A modelling and simulation framework for health care systems.

Vincent Augusto, Xiaolan Xie

A Modeling and Simulation Framework for Health Care Systems

Vincent Augusto, Xiaolan Xie, *Senior Member IEEE*

***Abstract*—In this paper, we propose a new modeling methodology named MedPRO for addressing organization problems of health care systems. It is based on a meta-model with three different views: process view (care pathways of patients), resource view (activities of relevant resources), and organization view (dependencies and organization of resources). The resulting meta-model can be instantiated for a specific health care system and be converted into executable model for simulation by means of a special class of Petri nets, called Health Care Petri Nets (HCPN). HCPN models also serve as a basis for short-term planning and scheduling of health care activities. As a result, the MedPRO methodology leads to a fast-prototyping tool for easy and rigorous modeling and simulation of health care systems. A case study is presented to show the benefits of the MedPRO methodology.**

***Index Terms*—health care, Petri nets, modeling, simulation, rapid prototyping.**

I. Introduction

A. Motivation

MANAGEMENT of health care systems has significantly evolved during the last decade. A modern health care system can be viewed as a complex network of medical units with a large diversity of health care professionals (both clinical and administrative), medical equipment and complex material flows connected through various information systems [17]. Appropriate coordination of these entities is needed to provide (i) better access, continuity and quality of care to patients, (ii) better working conditions for health care professionals, while (iii) compressing operating costs and investments in new technology and infrastructures.

Health care organizations such as hospitals are often compared to manufacturing systems and techniques from industry are adapted to solve various organization problems such as operating theater scheduling, patient transportation planning, logistic organization, etc. However the health care field presents several important special features such as complex patient flows, numerous human resources, dynamic evolution of patient's health state, coordination of separate medical units, etc. This implies a wide range of complex decision processes and behavior models. Thus most existing studies [1], [11] rely on case-specific models although the goals are often similar.

Xiaolan Xie is with the Center for Health and Engineering and CNRS UMR 6158 LIMOS-ROGI team, Ecole Nationale Supérieure des Mines, 158 cours Fauriel, 42023 Saint Etienne, France. E-mail: xie@emse.fr.

Xiaolan Xie is also with the Dept. Industrial Engr. & Logistics Management, Shanghai Jiao Tong University, Shanghai, China.

Vincent Augusto is with the Center for Health and Engineering and CNRS UMR 6158 LIMOS-ROGI team, Ecole Nationale Supérieure des Mines, 158 cours Fauriel, 42023 Saint Etienne, France. E-mail: augusto@emse.fr.

Manuscript received xxx, 2011; revised xxx, 2011.

This research is part of a project called MedPRO which aims at developing a Petri net based software for modeling, simulation and activity planning and scheduling of health care services. The goal of this study is to provide a mathematical framework to design models of relevant medical units of a hospital such as emergency department, operating theater, hospital pharmacy, etc. A new class of Petri nets is introduced for this purpose in order to model and simulate a wide range of health care services and organizations.

In order to effectively support decision makers of health care systems, we propose a new modeling methodology combining Petri net models and a light business process modeling (BPM) approach. The BPM approach makes use of the Unified Modeling Language (UML) to propose a simple interface between health care professionals and engineers enabling communication and better understanding of modeled organizations. The BPM approach allows easily modeling of relationship between different entities. It is also useful to design information systems, manage resource requirements and propose key performance indicators for continuous improvement.

B. Literature review

There are three types of relevant approaches in the literature for modeling complex systems. The first one includes some high-level enterprise modeling architectures and related business process modeling tools. The goal and the stake of the study is a must-have preliminary step allowing the definition of the model. Depending on the stake-holders the resulting model will be different in order to be understood by every party. Limitations of these tools have been identified in [23] which pointed out that human behavior is often not taken into account and control systems are rarely included. Transition from such high-level enterprise modeling to simulation is not obvious. Some of these tools have been applied to health care systems: SADT (Structured Analysis and Design Technic) [3], GRAI (Graph with Results and Interlinked Activities) [2], ARIS (Architecture of Integrated Information Systems) [20], [24] and UML [12], [19], [22].

An approach of the second type aims at designing a generic model for health care systems. An early attempt was proposed in [15]: a generic simulation tool is designed to: (i) summarize information about system state through graphs and reports, and (ii) test different control actions for decision aid. A modeling and simulation methodology with generic MDA-UML (Model-Driven Architecture-Unified Modeling Language) meta-models was presented in [18] where simulation models are instantiated from a generic framework using statecharts. Generic knowledge models were proposed in [4], allowing the modeling and simulation of several medical units as the emergency unit. Knowledge is structured using the ASCI methodology (Analysis, Specification, Conception, Implantation) [9].

Approaches of the third type propose modeling and simulation frameworks, featuring a large amount of tools for the modeling of health care systems. The medBPM framework [17] was designed to capture the complexity of the processes in health care systems. Dependencies and synchronization between people, information and material are taken into account. This framework is resource-oriented with a user-friendly formalism to model health care flows. A multi-agent simulation framework dedicated to health care systems was proposed in [14]. The framework is built on four key elements including agents, objects, environment and experience, and intends to allow a faster modeling process while minimizing frequent errors and element or interaction omissions. Finally, an analytic framework for modeling and analysis of hospital emergency departments (ED) was presented in [13] for addressing problems of performance analysis, bottleneck mitigation and workforce/resource allocation. The resulting framework is supposed to be adapted to any ED and support ED managers with efficient methods to reduce overcrowding in such department. To the best of our knowledge, frameworks presented in [17] and [14] have not been sufficiently experimented on various health care systems. Moreover, the multi-agent architecture requires a large amount of tests for validation, as operation processes are not formally defined.

Apart from these three types of approaches, various special purpose modeling and simulation frameworks can be found in the literature. A computational modeling program was proposed in [7] to model patient care units' achievement in terms of patient safety and quality outcomes. A simulation model was described in [16] to improve the quality of care of acute medical patients and to provide decision aid tools to avoid unnecessary admissions. Petri nets were used in several studies. An generic modeling approach of alarm management work flow in health care was proposed in [8] by using UML for communication and colored timed Petri nets for simulation. The framework was shown to have high potential capability for describing large and complex health care systems. A continuous Petri net framework was proposed in [5] to describe in a concise and effective way the structure and dynamics of a critical emergency cardiology department. Optimization problems were defined using the proposed model as well as a simulation study. UML activity diagrams and Petri nets were used in [6] to propose a better management of hospital departments and a case study on a pulmonary department was proposed. The authors claimed that their base model can be used to design and size any hospital department.

C. Scientific contribution

The main contribution of this paper is to propose a comprehensive and flexible type 3 approach for modeling and analyzing health care systems. It relies on a semi-automatic methodology allowing the rapid prototyping of these systems by appropriate combination of UML, Petri nets, discrete-event simulation and linear programming. It starts with a UML-based modeling framework to allow the representation of several complementary views of a system with different abstraction levels. UML is used to better communicate with health care professionals about the features of the system. Contrary to most frameworks proposed in the literature, the UML models are then automatically converted into formal colored Petri net models. The Petri net models are executable and are then used for simulation of the related health care systems. The relation between UML models and Petri net models allows easy visualization of the execution of the simulation model. Another important feature of the MedPRO approach is the automatic generation of optimization models for short term planning and scheduling of health care activities. Finally, the proposed framework intends to be adapted to any department of a hospital and a real case study of a hospital pharmacy will be presented to show its applicability.

Given the above features of our MedPRO approach, the following comparisons with respect to existing literature can be summarized as follows. With respect to high-level type 1 enterprise modeling architectures, our MedPRO framework has the following salient features: (i) efficient and user-friendly multi-view UML modeling better suited for health care applications than industry-oriented frameworks such as SADT, GRAI or ARIS; (ii) explicit modeling of human behaviors by defining special relations such as abilities, teams and replacement (as described in Section III-D); (iii) automatic generation of simulation models from UML representations, a feature badly missing in most commercial packages on enterprise modeling.

With respect to type 2 modeling approaches which intend to offer generic modeling tools, the MedPRO framework proposed in this paper is not a "ready-to-use" generic modeling and simulation tool. Instead of over complex generic models needing a tedious setup phase, the MedPRO proposes a "toolbox" for flexible modeling of any health care system model by describing different complementary views in UML. No high technical knowledge of modeling, simulation and optimization is needed.

With respect to type 3 approaches which are not based on formal modeling tools, our MedPRO approach makes use of formal Petri net models to convert UML representation into executable models and hence allows automatic generation of correct simulation models and planning and scheduling models.

Compared with other special purpose modeling and simulation framework, our MedPRO approach is closely related to the methodology proposed in [6] that also combined UML and Petri nets for modeling and simulation of health care systems. Special features of our MedPRO approach includes: (i) a comprehensive multi-view UML modeling that takes into account complex human aspects of health care systems, (ii) automatic conversion of the UML models into formal and executable Petri net models, and (iii) automatic generation of planning and scheduling models of health care activities. Further, our MedPRO approach has been tested on various health care systems such as the operating theater, the pharmacy

(Section V) and a short stay medical unit.

The remaining part of this paper is organized as follows. The MedPRO methodology is presented in Section II. The related UML modeling framework is described in Section III. In Section IV, we introduce a new class of Petri nets for health care systems as well as a conversion algorithm. Planning and scheduling techniques are also presented in this section. A case study is presented in Section V. Conclusions and perspectives are given in Section VI.

II. Methodology

This section outlines the MedPRO methodology followed in this paper for health care systems. Figure 1 illustrates the main steps of the MedPRO approach for the modeling and analysis of a health care system. For a given organizational problem, rapid prototyping is used to provide effective solutions quickly. After a preliminary domain analysis (step 1) of the health care system under consideration, a meta-model is built using a specific UML toolbox (step 2). The model is then validated with health care professionals (step 3) and eventually modified to fit the reality. Then an automatic conversion into a formal Petri net model is done (step 4). Using the properties of this formal model, planning and scheduling optimization can be performed (step 5) as well as discrete-event simulation (step 6). Methodological guidelines are also available to better understand special features of the modeled system. Discrete-event simulation is used to observe the behavior of the model, test new organization scenarios, extract and present results.

The underlying concepts of the proposed framework have been established using extensive literature reviews and field experience in three different health care systems (operating theater, hospital pharmacy, neuro-vascular department). Key features of the MedPRO approach are summarized as follows:

- A flexible multi-view high-level modeling formalism (BPM approach via UML). UML has been chosen in order to propose a simple and yet formal model for health care professionals. Multiple modeling views are used for ease of modeling and understanding. Relevant information (processing times, resource organization, arrival rates) are integrated in the model through an internal information system.
- Automated conversion of UML models into rigorous Petri net models. Taking advantage of the literature, Petri nets are used to: (i) analyze the model and identify bottlenecks and inaccuracies, (ii) provide optimal planning, and (iii) run relevant simulation scenarios.
- Integration of special features of health care systems such as specific decision processes, medical diagnosis models, competencies of human resources and team work.

To summarize, the MedPRO approach uses UML models for description and discussion purpose and rigorous Petri nets models for formal analysis and simulation purposes. Although using both models requires conversion procedures, the resulting framework remains light and covers the whole analysis process from specification to planning/scheduling and simulation. The following sections of this paper detail both BPM and Petri net models. A case study is also proposed to illustrate the performance of the method.

III. Operational system modeling

The MedPRO approach models a health care system from three views: process view (patient flow), resource view (health care activities of resources) and organization view (competencies, team formations, relations between resources). UML statechart diagrams are used to model various process flows. A statechart diagram (or statechart machine) depicts the dynamic behavior of an entity based on its response to events, showing how the entity reacts to various events depending on the current state that it is in. Statechart machines are used to explore the complex behavior of a system.

UML statechart diagrams are directed graphs where nodes denote states and connectors denote state transitions. In UML, states are represented as rounded rectangles with a state label. The transitions, represented as arrows, are labeled with the triggering events followed optionally by the list of executed actions. The initial transition originates from a solid circle and specifies the default state of the system. Every state diagram should have such a transition, which is labeled since it is not triggered by any event. The initial transition can have associated actions.

A. Preliminaries

UML models of this paper are built using a sub-class of statecharts defined as follows:

Definition 3.1 (Statechart): A statechart (SC) is a 3-tuple $M = (S, V, \zeta)$ where $S = \{s_1, s_2, \dots, s_n\}$ is a finite set of states, $V \subseteq (S \times S)$ is a finite set of transitions, and $\zeta : V \to \mathbb{B}$ is a the guard condition of a transition, where $\mathbb{B}$ is a set of eligible boolean conditions.

Statecharts are used to model flow of entities and behavior of resources. In the latter case, a statechart is called *mission*; a set of missions define the skills of a resource.

Definition 3.2 (Mission): A mission is a 4-tuple $mi = (M, s_{in}, s_{out}, mp)$ as M is a statechart, s_{in} (resp. s_{out}) is the input (resp. output) state of the mission, and $mp \in \mathbb{N}^*$ is the number of resource units needed to perform the mission.

Entities and resources are defined using UML classes. A resource may be a physician, an operating room, a medical team or a competency.

Definition 3.3 (Entity): An entity is a singleton $u = (A)$ where A is a list of attributes.

Definition 3.4 (Resource): A resource is a 3-tuple $r = (A, \mathcal{P}, \mathcal{M})$ where A is a list of attributes, $\mathcal{P}$ is a list of availability periods, and $\mathcal{M}$ is a set of missions. An availability period is a 3-tuple (t_1, t_2, n) where $t_1 \in \mathbb{N}$ is the starting time, $t_2 \in \mathbb{N}$ is the completion time and $n \in \mathbb{N}^*$ is the number of available resource units.

A set of variables are also defined in the framework in order to characterize the system. Variables are common to all objects of the model.

B. Process view

The process view is a set of flow diagrams modeling the care pathway of entities (usually patients) through a medical unit. The goal of the process view is double: (i) offer a precise

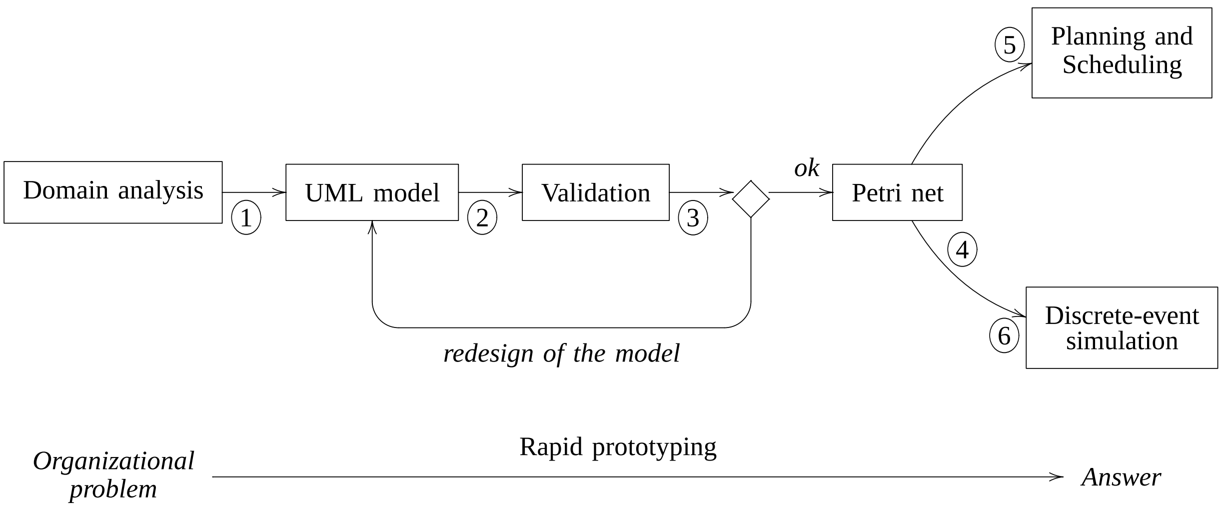


Fig. 1. Global methodology

representation of relevant care activities and care processes of each patient; (ii) propose a patient-centered model designed for communication and/or medical education.

A state of the process view (P-state) models an operation performed by/on the entity considered in the statechart. Resources are eventually needed. A state can model a static situation (‘wait for the porters’) or a dynamic one (‘take a meal’). In the latter case, the state has a *processing time*.

Definition 3.5 (P-state): A P-state is a 3-tuple $p^s = (l, \tau, R)$ where $l \in \mathbb{L}$ is a unique label, τ is the processing time, and $R = \{(r_1, mp_1), \ldots, (r_n, mp_n)\}$ is a list of resources types and the number of resource units needed.

Several types of resource selections can be defined for a P-state: (i) by name to choose a precise resource (‘Dr. Smith’); (ii) by competency to select a qualification shared by several resources (‘someone who knows how to install a blood transfusion’); (iii) by team to select a set of resources (‘surgery team S, made of surgeon A, nurses D, F and anesthesiologist X’). The selection process is undertaken by the control subsystem in charge of the planning and scheduling.

Entities are created at entry nodes and deleted at exit nodes. Let S be the set of states of a statechart.

Definition 3.6 (Entry node): The entry node of a statechart (S, V, ζ) is a couple $s_{in} = (u, G)$ such that $s_{in} \in S$, $s_{in} \neq w$ $\forall (v, w) \in V$. u is the type of entities created in the node and $G : \mathbb{N} \to \mathbb{N}$ with $G(x) \geq x$ is the generation scheme that generates the next entity creation time.

Let u_i be the i-th entity generated at time x_i. Three generation schemes are defined in this paper:

- Scheduled arrivals with given arrival dates x_i.
- Constant: $x_{i+1} = x_i + \delta$ where δ is a constant time interval.
- Random: $x_{i+1} = x_i + U_i$ where U_i are independent and identically distributed random numbers according to some common probability distribution.

Example 3.7 (P-statechart example): Figure 2 is a process view model for surgery operations. The task called `surgery` requires an `operating_room`, a `surgeon` and two `nurses`, and lasts 50 minutes. This state is formally defined by a 3-tuple (l, τ, R), where $l =$’`surgery`’, $\tau = 3000\ s$ and $R = (\{(r_1, 1), (r_2, 1), (r_3, 2)\})$. r_1, r_2 and r_3 correspond to `operating_room`, `surgeon` and `nurse` respectively.

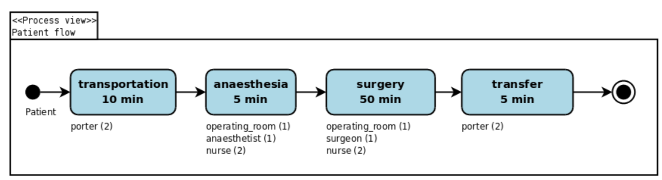


Fig. 2. A statechart in Process view

C. Resource view

The resource view models represent health care activities performed by various resources. In this paper, the resource view model of each resource are represented by statechart models for all needed missions. Each mission corresponds to a set of health care activities identified in the process view models and some other activities that are needed but are not seen by patients. A resource may have several missions depending of its involvement in the patient care process. This representation allows each health care professional to define accurately his own activities.

Definition 3.8 (R-state): An R-state is a couple $rs = (l, \tau)$ as $l \in \mathbb{L}$ is a unique label, and τ is the processing time.

An R-state in the statechart of a mission can be *synchronized* with a state of the process view corresponding to the use of the resource for a care activity.

Definition 3.9 (State synchronization): Let $v = (l, \tau, R)$ and $w = (l', \tau')$ two states from process and resource views respectively. v and w are synchronized if $l = l'$.

The execution of synchronized states is tied to the presence of an entity in the process view model and the presence of a resource in the resource view model. The proposed approach is close to the behavior of medical staff in hospitals, where medical activities have to be synchronized in order to ensure that all necessary resources are available. The execution is performed simultaneously in both views. Entities and resources are bound until the end of the synchronization.

The synchronization can be extended to a set of states in order to model the requirement of the same resource(s) for several consecutive activities.

Definition 3.10 (Multiple state synchronization): Let $M = (E, V, \zeta)$ be a subgraph of the statechart modeling a mission of a resource r and $M' = (E', V', \zeta')$ be a subgraph of a process view model. M and M' are synchronized if (i) subgraphs of M and M' are identical, (ii) any couple of states in E and E' are synchronized, and (iii) the number of units of resource r needed in all states of E is the same.

In the following, a mission is said *simple* if its statechart contains only one state corresponding to a P-state of the process view models. Otherwise, a mission is said *complex*. For a simple mission, the resource is needed when the corresponding care activity starts and is released immediately at the completion of the activity. For a complex mission, a resource might be needed before the corresponding care activity starts in order to perform some additional setup or preparation activities and the resource may be hold for several activities.

Example 3.11 (Synchronization example): Figure 3 presents two synchronized statecharts. In this example, the `anaesthetist` is a resource without complex mission: allocation is performed as soon as it is available and needed by states of process view models. To the contrary, the `operating_room` has a complex mission and needs first to be prepared before it can be used for activities of process view. Anaesthesia begins simultaneously in process and resource view. Depending on the patient type, a type A or B surgery is performed by a `surgeon` in the same `operating_room`. Patient's transfer is performed at the end of the surgery operation by a `porter` in the process view. The operating room is released and cleaned while the patient is transferred back to the ward.

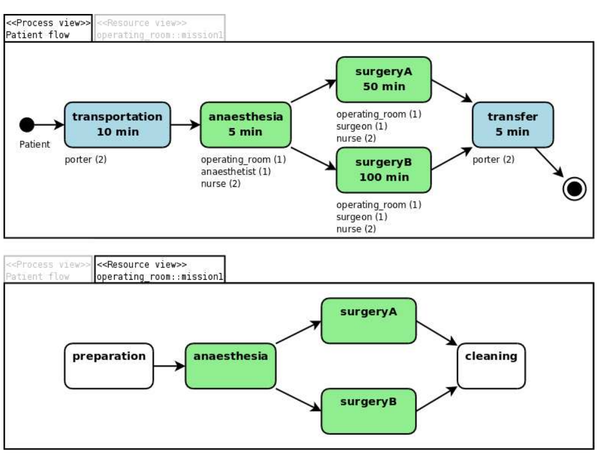


Fig. 3. Basic synchronization example

D. Organization view

The organization view offers a global representation of resources definitions, interactions, scope of actions and specialties. UML class diagrams are used to model such relations. Three types of relations have been identified in health care systems: (i) *inheritance*, to model resource specialization and hierarchy; (ii) *association*, to model resource sets or teams; (iii) *ability*, to model medical abilities or competencies of resources.

Definition 3.12 (Specialization relation): Let two resources r_1 and r_2 where $r_1 = (A_1, \mathcal{P}_1, \mathcal{M}_1)$ and $r_2 = (A_2, \mathcal{P}_2, \mathcal{M}_2)$. r_1 inherits r_2 if $A_2 \subseteq A_1$ and $\mathcal{M}_2 \subseteq \mathcal{M}_1$. We note $r_1 \triangleright r_2$.

The specialization relation allows clear and flexible representation of organizational charts using class diagrams and it also allows efficient definition of different types of resources. Medical specialization of surgeons and physicians can also be modeled (surgeon B has "better skill" for some surgeries than surgeon A) using attributes. Finally, inheritance relation is an important time saving for the modeler because a child class gets instantly attributes and missions of its parent class.

From an operational point of view, the inheritance relation allows the control system to replace an unavailable resource during simulation.

Definition 3.13 (Team class): A team class is a 3-tuple $t = (A, R, \mathcal{M})$ where A is a list of attributes, R is the set of team members, and $\mathcal{M}$ is the set of specific missions for this team.

Definition 3.14 (Team relation): Let $r = (A, \mathcal{P}, \mathcal{M})$ be a resource and $t = (A', R, \mathcal{M}')$ a team. r belongs to team t if $r \in R$ and if $\mathcal{M}' \subseteq \mathcal{M}$. We note $r \mapsto t$.

A team can perform various activities during missions. If one or more members of a team are not available, other resources are hold till the formation of the team. A resource may belong to several teams. Association conditions may be defined in the team class (efficiency of the team for some tasks for example).

The definition of teams allows simple and clear representation of resource requirement in process view models.

Definition 3.15 (Ability class): An ability class is a 3-tuple $a = (A, R, \mathcal{M})$ where A is a list of attributes, R is a set of resources having the ability, and $\mathcal{M}$ is a set of missions for this ability.

Definition 3.16 (Ability relation): Let $r = (A, \mathcal{P}, \mathcal{M})$ be a resource and $a = (A', R, \mathcal{M}')$ an ability. r has ability a if $r \in R$ and if $\mathcal{M}' \subseteq \mathcal{M}$. We note $r \succ a$.

All resources with a common ability inherit instantly missions defined for this ability. When an ability is defined as a class, it can be used in process and resource view as a simple resource, for the execution of a task for example. The control sub-system is invoked to choose the best qualified resource when needed.

E. Additional remarks

Statecharts used for patient flow and resource activity modeling offer a clear and complete representation of a health care system. Class diagrams allow the modeling of system organization and hierarchy as well as resource relations. UML has been chosen as a BPM tool for its straightforwardness and its modularity. Models are easy to understand during meetings with medical teams and enable easy communication with all concerned people. The multi-view approach has extra benefits for health care professionals and/or patients to understand, modify and validate their own models.

IV. A CLASS OF PETRI NETS FOR HEALTH CARE SYSTEMS

The high-level UML models of the previous section offer a clear and complete representation of the health care system under consideration. However those models are not executable and hence cannot be used directly to simulate the system for performance evaluation and optimization. Instead of proposing a new simulation algorithm for statecharts, the UML models are converted to an integrated executable Petri net model. A new class of Petri nets is defined in this section to take into account special features of the multi-view UML models.

A. Some basic notions of Petri nets

An ordinary Petri net (PN) is a 4-tuple $\mathcal{R} = (P, T, F, M_0)$ where P and T are two disjointed sets of nodes called respectively places and transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs, $M_0 : P \to \mathbb{N}$ is the initial marking of the net.

The set of input (resp. output) transitions of a place $p \in P$ is denoted by $\bullet p$ (resp. $p\bullet$). Similarly the set of input (resp. output) places of a transition $t \in T$ is denoted by $\bullet t$ (resp. $t\bullet$).

A transition $t \in T$ is said to be enabled at M_0 if for all $p \in \bullet t$, $M_0(p) \geq 1$. A transition may fire if it is enabled. The firing of a transition t at marking M removes one token from each of its input places and puts one token to each of its output places. This leads to a new marking, say M'. This process is denoted by $M_0[t > M'$. If M' is not explicitly mentioned, the process is denoted by $M[t >$, which means that t is enabled at M. These notations are also extended to sequences of firings, i.e. $M[\sigma > M'$, where σ is a sequence of transitions that brings M to M', and $M[\sigma >$, if M' is not explicitly mentioned. The set of all markings reachable from M_0 is denoted by $R(M_0)$.

The structure of a Petri net can also be represented by its incidence matrix $U = [u_{ij}]$ for $i \in \{1, \ldots, |P|\}$ and $j \in \{1, \ldots, |T|\}$ with $u_{ij} = 1$ if $t_j \in \bullet p_i$, $u_{ij} = -1$ if $t_j \in p_i\bullet$, and $u_{ij} = 0$ otherwise. Given the incidence matrix, the state equation is $M = M_0 + U\vec{\sigma}$ where M is the marking obtained by firing the sequence σ of transitions at M_0 and $\vec{\sigma}$, called the firing count vector, is a $|T| \times 1$ column vector whose i-th entry denotes the number of times that transition t_i appears in σ.

A source transition is a transition without any input place. A source transition is always enabled. A sink transition is a transition without any output place. When firing a sink transition, all tokens are removed respecting usual rules but no tokens are generated.

A T-timed Petri net is a 5-tuple $\mathcal{R} = (P, T, F, \theta, M_0)$ where $\theta : T \to \mathbb{N}$ assigns to each transition t its transition firing time $\theta(t)$. Firing a timed transition t at time d removes immediately one token from each input place but add tokens to its output places only at time $d + \theta(t)$.

A colored Petri net is a 7-tuple $\mathcal{CPN} = (P, T, C, A, W^+, W^-, M_0)$ where $C : (P \cup T) \to \Omega$, $C(p)$, $p \in P$, is the set of colors associated to a place p (i.e. the set of colors that place p may have), $C(t)$, $t \in T$, is the set of colors associated to a transition t (i.e. the set of ways to fire t), $W^-_{p,t} : C(t) \to \mathbb{N}^{|C(P)|}$ is the pre-condition of a transition in relation to a color which defines for each way of firing t the required combination of tokens of different colors in different places, $W^+_{t,p} : C(t) \to \mathbb{N}^{|C(P)|}$ is the post-condition of a transition in relation to a color which defines for each way of firing t the combination of tokens of different colors added to different places.

Let $\vec{w}$ be a t-invariant of a Petri net N, i.e. $U\vec{w} = 0$, where U is the incidence matrix of N. The Petri net $N_{\vec{w}}$ is $\vec{w}$-derived from the Petri net N if: (i) the set of transition of $N_{\vec{w}}$ is $||\vec{w}||$; (ii) $\forall t \in ||\vec{w}||$, $\bullet t$ and $t\bullet$ are identical in N and $N_{\vec{w}}$; (iii) each arc of $N_{\vec{w}}$ has the same weight as the corresponding arc in N.

We call $\vec{w}$-CFIO (*Conflict Free net with Input and Output transitions*) of N, where $\vec{w}$ is a t-invariant of N, the Petri net $N_{\vec{w}}$ $\vec{w}$-derived from N having the following properties: (i) $p\bullet$ is unique for all places of $N_{\vec{w}}$ (each place has exactly one output transition); (ii) there is at least one transition $t_1 \in ||\vec{w}||$ and one transition $t_2 \in ||\vec{w}||$ as $\bullet t_1 = \oslash$ (t_1 is a source transition) and $t_2\bullet = \oslash$ (t_2 is a sink transition); (iii) $N_{\vec{w}}$ has no cycle.

Let N be a Petri net and $\vec{w_1}, \ldots, \vec{w_k}$ a set of t-invariants of N such as: (i) the nets $N_{\vec{w_i}}$ $\vec{w_i}$-derived from N are $\vec{w_i}$-CFIO, with $i \in \{1, \ldots, k\}$; (ii) the $\vec{w_i}$-CFIO covers N: $N = N_{\vec{w_1}} \cup \cdots \cup N_{\vec{w_k}}$. Then N is said to be decomposable.

B. Conversion algorithm

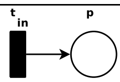
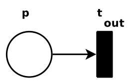
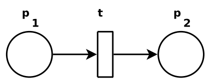
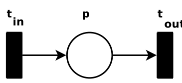
A sequential and automatic UML-Petri net conversion algorithm is proposed to define a new class of Petri nets: (i) the process view is examined and the patient pathway is converted; (ii) statecharts associated to resource missions are modeled; (iii) sub-nets corresponding to resources and their interactions described in the organization view are finally added.

1) Preliminaries: The following sets are defined from the UML model: let S (resp. V) be the set of states (resp. of UML transitions) in process and resource views; let R (resp. AB, TE) be the set of resources (resp. abilities, teams). Let $\mathcal{RP} = (P, T, F, W, \theta, M_0)$ be a T-timed Petri net.

Conversion rules are presented in Table I. An UML state is converted into a transition with one input and one output place. An UML entry node (resp. exit node) is converted into a source transition followed by a place (resp. a sink transition preceded by a place). The source transition models the arrival of entities into the model following a pattern defined in the UML model, whereas its output place models a waiting location before one or several conditional transition(s). A P-state (resp. R-state) is converted into a timed transition preceded by an input place (modeling a location where entities gather after one or several conditional transition(s) and waits in a queue for the activity) and followed by an output place (modeling a place with one or several exit(s) to conditional transition(s) depending on a post-activity branching condition). A base state for a resource is converted into a sub-net with one source transition (modeling the arrival of resources at the beginning of each working shift), one place (idle state), and one sink transition (modeling the departure of resources at the end of each working shift). Finally, those Petri net items are connected to each other using flow relations of the UML models.

Let $T_{in} \subset T$ (resp. $T_{out} \subset T$) be the set of source transitions (resp. sink transitions); let $TE \subset T$ (resp. $TC \subset T$) be the set of transitions corresponding to states (resp. corresponding to transitions between states); let $PB \subset P$ be the set of places corresponding to base states of resources.

TABLE I
CONVERSION RULES

MedPRO	Petri net	Sets
Entry node		$t_{in} \in T_{in}$; $p \in P$
Exit node		$t_{out} \in T_{out}$; $p \in P$
P-state/R-state		$p_1, p_2 \in P$; $t \in TE$
Base state		$t_{in} \in T_{in}$; $p \in PB$; $t_{out} \in T_{out}$
Cond. transition		$t \in TC$

Let $\gamma : S \rightarrow TE$ with $\gamma(s)$ being the transition corresponding to an UML state $s \in S$. Let $\eta : V \rightarrow TC$ with $\eta(v)$ being the transition corresponding to the UML transition $v \in V$. Let $\Gamma : R \cup AB \cup TE \rightarrow PB$ where $\Gamma(r)$ is the idle place corresponding to a resource r. Finally, let $\xi : TC \rightarrow \mathbb{E}$ where $\xi(t)$ is the condition of transition $t \in TC$.

2) Process view conversion: The first step consists in building the Petri net associated to the process view. Resource requirements are not taken into account. For each statechart $PM = (S, V, \zeta)$ of the process view, the following algorithm is applied:

i Convert the UML entry node into a pair of source transition $t_{in} \in T_{in}$ and output place $p \in P$ and add arc (t_{in}, p) to F.

ii Convert the UML exit node into a pair of input place $p \in P$ and sink transition $t_{out} \in T_{out}$ and add arc (p, t_{out}) to F.

iii Convert each P-state $ps \in E$ with duration τ into a sub-net made of an input place $p_1 \in P$, a transition $t \in TE$ and an output place $p_2 \in P$; add arc (p_1, t) and arc (t, p_2) to F. Transition t is associated to state ps with $\gamma(ps) = t$ and the firing duration of t is $\theta(t) = \tau$.

iv For each UML transition $(v, w) \in V$: 1. create a transition $t \in TC$; 2. add $(\gamma(v)\bullet, t)$, $(t, \bullet\gamma(w))$ to F; 3. associate transition t to (v, w): $\eta(v, w) = t$; 4. associate the boolean condition to t: $\xi(t) = \zeta(v, w)$.

The Petri net is ordinary, i.e. $W(x, y) = 1 \; \forall \; (x, y) \in F$. Finally, each statechart of the process view is associated with a flat stand-alone Petri sub-net.

Example 4.1: Figure 4 presents a P-state modeling without resource constraints. States `E1`, `E2`, `E3` and `E4` correspond respectively to transitions t_3, t_5, t_8 and t_{10}. Following the conversion algorithm, we build the Petri net $\mathcal{RP} = (P, T, F, W, \theta, M_0)$ where (i) $P = \{p_1, \ldots, p_{10}\}$, (ii) $T = \{t_1, \ldots, t_{12}\}$ (with $TE = \{t_3, t_5, t_8, t_{10}\}$ and $TC = \{t_2, t_4, t_6, t_7, t_9, t_{11}\}$), (iii) $F = \{(t_1, p_1), (p_1, t_2), \ldots\}$, and (iv) $W(x, y) = 1 \; \forall (x, y) \in F$. Initial marking is null.

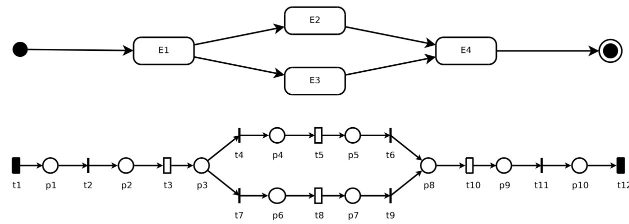


Fig. 4. Conversion of a process view UML model

3) Resource view conversion: Once the process view models are converted, base states of resources, teams and abilities defined in the organization view are added to the net. For each ordinary resource $r \in R$ available in mp units, a Petri sub-net connecting a source transition $t_{in} \in T_{in}$ to a idle place $p \in PB$ to a sink transition $t_{out} \in T_{out}$ is created. The idle place p is associated to the resource r ($\Gamma(r) = p$) and $W(t_{in}, p) = W(p, t_{out}) = mp$. For each team and ability $r \in TE \cup AB$, an idle place $p \in PB$ is created. This place is associated to the resource r ($\Gamma(r) = p$).

At the end of this step, we have a Petri net with all idle places of resources defined in the organization view. We still have to convert statecharts of missions.

Consider first a complex mission $mi = (M, e_{in}, e_{out}, mp_{mi})$ defined for a resource $r \in R \cup AB \cup TE$, where $M = (S, V, \zeta)$. We first convert the statechart associated to mission mi the same way as for conversion of process view models. We get the Petri net associated to the mission mi, where input and output are modeled using two transitions t_{in} and t_{out} respectively. This Petri net and the Petri net associated to the process view are united to model the synchronization mechanism by merging common places and transitions. Complete merging mechanism is detailed in the following through an example.

Example 4.2: Figure 5 presents a conversion example with synchronization between the process view and the resource view. The Petri sub-net G_1 is the conversion of the process view with states `State1`, `State2`, `State3`, `State4` and `State5` modeled by transitions t_1, t_3, t_5, t_8 and t_{10}. p_9 is the resource idle place of `R1`. The Petri sub-net G_2 is the conversion of the mission presented in the resource view where states `State2`, `State3`, `State4` and `State6` are modeled by transitions t'_3, t'_5, t'_8 and t'_{10}; input and output of the mission are modeled with transitions t'_1 and t'_{12} respectively. The third sub-net presents G_1 and G_2 when merged; input and output of the mission are linked to the base place of the resource.

Simple missions can be easily taken into account by connecting resource idle place to the transitions corresponding to the state of the process view. For each state $s = (l, \tau, R) \in E$, for each resource $(r, mp) \in R$ needed as a simple mission, two arcs of weight mp are added to connect resource idle place to and from the transition $\gamma(s)$ of the UML state s.

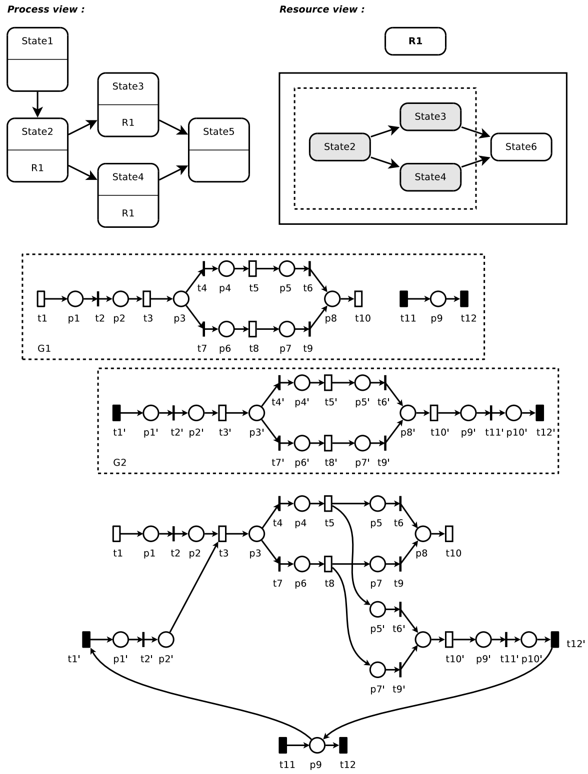


Fig. 5. Statechart conversion with synchronization

Example 4.3: Figure 6 presents a modeling example of a P-state with simple mission resources. States E1 and E2 correspond to transitions t_1 and t_2. The execution of the task corresponding to E2 requires a unit of R1, a unit of R2 and two units of R3. All required resources are ordinary resources modeled with places pr_1, pr_2 and pr_3. Following the conversion algorithm, we build the Petri net $\mathcal{RP} = (P, T, F, W, \theta, M_0)$ where (i) $P = \{p_1, \ldots, p_6, pr_1, pr_2, pr_3\}$ and $P_{base} = \{pr_1, pr_2, pr_3\}$, (ii) $T = \{t_{in}, t_{out}, t_1, t_2, t'_1, t'_2, t'_3\}$, $TE = \{t_1, t_2\}$ and $TC = \{t'_1, t'_2, t'_3\}$, (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) $W(pr_3, t_2) = W(t_2, pr_3) = 2$ and $W(x, y) = 1$ $\forall (x, y) \in F \setminus \{(pr_3, t_2), (t_2, pr_3)\}$. We also have $\theta(t_1) = 25\ min$ and $\theta(t_2) = 30\ min$.

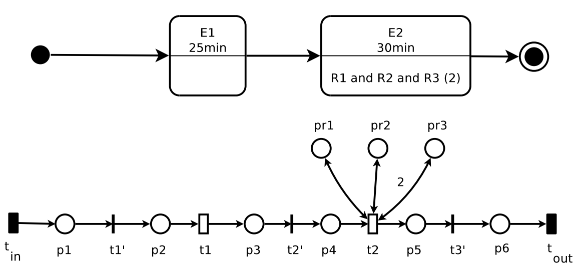


Fig. 6. Statechart conversion with simple mission resources

4) Organization view conversion: The last conversion step consists in modeling relations between resources (inheritance, abilities and teams). Because of space limitations, we will only give an overview of these algorithms in the following.

First team relations are considered. Let $TE = \{te_1, \ldots, te_n\}$ be the set of team classes. Two transitions t_a and t_b are defined to get or release all team members. On the other hand n places and transitions are defined to pre-select team members. From a dynamic point of view, setting up or breaking up a team can be done only under special conditions like resource allocation for a task or beginning a mission. Figure 7 presents the sub-net associated to a team.

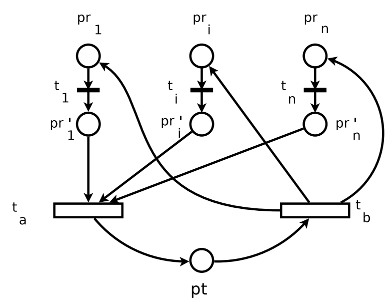


Fig. 7. Team sub-net

To model inheritance and ability relations, colored Petri nets are required. Let Ω be the set of colors. A color $c(r)$ is associated to each ordinary resource r in order to keep track of its resource type identity when it is used in a team or when it replaces another resource of a different type. Hence we have $\Omega = \{c(r_1), \ldots, c(r_n)\}$ with $r_1, \ldots, r_n \in R$. Functions C, W^-, W^+ are defined in Section IV-A.

Let $CH = \{(r_1, q_1), \ldots, (r_n, q_n)\}$ be a set of ordinary resource pairs (r_i, q_i) where $r_i \triangleright q_i$. The inheritance relation allows resource r_i to replace q_i for one or several task(s). For each resource pair (r, q), two transitions t_a and t_b are defined. There is only one way to fire t_a and t_b. Color of tokens in place $\Gamma(q)$ can be $c(q)$ or $c(r)$. Pre-conditions and post-conditions of these transitions are used to allow replacement in only one way. From a dynamic point of view, a replacement occurs only in special conditions like resource allocation for a task or a mission. Transition firing is decided by the control sub-system. Figure 8 presents the sub-net associated to an inheritance relation where pr and pq are idle places of resources r and q.

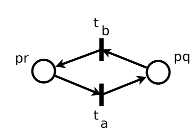


Fig. 8. Inheritance sub-net

Finally, we have to link each ability to ordinary resources having the ability. For each ordinary resource r_i which has ability co, two transitions $t_{1,i}$ and $t_{2,i}$ are defined. As said before, the control sub-system decides which resource will be selected to perform a task requiring a certain ability. Figure 9 presents the sub-net associated to an ability relation where pr_i is the idle place of resource r_i and pa the idle place of the ability.

5) Merging sub-models: Finally, Petri nets generated in different views are merged into a single integrated Petri net using the following rules:

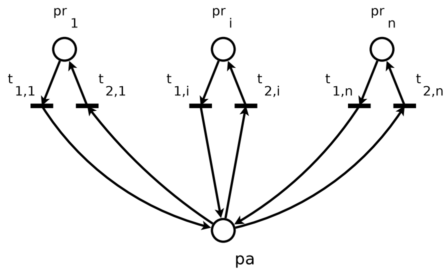


Fig. 9. Ability sub-net

- Process view and resource view models are integrated by merging common transitions corresponding to synchronized states defined in Definition 3.9 and common sub-nets of multiple state synchronization defined in Definition 3.10. For example, if a resource is needed for several consecutive synchronized states, the whole sub-Petri net corresponding to those states in the resource view will be merged with the corresponding part in the process view.
- Resource view and organization view are integrated by merging idle places of the resource view Petri net (Section IV-B3) with all corresponding places in the organization view Petri nets. For the same resource r, its idle place bp in the resource view model and its idle place p in the organization view model will be merged together.

We finally define a Health Care Petri Net (HCPN) as a set of synchronized statecharts with resource allocation in which resources are linked together by several types of relations allowing the modeling of teams, shared abilities and replacements. Colored Petri nets are used to model such relations. Colors are used to preserve resource identity in the Petri net.

Example 4.4: Figure 10 presents a complete conversion example. The UML model is presented in the upper part of the figure (process, resource and organization views). The corresponding Petri net is presented in the lower part. Process view has a partial statechart with five states `State1` to `State5`. Five resource classes `R1`, ... ,`R5` and an ability class `A1` are considered. Finally, a complex mission is defined for resource class `R1` whose statechart has three states `State2`, `State3` and `State6` ; `State2` and `State3` form a multiple synchronized state set. Simple missions are obvious and are not explicitly represented.

In the Petri net, transitions t_1, t_3, t_5, t_8 and t_{10} correspond to states from the process view, and transition t_{16} correspond to state `State6`. Places $pr_1, \ldots, pr_5$ correspond to resources classes and place pc_1 corresponds to the ability class. For a better readability, source/sink transitions are not shown on the diagram. The Petri net of Figure 10 is defined as follow:

1) P is the set of places with $PB = \{pr_1, pr_2, pr_3, pr_4, pr_5, pc_1\}$.
2) T is the set of transitions with $TE = \{t_1, t_3, t_5, t_8, t_{10}\}$ and $TC = \{t_2, t_4, t_6, t_7, t_9, t_{14}, t_{15}, t_{17}\}$.
3) $F = \{(t_1, p_1), \ldots\}$.

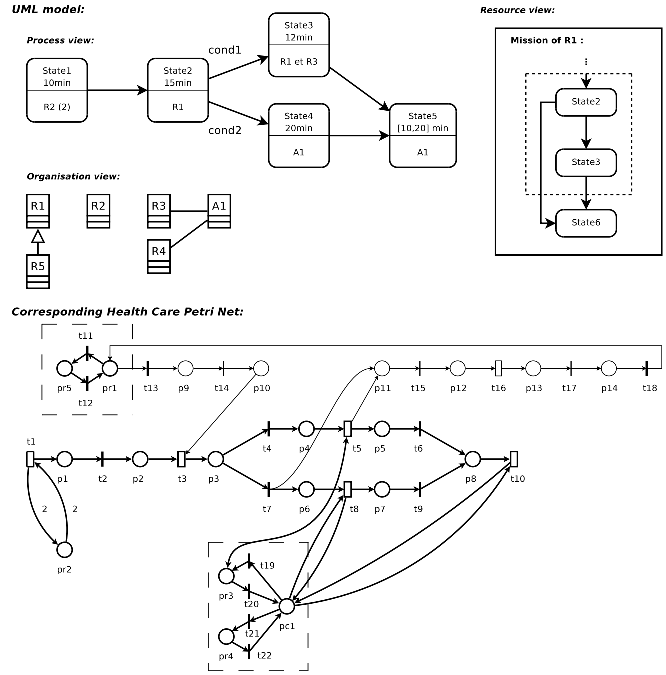


Fig. 10. Complete statechart conversion

4) $\Omega = \{c(r_1), c(r_2), c(r_3), c(r_4), c(r_5)\}$ is the set of colors. $C(pr_1) = \{c(r_1), c(r_5)\}$, $C(pr_2) = \{c(r_2)\}$, $C(pr_3) = \{c(r_3)\}$, $C(pr_4) = \{c(r_4)\}$, $C(pr_5) = \{c(r_5)\}$, $C(pc_1) = \{c(r_3), c(r_4)\}$.
5) The colored sub-nets are framed with dotted lines: $W^-_{pr_3,t_{20}}(1) = W^+_{t_{20},pc_1}(1) = W^-_{pc_1,t_{19}}(1) = W^+_{t_{19},pr_3}(1) = [0,0,1,0,0]$; $W^-_{pr_4,t_{22}}(1) = W^+_{t_{22},pc_1}(1) = W^-_{pc_1,t_{21}}(1) = W^+_{t_{21},pr_4}(1) = [0,0,0,1,0]$; $W^-_{pr_5,t_{12}}(1) = W^+_{t_{12},pr_1}(1) = W^-_{pr_1,t_{11}}(1) = W^+_{t_{11},pr_5}(1) = [0,0,0,0,1]$.
Other weight functions are defined as follow: $W(pr_2, t_1) = W(t_1, pr_2) = 2$ and $W(x, y) = 1$ $\forall (x, y) \in F \setminus \{(r_2, t_1), (t_1, r_2)\}$.
6) $\theta(t_1) = 10$ min, $\theta(t_3) = 15$ min, $\ldots$, $\theta(t_{10}) = UNIF(10, 20)$ min.
7) $\xi(t_4) = \text{cond1}$, $\xi(t_7) = \text{cond2}$.

Colors are useful to identify resources in the net: `R1`'s mission can be carried out by `R5` because `R5` inherits `R1`. However token color during the mission is not considered because it has no impact on the dynamic behavior; color is verified at the beginning and at the end of the mission. The same remark is true for `A1`.

C. Planning elective patients using Petri nets

Using the Health Care Petri Net obtained by automatic conversion of the multi-view UML models, we now propose an optimization model to determine the short term planning of elective patients. Short term planning essentially applies to medical units where patient care or entity processes does not exceed one day. Ambulatory services or operating theaters belong to this category where activity planning is required.

In order to apply short-time planning methods to a Health Care Petri Net, several restrictions are considered:

- H_1: The Petri net model of the process view has no cycle.
- H_2: Each complex mission corresponds to a unique path. Selection of alternative paths is made a priori.
- H_3: Resource requirements of activities belonging to resource view are not considered.

The Petri net model of the process view can be split into different CFIO (Conflict Free net with Input and Output transitions) in order to explicitly represent different choice-free care pathways of each patient.

As each Petri net of the process view model is a state machine without synchronization, it is decomposable and is covered by a finite set of CFIO corresponding to the set of minimal t-invariants $\{W_1, W_2, \ldots, W_n\}$. Let N_{W_s} be the W_s-CFIO. In the following, we denote $W_s = [w_{s,1}, \ldots, w_{s,q}]$ where q is the number of transitions of the Petri net of the process view.

Let $\{u_1, \ldots, u_k, \ldots, u_p\}$ be the set of patients to plan on time periods $1, \ldots, H$. We define the following decision variable:

$$y^j_{s,k} = \begin{cases} 1 & \text{if CFIO } N_{W_s} \text{ has been activated during time} \\ & \text{period } j \text{ for patient } k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Let $r_1, \ldots, r_m$ be the resources and T^g_s, $g \in \{1, \ldots, m\}$, the set of transitions of CFIO N_{W_s} requiring resource r_g. Clearly T^g_s is a set of transitions of the process view model. Capacity constraints may be written as follows:

$$\sum_{k=1}^{p} \sum_{s=1}^{n} \left(y^j_{s,k} \Big(\sum_{t \in T^g_s} w_{s,h(t)} \theta_{t,k} \Big) + \alpha_{s,g} \right) \le \tau \quad \forall j \in \{1, \ldots, H\},\ \forall g \in \{1, \ldots, m\} \tag{2}$$

where $h(t)$ is the CFIO of transition t, $\alpha_{s,g}$ is the total time of resource specific activities for resource r_g in CFIO N_{W_s}, τ is the duration of each time period and $\theta_{t,k}$ is the firing time of transition t for patient k.

Each patient must be assigned to one time period and one CFIO:

$$\sum_{j=1}^{H} \sum_{s=1}^{n} y^j_{s,k} = 1 \quad \forall k \in \{1, \ldots, p\} \tag{3}$$

Finally the optimization problem can be written as follows:

$$\text{Optimize } C(\{y^j_{s,k}\}) \tag{4}$$

under constraints (2) and (3). C is the criterion to optimize, with $y^j_{s,k} \ge 0\ \forall j, k, s$.

A scheduling optimization model can also be generated using a similar technique. These techniques will not be described in this paper due to space limitations. A practical application will be presented in the following section.

D. Simulation and control

The Health Care Petri Net defined above is executable and can be directly used to simulate the behavior of the related health care system. Discrete-event simulation is used. Tokens are generated following probability distributions and schedules provided in the UML models. All timed transitions are fired as soon as possible. Conditional transitions are fired depending on their corresponding conditions which may involve mathematical expression evaluations. Advanced control is performed automatically during the simulation. For example, resource replacement is automatically proposed when possible by using organization view models in order to avoid blocking situations. The decision maker can propose a simple selection rule, or define a more complex optimization decision process. Specific medical decisions can also be programmed in order to model relevant physician diagnosis decisions.

V. Modeling and analysis of a pharmacy delivery process

The section presents the application of the MedPRO approach to model and analysis the pharmacy delivery process of a local hospital. We will model the complete pharmacy delivery process and take into account the organization issues and all relevant actors. The MedPRO approach will be used to optimize simultaneously medicine transportation and workload of relevant human resources while taking into account various organizational and regulation constraints. Optimization algorithms are combined with the discrete-event simulation of the whole pharmacy delivery process to provide trustful and optimized results using the MedPRO methodology.

A. Problem setting

Pharmacy delivery process can be described as follows. First all medicines arrive to a central pharmacy from outside suppliers each weekday. These medicines are sorted and stored in the main preparation room of the pharmacy in shelves. Each medical unit of the hospital has a mobile medicine closet with medicines related to the unit. Once a week, each closet is conveyed to the pharmacy by transporters to be refilled by pharmacy assistants. Closets are then transported back to their units. The pharmacy also delivers nominated medicines for inpatients with special needs: prescriptions for nominated medicines are sent to the pharmacy by physicians and gathered in the morning to be treated in the afternoon by pharmacy assistants. Urgent prescriptions are treated immediately. These medicines are also conveyed to the appropriate unit by a transporter. Finally, medicines can also be delivered to outpatients who come to the hospital pharmacy to get specific drugs which cannot be found in town dispensaries.

In this organization, medicines are kept in a mobile medicine closet in each medical unit; pharmacy assistants remain in the central pharmacy to check inventory and refill closets, while transporters convey these closets. These closets are conveyed either by foot (if the unit is located in the same building), by tractor (if the unit is located in a nearby building) or by truck (if the unit is located in a building outside of the hospital site).

The goal of this case study consists in (i) providing a formal representation of the organization for the central pharmacy, (ii) determining delivery plans for all medical units of the hospital, (iii) simulating the pharmacy delivery process to test the robustness of the delivery plan computed in (ii) under stochastic conditions.

B. System model

In order to capture both pharmacy and transportation processes, two views are considered:

- **Pharmacy centered-view:** in this process view, we only model internal flows of the pharmacy including inventory checking and refill of mobile medicine closets, preparation of urgent medicine requests... Medicines and closets are considered as entities while chemists and assistants are considered as resources.
- **Transporter centered-view:** medicine transportation flows are modeled in this resource view. It includes delivery by foot to units near the pharmacy, delivery by tractor to neighboring buildings and delivery by truck to other facilities.

The MedPRO framework perfectly fits the dual-view modeling. The process view is used to model processes related to mobile closets and medicines, whereas the resource view is used to model transporters' activities. As seen before, synchronization is included in the framework. UML models ensure the consistency and the clarity of the model. Process and resource views of the MedPRO approach are presented in Figures 11 and 12 respectively. Note that missions of pharmacy assistants are all simple missions and are not explicitly represented in the resource view.

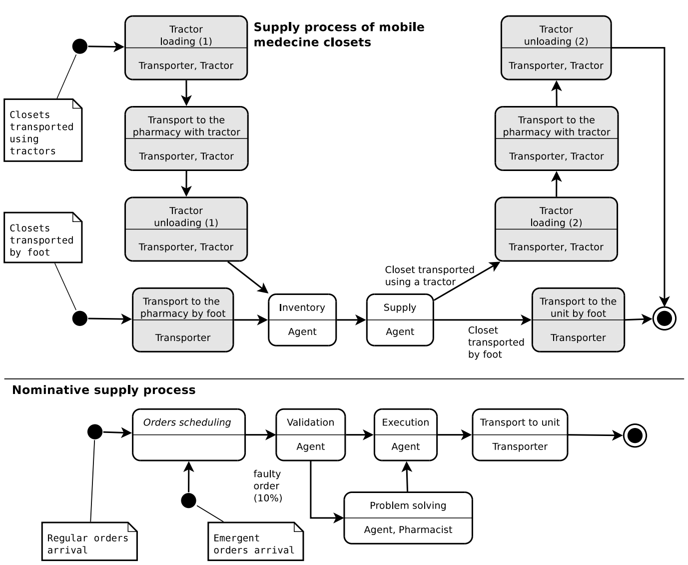


Fig. 11. Process view of pharmacy

Two processes are presented in the process view: (i) mobile closets conveyed by foot and by tractor, and (ii) nominated medicines delivery. We focus on transporters' activity in the resource view. For closets supply, synchronization between the transporter and the process view is required for all transportation activities (grey boxes on the figures). Activities of

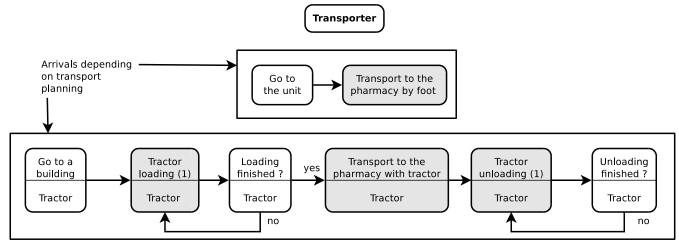


Fig. 12. Resource view of pharmacy for transporters with two missions

transporters are presented in Figure 12. Transporters by foot have to go to a medical unit to take a closet. Transporters using tractors must go to a building, load closets, and transport them to the pharmacy. These missions are planned using optimization techniques in the following subsection. Assistants and pharmacists are modeled as ordinary resources without complex missions.

C. Planning and scheduling using Petri nets

Using the Health Care Petri Net deriving from the UML models, we propose here an optimization model to determine: (i) weekly medicine closet supply planning and (ii) scheduling of transportation tasks.

A solution approach in two steps is proposed. It first determines a set with the minimal number of pickup routes subject to carrier capacity constraints and other constraints such as of the availability of at least one closet at all time in each medical unit.

The second step determines the optimal closet supply planning under capacity constraints of transporters and carriers and the closet availability constraints of medical units. The goal of closet supply planning is to best balance the per pharmacy assistant workload C over the week defined as follows:

$$C \geq \sum_{i \in L} \frac{p_i}{n_t S_t} x_{it}, \ \forall t \in \{1 \dots T\} \tag{5}$$

where L is the set of pick-up routes in the hospital, i is a route, T is the number of periods (generally a half-day) in a week, t is a period, p_i is the total supply time for all closets of route i, n_t is the number of assistants available during period t, S_t is the working hours of assistants during period t. The binary decision variable x_{it} is equal to 1 if route i is assigned to period t.

The method described in Section IV-C has been applied to our case study using real data sets from the hospital. Medicine closet supply planning is computed using short term planning optimization whereas task scheduling for transporters is computed using scheduling optimization.

The optimization method is tested on a real scenario of two transporters, ten assistants, one tractor and one truck. Table II presents the workloads of transporters and assistants obtained with the planning model for a regular week.

Our planning and scheduling models reach a very good balance of workloads for transporters and for assistants over a week. Workloads of pharmacy assistants presented in Table II does not take into account administration duties. For this

TABLE II
GENERATION AND ASSIGNMENT OF ROUTES ON A REGULAR WEEK

Half-day	Transportation time (transporters)	Preparation time (assistants)
Monday AM	55.5 min	178.5 min
Tuesday AM	56.7 min	189.0 min
Tuesday PM	57.7 min	179.7 min
Wednesday AM	56.7 min	178.5 min
Thursday AM	54.4 min	178.5 min
Thursday PM	56.6 min	178.5 min
Friday AM	58.7 min	178.5 min
Standard deviation	1.4 min	3.9 min

TABLE III
SIMULATION RESULTS: WORKLOADS OF ASSISTANTS

Half-day	Avg. workload	Std. deviation	Min.	Max.
Mon AM	2 h 32 min	2 min 35 s	2 h 10 min	2 h 59 min
Tues AM	2 h 35 min	2 min 54 s	2 h 11 min	3 h 12 min
Tues PM	2 h 27 min	3 min 10 s	2 h 00 min	3 h 09 min
Wed AM	2 h 31 min	2 min 51 s	2 h 03 min	3 h 02 min
Thurs AM	2 h 31 min	2 min 54 s	2 h 03 min	3 h 09 min
Thurs PM	2 h 32 min	2 min 51 s	2 h 02 min	3 h 07 min
Fri AM	2 h 36 min	1 min 21 s	2 h 19 min	2 h 54 min

reason, workload ratios are always lower than fifty percents. Notice that using the MedPRO methodology, the mathematical model could be built automatically from the Health Care Petri Net, generated from the UML model. No additional programming is required since the first UML model.

D. Simulation

The simulation model was built from the MedPRO model using both the proposed methodology and the Arena simulation software. The Arena model has been built following the logic of the UML model of the pharmacy department: beginning and ending nodes are converted to *Create* and *Dispose* blocks, whereas activities are converted using *Process* blocks. Some intermediary blocks have been used to manage variables, attributes and decisions. The synchronization between the transportation flow and the supply flow has been modeled using *Wait* and *Signal* blocks: this is the main drawback of Arena compared to the proposed framework, where process flows and resource flows are defined in two separate views. Data sets are presented in the Appendix.

The scenario used for simulation has the following features:

- Ten assistants are assigned to supply mobile medicine closets. Another assistant is available for urgent tasks when needed. One regular assistant can be allocated to other tasks in the pharmacy.
- Each assistant have a total of 3 hours to supply medicine closets in each period.
- Transporters have 1 hour to pick-up or deliver closets in each period. One transporter is assigned to external delivery with a truck, one transporter to internal delivery with a tractor or by foot. The last transporter has no specific tasks, he delivers urgent medicines during the day, helps his co-worker when needed...

As expected, the same results are obtained from both simulation models, although minor differences due to different random number generations. Twenty replications were run, each replication having a length of ten regular weeks. The simulation model provides statistics such as staff utilization, length of stay of mobile medicine closets, and turnaround time for emergency medicine delivery. Results are summarized in tables III and IV.

The mean workload of pharmacy assistants over the ten weeks is 2 hours and 32 minutes per half-day. Standard deviation is lower than 3 minutes. The maximum workload of 3 hours per half-day is only exceeded on Tuesday AM by 12 minutes in the worst case scenario. The results take into account urgent medicine demands, outpatients and nurses, and unexpected activities such as prescription checking with chemists.

TABLE IV
SIMULATION RESULTS: INTERNAL TRANSPORTS

		Transporter			
		Pick-up		Delivery	
Mon AM	Planned	7:50	9:00	10:20	11:30
	Simulated	7:50	**8:53**	10:20	**11:21**
Tues AM	Planned	7:50	9:00	10:20	11:30
	Simulated	7:50	**8:46**	10:20	**11:15**
Tues PM	Planned	12:00	13:20	-	-
	Simulated	12:00	**13:08**	-	-
Wed AM	Planned	7:50	9:00	10:20	11:30
	Simulated	7:50	**8:56**	10:20	**11:27**
Thurs AM	Planned	7:50	9:00	10:50	11:30
	Simulated	7:50	**8:36**	10:50	**11:06**
Thurs PM	Planned	12:00	13:20	-	-
	Simulated	12:00	**12:13**	-	-
Fri AM	Planned	7:50	9:00	10:40	11:30
	Simulated	7:50	**8:42**	10:40	**11:06**

The workload assignment over the week for both transporters and assistants is appropriate because time constraints are not violated during the simulation over a week. Assistant workloads are mostly lower than the given maximum of 3 hours and the overtime is within an acceptable range, and transportation of medicine closets always ends before noon (even in the worst case scenario). Taking into account task duration uncertainty, we are able to propose to the pharmacy a very robust schedule. Even if unexpected events occur after the planning generation, the lead assistant can reschedule tasks to take into account urgent prescription requests or expected absenteeism of assistants during the week.

The work plan generated by the optimization program is a good solution because workloads of assistants and transporters are well balanced over the week. The second transporter is mainly assigned to end-of-day transportation, to urgent delivery and other minor tasks. Finally, enough assistants are always planned to prepare the mobile medicine closets. These results have been shown to the entire pharmacy team during the final meeting and the new organization has been validated.

E. Discussion

The pharmacy staff was very interested in the optimization tool and the modeling methodology included in the MedPRO framework for its flexibility and clarity. The UML models were used during meetings with pharmacists and assistants, who were able to comment and directly correct the models of their pharmacy. Each worker was able to visualize his own activity (especially transporters) and adjust the operational model thanks to the multi-view approach. Numerical data were collected easily using the MedPRO model. Unavailable data were generated randomly using data history and relevant probability distributions.

The automatic conversion of the UML models generated a Health Care Petri Net and a short term planning model, which was modified in order to take into account the objective of the decision maker (balancing workloads of the assistants over the planning horizon). The optimization phase produced a work plan for assistants and transporters which was simulated in the simulation model. The proposed plan is robust enough to take into account unexpected events and uncertainties related to task duration and workforce absenteeism. Final results were presented to the pharmacy staff and a simulation demonstration was performed.

VI. Conclusion

We proposed in this paper an integrated approach for modeling and analysis of health care systems. It starts with a multi-view UML modeling approach to represent patient care pathways, resource behaviors and relations between the different resources. The dynamic behavior of the model is described using a special class of colored Petri nets called Health Care Petri Nets obtained by automatic conversion of multi-view UML models. Petri net models are executable and are used directly for simulation and for building optimization models for planning and scheduling of health care activities. Because of space limitations, we were not able to present in enough details the control system which will be detailed in another paper.

This paper can be extended in several directions. First it is necessary to test the MedPRO framework on various health care systems and to compare it with other modeling and simulation tools. The multi-view UML models can also be extended: team and ability concepts are too restrictive, models of absenteeism and replacement of resources are needed. More generally, deadlock prevention and avoidance are not addressed and appropriate techniques are needed to manage resource assignment in order to avoid deadlock situations. Finally, holonic systems [21] may be a relevant tool to complete the control strategy.

Appendix
Data sets for the pharmacy case study

Data sets for the case study are presented in Tables V, VI, VII and VIII. Medical units are grouped together depending on their location in the hospital. Tables present the name of the service (*Service*), the types of closets (*Type*, with 1 for a small mobile closet, 2 for a big mobile closet, 3 for a fixed closet), the floor of the service (*Floor*), and finally the distances between elevators and services in meters (E1 is near the pharmacy and E2 is near the service).

Service	Type	Floor	Distances		
			Ph-E1	E1-E2	E2-Serv
HMU Soins intensifs	3	2	5	180	40
HMU BV	3	2	5	180	40
Urgences Psy.	3	1	5	180	30
Réanimation Poly. BV	3	0	5	180	40
SRPR	3	0	5	180	35
UHCD	3	-1	5	150	20
Urgences graves	3	-1	5	180	30
Urgences fonctionnelles	3	-1	5	180	20
Maladies infectieuses hosp. A	2	3	5	220	40
Maladies infectieuses hosp. B	2	3	5	220	40
Hosp. de semaine rhumato.	2	3	5	220	30
Hosp. complète rhumato.	2	3	5	250	20
Endocrinologie	2	3	5	250	40
Neurologie 12B U1	2	2	5	220	40
Neurologie 12C U2	2	2	5	220	30
Médecine interne 5 EF	2	2	5	250	20
Médecine interne 5 CD	2	2	5	250	40
Consult. HJ Maladies infectieuses	1	1	5	220	20
Rhumato./Endocrino. HJ	2	1	5	250	20

TABLE V
Emergency & consultation building

Service	Type	Level	Distances		
			Ph-E1	E1-E2	E2-Serv
Maladies infectieuses hosp. A	2	3	5	220	40
Maladies infectieuses hosp. B	2	3	5	220	40
Hosp. de semaine rhumato.	2	3	5	220	30
Hosp. complète rhumato.	2	3	5	250	20
Endocrinologie	2	3	5	250	40
Neurologie 12B U1	2	2	5	220	40
Neurologie 12C U2	2	2	5	220	30
Médecine interne 5 EF	2	2	5	250	20
Médecine interne 5 CD	2	2	5	250	40
Consult. HJ Maladies infectieuses	1	1	5	220	20
Rhumato./Endocrino. HJ	2	1	5	250	20

TABLE VI
Consultation building

Service	Type	Floor	Distances		
			Ph-E1	E1-E2	E2-Serv
Ophtalmologie	2	4	5	130	60
Neurochirurgie	2	4	5	130	50
Orthopédie traumatologie U1	2	3	5	130	60
Orthopédie traumatologie U2	2	3	5	130	50
Stomato./Ortho. de sem.	2	2	5	130	60
ORL Unités 1 et 2	2	2	5	130	50
Chirurgie ambulatoire	2	0	5	130	80

TABLE VII
Surgery buildings

Service	Type	Floor	Distances		
			Ph-E1	E1-E2	E2-Serv
Pédiatrie C	2	3	5	85	30
Soins intensifs neonatalité	1	3	5	85	40
Neonatalité	2	3	5	85	40
Pédiatrie B	2	2	5	85	30
Pédiatrie A HJ	1	2	5	85	10
Maternité A	2	1	5	85	30
Grossesse pathologique	2	1	5	85	30
Maternité B	2	1	5	85	60
Réa. néonatalité sect. 1	1	0	5	85	30
Réa. néonatalité sect. 2	1	0	5	85	30
Réa. pédiatrique	3	0	5	85	30
Bloc opératoire gynécologie	3	0	5	85	30
Gynécologie C	2	0	5	85	60
Radiologie mère-enfant	1	-1	60		
Consultation gynécologie	1	-1	80		
MPR Pédiatrie	2	-1	100		
IVG chirurgie gynécologie	2	-1	80		
Urgences pédiatriques	3	-1	150		

TABLE VIII
Maternity building

Acknowledgment

The authors would like to thank the reviewers for their useful comments.

References

[1] V. Augusto and X. Xie, "Modélisation et analyse de flux par la simulation en milieu hospitalier : état de l'art," in *Actes de la conférence GISEH*, pp. 519–528, 2006.

[2] B. Besombes and L. Merchier, "Regroupement des plateaux médico-techniques ; une approche basée sur la modélisation d'entreprise à partir de GRAI," in *Actes de la conférence GISEH*, pp.319–328, 2004.

[3] S. Chaabane, "Gestion prédictive des blocs opératoires," Ph.D. dissertation, Informatique et Information pour la Société, Institut National des Sciences Appliquées de Lyon, 2004.

[4] J. Chauvet, P. Féniès, and M. Gourgand, "A contribution to emergency medical systems modelling and simulation," in *Proceedings of the IEEE Conference on Industrial Engineering and System Management*, pp. 296–301, 2007.

[5] M. Dotoli, M. P. Fanti, A. M. Mangini, and W. Ukovich, "A continuous petri net model for the management and design of emergency cardiology departments," in *3rd IFAC Conference on Analysis and Design of Hybrid Systems, Zaragoza, Spain*, 2009.

[6] M. Dotoli, M. P. Fanti, G. Iacobellis, L. Martino, A. M. Moretti, and W. Ukovich, "Modeling and management of a hospital department via petri nets," in *Proceedings of WHCM 2010: IEEE Workshop on Health Care Management, Venice, Italy*, 2010.

[7] J. A. Effken, B. B. Brewer, A. Patil, G. S. Lamb, J. A. Verran, and K. Carley, "Using orgahead, a computational modeling program, to improve patient care unit safety and quality outcomes," *International Journal of Medical Informatics*, vol. 74, pp. 605–613, 2005.

[8] M. P. Fanti, S. Mininel, W. Ukovich, and F. Vatta, "Modelling alarm management workflow in healthcare according to IHE framework by coloured Petri Nets," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 4, pp. 728–733, 2012.

[9] M. Gourgand and P. Kellert, "Conception d'un environnement de modélisation de systèmes de production," in *Actes du troisième congrès international de génie industriel, Tours, France*, 1991, pp. 191–203.

[10] M. D. Jeng and F. DiCesare, "Synthesis using resource control nets for modeling shared-resource systems," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 317–327, 6 1995.

[11] J. B. Jun, S. H. Jacobson, and J. R. Swisher, "Application of discrete-event simulation in health care clinics: A survey," *Journal of the Operational Research Society*, vol. 50, no. 2, pp. 109–123, 1999.

[12] C. H. Kil, R. H. Weston, A. Hodgsin, and K. H. Lee, "The complementary use of idef and uml modelling approaches," *Computers in Industry*, vol. 50, pp. 35–56, 2003.

[13] J. Li and P. K. Howard, "Modeling and analysis of hospital emergency department: An analytical framework and problem formulation," in *Proceedings of the 6th annual IEEE Conference on Automation Science and Engineering, Toronto, Canada*, 2010.

[14] B. Montreuil, M. Charfeddine, O. Labarthe, and A. Côté, "A generic agent-based framework for simulation in distributed healthcare systems," in *Proceedings of the IEEE Conference on Industrial Engineering and Systems Management*, pp. 302–307, 2007.

[15] L. Moreno, R. Aguilar, C. Martín, J. Pineiro, J. Estévez, J. Sigut, J. Sánchez, and V. Jiménez, "Patient-centered simulation tool for aiding in hospital management," *Simulation Practice and Theory*, vol. 7, pp. 373–393, 1999.

[16] J. P. Oddoye, D. F. Jones, M. Tamiz, and P. Schmidt, "Combining simulation and goal programming for healthcare planning in a medical assessment unit," *European Journal of Operational Research*, vol. 193, pp. 250–261, 2009.

[17] A. Ramudhin, E. Chan, and A. Mokadem, "A framework for the modelling, analysis and optimization of pathways in healthcare," in *Proceedings of the IEEE Conference on Service Systems and Service Management*, 2006, pp. 698–702.

[18] O. Roux, C. Combes, and 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," in *Proceedings of the IEEE Conference on Service Systems and Service Management*, 2006, pp. 692–697.

[19] P. Staccini, M. Joubert, J. M. Quaranta, D. Fieschi, and M. Fieschi, "Modeling health care processes for eliciting user requirements : a way to link quality paradigm and clinical information system design," *Int. Journal of Medical Informatics*, vol. 64, pp. 129–142, 2001.

[20] L. Trilling, B. Besombes, S. Chaabane, and A. Guinet, "Investigation et comparaison des méthodes et outils d'analyse pour l'étude des systèmes hospitaliers," Rapport de recherche sur le projet HRP2, Tech. Rep., Institut National des Sciences Appliquées de Lyon, 2004.

[21] P. Valckenaers, H. V. Brussel, L. Bongaerts, and J. Wyns, "Holonic manufacturing systems," *Integrated Computers Aided Engineering*, vol. 4, pp. 191–201, 1997.

[22] C. Vasilakis and L. Kuramoto, "Comparing two methods of scheduling outpatient clinic appointments using simulation experiments," *Clinical & Investigative Medicine*, vol. 28, pp. 368–370, 2005.

[23] F. Vernadat, *Techniques de Modlisation en Entreprise : Application aux Processus Oprationnels*. Economica, 1999.

[24] T. Wang, A. Guinet, A. Belaidi, and B. Besombes, "Modeling and simulation of emergency services with aris and arena - case study: the emergency department of saint joseph and saint luc hospital," *Production Planning and Control*, vol. 20, no. 6, pp. 484–495, 2009.

[25] X. Xie and M. D. Jeng, "Ercn-merged nets and their analysis using siphons," *IEEE Trans. Robot. Automat.*, vol. 15, no. 4, pp. 692–703, 8 1999.

Vincent Augusto Vincent Augusto received his Ph.D. in Industrial Engineering from the École Nationale Superieure des Mines de Saint-Étienne, France, in 2009. He was a visiting scholar at CIRRELT (Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport), University of Laval, Quebec, Canada in 2009. Currently, he is an assistant professor in the Department of Health care Engineering at the ENSM.SE. His research interests include modeling, simulation, optimization of health care systems and their supply chains.

Xiaolan Xie Xiaolan XIE received his Ph.D degree from the University of Nancy I, Nancy, France, in 1989, and the Habilitation à Diriger des Recherches degree from the University of Metz, France, 1995.

Currently, he is a professor of industrial engineering, the head of the department of Healthcare Engineering of the Center for Health Engineering and the head of IEOR team of CNRS UMR 6158 LIMOS, Ecole Nationale Suprieure des Mines (ENSMSE), Saint Etienne, France. He is also a chair professor and director of the Center for healthcare engineering at the Shanghai Jiao Tong University, China. Before Joining ENSMSE, he was a Research Director at the Institut National de Recherche en Informatique et en Automatique (INRIA) from 2002 to 2005, a Full Professor at Ecole Nationale d'Ingnieurs de Metz from 1999 to 2002, and a Senior Research Scientist at INRIA from 1990 to 1999. His research interests include design, planning and scheduling, supply chain optimization, and performance evaluation, of healthcare and manufacturing systems. He is author/coauthor of over 250 publications including over 90 journal articles and five books. He has rich industrial application experiences with European industries. He is PI for various national and international projects including ANR-TECSAN HOST on management of winter epidemics, NSF China key project on planning and optimization of health care resources, French Labex IMOBS3 project on home health cares, FP6-IST6 IWARD on swarm robots for health services, FP6-NoE I*PROMS on intelligent machines and production systems, the FP5-GROWTH-ONE project for the strategic design of supply chain networks, the FP5-GRWOTH thematic network TNEE on extended enterprises.

He has been an associate editor for IEEE Transactions on Automation Science & Engineering, IEEE Transaction on Automatic Control, IEEE Transactions on Robotics & Automation and has been on the (conference) editorial board of the IEEE Robotics and Automation Society, Health Systems, and Int. J. Simulation & Process Modelling. He is a Guest Editor of various special issues on healthcare engineering and manufacturing systems. He is general chair of ORAHS'2007 (Operational Research Applied to Health care Systems), IPC chair of the IEEE Workshop on Health Care Management WHCM'2010 and IPC member for many other conferences.