franchissement, et les traitements spécifiques au milieu hospitalier. L'approche modulaire proposée permet le remplacement d'un système de pilotage par un autre sans modifier le modèle physique.

Le système de pilotage en temps réel est également destiné à contrôler la mise en œuvre du planning obtenu à l'issu de la première phase : l'application de techniques MES (*Manufactufing Execution System*) est envisagée pour permettre l'ajustement du planning théorique durant l'exécution de la simulation.

Le plan de ce chapitre est le suivant : après une brève revue de littérature décrivant les différentes architectures de pilotage et quelques une de leurs applications (section 5.2), l'application des techniques de planification et d'ordonnancement sont décrites section 5.3. Le principe du système de pilotage en temps réel ainsi que son intégration pendant la simulation sont détaillés section 5.4. La section 5.5 fait la synthèse des techniques présentées dans ce chapitre et discute de l'utilisation et de l'intérêt de telles méthodes lorsqu'elles sont appliquées dans le domaine hospitalier.

5.2 Structures décisionnelles et architectures de pilotage

Malgré le fait que la simulation à événements discrets est souvent recommandée pour la représentation de systèmes de production complexes, son utilisation présente quelques inconvénients : la phase de modélisation d'un projet de simulation est excessivement lourde, ce qui entraîne des coûts de développement importants. De plus un modèle unique est très sensible au moindre changement, notamment dans les aspects de contrôle : la mise en œuvre de politiques de pilotage pour différents scénarios de simulation implique un investissement très couteux en temps. Des outils de simulation très performants (tel Arena) qui conviennent parfaitement pour simuler les aspects physiques du système de production ne possèdent aucune structure de modélisation adaptée pour la représentation de stratégies de pilotage.

Le système de décision est généralement issu d'une décomposition systémique du système de production considéré (c.f. section 2.3.2). Trois grandes structures décisionnelles existent dans la littérature :

Hiérarchie : Les systèmes de décision des entreprises sont traditionnellement structurés en hiérarchie, permettant aux acteurs de haut niveau du système de prendre des décisions avec une vision globale. Le modèle de planification le plus couramment utilisé est basé sur la structure à cinq étapes MPCS (*Manufacturing Planning and Control System*) du MRP2 (Vollman *et al.*, 1988), où les cadres de décision sont de plus en plus contraints à mesure que l'horizon diminue. Les décisions sont stratégiques ou opérationnelles. L'optimalité de la solution est assurée, mais lorsqu'un événement imprévu se produit au niveau le plus bas, l'information doit être remontée aux niveaux supérieurs afin de fixer un nouveau cadre de modélisation, ce qui réduit la réactivité du système.

Hétérarchie : Le pilotage hétérarchique consiste à distribuer l'intégralité du pouvoir de décision : les entités sont indépendantes et évoluent dans un environnement qu'elles perçoivent et sur lequel elles peuvent agir, mais aucune ne possède une vue globale du système de production. L'approche hétérarchique la plus classique repose sur le concept des systèmes multi-agents (SMA). Chaque décision est prise de manière locale, ce qui constitue par ailleurs le défaut majeur d'une telle approche : l'optimalité de la solution n'est pas assurée. De plus l'absence de coordination peut mener à un blocage du système.

Holarchie : L'approche holonique, proposée par Valckenaers *et al.* (1997) pour les systèmes manufacturiers, permet de mêler les deux approches précédentes. En se basant sur le concept de holon défini par Koestler (1989), les auteurs proposent de définir un ensemble de sous-systèmes stables et hiérarchisés, dont les caractéristiques sont l'autonomie et la coopération. Cette approche est mieux adaptée pour la gestion et le pilotage des systèmes de production au niveau opérationnel que l'approche purement hétérarchique, qui convient mieux à la gestion d'entreprises au niveau tactique.

Les systèmes holarchiques proposent en effet de combiner la stabilité de systèmes hiérarchiques et la flexibilité des systèmes hétérarchiques, tirant ainsi avantage de ces deux structures.

La problématique actuelle des architectures hybrides réside dans le manque de validation industrielle : beaucoup de travaux de recherche ont été menés et validés sur des plate-formes de laboratoire ou par la simulation uniquement et les applications industrielles sont rares. Klein et Thomas (2006) proposent une architecture de simulation permettant de distinguer la structure de pilotage de la partie d'émulation du système physique dans le but de comparer plusieurs modèles de pilotage. Une structure de pilotage hybride hiérarchique/hétérarchique est utilisée et appliquée sur le cas réel d'un fabriquant de meubles en kits. L'originalité de cette étude réside dans le développement d'un modèle de contrôle interfaçable avec Arena permettant la résolution de problèmes de planification en relation avec le problème industriel traité. Une approche similaire est proposée dans (Haouzi et Thomas, 2005), où une approche de simulation de systèmes de production avec contrôle distribué basée sur la méthodologie ADSI (*Analysis Specification Design Implementation*) est proposée (Kellert et Ruch, 1998).

Blanc *et al.* (2006) optent pour une approche multi-agents reposant sur une structure decisionnelle holarchique afin de résoudre les problèmes d'hétérogéneité des modèles, et de distribution du pilotage. Cette approche est appliquée à un cas industriel ; chaque agents ressource est en charge du système de décision d'une ressource qui consiste principalement à réaliser l'ordonnancement des opérations sur celle-ci. Plusieurs problèmes d'optimisation liés au processus industriel sont résolus de cette manière. L'évaluation du système de pilotage se fait en utilisant la simulation à événements discrets. L'outil combinant Arena avec un système multi-agents doit permettre la capitalisation de la connaissance du modèle pour sa réutilisation pour le pilotage du système réel.

Nous privilégions une approche hybride hierarchie/hétérarchie pour notre propre système de décision, et nous plaçons donc dans la lignée des travaux de Klein et Thomas (2006).

5.3 Planification et ordonnancement

5.3.1 Position du problème

Plaçons nous dans le cas d'un système hospitalier quelconque. Les patients entrent dans ce système à cause d'un problème de santé et suivent un processus de prise en charge pour être soignés (intervention chirurgicale, séjour dans une unité de soins, examens médicaux). À la différence d'un système de fabrication capable de fournir plusieurs types de produits, un système hospitalier ne fait correspondre qu'une seule sortie à chacune de ses entrées.

Nous définissons H périodes élémentaires durant lesquelles un certain nombre de patients pourront être soignés dans le système. Pour fixer les idées, une période élémentaire peut être une journée et H peut être fixé à 5 : nous serons alors dans le cas d'un système hospitalier géré au quotidien sur un horizon d'une semaine ouvrable, c'est à dire cinq jours. Comme pour les systèmes manufacturiers, nous supposons que la durée d'une période élémentaire est largement supérieure à la durée de prise en charge d'un patient quel qu'il soit.

Comme pour la fabrication d'un produit, la prise en charge d'un patient se caractérise par les examens et les interventions (les opérations) effectuées par le personnel hospitalier (les ressources) et les temps nécessaires pour la réalisation de ces tâches. Nous connaissons la durée T de chaque période élémentaire : dans le cas d'un bloc opératoire, la période élémentaire correspond à la durée d'ouverture du bloc pendant la journée.

Le problème consiste à décider du nombre de patients à prendre en charge durant chaque période élémentaire en tenant compte de la capacité globale du système. Il s'agit ensuite de décider, à l'intérieur de chaque période élémentaire, des ressources qui vont exécuter les différentes opérations et des instants

de début de ces opérations :

1. **Planification à court terme :** affectation des patients à chacune des H périodes élémentaires en se basant sur les capacités globales des ressources sans tenir compte des ordres de passage. La durée de travail utilisée pour la planification dans cette phase est inférieure à T, offrant une plus grande souplesse pour l'ordonnancement.
2. **Ordonnancement :** affectation des opérations aux ressources. L'objectif est de trouver un ordonnancement permettant la prise en charge des patients planifiés lors de la phase précédente. Si aucun ordonnancement n'a pu être trouvé à l'issu de cette phase, une nouvelle planification est réalisée en réduisant la durée de travail de chaque période élémentaire.

Les critères habituellement considérés dans le milieu hospitalier pour la planification sont :
- la minimisation du retard par rapport aux préférences du patient ;
- la minimisation des heures supplémentaires ;
- la minimisation du temps total d'occupation du système hospitalier.

Les procédures permettant la planification et l'ordonnancement sont décrites dans les sections 5.3.4 et 5.3.5 respectivement. Les hypothèses considérées sont présentées dans la section 5.3.2.

5.3.2 Hypothèses préalables

Afin d'appliquer les méthodes de planification à court terme et d'ordonnancement à un réseau de Petri de santé, plusieurs hypothèses doivent être considérées :
- H_1 : le réseau de Petri considéré est acyclique. Un système hospitalier est effectivement comparable à un système acyclique dans la mesure où un tel système est appelé à répondre à des demandes exogènes qui évoluent dans le temps.
- H_2 : un chemin unique est considéré pour chaque mission déclarée dans la vue ressource : les branchement conditionnels sont évalués à l'avance pour chaque patient.
- H_3 : les allocations ponctuelles de ressources dans des R-états non-synchronisés ne sont pas prises en compte.

D'une manière générale, nous supposons que tous les branchements conditionnels sont connus à l'avance pour chaque entité et dans chaque machine d'état. Les activités définies exclusivement pour des ressources (R-états non-synchronisés) ne sont pas prises en compte durant la phase d'ordonnancement.

5.3.3 Modélisation du point de vue de la planification et de l'ordonnancement

Décomposition en sous-réseaux de Petri contrôlables

Soit H un réseau de Petri de santé composé d'un ensemble de machines d'état associées aux flux de la vue processus et d'un ensemble de machines d'états associées aux missions de la vue ressource. Afin d'illustrer nos propos, nous nous baserons sur l'exemple présenté figure 5.1 : ce réseau comporte une machine d'état associée à la vue processus et une machine d'état associée à l'unique mission de la ressource `R1`.

Seul le réseau de Petri correspondant à la vue processus (c.f. définition 4.3.1) est utilisé pour la planification et l'ordonnancement : il s'agit d'une machine d'état particulière modélisant uniquement le parcours patient. Soit $M = (P, T, F, M_0)$ ce réseau. Soit $TE \subset T$ l'ensemble des transitions associées à des états. Nous appellons $r_1, \ldots, r_m$ les ressources simples utilisées. Nous associons à chaque transition $t \in TE$ de ce réseau les combinaisons de ressources requises sans tenir compte dans un premier temps des éventuelles missions et/ou allocations ponctuelles de ressources.

Les combinaisons de ressources sont déterminées de la manière suivante : une compétence requise est remplacée par la liste de ressources simples possédant cette compétence séparées par des « ou » ; une

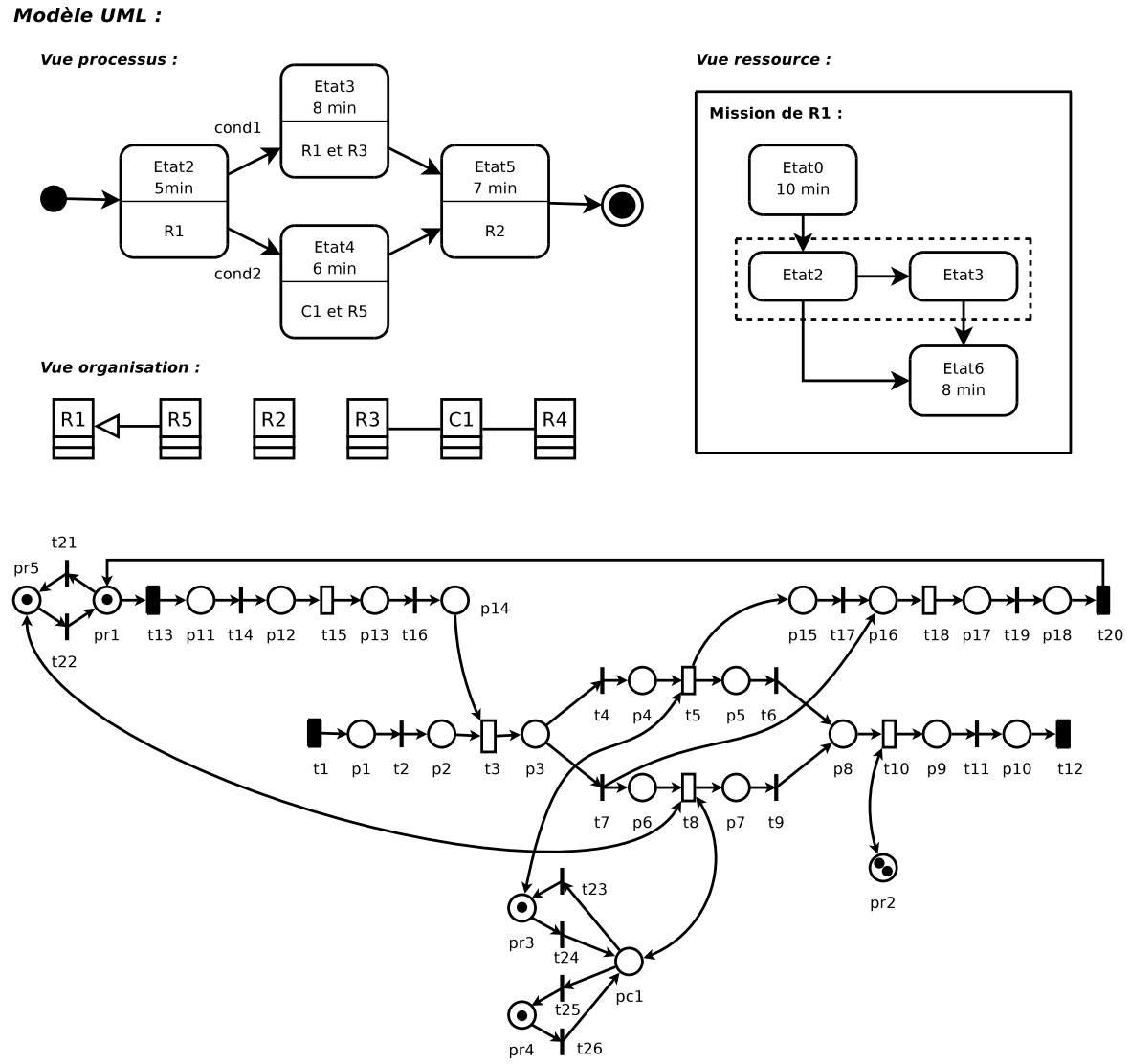


FIG. 5.1 – Un réseau de Petri de santé pour la planification

équipe est remplacée par la liste des ressources simples figurant dans cette équipe séparées par des « et » ; une ressource simple possédant un ou plusieurs remplaçant(s) est remplacée par la liste constituée de cette ressource et de ses remplaçants séparés par des « ou ». L'expression ainsi obtenue pour chaque transition ne contient plus que des ressources simples et est développée selon l'opérateur « ou ».

La figure 5.2 présente l'exemple de l'état `Etat` qui nécessite une ressource possédant la compétence `Comp1`, la ressource simple `R1` et la ressource simple `R2` qui peut être remplacée par `R5`. L'expression s'écrit alors $c_1 \wedge r_1 \wedge r_2$. Selon les relations de la vue organisation, nous pouvons réécrire l'expression : $(r_3 \vee r_4) \wedge r_1 \wedge (r_2 \vee r_5)$. La combinaison de ressources finale s'écrit donc $(r_3 \wedge r_1 \wedge r_2) \vee (r_4 \wedge r_1 \wedge r_2) \vee (r_3 \wedge r_1 \wedge r_5) \vee (r_4 \wedge r_1 \wedge r_5)$.

La figure 5.3 offre une représentation de la machine d'état associée à la vue processus de l'exemple précédent.

Chaque composante de ce réseau de Petri est alors décomposée selon les combinaisons de ressources associées à chaque transition. La figure 5.4 reprend l'exemple précédent et présente un exemple de décomposition :

– chaque transition est décomposée en autant de transitions qu'il existe de combinaisons de ressources

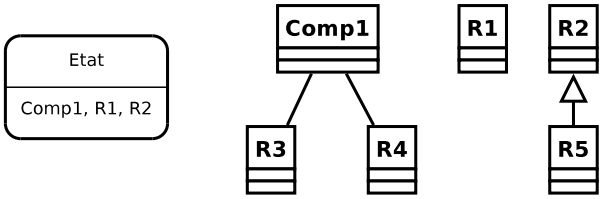


FIG. 5.2 – Combinaison de ressources

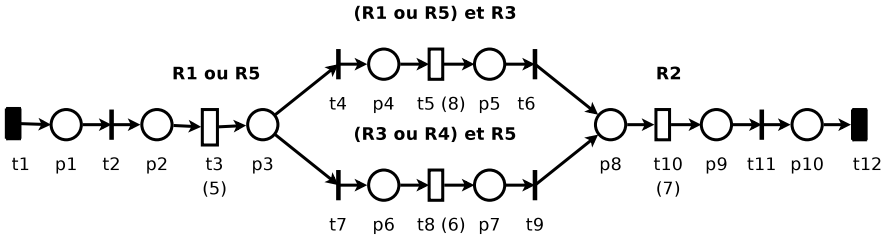


FIG. 5.3 – Machine d'état considérée pour la planification

admissibles pour cette tâche : t_3 qui nécessite r_1 ou r_5 est décomposée en deux transitions, t'_3 qui nécessite r_1 et t'_{16} qui nécessite r_5 ;

– les places de confluence sont également décomposées : la place p_8 est décomposée en six places : p'_6, p'_{10}, p'_{13}, p'_{18}, p'_{22} et p'_{25}.

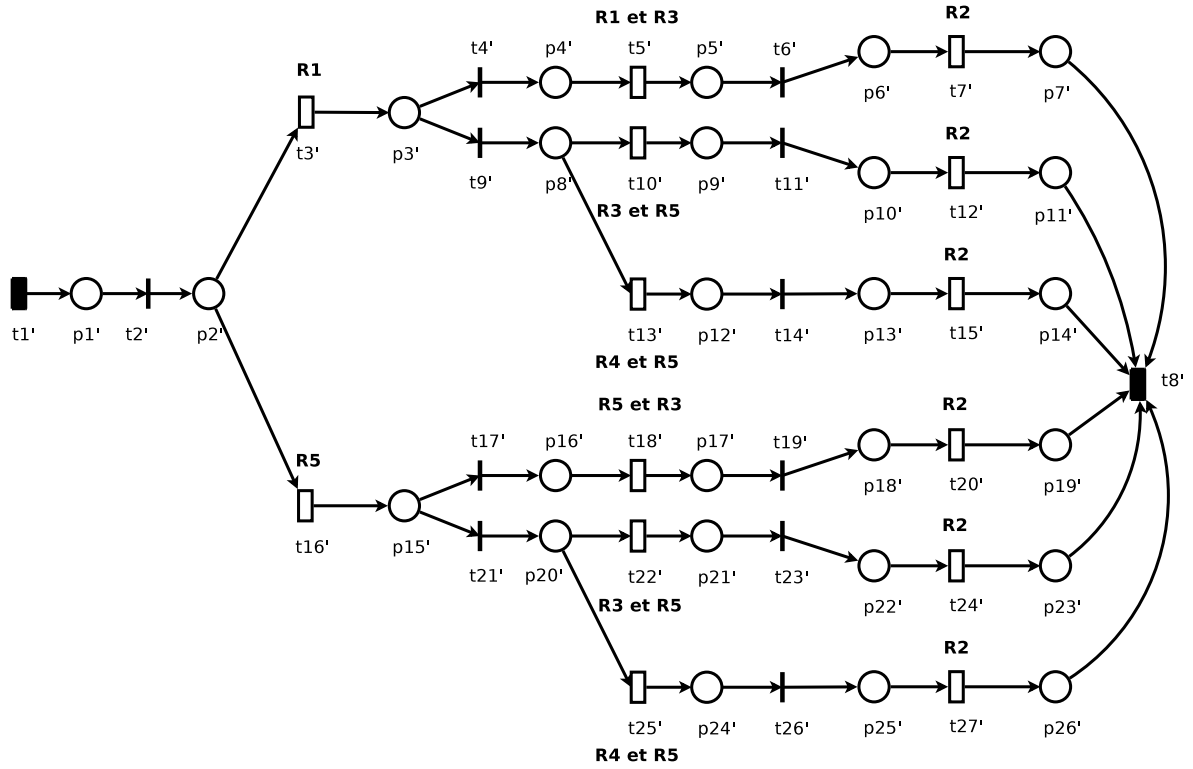


FIG. 5.4 – Réseau de Petri décomposé

Chaque composante du réseau résultant est un arbre permettant de découvrir tous les marquages que l'on peut atteindre à partir du marquage initial. Chacune de ces composantes est un réseau de Petri contrôlable sans place de ressource, donc décomposable.

Durée de montage/démontage

Pour chaque ressource r_g requise au cours du processus, nous définissons une durée de « montage/démontage » (*setup time*) α_g modélisant la durée globale des activités propres à la ressource considérée pour un parcours donné. Dans le cas d'une ressource r_g allouée de manière ponctuelle, nous avons très simplement $\alpha_g = 0$, car il n'existe aucune activité réalisée par la ressource avant ou après l'état modélisé par la transition t. Dans le cas d'une ressource allouée au sein d'une mission définie par l'utilisateur, les temps de montage et de démontage dépendent des activités qui se trouvent avant et après la synchronisation : ces durées sont évaluées globalement selon l'état de la mission considéré.

Exemple 5.3.1 *Si nous reprenons l'exemple précédent, la ressource r_2 est allouée ponctuellement : $\alpha_2 = 0$. En revanche, la ressource r_1 intervient dans le cadre d'une mission : $\alpha_1 = 10+8$ (durées des l'activités correspondant aux états* `Etat0` *et* `Etat6`*).*

5.3.4 Planification à court terme

La planification à court terme concerne essentiellement les services de soins où l'ensemble des activités constituant la prise en charge du patient ne dépasse pas une journée. Les services ambulatoires, ainsi que le bloc opératoire font partie de cette catégorie de services où la planification est requise. La durée d'une période élémentaire correspond à la durée d'ouverture du service pendant la journée, et l'horizon est généralement fixé à une semaine.

Problème d'optimisation

Chacune des composantes du réseau de Petri décrit dans la section précédente est un réseau de Petri décomposable : il existe donc un ensemble $\{W_1, W_2, \ldots, W_n\}$ de t-invariants du modèle N tels que les W_s-CFIO N_{W_s}, $s \in \{1, \ldots, n\}$, recouvrent N. Dans la suite, nous noterons $W_s = [w_{s,1}, \ldots, w_{s,q}]$ où q est le nombre de transitions de N.

Soit $\{u_1, \ldots, u_k, \ldots, u_p\}$ l'ensemble patients à planifier sur les périodes $1, \ldots, H$. Nous définissons la variable $y^j_{s,k}$ de la manière suivante :

$$y^j_{s,k} = \begin{cases} 1 & \text{si le CFIO } N_{W_s} \text{ a été activé durant la période } j \text{ pour le patient } k \\ 0 & \text{sinon} \end{cases} \tag{5.1}$$

Soient $r_1, \ldots, r_m$ les ressources utilisées et T^g_s, $g \in \{1, \ldots, m\}$, l'ensemble des transitions correspondant à la ressource r_g dans le CFIO N_{W_s}. Les contraintes de capacités peuvent être exprimées de la manière suivante :

$$\sum_{k=1}^{p} \sum_{s=1}^{n} \left(y^j_{s,k} \Big(\sum_{t \in T^g_s} w_{s,h(t)} \theta_{t,k} \Big) + \alpha_{s,g} \right) \leq \tau \quad \forall j \in \{1, \ldots, H\},\ \forall g \in \{1, \ldots, m\} \tag{5.2}$$

avec $h(t)$ le rang du t-invariant correspondant à la transition t, $\alpha_{s,g}$ la durée globale de montage/démontage pour la ressource r_g dans le CFIO N_{W_s}, τ le temps utilisable durant chaque période élémentaire et $\theta_{t,k}$ le temps de franchissement de la transition t pour le patient k.

Chaque patient doit être affecté à une période et à un CFIO exactement :

$$\sum_{j=1}^{H} \sum_{s=1}^{n} y^j_{s,k} = 1 \quad \forall k \in \{1, \ldots, p\} \tag{5.3}$$

Finalement le problème s'écrit de la manière suivante :

$$\text{Optimiser } C(\{y^j_{s,k}\}) \tag{5.4}$$

sous les contraintes (5.2) et (5.3), avec $y^j_{s,k} \geq 0$, $\forall j, k, s$. C est le critère à optimiser.

Exemple

Nous considérons le réseau représenté dans la figure 5.4. Supposons que dix patients $u_1, \ldots, u_{10}$ doivent être planifiés sur un horizon de trois jours : $p = 10$ et $H = 4$ car nous ne savons pas si tous les patients pourront être pris en charge sur trois périodes élémentaires. Nous supposons que les durées de franchissement ne dépendent ni des patients ni des ressources mises en jeu. Nous fixons $\tau = 100$. Les t-invariants $W_1, \ldots, W_6$ de cet exemple sont définis de la manière suivante :

- $W_1 = [1, 1, 1, 1, 1, 1, 1, 1, 0, \ldots, 0]$,
- $W_2 = [1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, \ldots, 0]$,
- $W_3 = [1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, \ldots, 0]$, etc.

Nous supposons que le parcours patient est connu : la condition `cond1` est supposée vraie pour les patients $u_1, \ldots, u_5$ et la condition `cond2` est supposée vraie pour les patients $u_6, \ldots, u_{10}$ (c.f. figure 5.1). Nous en déduisons les contraintes suivantes :

$$y^j_{s,k} = 0 \quad \forall j \in \{1, \ldots, H\},\ \forall k \in \{1, \ldots, 5\},\ \forall s \in \{4, \ldots, 6\} \tag{5.5}$$

$$y^j_{s,k} = 0 \quad \forall j \in \{1, \ldots, H\},\ \forall k \in \{6, \ldots, 10\},\ \forall s \in \{1, \ldots, 3\} \tag{5.6}$$

Nous cherchons à minimiser le coût d'utilisation des ressources. Soit $c_{s,k}$ le coût global lié à l'activation du CFIO N_{W_s} pour le patient k. Le critère est donc défini de la manière suivante :

$$\min \sum_{k=1}^{10} \sum_{s=1}^{6} \sum_{j=1}^{4} c_{s,k} y^j_{s,k} \tag{5.7}$$

Les contraintes relatives à chaque ressource $r_1, \ldots, r_5$ sont réunies dans la tables 5.1. Avec les contraintes relatives à l'affectation des patients à une période et à un CFIO, ce problème peut se transformer en un problème de programmation linéaire par l'introduction de variables supplémentaires. Les variables $y^j_{s,k}$ à déterminer sont des entiers positifs ou nuls.

Ressource	**Contrainte** $\forall j \in \{1, 2, 3\}$
r_1	$\sum_{k=1}^{10} \sum_{s=1}^{3} y^j_{s,k}(5w_{s,3} + 8w_{s,5}) + \alpha_{s,1} \leq 100$
r_2	$7 \sum_{k=1}^{10} \sum_{s=1}^{6} y^j_{s,k} + \alpha_{s,2} \leq 100$
r_3	$\sum_{k=1}^{10} \sum_{s=1}^{6} y^j_{s,k}(8w_{s,5} + 6w_{s,10} + 8w_{s,18} + 6w_{s,22}) + \alpha_{s,3} \leq 100$
r_4	$\sum_{k=1}^{10} \sum_{s=1}^{6} y^j_{s,k}(6w_{s,13} + 6w_{s,25}) + \alpha_{s,4} \leq 100$
r_5	$\sum_{k=1}^{10} \sum_{s=1}^{6} y^j_{s,k}(5w_{s,16} + 6w_{s,10} + 6w_{s,13} + 8w_{s,18} + 6w_{s,22} + 6w_{s,25}) + \alpha_{s,5} \leq 100$

TAB. 5.1 – Contraintes de capacité relatives à l'exemple de la figure 5.4

5.3.5 Ordonnancement

Le dimensionnement des ressources nécessaires durant le séjour du patient dans un service de soins permet une évaluation du coût de prise en charge. Connaissant les dates d'arrivée des patients dans le système, nous nous proposons de déterminer :

– les horaires de disponibilité des ressources humaines et matérielles ;
– les effectifs minimum pour chaque ressource intervenant dans le processus de prise en charge du patient.

Ces informations permettrons à l'utilisateur de tester en premier lieu des scénarios de simulation mettant en œuvre une distribution de ressources optimale selon les critères de son choix.

Problème d'optimisation

Nous supposons le problème de planification à court terme résolu : l'ordre et les dates d'arrivée des entités sont connues, ainsi que le parcours de ces entités. Nous cherchons donc à planifier les tâches qui concernent les patients sur chacune des périodes élémentaires définies pour la planification et à affecter les ressources pour ces tâches de manière à optimiser un certain critère.

Pour cette étape nous reprenons le réseau de Petri décomposé présenté figure 5.4. Nous considérons l'ensemble $\{W_1, W_2, \ldots, W_n\}$ de t-invariants du modèle N tels que les W_s-CFIO N_{W_s}, $s \in \{1, \ldots, n\}$, recouvrent N. Nous supposons que les t-invatiants de W_s sont ordonnés en fonction du parcours du patient.

Soit $\{u_1, \ldots, u_k, \ldots, u_p\}$ l'ensemble patients à planifier sur les périodes $1, \ldots, j, \ldots, H$. Nous définissons les variables $y_{s,k}$ et $x_{i,k}$ de la manière suivante :

$$y_{s,k} = \begin{cases} 1 & \text{si le CFIO } N_{W_s} \text{ est activé pour le patient } k \\ 0 & \text{sinon} \end{cases} \tag{5.8}$$

La variable $x_{i,k}$ est la date de tir du ième t-invariant pour le patient k. Les contraintes de précédence peuvent être exprimées de la manière suivante :

$$x_{i,k} + y_{s,k} w_{s,i} \theta_{i,k} \leq x_{i+1,k} \quad \forall k \in \{1, \ldots, p\},\ \forall i \in \{1, \ldots, q-1\},\ \forall s \in \{1, \ldots, n\} \tag{5.9}$$

Soit $\delta^j_{i,k,s}$ une variable booléenne permettant de connaître si la ième transition du CFIO N_{W_s} pour le patient k est en cours de franchissement à l'instant t :

$$\delta^j_{i,k,s} = \mathbf{1}\{x_{i,k} \leq t \leq x_{i,k} + y_{s,k} w_{s,i} \theta(i,k) - 1\} \quad \forall k, i, s, j \tag{5.10}$$

Soient $r_1, \ldots, r_m$ les ressources utilisées et T^g_s, $g \in \{1, \ldots, m\}$, l'ensemble des transitions correspondant à la ressource r_g dans le CFIO N_{W_s}. Les contraintes de capacités relatives aux ressources peuvent être exprimées de la manière suivante :

$$\sum_{k=1}^{p} \sum_{s=1}^{n} \left(y_{s,k} \sum_{t \in T^g_s} \delta^j_{h(t),k,s} w_{s,h(t)} \right) \leq N_g \quad \forall j \in \{1, \ldots, H\},\ \forall g \in \{1, \ldots, m\} \tag{5.11}$$

avec $h(t)$ le rang du t-invariant correspondant à la transition t et N_g le nombre total d'instances de la ressource r_g disponibles.

Enfin, chaque patient doit être affecté à une période et à un CFIO exactement :

$$\sum_{s=1}^{n} y_{s,k} = 1 \quad \forall k \in \{1, \ldots, p\} \tag{5.12}$$

Finalement le problème s'écrit de la manière suivante :

$$\text{Optimiser } C(\{y_{s,k}, x_{i,k}\}) \tag{5.13}$$

sous les contraintes (5.9), (5.10), (5.11) et (5.12), avec $y_{s,k} \geq 0\ \forall s, k$ et $x_{i,k} \geq 0\ \forall i, k$. C est le critère à optimiser.

Le choix de la fonction objectif doit être réalisé en prenant en considération les caractéristiques du système hospitalier en cours d'analyse ainsi que le but de l'étude de modélisation/simulation. Une fois le problème d'ordonnancement résolu, nous obtenons les dates de franchissement des différentes transitions du réseau considéré ainsi que les ressources impliquées dans ces franchissement. Si nous prenons en compte les temps de montage/démontage $\alpha_{s,g}$ définis pour chaque ressource r_g et pour chaque CFIO N_{W_s}, nous sommes en mesure de calculer les effectifs nécessaires durant chaque période élémentaire de l'horizon considéré. L'utilisateur a donc la possibilité d'ajuster les emplois du temps de chaque ressource et/ou de modifier les allocations de ressources au cours du parcours pour approcher l'objectif fixé.

Exemple et interprétation de l'ordonnancement

Reprenons l'exemple de la figure 5.1 et la décomposition présentée figure 5.4. Nous considérons la phase de planification terminée, permettant l'initialisation des données du problème d'ordonnancement sur une période élémentaire. Supposons quatre patients $u_1, \ldots, u_4$ planifiés pour la période considérée. Les t-invariants $W_1, \ldots, W_6$ sont identiques et le parcours patient est supposé connu.

Nous cherchons à minimiser la date de sortie du dernier patient (makespan ou C_{max}). Soit $c_{i,k}$ la date de fin de franchissement du i^e t-invariant du patient k :

$$c_{i,k} = x_{i,k} + y_{s,k} w_{s,i} \theta(i,k) \quad \forall i,k,s \tag{5.14}$$

Le critère est donc défini de la manière suivante :

$$\min \max_{i,k} c_{i,k} \tag{5.15}$$

Considérons la solution présentée dans la table 5.2 et illustrée figure 5.5. Le makespan pour cette solution est égal à 28. L'observation du chronogramme présenté figure 5.5 permet de comprendre que la ressource `R5` est une ressource critique : dans ce cas de figure, il serait judicieux d'ajouter une instance de cette ressource pour désengorger les tâches associées aux transitions t_5 et t_8.

Patient	**CFIO**	t_3	t_5	t_8	t_{10}
u_1	1	0	5	-	13
u_2	6	0	-	5	11
u_3	1	5	13	-	21
u_4	6	5	-	11	18

TAB. 5.2 – Indice du CFIO activé et dates de début de franchissement pour chaque patient

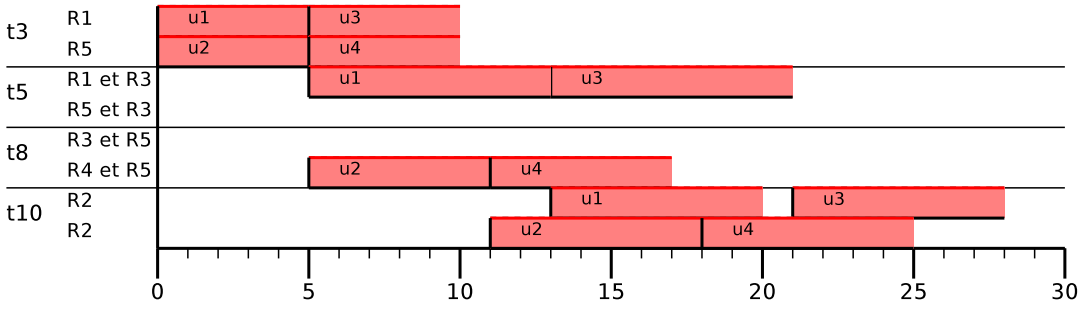


FIG. 5.5 – Évolution des franchissement

Si la période élémentaire est plus grande ou égale à 28 unités de temps, l'ordonnancement considéré est admissible et pourra être conservé.

5.4 Pilotage en temps réél

5.4.1 Principe du système de contrôle proposé

Le couple sous-système de pilotage - sous-système d'information est appelé modèle de contrôle. Il s'agit d'un système à événements discrets qui réagit aux événements externes provenant du modèle d'émulation, de l'utilisateur ou d'une quelconque autre source. Indépendant du sous-système physique, ce modèle est théoriquement interchangeable selon les besoins. Dans notre cas, le modèle de contrôle est constitué d'un certain nombre de modules implémentés sous formes de classes en interaction avec le système d'information, permettant la réalisation de tâches de décision.

Comme cela a été décrit dans le chapitre 4, le sous-système physique, ou modèle d'émulation, n'intègre aucune règle de décision. La simulation est interrompue en amont et en aval de chaque transition du réseau de Petri de santé jusqu'à réception d'un ordre de redémarrage. Sur réception d'un événement de synchronisation, le modèle de contrôle peut lancer un processus de décision encapsulé dans un module spécifique et paramétrer le modèle d'émulation en fonction des résultats par l'intermédiaire du système d'information. Cette phase de paramétrage peut prendre différentes formes : tirage de transitions, modification d'attributs, génération de variables aléatoires, etc.

La figure 5.6 offre une représentation de la synchronisation existant entre le modèle d'émulation et le modèle de contrôle. Les points de synchronisation sont placés en amont et en aval de toute transition du réseau de Petri de santé. L'interface de communication favorise interactions et échanges entre les deux modèles par le biais de procédures d'appels standardisées, permettant éventuellement le remplacement du modèle de contrôle pour le test de différentes stratégies de pilotage par exemple. Le réseau de Petri de santé en lui-même n'offre aucun indice quant à la prise de décision au cours de la simulation.

L'interface de communication qui fait le lien entre le modèle d'émulation et le modèle de contrôle doit permettre au modèle de pilotage de maintenir sa représentation du système contrôlé à jour afin d'assurer la cohérence des décisions prises avec la situation réelle du système hospitalier. Les messages échangés entre les deux modèles doivent donc être standardisés et contenir un minimum d'informations :

- l'événement à l'issu duquel un appel au modèle de contrôle à été réalisé, permettant l'identification de la transition qui a été ou qui va être tirée ;
- l'entité ou le groupe d'entités impliquée(s) dans l'événement ;
- le marquage du réseau au moment de la synchronisation.

Le modèle de pilotage est donc appelé à manipuler des données invisibles du point du vue du réseau de Petri de santé, tels les attributs des objets qui circulent dans le réseau ou les données du système d'information. Le formalisme orienté objet utilisé pour la modélisation du système grâce à la plate-forme medPRO est directement réutilisé : classes d'entités, de ressources, de compétences, etc. ainsi que leurs relations peuvent être interprétées nativement.

Comme cela a été dit précédemment, le modèle de contrôle est organisé sous forme de modules interchangeables chargés d'interpréter et de répondre aux requêtes en provenance du modèle d'émulation. Nous définissons plusieurs catégories de modules :

Modules de traitement pré-franchissement : réalisation de tâches liées à la sélection de ressources et test de faisabilité de franchissement ;

Modules de traitement post-franchissement : évaluation de conditions liées à des transitions et orientation de jetons à travers le réseau ;

Modules spécifiques : les procédures encapsulées dans ces modules permettent la réalisation de tâches particulières liées à la modélisation de processus de décisions spécifiques aux systèmes hospitaliers.

Les contenus de chacun de ces ensembles sont décrits dans les sections suivantes.

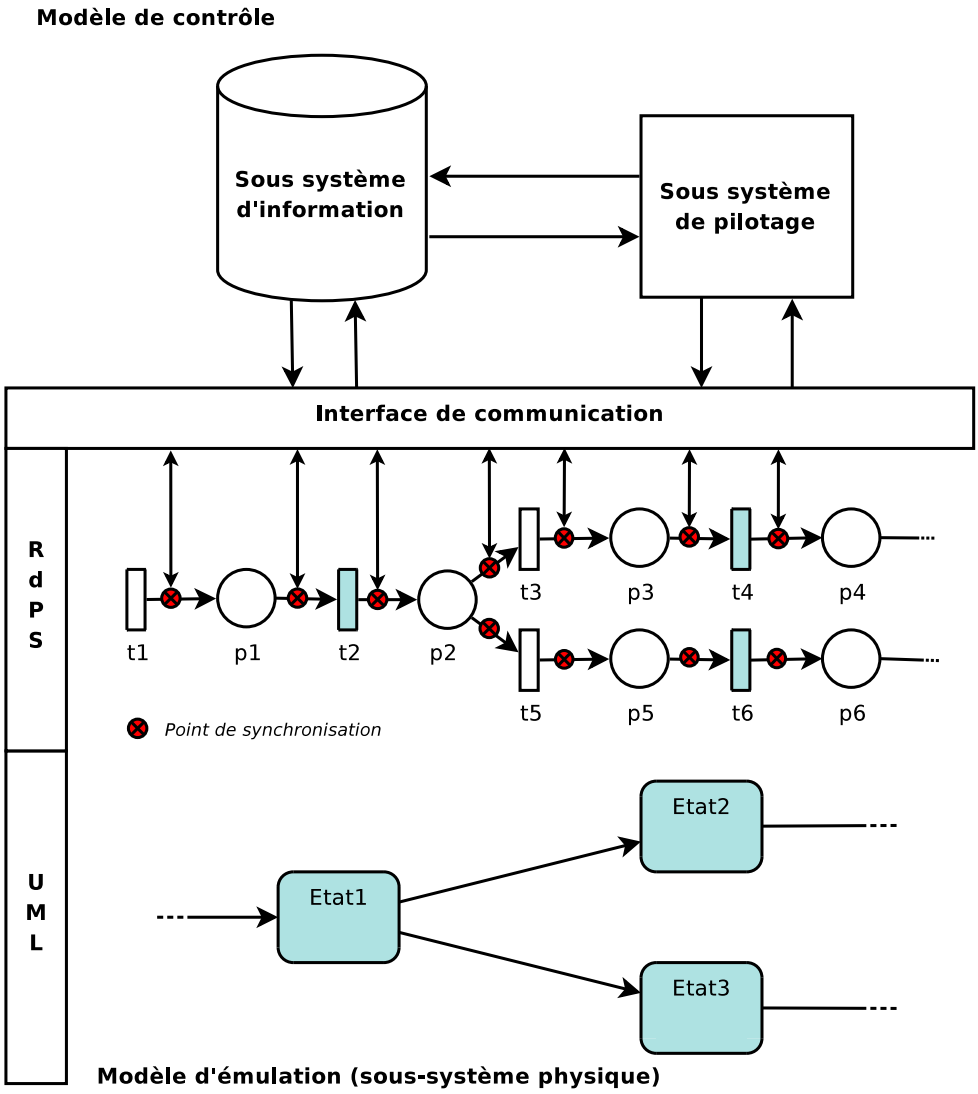


FIG. 5.6 – Intéractions entre les trois sous-systèmes

5.4.2 Modules de traitement pré-franchissement

La synchronisation liée à l'appel du sous-système de pilotage est réalisée juste avant le début du franchissement d'une transition du réseau de Petri de santé. Il s'agit donc du traitement d'un événement de type $(d, t, 0)$. La nature du module de contrôle mis en œuvre pour la prise de décision dépend de la nature de la transition t qui doit être tirée :

1. t est la première transition d'une mission, impliquant éventuellement la sélection d'une ressource selon ses compétences, la constitution d'une équipe ou encore le remplacement d'une ressource.
2. $t \in TE$, le module de sélection de ressource est appelé afin de déterminer la meilleure combinaison de ressources pour l'accomplissement de la tâche liée à la transition t.
3. t est une transition conditionnelle ($t \in TC$) dont la condition de garde a déjà été validée (c.f. modules de traitement post-franchissement), le tirage de la transition entraînant éventuellement une mise à jour de l'état du système.

Les arguments passés aux modèle de contrôle sont l'événement $(d, t, 0)$ contenant la date courante $d = t_{now}$ et la transition t à tirer, ainsi que le marquage courant du réseau M^d. Si t est une transition

modélisant un P-état, l'entité u est également transmise ainsi que la place $p \in \bullet t$ où elle se trouve. Si t est une transition modélisant un R-état non-synchronisé, la ressource r est transmise ainsi que la place $p \in \bullet t$ où elle se trouve.

Début d'une mission

L'événement transmis au modèle de contrôle est le début du franchissement de la première transition d'une mission :

Cas d'une ressource simple : Si la ressource simple initialement prévue est indisponible, un remplaçant doit être déterminé pour que la mission puisse débuter. Si aucun replaçant n'est disponible, la transition ne peut être tirée et sera mise en attente. L'algorithme de sélection est détaillé dans l'annexe B.5.

Cas d'une compétence : Une ressource susceptible d'accomplir cette mission doit être sélectionnée. La procédure consiste à parcourir l'ensemble des ressources possédant cette compétence pour sélectionner le(s) meilleur(s) candidat(s). La mission peut débuter si une ressource a finalement été trouvée. L'algorithme de sélection est détaillé dans l'annexe B.6.

Cas d'une équipe : Les membres de l'équipe doivent être réunis pour permettre le commencement de la mission. Malheureusement, tous les membres d'une équipe ne sont pas toujours disponibles au même moment ; une stratégie permettant la sélection de ressources a été mise en place au sein du modèle de contrôle afin de faire face à ces situations. Le système de contrôle tente tout d'abord de réunir l'ensemble des membres de l'équipe. Si l'équipe est complète la mission peut commencer, sinon les ressources présentes sont mises en attente. Une équipe intervient toujours dans le cadre d'une mission. L'algorithme complet permettant la constitution d'une équipe est décrit dans l'annexe B.7.

Sélection de ressources pour la réalisation d'une tâche liée à un état

La vérification de la disponibilité des ressources et leur sélection constitue une tâche primordiale du modèle de contrôle. Si une combinaison de ressources disponibles en accord avec les spécifications de l'utilisateur a pu être déterminée, la transition peut être tirée (la tâche liée à la transition commence et l'état devient actif). Dans le cas contraire, la transition ne pourra être tirée et le système sera mis en attente. Dans ce dernier cas, une décision de l'utilisateur peut être requise pour débloquer la situation, décision qui pourra être réutilisée à l'avenir si le cas se présente à nouveau.

Nous nous plaçons dans le cas où la transition t qui doit être tirée est liée à un P-état ou à un P-état synchronisé avec un R-état. Les ressources ou compétences nécessaires sont connues grâce à l'ensemble $\bullet t$ et grâce aux spécifications de l'état correspondant à la transition t.

Lorsque les ressources nécessaires pour la réalisation d'une tâche ne sont pas explicitement définies (sélection sur compétence par exemple), un processus de décision est mis en œuvre pour réaliser la sélection de ressources. Ce processus est modélisé sous la forme d'un programme linéaire à résoudre. La fonction objectif et les contraintes du problème linéaire sont générées automatiquement en fonction de l'organisation et de l'état du système.

De manière plus formelle, nous définisson les ensembles suivants :

- soit $R = \{r_1, \ldots, r_m\}$ l'ensemble des $m \in \mathbb{N}^*$ ressources potentiellement requises à l'instant t_{now} définies dans la vue organisation ;
- soit $A = \{A_1, \ldots, A_n\}$ une liste de $n \in \mathbb{N}^*$ ensembles de ressources requises pour la réalisation de l'activité. Chaque liste A_i est un ensemble de p_i ressources candidates sélectionnées dans R ;

Soit e_j, $j \in \{1, \ldots, m\}$ le nombre total d'instances de la ressource r_j disponibles. Le problème réside dans le choix d'une ressource dans chaque liste A_i afin d'optimiser une fonction objectif F. Soit x_{ij} les

variables de décision booléennes définies comme suit :

$$x_{ij} = \begin{cases} 1 & \text{si la ressource } r_j \text{ est choisie pour l'ensemble } A_i \\ 0 & \text{sinon} \end{cases} \tag{5.16}$$

La fonction objectif du problème peut être exprimée comme suit :

$$F = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} f(r_j) \tag{5.17}$$

où f est une fonction définie par l'utilisateur. Les contraintes suivantes sont établies :

$$x_{ij} = 0 \ \forall r_j \notin A_i \tag{5.18a}$$

$$\sum_{j} x_{ij} = 1 \ \forall i \tag{5.18b}$$

$$\sum_{i} x_{ij} \leq e_j \ \forall j \tag{5.18c}$$

La contrainte (5.18a) permet de limiter le domaine de définition de x_{ij} selon les listes A_i. La contrainte (5.18b) vérifie qu'exactement une ressource est sélectionnée par liste A_i. La contrainte (5.18c) vérifie que le nombre total de ressources sélectionnées n'excède pas le nombre total d'instances disponibles de cette ressource.

Afin de mettre en œuvre le processus de sélection de ressources tel qu'il a été décrit, les ensembles C et A doivent être initialisés. C'est le rôle de l'algorithme décrit dans l'annexe B.8. Cette procédure examine l'ensemble des places $p_i \in {}^\bullet t \setminus \{p\}$. En fonction de la nature de p_i et de la ressource r_i qui s'y rapporte les ensembles C et A sont mis à jour.

La fonction objectif est définie en fonction des attributs des différentes ressources mises en jeu. Lorsqu'une solution optimale a été déterminée, l'état du système est mis à jour : les transitions permettant la sélection des ressources sont tirées, le marquage est mis à jour et la durée de franchissement de la transition $\theta(t)$ est éventuellement modifiée en fonction des ressources qui ont été sélectionnées.

L'ultime étape consiste à tirer une variable aléatoire si la durée du franchissement de la transition est exprimée de manière stochastique. Les distributions de probabilités sont implémentées au sein du système de contrôle et choisies par l'utilisateur au moment de la modélisation.

Exemple de sélection automatique de ressources

La figure 5.7 présente un exemple de mise en œuvre de sélection de ressource automatisée pour le tirage de la transition t. Les ressources nécessaires sont : une instance d'une ressource possédant la compétence c_1, deux instances de la ressource r_5, une instance de la ressource r_9 et une instance d'une ressource possédant la compétence c_2. La ressource r_9 est synchronisée tandis que c_1, c_2 et r_5 sont des ressources allouées ponctuellement (libérées à l'issu du tir de t).

D'après le réseau présenté figure 5.7, les données du problème d'optimisation sont :

- $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$, R contient toutes les ressources potientiellement impliquées dans le franchissement de t ;
- $A = \{A_1, A_2, A_3, A_4\}$ avec $A_1 = \{r_1, r_2, r_3\}$, $A_2 = A_3 = \{r_4, r_5\}$, $A_4 = \{r_6, r_7, r_8\}$; $A_2 = A_3$ car deux instances de r_5 sont requises ;
- $e_1 = e_6 = 0$, $e_2 = e_3 = e_4 = e_5 = e_8 = 1$, $e_7 = 2$.

Une solution admissible pour cet exemple serait $x_{12} = x_{25} = x_{34} = x_{47} = 1$ avec $F = f(r_2) + f(r_5) + f(r_4) + f(r_7)$.

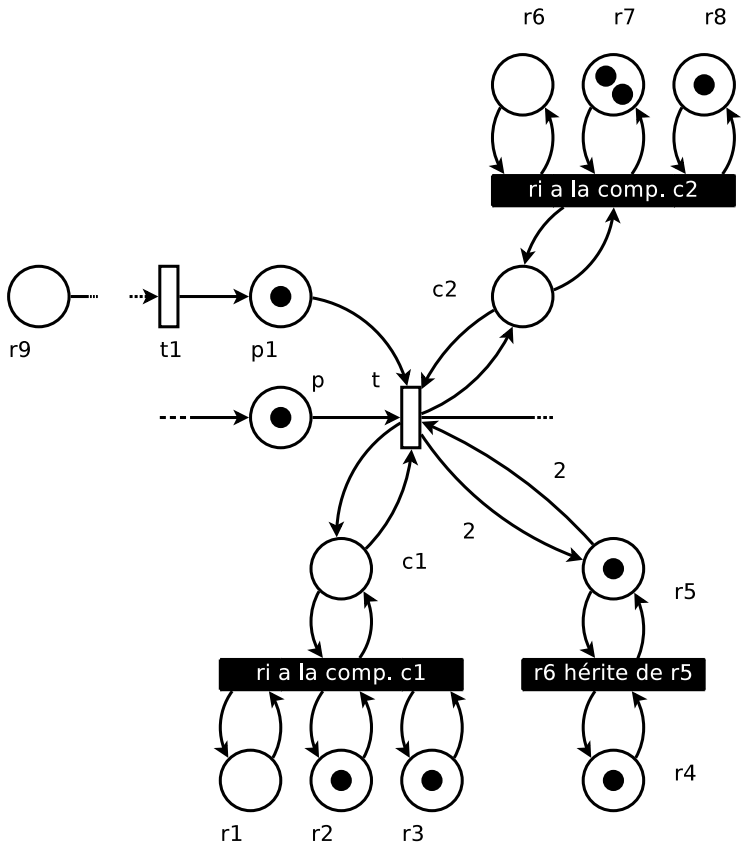


FIG. 5.7 – Exemple de sélection de ressources

5.4.3 Modules de traitement post-franchissement

Une nouvelle synchronisation est réalisée juste après le franchissement d'une transition du réseau de Petri de santé. Un appel au système de pilotage est réalisé pour traiter un événement de type $(d, t, 1)$. La nature du module de contrôle mis en œuvre pour la prise de décision dépend de la nature de la transition t qui doit être tirée :

1. t est la dernière transition d'une mission, impliquant éventuellement la dissolution d'une équipe ou la libération d'une ressource selon une compétence.
2. $t \in TE$, le franchissement d'une transition correspondant à une tâche est terminé et il s'agit de déterminer la prochaine transition conditionnelle à tirer.
3. t est une transition conditionnelle ($t \in TC$) et aucune décision n'est requise pour mettre à jour le système.

De la même manière que pour le traitement pré-franchissement, les arguments passés aux modèle de contrôle sont l'événement $(d, t, 1)$ contenant la date courante $d = t_{now}$ et la transition t à tirer, ainsi que le marquage courant du réseau M^d. Si t est une transition modélisant un P-état, l'entité u est également transmise ainsi que la place $p \in t\bullet$ où elle se trouve. Si t est une transition modélisant un R-état non synchronisé, la ressource r est transmise ainsi que la place $p \in t\bullet$ où elle se trouve.

Fin d'une mission

Le traitement d'une fin de mission consiste à rendre la ressource à nouveau disponible. Dans le cas d'une ressource simple, il s'agit de restituer les jetons de cette ressource simple dans sa place de base.

Dans le cas d'une équipe, les membres de l'équipe doivent réintégrer leurs places de base respectives. Enfin, dans le cas d'une mission définie pour une compétence, il s'agit de rediriger l'instance de ressource sélectionnée en début de mission vers la bonne place de base. Les algorithmes correspondant de sont pas détaillés ici car ils sont les symétriques des algorithmes de traitement de début de mission définis plus haut.

Branchement conditionnel à l'issu d'une tâche : centre de décision

À l'issu du franchissement d'une transition liée à une tâche, un jeton est ajouté dans une place p de sortie et se retrouve donc dans un centre de décision. L'évaluation de conditions associées aux transitions en sortie de p est réalisée par le système de contrôle. Les expressions admissibles qui doivent être évaluées ont été listées dans la table 3.1 de la section 3.3.4 ; l'expression considérée peut être récupérée grâce à la fonction ξ. Les conditions sont évaluées selon l'ordre précisé par l'utilisateur et toutes les conditions d'un centre de décision doivent être de même nature :

Test sur la valeur d'un attribut ou d'une variable : les valeurs des attributs et/ou variables sont récupérées et l'expression est évaluée.

Test sur la valeur de l'horloge globale ou d'une horloge locale : l'expression inclue l'horloge globale (t_{now}) ou une horloge locale définie par l'utilisateur ; ce type de condition permet généralement de tirer la transition à une date précise dans le temps.

Test sur l'état d'une entité ou d'une ressource : ce type de condition permet de scruter si une instance d'entité ou de ressource se trouve dans un état particulier au moment du test.

Tirage aléatoire : le système de contrôle tire la transition de sortie du centre de décision aléatoirement grâce à la loi uniforme $U \sim U(0, 1)$. Les probabilités de sortie doivent être exprimées sous forme de pourcentages.

L'algorithme détaillé permettant de choisir la prochaine transition à tirer est présenté dans l'annexe B.9.

5.4.4 Module de contrôle spécifique : demande d'examens médicaux

Ce module permet la modélisation du processus de demande d'examens pour les patients d'un service de soins. Ce processus consiste à prendre un rendez-vous auprès d'un service externe pour un patient : scanner, échographie, etc. Le service prestataire gère son carnet de rendez-vous de manière indépendante. Cette pratique est courante dans la plupart des centres hospitaliers où le fonctionnement des différents services de soins est dissocié. Nous nous proposons de modéliser ce processus de prise de rendez-vous du point de vue du système modélisé, sans tenir compte de l'organisation des services prestataires.

Un état particulier dont le nom est typographié en italique est utilisé dans la plate-forme medPRO pour modéliser ce module spécifique, présenté figure 5.8. Un algorithme de contrôle spécifique est associé à cet état : cet algorithme permet de mettre à jour la durée de séjour du patient en fonction des dates de rendez-vous obtenues.

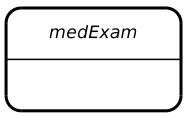


FIG. 5.8 – État correspondant au module de demande d'examens médicaux

L'utilisation de ce module implique l'initialisation de certaines données. Nous définissons tout d'abord l'ensemble $E = \{e_1, \ldots, e_m\}$ des examens potentiellement requis pour l'unité de soins considérée. Les fonctions suivantes sont définies :

- $P_i : E \rightarrow [0,1]$ l'application qui associe à un examen $e \in E$ sa probabilité d'être demandé au ième jour du séjour du patient ;
- $PU : E \rightarrow [0,1]$ l'application qui associe à un examen $e \in E$ sa probabilité d'être demandé en urgence ;
- $D : E \rightarrow \mathbb{N}$ l'application qui associe à un examen $e \in E$ son délai normal d'obtention ;
- $DU : E \rightarrow \mathbb{N}$ l'application qui associe à un examen $e \in E$ son délai d'obtention s'il est demandé en urgence ;
- $D_{max} : E \rightarrow \mathbb{N}$ l'application qui associe à un examen $e \in E$ le délai d'attente maximum admissible pour un patient en cours de séjour.

Chaque patient susceptible d'entrer dans ce P-état doit posséder les attributs suivants :

- $n \in \mathbb{N}$ permettant de connaître de nombre d'examens programmés pour ce patient ;
- $l \in \mathbb{N}$ est la durée de séjour du patient ;
- $s \in \mathbb{N}$ est la date d'entrée du patient dans le service ;
- $d : E \rightarrow \mathbb{N}$ l'application qui associe à un examen particulier du patient un délai d'obtention : il s'agit du délai entre la date de demande et la date du rendez-vous pour ce patient.

Enfin, $t_{now} \in \mathbb{N}$ désigne le moment où la demande d'examen est réalisée. Nous considérons qu'un examen médical peut être demandé à tout moment durant le séjour du patient, lors de la visite du médecin par exemple. L'ensemble des examens E est alors parcouru pour déterminer quels examens le patient considéré est susceptible de subir.

Pour chaque examen $e \in E$ sélectionné, un nouveau tirage permet de savoir si cet examen est demandé normalement ou en urgence : ce tirage permettra de déduire le délai d'attente $d(e)$ associé à e. La dernière étape consiste à déterminer l'impact de ce délai sur la durée de séjour du patient : si le délai d'attente est trop long ($d(e) > D_{max}(e)$), la durée de séjour l est inchangée. Sinon, l prend la valeur $\max(los - (t_{now} - s), d(e)) + \epsilon$, où ϵ est une variable permettant d'ajuster la durée de séjour.

Une autre stratégie de prise de rendez-vous consiste à allouer des créneaux réservés pour le service de soins considéré. Dans ce cas, l'algorithme de contrôle associé au module est légérement différent. Nous définisson la fonction $C : \mathbb{N} \times E \rightarrow \mathbb{N}$ qui associe à une période de temps quelconque et à un examen un nombre de créneaux disponibles. Chaque examen $e \in E$ sélectionné pour un patient, sera alloué à un créneau disponible dans la période courante. Si aucun créneau n'est disponible, un délai d'attente sera calculé de la manière habituelle.

Pour chaque module inclu dans le modèle, deux configurations sont possibles :

Configuration automatique : selon l'unité de soins modélisée, l'utilisateur peut choisir une configuration automatique. Par exemple, le module peut être pré-configuré pour l'unité neuro-vasculaire : les types d'examens sont sélectionnés et les lois de probabilité sont configurées selon la stratégie d'allocation choisie (c.f. chapitre 6).

Configuration manuelle : la configuration manuelle permet à l'utilisateur de paramétrer entièrement le module selon les données relevées dans le système d'origine.

Les indicateurs de performance inclus dans ce module permettent de réaliser des statistiques sur les demandes et les délais d'attente des examens requis. L'impact sur la durée de séjour du patient peut également être relevée afin d'évaluer les coûts supplémentaires dûs aux retards éventuels.

5.4.5 Module de contrôle spécifique : dotation nominative de médicaments

Les préparateurs de la pharmacie centrale d'un hôpital sont impliqués dans un certain nombre d'activités nécessitant une planification précise des tâches à réaliser, principalement en raison de contraintes de dates de livraison à respecter. Nous prenons ici pour exemple deux problèmes similaires du point de vue de l'ordonnancement, à savoir la réalisation de dotations nominatives de médicaments pour un service

de soins particulier (dispensation nominative) ou encore la fabrication de lots de médicaments par service (flux de fabrication magistrale).

Dans le cas de la dispensation nominative, un préparateur doit traiter un certain nombre d'ordonnances sur un certain horizon, généralement une demi-journée. Toutes les ordonnances sont disponibles au début de l'horizon de planification, et chaque ordonnance possède une date de livraison, correspondant généralement au délai d'attente maximum admissible pour le patient. Chaque ordonnance doit être vérifiée par le préparateur, qui doit ensuite réunir les médicaments nécessaires dans une boîte qui sera transportée auprès du patient par un transporteur. Au cours de l'horizon de planification, des ordonnances considérées « urgentes » peuvent arriver par fax à la pharmacie : les dotations correspondantes doivent alors être réalisées en priorité.

Dans le cas de la fabrication magistrale de médicaments, le processus de traitement est similaire : un certain nombre d'ordonnances où sont précisés les différents composants chimiques nécessaires pour chaque patient sont traitées de manière séquentielle par un préparateur. Les produits fabriqués sont ensuite transportés dans le service de soins par un transporteur.

Ces deux problèmes d'ordonnancement peuvent être modélisés par un flowshop à deux machines avec dates d'échéances, où la première machine modélise le préparateur et la deuxième machine modélise le transporteur. Nous associons à chaque dotation i sa durée de préparation $p_{i,1}$ et sa durée de transport $p_{i,2}$. Soit C_i la date d'arrivée de la dotation dans le service. Nous cherchons à minimiser le retard maximum $T_{max} = \max T_i$ où $T_i = \max\{C_i - d_i, 0\}$ est le retard pour la dotation i. Nous ne prenons pas en compte dans un premier temps les demandes urgentes qui apparaissent au cours du temps. Le problème s'écrit donc $M2|d_i|T_{max}$. Ce problème est NP-difficile au sens fort. Lin (2001) propose une méthode de résolution basée sur une réduction polynomiale de 3-Partition. Birman et Mosheiov (2004) proposent une méthode en $O(n^2 \log(n))$ basée sur l'algorithme de (Johnson, 1954).

Nous proposons un état de contrôle particulier permettant de modéliser ce module spécifique, présenté figure 5.9. Une stratégie de contrôle spécifique est associée à cet état, mettant en œuvre un algorithme d'ordonnancement lié à ce problème.

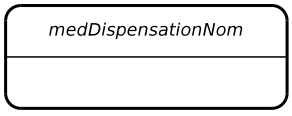


FIG. 5.9 – État correspondant au module de dispensation nominative de médicaments

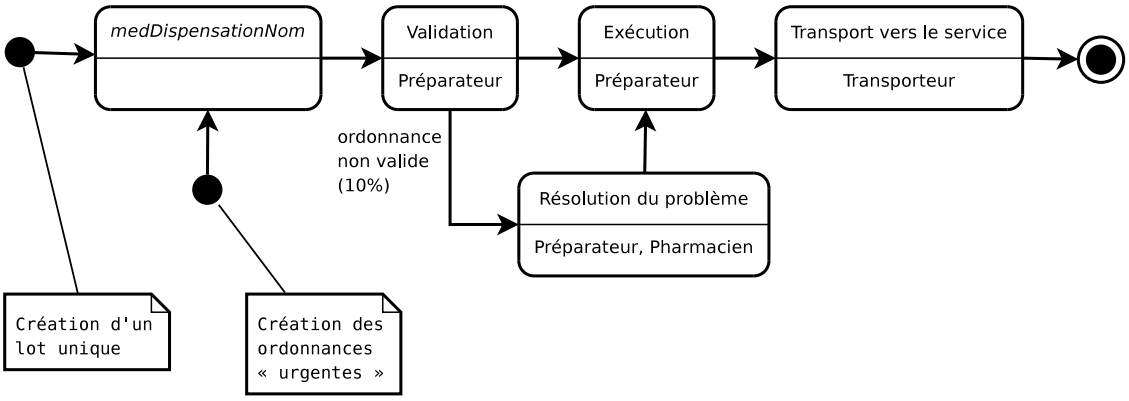


FIG. 5.10 – Processus de dotation nominative de médicaments

Dans ce cas de figure, l'ordonnancement des dotations est réalisé au début du processus uniquement. L'arrivée de demandes urgentes est modélisée en utilisant le même module de contrôle : lorsqu'une nouvelle

ordonnance est générée, l'algorithme est à nouveau exécuté pour insérer la nouvelle dotation à réaliser en première place et pour calculer un nouvel ordonnancement qui prendra en compte cette perturbation. Ce changement est transparent pour le préparateur qui continuera à traiter ses dotations selon l'ordre géré par le module de contrôle. La figure 5.10 illustre le processus de dispensation nominative de médicaments dans son intégralité.

5.5 Synthèse

Nous avons présenté dans ce chapitre une structure décisionnelle originale pour la plate-forme medPRO organisée sur deux plans : (i) la planification et l'ordonnancement, et (ii) le pilotage en temps réel.

La transposition dans un contexte hospitalier de méthodes de planification à court terme et d'ordonnancement initialement développées pour des systèmes de production industriels se révèle intéressante. Nous avons exploités pour cela les propriétés des réseaux de Petri de santé décomposables. Nous sommes ainsi en mesure de présenter deux méthodes permettant d'une part la planification à court terme de patients (échelle macroscopique, horizon d'une semaine), et d'autre part l'ordonnancement de ressources (échelle microscopique, horizon d'une journée) tout en tenant compte des spécificités du modèle développé sur la plate-forme medPRO. L'application de ces méthodes est immédiate. Les résultats obtenus sont injectés dans le modèle de simulation : la robustesse des plannings peut ainsi être évaluée au moyen de scénarios de simulation définis par l'utilisateur.

Ces méthodes ne peuvent cependant pas être appliquées à n'importe quel type de système hospitalier. Nous nous plaçons ici dans le cas de systèmes où la durée de séjour des patients est très courte (moins d'une journée). Ces méthodes sont donc inadaptées aux études concernant des services de long et moyen séjour. En revanche, la planification et l'ordonnancement peuvent être appliquées à des unités médicales ambulatoires (centres de chimiothérapie, cabinets de médecins) ou à des systèmes possédant une durée de prise en charge très rapide, comme le bloc opératoire.

Nous avons ensuite présenté une architecture de pilotage en temps réel dédiée aux systèmes hospitaliers. Nous avons opté pour une approche combinant organisations hiérarchique et hétérarchique afin de proposer un module de pilotage indépendant du système physique. La mise en place d'une interface standardisée entre ces deux sous-systèmes offre une grande flexibilité pour le test de stratégies de contrôle différentes sur un même système physique. La simulation est ainsi pilotée par un ensemble de modules configurés pour une meilleure adaptation au système de production de soins considéré. La modification de modules existants et/ou l'ajout de modules spécifiques permet d'augmenter le degré de fidélité de la modélisation en fonction du système étudié.

L'intérêt majeur de la modularité du système de pilotage est d'offrir une meilleure visibilité au travers de la modélisation UML. L'intégration d'algorithmes de traitements (résolution de programmes linéaires, procédures d'ordonnancement, etc.) est souvent fastidieuse et lourde à mettre en œuvre dans un outil de modélisation/simulation classique. L'aspect modulaire permet également un enrichissement de la bibliothèque de la plate-forme en fonction des études menées.

Nous n'avons exploré dans ce chapitre qu'une fraction des possibilités offertes par une telle structure décisionnelle : l'application de méthodes d'optimisation permettant le test automatique de scénarios de simulation avec variation des paramètres fait partie de nos perspectives de recherche dans ce domaine. Le développement de modèles de contrôle différents sera également exploré. Enfin, le paradigme holonique représente également un axe de recherche très séduisant de par sa grande flexibilité.

Deuxième partie

Applications

Chapitre 6

L'unité neuro-vasculaire

Ce chapitre est consacré à l'étude des durées de séjour des patients victimes d'accidents vasculaires cérébraux (AVC) au sein de l'unité neuro-vasculaire du CHU de Saint-Étienne. Deux modèles de simulation réalisés sous Arena et medPRO ont été mis en œuvre et comparés afin de mettre en valeur les modules de contrôle intégrés à la plate-forme medPRO. Une organisation alternative est proposée et discutée.

6.1 Introduction

L'unité neuro-vasculaire constitue un cas d'étude intéressant dans la mesure où nous nous intéressons à la durée moyenne de séjour (DMS) des patients pris en charge dans l'unité neuro-vasculaire du CHU de Saint-Étienne. La DMS est fonction de plusieurs facteurs : durée du traitement médical, délai avant réception d'examens médicaux, date d'entrée dans un établissement de soins de suite, etc. Son évaluation est donc difficile à réaliser et nécessite la mise en œuvre d'une logique de contrôle avancée si l'on désire simuler le fonctionnement d'un tel système. Cette étude concerne et vise à mettre en évidence l'impact des délais d'attente d'examens complémentaires sur les durées moyennes de séjour des patients atteints d'accidents vasculaires cérébraux (AVC).

La plupart des études de la littérature portent sur l'organisation du service d'imagerie lui-même ; nous avons trouvé peu de références sur l'influence des délais des rendez-vous donné par ce même service sur la durée de séjour patient. Marshall *et al.* (2004) suggèrent une méthodologie pour modéliser les différentes durées de séjour des personnes agées. Les auteurs utilisent pour cela des chaînes de Markov. Une classification préalable au moyen de réseaux bayesiens permet d'identifier plusieurs groupes de patients. Les auteurs démontrent ainsi qu'il existe une relation entre le niveau de dépendance du patient, sa durée de séjour et l'issue de son séjour. Nous remarquons ainsi que la durée de séjour est fortement proportionnelle aux ressources nécessaires pour prendre en charge le patient, et qu'un séjour long aboutit presque toujours à un transfert dans un centre spécialisé.

Après une rapide description des pathologies traitées dans une telle unité, l'objectif de l'étude est détaillé section 6.2. Nous décrivons ensuite l'organisation du service ainsi que le parcours patient section 6.3. La collecte et l'analyse des données sont détaillées dans la section 6.4. Une modélisation du système sous medPRO est proposée section 6.5. Plusieurs scénarios de simulation mettant en œuvre une organisation alternative sont présentés section 6.6. Enfin les résultats de la simulation sont présentés section 6.7. Le bilan de cette étude sera détaillé dans la section 6.8.

Nous aimerions remercier très chaleureusement les Docteurs Pierre Garnier et Stéphanie Demasles, Mmes Martine Borel et Monique Pichon-Galland, ainsi que l'ensemble du personnel des services 12C et 5EF du CHU de Saint-Étienne pour leur disponibilité et leur patience durant cette étude, qui n'aurait pas pu être menée à son terme sans leur précieuse aide.

6.2 Contexte de l'étude

Un accident vasculaire cérébral (AVC), aussi appelé « attaque cérébrale », est un déficit neurologique soudain d'origine vasculaire. L'âge moyen des patients touchés est de soixante-dix ans, mais un AVC peut se produire à tout âge. Les AVC représentent la majorité des causes d'hémiplégie (paralysie d'un côté) récente et frappent environ 130.000 nouveaux cas par an en France. La mortalité à six mois est évaluée à 30 %.

Les AVC sont classés deux catégories :

Accident vasculaire ischémique : les accidents ischémiques sont plus fréquents (80 % des cas). Ils sont dus à l'occlusion d'une artère cérébrale ou à destination cérébrale : l'apparition d'un caillot de sang empêche l'irrigation du cerveau. L'accident vasculaire cérébral peut être transitoire avec retour rapide à l'état normal, sans séquelles. Le déficit peut être au contraire permanent. On parle alors d'accident vasculaire cérébral ischémique constitué.

Accident vasculaire hémorragique : les accidents hémorragiques représentent 20 % des AVC et sont en rapport avec la rupture d'un vaisseau sanguin.

La figure 6.1 présente un cas d'accident ischémique (un caillot est apparu dans l'hémisphère droit du cerveau, et la zone grisée n'est plus irriguée), ainsi qu'un cas d'accident hémorragique dans l'hémisphère gauche du cerveau.

Le diagnostic d'un accident vasculaire cérébral se décompose en plusieurs phases distinctes. L'interrogatoire permettra de préciser les circonstances d'installation des facteurs de risque personnels ou familiaux. Un examen neurologique permet de préciser le diagnostic grâce à des tests visuels et physiques simples et de déduire les fonctions touchées (force musculaire et sensibilité, réflexes, sensorialité, état de conscience, coordination, équilibre, langage et vision). Un examen cardiovasculaire permettant la détection d'anomalies cardiaques et artérielles est systématique. L'imagerie (scanner et/ou IRM) permet de préciser le mécanisme de l'accident (ischémique ou hémorragique) et d'éliminer les autres diagnostics (tumeurs par exemple). D'autre part l'IRM permet de diagnostiquer un AVC ischémique dans les minutes qui suivent l'arrêt de l'oxygénation d'une zone du cerveau. Néanmoins, le scanner est la plupart du temps plus rapide et plus facile d'accès, il est très souvent réalisé en première intention.

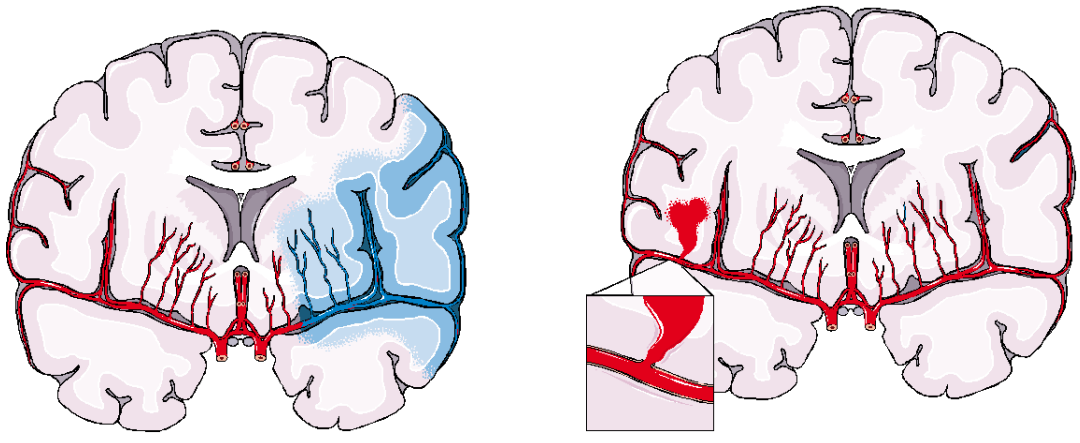


FIG. 6.1 – Accident vasculaire ischémique (à gauche) et hémorragique (à droite)

De par les spécificités de l'accident vasculaire cérébral, l'unité neuro-vasculaire est un service très demandeur d'examens complémentaires qui permettront de préciser le diagnostic. Cette étude a pour objectif l'évaluation de l'impact sur les durées moyennes de séjour (DMS) des délais de rendez-vous d'examens (imagerie, etc.) pour les patients admis dans l'unité neuro-vasculaire. Nous étudierons en particulier les délais des examens suivants :

Scanner : le tomodensitomètre (souvent appelé « scanner ») est une technique d'imagerie médicale qui consiste à calculer une reconstruction 3D des tissus à partir d'une analyse tomographique obtenue en soumettant le patient au balayage d'un faisceau de rayons X ;

IRM : le scanner IRM désigne l'appareil d'imagerie par résonance magnétique ; cette méthode repose sur le principe de la résonance magnétique nucléaire et n'utilise pas de rayons X ni de traceur radioactif, ce qui en fait une technique de choix pour visualiser les organes de façon non-invasive ;

Échocardiographie transthoracique/transœsophagienne : l'échographie est une technique d'imagerie médicale employant les ultrasons. L'échographie cardiaque transœsophagienne (ETO) consiste à introduire, après anesthésie locale, un endoscope souple doté d'un capteur dans l'œsophage du patient. Le but principal de cet examen est d'obtenir des images des zones du cœur difficilement accessibles par la technique plus courante de l'échocardiographie transthoracique (ETT).

Holter tensionnel ECG ou R-test : le holter tensionnel (holter TA) est un dispositif portable permettant l'enregistrement en continu de la pression artérielle pendant au moins vingt-quatre heures. Le holter ECG permet d'identifier les anomalies du rythme cardiaque. Le holter R-test possède un fonctionnement similaire, mais il permet un enregistrement plus long, jusqu'à une semaine.

6.3 Description du système

La grande majorité des patients victimes d'AVC arrivent à l'hôpital aux urgences. Un premier passage au scanner et/ou à l'échographie cardiaque dans le pavillon d'urgence est prévu pour identifier le type d'AVC (accident ischémique ou hémorragique).

Après les premiers soins, les patients sont admis dans les services prenant en charge les AVC. Les patients atteints d'AVC sont actuellement répartis dans deux pavillons de l'hôpital Bellevue (neurologie et médecine interne). Les patients sont admis indifféremment dans un pavillon ou dans l'autre en fonction de l'occupation des lits. Un premier bilan est réalisé par l'interne, le patient est ensuite examiné minutieusement par un médecin sénior. L'AVC doit être traité durant le séjour du patient pour prévenir les récidives. Il s'agit également de surveiller le patient, d'identifier les causes et les facteurs risques. Pour cela un certain nombre d'examens listés section 6.2 sont requis (scanner, IRM, échographie, holter).

Pour obtenir un rendez-vous, le médecin doit remplir et signer un bon, qui est faxé au service prestataire. Les secrétaires du service traitent les demandes et attribuent ensuite des rendez-vous aux patients des différents services. Nous supposons les règles de gestion des rendez-vous au sein des services prestataires inconnues. Les médecins peuvent parfois être amenés à prendre rendez-vous directement par téléphone dans les cas urgents. Les demandes urgentes sont traitées le plus tôt possible, en insérant les examens urgents entre deux rendez-vous.

Selon la gravité de l'AVC, le patient peut être autorisé à sortir avant réception des résultats d'examens si son état le permet. Dans le cas inverse, la durée de séjour du patient est fonction du délai d'obtention de ces résultats. La récupération peut être rapide ou non, tout dépend des zones du cerveau qui ont été atteintes. En fin de séjour le patient peut rentrer chez lui ou être transféré vers un établissement spécialisé pour sa rééducation. La date du transfert dépend aussi de la date de réception des examens, la coordination entre ces contraintes est difficile.

Plusieurs visites sont organisées durant la semaine par plusieurs médecins. Lors de cette visite, le médecin est informé par l'interne de l'état du patient, examine les résultats d'examens, prend des décisions. Chaque patient possède un dossier contenant une fiche de suivi d'examens et d'observations.

6.4 Données

Afin de mettre en œuvre la modélisation et la simulation de l'unité cardio-vasculaire, nous avons collecté anonymement les données des patients atteints d'AVC sur quatre mois (du premier janvier au 30 avril 2008). Ces données se répartissent en trois catégories distinctes : (i) les données temporelles liées au parcours patient, (ii) les données liées au dossier patient, et (iii) les données liées aux délais de réception des examens. L'accès à ces données permet l'observation de tendances en comparant temps de séjour et délais d'obtention d'examens.

6.4.1 Nature des données

Les données temporelles liées au parcours patient ont été relevées sur le terrain grâce aux observations réalisées. Il s'agit essentiellement de durées d'activités (visites, repas, etc.), permettant la simulation du parcours patient. Ces donnéees ont un faible impact sur l'étude car elle ne déterminent pas la durée du séjour patient. L'ensemble des mesures effectuées apparaissent dans les différentes vues du modèle proposé dans la section 6.5.

Les données relevées dans le dossier patient sont les dates d'entrée et de sortie du service ; les données liées aux délais de réception des examens sont les plus difficiles à obtenir, car elles dépendent de multiples facteurs et sont liées à plusieurs services. Étant donné qu'il est impossible d'accéder au dossier médical du patient après son départ, ces données ont été collectées à partir du premier janvier 2008. Nous avons ainsi formé une base de données contenant les informations suivantes : (i) nombre et types d'examens demandés par patient, (ii) date de chaque demande, (iii) date de chaque rendez-vous, et (iv) nature du rendez-vous demandé (régulier, urgence ou récupéré). Un rendez-vous récupéré est un rendez-vous déjà obtenu pour le service réattribué à un patient différent.

6.4.2 Analyse statistique préliminaire

Le relevé et l'observation de ces données nous permet d'établir un lien entre les examens prévus, la pathologie du patient et la gravité de son cas. Afin d'éviter la modélisation du fonctionnement de tous les services prestataires, nous proposons dans un premier temps une étude statistique de ces données. Dans un second temps, nous proposerons une loi permettant de générer des délais d'attente proches de la réalité, permettant de simuler le fonctionnement du système dans plusieurs conditions.

Un certain nombre d'observations générales peuvent être réalisées sur l'échantillon de données, regroupées dans la table 6.1. Durant les quatre mois de l'observation, 109 patients ont été admis en neuro-vasculaire et 295 examens ont été demandés. La quasi-totalité des entrées sont des urgences (non-programmés). La durée moyenne de séjour mesurée est de 13,8 jours. Les délais moyens d'attente d'examens sont déclinés selon la nature du rendez-vous (régulier, urgence ou récupéré). Nous constatons que les délais moyens sont très faibles lorsque la demande est réalisée en urgence, contrairement au délai d'une demande réalisée de manière régulière. La réutilisation de rendez-vous annulés offre une certaine souplesse ; néanmoins cette pratique est risquée et irrégulière.

Nature de l'examen	**Régulier**		**Urgence**		**Récupéré**	
	Délai moyen	Proportion	Délai moyen	Proportion	Délai moyen	Proportion
IRM	38,6 jours	55,94 %	3,8 jours	25,42 %	5,5 jours	18,64 %
Scanner	12,8 jours	46,88 %	4,4 jours	40,62 %	5,2 jours	12,5 %
Holter TA-ECG	10,3 jours	88,16 %	3,5 jours	2,63 %	5,1 jours	9,21 %
Holter TA-R-test	10,9 jours	92,86 %	-	0 %	1 jour	7,14 %
ETT+ETO	13,1 jours	76,6 %	7 jours	2,13 %	7,9 jours	21,27 %

TAB. 6.1 – Observations générales sur un échantillon de 295 examens pour 109 patients

La figure 6.2 représente la demande de rendez-vous pour les IRM, scanner et holters TA-ECG par semaine. Nous constatons que les demandes pour ces trois examens sont relativement irrégulières : les demandes d'IRM varient entre 0 et 7 par semaine, tandis que les demandes de scanners varient entre 0 et 4 par semaine. Les holters TA-ECG, plus demandés, sont requis 1 à 8 fois par semaine. Ces données nous permettrons de proposer une ou plusieurs organisation(s) alternative(s) pour la gestion des rendez-vous.

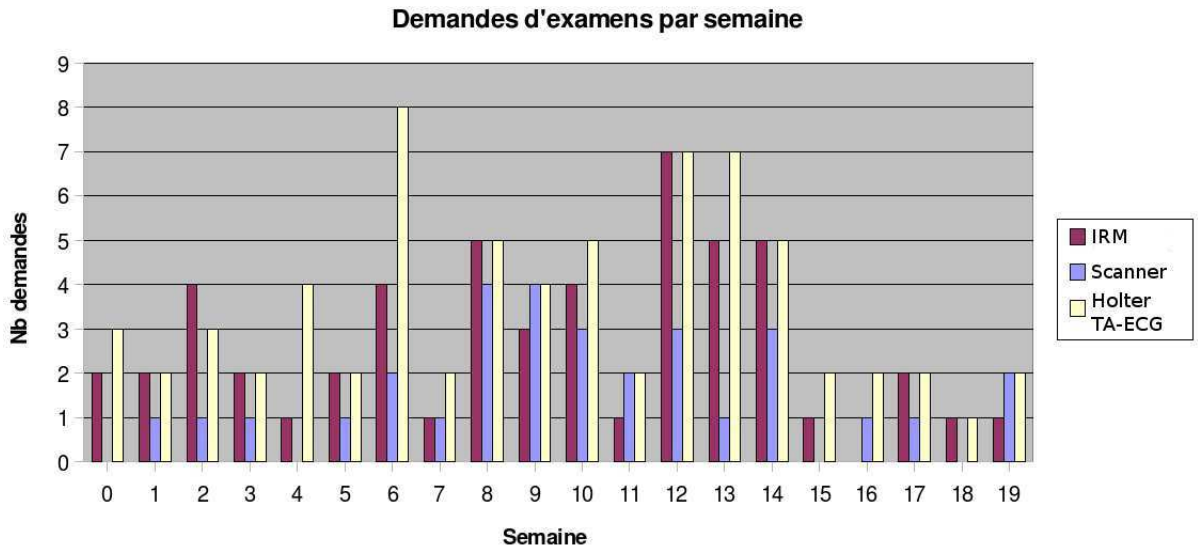


FIG. 6.2 – Nombre d'examens demandés par semaine

Nous pouvons estimer à partir de cet échantillon un certain nombre de lois permettant de modéliser le schéma d'arrivée des patients dans le service ainsi que les délais d'obtention des examens. Ces valeurs sont injectées dans le modèle présenté dans la section 6.5.

6.4.3 Arrivées des patients

Les dates d'arrivée des patients atteints d'AVC dans l'unité neuro-vasculaire entre janvier et mai 2008 nous permettent d'identifier la distribution la plus adéquate. La distribution exponentielle est choisie pour modéliser la distribution des temps inter-arrivées.

6.4.4 Délais d'attente

Une analyse de l'échantillon de données permet de déduire un certain nombre de distributions permettant de modéliser les délais d'attente pour les différents examens selon le type de demande. La table 6.2 regroupe l'ensemble de ces lois.

Nature de l'examen	Régulier	Urgence	Récupéré
IRM	$NORM(38; 13, 4)$	$WEIB(4, 29; 0, 846)$	$WEIB(8, 98; 1, 23)$
Scanner	$0, 5 + 31BETA(0, 58; 0, 657)$	$LOGN(2, 16; 2, 14)$	$LOGN(2, 26; 2.33)$
Holter TA-ECG	$0, 5 + ERLA(5, 06; 2)$	1	$UNIF(1; 6)$
Holter TA-R-test	$1, 5 + 49BETA(0, 628; 1, 96)$	1	1
ETT+ETO	$1, 5 + ERLA(6, 46; 2)$	1	$0, 5 + GAMMA(5, 02; 1.37)$

TAB. 6.2 – Distributions modélisant le délai d'attente d'un examen

6.5 Modélisation du système

Nous proposons une modélisation du parcours patient au sein de l'unité neuro-vasculaire grâce à la plate-forme medPRO. Les activités liées au parcours patient ont pour la plupart une durée négligeable comparée au délai d'obtention d'un examen quelconque ; cependant la modélisation du parcours et du séjour du patient offre une meilleure compréhension du fonctionnement du service : la compréhension des pathologies traitées, de la prise en charge médicale du patient et de l'intérêt de chaque examen est importante pour proposer des solutions viables. Les ressources impliquées dans le processus sont le chef de clinique, le médecin spécialisé neuro-vasculaire et l'interne. La figure 6.3 présente le parcours patient (vue processus) ainsi que les classes déclarées (vue organisation) sous medPRO.

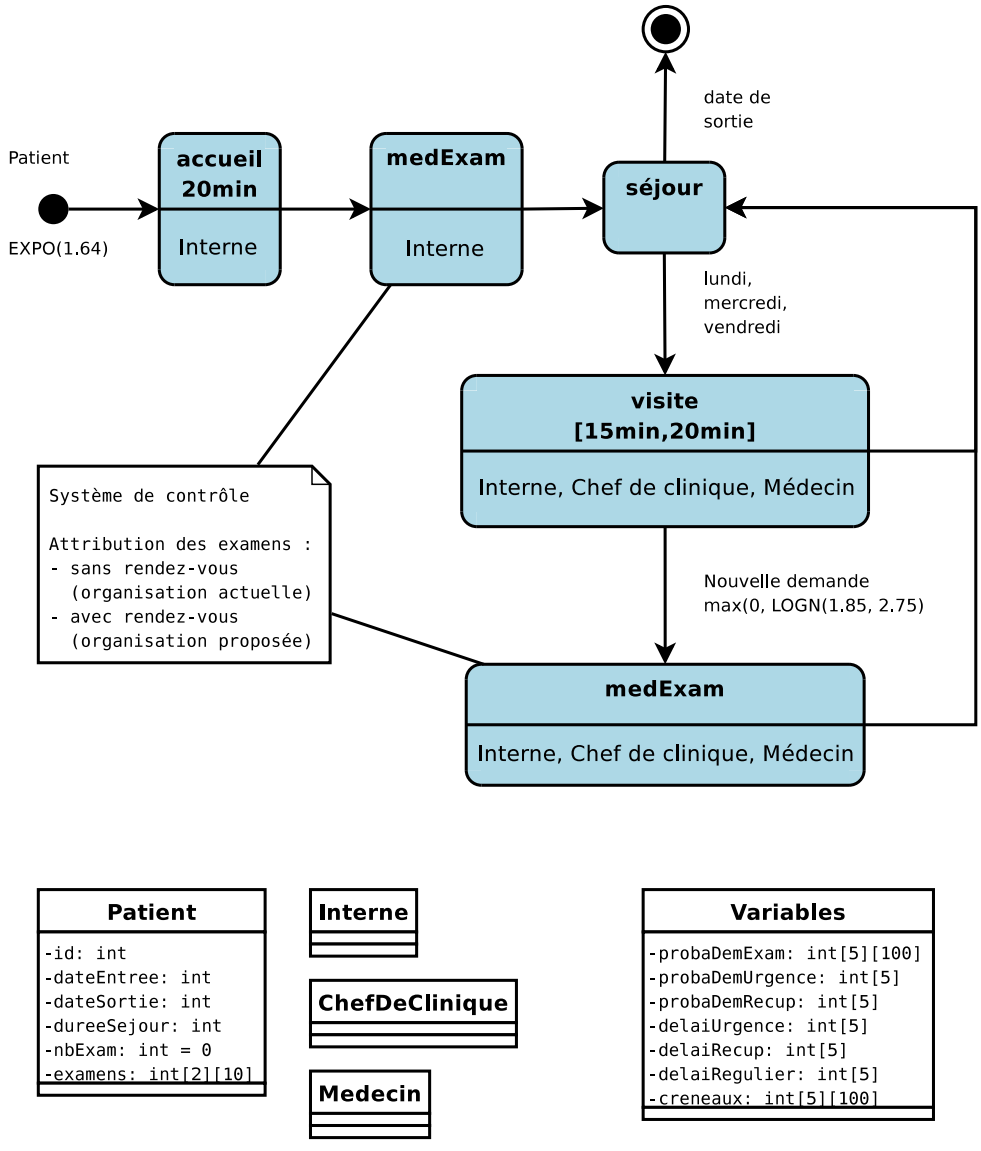


FIG. 6.3 – Modélisation medPRO de l'unité neuro-vasculaire

Chaque nouveau patient arrive dans l'unité neuro-vasculaire depuis le pavillon d'urgence. L'interne accueille le patient et réalise les premières demandes d'examen. Le séjour du patient commence ensuite ; trois visites sont programmées par semaine afin de mesurer l'évolution de l'état de santé du patient. De nouveaux examens peuvent être requis et la date de sortie du patient est éventuellement modifiée. Enfin, le patient sort du système lorsque la date de sortie programmée est atteinte.

6.6 Scénarios de simulation

Plusieurs scénarios de simulation ont été mis en œuvre pour valider les lois de probabilité d'une part, et d'autre part pour tester un certain nombre d'organisations alternatives. Le nombre de réplications est fixé à 20. La longueur de chaque réplication est fixée à un an.

Organisation actuelle

Le système correspondant à l'organisation actuelle a été modélisé afin de valider les lois de probabilités choisies. Les statistiques recueillies dans le rapport de simulation ont été comparées aux statistiques présentées dans la section 6.4.

Organisation alternative : la contractualisation

La contractualisation des interactions entre le(s) service(s) pretataire(s) et le service de neurologie permettrait une meilleure coordination des rendez-vous pour les patients. Aucun accord n'existe à l'heure actuelle entre ces services, expliquant les grandes disparités entre les délais d'attente pour un rendez-vous selon le patient, la période de l'année, etc. La mise en place d'un contrat permettrait d'établir une relation fiable pour la prise de rendez vous.

Dans ce cadre, nous proposons la réservation d'un créneau dédié au service de neurologie dans chacun des services liés par ce contrat. Une telle contractualisation permet de garantir un rendez-vous selon une échéance régulière. De plus, le rendez-vous peut être programmé judicieusement par rapport aux horaires des visites des médecins. Côté prestataire, la présence du patient à ce rendez-vous est garantie ; de plus, la programmation de rendez-vous dédiés permet de diminuer les temps de traitement liés à la prise en charge de la demande (réception et traitement d'un fax ou d'un appel, programmation du rendez-vous).

Les scénarios de simulation sont définis dans la table 6.3. Trois scénarios sont prévus, permettant le test de différentes configurations pour la gestion des rendez-vous. Les deux premiers scénarios permettent de tester deux propositions de contractualisation différentes. La fréquence des rendez-vous planifiés dans le scénario 1 correspond à la fréquence des demandes observées lors de la collecte de données. Le scénario 2 correspond à une organisation proposée par le personnel hospitalier de l'unité neuro-vasculaire, avec une fréquence légèrement supérieure. Enfin, le scénario 3 vise à tester le fonctionnement du service lorsque la capacité d'accueil sera doublée.

Nature de l'examen	**Scénario 1**	**Scénario 2**
Nombre de patients	1 :1	1 :1
IRM	lundi (2), vendredi (1)	lundi (2), mercredi (1), vendredi (1)
Scanner	lundi (1)	lundi (1), vendredi (1)
Holter TA-ECG	lundi (2), vendredi (2)	lundi (2), mercredi (1), vendredi (2)
Holter TA-R-test	mercredi (1)	lundi (1), vendredi (1)
ETT+ETO	lundi (1), vendredi (1)	lundi (2), mercredi (1), vendredi (1)

Nature de l'examen	**Scénario 3**
Nombre de patients	2 :1
IRM	6 par semaine
Scanner	lundi (1), mercredi (1), vendredi (1)
Holter TA-ECG	8 par semaine
Holter TA-R-test	lundi (1), mercredi (1), vendredi (1)
ETT+ETO	lundi (2), mercredi (1), vendredi (1)

Tab. 6.3 – Scénarios de simulation

6.7 Simulation et résultats

Pour réaliser une comparaison pertinente, nous avons mesuré pour chaque scénario et pour chaque type d'examen la proportion de rendez-vous pris hors-réservation, la proportion de rendez-vous annulés, ainsi que le nombre de patients accueillis dans l'unité et la durée moyenne de séjour observée.

		Sys. actuel	**Scénario 1**	**Scénario 2**	**Scénario 3**
Nombre de patients		301	296	294	603
Durée moyenne de séjour		12,56 ± 1 j	8,67 ± 1 j	5,29 ± 1 j	7,31 ± 1 j
Hors réserv.	IRM	-	12,46 %	3,14 %	8,87 %
	Scanner	-	33,52 %	2,86 %	4,91 %
	Holter TA-ECG	-	7,81 %	1,62 %	4,77 %
	Holter TA-R-test	-	33,25 %	3,71 %	7,44 %
	ETT+ETO	-	26,93 %	5,63 %	22,31 %
Annulations	IRM	-	11,12 %	26,85 %	6,29 %
	Scanner	-	19,52 %	45,96 %	22,89 %
	Holter TA-ECG	-	12,93 %	28,44 %	10,06 %
	Holter TA-R-test	-	19,52 %	39,9 %	17,76 %
	ETT+ETO	-	7,74 %	32,67 %	3,12 %

Tab. 6.4 – Résultats de simulation de l'unité neuro-vasculaire

Les résultats présentés dans la table 6.4 indiquent une diminution flagrante de la durée moyenne de séjour dans les trois scénarios de simulation, même dans le cas où la capacité d'accueil de l'unité serait doublée (S3). Nous observons logiquement un meilleur remplissage des créneaux réservés dans S2 par rapport à S1, mais aussi une augmentation flagrante des annulations. Le scénario S3 permet d'observer un fonctionnement grandeur nature, et nous pouvons observer un équilibre assez bon entre la proportion d'examens hors réservation et la proportion d'annulations. La durée moyenne de séjour associée à S3 avoisine la semaine et concorde avec les normes données par la T2A (9 jours pour un séjour dans une unité neuro-vasculaire).

Nous ajoutons un coefficient d'erreur à plus ou moins un jour concernant la DMS ; ce coefficient reflète l'approximation sur le calcul de la durée de séjour vis à vis du diagnostic médical du médecin. En effet, il nous est impossible de modéliser précisément les aléas pathologiques dûs à la prise en charge d'un patient.

6.8 Synthèse

Nous avons proposé et testé plusieurs organisations mettant en œuvre une contractualisation de la prise de rendez-vous pour les patients atteints d'AVC. Nous avons pu démontrer par la simulation que la planification de rendez-vous permet de diminuer le temps de séjour du patient de quatre à sept jours selon l'organisation adoptée. D'autres organisations peuvent également être testées, selon les souhaits des responsables des différents services prestataires.

Une planification semaine par semaine peut être établie grâce aux méthodes utilisées dans le milieu du génie industriel. Une étude sur ce point pourrait donner lieu à la création d'un outil informatique permettant la planification par la cadre-infirmière de l'unité neuro-vasculaire semaine après semaine.

L'unité neuro-vasculaire du CHU de Saint-Étienne a pu être modélisée et simulée grâce à la plate-forme medPRO, offrant une visualisation claire des processus opérationnel et décisionnels mis en jeu. La séparation de l'opérationnel, de l'informationnel et du décisionnel offre une plus grande souplesse dans la modélisation et la simulation de scénarios différents.

Chapitre 7

La pharmacie

Ce chapitre est consacré à l'étude du problème de transport de médicaments au sein du CHU de Saint-Étienne. Chaque unité médicale de l'hôpital possède un chariot de médicaments qui doit être transporté chaque semaine par les transporteurs jusqu'à la pharmacie centrale pour être réapprovisionné par les préparateurs. Le transport est réalisé à pied, en tracteur ou en camion selon la localisation de l'unité. Le problème consiste à créer un planning de transport et de réapprovisionnement sur une semaine en équilibrant la charge des transporteurs et des préparateurs. Une méthode de programmation linéaire en deux étapes est proposée et intégrée dans la plate-forme medPRO. Les planning ainsi créés sont simulés sous medPRO pour tester leur robustesse dans des conditions stochastiques. La méthodologie de cette étude de réingénierie est également présentée.

7.1 Introduction

Le Centre Hospitalier Universitaire de Saint-Étienne est un complexe hospitalier doté de 1.921 lits et de 56 services de soins répartis sur cinq sites géographiques. En raison de la taille du complexe, il est nécessaire de contrôler rigoureusement l'organisation de l'ensemble des processus médicaux de prise en charge des patients. Dans cette optique, le CHU est impliqué depuis 2001 dans un vaste projet de réingenierie appelé *Modernisation du CHU* visant à réduire les coûts en réunissant les cinq hôpitaux actuels : la plupart des activités du CHU seront regroupées sur le site le plus important (hôpital Nord), tandis que les activités de long séjour seront regroupées sur un second site (hôpital Bellevue).

Le CHU de Saint-Étienne possède actuellement deux pharmacies : la pharmacie centrale située dans l'hôpital Nord et une pharmacie annexe située dans l'hôpital Bellevue. La réorganisation du CHU aura un impact important sur l'organisation de la pharmacie ; en effet, il a été décidé de regrouper les deux pharmacies actuelles sur l'hôpital Nord. La pharmacie de Bellevue fermera ses portes en 2008 et l'ensemble des activités de préparation et de livraison de médicaments seront assurées par la pharmacie centrale de l'hôpital Nord. Toutes les ressources (humaines et matérielles) seront également regroupées dans la nouvelle pharmacie centrale. Ainsi l'organisation complète de la livraison des médicaments doit être remise en cause pour assurer l'approvisionnement depuis la nouvelle pharmacie centrale vers l'ensemble des services du CHU.

L'objectif de cette étude est de proposer une approche pour optimiser le processus de livraison des médicaments au sein du CHU de Saint-Étienne.

Le processus de livraison des médicaments a été peu abordé dans la littérature. Wong *et al.* (2003) ont utilisé la simulation afin de montrer l'efficacité de l'implémentation d'un système informatisé pour le processus administratif concernant la rédaction et la délivrance d'ordonnances de la part des médecins. Deux modèles ont été créés, l'un décrivant le système actuel, l'autre le système en prévision. Les critères de com-

paraisons sont le taux d'erreur en matière de prescription et le niveau requis de ressources. La modélisation des processus a été réalisée grâce à la méthodologie IDEF0, et le modèle de simulation implémenté sous MedModel. Outre la simplification du processus opératoire, les performances sont améliorées et les taux d'erreurs concernant les prescriptions est réduit. La simulation a également été utilisée dans (Anderson, 2002) pour mesurer l'impact des ordonnances informatisées sur le taux d'erreurs de prescription dues aux médecins. Finalement Spry et Lawley (2005) ont créé un modèle de simulation afin d'analyser l'impact de différentes stratégies de planification au sein de la pharmacie interne d'un hôpital. Le modèle permet d'estimer les effets de changements de planification du travail des employés sur la durée entre le moment où la demande est reçue et le moment où les médicaments sont prêts. Le flux concernant le traitement d'une demande classique est considéré, ainsi que les tâches additionnelles réalisées par les employés de la pharmacie. À terme, un logiciel de simulation a pu être développé, permettant aux pharmaciens de tester l'impact de changements organisationnels au sein de leur service.

La principale contribution de ce travail consiste en l'approche proposée pour organiser et réorganiser les processus de livraison de médicaments d'un complexe hospitalier. Cette approche repose sur la modélisation complète des flux de livraison et tient compte de l'organisation du centre hospitalier. En particulier, le transport de médicaments et l'utilisation des ressources sont conjointement optimisés tout en tenant compte les contraintes organisationnelles du CHU.

Ce chapitre est organisé de la manière suivante : une description générique du processus de livraison de médicaments est proposée section 7.2. Une approche en deux étapes est décrite pour l'optimisation du processus de livraison section 7.3. La section 7.4 décrit le processus de réingénierie dans lequel l'approche proposée est appliquée pour déterminer l'organisation optimale de la nouvelle pharmacie. Une modélisation du système pour la simulation est également proposée sous medPRO. Finalement, la section 7.5 dresse le bilan de cette étude. Cette étude a été présentée dans (Augusto et Xie, 2008).

Nous aimerions également remercier très chaleureusement Mme Marie-Claire Veyre, Mme Françoise Lorca ainsi que l'ensemble du personnel de la pharmacie du CHU de Saint-Étienne pour leur disponibilité et leur patience durant cette étude.

7.2 Description du problème

7.2.1 Description du processus de livraison des médicaments

Le processus de livraison des médicaments considéré peut être décrit comme suit : avant tout l'intégralité des médicaments est livrée quotidiennement à la pharmacie centrale par des transporteurs externes pendant la semaine. Ces médicaments sont classés et rangés dans la salle de préparation principale de la pharmacie. Chaque unité du CHU possède un chariot contenant les médicaments liés à la spécialité de l'unité. Ce chariot est transporté à la pharmacie par des transporteurs et réapprovisionné une fois par semaine par les préparateurs ; le chariot est ensuite ramené dans son unité. La pharmacie livre également des médicaments nominatifs pour certains patients : les prescriptions sont envoyées à la pharmacie par les médecins et traitées dans l'après-midi par les préparateurs ; les prescriptions urgentes sont traitées immédiatement. Ces médicaments sont aussi transportés par un transporteur. Enfin, certains médicaments sont également délivrés à des patients venant de l'extérieur.

La figure 7.1 offre une représentation simplifiée du processus de livraison de médicaments. Trois types de services sont identifiés : les services livrés à pied, les services livrés en tracteur, et les services localisés dans d'autres bâtiments livrés par camion.

Dans cette organisation, les médicaments sont stockés dans un chariot dans chaque unité ; les préparateurs restent dans la pharmacie centrale pour inventorier les chariots et les réapprovisionner, tandis que les transporteurs vont chercher et ramènent ces même chariots. Ce type d'organisation présente

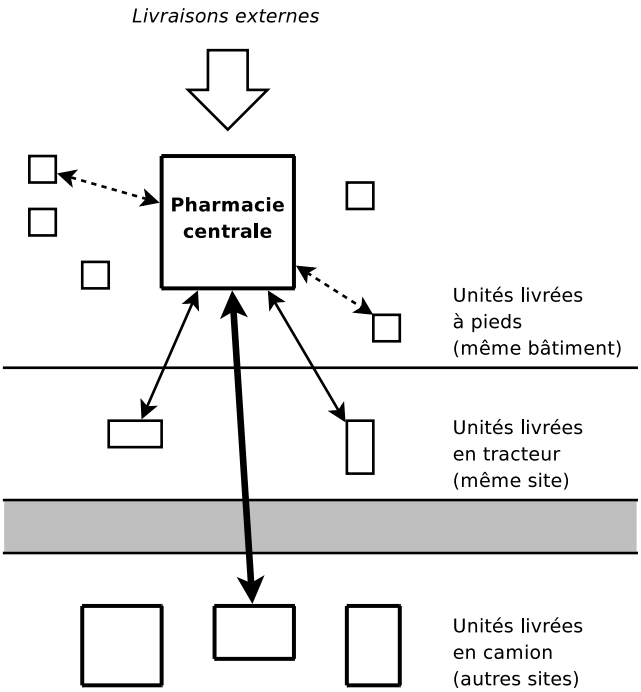


FIG. 7.1 – Livraison des médicaments à pied, en tracteur et en camion

plusieurs avantages : (i) les infirmières dans les services n'ont pas à gérer le stock de médicaments, (ii) les médicaments périmés sont rapidement identifiés et jetés, et (iii) l'inventaire est réalisé directement dans la pharmacie, permettant un réapprovisionnement plus rapide. Trois types de ressources humaines sont considérées dans ce système :

- **Pharmaciens :** les pharmaciens ont l'expertise nécessaire pour approuver ou rejeter certaines prescriptions (incompatibilité entre deux médicaments par exemple). Ils gèrent également le stock de médicaments et examinent les nouveaux marchés.
- **Préparateurs :** les préparateurs constituent la principale ressource de la pharmacie. Ils sont chargés de l'approvisionnement des chariots, de la fabrication de médicaments et de la dotation nominative en médicaments.
- **Transporteurs :** les transporteurs sont chargés d'aller chercher et de ramener les chariots dans les services de soins de l'hôpital.

7.2.2 Problème de transport

Chaque unité du centre hospitalier possède un chariot ; les unités sont réparties sur l'ensemble des bâtiments du CHU et sur plusieurs étages. Les unités médicales sont regroupées en services. L'horizon de planification pour le réapprovisionnement (généralement une semaine) est divisé en périodes élémentaires (généralement une demi-journée). Pour chaque période élémentaire, le nombre de préparateurs disponibles est connu. La durée nécessaire pour le réapprovisionnement de chaque chariot est également connue. Au moins un chariot doit rester dans chaque service à tout moment pour assurer la disponibilité des médicaments pendant le réapprovisionnement. Cette contrainte est appelée « contrainte de disponibilité ».

Le problème est décrit comme suit. Selon la localisation de chaque unité, nous savons si le chariot associé doit être transporté à pied, en tracteur ou en camion. Concernant le transport à pied, la durée nécessaire au transport est connue pour chaque unité. Le transport en tracteur ou en camion est réalisé bâtiment par bâtiment : tous les chariots d'une même tournée réalisée en tracteur ou en camion ap-

partiennent au même bâtiment. Chaque tracteur ou camion peut transporter un nombre maximum de chariots à la fois lors d'une tournée. La durée totale de chaque tournée comprend : (i) la durée totale nécessaire pour amener tous les chariots de la tournée jusqu'au véhicule, (ii) la durée de transport depuis le bâtiment considéré jusqu'au bâtiment de la pharmacie, et (iii) la durée totale nécessaire pour amener tous les chariots à la pharmacie.

Pour chaque période élémentaire, le nombre de transporteurs, de tracteurs et de camions disponibles est connu. La plage horaire de travail de chaque travailleur est également connue.

Nous cherchons ainsi à équilibrer la charge de travail des préparateurs sur l'horizon de planification en tenant compte des contraintes de capacité des préparateurs et des transporteurs et en respectant la contrainte de disponibilité.

7.3 Résolution du problème de transport

Le problème de transport peut être décrit de la manière suivante : construire des tournées de ramassage et de livraison de chariots et répartir ces tournées sur les périodes élémentaires disponibles en équilibrant la charge des transporteurs et des préparateurs. Le problème de tournée de véhicules étant hautement combinatoire, ce problème est trop complexe pour être résolu de manière exacte. Nous proposons ainsi une approche en deux étape :

1. Contruire les tournées de ramassage pour les transporteurs qui conduisent un tracteur ou un camion en respectant les contraintes de disponibilité et de capacité. Chaque tournée réalisée à pied correspond à un chariot transporté à pied.
2. Répartir les tournées obtenues sur les périodes élémentaires de l'horizon de planification pour équilibrer conjointement les charges des transporteurs et des préparateurs.

7.3.1 Génération des tournées de ramassage

Un modèle linéaire en nombres entiers est proposé pour la construction des tournées pour les transporteurs en tracteur ou en camion. Chaque bâtiment de l'hôpital (les bâtiments livrés à pieds ne sont pas pris en compte) est considéré séparément. Soit U l'ensemble des chariots (ou des unités) du bâtiment considéré. Soit U_k, $k \in \{1, \ldots, K\}$, l'ensemble des chariots du service k. Nous avons $U = U_1 \cup U_2 \cup \cdots \cup U_K$.

Soit N la capacité du moyen de transport (tracteur ou camion, selon le bâtiment considéré). Soit J le nombre maximum de tournées nécessaires pour le bâtiment. Nous avons $J \leq |U|$.

La variable de décision y_{ij} vaut 1 si le chariot i est affecté à la tournée j, 0 sinon. La variable de décision z_j vaut 1 si la tournée j est ouverte, 0 sinon. Le problème s'écrit donc de la manière suivante.

$$\text{Minimiser} \sum_{j=1}^{J} z_j \tag{7.1}$$

Sous les contraintes suivantes :

$$\sum_{j=1}^{J} y_{ij} = 1, \ \forall i \in U \tag{7.2a}$$

$$\sum_{i \in U} y_{ij} \leq N, \ \forall j \in \{1 \ldots J\} \tag{7.2b}$$

$$\sum_{i \in U_k} y_{ij} \leq |U_k| - 1, \ \forall j \in \{1 \ldots J\} \tag{7.2c}$$

$$z_j \geq y_{ij}, \ \forall i \in U, \ \forall j \in \{1 \ldots J\} \tag{7.2d}$$

Nous cherchons à minimiser le nombre de tournées (7.1). La contrainte (7.2a) assure que chaque chariot est affecté à une tournée exactement. La contrainte (7.2b) est la contrainte de capacité du moyen de transport. La contrainte (7.2c) est la contrainte de disponibilité et la contrainte (7.2d) assure que chaque chariot est affecté à une tournée ouverte.

7.3.2 Génération du planning de réapprovisionnement

Le modèle mathématique pour la résolution de ce problème est défini de la manière suivante. Soient L l'ensemble des S tournées générées précédemment, $LF \subset L$ le sous-ensemble de tournées en tracteur et $LT \subset L$ le sous-ensemble de tournées en camion. Soit T le nombre de périodes.

Pour chaque tournée i, nous définissons sa durée de transport q_i et sa durée de réapprovisionnement globale p_i dans la pharmacie. Pour chaque période t, nous connaissons le nombre de préparateurs n_t, le nombre de transporteurs m_t, le nombre de tracteurs v_t et de camions w_t disponibles. Soit S_t (resp. T_t) la durée de la plage de travail des préparateurs (resp. des transporteurs) par période.

Enfin, Ω_r avec $r \in \{1 \dots R\}$ est l'ensemble des tournées qui ne peuvent être affectées à la même période à cause de la contrainte de disponibilité. L'ensemble Ω_r correspond aux tournées qui ne peuvent être affectées en même temps à la même période.

La variable de décision x_{it} vaut 1 si la tournée i est affectée à la période t. Le problème s'écrit donc de la manière suivante.

$$\text{Minimiser } C \tag{7.3}$$

Sous les contraintes suivantes :

$$C \geq \sum_{i \in L} \frac{p_i}{n_t S_t} x_{it},\ \forall t \in \{1 \dots T\} \tag{7.4a}$$

$$\sum_{i \in \Omega_r} x_{it} \leq |\Omega_r| - 1,\ \forall t \in \{1 \dots T\},\ \forall r \in \{1 \dots R\} \tag{7.4b}$$

$$\sum_{i \in L} q_i x_{it} \leq m_t T_t,\ \forall t \in \{1 \dots T\} \tag{7.4c}$$

$$\sum_{i \in LF} q_i x_{it} \leq v_t T_t,\ \forall t \in \{1 \dots T\} \tag{7.4d}$$

$$\sum_{i \in LT} q_i x_{it} \leq w_t T_t,\ \forall t \in \{1 \dots T\} \tag{7.4e}$$

$$\sum_{t=1}^{T} x_{it} = 1,\ \forall i \in L \tag{7.4f}$$

Nous cherchons à minimiser la charge des préparateurs (7.3), (7.4a). La contrainte (7.4b) est la contrainte de compatibilité entre tournées. La contrainte (7.4c) est la contrainte de capacité des transporteurs. Les contraintes (7.4d) et (7.4e) sont les contraintes de disponibilité des tracteurs et des camions respectivement, et la contrainte (7.4f) assure que chaque tournée est affectée à une période exactement.

7.4 Étude de la pharmacie du CHU de Saint-Étienne à l'aide de medPRO

La méthode décrite dans la section 7.3 pour la résolution du problème de transport a été appliquée à la pharmacie du CHU de Saint-Étienne. Les deux pharmacies existantes dans le système actuel (as-is) de l'hôpital Nord et de l'hôpital Bellevue vont être réunies en une seule pharmacie dans le système futur

(to-be) dans l'hôpital Nord. La localisation de toutes les unités médicales va également changer. Les médicaments seront livrés directement aux unités à Nord et à Bellevue quotidiennement.

La figure 7.2 présente une carte de la ville de Saint-Étienne pour une meilleure visualisation de l'emplacement des différentes sites qui constituent le CHU. Les quatre hôpitaux actuels sont l'hôpital Nord, l'hôpital Bellevue, l'hôpital de la Charité et l'hôpital de Saint-Jean-Bonnefonds. Dans le système to-be, seuls l'hôpital Nord et l'hôpital Bellevue resteront ouverts.

FIG. 7.2 – Localication des sites du CHU de Saint-Étienne

7.4.1 Processus de réingénierie

Les processus de l'actuelle pharmacie ont été observés pendant deux mois afin de capter les particularités du système et de détecter d'eventuels problèmes d'organisation. Les réunions avec l'équipe de la pharmacie ont été planifiées toutes les deux semaines afin de valider notre vision du système et la précision des données collectées. Ces réunions ont également été utiles pour corriger notre modèle et mettre d'accord les membres de l'équipe entre eux. Plusieurs points faibles ont été détectés dans la pharmacie actuelle :

- il n'existe aucun planning de réapprovisionnement établi à l'avance ;
- le travail des préparateurs et des transporteurs n'est pas équilibré sur la semaine, entraînant la création de disparités d'un jour à l'autre ;
- les plannings des semaines réduites (avec un jour férié) sont établis au jour le jour ; il en va de même pour la gestion des absences.

Nous devons également prévoir les problèmes d'organisation qui se poseront dans le système to-be. Le processus de transport sera profondément modifié pour plusieurs raisons :

- lorsque la pharmacie de Bellevue fermera ses portes, toutes les unités de cet hôpital devront être livrées directement depuis la pharmacie centrale de l'hôpital Nord quotidiennement, entraînant une augmentation du trafic vers ce site ;
- la plupart des unités médicales vont déménager à l'hôpital Nord pendant la réorganisation, rendant la phase de livraison interne dans cet hôpital plus lourde ;

– de nouvelles régulations interdisent la circulation de chariots dans les couloirs « publics » de l'hôpital, les chemins empruntés doivent être modifiés ; les chariots devront être transportés principalement en sous-sol.

7.4.2 Collecte des données

Une base de donnée a été mise en place pour stocker efficacement les données collectées. Les durées d'approvisionnement, les distances entre unités, les plannings de livraison et les caractéristiques de toutes les unités du CHU ont été organisés selon un modèle conceptuel de données approprié. Comme la pharmacie du CHU ne possède aucune base de donnée informatisée, toutes les données temporelles ont dû être chronométrées sur le terrain. La table 7.1 présente les durées mesurées pour l'inventaire et le réapprovisionnement des chariots selon leur type. La distribution uniforme a été sélectionnée pour modéliser ces durées.

Type de chariot	**Durée de l'inventaire (min)**	**Durée du réappro. (min)**
Petit	[15,25]	[25,35]
Moyen	[20,40]	[30,60]
Grand	[35,50]	[40,70]

Tab. 7.1 – Durées d'inventaire et de réapprovisionnement des chariots

Les durées de transport ainsi que les distances entre les différentes unités du CHU ont été collectées et organisées sous forme de matrice. Pour chaque unité du CHU, un chemin idéal a été établi sur une carte en accord avec les transporteurs. Les durées de transport ont été estimées pour les unités qui n'existent pas encore à l'hôpital Nord. Les chariots de la pharmacie to-be sont similaires aux anciens chariots en terme de durée de réapprovisionnement et de poids.

Selon la méthode de transport, nous déterminons la durée associée sur chaque chemin pour chaque tournée en tenant compte de l'étage de chaque unité. Les données recueillies sont partiellement présentées dans la table 7.2. La première colonne indique le nom de l'unité ; les deuxième et troisième colonnes indiquent le bâtiment et l'étage de cette unité. Les quatre dernières colonnes indiquent respectivement la durée de transport depuis la pharmacie (P) jusqu'à l'ascenseur (A), la durée d'ascension, la durée de transport depuis l'ascenseur jusqu'à l'unité (U), et enfin la durée de préparation. Le CHU possède un total de 35 unités dans le même bâtiment que la pharmacie, et 29 unités dans d'autres bâtiments.

Unité	**Bâtiment**	**Niveau**	**P↔E(s)**	**E(s)**	**E↔U(s)**	**Prep.(s)**
RADIOLOGIE	Maternité	-1	43,2	0	0	3000
GYNECOLOGIE	Maternité	-1	57,6	0	0	3000
PEDIATRIE	Maternité	-1	72	0	0	5700
CHIR. GYNAECO.	Maternité	-1	57,6	0	0	5700
CARDIOLOGIE A	Principal	7	18	24	21,6	5700
CARDIOLOGIE B	Principal	7	18	24	36	5700
CARDIOLOGIE C	Principal	7	18	24	36	5700
PNEUMOLOGIE A	Principal	6	18	22	21,6	3000
PNEUMOLOGIE B	Principal	6	18	22	36	5700
PNEUMOLOGIE C	Principal	6	18	22	36	5700
BRONCHOSCOPIE	Principal	6	18	22	21,6	3000
GASTROENTERO. A	Principal	4	18	18	36	5700

Tab. 7.2 – Liste partielle des unités de l'hôpital Nord

Des durées stochastiques ont été introduites dans le modèle pour rester fidèle à la réalité. Le processus d'arrivée des ordonnances urgentes suit une loi de Poisson. Un modèle de simulation du système actuel a été établi pour valider les données collectées. Ce modèle n'est pas reproduit ici pour ne pas surcharger ce chapitre.

7.4.3 Génération du planning d'approvisionnement pour le CHU

La méthode décrite dans la section 7.3 a été appliquée à notre cas d'étude en injectant les données réelles présentées dans la section précédente. Nous avons établi avec les pharmaciens que les ressources disponibles seraient deux transporteurs, un tracteur et un camion. La table 7.3 présente les durées de transport et les charges de travail des préparateurs obtenues par notre méthode pour une semaine régulière. La table 7.4 présente le même type de résultats obtenus en considérant une semaine avec un jour férié.

Demi-journée	**Durée de transport (min)**	**Charge des préparateurs (%)**
Lundi matin	55.5	42.5
Mardi matin	56.7	45
Mardi ap.-midi	57.7	42.8
Mercredi matin	56.7	42.5
Jeudi matin	54.4	42.5
Jeudi ap.-midi	56.6	42.5
Vendredi matin	58.7	42.5
Écart-type	1.4	0.93

TAB. 7.3 – Génération et affection de tournées sur une semaine régulière

Demi-journée	**Durée de transport (min)**	**Charge des préparateurs (%)**
Lundi matin	82.5	68.2
Mardi matin	81.2	65.3
Mardi ap.-midi	82.3	68.3
Mercredi matin	81.3	69.7
Vendredi matin	81.8	66.4
Écart-type	0.47	5.12

TAB. 7.4 – Génération et affectation de tournées sur une semaine avec un jour férié

Notre approche offre une solution avec un bon équilibre des charges pour les transporteurs et les préparateurs sur une semaine. Ces résultats ne sont pas surprenant car le jeu de données réel possède une petite taille. Les résultats globaux pour la semaine amputée d'un jour restent très bons si nous prenons en compte les commentaires des pharmaciens lorsque ces résultats leur ont été présentés. Il faut noter que les charges des préparateurs indiquées dans les tables 7.3 et 7.4 tient uniquement compte des durées de préparation ; les tâches administratives ne sont pas considérées, d'où une charge de travail toujours bien inférieure à 100 %.

7.4.4 Modélisation

Afin de tester la robustesse du planning de transport dans des conditions stochastiques, nous avons développé un modèle de simulation sous medPRO.

La structure de l'outil de modélisation intégré à medPRO est parfaitement adapté à ce type de modèle : la vue processus est utilisée pour modéliser les différents traitements sur les chariots, médicaments, et ordonnances, tandis que la vue ressource permet la modélisation séparée des transports. La synchronisation entre ces vues est intégrée dans les spécification du modèle UML.

Lorsque les transporteurs retournent à la pharmacie centrale, les processus appropriés de la vue chariot/médicament sont activés par synchronisation. Par exemple, lorsqu'un chariot est livré à la pharmacie au début d'une période, un préparateur est requis pour le réapprovisionner. Le chariot est ensuite mis en attente pour être ramené dans son unité à la fin de la période élémentaire. Nous sommes donc capable de modéliser les taux d'occupation des pharmaciens, des préparateurs et des transporteurs. Chaque vue est synchronisée lorsqu'une entité arrive ou quitte la pharmacie.

Nous sommes ainsi en mesure de créer un modèle cohérent structurellement et clair pour chaque intervenant. Les vues processus et ressources du modèle medPRO sont présentées figures 7.3 et 7.4 respectivement.

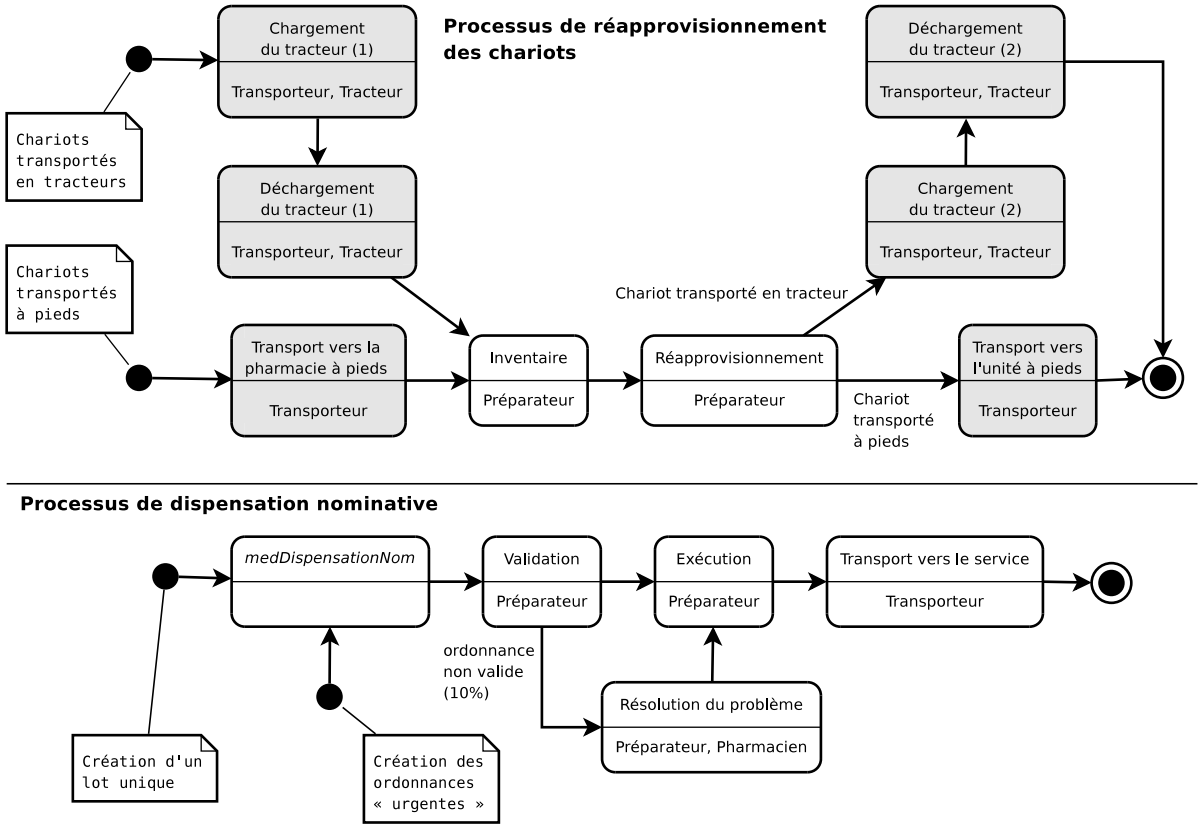


FIG. 7.3 – Vue processus de la pharmacie

Nous avons uniquement représenté le processus de réapprovisionnement des chariots pour le transport à pieds et en tracteur et le processus de dispensation nominative dans la vue processus de la figure 7.3 afin de ne pas surcharger ces pages. Concernant le réapprovisionnement des chariots, une synchronisation avec le transporteur est nécessaire pour l'ensemble des activités de transports, grisées sur la figure. Les activités du transporteur sont représentées figure 7.4 : s'il est à pied, sa mission consiste à aller dans l'unité pour ramener un chariot ; s'il est en tracteur, le transporteur doit aller vers un bâtiment en tracteur, charger ses chariots, et les transporter à la pharmacie pour les décharger. Ces missions débutent à la date prévue par le planning.

Les préparateurs et les pharmaciens sont considérés dans ce modèle comme des ressources sans missions allouées ponctuellement, mais il est tout à fait envisageable de déclarer des missions relative à

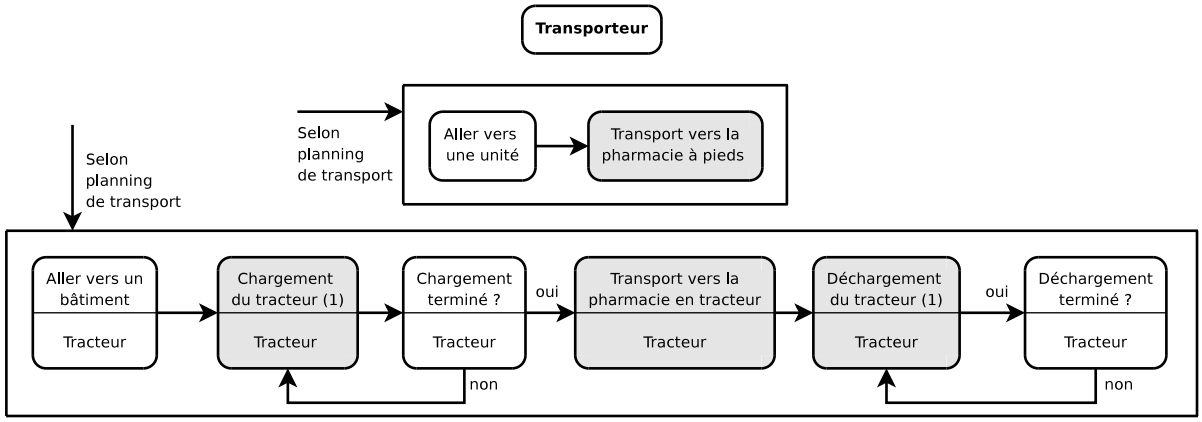


Fig. 7.4 – Vue ressource de la pharmacie

l'approvisionnement des chariots pour les préparateurs afin d'accéder à une modélisation plus fine des processus faisant apparaître les tâches administratives par exemple.

L'utilisation de medPRO permet en outre d'éviter les multiples conversions de données entre le modèle d'optimisation et le modèle de simulation étant donné que l'outil est intégré dans la couche supérieure du logiciel medPRO : la base de données liée au système d'information est commune. Enfin, nous avons la possibilité d'intégrer des modules de décision en rapport avec le système étudié, tel le module d'ordonnancement dédié à la dispensation nominative dont le fonctionnement a été décrit dans la section 5.4.5.

7.4.5 Validation du planning et simulation

Le modèle de simulation a été implémenté en utilisant les plannings générés par notre approche. Vingt réplications d'une longueur de dix semaines ont été lancées. Outre les résultats concernant les plannings de transport, la simulation renvoie diverses statistiques (utilisation des ressources, durée de séjour des chariots dans la pharmacie, durée de traitement des prescriptions urgentes). Les résultats sont présentés dans les tables 7.5, 7.6 et 7.7.

Demi-journée	**Charge moyenne**	**Écart-type**	**Min.**	**Max.**
Lundi matin	2 h 32 min	2 min 35 s	2 h 10 min	2 h 59 min
Mardi matin	2 h 35 min	2 min 54 s	2 h 11 min	3 h 12 min
Mardi a.-midi	2 h 27 min	3 min 10 s	2 h 00 min	3 h 09 min
Mercredi matin	2 h 31 min	2 min 51 s	2 h 03 min	3 h 02 min
Jeudi matin	2 h 31 min	2 min 54 s	2 h 03 min	3 h 09 min
Jeudi a.-midi	2 h 32 min	2 min 51 s	2 h 02 min	3 h 07 min
Vendredi matin	2 h 36 min	1 min 21 s	2 h 19 min	2 h 54 min

Tab. 7.5 – Résultats de la simulation : charge des préparateurs

La charge moyenne des préparateurs sur dix semaines est égale à 2 heures et 32 minutes par demi-journée. L'écart type est inférieur à 5 minutes. La charge maximale de 3 heures par demi-journée est dépassée le jeudi matin de 12 minutes. Ces résultats tiennent compte des demandes urgentes en médicaments, de la réception des patients externes, et d'activités imprévues telle la double vérification d'ordonnances par les pharmaciens.

Le planning établi sur la semaine pour les transporteurs et les préparateurs est correct car les con-

		Transporteur 1				Transporteur 2			
		Ramassage		Livraison		Ramassage		Livraison	
L. matin	Planifié	7 :50	9 :00	10 :20	11 :30	-	-	-	-
	Simulé	7 :50	**8 :53**	10 :20	**11 :21**	-	-	-	-
M. matin	Planifié	7 :50	9 :00	10 :20	11 :30	-	-	-	-
	Simulé	7 :50	**8 :46**	10 :20	**11 :15**	-	-	-	-
M. a.-midi	Planifié	12 :00	13 :20	-	-	-	-	-	-
	Simulé	12 :00	**13 :08**	-	-	-	-	-	-
Me. matin	Planifié	7 :50	9 :00	10 :20	11 :30	-	-	-	-
	Simulé	7 :50	**8 :56**	10 :20	**11 :27**	-	-	-	-
J. matin	Planifié	7 :50	9 :00	10 :50	11 :30	-	-	10 :50	11 :30
	Simulé	7 :50	**8 :36**	10 :50	**11 :06**	-	-	10 :50	**11 :05**
J. a.-midi	Planifié	12 :00	13 :20	-	-	12 :00	13 :30	15 :30	16 :40
	Simulé	12 :00	**12 :13**	-	-	12 :00	**12 :21**	15 :30	**16 :02**
V. matin	Planifié	7 :50	9 :00	10 :40	11 :30	-	-	10 :40	11 :30
	Simulé	7 :50	**8 :42**	10 :40	**11 :06**	-	-	10 :40	**11 :13**

Tab. 7.6 – Résultats de la simulation : transporteurs à pied et en tracteur

		Tournée 1		Tournée 2		Tournée 3		Tournée 4	
Lundi	Planifié	6 :30	7 :20	7 :30	9 :10	9 :30	11 :20	11 :30	14 :10
	Simulé	6 :30	**6 :43**	7 :30	**8 :52**	9 :30	**11 :09**	11 :30	**13 :56**
Mardi	Planifié	6 :30	7 :20	7 :30	9 :10	9 :30	11 :20	11 :30	14 :10
	Simulé	6 :30	**6 :45**	7 :30	**8 :53**	9 :30	**11 :21**	11 :30	**13 :59**
Mercredi	Planifié	6 :30	7 :20	7 :30	9 :10	9 :30	11 :20	11 :30	14 :10
	Simulé	6 :30	**6 :46**	7 :30	**8 :51**	9 :30	**11 :13**	11 :30	**13 :59**
Jeudi	Planifié	8 :00	11 :00	11 :30	15 :40	-	-	-	-
	Simulé	8 :00	**10 :44**	11 :30	**14 :37**	-	-	-	-
Vendredi	Planifié	8 :00	11 :30	12 :00	15 :40	-	-	-	-
	Simulé	8 :00	**11 :05**	12 :00	**15 :10**	-	-	-	-

Tab. 7.7 – Résultats de la simulation : transporteur en camion

traintes sur les dates de livraison ne sont pas violées durant la simulation. Les charges de travail des préparateurs sont en général inférieures à la durée maximale fixée à 3 heures, et la durée du temps de travail en heures supplémentaires est acceptable. Le transport des chariots se termine toujours à temps dans chaque période de l'horizon de planification. Le deuxième transporteur est principalement affecté aux transports de fin de journée ; cette ressource est également utilisée pour le transport de médicaments urgents pendant la journée, activité qui n'apparaît pas ici. Enfin, l'effectif de préparateurs convenu convient pour garantir l'approvisionnement des chariots lorsque le processus de déménagement sera terminé.

7.5 Synthèse

Nous avons proposé dans ce chapitre une méthode générique originale pour la résolution du problème de livraison de médicament au sein d'un centre hospitalier. Cette méthode a été appliquée au CHU de Saint-Étienne avec succès et la réorganisation des flux de médicaments liés à la pharmacie est terminée. Plusieurs réunions avec l'équipe de la pharmacie ont permis la sélection d'une meilleure organisation pour la nouvelle pharmacie centrale :

- un modèle d'optimisation nous a permis de construire un planning de réapprovisionnement pour les préparateurs et les transporteurs, tout en respectant les contraintes imposées par le CHU ;
- un modèle de simulation a été construit pour tester l'efficacité de la nouvelle organisation et pour valider les plannings de travail de chaque intervenant.

L'intérêt tout particulier de la pharmacie pour l'outil d'optimisation intégré au sein de la plate-forme de simulation medPRO est particulièrement encourageant. Le formalisme de modélisation utilisé s'est révélé particulièrement parlant pour l'ensemble des intervenants, qui ont pu réagir et corriger le modèle de leur propre pharmacie. Chaque travailleur a pu visualiser et commenter son activité grâce aux différentes vues proposées. Les résultats finaux ont été présentés à l'ensemble de l'équipe et une démonstration de la simulation a été réalisée. Les pharmaciens se sont montrés très intéressés par la flexibilité de l'outil d'optimisation qui pourrait être installé dans la future pharmacie.

Chapitre 8

Le bloc opératoire

Ce dernier chapitre est consacré à une étude de cas concernant la planification des interventions chirurgicales dans un bloc opératoire spécialisé du CHU de Saint-Étienne : les flux de ce système ont été modélisés avec medPRO et SADT, permettant de mettre en valeur les avantages de notre approche. Nous proposons également un outil de planification dédié au bloc opératoire : une version simplifiée du problème d'ordonnancement a été résolue grâce à la relaxation Lagrangienne. Cet outil constitue un module implantable dans la plate-forme medPRO.

8.1 Introduction

La planification des interventions chirurgicales et l'affectation de ressources dans le bloc opératoire relèvent de problématiques cruciales à la fois pour la prise en charge du patient et l'optimisation de ressources. Le processus de chirurgie est un processus critique à cause (i) des coûts importants associés aux salles opératoires et aux équipements médicaux, et (ii) de son impact direct sur l'état de santé du patient et par extension sur sa satisfaction. Gordon *et al.* (1988) ont déjà mis en avant dans les années 80 que 9 % des dépenses hospitalières sont dues aux procédures chirurgicales ; Dexter (2002) a prouvé que le coût de l'heure supplémentaire d'utilisation d'une salle opératoire est multiplié par 1,75 par rapport à son coût régulier. Ainsi le problème de la planification chirurgicale doit être étudié afin de garantir pour le patient une prise en charge optimale, tout en utilisant au mieux les ressources à moindre coût.

Nous nous focalisons dans cette étude sur une politique particulière en matière de planification chirurgicale (*open scheduling*) et proposons une méthode pratique pour la construction d'un planning de bloc opératoire. Ce planning doit tenir compte des ressources disponibles mises en jeu durant l'horizon de planification et maximiser le taux d'utilisation du bloc. De plus, nous considérons l'ensemble du processus de chirurgie, du brancardage du patient depuis sa chambre jusqu'au brancardage retour.

Après une brève revue de littérature présentée section 8.2, nous proposons un modèle du processus opératoire section 8.3 réalisé sous medPRO. Nous comparons cette approche avec une analyse SADT dans la section 8.4. Le problème au centre de l'étude est détaillé section 8.5 et la méthode de résolution adoptée section 8.6. Les résultats et le bilan de l'étude sont présentés sections 8.7 et 8.8 respectivement. Cette étude a été présentée dans (Augusto *et al.*, 2008).

8.2 Revue de littérature

Le problème de la planification chirurgicale consiste à planifier les interventions sur une période donnée (généralement une semaine) en tenant compte des ressources disponibles. Les blocs opératoires peuvent être répartis en plusieurs catégories selon les équipements médicaux, les spécificités des salles d'opération,

et leur localisation dans l'hôpital (un bloc centralisé ou un bloc par spécialité médicale). Selon le type de bloc opératoire, une politique de planification peut être adoptée : *open scheduling* ou *block scheduling* (Jebali, 2004). Dans la première politique il n'existe aucune contrainte sur les salles d'opérations : les interventions sont programmées sur l'ensemble du bloc. Dans la deuxième politique, un planning préalable (planning chirurgical maître) est établi : les créneaux sont alloués aux chirurgiens, groupes de chirurgiens ou spécialités médicales. Le problème consiste alors à planifier les interventions au sein de chaque créneau pour chaque spécialité.

Le bloc opératoire étant un environnement stochastique, l'avantage de la politique *open scheduling* repose sur sa flexibité (Jebali, 2004), même si la détermination d'un planning optimal est un problème difficile (Pinedo, 1995). Cette politique est donc utile pour des blocs opératoires de petite taille (moins de dix salles opératoires) dédiés à des spécialités médicales. Le problème de planification d'interventions chirurgicales en tenant compte des contraintes de réveil et de brancardage est peu abordé dans la littérature. Marcon *et al.* (2003) ont développé un modèle de simulation pour trouver une stratégie de planification optimale tout en considérant l'ensemble du processus de chirurgie. Un modèle mathématique est proposé pour minimiser la durée globale d'ouverture des salles opératoires. Un modèle de simulation est utilisé pour modéliser le parcours patient à travers le bloc opératoire. Les résultats indiquent que les brancardiers sont la ressource goulot de ce système. Fei *et al.* (2006a) ont résolu un problème similaire avec la construction de plannings quotidiens pour le bloc opératoire. Les salles opératoires et les lits de réveil sont identiques. Le réveil du patient dans les salles opératoires est autorisé. Les brancardiers ne sont pas considérés. Un algorithme génétique hybride est utilisé pour résoudre le problème. La prise en compte des urgences dans le bloc opératoire a été étudiée par Lamiri *et al.* (2006) : un modèle stochastique de planification a été présenté dans cette optique. Une étude similaire présentée dans (Lamiri *et al.*, 2007) décrit la résolution du même problème en utilisant la technique de génération de colonnes.

La politique de *block scheduling* est privilégiée pour des blocs opératoires de taille plus importante où plusieurs spécialités médicales se côtoient. Les interventions chirurgicales sont centralisées. Guinet et Chaabane (2003) ont proposé une approche en deux étapes pour la planification : les patients sont tout d'abord affectés aux salles opératoires, puis les interventions de chaque salle sont planifiées en utilisant une extension de la méthode hongroise. Jebali *et al.* (2006) ont étudié un problème similaire en proposant une formulation sous forme de modèle mathématique résolu grâce à un solveur commercial. Fei *et al.* (2006b) ont également proposé une méthode en deux étapes : un planning hebdomadaire est tout d'abord généré grâce à la génération de colonnes, puis le problème de planification quotidien est résolu grâce à un algorithme génétique hybride. Enfin, Kharraja *et al.* (2006) ont proposé un outil permettant la construction d'un planning chirurgical maître où les chirurgiens sont considérés individuellement dans un premier temps, puis comme membres de groupes. Cette dernière approche donne de meilleurs résultats car les chirurgiens en groupe ont plus de flexibilité pour assurer leurs interventions.

8.3 Description et modélisation du système

Le bloc opératoire est constitué de plusieurs salles opératoires et d'une salle de réveil (également appelée SSPI, *Salle de Soins Post-Intervention*) qui contient au moins autant de lits de réveil qu'il existe de salles opératoires. Généralement, au moins une salle opératoire est réservée pour les urgences : nous considérons dans cette étude que les interventions d'urgence sont toujours réalisées dans cette salle, sans perturber la programmation des salles régulières. Jebali *et al.* (2006) décrivent deux phases principales dans le processus : (i) la phase pré-intervention, qui a lieu dans la salle opératoire, où les activités de préparation, d'intervention et de nettoyage sont entreprises, et (ii) la phase post-anesthésie, où le patient se réveille dans la salle de réveil.

Nous considérons le processus d'intervention chirurgicale suivi par un patient. Le jour de son interven-

tion, le patient est transporté au bloc opératoire par deux brancardiers. L'intervention est réalisée par une équipe chirurgicale. Lorsque l'intervention est terminée, le patient est immédiatement transporté dans un lit de réveil de la SSPI. Au même moment, le nettoyage de la salle commence. Si aucun lit de réveil n'est disponible lorsque l'intervention se termine, une partie du réveil du patient peut éventuellement avoir lieu dans la salle opératoire. Lorsque le patient est réveillé, ce dernier est transporté dans sa chambre par deux brancardiers.

La figure 8.3 présente le processus décrit dans la paragraphe précédent. Il s'agit d'une représentation simplifiée du processus d'intervention chirurgicale. Grâce aux macro-états, nous proposons une représentation plus détaillée du processus de chirurgie.

La figure 8.4 décrit les états par lesquels passe le patient dans la phase d'anesthésie (se reporter à l'annexe D pour l'intégralité du modèle). Ces opérations impliquent un certain nombre d'intervenant : chirurgiens (Chirurgien), médecins anesthésistes réanimateurs (MAR), infirmiers diplômés d'état (IDE), infirmiers de bloc opératoire diplômés d'état (IBODE), infirmiers anesthésistes diplômés d'état (IADE), et des aide-soignants (AS). Nous n'entrerons pas dans le détail de ces processus qui décrivent avec précision chaque étape d'une intervention chirurgicale.

La figure 8.5 détaille les activités concernant la salle opératoire (SO) et les brancardiers. Un état grisé désigne un état synchronisé avec la vue processus. Certains états impliquant plusieurs ressources (tel « Préparer le matériel de chirurgie ») apparaissent de manière rigoureusement identique dans les machines d'états de la salle opératoire et du chirurgien. Les activités des ressources IDE, IADE, IBODEc et IBODEi ne sont pas décrites en détail, et seront considérées comme des ressources simples dans cette modélisation.

Les durées des différentes activités manquent dans ces diagrammes car nous n'avons pu collecter ces données. Cependant il sera admis que les activités concernant l'intervention et le réveil du patient sont les tâches critiques car bien plus longues que le brancardage ou le nettoyage de la salle. L'estimation de la durée opératoire est difficile à estimer et affecte directement la charge des ressources les plus critiques : les salles opératoires. La durée de l'intervention dépend essentiellement du chirurgien et de la pathologie du patient (Wright *et al.*, 1996).

8.4 Comparaison SADT/medPRO

Nous nous proposons de comparer la modélisation complète du processus opératoire décrite dans la section précédente avec la modélisation réalisé grâce à SADT. Le modèle correspondant présenté dans (Chaabane, 2004) est rappelé dans l'annexe C. Le processus général et le processus d'anesthésie sont cependant rappelés figures 8.1 et 8.2. Une approche hiérarchique descendante est proposée, permettant une appréhension modulaire et progressive de la complexité du système. Le symbolisme graphique de SADT offre un outil de communication universel, clair et synthétique. Une analyse du système par les activités et les données est fournie, définissant un moyen d'expression qui favorise un dialogue de qualité tout en minimisant les problèmes d'interprétation. Onze diagrammes sont proposés dans (Chaabane, 2004) sous la forme d'une arborescence, détaillant les processus pré-opératoire (des consultations à l'hospitalisation), per-opératoire (de l'admission à la sortie du bloc) et post-opératoire (du retour de bloc à la sortie de l'hôpital). Pour la comparaison avec medPRO, nous ne nous intéresserons qu'à la phase per-opératoire, constituée de huit diagrammes, regroupés dans l'annexe C.

Une première analyse comparative entre une modélisation sous SADT et medPRO permet de mettre en avant les différences suivantes :

1. Les activités concernant le patient sont mêlées aux activités concernant les intervenants hospitaliers et/ou les ressources matérielles dans SADT : par exemple le diagramme présenté figure 8.1 présente les activités de préparation de la salle opératoire qui se trouvent pourtant sur un plan différent de

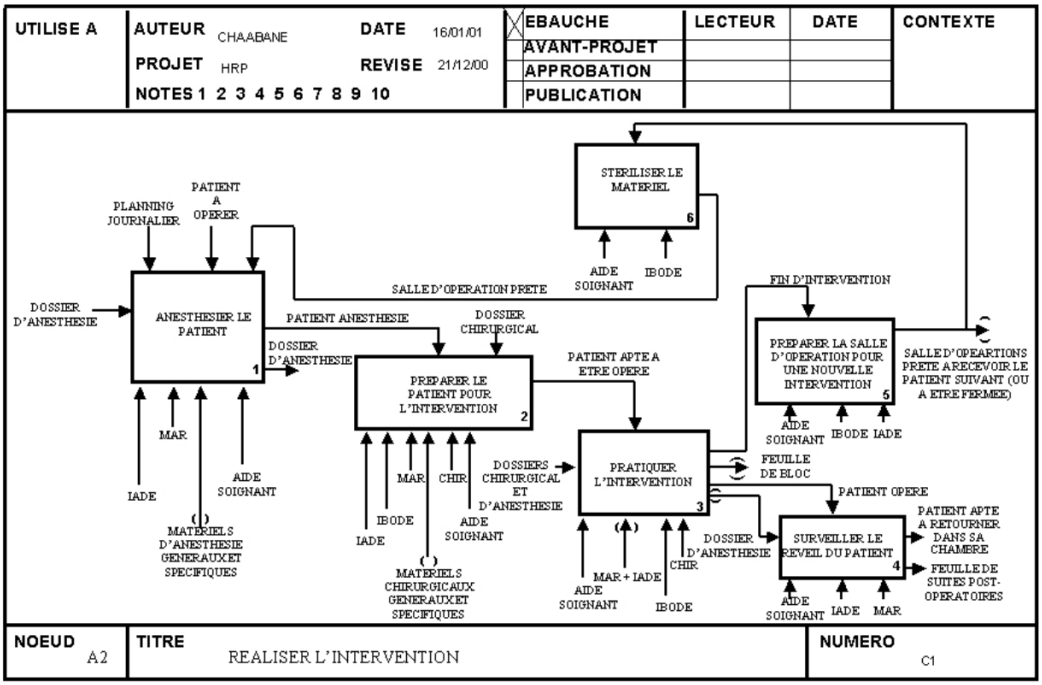


FIG. 8.1 – A2 – Réaliser l'intervention

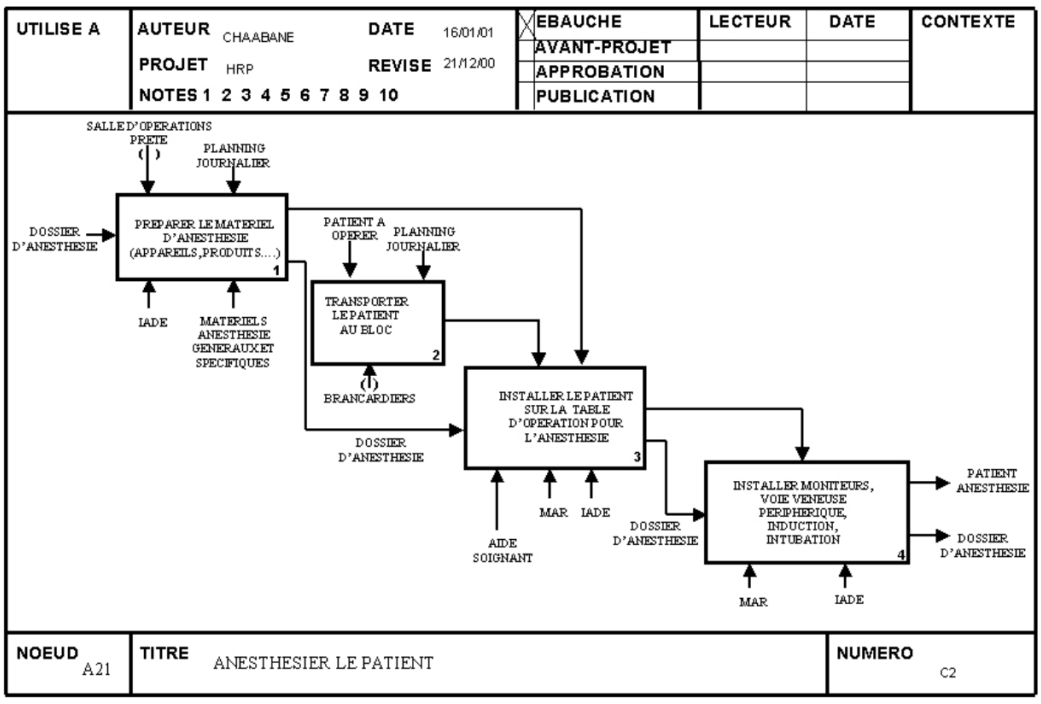


FIG. 8.2 – A21 – Anesthésier le patient

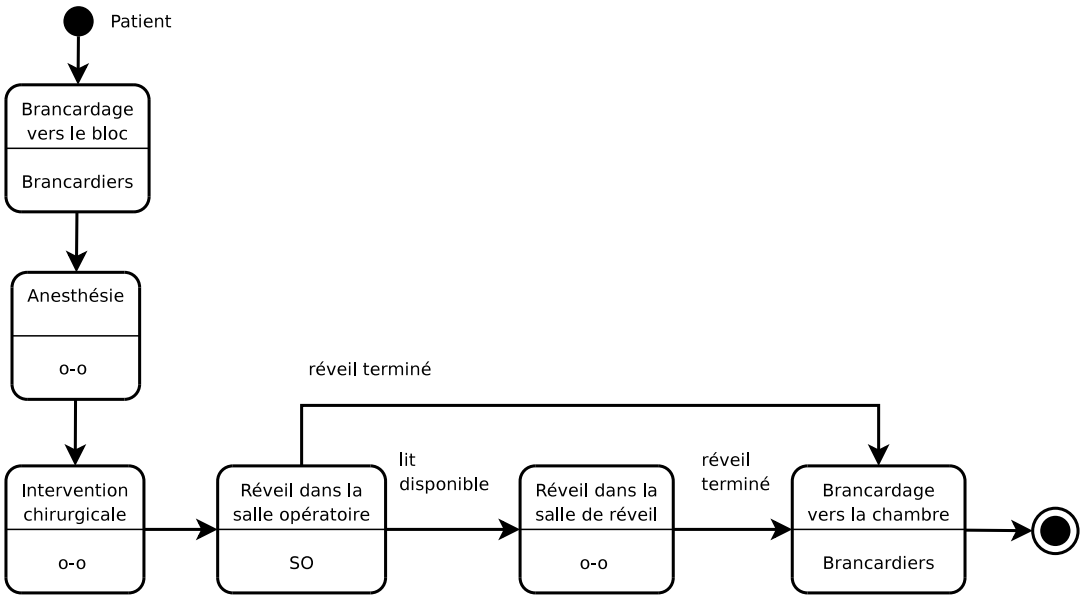


FIG. 8.3 – Vue processus macroscopique de l'intervention chirurgicale

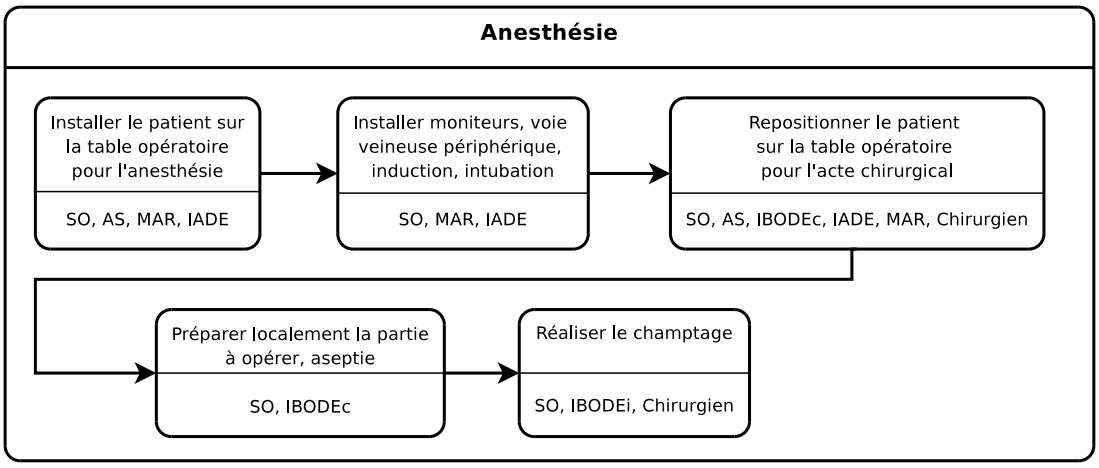


FIG. 8.4 – État composite pour l'Anesthésie

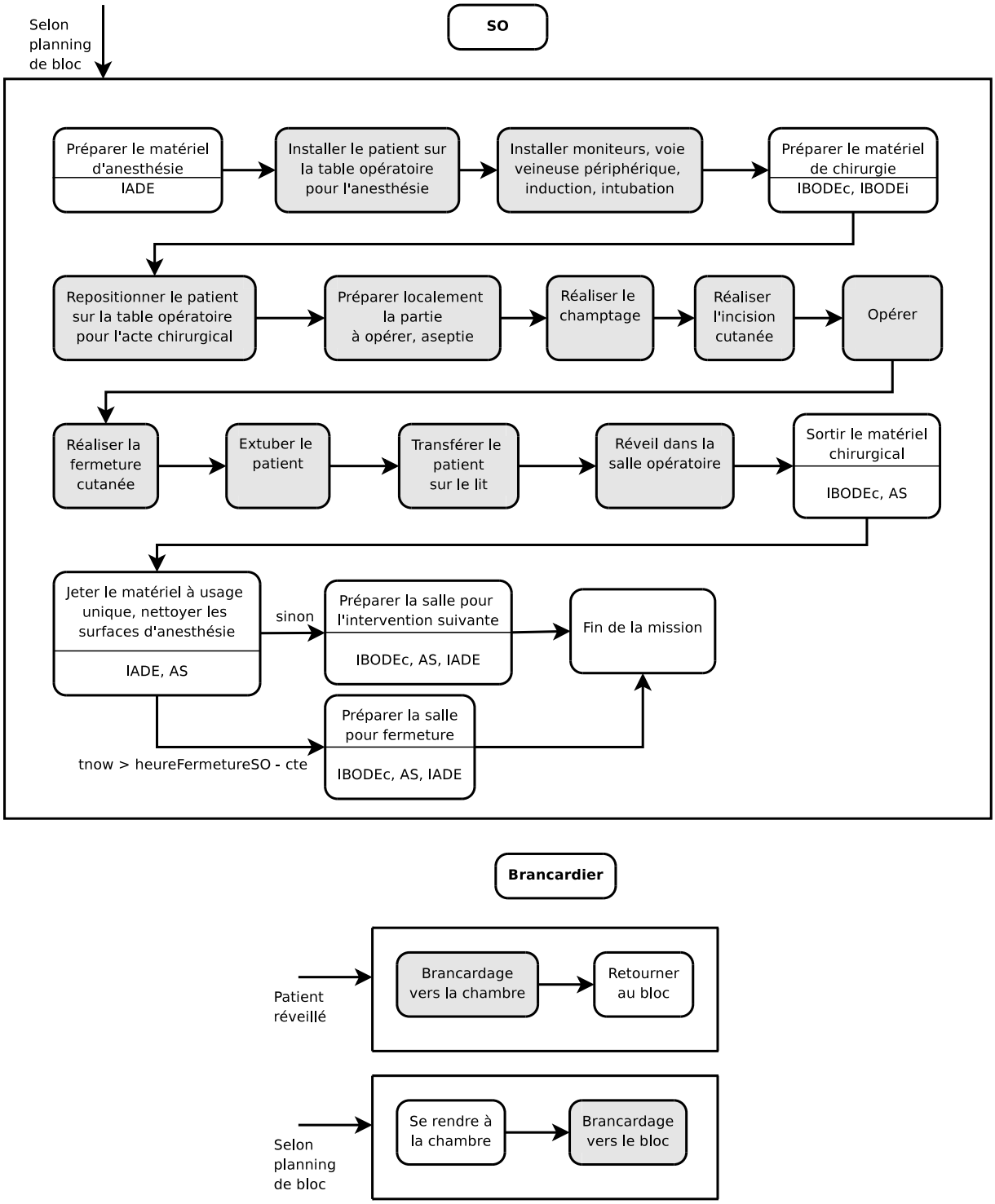


FIG. 8.5 – Vue ressource : Salle opératoire, Brancardier

la prise en charge du patient ; de même pour le diagramme présenté figure 8.2 où la préparation du matériel d'anesthésie est incluse dans le processus patient, qui n'est pourtant pas concerné par ces activités. La modélisation medPRO permet de séparer sans équivoque ces activités grâce à la vue ressource : la préparation de la salle n'apparaît que pour la salle (c.f. figure 8.5).

2. Dans la continuité de la remarque précédente, il est difficile de comprendre l'enchaînement des activités assurées par les différents intervenants dans SADT. La vue ressource permet dans medPRO de modéliser précisément le processus de prise en charge assuré par le chirurgien ou par l'anesthésiste. De plus, la modélisation du processus opératoire du point de vue de la salle opératoire (c.f. figure 8.5) permet d'évaluer globalement les coûts d'utilisation associés à cette ressource.
3. L'approche hiérarchique descendante constitue une approche intéressante pour l'organisation des diagrammes selon les principales phases du processus opératoire. Une approche similaire est présentée dans la modélisation UML grâce aux états composites.
4. SADT permet de faire apparaître dans le modèle les flux informationnels (dossier chirurgical, feuille de bloc, etc.) associés au parcours patient. Ces flux n'apparaissent pas dans la modélisation medPRO afin de ne pas surcharger le modèle, étant donné que nous nous intéressons exclusivement au parcours patient. Le traitement de données de ce type est réalisé par l'intermédiaire du système d'information qui pourra être mis à jour au cours du processus modélisé. L'ajout de commentaires permet de préciser à quels moments un dossier ou une information est requise.
5. SADT ne comporte aucune information dynamique (durée opératoire, expressions booléennes, etc.) et le modèle proposé ne peut être directement simulé. Nous avons présenté dans ce mémoire les processus mis en place pour pallier à ce problème grâce à la plate-forme medPRO.

En plus de pallier aux problèmes récurrents de SADT, la plate-forme medPRO permet d'apporter une précision de modélisation supplémentaire tout en concervant les avantages de SADT, à savoir sa modularité et sa clarté. Au final le modèle UML proposé comporte un nombre d'états plus important du fait de la mise en place de différentes vues, mais l'organisation proposée permet une meilleure visualisation des processus opératoires. Enfin, la possibilité de convertir directement le modèle UML en modèle de simulation offre un gain de temps indéniable par rapport à des outils tels que SADT.

8.5 Problème d'ordonnancement

Nous nous intéressons dans cette étude à l'ordonnancement des patients dont l'intervention est programmée sur une certaine durée afin de proposer un outil de planification efficace pour le bloc opératoire observé. Cet outil pourra s'intégrer dans l'interface utilisateur pour l'injection de planning dans le modèle de simulation décrit plus haut. Pour résoudre ce problème, nous considérons le problème simplifié décrit figure 8.3.

De manière formelle, N patients doivent être opérés sur un certain horizon H, un jour ou une semaine par exemple. Pour chaque patient i, nous introduisons les cinq tâches suivantes de durées p_{ij}, $i \in \{1, \ldots, N\}$ et $j \in \{1, \ldots, 5\}$:

1. $TR1_i$ (p_{i1}) : brancardage du patient de sa chambre vers la salle opératoire.
2. SUR_i (p_{i2}) : intervention dans une salle opératoire quelconque (anesthésie comprise).
3. REC_i (p_{i3}) : réveil dans un lit de la salle de réveil.
4. $TR2_i$ (p_{i4}) : brancardage du patient de la salle opératoire vers sa chambre.
5. CL_i (p_{i5}) : nettoyage de la salle opératoire.

Nous considérons les hypothèses suivantes :

- H_1 : les interventions urgentes ne sont pas prises en compte.
- H_2 : la salle opératoire, les intervenants et le matériel nécessaire à l'intervention sont considérés comme une unique ressource appelée « salle opératoire ».
- H_3 : le réveil REC_i commence dans un lit de réveil immédiatement après la fin de l'intervention SUR_i.
- H_4 : $p_{i5} \leq p_{i3} + p_{i4}$.

Les ressources suivantes sont considérées :

- N_t : nombre de salles opératoires à l'instant t.
- L_t : nombre de brancardiers à l'instant t.
- M_t : nombre de lits de réveil à l'instant t.

Les variables de décision sont s_{ij}, la date de début de la tâche j pour le patient i. Nous pouvons maintenant poser les contraintes du problème :

$$s_{i1} + p_{i1} \leq s_{i2} \tag{8.1a}$$
$$s_{i2} + p_{i2} = s_{i3} \tag{8.1b}$$
$$s_{i2} + p_{i2} = s_{i5} \tag{8.1c}$$
$$s_{i3} + p_{i3} \leq s_{i4} \tag{8.1d}$$
$$c_{ij} = s_{ij} + p_{ij} \tag{8.1e}$$
$$\delta_{ijt} = \mathbf{1}\{s_{ij} \leq t \leq s_{ij} + p_{ij} - 1\} \tag{8.1f}$$
$$\sum_i (\delta_{i1t} + \delta_{i4t}) \leq L_t \tag{8.1g}$$
$$\sum_i (\delta_{i2t} + \delta_{i5t}) \leq N_t \tag{8.1h}$$
$$\sum_i \delta_{i3t} \leq M_t \tag{8.1i}$$

Les contraintes (8.1a–8.1d) sont les contraintes de précédence. La contrainte (8.1e) définit la date de fin c_{ij} de la tâche j pour le patient i. La variable δ_{ijt} (contrainte (8.1f)) précise si la tâche j pour le patient i est active durant la période t. Les contraintes (8.1g–8.1i) sont les contraintes de capacité pour les brancardiers, les salles opératoires et les lits de réveil respectivement.

Le critère à optimiser J est la somme d'une fonction de c_i, où c_i est la date de fin de la dernière tâche pour le patient i ($c_i = c_{i4}$ selon H_4).

$$J = \min \sum_i f(c_i) \tag{8.2}$$

Le makespan ($\max(c_i)$) est le critère le plus classique si l'on considère la littérature de la recherche opérationnelle. Cependant nous ne pouvons utiliser la relaxation Lagrangienne pour résoudre ce problème avec un tel critère ; nous nous tournons ainsi vers des critères de type additif, par exemple $f(c_i) = c_i^2$ ou $f(c_i) = c_i^3$. Avec un critère tel que $f(c_i) = c_i^n$ avec n grand, la minimisation du critère revient à minimiser le c_i maximum, i.e. à minimiser le makespan. Notre approche peut donc être appliquée pour la minimisation du makespan.

8.6 Relaxation Lagrangienne

Cette section décrit l'application de la relaxation Lagrangienne pour la résolution du problème décrit dans la section précédente. La relaxation Lagrangienne est une technique mathématique utilisée pour

la résolution de problèmes d'optimisation fortement contraints (Luenberger, 1984). Cette méthode s'est démarquée comme une technique fiable pour la résolution de problèmes d'ordonnancement complexes (Chen *et al.*, 1998; Hoitomt *et al.*, 1993; Wang *et al.*, 1997). Nous l'utilisons ici pour décomposer le problème d'ordonnancement en sous-problèmes au niveau du patient.

8.6.1 Relaxation Lagrangienne du problème

Afin d'appliquer cette méthode, nous choisissons de relaxer les contraintes de capacité (8.1g–8.1i) en utilisant les multiplicateurs de Lagrange, qui modélisent dans ce cas le coût d'utilisation des ressources : les multiplicateurs λ_t, β_t et γ_t représentent les coûts pour utiliser brancardiers, salles opératoires et lits de réveil, respectivement. Nous en déduisons le problème relaxé suivant :

$$RP = L(\lambda_t, \beta_t, \gamma_t) \tag{8.3}$$

où :

$$\begin{aligned} L(\lambda_t, \beta_t, \gamma_t) = \min \Big\{ & \sum_i f(c_i) + \sum_t \lambda_t \Big(\sum_i (\delta_{i1t} + \delta_{i4t}) - L_t \Big) \\ & + \sum_t \beta_t \Big(\sum_i (\delta_{i2t} + \delta_{i5t}) - N_t \Big) + \sum_t \gamma_t \Big(\sum_i \delta_{i3t} - M_t \Big) \Big\} \end{aligned} \tag{8.4}$$

sous les contraintes :

$$(8.1\text{a} - 8.1\text{f})$$

Le problème (8.3) peut être reformulé de la manière suivante :

$$L(\lambda_t, \beta_t, \gamma_t) = -\sum_t \lambda_t L_t - \sum_t \beta_t N_t - \sum_t \gamma_t M_t + \sum_i SP_i(\lambda_t, \beta_t, \gamma_t) \tag{8.5}$$

où :

$$SP_i(\lambda_t, \beta_t, \gamma_t) = \min \Big\{ f(c_i) + \sum_t \lambda_t \delta_{i1t} + \sum_t \lambda_t \delta_{i4t} + \sum_t \beta_t (\delta_{i2t} + \delta_{i5t}) + \sum_t \gamma_t \delta_{i3t} \Big\} \tag{8.6}$$

sous les contraintes :

$$(8.1\text{a} - 8.1\text{f})$$

Nous pouvons simplifier l'expression 8.6 ainsi :

$$SP_i(\lambda_t, \beta_t, \gamma_t) = \min \Big\{ f(c_i) + \sum_{t=s_{i1}}^{s_{i1}+p_{i1}-1} \lambda_t + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t + \sum_{t=s_{i2}}^{s_{i2}+p_{i2}+p_{i5}-1} \beta_t + \sum_{t=s_{i2}+p_{i2}}^{s_{i2}+p_{i2}+p_{i3}-1} \gamma_t \Big\} \tag{8.7}$$

sous les contraintes :

$$(8.1\text{a} - 8.1\text{f})$$

Afin de résoudre le problème relaxé (8.3), nous devons resoudre les sous-problèmes (8.7) en utilisant la programmation dynamique.

8.6.2 Résolution des sous-problèmes relaxés à l'aide de la programmation dynamique

Les sous-problèmes (8.7) peuvent être résolus en utilisant la programmation dynamique. Soient SPP_{4i}, SPP_{2i} et SPP_{1i} les sous-problèmes suivants :

$$\begin{gathered} SPP_{4i}(T) = f(c_i) + \sum_{t=s_{i4}}^{s_{i1}+p_{i4}-1} \lambda_t \\ s.c.\ (1\mathrm{a}-1\mathrm{i}),\ s_{i4}+p_{i4}-1 \leq H,\ s_{i4}=T \end{gathered} \tag{8.8}$$

$$\begin{gathered} SPP_{2i}(T) = \min\Big\{ f(c_i) + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t + \sum_{t=s_{i2}}^{s_{i2}+p_{i2}+p_{i5}-1} \beta_t + \sum_{t=s_{i2}+p_{i2}}^{s_{i2}+p_{i2}+p_{i3}-1} \gamma_t \Big\} \\ s.c.\ (1\mathrm{a}-1\mathrm{i}),\ s_{i2}+p_{i2}+p_{i3}+p_{i4}-1 \leq H,\ s_{i2}=T \end{gathered} \tag{8.9}$$

$$\begin{gathered} SPP_{1i}(T) = \min\Big\{ f(c_i) + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t + \sum_{t=s_{i2}}^{s_{i2}+p_{i2}+p_{i5}-1} \beta_t + \sum_{t=s_{i2}+p_{i2}}^{s_{i2}+p_{i2}+p_{i3}-1} \gamma_t + \sum_{t=s_{i1}}^{s_{i1}+p_{i1}-1} \lambda_t \Big\} \\ s.c.\ (1\mathrm{a}-1\mathrm{i}),\ s_{i1}+\sum_{j=1}^{4} p_{ij} - 1 \leq H,\ s_{i1}=T \end{gathered} \tag{8.10}$$

qui peut être réécrit de la manière suivante :

$$SPP_{2i}(T) = \sum_{t=T}^{T+p_{i2}+p_{i5}-1} \beta_t + \sum_{t=T+p_{i2}}^{T+p_{i2}+p_{i3}-1} \gamma_t + \min_{s_{i2}+p_{i2}+p_{i3} \leq t \leq H} SPP_{4i}(t) \tag{8.11}$$

$$SPP_{1i}(T) = \min \sum_{t=T}^{T+p_{i1}-1} \lambda_t + \min_{s_{i1}+p_{i1} \leq t \leq H} SPP_{2i}(t) \tag{8.12}$$

où la solution de $SPP_{4i}(T)$ est évidente. Le sous-problème peut ainsi être résolu en utilisant la programmation dynamique : nous déterminons tout d'abord SPP_{4i} pour toutes les dates de début admissibles $s_{i4} = T$. Puis, nous injectons les valeurs de SPP_{4i} dans (8.11) et calculons SPP_{2i} pour toutes les dates de début admissibles $s_{i2} = T$. Cette procédure est réitérée pour SPP_{1i}. Nous déterminons les dates de début s_{i1} qui minimisent SPP_{1i}, et nous en déduisons les dates de début s_{i2} et s_{i4} qui nous ont permis de trouver cette solution par backtracking.

Une fois tous les sous-problèmes résolus, la solution du problème relaxé est immédiate ; nous obtenons ainsi une borne inférieure du problème initial J (Luh et Hoitmt, 1993).

8.6.3 Construction d'un planning de chirurgie faisable

La solution du problème relaxé n'est pas faisable en général. Le planning construit avec les dates de début obtenues en résolvant le problème relaxé pourrait violer les contraintes de capacité pour certaines périodes élémentaires. Ainsi il est nécessaire de développer des algorithmes permettant de transformer une solution non-faisable en une solution faisable. Deux algorithmes ont été développés pour ce problème, prenant en argument les dates de début s_{ij} de la solution obtenue en résolvant le problème relaxé et renvoyant une solution faisable s_{ij}^F. La solution faisable nous permet de calculer le gap de dualité, mesure de l'optimalité de la solution faisable.

Nous conservons l'ordre des interventions chirurgicales obtenu en résolvant le problème relaxé (ligne 1), car les salles opératoires représentent une ressource critique. Puis chaque patient est placé le plus tôt possible en respectant les contraintes de capacité (lignes 2–4). Le brancardage depuis la chambre est programmé en premier, puis le bloc intervention-réveil-nettoyage, et enfin le brancardage retour. Les

Algorithme 8.1 Génération d'une solution faisable (1)

1 Les patients sont triés selon l'ordre des interventions donné par la solution non-faisable, i.e. $s_{[1]2} \leq s_{[2]2} \leq \cdots \leq s_{[n]2}$.
2 **pour** chaque patient i **faire**
3 Placer le patient i le plus tôt possible, en respectant les contraintes de capacité.
4 **fin pour**

Algorithme 8.2 Génération d'une solution faisable (2)

1 Conserver le planning non-faisable original S.
2 **faire**
3 Pour chaque patient i impliqué dans une violation de capacité, tenter de reprogrammer le patient le plus tôt possible en respectant les contraintes de capacité. Soit C_i la date de sortie associée à ce patient.
4 Modifier définitivement le planning pour le patient dont le C_i est minimal.
5 Recommencer jusqu'à ce que tous les patients en conflit soient insérés.
6 Essayer de retirer et d'insérer tous les patients jusqu'à ce que la solution ne puisse plus être améliorée.

patients sont affectés aux ressources le plus tôt possible dès qu'elles deviennent disponibles ; de cette manière, cet algorithme n'autorise pas l'apparition de conflits de ressources.

L'algorithme 8.2 prend en argument le planning complet des patients obtenu en résolvant le problème relaxé (ligne 1). Chaque patient en conflit est retiré du planning et inséré le plus tôt possible en respectant les contraintes de capacité. La date de sortie de ce patient C_i est évaluée, et le patient est réinséré à sa place. Nous répétons la même opération pour chaque patient en conflit (ligne 3). Le patient dont le critère C_i est minimal est définitivement inséré dans le planning (ligne 4). Nous réitérons la même procédure jusqu'à ce que la solution devienne faisable. Finalement, lorsque la solution est faisable, nous essayons de retirer et d'insérer chaque patient le plus tôt possible jusqu'à ce que la solution ne puisse plus être améliorée (ligne 6). Le planning de chirurgie obtenu inclut tous les patients planifiés et respecte les contraintes de capacité.

Ce dernier algorithme vise à conserver la solution non-faisable originale plutôt que de ne réuiliser que l'ordre donné par cette même solution. De cette manière, l'algorithme essaie de minimiser la déviation par rapport à la solution initiale.

8.6.4 Résolution du problème relaxé

L'objectif de la relaxation Lagrangienne est d'obtenir la meilleure borne inférieure possible de la solution optimale. La méthode du sous-gradient est appliquée itérativement pour résoudre le problème relaxé, et peut être résumée de la manière suivante :

1. Initialiser les multiplicateurs de Lagrange λ_t, β_t et γ_t avec $\lambda_t^{(0)}$, $\beta_t^{(0)}$ et $\gamma_t^{(0)}$. $n \leftarrow 0$.
2. Résoudre $RP = L(\lambda_t, \beta_t, \gamma_t)$. Nous obtenons la solution non-faisable s_{ij} et L.
3. En déduire une solution faisable. Nous obtenons les s_{ij}^F et J.
4. Évaluer le gap : $gap = \dfrac{J - L^*}{L^*}$.
5. Mettre à jour les multiplicateurs : $\alpha^{(n+1)} \leftarrow \alpha^{(n)} + S^{(n)} \nabla L(\alpha^{(n)})$

avec $S^{(n)} = \epsilon \dfrac{L^* - L(\alpha^{(n)})}{||\nabla L(\alpha^{(n)})||}$ et $\alpha = (\lambda; \beta; \gamma)$
$n \leftarrow n + 1$.

6. Aller à l'étape 2 jusqu'à ce que $gap < G$ ou $n > M$.

Ainsi le problème relaxé est résolu itérativement jusqu'à ce que $gap < G$ ou que $n > M$ (G et M sont fixés). Le gap de dualité est une mesure de la qualité de la solution à l'étape n, et est calculé à chaque itération (étape 4). Les multiplicateurs de Lagrange sont mis à jour en utilisant un pas de taille $S^{(n)}$ permettant la convergence (Polyak, 1969) (étape 5) où L^* est le dual optimal, $L(\alpha^{(n)})$ est la valeur du problème relaxé à la n^e itération et $\nabla L(\alpha^{(n)})$ est le sous-gradient du problème relaxé L en respectant les multiplicateurs. ϵ est un facteur variable utilisé pour ajuster la convergence. ϵ est choisi entre 0 et 2 et initialisé à $0,5$.

Remarquons que la valeur de L^* est inconnue durant le processus car il s'agit de la valeur objectif optimale que nous recherchons. Celle-ci est cependant requise pour calculer le gap de dualité et la taille du pas $S^{(n)}$. L^* est donc remplacé par la meilleure valeur de L obtenue jusqu'à maintenant.

8.7 Résultats

Afin de tester l'efficacité de la méthode proposée, plusieurs instances de problèmes ont été générées et testées. Nous n'avons pas pu obtenir des jeux de données réels car les durées des interventions chirurgicales et des réveils des patients sont rarement enregistrées. Cependant, nous avons généré des données aussi proche de la réalité que possible selon nos observations.

L'heuristique et les différents algorithmes décrits dans ce chapitre ont été codés en C++. Les tests ont été réalisés sur une machine de type Pentium 4 cadencé à 3,2 GHz.

8.7.1 Jeux de données

Les données de neuf instances de problèmes ont été générées. Ces instances consignées dans la table 8.1 possèdent des caractéristiques différentes. La première colonne indique l'indice de l'instance ; la deuxième colonne indique le nombre de patients opérés sur l'horizon de planification. Les trois colonnes suivantes indiquent respectivement le nombre de brancardiers, de salles opératoires et de lits de réveil disponibles. Les quatre dernières colonnes présentent les intervalles à partir desquels sont tirées les durées des activités du processus opératoire.

Classe	n	L	N	M	$p_{i,1}, p_{i,4}$	$p_{i,2}$	$p_{i,3}$	$p_{i,5}$
1	10	2	4	6	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
2	15	2	4	6	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
3	20	2	4	6	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
4	30	2	6	10	$\{2,6\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
5	10	1	2	4	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
6	10	2	4	6	$\{1,3\}$	$\{18,24\}$	$\{18,24\}$	$\{2,4\}$
7	10	2	4	6	2	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
8	30	2	6	10	2	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
9	30	3	6	10	$\{2,6\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$

TAB. 8.1 – Description des instances de test

L'horizon de planification est discrétisé en périodes élémentaires de dix minutes chacune. L'horizon H considéré est fixé à 100 ou 200 unités pour les problèmes de petite et de grande taille respectivement.

Nous avons considéré un horizon commun afin de comparer nos résultats avec des heuristiques classiques ; il est cependant possible d'utiliser des horizons de planification plus réalistes comme une semaine ou une journée. Le programme est stoppé après 3.000 itérations et le pas de convergence ϵ est fixé à $1,9$. Quatre critères sont testés et comparés : (i) $f_1(c_i) = \sum c_i$; (ii) $f_2(c_i) = \sum c_i^2$; (iii) $f_3(c_i) = \sum c_i^3$; (iv) $f_4(c_i) = \sum (c_i - LB)^+$. LB est une borne inférieure pour le makespan, calculée en sommant les durées des interventions et du nettoyage pour tous les patients, et le résultat est divisé par le nombre total de salles opératoires disponibles.

$$LB = \frac{\sum_i (p_{i2} + p_{i5})}{\max_t (N_t)} \tag{8.13}$$

Utilisé avec la relaxation Lagrangienne, f_4 est le critère qui donne les meilleurs résultats si l'on considère le C_{max} : LB est une borne inférieure grossière qui ne tient compte que de la charge des salles opératoires. Comme nous allons le voir par la suite, les solutions faisables sont plus éloignées de la solution optimale avec les critères f_3 ou f_4, mais les makespans obtenus sont meilleurs comparés aux solutions obtenues avec les critères f_1 et f_2.

Quinze jeux de données ont été générés pour chaque classe de problème. Les résultats présentés dans les tables 8.2 et 8.3 correspondent à la moyenne sur ces 15 jeux. Toutes les durées opératoires pour chaque jeu de données sont générés aléatoirement. La table 8.1 montre que la durée du brancardage des patients varie toujours entre 10 et 30 minutes avec quelques exceptions ; la durée des interventions entre 40 et 220 minutes ; la durée du réveil entre 1 et 4 heures et la durée du nettoyage entre 20 et 30 minutes.

8.7.2 Résultats numériques

Les tables 8.2 et 8.3 présentent les résulats numériques. La table 8.2 liste les gaps obtenus exprimés en pourcentage pour les différents critères testés ainsi que les durées de calcul mesurées en secondes.

Classe	**Algorithme 1**				**Algorithme 2**			
	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4
1	0.06%	0.11%	0.35%	0.82%	0.04%	0.07%	0.21%	0.67%
	2.3s	2.5s	2.5s	2.6s	2.9s	3.2s	3.1s	3.2s
2	0.11%	0.18%	0.33%	3.66%	0.09%	0.13%	0.34%	4.20%
	7.2s	7.3s	7.3s	7.7s	9.0s	8.8s	9.1s	9.6s
3	0.14%	0.29%	0.54%	2.23%	0.13%	0.49%	1.54%	21.30%
	15.9s	16.4s	16.3s	16.9s	18.7s	30.9s	19.5s	21.3s
4	3.09%	7.13%	11.19%	14.92%	4.80%	8.97%	13.26%	23.18%
	24.3s	24.4s	24.8s	26.1s	34.2s	33.0s	32.5s	33.5s
5	0.13%	0.26%	0.40%	3.73%	0.11%	0.15%	0.25%	6.64%
	5.0s	5.1s	5.2s	5.5s	5.6s	6.1s	6.9s	6.3s
6	0.06%	0.15%	0.29%	0.48%	0.05%	0.16%	0.29%	0.42%
	4.9s	5.2s	5.3s	5.4s	5.9s	6.6s	6.6s	6.3s
7	0.04%	0.10%	0.25%	1.01%	0.03%	0.09%	0.21%	1.35%
	2.2s	2.5s	2.5s	2.7s	2.5s	2.7s	3.1s	3.3s
8	0.42%	0.59%	0.75%	2.59%	0.56%	0.74%	1.91%	33.65%
	23.8s	24.6s	23.9s	26.2s	34.3s	30.7s	31.7s	36.5s
9	0.80%	1.44%	2.06%	6.76%	2.14%	3.80%	4.44%	24.03%
	24.2s	24.6s	24.7s	25.85s	31.6s	31.8s	32.1s	33.3s

TAB. 8.2 – Performances de la relaxation Lagrangienne (gap et durée)

Classe	Critère	Algorithme 1			Algorithme 2		
		SPT	**LPT**	**NEH**	**SPT**	**LPT**	**NEH**
1	f_3	29.62	5.56	1.81	28.84	4.91	1.23
	f_4	29.94	5.81	2.09	25.90	2.52	-1.08
2	f_3	31.15	6.10	1.65	25.94	1.88	-2.36
	f_4	32.83	7.46	2.98	25.35	1.41	-2.82
3	f_3	23.95	5.59	2.10	23.29	5.02	1.60
	f_4	26.86	8.07	4.54	20.44	2.60	-0.75
4	f_3	23.53	5.76	0.91	29.66	11.02	5.93
	f_4	23.18	5.47	0.64	29.35	10.75	5.68
5	f_3	19.46	3.38	1.07	19.36	3.29	1.00
	f_4	21.21	4.89	2.56	18.99	2.97	0.69
6	f_3	8.19	4.81	1.41	7.62	4.26	0.90
	f_4	7.31	3.95	0.60	7.47	4.11	0.75
7	f_3	30.60	3.39	-0.36	30.27	3.13	-0.49
	f_4	31.24	3.90	0.25	27.82	1.19	-2.36
8	f_3	20.13	0.99	0.99	18.95	0.00	0.00
	f_4	22.92	3.34	3.34	13.37	-4.69	-4.69
9	f_3	24.82	5.94	2.79	24.31	5.51	2.37
	f_4	27.23	7.99	4.78	19.80	1.68	-1.34

TAB. 8.3 – Comparaison avec des heuristiques classiques pour le makespan

Concernant les résultats présentés dans la table 8.2, la relaxation Lagrangienne donne de bons résultats pour la plupart des instances, exceptée l'instance 4. Cette instance est un cas irréaliste et difficile où les transporteurs représentent la ressource la plus critique avec 137 % de la charge des salles opératoires. Dans ce cas le plus petit gap vaut 3 % pour f_1 avec l'algorithme 1 et le plus grand vaut 23 % pour f_4 avec l'algorithme 2. L'instance 9 reprend les même conditions que l'instance 4 avec trois transporteurs : dans ce cas, le gap de dualité est significativement plus petit avec l'algorithme 1, avec un gap variant entre 0,8 % et 6,76 %. Pour les autres cas, le gap pour l'algorithme 1 varie entre 0,04 % et 3,73 %. En général, l'algorithme 2 donne de meilleurs résultats lorsque le nombre de patients est plus petit que 15, bien que les temps de calcul restent supérieurs. Enfin, nous observons que le problème devient plus difficile selon l'ordre des critères f_1, f_2, f_3 et f_4. Le plus grand gap de dualité est toujours atteint avec f_4.

Dans la suite, nous comparons l'approche utilisant relaxation Lagrangienne avec les extensions des règles d'ordonnancement classiques LPT, SPT et NEH pour les problèmes de flow-shop. Le makespan C_{max} est utilisé comme critère pour cette comparaison. Comme cela a été dit auparavant, le makespan est le critère le plus classique pour la mesure de performances, mais la relaxation Lagrangienne ne peut être utilisée avec ce critère. À la place nous utilisons le critère $\sum_i f(c_i)$ afin de trouver des solutions qui minimisent également le makespan. Nous ne comparons pas les heuristiques sur le critère $\sum_i f(c_i)$ car cela favoriserait l'approche basée sur la relaxation Lagrangienne. Les plannings obtenus avec la relaxation Lagrangienne sont générés avec les deux algorithmes décrit précédemment et avec les critères f_3 et f_4 qui se rapprochent le plus du critère C_{max}. Les règles LPT, SPT et NEH sont adaptées pour tenir compte de la contrainte no-wait et de la réentrance des brancardiers.

La table 8.3 presente les résultats de la comparaison et donne le résultat suivant :

$$\frac{C_{max}^{Heur} - C_{max}^{Lag}}{C_{max}^{Lag}} \times 100\% \tag{8.14}$$

où C_{max}^{Heur} est le makespan du planning donné par l'heuristique considérée et C_{max}^{Lag} est le meilleur makespan généré pendant la procédure de relaxation Lagrangienne avec les critères f_3 ou f_4 et avec l'un des algorithmes proposés. Les observations suivantes ont été réalisées :

1. Concernant les heuristiques, SPT donne les plus mauvais résultats et NEH donne les meilleurs résultats ; la règle SPT est toujours dominée par la relaxation Lagrangienne.
2. En général, la relaxation Lagrangienne donne de meilleurs résultats avec l'algorithme 1 qu'avec l'algorithme 2. La relaxation Lagrangienne avec l'algorithme 1 domine toutes les heuristiques pour toutes les instances considérées, excepté pour l'instance 7 pour laquelle NEH donne des résultats légèrement meilleurs lorsque f_3 est utilisé.
3. En général, la relaxation Lagrangienne donne de meilleurs résultats avec le critère f_4. La relaxation Lagrangienne avec l'algorithme 1 et le critère f_4 domine toutes les heuristiques pour toutes les instances. Il est intéressant de noter que le gap de dualité est le plus mauvais pour la relaxation Lagrangienne avec le critère f_4.

Le gap est le plus mauvais lorsque la relaxation Lagrangienne est utilisée avec le critère f_4, mais les makespans sont meilleurs : lorsqu'une solution faisable est construite en utilisant le critère f_4, le C_{max} de la solution est très bon car le critère a été choisi pour l'obtention de résultats similaires au makespan. Cependant, cette solution peut être en même temps éloignée de l'optimum, calculée sur la base de la meilleure solution non-faisable trouvée jusqu'à maintenant : les gaps sont donc plus grands pour la relaxation Lagrangienne avec les critères f_3 et f_4.

8.8 Synthèse

Une approche utilisant la relaxation Lagrangienne a été proposée pour l'ordonnancement du bloc opératoire. La disponibilité des lits de réveil et des transporteurs est prise en compte, et nous avons considéré un critère général séparable qui dépend de la date de sortie de chaque patient. Les résultats numériques montrent que cette méthode permet l'obtention d'une borne inférieure réduite et de bonnes solutions faisables avec un gap de dualité faible. Malgré le fait que la relaxation Lagrangienne n'ait pas été développée pour des critères non séparables comme la minimisation du makespan, les résultats numériques démontrent, avec un critère séparable approprié, que la relaxation Lagrangienne se révèle bien plus efficaces que des règles d'ordonnancement propres au flowshop telles que SPT, LPT ou NEH.

Ces approches ont été intégrées dans la première couche logiciel de la plate-forme medPRO et permettent la constitution d'un planning opératoire. Ce planning peut ensuite être simulé afin de tester sa robustesse dans des conditions stochastiques.

Le problème d'ordonnancement du bloc opératoire où le réveil du patient (en partie ou intégral) est autorisé dans la salle opératoire lorsqu'aucun lit de réveil n'est disponible est considéré dans (Augusto *et al.*, 2007). La relaxation Lagrangienne offre dans ce cas également de bons résultats. En matière de perspectives il serait intéressant d'étudier le même problème en faisant varier les ressources disponibles au cours du temps. Étudier l'importance des brancardiers constitueraient également un travail intéressant dans la mesure où ces derniers constituent une ressource goûlot du bloc opératoire.

Conclusion générale

Synthèse

Les systèmes hospitaliers sont des faux-amis des systèmes de production industriels. Comparables sur de nombreux points, l'application de méthodes analytiques issues du génie industriel se révèle pourtant difficile à mettre en œuvre dans le domaine médical : les disparités organisationnelles entre hôpitaux, la nature particulière et la diversité des flux hospitaliers, le caractère hautement stochastique de l'évolution de l'état de santé des patients sont autant d'obstacles à la mise en place de méthodologies de réingénierie classique pour le prototypage rapide de ces systèmes.

Afin de répondre à cette problématique, nous avons spécifié et développé dans cette thèse une plate-forme de modélisation et d'analyse de flux spécifiquement dédiée aux systèmes hospitaliers. Nous nous sommes appuyés sur un large état de l'art portant sur l'application de méthodes de modélisation et de simulation en milieu hospitalier et sur nos propres observations pour spécifier cet outil. La structure logicielle adoptée permet de proposer à chaque type d'utilisateur (hospitalier, ingénieur, développeur) une interface intuitive. L'architecture systémique adoptée offre une décomposition rigoureuse du système en trois sous-systèmes : physique, informationnel et décisionnel.

Le système physique offre une représentation des flux opérationnels du système considéré. UML a été choisi pour la modélisation. Trois vues sont proposées : la vue processus offre une représentation du système centrée sur le patient (ou le produit) ; la vue ressource détaille le comportement de chaque intervenant sous forme de missions ; la vue organisation permet la déclaration de l'ensemble des acteurs en interaction et leurs relations (spécialisation, compétences et équipes). L'organisation du système physique alliée au formalisme graphique intuitif d'UML offre une représentation limpide des flux d'un système complexe sans ambiguïté. La construction et la modification du modèle medPRO peuvent être réalisées en collaboration avec le personnel soignant car cette phase n'exige aucun pré-requis scientifique.

Le comportement dynamique du système est décrit au moyen des réseaux de Petri. La conversion est automatique, permettant la simulation du modèle medPRO/UML. Les réseaux de Petri permettent en outre de spécifier l'algorithme de simulation à événements discrets de manière formelle et sans équivoque.

Le système informationnel regroupe les données de l'unité médicale représentée dans le système physique. Ces données sont organisées grâce à un modèle conceptuel et stockées dans une base de données. Cette organisation très simple d'un point de vue informatique est souvent inexistante dans bon nombre d'outils de modélisation et/ou de simulation. L'extraction d'informations dans un but statistique est immédiate grâce à des langages d'interrogation de bases de données simples comme SQL. La flexibilité offerte permet de supprimer les intermédiaires et de minimiser le temps perdu en conversion de données. Enfin, la sécurisation de données confidentielle est également possible.

Le système décisionnel implémente un certain nombre de méthodes pour la planification et l'ordonnancement d'une part, et le pilotage en temps réel d'autre part. Nous avons adapté deux méthodes issues du milieu industriel pour la planification à court terme et l'ordonnancement de systèmes hospitaliers. Ces approches permettent, uniquement à partir du modèle medPRO/UML, de planifier des entités et d'ajuster

les ressources nécessaires aux traitements modélisés. Le pilotage en temps réel est utilisé pour contrôler la simulation et gérer l'ensemble des aspects décisionnels : sélection de ressources, branchements conditionnels, prise de décision médicale, etc. Basé sur une approche modulaire hiérarchique/hétérarchique, ce système de contrôle est destiné à être étendu pour couvrir l'ensemble des processus décisionnels propres au domaine médical.

La méthodologie proposée dans cette thèse a été appliquée à trois études de cas réels liées à trois services très différents du CHU de Saint-Étienne :

L'unité neuro-vasculaire : nous avons étudié les facteurs qui influent sur la durée moyenne de séjour de patients atteints d'accidents vasculaires cérébraux.

La pharmacie : le problème de transport et de livraison de médicament dans le cadre d'une réorganisation générale de l'hôpital a été résolu.

Le bloc opératoire : le processus de chirurgie a été étudié et modélisé ; nous avons également extrait et résolu un problème d'ordonnancement.

Nous avons pu constater que la modélisation medPRO permet la mise en valeur des aspects organisationnels critiques de ces systèmes, offrant une plus-value intéressante comparée à des outils de modélisation/simulation classiques. Nous avons démontré que l'utilisation d'un formalisme de modélisation simple permet de minimiser les erreurs d'incompréhension et d'accélerer la phase de collecte et de validation des données. L'intégration du système d'information sous forme d'une base de donnée au sein de la plate-forme représente une structure parfaitement adaptée à l'organisation, au traitement, et à la sécurisation de données. La séparation des systèmes physique et décisionnel se révèle essentielle pour une modélisation fidèle de processus décisionnels particuliers liés au milieu médical, comme nous avons pu le constater dans l'étude concernant l'unité neuro-vasculaire. Enfin, l'organisation de la plate-forme medPRO en couches logicielles dédiées permet l'intégration d'outils adaptés (non génériques) aux besoins du personnel hospitalier : chacune de ces études a d'ailleurs permis la création d'un outil d'optimisation spécifique et son intégration à la plate-forme medPRO.

Les pratiques observées ont été recueillies et consignées dans un recueil, constituant le guide méthodologique associé à l'unité médicale étudiée. Par exemple, l'utilisateur désireux de modéliser le bloc opératoire d'un hôpital particulier pourra s'y référer pour comprendre rapidement les spécificités d'un tel système et éventuellement réutiliser les modèles présentés dans ce chapitre. Nous avons pu constater que les spécificités médicales que nous nous sommes efforcés de mettre en avant tout au long de ce mémoire constituent les principaux obstacles à une étude de prototypage rapide, d'où la nécessité de mettre en place un guide méthodologique adapté. L'architecture, la décomposition en vues et la modularité de la plate-forme medPRO permettent ainsi une modélisation intuitive et compréhensible. Enfin, l'intégration de modules spécifiques aux systèmes étudiés est possible au travers de l'interface utilisateur (couche logicielle la plus élevée), comme nous avons pu le démontrer dans les études de cas présentées dans ce mémoire.

Perspectives

Nous avons spécifié et développé une plate-forme de modélisation et de simulation pour les systèmes hospitaliers : nous nous sommes restreints dans cette optique à l'étude de systèmes à l'échelle de services de soins afin de proposer à l'utilisateur un panel d'outils et un guide méthodologique en rapport avec un type d'unité bien particulier. Ainsi nous sommes en mesure de proposer trois cadres de modélisations dédiés à l'unité neuro-vasculaire, à la pharmacie hospitalière et au bloc opératoire, comportant une déclinaison d'outils et de modules spécifiques. La suite logique de ce travail consisterait donc à étendre nos observations à d'autres types de services médicaux pour enrichir la bibliothèque que représente medPRO : cet enrichissement peut être réalisé de manière communautaire par n'importe quel utilisateur

de la plate-forme. Nous compensons ainsi l'impossibilité de réutiliser un précédent modèle dans divers hôpitaux par une aide riche permettant l'adaptation de modèles existants à d'autres systèmes.

Les réseaux de Petri sont utilisés dans cette thèse pour formaliser le comportement dynamique des modèles UML. Les possibilités offertes par cet outil sont très vastes et mériteraient d'être explorées plus avant afin de mettre en valeur certaines propriétés mathématiques pour l'analyse de systèmes hospitaliers. L'application de telles méthodes permettrait de détecter avant la phase de simulation des problèmes structurels ou organisationnels. L'extension des méthodes de planification et d'ordonnancement proposées dans ce mémoire entre dans le même cadre : il serait judicieux d'enrichir ces méthodes pour une meilleure prise en compte des spécificités des systèmes modélisés grâce aux réseaux de Petri de santé.

Le système de pilotage constitue un chantier scientifique également très vaste : la nature même du réseau social existant dans les hôpitaux se prête parfaitement à l'application de structures holoniques, permettant la mise en œuvre de techniques de négociation et d'apprentissage au cours de la simulation. Sans aller jusqu'à l'application de techniques de simulation multi-agents, l'enrichissement des modules de décision proposés dans le cadre du système de contrôle est inévitable, au même titre que l'analyse de systèmes hospitaliers supplémentaires.

Enfin le cadre de modélisation UML proposé peut être étendu afin de formaliser certains processus complexes : la section 3.8 offre un aperçu de projets d'extensions permettant d'accroître la précision de la modélisation. La préemption d'activités est une notion courante dans le milieu médical (essentiellement dans un contexte d'urgence) qui mériterait d'être étudiée.

Pour conclure ce mémoire, nous aimerions attirer l'attention du lecteur sur l'enthousiasme des différents intervenants (chirurgiens, médecins, pharmaciens, infirmiers, etc.) quant à la mise en œuvre de la plate-forme medPRO pour la résolution des problèmes organisationnels cités plus haut. Le succès d'une étude visant à réorganiser un système hospitalier repose essentiellement sur la méthodologie de travail adoptée et sur l'effort de communication fourni : la prise en compte du réseau social (individualités et équipes) que nous avons tenté d'intégrer dans notre approche est un élément déterminant pour la réussite de toute étude scientifique.

Rappelons enfin que l'organisation d'un hôpital est unique : l'application de méthodes ou d'outils dits « génériques » ne sera jamais aisée, car c'est avec des humains que nous travaillons et non avec des machines.

Annexes

Annexe A

Notions de base sur les réseaux de Petri

Nous proposons dans cette annexe un récapitulatif de l'ensemble des notions utilisées dans ce mémoire sur les réseaux de Petri. Un réseau de Petri est un modèle mathématique permettant la représentation de systèmes distribués discrets (informatiques, industriels, etc.) introduit par Petri (1962). Un réseau de Petri est également un langage de modélisation représenté sous forme d'un graphe biparti orienté. La majeure partie des définitions et propriétés présentées dans cette annexe sont tirées de (Proth et Xie, 1995a).

A.1 Définition informelle

Un réseau de Petri est représenté sous forme d'un graphe biparti orienté composé de nœuds appelés *places* et *transitions*. Places et transitions sont connectées grâce à des arcs orientés possédant un certain poids. Une place précédant une transition liée par un arc est appelée place d'entrée de la transition. Une place succédant une transition liée par un arc est appelée place de sortie de la transition. Les places peuvent contenir un certain nombre de *jetons*. La distribution des jetons dans les places du réseau est appelée le marquage du réseau de Petri. Les transitions agissent sur les jetons en entrée grâce à un processus appelé *tir*. Une transition est validée si elle peut être tirée, i.e. si un nombre de jetons égal au poids de l'arc sont présents dans chaque place d'entrée. Lorsqu'une transition est tirée, les jetons en entrée sont retirés et un certain nombre de jetons sont restitués dans les places de sortie.

L'exécution d'un réseau de Petri est stochastique : (i) plusieurs transitions peuvent être validées au même moment, et (ii) aucune règle n'exige que ces transitions soient tirées à un moment précis. Ainsi les réseaux de Petri sont particulièrement adaptés à la modélisation de comportements concurrents de systèmes distribués. Des règles de tir peuvent être fixées pour piloter l'exécution du réseau.

A.2 Définition formelle

Desel et Gabriel (2001) proposent de définir un réseau de Petri de la manière suivante :

Définition A.2.1 (Réseau de Petri) *Un réseau de Petri est un quintuplet* $\mathcal{R}$ *tel que* $\mathcal{R} = (P, T, F, M_0, W)$*, où :*

- P *est l'ensemble des places du réseau ;*
- T *est l'ensemble des transitions du réseau,* $P \cap T = \oslash$ *;*
- F *est l'ensemble des arcs du réseau,* $F \subseteq (S \times T) \cup (T \times S)$ *;*

– $M_0 : S \to \mathbb{N}$ *est le marquage initial du réseau ;*
– $W : F \to \mathbb{N}^+$ *est l'ensemble des poids des arcs du réseau.*

Remarque A.2.1 *Notons que* $P \cap T = \oslash$.

A.3 Fonctionnement d'un réseau

Dans la suite de ce mémoire nous notons :
$\bullet t$ l'ens. des places d'entrée de la transition t, i.e. l'ens. des places p telles que $(p,t) \in A$.
$t\bullet$ l'ens. des places de sortie de la transtion t, i.e. l'ens. des places p telles que $(t,p) \in A$.
$\bullet p$ l'ens. des transitions d'entrée de la place p, i.e. l'ens. des transitions t telles que $(t,p) \in A$.
$p\bullet$ l'ens. des transitions de sortie de la place p, i.e. l'ens. des transitions p telles que $(p,t) \in A$.

Définition A.3.1 (Tir d'une transition) *Une transition t est tirable, franchissable ou validée lorsque :*

$$\forall p \in \bullet t,\ M(p) \geq W(p,t)$$

Lorsqu'une transition est validée dans le marquage M_0*, on note* $M_0[t>$.

Tirer une transition consiste à :
– retirer $W(p,t)$ jetons de toute place $p \in \bullet t$;
– ajouter $W(t,p)$ jetons dans toute place $p \in t\bullet$.

Définition A.3.2 (Évolution du marquage) *Le franchissement d'une transition* t *de* T *validée dans le marquage* M *conduit au marquage* M_1*, défini par :*

$$\forall p \in P,\ \forall t \in T,\ M_1(p) = M(p) + W(t,p) - W(p,t)$$

On note $M[t > M_1$*. L'ensemble des marquages qu'il est possible d'atteindre à partir de* M *en effectuant un ou plusieurs tirages est noté* $R(M)$.

Définition A.3.3 (Séquence de franchissement) *Une séquence de franchissement est un mot construit sur l'alphabet* T^* *des transitions de* T*. On note* σ *une séquence de franchissements.*

Pour caractériser une séquence de franchissement σ*, on utilise son image commutative* $\vec{\sigma}$.

Exemple A.3.1

$\sigma = t_1 t_2$ *et* $M[\sigma > M_2$*. Si* $T = \{t_1, t_2, t_3\}$, $\vec{\sigma} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$.

Chaque composante de l'image commutative est le nombre d'occurrences de la transition correspondante dans σ.

A.4 Définitions particulières

Définition A.4.1 (Transition source et transition puits) *Une transition sans place d'entrée est appelée transition source. Une telle transition est toujours tirable. Une transition sans place de sortie est appelée transition puits. Si une telle transition est tirée, les jetons sont prélevés suivant les règles habituelles, mais aucun jeton n'est produit en sortie.*

Définition A.4.2 (Réseau pur) *Un réseau de Petri est dit pur s'il ne contient pas de boucle* (p,t) *telle que* $p \in \bullet t$ *et* $p \in t\bullet$.

A.5 Matrice d'incidence et équation d'état

Définition A.5.1 (Matrice d'incidence) *La matrice d'incidence $U = [u_{ij}]$ ($i \in \{1, \ldots, n\}$ et $j \in \{1, \ldots, m\}$) d'un réseau de Petri pur est définie de la manière suivante :*

$$u_{ij} = \begin{cases} W(t_j, p_i) & si\ t_j \in \bullet p_i \\ -W(p_i, t_j) & si\ t_j \in p_i \bullet \\ 0 & sinon \end{cases}$$

n étant le nombre de places et m le nombre de transitions du réseau considéré.

Théorème A.5.1 (Équation d'état) *Soit σ une séquence finie de transitions tirables depuis un marquage M d'un réseau $\mathcal{R}$ de matrice d'incidence U. On a le résultat fondamental suivant :*

$$M[\sigma > M_1 \Rightarrow M_1 = M + U\vec{\sigma} \tag{A.1}$$

$\vec{\sigma}$ est appelé vecteur de comptage de σ.

A.6 t-invariant

Définition A.6.1 (t-invariant) *Soit $\vec{w}$ un vecteur de dimension q (q étant le nombre de transitions du réseau de Petri considéré) dont les composantes sont entières et non négatives. $\vec{w}$ est un t-invariant si $U\vec{w} = 0$, où U est la matrice d'incidence du réseau de Petri.*

Théorème A.6.1 *Soit σ une séquence tirable et $\vec{\sigma}$ le vecteur de comptage correspondant. Si $\vec{\sigma}$ est un t-invariant, M_0 le marquage initial et $M \in R(M_0)$ le marquage obtenu après le tirage de σ, alors $M = M_0$.*

Preuve A.6.1 *Considérons l'équation d'état (A.1) : $M = M_0 + U\vec{\sigma}$. Comme $\vec{\sigma}$ est un t-invariant, alors par définition $U\vec{\sigma} = 0$ et l'équation d'état se réduit à $M = M_0$.*

Définition A.6.2 (Support du t-invariant) *L'ensemble des transitions qui correspondent aux composantes non nulles d'un t-invariant est appelé support du t-invariant considéré. Si $\vec{w}$ est un t-invariant, son support sera noté $||\vec{w}||$.*

Définition A.6.3 (Support minimal) *Le support d'un t-invariant est dit minimal s'il ne contient aucun ensemble non vide qui soit un support de t-invariant.*

Définition A.6.4 (t-invariant minimal) *Un t-invariant minimal est un t-invariant $\vec{w}$ tel qu'il n'existe aucun autre t-invariant ayant toutes ses composantes inférieures ou égales aux composantes correspondantes de $\vec{w}$.*

Théorème A.6.2 *Tout t-invariant est une combinaison linéaire de t-invariants minimaux.*

A.7 Réseau de Petri T-temporisé

Deux types de temporisations sont présentées dans la littérature : la temporisation de places et la temporisation de transitions. Nous décrirons plus avant la temporisation de transitions dans cette section étant donné que ce mécanisme sera utilisé dans la suite de ce mémoire.

Définition A.7.1 (Réseau de Petri T-temporisé) *Soit $\mathcal{R}$ un réseau de Petri. Un réseau de Petri T-temporisé est un couple $R = (\mathcal{R}, \theta)$ dans lequel :*

- *$\mathcal{R}$ est le réseau sous-jacent ;*

– $\theta : T \rightarrow \mathbb{N}^+$ *est l'application de temporisation qui associe à toute transition* t *sa durée de séjour* $\theta(t)$.

L'algorithme de franchissement d'une transition t peut être décrit comme suit. On associe le temps θ à la transition t dont le franchissement débute à l'instant T_0.

1. retirer $W(p,t)$ jetons de tout $p \in {}^\bullet t$ (i.e. de toutes les places d'entrée de t) à l'instant T_0.
2. ajouter $W(t,p)$ jetons dans tout $p \in t^\bullet$ (i.e. dans toutes les places de sortie de t) à l'instant $T_0 + \theta$.

Les jetons sont supposés séjourner dans la transition entre les instants T_0 et $T_0 + \theta$.

Remarque A.7.1 *La durée de séjour d'un ou plusieurs jetons dans une transition temporisée* t *est supérieure ou égale à* $\theta(t)$.

A.8 Réseau de Petri décomposable

Seuls les réseaux de Petri possédant des transitions d'entrée (transitions sources) et des transitions de sortie (transitions puits) sont considérés dans cette section.

Définition A.8.1 *Soit* $\vec{w}$ *un t-invariant d'un réseau de Petri* N. *Un RdP* $N_{\vec{w}}$ *est dit* $\vec{w}$*-dérivé du RdP* N *ayant des places d'entrée et de sortie si :*
(i) l'ensemble des transitions de $N_{\vec{w}}$ *est* $||\vec{w}||$;
(ii) $\forall t \in ||\vec{w}||$, ${}^\bullet t$ *et* $t^\bullet$ *sont identiques dans* N *et* $N_{\vec{w}}$;
(iii) tout arc de $N_{\vec{w}}$ *a le même poids que l'arc correspondant de* N.

Définition A.8.2 (CFIO) *Nous appelons* $\vec{w}$*-CFIO de* N, *où* $\vec{w}$ *est un t-invariant de* N, *le réseau de Petri* $N_{\vec{w}}$ $\vec{w}$*-dérivé de* N *lorsqu'il possède les propriétés suivantes :*
(i) $p^\bullet$ *est unique pour toutes les places de* $N_{\vec{w}}$ *(chaque place a exactement une transition de sortie) ;*
(ii) il existe au moins une transition $t_1 \in ||\vec{w}||$ *et une transition* $t_2 \in ||\vec{w}||$ *telles que* ${}^\bullet t_1 = \oslash$ *(*t_1 *est une transition source) et* $t_2{}^\bullet = \oslash$ *(*t_2 *est une transition puits) ;*
(iii) $N_{\vec{w}}$ *est acyclique, i.e.* $N_{\vec{w}}$ *ne contient aucun circuit élémentaire.*

L'acronyme CFIO signifie *Conflict Free net with Input and Output transitions* (i.e. *Réseau de Petri sans conflit avec transitions d'entrée et transitions de sortie*). Nous pouvons maintenant définir ce qu'est un réseau de Petri décomposable.

Définition A.8.3 (Réseau de Petri décomposable) *Soit* N *un réseau de Petri et* $\vec{w_1}, \ldots, \vec{w_k}$ *un ensemble de t-invariants de* N *tels que :*
(i) les réseaux $N_{\vec{w_i}}$ $\vec{w_i}$*-dérivés de* N *sont des* $\vec{w_i}$*-CFIO, avec* $i \in \{1, \ldots, k\}$;
(ii) les $\vec{w_i}$*-CFIO couvrent* N *:* $N = N_{\vec{w_1}} \cup \cdots \cup N_{\vec{w_k}}$.
Alors N *est dit décomposable.*

Les réseaux de Petri décomposables sont utilisés pour modéliser des systèmes de production si on les considère du point de vue de leur planification à court terme. Dans ce cas il est possible de les compléter pour aboutir à un modèle utilisable pour l'ordonnancement.

A.9 Réseau de Petri coloré

Les réseaux de Petri colorés permettent de réduire la taille de modèles réalisés avec les réseaux de Petri ordinaires introduits précédemment. Ils sont donc souvent utilisés pour modéliser des systèmes de

production. Cependant les réseaux de Petri colorés possèdent très peu de propriétés analytiques, leur analyse n'est donc possible que par la simulation.

Les réseaux de Petri colorés permettent l'utilisation de données typées ainsi que la manipulation de données complexes : chaque jeton possède une valeur appelée « couleur du jeton ». Différents types de franchissements sont associés à chaque transition : les jetons évoluent dans le réseau par les tirs des transitions suivant les différents types de franchissements possibles.

Formellement, un RdP coloré est défini de la manière suivante. Soit Ω l'ensemble des couleurs du réseau.

Définition A.9.1 (Réseau de Petri coloré) *Un réseau de Petri coloré est un septuplet* $\mathcal{C} = (P, T, F, C, W^+, W^-, M_0)$ *tel que :*

(i) P *est l'ensemble des places ;*

(ii) T *est l'ensemble des transitions ;*

(iii) $F \subseteq (P \times T) \cup (T \times P)$ *est l'ensemble fini des arcs : un arc relie une place à une transition ou une transition à une place ;*

(iv) $C : (P \cup T) \to \Omega$ *est une fonction attachée aux places et aux transitions du réseau :* $C(p)$, $p \in P$, *est l'ensemble des couleurs associées à la place* p *(i.e. l'ensemble des couleurs que peut contenir la place* p*) et* $C(t)$, $t \in T$, *est l'ensemble des couleurs associées à la transition* t *(i.e. l'ensemble des manières de franchir la transition* t*) ;*

(v) $W^-_{p,t} : C(t) \to \mathbb{N}^{|C(P)|}$ *est l'application qui associe à chaque manière de franchir* t *(i.e. à chaque couleur associée à* t*) un vecteur qui a autant d'éléments que de couleurs dans les places du RdP. La valeur du ième élément de ce vecteur est un entier qui indique le nombre de jetons de cette couleur nécessaires dans* p *pour franchir la transition* t *;*

(vi) $W^+_{t,p} : C(t) \to \mathbb{N}^{|C(P)|}$ *est l'application qui associe à chaque manière de franchir* t *(i.e. à chaque couleur associée à* t*) un vecteur dont les composantes sont des entiers qui indiquent le nombre de jetons de chaque couleur obtenus à l'issue du tirage en question ;*

(vii) M_0 *est le marquage initial permettant de connaître le nombre de jetons de chaque couleur qui se trouvent initialement dans chacune des places du réseau.*

Le franchissement d'une transition t par rapport à une couleur $a \in C(t)$ consiste à (i) soustraire du marquage de tout $p \in {\bullet t}$ le vecteur $W^-_{p,t}(a)$, puis à (ii) ajouter au marquage de tout $p \in t{\bullet}$ le vecteur $W^+_{t,p}(a)$.

Annexe B

Algorithmes

Cette annexe regroupe les algorithmes utilisés pour la conversion du modèle UML appartenant au sous-système physique de la plate-forme medPRO en réseau de Petri de santé. Ces algorithmes sont donnés dans l'ordre d'exécution normal pour la conversion intégrale d'un modèle. Nous invitons le lecteur à se reporter à la section 4.2.1 pour la définition des ensembles et fonctions utilisés au cours de la procédure de conversion.

Cette annexe comprend également les algorithmes utilisés par le système de contrôle pendant la simulation pour la sélection de ressources au début d'une mission ou pour la réalisation d'une tâche liée à une transition. L'ensemble de ces algorithmes utilisent une primitive appelée **tirer**, permettant de tirer une transition. Par exemple, **tirer** $t(1)$ permet de tirer la transition t avec le type de franchissement 1.

B.1 Conversion de la vue processus

Cet algorithme permet la création du réseau de Petri associé à la vue processus sans tenir compte des allocation/libération de ressources. Pour chaque machine d'état $PM = (E, V, \zeta)$ appartenant à la vue processus, l'algorithme de conversion suivant est appliqué :

1. Convertir l'entrée en un couple transition source $t_{in} \in T_{in}$ – place $p \in P$; ajouter l'arc (t_{in}, p) à l'ensemble F.
2. Convertir la sortie en un couple place $p \in P$ – transition puits $t_{out} \in T_{out}$; ajouter l'arc (p, t_{out}) à l'ensemble F.
3. Convertir chaque P-état $e = (l, \tau, R) \in E$ en un triplet place d'entrée $p_1 \in P$ – transition $t \in TE$ – place de sortie $p_2 \in P$; ajouter les arcs (p_1, t) et (t, p_2) à l'ensemble F. La transition t est associée à l'état e : $\gamma(e) = t$. La durée de franchissement de t est initialisée : $\theta(t) = \tau$.
4. Pour chaque transition conditionnelle $(e, f) \in V$: (i) créer une transition $t \in TC$, (ii) ajouter $(\gamma(e)\bullet, t)$, $(t, \bullet\gamma(f))$ à l'ensemble F, (iii) associer la transition t à (e, f) : $\eta(e, f) = t$ et (iv) associer la condition à t : $\xi(t) = \zeta(e, f)$.
5. $W(x, y) = 1 \ \forall \ (x, y) \in F$.

Algorithme B.1 Conversion de la vue processus

1 $T_{in} \leftarrow T_{in} \cup \{t_{in}\}$; $T_{out} \leftarrow T_{out} \cup \{t_{out}\}$
2 $P \leftarrow P \cup \{p_1,\ p_2\}$
3 $F \leftarrow F \cup \{(t_{in}, p_1),\ (p_2, t_{out})\}$
4 $W(t_{in}, p_1) = W(p_2, t_{out}) = 1$
5 **pour** chaque état $e_i = (l_i, \tau_i, R_i) \in E$, $i \in \{1, \ldots, n\}$ **faire**
6 $\quad TE \leftarrow TE \cup \{t_i\}$
7 $\quad P \leftarrow P \cup \{p_{i,1},\ p_{i,2}\}$
8 $\quad F \leftarrow F \cup \{(p_{i,1}, t_i),\ (t_i, p_{i,2})\}$
9 $\quad W(p_{i,1}, t_i) = W(t_i, p_{i,2}) = 1$
10 $\quad \gamma(e_i) = t_i$; $\theta(t_i) = \tau_i$
11 **fin pour**
12 **pour** chaque transition $(e_i, f_i) \in V$, $i \in \{1, \ldots, n\}$ **faire**
13 $\quad TC \leftarrow TC \cup \{t_i\}$
14 $\quad F \leftarrow F \cup \{(\gamma(e)\bullet, t),\ (t, \bullet\gamma(f))\}$
15 $\quad W(\gamma(e)\bullet, t) = W(t, \bullet\gamma(f)) = 1$
16 $\quad \eta(e_i, f_i) = t_i$; $\xi(t_i) = \zeta(e, e_i)$
17 **fin pour**

B.2 Conversion des ressources déclarées vue organisation

Une fois la vue processus convertie en réseau de Petri, nous allons ajouter les états de base correspondant aux ressources simples, équipes et compétences déclarées vue organisation :

1. Pour chaque ressource simple $r \in R$ d'effectif eff, créer un triplet transition source $t_{in} \in T_{in}$ – place de base $p \in PB$ – transition puits $t_{out} \in T_{out}$; ajouter les arcs (t_{in}, p) et (p, t_{out}) à l'ensemble F. La place de base p est associée à la ressource simple $r : \Gamma(r) = p$. De plus nous avons $W(t_{in}, p) = W(p, t_{out}) = eff$.
2. Pour chaque équipe et compétence $r \in EQ \cup CO$, créer une place de base $p \in PB$. Cette place est associée à la ressource $r : \Gamma(r) = p$.

Algorithme B.2 Conversion des ressources déclarées dans la vue organisation

1 **pour** chaque ressource $r \in R$ d'effectif eff **faire**
2 $\quad T_{in} \leftarrow T_{in} \cup \{t_{in}\}$
3 $\quad T_{out} \leftarrow T_{out} \cup \{t_{out}\}$
4 $\quad PB \leftarrow PB \cup \{p_{base}\}$
5 $\quad F \leftarrow F \cup \{(t_{in}, p_{base}),\ (p_{base}, t_{out})\}$
6 $\quad \Gamma(r) = p_{base}$
7 $\quad W(t_{in}, p_{base}) = W(p_{base}, t_{out}) = eff$
8 **fin pour**
9 **pour** chaque ressource $r \in CO \cup EQ$ **faire**
10 $\quad PB \leftarrow PB \cup \{p_{base}\}$
11 $\quad \Gamma(r) = p_{base}$
12 **fin pour**

B.3 Conversion des missions déclarées dans la vue ressource

Soit $mi = (M, e_{in}, e_{out}, eff_{mi})$ avec $M = (E, V, \zeta)$ une mission définie pour la ressource $r = (A, EDT, MI, \zeta, eff_r)$. Nous convertissons tout d'abord la machine d'état correspondant à la mission mi de la même manière qu'une machine d'état de la vue processus :

1. Créer un couple transition source $t_{in} \in T$ – place $p \in P$; ajouter l'arc (t_{in}, p) à l'ensemble F. Ce sous-réseau constitue l'entrée de la mission.
2. Créer un couple place $p \in P$ – transition puits $t_{out} \in T$; ajouter l'arc (p, t_{out}) à l'ensemble F. Ce sous-réseau constitue la sortie de la mission.
3. Convertir chaque R-état $e = (l, \tau) \in E$ en un triplet place d'entrée $p_1 \in P$ – transition $t \in TE$ – place de sortie $p_2 \in P$; ajouter les arcs (p_1, t) et (t, p_2) à l'ensemble F. La transition t est associée à l'état e : $\gamma(e) = t$. La durée de franchissement de t est initialisée : $\theta(t) = \tau$.
4. Pour chaque transition conditionnelle $(e, f) \in V$:
   - créer une transition $t \in TC$;
   - ajouter $(\gamma(e)\bullet, t)$, $(t, \bullet\gamma(f))$ à l'ensemble F ;
   - associer la transition t à (e, f) : $\eta(e, f) = t$;
   - associer la condition à t : $\xi(t) = \zeta(e, f)$.
5. $W(x, y) = 1 \ \forall \ (x, y) \in F$.

Algorithme B.3 Conversion des missions déclarées dans la vue ressource

```
1   T ← T ∪ {t_in, t_out}
2   P ← P ∪ {p_1, p_2}
3   F ← F ∪ {(t_in, p_1), (p_2, t_out)}
4   W(t_in, p_1) = W(p_2, t_out) = 1
5   pour chaque état e_i = (l_i, τ_i, R_i) ∈ E, i ∈ {1, ..., n} faire
6       TE ← TE ∪ {t_i}
7       P ← P ∪ {p_i,1, p_i,2}
8       F ← F ∪ {(p_i,1, t_i), (t_i, p_i,2)}
9       W(p_i,1, t_i) = W(t_i, p_i,2) = 1
10      γ(e_i) = t_i
11      θ(t_i) = τ_i
12  fin pour
13  pour chaque transition (e_i, f_i) ∈ V, i ∈ {1, ..., n} faire
14      TC ← TC ∪ {t_i}
15      F ← F ∪ {(γ(e)•, t), (t, •γ(f))}
16      W(γ(e)•, t) = W(t, •γ(f)) = 1
17      η(e_i, f_i) = t_i
18      ξ(t_i) = ζ(e, e_i)
19  fin pour
```

B.4 Synchronisation des réseaux des missions et de la vue processus

L'algorithme B.3 permet d'obtenir le réseau de Petri associé à la mission mi, dont l'entrée et la sortie sont modélisées par une transition source et une transition puits respectivement. La synchronisation entre ce réseau de Petri et le réseau associé à la vue processus créé précédemment est réalisée en fusionnant les transitions communes et les sous-réseaux de transitions communs de ces machines d'état.

B.5 Allocation ponctuelle de ressources

Reproduire le mécanisme d'allocation ponctuelle (la ressource est allouée pour une seule tâche et libérée immédiatement) consiste à relier une place de base à une transition modélisant un état. Pour chaque état $e = (l, \tau, R)$, l'ensemble R est examiné. Pour chaque ressource $(r, eff) \in R$, les arcs $(\gamma(pe), \Gamma(r))$ et $(\Gamma(r), \gamma(pe))$ sont ajoutés à F. De plus nous avons $W(\gamma(pe), \Gamma(r)) = W(\Gamma(r), \gamma(pe)) = eff$.

Algorithme B.4 Allocation ponctuelle de ressources

```
1  pour chaque état e = (l, τ, R) faire
2      pour chaque couple (r, eff) ∈ R faire
3          F ← F ∪ {(Γ(r), γ(pe)), (γ(pe), Γ(r))}
4          W(γ(pe), Γ(r)) = W(Γ(r), γ(pe)) = eff
5      fin pour
6  fin pour
```

B.6 Sélection d'une ressource et/ou de remplaçants

L'algorithme décrit ici permet de sélectionner une ressource ou un remplaçant. Soit r une ressource associée à la place de base pr. L'effectif requis pour la ressource r est fixé à eff. La ressource r possède n suppléants $q_1, \ldots, q_n$ dont les places de base sont $pq_1, \ldots, pq_n$ ($q_i \triangleright r \ \forall i$). Chaque place de base pq_i est liée à pr par l'intermédiaire d'une transition t_i. La figure B.1 illustre ce sous-réseau.

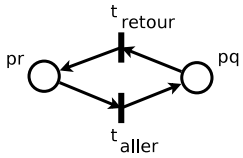


FIG. B.1 – Sous-réseau de Petri correspondant à un remplacement

La variable cpt initialisée à $eff - M(pr)$ ligne 1 permet de connaître le nombre d'instances de la ressource r manquantes. Tant que des instances de ressource manquent ($cpt > 0$), les remplaçants de r sont examinés et alloués s'ils sont disponibles (lignes 3–9). Lorsque l'ensemble des remplaçants a été parcouru, la fonction renvoie 1 si un nombre suffisant d'instances ont été allouées, 0 sinon.

Algorithme B.5 Sélection d'une ressource ou d'un remplaçant

```
1   cpt ← eff − M(pr)
2   i ← 1
3   tant que cpt > 0 ∧ i ≤ n faire
4       tant que cpt > 0 ∧ M(pq_i) > 0 faire
5           tirer t_i
6           cpt ← cpt − 1
7       fin tant que
8       i ← i + 1
9   fin tant que
10  si cpt ≤ 0
11      renvoyer 1
12  sinon
13      renvoyer 0
14  fin si
```

B.7 Sélection d'une ressource sur compétences

L'algorithme décrit ici permet de sélectionner une ressource en fonction d'une compétence requise. Nous considérons le sous-réseau de Petri associé à la relation de compétence présenté figure B.2. L'effectif requis pour la compétence considérée est fixé à eff. Soit $pr_1, \ldots, pr_n$ l'ensemble des places de base des ressources $r_1, \ldots, r_n$ possédant la compétence considérée. Soit t la transition permettant l'allocation de la ressource désirée, il existe n manières de tirer t. La place de base de la compétence considérée est p. Nous supposons que eff instances de ressources possédant la compétence considérée sont requises.

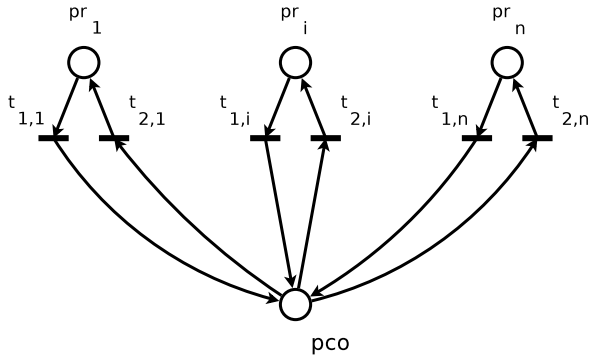


FIG. B.2 – Sous réseau de Petri correspondant à une compétence

La variable *cpt* initialisée à $eff - M(p)$ ligne 1 permet de connaître le nombre d'instances de ressource manquantes pour la compétence considérée. Toutes les ressources r_i possédant la compétence requise sont examinées une première fois, la transition $t(i)$ étant tirée autant de fois que possible (lignes 3–9). La fonction renvoie 1 si un nombre suffisant d'instances ont été allouées, 0 sinon ; les instances de ressource déjà réservées se trouvent toutes dans la place p.

Algorithme B.6 Sélection d'une ressource sur compétences

```
1   cpt ← eff − M(p)
2   i ← 1
3   tant que cpt > 0 ∧ i ≤ n faire
4       tant que cpt > 0 ∧ M(pr_i) > 0
5           tirer t(i)
6           cpt ← cpt − 1
7       fin tant que
8       i ← 1
9   fin tant que
10  renvoyer(cpt ≤ 0)
```

B.8 Constitution d'une équipe

Nous considérons le sous-réseau de Petri associé à la relation d'équipe présenté figure B.3. Soit $pr_1, \ldots, pr_n$ l'ensemble des places de base des ressources $r_1, \ldots, r_n$ membres de l'équipe considérée. Soit t la transition permettant l'allocation des membres de l'équipe. Les places $pr'_1, \ldots, pr'_n$ permettent d'allouer temporairement les membres de l'équipe.

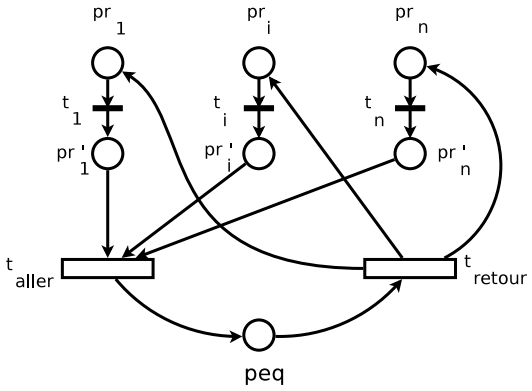


FIG. B.3 – Sous-réseau de Petri correspondant à une équipe

La variable *cpt* initialisée à $W(pr'_i, t) - M(pr'_i)$ ligne 3 permet de connaître le nombre d'instances de ressource manquantes pour chacun des membres. Toutes les ressources r_i de l'équipe sont traitées : si des instances de la ressource r_i manquent ($cpt > 0$), la transition t_i est tirée autant de fois que possible, jusqu'à ce que la place pr_i soit vide ou jusqu'à ce que le nombre d'instances requises soit atteint (lignes 4–7). Si toutes les ressources nécessaires ont pu être réunies, la transition t est tirée pour former l'équipe sous forme d'un jeton. Sinon, les ressources sont mises en attente. La fonction renvoie 1 si la constitution de l'équipe a réussi, 0 sinon.

Algorithme B.7 Constitution d'une équipe

```
1   i ← 1
2   tant que i ≤ n faire
3       cpt ← W(pr'_i, t) − M(pr'_i)
4       tant que cpt > 0 ∧ M(pr_i) > 0
5           tirer t_i
6           cpt ← cpt − 1
7       fin tant que
8       i ← i + 1
9   fin tant que
10  si cpt = 0
11      tirer t
12  fin si
13  renvoyer(cpt = 0)
```

B.9 Initialisation du PLNE de sélection de ressources

L'algorithme B.8 permet de mettre en œuvre le processus de sélection de ressources tel qu'il a été décrit dans la section 5.4.2. Il s'agit d'une procédure permettant d'initialiser les ensembles C et A requis dans les données du problème. Toutes les places $p_i \in {}^\bullet t$ sont examinées (ligne 2) :

- si p_i est la place de base d'une compétence, un choix doit être réalisé parmi les ressources possédant la compétence requise pour le franchissement de cette transition (lignes 4–9) ;
- sinon p_i est la place de base d'une ressource nous sommes dans le cas d'une affectation ponctuelle où la ressource requise pour le franchissement de cette transition (lignes 10–13) ; s'il existe un ensemble de ressources héritières ou remplaçantes, alors la classe de ressource requise peut éventuellement être remplacée par l'une de ses héritières (lignes 14–18).

Si p_i n'est pas une place de base et appartient à la machine d'état d'une mission, alors aucune marge de manœuvre n'existe, le choix de la ressource effectuant cette mission ayant été réalisé en amont. Enfin L'ensemble A_i est dupliqué en fonction de l'effectif nécessaire requis pour chaque ressource (ligne 22).

Algorithme B.8 Initialisation du PLNE de sélection de ressources

```
1   j ← 0
2   pour tout p_i ∈ •t, p_i ∉ PP_in, i ∈ {1,...,n}, faire
3       // p_i est une place appartenant à la méta-ressource c_i
4       si p_i ∈ P_base et c_i = (A, R, MI) ∈ CC
5           pour tout r_k ∈ R, k ∈ {1,...,m} faire
6               A_i ← A_i ∪ {r_k}
7               C ← C ∪ {r_k}
8               e_j ← M(Γ(r_k)) ; j ← j + 1
9           fin pour
10      sinon si p_i ∈ P_base
11          A_i ← {c_i}
12          C ← C ∪ {c_i}
13          e_j ← M(p_i) ; j ← j + 1
14          pour tout r_k ∈ R tel que r_k ▷ Γ(p_i), k ∈ {1,...,m} faire
15              A_i ← A_i ∪ {r_k}
16              C ← C ∪ {r_k}
17              e_j ← M(Γ(r_k)) ; j ← j + 1
18          fin pour
19      fin si
20      A ← A ∪ {A_i}
21  fin pour
22  pour tout A_i ∈ A dupliquer A_i W(p_i, t) fois fin pour
```

B.10 Centre de décision

Le système de pilotage intervient lorsqu'une instance d'entité ou de ressource entre dans la place p d'un centre de décision. Cette place est liée à n transitions conditionnelles en sortie $t_1, \ldots, t_n$. L'algorithme décrit ici permet de choisir la prochaine transition t_i à tirer en évaluant sa condition $\xi(t_i)$. Nous supposons que les transitions sont déjà classées par l'utilisateur et que les expressions $\xi(t_i)$ sont de même nature. Si $\xi(t_i)$ est un test sur une valeur d'attribut, de variable ou d'horloge, les valeurs sont comparées et le résultat est renvoyé (lignes 4–5). Si $\xi(t_i)$ est un test sur l'état d'une entité ou d'une ressource, le marquage de ces places est examiné (lignes 6–15). Enfin, si $\xi(t_i)$ est un tirage aléatoire, la valeur de la variable aléatoire X tirée en début de procédure est comparée avec les probabilités cumulées (lignes 16–20).

Algorithme B.9 Traitement d'un centre de décision

```
1   tirer X ~ U(0,1)
2   s ← 0
3   pour tout t_i ∈ p•, i ∈ {1,...,n}, faire
4       si ξ(t_i) est un test sur la valeur d'un attribut, d'une variable ou d'une
        horloge
5           renvoyer(ξ(t_i))
6       sinon si ξ(t_i) est un test sur l'état d'une entité ou d'une ressource
7           // ξ(t_i) = x < operateur > e, x ∈ U ∪ R et e ∈ E
8           si < operateur >=entre dans
9               renvoyer(M(•γ(e)) > 0)
10          sinon si < operateur >=sort de
11              renvoyer(M(γ(e)•) > 0)
12          sinon si < operateur >=est dans
13              // e est l'état de base de la ressource r
14              renvoyer(M(Γ(r)) > 0)
15          fin si
16      sinon si ξ(t_i) est un tirage aléatoire
17          // ξ(t_i) = x, x ∈ [0,1]
18          s ← s + ξ(t_i)
19          renvoyer(X ≤ s)
20      fin si
21  fin pour
```

Annexe C

Modélisation SADT du processus d'intervention chirurgicale

Nous proposons dans cette annexe le modèle du processus d'intervention chirurgicale présenté dans (Chaabane, 2004) dans le cadre du projet HRP financé par la région Rhône-Alpes. Onze diagrammes SADT organisés sous la forme d'une arborescence détaillent les processus pré-opératoire (des consultations à l'hospitalisation), per-opératoire (de l'admission à la sortie du bloc) et post-opératoire (du retour du bloc à la sortie de l'hôpital). Nous nous focaliserons ici sur les huit diagrammes concernant le processus per-opératoire (A0, A2, A21, A22, A23, A231, A24 et A25).

Ce processus nécessite la disponibilité de plusieurs intervenants : Chirurgiens (CHIR), Médecins Anesthésistes Réanimateurs (MAR), Infirmiers Diplômés d'Etat (IDEs), Infirmiers de Bloc Opératoire Diplômés d'Etat (IBODEs), Infirmiers Anesthésistes Diplômés d'Etat (IADEs), secrétaires, et aides soignants. Le processus décrivant l'intervention sur le patient (A2) se décompose en six phases :

1. Anesthésier le patient (A21) ;
2. Préparer le patient à l'intervention (A22) ;
3. Pratiquer l'intervention (A23) ;
4. Surveiller le réveil du patient (A24) ;
5. Préparer la salle opératoire (A25).

Nous invitons le lecteur à se reporter à (Chaabane, 2004) pour une description complète du processus opératoire.

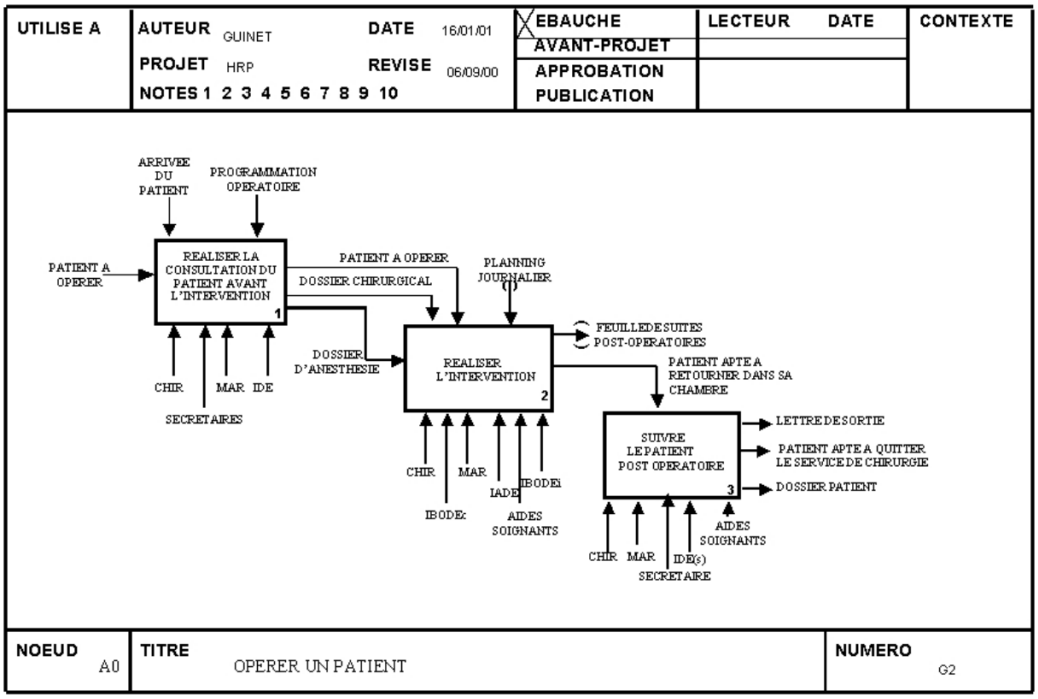


FIG. C.1 – A0 – Opérer un patient

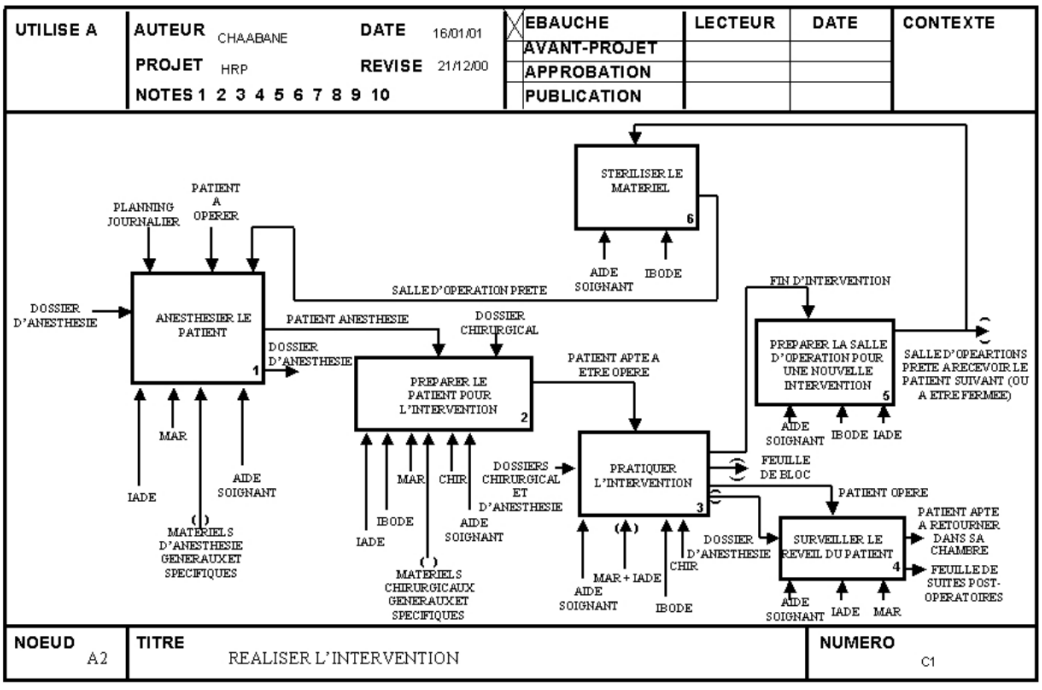


FIG. C.2 – A2 – Réaliser l'intervention

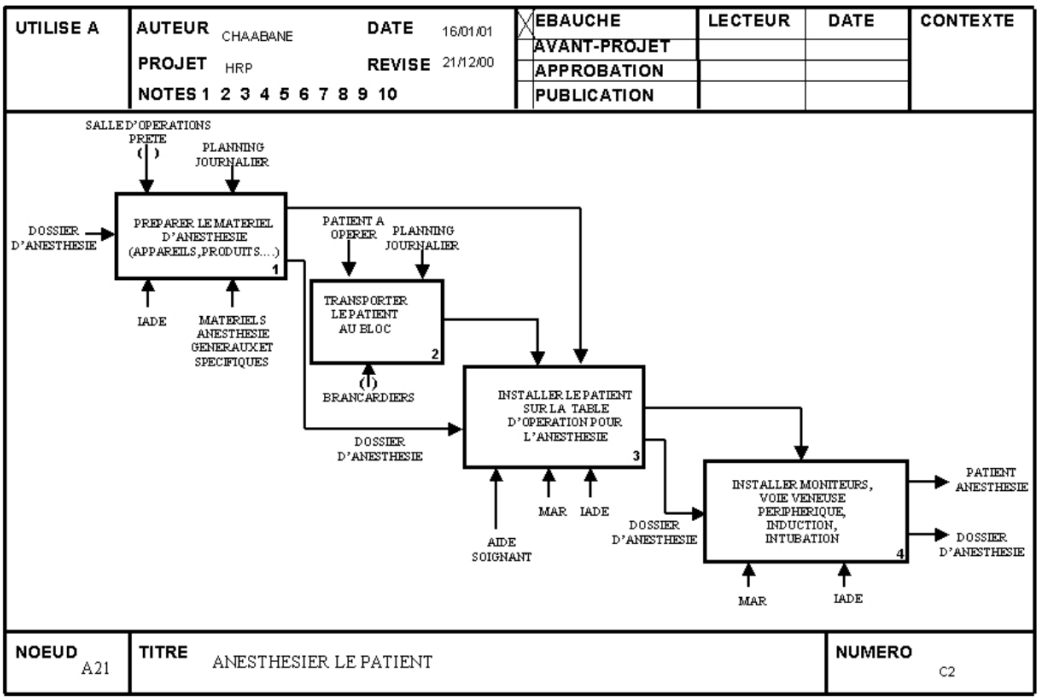


FIG. C.3 – A21 – Anesthésier le patient

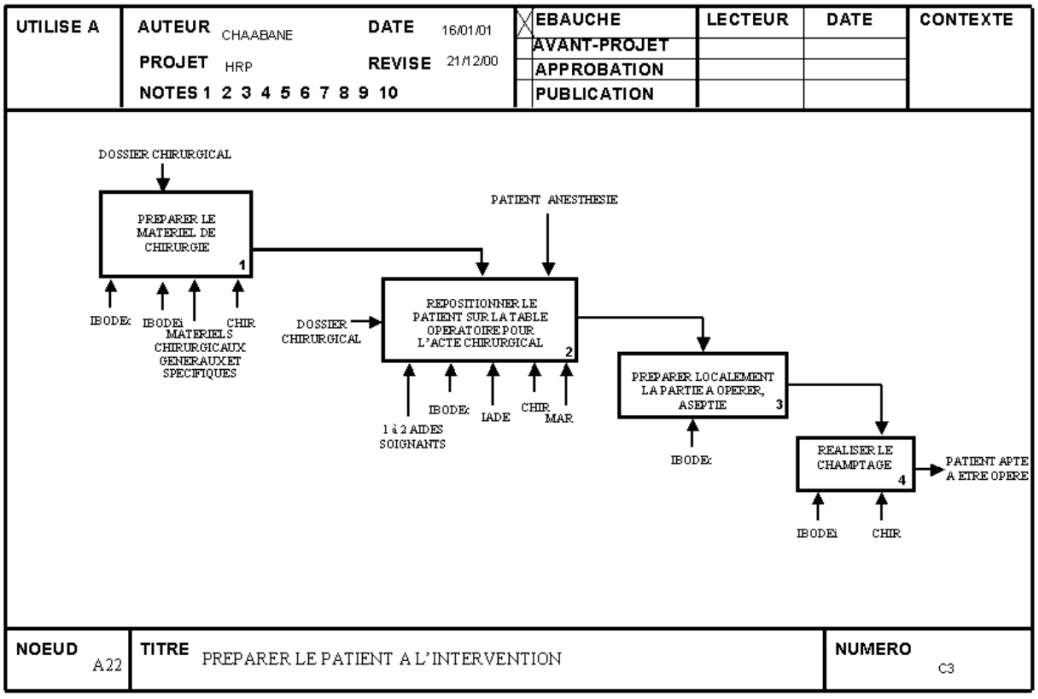


FIG. C.4 – A22 – Préparer le patient à l'intervention

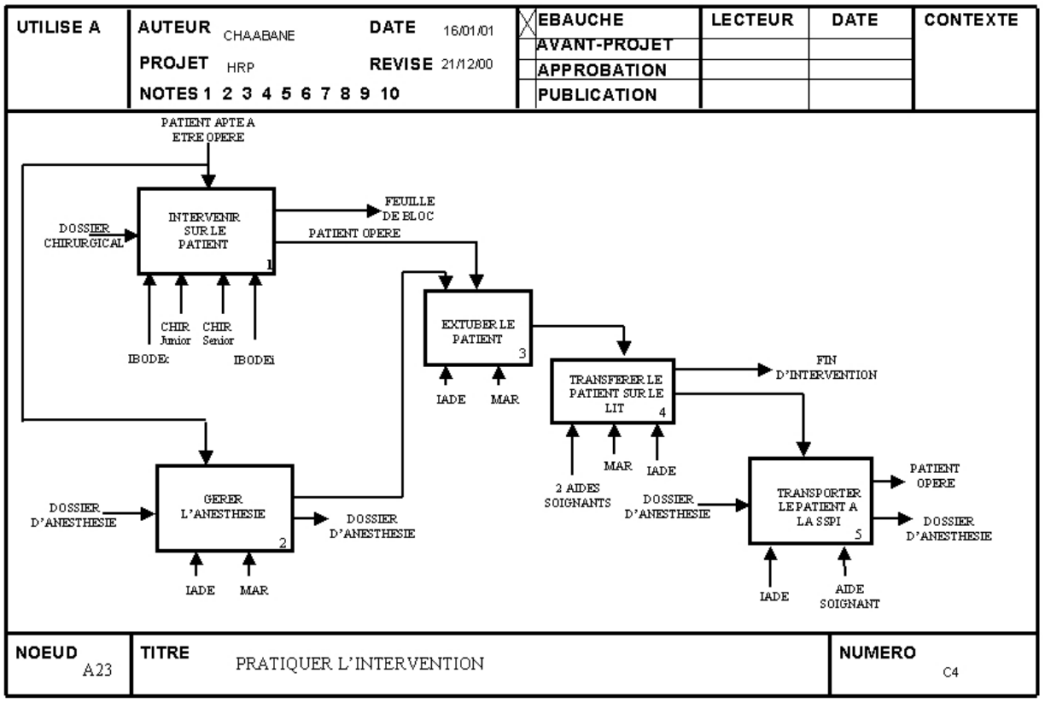


FIG. C.5 – A23 – Pratiquer l'intervention

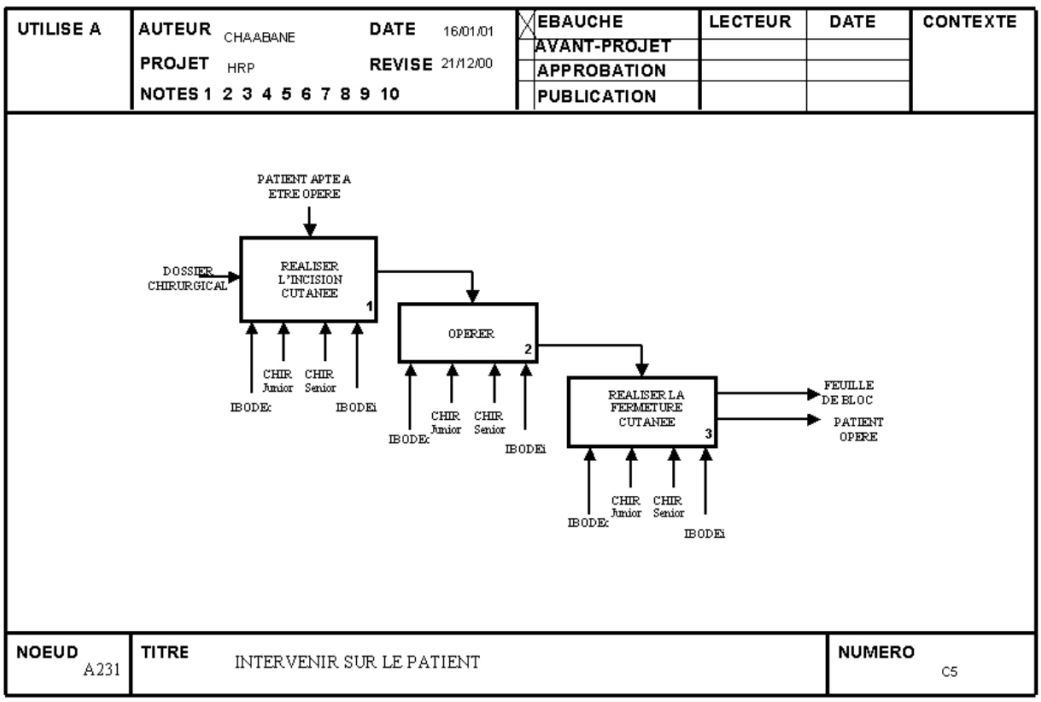


FIG. C.6 – A231 – Intervenir sur le patient

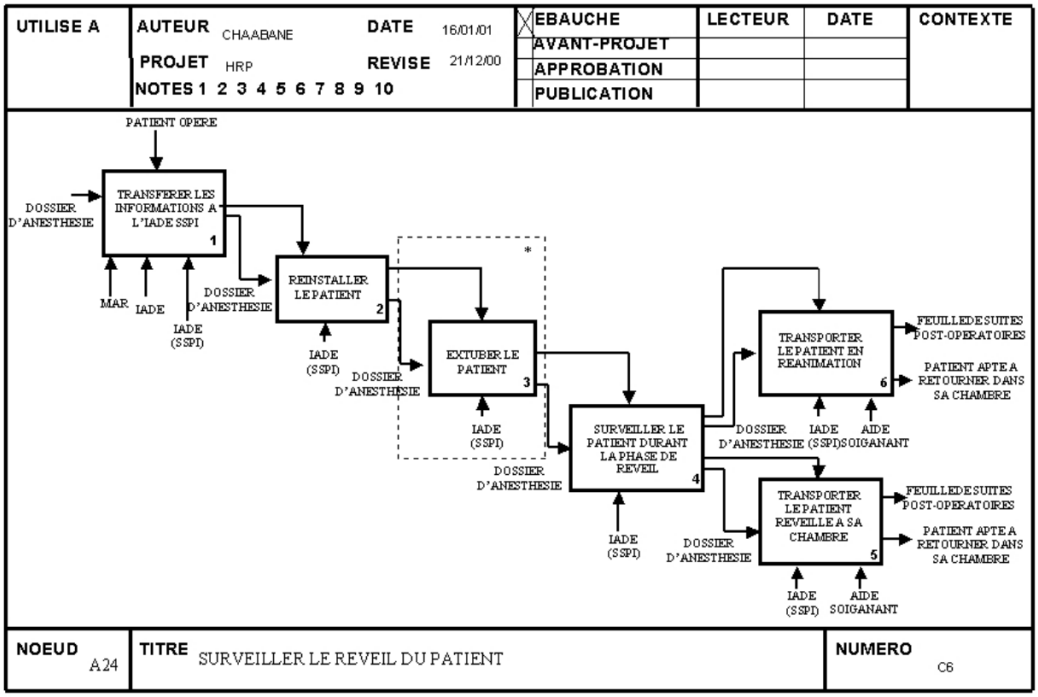


FIG. C.7 – A24 – Surveiller le réveil du patient

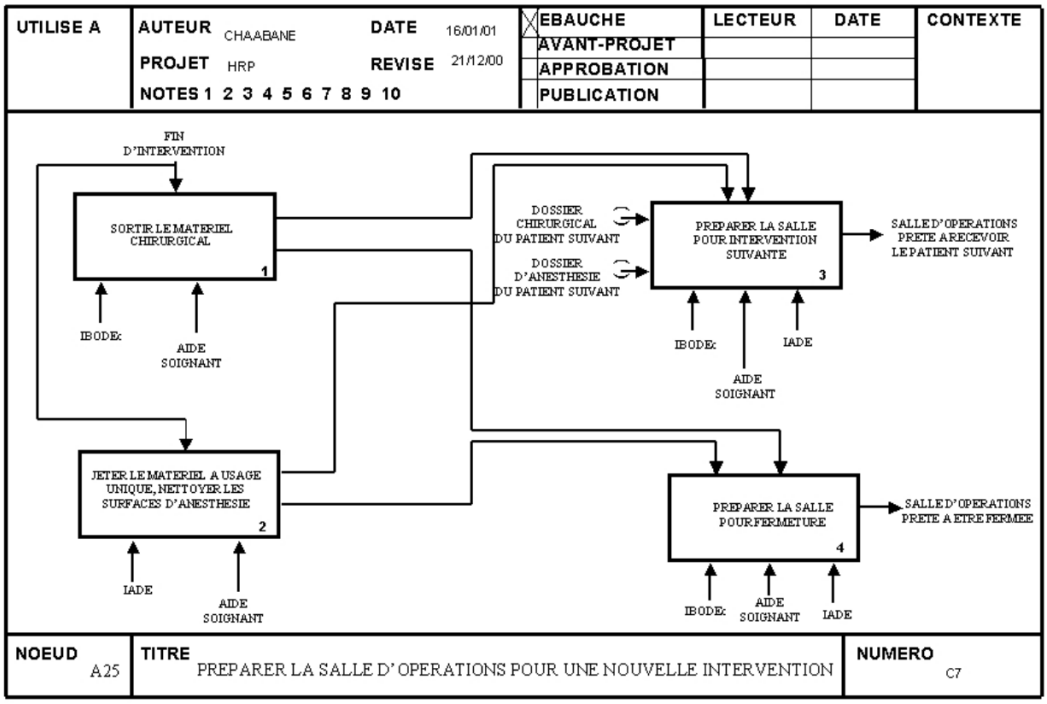


FIG. C.8 – A25 – Préparer la salle d'opérations

Annexe D

Modélisation medPRO du processus d'intervention chirurgicale

Nous proposons dans cette annexe l'intégralité du modèle du processus d'intervention chirurgicale medPRO présenté dans le chapitre 8 de ce mémoire. La description de ce processus et la comparaison avec SADT (c.f. annexe C) sont décrites section 8.4.

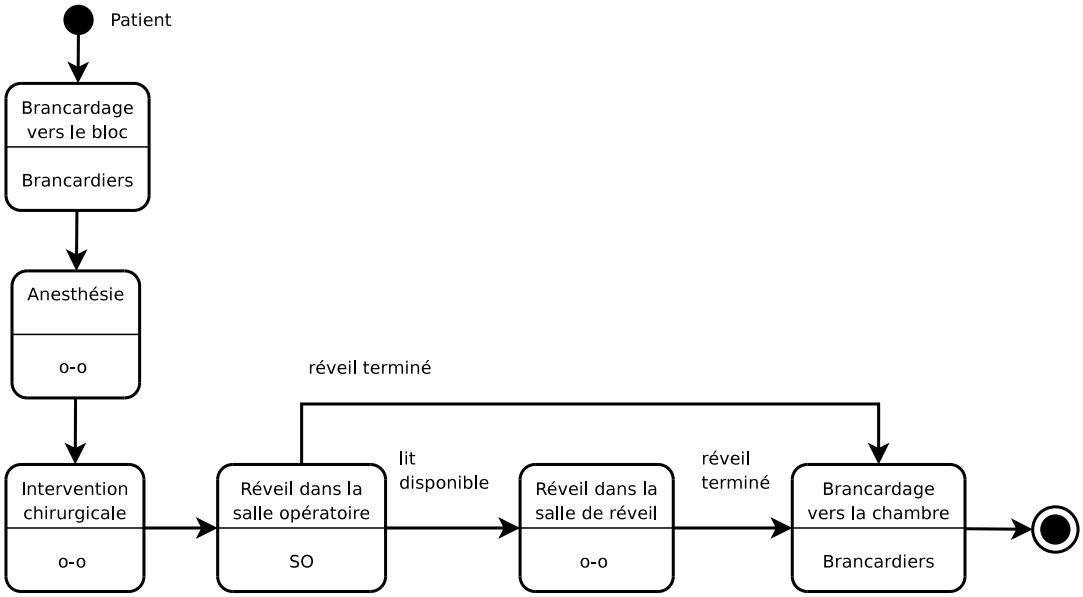


Fig. D.1 – Vue processus macroscopique de l'intervention chirurgicale

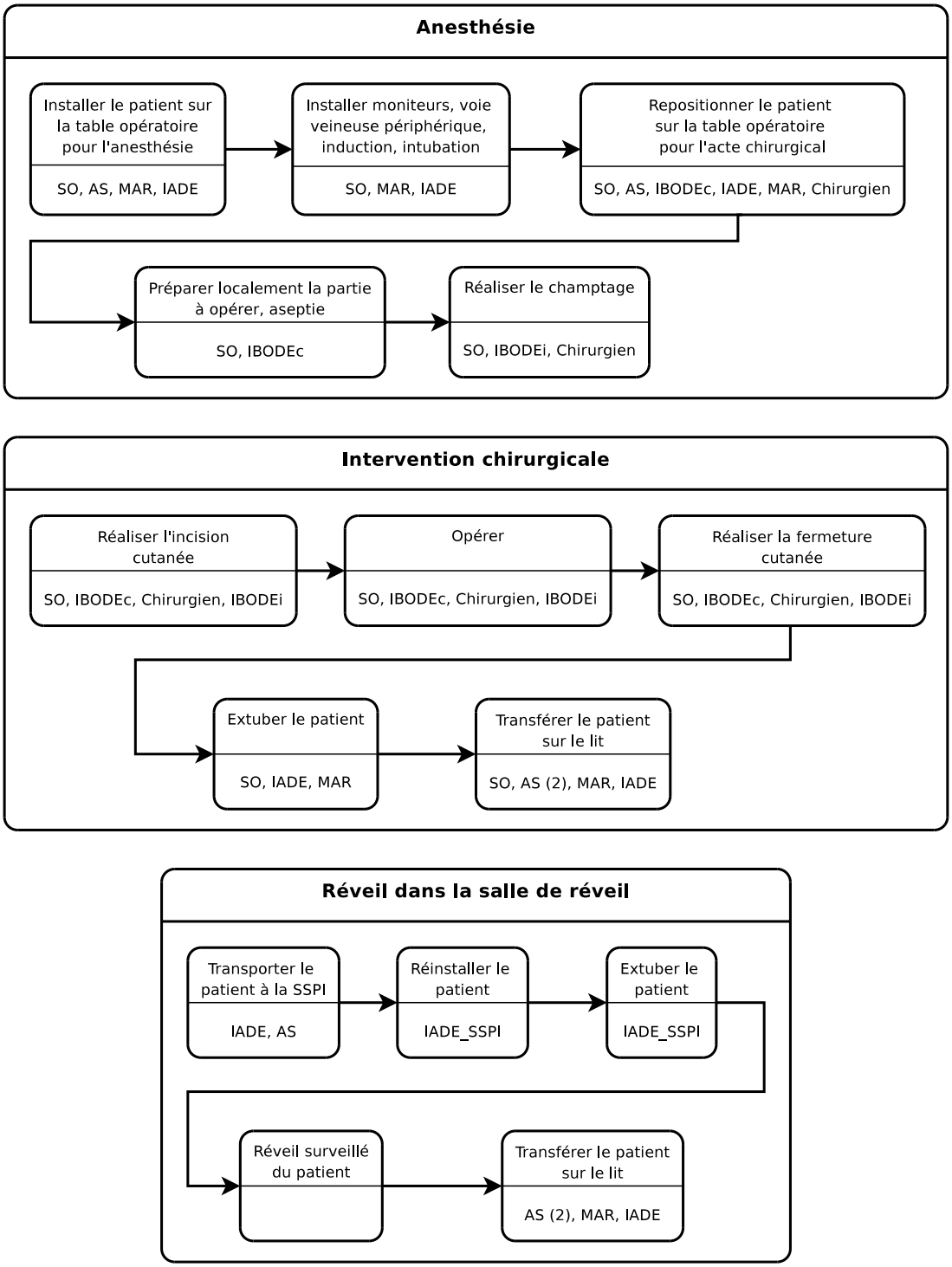


FIG. D.2 – États composites : Anesthésie, Intervention chirurgicale, Réveil dans la SSPI

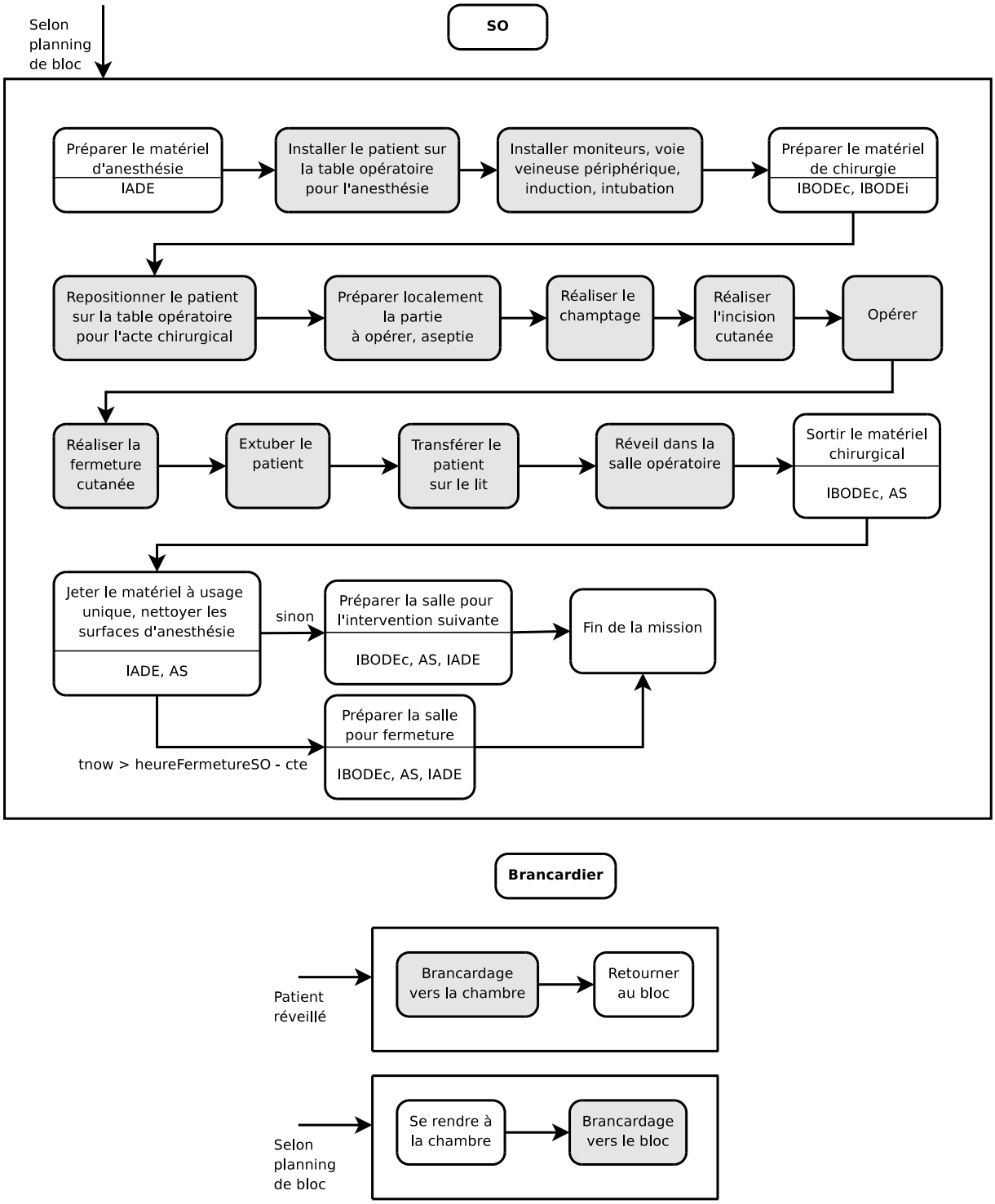


Fig. D.3 – Vue ressource : Salle opératoire, Brancardier

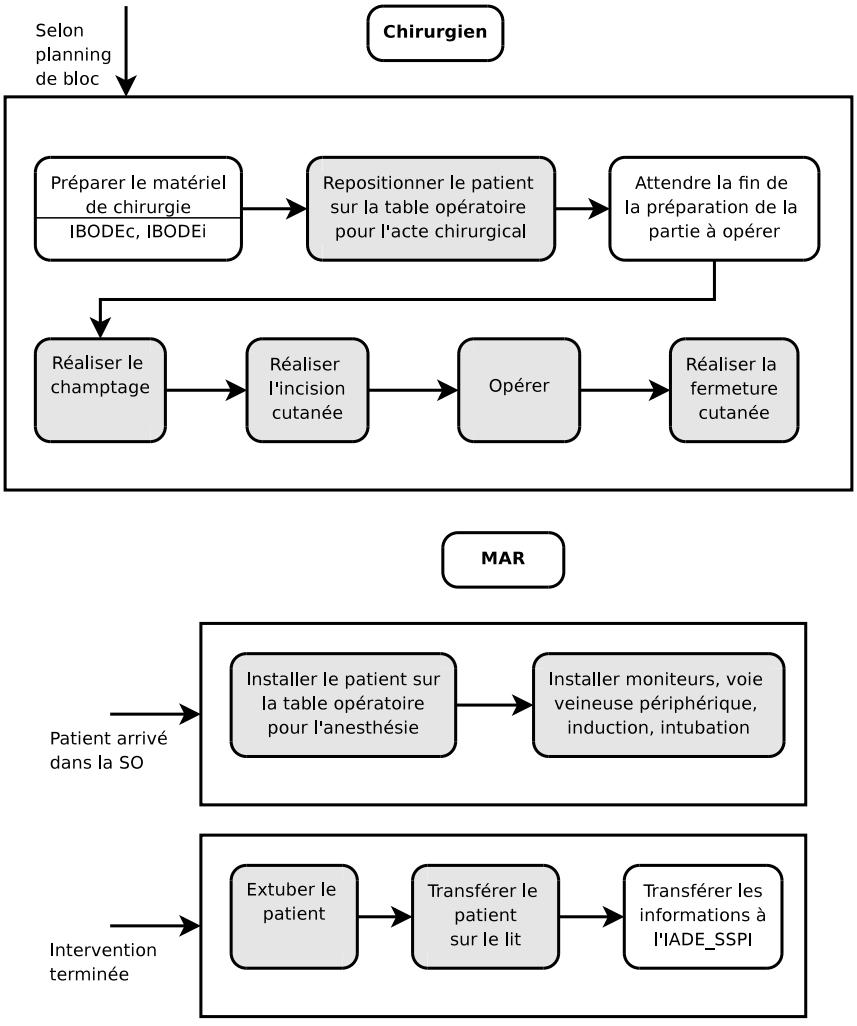


FIG. D.4 – Vue ressource : Chirurgien, MAR

Liste de publications

Publications dans des revues internationales :

- Augusto V. and X. Xie. (200x) Redesigning pharmacy delivery of a health care complex. *Health Care Management Science* (accepté).
- Augusto V., X. Xie and V. Perdomo. (2008) Operating theatre scheduling using Lagrangian relaxation. *European J. Industrial Engineering*, Vol. 2, No. 2, pp.172–189.

Articles soumis ou en cours :

- Augusto V. and X. Xie. (200x) A modelling and simulation framework for health care systems. *European J. Industrial Engineering.*
- Augusto V., X. Xie and V. Perdomo. (200x) Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers and Industrial Engineering.*

Conférences internationales avec actes :

- Lamiri M., V. Augusto, X. Xie. (2008) Patients Scheduling in a Hospital Operating Theatre. *Proceedings of the IEEE Conference on Automation Science and Engineering, Washington DC, USA.*
- Augusto V., X. Xie and F. Grimaud. (2007) A framework for the modeling and simulation of health care systems. *Proceedings of the IEEE Conference on Automation Science and Engineering, Scottsdale, USA.*
- Augusto V., X. Xie and V. Perdomo. (2007) Operating theatre scheduling with limited recovery beds and patient recovery in operating rooms. *Proceedings of the IEEE Conference on Industrial Engineering and System Management, Beijing, China.*
- Perdomo V., V. Augusto, and X. Xie. (2006) Operating theatre scheduling using Lagrangian relaxation. *Proceedings of the IEEE Conference on Service Systems and Service Management, Troyes, France.*

Conférences nationales avec actes :

- Augusto V. et X. Xie. (2008) Une plate-forme de modélisation/simulation pour les systèmes hospitaliers : application à une unité neuro-vasculaire. *Gestion et Ingénierie des SystÈmes Hospitalier (GISEH), 4-6 septembre 2008, Lausanne, Suisse.*
- Augusto V. et X. Xie. (2008) Une plate-forme de modélisation et simulation pour les systèmes hospitaliers. *7e Conférence Francophone de MOdélisation et SIMulation (MOSIM'08), 31 mars-2 avril 2008, Paris, France.*
- Augusto V. et X. Xie. (2006) Modélisation et analyse de flux par la simulation en milieu hospitalier : état de l'art. *Gestion et Ingénierie des SystÈmes Hospitalier (GISEH), 14-16 septembre*

2006, Luxembourg, Grand-Duché de Luxembourg.

Workshops et colloques sans actes :

- Augusto V., X. Xie and F. Grimaud. (2007) A simulation framework for health care systems. *EURO ORAHS'07 (Operational Research Applied to Health Services), Saint-Étienne, France.*
- Augusto V. et X. Xie. (2007) Réorganisation de la pharmacie d'un complexe hospitalier. *ROADEF'07 (Société française de Recherche Opérationnelle et d'Aide à la Décision), Grenoble, France.*
- Augusto V., L. Wang and X. Xie. (2006) Reengineering medication delivery of a healthcare complex using simulation. *EURO ORAHS'06 (Operational Research Applied to Health Services), Wroclaw, Poland.*

Bibliographie

F. Albert et E. Marcon : Computer-aided decision support tool applied in radiology department reengineering. *Dans Proceedings of the IEEE Conference on Industrial Engineering and System Management*, 2005.

C. Alexopoulos, D. Goldsman, J. Fontanesi, M. Sawyer, M. De Guire, D. Kopald et K. Holcomb : A discrete-event simulation application for clinics serving the poors. *Dans Proceedings of the 2001 Winter Simulation Conference*, p. 1386–1391, 2001.

J. G. Anderson : Evaluation in health informatics : computer simulation. *Computers in Biology and Medicine*, 32:151–164, 2002.

R. Ashton, L. Hague, M. Brandreth, D. Worthington et S. Cropper : A simulation-based study of a nhs walk-in centre. *J of the Operational Research Society*, 56:153–161, 2005.

V. Augusto et X. Xie : Modélisation et analyse de flux par la simulation en milieu hospitalier : état de l'art. *Dans Actes de la conférence GISEH*, 2006.

V. Augusto et X. Xie : Redesigning pharmacy delivery processes of a health care complex. *Health Care Management Science*, A paraître, 2008.

V. Augusto, X. Xie et V. Perdomo : Operating theatre scheduling with limited recovery beds and patient recovery in operating rooms. *Dans Proceedings of the IEEE Conference on Industrial Engineering and System Management*, 2007.

V. Augusto, X. Xie et V. Perdomo : Operating theatre scheduling using lagrangian relaxation. *European J. Industrial Engineering*, 2(2):172–189, 2008.

F. F. Baesler, H. E. Jahnsen et M. DaCosta : The use of simulation and design of experiments for estimating maximum capacity in an emergency room. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1903–1906, 2003.

F. F. Baesler et J. A. Sepúlveda : Multi-objective simulation optimization for a cancer treatment center. *Dans Proceedings of the 2001 Winter Simulation Conference*, p. 1405–1411, 2001.

A. Bartelt, W. Lamersdorf, T. O. Paulussen et A. Heinzl : Agent oriented specification for patient-scheduling systems in hospitals. Dagstuhl Article, 2002.

F. Bellifemine, G. Cairea, A. Poggib et G. Rimassa : Jade : A software framework for developing multi-agent applications. lessons learned. *Information and Software Technology*, 50:10–21, 2007.

B. Besombes et L. Merchier : Regroupement des plateaux médico-techniques ; une approche basée sur la modélisation d'entreprise à partir de grai. *Dans Actes de la conférence GISEH*, 2004.

M. Birman et G. Mosheiov : A note on a due-date assignment on a two-machine flow-shop. *Computers and Operations Research*, 31(3):473–480, 2004.

P. Blanc, P. Castagna et I. Demongodin : Pilotage multi-agents d'un système de fabrication de vitres de sécurité. *Dans Actes de MOSIM'06, Rabat (Maroc)*, 2006.

A. Boumane, A. Talbi, C. Tahon et D. Bouami : Identification des compétences requises en milieu hospitalier : application aux médecins. *Dans Actes de la conférence GISEH*, 2006.

M. W. Carter : Simulation modelling in health care : Some examples and lessons learned. Tutorials of the 2007 Operational Research Applied to Health Services, 2007.

M. W. Carter et J. T. Blake : *Using simulation in an acute-care hospital : easier said than done*, chapitre 8, p. 191–215. Kluwer Academic Publishers, 2004.

G. Celano, S. Fichera, M. G. Guglielmino et M. Trefiletti : Activity simulation within a radiology department with limited nurse resources. *Dans INCOM'2006 Preprints*, vol. 3, p. 695–700, 2006.

M. A. Centeno, R. Giachetti, R. Linn et A. M. Ismail : A simulation-ilp based tool for scheduling er staff. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1930–1938, 2003.

M. A. Centeno, M. A. Lee, E. Lopez, H. R. Fernandez, M. Carrillo et T. Ogazon : A simulation study of the labor and delivery rooms at jmh. *Dans Proceedings of the 2001 Winter Simulation Conference*, p. 1392–1400, 2001.

S. Chaabane : *Gestion prédictive des Blocs Opératoires*. Thèse de doctorat, Informatique et Information pour la Société, 2004.

M. Chabrol, P. Féniès, M. Gourgand et N. Tchernev : Un environnement de modélisation pour la supply chain hospitalière : application sur le nouvel hôpital d'estaing. *Ingénierie des Systèmes d'Information*, 11(1):137–162, 2006.

M. Chabrol et D. Sarramia : Object oriented methodology based on uml for urban traffic system modeling. *Dans* Springer-Verlag, éditeur : *Third International Conference on the Unified Modeling Language (UML2000), volume 1939 of LNCS*, 2000.

M. Charfeddine et B. Montreuil : Une approche de mappage stratégique de réseaux de santé intégrés. *Dans Actes de la conférence GISEH*, 2008.

J. Chauvet, P. Féniès et M. Gourgand : A contribution to emergency medical systems modelling and simulation. *Dans Proceedings of the IEEE Conference on Industrial Engineering and System Management*, 2007.

C. Chen, C. Chu et J.-M. Proth : An improvement of the lagrangian relaxation approach for job-shop scheduling : A dynamic programming method. *IEEE Transactions on Robotics and Automation*, 14 (5):786–795, 1998.

J. Desel et G. Gabriel : What is a petri net ? informal answers for the informed reader. *Lecture Notes in Computer Science*, 2128:1–25, 2001.

F. Dexter : Cost implications of various operating room scheduling strategies. *American Society of Anesthesiology - Refresher Courses in Anesthesiology*, 30(1):87–95, 2002.

G. Doumeingts : *Méthode GRAI : Méthode de conception des systèmes en productique*. Thèse de doctorat, Université de Bordeaux I, 1984.

A. Dussauchoy, C. Combes and. F. Gouin et G. Botti : Simulation de l'activité d'un bloc opératoire en utilisant des données recueillies au niveau d'un département d'anesthésie. *Dans Actes de la conférence GISEH*, 2003.

E. El-Darzi, C. Vasilakis, T. Chaussalet et P.H. Millard : A simulation modelling approach to evaluating length of stay, occupancy, emptiness and bed blocking in a hospital geriatric department. *Health Care Management Science*, 1:143–149, 1998.

T. Eldabi et R. J. Paul : A proposed approach for modeling healthcare systems for understanding. *Dans Proceedings of the 2001 Winter Simulation Conference*, 2001.

H. Fei, D. Duvivier, N. Meskens et C. Chu : Ordonnancement journalier dans un bloc opératoire dans le cadre d'une stratégie open scheduling. *Dans Actes de la conférence GISEH*, p. 615–622, 2006a.

H. Fei, N. Meskens et C. Chu : An operating theatre planning and scheduling problem in the case of a "block scheduling" strategy. *Dans Proceedings of the IEEE Conference on Service Systems and Service Management*, p. 422–428, 2006b.

D. M. Ferrin, M. J. Miller, S. Wininger et M. S. Neuendorf : Analysing incentives and scheduling in a major metropolitan hospital operating room through simulation. *Dans Proceedings of the 2004 Winter Simulation Conference*, p. 1975–1980, 2004.

D. Fone, S. Hollinghurst, M. Temple, A. Round, N. Lester, A. Weightman, K. Roberts, E. Coyle, G. Bevan et S. Palmer : Systematic review of the use and value of computer simulation modelling in population health and health care delivery. *Journal of Public Health Medicine*, 25 (4):325–335, 2003.

B. Glaa, S. Hammadi et C. Tahon : Modeling the emergency path handling and emergency department simulation. *Dans Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, p. 4585–4590, 2006a.

B. Glaa, C. Tahon, S. Hammadi et A. O. Daem : Gestion optimisée des infirmiers/infirmières diplômés d'état dans le service des urgences du centre hospitalier de valenciennes. *Dans Actes de la conférence GISEH*, p. 121–130, 2006b.

T. Gordon, S. Paul, A. Lyses et J. Fountain : Surgical unit time utilization review : resource utilization and management implications. *J. of Med. Sys.*, 12(3):169–179, 1988.

M. Gourgand et P. Kellert : Conception d'un environnement de modélisation de systèmes de production. *Dans Actes du troisième congrès international de génie industriel, Tours, France*, p. 191–203, 1991.

J. D. Griffiths, N. Price-Lloyd, M. Smithies et J. E. Williams : Modelling the requirement for supplementary nurses in an intensive care unit. *Journal of the Operational Research Society*, 56:126–133, 2005.

A. Guinet et S. Chaabane : Operating theatre planning. *International Journal of Production Economics*, 85:69–81, 2003.

H. El Haouzi et A. Thomas : A methodological approach to build simulation models of manufacturing systems with distributed control. *Dans Proceedings of the IEEE Conference on Industrial Engineering and System Management, Marrakech, Morocco*, 2005.

C. R. Harrell et V. Lange : Healthcare simulation modeling and optimization using medmodel. *Dans Proceedings of the 2001 Winter Simulation Conference*, p. 233–238, 2001.

D. L. Heflin et C. R. Harrell : Healthcare simulation modeling and optimization using medmodel. *Dans Proceedings of the 1998 Winter Simulation Conference*, p. 185–189, 1998.

D. J. Hoitomt, P. B. Luh et K. R. Pattipati : A practical approach to job shop scheduling problems. *IEEE Transactions on Robotics and Automation*, 9(1):1–13, 1993.

A. Jebali : *Vers un outil d'aide à la planification et à l'ordonnancement des ressources dans les services de soins.* Thèse de doctorat, Laboratoire d'Automatique de Grenoble (LAG) - Ecole Doctorale Organisation Industrielle et Systèmes de Production, 2004.

A. Jebali, A. B. Hadj Alouane et P. Ladet : Operating rooms scheduling. *Int. J. Production Economics*, 99:52–62, 2006.

M. D. Jeng et F. DiCesare : Synthesis using resource control nets for modeling shared-resource systems. *IEEE Trans. Robot. Automat.*, 11:317–327, 6 1995.

S. M. Johnson : Optimal two- and three-stage production schedules with setup times included. *Naval Res Log Quart*, 1:61–68, 1954.

J. B. Jun, S. H. Jacobson et J. R. Swisher : Application of discrete-event simulation in health care clinics : A survey. *Journal of the Operational Research Society*, 50(2):109–123, 1999.

E. P. C. Kao et G. G. Tung : Bed allocation in a public health care delivery system. *Management Science*, 27(5):507–520, 1981.

P. Kellert et S. Ruch : Méthodologie de modélisation orientée objets de systèmes de production - un processus de construction/validation du modèle générique orienté objets d'un système de production. *Journal Européen des Systèmes Automatisés*, 32(1):51–105, 1998.

S. Kharraja, P. Albert et S. Chaabane : Block scheduling : Toward a master surgical schedule. *Dans Proceedings of the IEEE Conference on Service Systems and Service Management*, p. 429–435, 2006.

C. H. Kil, R. H. Weston, A. Hodgsin et K. H. Lee : The complementary use of idef and uml modelling approaches. *Computers in Industry*, 50:35–56, 2003.

S.-C. Kim, I. Horowitz, K. K. Young et T. A. Buckley : Flexible bed allocation and performance in the intensive care unit. *Journal of Operations Management*, 18:427–443, 2000.

S. Kirn, C. Heine, R. Herrler et K.-H. Krempels : Agent.hospital - framework for clinical applications in agentcities. *Dans Proceedings of the International Workshop on Enabling Technologies : Infrastructure for Collaborative Enterprises*, p. 67–85, 2003.

T. Klein et A. Thomas : Dévoloppement d'un modèle de simulation pour l'évaluation des systèmes de pilotage distribués. *Dans Actes de MOSIM'06, Rabat (Maroc)*, 2006.

A. Koestler : *The Ghost in the Machine.* Artana Books, London, 1989.

M. Lamiri, X. Xie, A. Dolgui et F. Grimaud : A stochastic model for operating room planning with elective and emergency demand for surgery. *European J of Operational Research*, accepted, in press, 2006.

M. Lamiri, X. Xie et S. Zhang : Column generation for operating theatre planning with elective and emergency patients. *IIE Transactions*, accepted, in press, 2007.

A. M. Law et W. D. Kelton : *Simulation Modeling and Analysis*. McGraw-Hill, 2004a.

A. M. Law et W. D. Kelton : *Simulation Modeling and Analysis*, chapitre 8. McGraw-Hill, 2004b.

B. M. T. Lin : Scheduling in the two-machine flowshop with due date constraints. *International Journal of Production Economics*, 70(2):117–123, 2001.

D. G. Luenberger : *Linear and non linear programming (second edition)*. Addison-Wesley, Reading, 1984.

P. B. Luh et D. Hoitmt : Scheduling of manufacturing systems using lagrangian relaxation technique. *IEEE Transactions on Robotics and Automation*, 9(7):1066–1079, 1993.

E. Marcon, S. Kharraja, N. Smolski, B. Luquet et J.-P. Viale : Determining the number of beds in the postanesthesia care unit : A computer simulation flow approach. *Anesth Analg*, 96:1415–1423, 2003.

A. H. Marshall, S. I. McClean et P. H. Millard : Addressing bed costs for the elderly : A new methodology for modelling patient outcomes and length of stay. *Health Care Management Science*, 7:27–33, 2004.

P. H. Millard, M. Mackay, C. Vasilakis et G. Christodoulou : Measuring and modelling surgical bed usage. *Annals of the Royal College of Surgeons of England*, 82:75–82, 2000.

M. J. Miller, D. M. Ferrin et M. G. Messer : Fixing the emergency department : a transformation journey with edsim. *Dans Proceedings of the 2004 Winter Simulation Conference*, p. 1988–1993, 2004.

M. J. Miller, D. M. Ferrin et J. M. Szymanski : Simulating six sigma improvement ideas for a hospital emergency department. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1926–1929, 2003.

B. Montreuil, M. Charfeddine, O. Labarthe et A. Côté : A generic agent-based framework for simulation in distributed healthcare systems. *Dans Proceedings of the IEEE Conference on Industrial Engineering and Systems Management*, 2007.

L. Moreno, R.M. Aguilar, C.A. Martín, J.D. Pineiro, J.I. Estévez, J.F. Sigut, J.L. Sánchez et V.I. Jiménez : Patient-centered simulation tool for aiding in hospital management. *Simulation Practice and Theory*, 7:373–393, 1999.

B. P. Morrison et B. C. Bird : A methodology for modeling front office and patient care processes in ambulatory health care. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1882–1886, 2003.

V. Z. Osidach et M. C. Fu : Computer simulation of a mobile examination center. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1868–1875, 2003.

T. O. Paulussen, N. R. Jennings, K. S. Decker et A. Heinzl : Distributed patient scheduling in hospitals. *Dans International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, 2003.

C. A. Petri : *Kommunikation mit Automaten*. Thèse de doctorat, University of Bonn, 1962.

D. Pilone et N. Pitman : *UML 2 en concentré*. Éditions O'Reilly, Paris, 2006.

M. Pinedo : *Scheduling theory, algorithms and systems*. Prentice Hall, 1995.

M. Pitt : A generalised simulation system to support strategic resource planning in healthcare. *Dans Proceedings of the 1997 Winter Simulation Conference*, 1997.

B. T. Polyak : Minimization of unsmooth functionals. *USSR Comput. Math. and Math. Phys.*, 9:14–29, 1969.

J.-M. Proth, L. Wang et X. Xie : A class of petri nets for manufacturing system integration. *IEEE Transactions on Robotics and Automation*, 13(3):317–326, 1997.

J.-M. Proth et X. Xie : *Les réseaux de Petri pour la conception et la gestion des systèmes de production.* Masson, 1995a.

J.-M. Proth et X. Xie : *Les réseaux de Petri pour la conception et la gestion des systèmes de production*, chapitre 6. Masson, 1995b.

F. J. Ramis, J. L. Palma et F. F. Baesler : The use of simulation for process improvement at an ambulatory surgery center. *Dans Proceedings of the 2001 Winter Simulation Conference*, 2001.

A. Ramudhin, E. Chan et A. Mokadem : A framework for the modelling, analysis and optimization of pathways in healthcare. *Dans Proceedings of the IEEE Conference on Service Systems and Service Management*, p. 698–702, 2006.

M. Roboam : *La Méthode GRAI, Principes, Outils, Démarche et Pratique.* Teknéa, 1993.

D. T. Ross : Structured analysis (sa) : a language for communicating ideas. *IEEE Transactions on Software Engineering (TSE)*, 3, 1977.

D. T. Ross : Applications and extensions of sadt. *IEEE Computer*, 18:25–34, 1985.

O. Roux, C. Combes et D. Duvivier : A modeling methodology dedicated to simulation and based on genric meta-models using mda-uml : an application to a chronic renal dialysis unit. *Dans Proceedings of the IEEE Conference on Service Systems and Service Management*, p. 692–697, 2006.

S. Samaha, W. S. Armel et D. W. Starks : The use of simulation to reduce the length of stay in an emergency department. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1907–1911, 2003.

R. Sampath, H. Darabi, J. Lin et B. Galanter : Modeling and integration of hospital information systems by petri nets. *IEEE Transactions on Systems, Man and Cybernetics*, A paraître, 2006.

S. M. Sanchez, D. M. Ferrin, T. Ogazon, J. A. Sepúlveda et T. J. Ward : Emerging issues in healthcare simulation. *Dans Proceedings of the 2000 Winter Simulation Conference*, p. 1999–2003, 2000.

D. Sinreich et Y. N. Marmor : A simple and intuitive simulation tool for analyzing emergency department operations. *Dans Proceedings of the 2004 Winter Simulation Conference*, p. 1994–2002, 2004.

C. W. Spry et M. A. Lawley : Evaluating hospital pharmacy staffing and work scheduling using simulation. *Dans Proceedings of the 2005 Winter Simulation Conference*, p. 2256–2263, 2005.

P. Staccini, M. Joubert, J. M. Quaranta, D. Fieschi et M. Fieschi : Modeling health care processes for eliciting user requirements : a way to link quality paradigm and clinical information system design. *International Journal of Medical Informatics*, 64:129–142, 2001.

M. D. Stevenson, P. A. Oakley, S. M. Beard, A. Brennan et A. L. Cook : Triaging patients with serious head injury : results of a simulation evaluating strategies to bypass hospitals without neurosurgical facilities. *Injury*, 32:267–274, 2001.

S. Su et C.-L. Shih : Modeling an emergency medical services system using computer simulation. *International Journal of Medical Informatics*, 72:57–72, 2003.

S. Takakuwa et H. Shiozaki : Functional analysis for operating emergency department of a general hospital. *Dans Proceedings of the 2004 Winter Simulation Conference*, p. 2003–2011, 2004.

B. A. Tan, A. Gubaras et N. Phojanamongkolkij : Simulation study of dreyer urgent care facility. *Dans Proceedings of the 2002 Winter Simulation Conference*, 2002.

L. Trilling, B. Besombes, S. Chaabane et A. Guinet : Investigation et comparaison des méthodes et outils d'analyse pour l'étude des systèmes hospitaliers. Rapport technique, Rapport de recherche sur le projet HRP2, 2004.

D. C. Tyler, C. A. Pasquariello et C.-H. Chen : Determining optimum operating room utilization. *Anesth Analg*, 96:1114–1121, 2003.

P. Valckenaers, H. Van Brussel, L. Bongaerts et J. Wyns : Holonic manufacturing systems. *Integrated Computers Aided Engineering*, 4:191–201, 1997.

C. Vasilakis et L. Kuramoto : Comparing two methods of scheduling outpatient clinic appointments using simulation experiments. *Clinical & Investigative Medicine*, 28:368–370, 2005.

F. Vernadat : *Techniques de Modélisation en Entreprise : Application aux Processus Opérationnels*. Economica, 1999.

T. Vollman, W. Berry et D. Whybark : *Manufacturing Planning and Control Systems*. Irwin, 1988.

L. von Bertalanffy : *General System Theory : Foundations, Development, Applications*. George Braziller, New York, 1968.

J. Wang, P. B. Luh et X. Zhao : An optimization-based algorithm for job shop scheduling. *SADHANA, in Journal of Indian Acacdemy of sciences, Special Issue on Competitive Manufacturing Systems*, 22: 241–256, 1997.

A. Wiinamaki et R. Dronzek : Using simulation in the architectural concept phase of an emergency department design. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1912–1916, 2003.

A. Wijewickrama et S. Takakuwa : Simulation analysis of appointment scheduling in an outpatient department of internal medicine. *Dans Proceedings of the 2005 Winter Simulation Conference*, p. 2264–2273, 2005.

C. Wong, G. Geiger, Y. D. Derman, C. R. Busby et M. W. Carter : Redesigning the medication ordering, dispensing, and administration process in an acute care academic health sciences center. *Dans Proceedings of the 2003 Winter Simulation Conference*, p. 1894–1902, 2003.

I. H. Wright, C. Kooperberg, B. A. Bonar et G. Bashein : Statistical modeling to predict elective surgery time. *Anesthesiology*, 85:1235–1245, 1996.

X. Xie et M. D. Jeng : Ercn-merged nets and their analysis using siphons. *IEEE Trans. Robot. Automat.*, 15(4):692–703, 8 1999.

École Nationale Supérieure des Mines
de Saint-Étienne

N° d'ordre : 496 GI

Vincent AUGUSTO

Modelling, simulation and control of health care flows using UML and Petri nets

Speciality : Industrial Engineering

Keywords : modelling, simulation, control, UML, nets, Petri, framework, planning, scheduling, rapid prototyping

Abstract :

Traditional business process modelling approaches and operational research methods have been used to model and understand complex health care systems. However, special features of these systems cannot be captured by traditional modelling tools. Our goal is to develop a methodology which is able to capture particularities of health care system. We introduce the medPRO (medical Process-Resource-Organisation) modelling and simulation framework in this thesis, designed for analysis and simulation of health care systems. This framework consists of a list of generic resources and activities description, and a handbook describing the base system. UML (Unified Modelling Language) has been chosen to model these systems. Several points of view have been selected : Process (patient centred view), Resource (behaviour of human and material resources) and Organisation (relation between actors). Dynamic behaviour of the model is specified using a special class of Petri nets, called Health Care nets : a discrete-event simulation algorithm has also been developped for Petri nets. A large part of this work is dedicated to the decision system, which is used (i) to apply planning and scheduling methods from industrial engineering to health care systems, and (ii) to control the simulation process in real-time using a hierarchical/heterarchical hybrid approach.

Three case studies are also presented to show the efficiency of the medPRO framework : we investigated the neuro-vascular unit, the pharmacy unit and the operating theatre of the universitary teaching hospital of Saint-Étienne (France). Special optimization tools were developped and included in the framework.

École Nationale Supérieure des Mines
de Saint-Étienne

N° d'ordre : 496 GI

Vincent AUGUSTO

Modélisation, simulation et pilotage de flux en milieu hospitalier à l'aide d'UML et des réseaux de Petri

Spécialité : Génie Industriel

Mots-clefs : modélisation, simulation, pilotage, UML, réseaux, Petri, plate-forme, planification, ordonnancement, prototypage rapide

Résumé :

La modélisation et l'analyse de systèmes hospitaliers sont traditionnellement réalisées en utilisant méthodes et outils issus du génie industriel. Cependant, les caractéristiques de ces systèmes sont difficiles à capturer avec les outils de modélisation et de simulation classiques. Notre objectif est de spécifier et de développer une plate-forme de modélisation et de simulation dédiée aux systèmes hospitaliers, appelée medPRO (medical Process-Resource-Organisation), accompagnée d'une méthodologie d'analyse adaptée au domaine médical. Cette plate-forme est construite autour d'un cadre de modélisation et d'un guide méthodologique conçus pour un système hospitalier particulier. UML (Unified Modelling Language) a été choisi pour la modélisation de ces systèmes. Plusieurs points de vue sont proposés : Processus (vue centrée sur le patient), Ressource (comportement des ressources humaines et matérielles), et Organisation (relation entre les intervenants). Le comportement dynamique du modèle est spécifié grâce à une classe dédiée de réseaux de Petri, appelés réseaux de Petri de Santé : un algorithme de simulation à événements discrets a également été développé pour les réseaux de Petri. Une large partie de ce travail est dédiée au système de décision, qui est utilisé (i) pour appliquer des méthodes de planification et d'ordonnancement issues du génie industriel à des systèmes hospitaliers, et (ii) pour piloter le déroulement de la simulation en temps réel au travers d'une approche hybride hiérarchique/hétérarchique.

Trois études de cas sont également présentées pour montrer l'efficacité de la plate-forme medPRO : nous nous sommes intéressés à l'unité neuro-vasculaire, à la pharmacie et au bloc opératoire du CHU de Saint-Étienne (France). Plusieurs outils d'optimisation spécifiques ont été développés et inclus dans la plate-forme.