Modeling and Integration of Hospital Information Systems with Petri Nets

Houshang Darabi
Dept. of Mechanical and Industrial Engineering
University of Illinois at Chicago
hdarabi@uic.edu

William L. Galanter
Depts. of Medicine and Pharmacy Practice
University of Illinois at Chicago
billg@uic.edu

Janet Yueh-Yun Lin
Dept. of Emergency Medicine
University of Illinois at Chicago
jlin7@uic.edu

Ugo Buy
Dept. of Computer Science
University of Illinois at Chicago
buy@cs.uic.edu

Rupa Sampath
University of Chicago Hospitals
Rupa.Sampath@uchospitals.edu

***Abstract*— While the use of information technologies is becoming increasingly widespread in healthcare organizations, such as large hospitals, to date these organizations lack unified information systems providing a comprehensive view of the organization's state. We define a technique for building formal models capturing the state of hospital departments and the interactions among departments during hospital operations. These models, based on Petri nets, will support a variety of management and decision-support tools, such as statistical analysis, simulation, and dynamic reconfiguration. These models are at the core of a new software prototype that we are developing for the University of Illinois Medical Center at Chicago.**

I. INTRODUCTION

We present a method for creating comprehensive formal models of workflows within a large healthcare delivery facility, such as a major hospital, using Petri nets. Our models integrate all relevant aspects of hospital operations including the status and availability of various classes of hospital personnel, the history and status of all patients currently in the hospital, and the availability of a broad range of healthcare resources (e.g., operating rooms, hospital beds, radiographic equipment, and space in various hospital departments).

The work reported here is part of a project aimed at creating a comprehensive model of the University of Illinois Hospital (UIH). The project's goal is to develop a management and decision-support system for the UIH, a large academic urban tertiary hospital. Here we show how Petri nets (PNs) can be used as a tool for information management by integrating PNs with the different software programs and databases currently in use at the UIH. The results presented in this paper are based on a detailed study and collaboration between an engineering group and members of the information services and medical personnel at the University of Illinois at Chicago. Here we show some of the models that we developed and we describe our approach for model construction, which is broadly applicable to a large class of healthcare facilities within the USA and internationally.

We use Petri nets as our models for two reasons. First, Petri nets have long been recognized to be quite well-suited for workflow modeling in general and for hospital workflows in particular. To date, excellent methods exist for mining Petri nets, for instance, from patient workflow logs [4], [8]. Second, Petri nets have been studied extensively, resulting in a large body of knowledge and software tools for analysis [1]. We use the existing knowledge when mining workflows from patient logs [10] and building optimal recofinguration [7] algorithms in response to changes in service demands and resource availability in a hospital.

When modeling hospital operations, the definition of an appropriate abstraction level is crucial for successful application of the models. An important choice is whether to model each individual patient or to cluster patients with similar symptoms within a patient class. Similarly, we must decide whether to model each resource, whether a human or inanimate resource, or to aggregate resources of each kind into classes. We chose to model each invididual patient in the hospital at any given time, but to cluster resources by resource classes. On the one hand, the workflow of a patient is more unpredictable than that of a resource. Also, a patient's workflow may involve many events of different kinds, resulting in relatively long event logs. Thus, we use a separate subnet for each patient. On the other hand, resources belonging to the same class (e.g., emergency room nurse) exhibit similar behaviors. For each resource class we need to model the number of resources in a given status (e.g., busy, idle and unavailable). Our strategy of modeling single patients and resource classes results in a high modeling accuracy while avoiding unnecessarily large models.

Other works have used various kinds of Petri nets to model specific aspects of healthcare systems. (See, e.g., [4], [6].) Most of those works have modeled individual hospital units without providing a collective view that shows the interactions among units and the magnitude of the resulting information exchanges. An additional limitation is that those models are based on static hospital representations. For example, a PN model of the Emergency Department (ED) workflow developed on the assumption that all resources associated with the ED will always be available is not realistic. In practice, resources sometimes become unavailable. Even the few models that do consider this possibility do not address how a PN model can be updated based on real-time information. Finally, existing models were generally not implemented, let alone integrated with existing hospital information systems.

Our method for generating comprehensive models of hospital operations starts by defining Petri net models of four key hospital components: (1) patient workflows, (2) human resource classes, (3) inanimate resource classes, and (4) hospital functions. Hospital functions capture activities that can be performed by different kinds of resources. For instance, a patient on a stretcher could be moved between hospital rooms by a transport, a registered nurse (RN), or a physician. We defined models for resource classes and for hospital functions based on their behavior. The generation of models for patient workflows is much more difficult because in general hospitals do not have such models. Fortunately, hospitals keep, for each patient, a log of activities performed on the patient. In addition, there are many techniques for mining workflows from activity logs [8], [9], [10]. The result of the mining activity is a set of PNs for each kind of patient, based on the patient's symptoms.

The generation of a comprehensive hospital workflow model is built upon patient workflows, which are created and deleted as patients enter and exit the hospital. When a patient enters the hospital, an instance of the workflow matching the patient's symptoms is added to the overall hospital model. Next, the resources and functions required for the activities contained in the workflow are connected to the patient's workflow. Finally, the status of the patient's workflow is updated as the patient receives treatment.

This paper is organized as follows. Section II discusses Petri nets in general and the special case of nets obtained from workflow logs. Section III describes the UIH. Our method for modeling resources and patients in the center is discussed in Section IV. Section V reports on the creation of integrated hospital models.

II. WORKFLOW NETS

A generalized Petri net is a 5-tuple $\mathscr{N} = (P,T,F,W,M_0)$, where P is a finite set of places, T is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs connecting the places and transitions, $W : F \rightarrow \mathbb{N}$ is a weight function of the arcs, and $M_0 : P \rightarrow \mathbb{N}$ is called the *initial marking* of $\mathscr{N}$ [5]. Symbol $\mathbb{N}$ denotes the set of nonnegative integers. In brief, a Petri net is a directed graph containing two kinds of nodes, places (circles) and transitions (rectangles), connected by directed arcs. A net state (aka marking) assigns a nonnegative number of so-called tokens to each place. The input (output) set of a node n is denoted by $^\bullet n$ ($n^\bullet$). A transition $t \in T$ is enabled in a state M_1 if $\forall p \in {^\bullet t} : M_1(p) \geq W(p,t)$, where $M_1(p)$ denotes the number tokens in place p for a given state M_1. If t is enabled in M_1, t can fire resulting in a state M_2 defined by $\forall p \in P : M_2(p) = M_1(p) - W(p,t) + W(t,p)$. In short, a transition may fire when a sufficient number tokens is present in its input place(s). When the transition is fired, tokens are consumed from its input place(s) and produced for its output place(s).

We also report some key definitions of workflow management as applied to a healthcare system. An *operation* or *task* is a piece of work to be done with one or more resources in a pre-determined time interval. A task is atomic, meaning that it cannot be split into smaller tasks. Examples include taking a CT scan and transporting a patient. A *resource* can perform tasks (on its own or together with other resources). A *resource class* is a set of resources of the same type. A hospital *unit* (department) has a set of resources that perform certain operations in the unit. Resources include both human and inanimate (i.e., non-human) resources. In the case of an X-ray technician taking the X-ray of a patient in the X-ray unit, taking the X-ray is the operation, whilst X-ray technician and the X-ray equipment are the resources. The X-ray technician belongs to the class of X-ray technicians, and the X-ray unit is the unit.

Let A be a set of observable events of a workflow $\mathscr{W}$. A workflow log $\mathscr{L}$ is a set of finite sequences over A, with each string $\mathscr{L} \in A^*$. Given a workflow log $\mathscr{L}$, we define the *workflow net* of $\mathscr{L}$ to be a Petri net $\mathscr{N} = (P,T,F,W,m_0)$ whose transition set T contains $\mathscr{L}$'s event set A. Ideally, all strings in $\mathscr{L}$ can be generated as firing sequences of $\mathscr{N}$. In addition, $\mathscr{N}$ should not generate any firing sequences that do not correspond to valid event sequences in the underlying workflow $\mathscr{W}$. The problem of deriving a workflow net from a workflow log is quite challenging; this problem is sometimes known as *workflow mining*. In general, the log must satisfy given completeness properties in order for workflow mining to be successful. Van der Aalst et al. defined reasonably permissive completeness criteria for workflow logs and an efficient method for generating workflow nets from complete logs [8]. Van der Aalst et al. subsequently refined their technique and implemented the ProM toolset, which gave excellent empirical results [9], [10].

For space reasons, we do not report Van der Aalst et al.'s method here. We just report a simple example showing how Petri nets can model a workflow log consisting of the following three event sequences, for a patient with an initial symptom of right lower quadrant abdominal pain:

1) ED, CT scan, SC, OR, SF, home.
2) ED, CT scan, SC, OR, SF, SICU, SF, home.
3) ED, CT scan, SC, OR, SF, SICU, death.

In the above, ED denotes admission to the hospital's Emergency Department, SC means surgical consultation, OR means operating room, SF means surgical floor, and SICU means surgical intensive care unit. A workflow net capturing the above workflow log appears in Figure 1. Transitions *ED, CT scan, SC, OR, SF, home, SICU, and death* capture the corresponding events in the workflow log. The first five transitions corresponding to a common prefix consisting of five initial events in all three sequences in the log. After the *SF* transition is fired, transition *dummy* captures a nondeterministic choice between a patient being sent to the SICU or discharged from the hospital. The firing of the latter transition results in an additional choice between transitions *SF* and *death*.

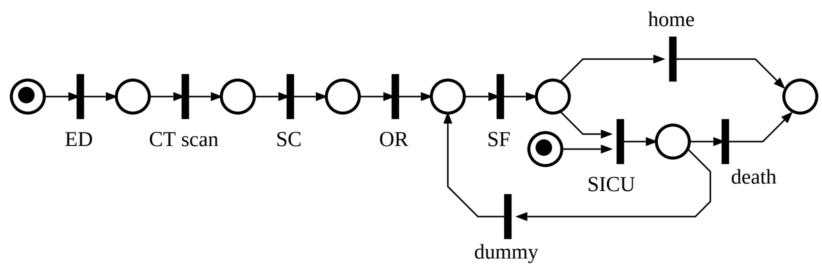

Fig. 1. A simple workflow net.

III. SYSTEM DESCRIPTION AND MODELING APPROACH

The University of Illinois Medical Center at Chicago (UIMCC) is an urban medical center supporting one of the largest medical schools in the USA. The center includes a 500-bed hospital in addition to over 40 primary care and specialty clinics. It has roughly 20,000 yearly admissions, of which approximately 40% come through the Emergency Department (ED). The ED, the ambulatory center and the hospital are busy with a relatively high occupancy rate in the hospital. The medical center has a large single site ambulatory center, inpatient and ambulatory radiology suites, a surgical area that maintains surgical services for both same day ambulatory and inpatient procedures. The UIMCC is a major referral center, attracting cases from around the world as well as from the USA Midwest region.

The hospital has 54 units. Every unit in the hospital is assigned a set of resource classes and shares some resources with other units. The hospital has the following classes of resources: (1) Human resources (e.g., physicians, physician assistants, nurses, and transports), (2) Equipment resources (e.g., CT scanners, X-ray machines, vitals monitors), and (3) Facility resources (e.g., operating rooms, recovery room, labs, and general beds).

The UIMCC utilizes a commercially available Electronic Medical Record (EMR) (Millennium®, Cerner Corporation) as its repository of patient-specific information. This system tracks orders and clinical data along with demographic and location information. However, the EMR does not have information about different UIMCC resources and their status.

The UIMCC uses additional computer systems for several functions including, among others, billing, patient admissions, nurse staffing, financial decision support and ambulatory patient scheduling. Although certain interfaces among electronic systems exist, there is no global integration of all the systems that contain patient specific, administrative and financial data. Consequently, there is no centralized data repository capturing the state of the medical center at any given point in time.

The lack of a central, integrated Information System (IS) adversely affects the decision-making process in different ways. The IS cannot provide a snapshot of resources available in the different hospital units at a given time. Any resource assignment decision can affect the rest of the workflow as tasks can require the same resources (e.g., transports and gurneys). Any effective decision-support system must have all the relevant information to evaluate feasible alternative strategies for meeting a given goal. The current collection of disparate databases does not provide ideal decision support. A centralized database could provide support for real-time resource assignment decisions. Also, as resource assignment decisions are not done in a fully informed manner, it is more difficult to make sound strategic resource management plans. Operations research and simulation tools could be used effectively once a centralized IS is created.

Our approach to modeling the UIH consists of two phases. First, we build models of patient workflows and resource classes. Next, we integrate such models into a comprehensive model of the entire center. Patient workflow modeling uses patient logs and the workflow mining method that we mentioned earlier [10]. Resource modeling uses two types of resources, namely Human Resources (HR) and non-human resources (NHR, aka inanimate resources). The latter category groups together equipment resources and facility resources. A difference between human and non-human resources is that NHRs can only perform one function, whereas HRs can perform multiple functions, as we explain below.

HRs can be on-site or on-call. The HRs performing the same set of tasks are grouped in a resource class. Hence the set of all RNs who work in the OR belong to the HR class called OR RN. The resources are identified by their class and not individually. For example, a RN can belong to one of the following classes: OR RN, ED RN and clinical RN. Every HR class performs at least one function. For example, the function of the transport resource class is to transport patients. Here “transport” is the resource class and “transporting” is its function. Some HR classes can perform more than one function. For example, the classes of clinical Medical Doctor (MD), clinical RN, ED MD, and ED RN could all perform the function of transporting a patient.

The set of resource classes (and resources) of a hospital is fixed. All resources are in one of three possible states: (1) Available; (2) Assigned; and (3) Unavailable. They can undergo a change of state. The HR classification is summarized as follows:

1) Skills: The resource classes that can perform only one function and those that can perform multiple functions.
2) Department affiliation: Information about the HR department affiliations.

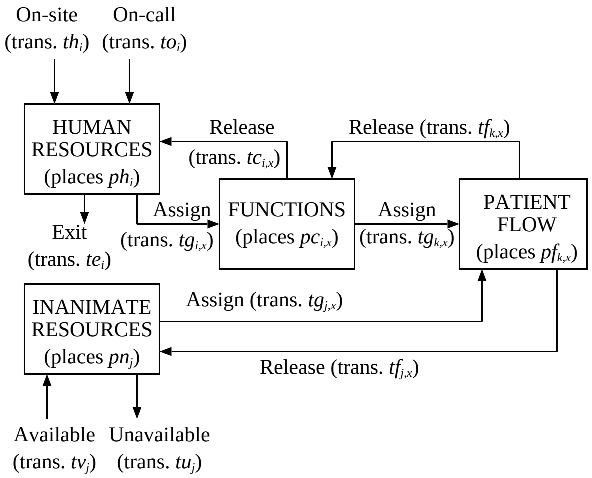

Fig. 2. Abstact view of our model.

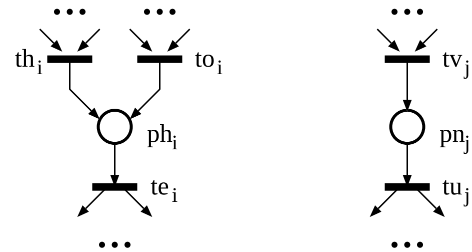

Fig. 3. Subnets for modeling a human and non-human resource classes.

3) Presence: Being on-site or on-call.

At any given time, the number of resources that can perform a certain function depends on the number of available resources of different resource classes that can perform the same function. For example, if there are 2 ED MDs and 2 transports (unassigned) in a hospital at a specific time, the number of available resources for the transport function is 4.

Non-human resources do not have functions since no two NHR classes can do the same tasks; they are directly assigned to a task based on their resource class. Figure 2 explains the resource allocation strategy of our model. Rectangles represent the four major model components. Arcs represent information flows among model components. The figure shows also the names that label places and transitions in the model. For instance, places in the subnets modeling patient workflows will be denoted by the label pf_i. Likewise, places in the subnets modeling human resources, non-human resources, and functions will be labeled by ph_i, pn_i, and pc_i. Transitions modeling the flow of information between model components will be labeled by th_i, to_i, tv_i, tu_i, tc_i, td_i, and tf_i. Next, we discuss the definition of PN models of resources and functions.

IV. Resource Modeling

Resource subnets are Petri nets modeling the different human and non-human resources of the UIH. For each HR class H_i with $i \in [1,h]$, where h is the total number of HR classes, we define a place ph_i in the HR component shown in Figure 2. At any time, the number of tokens in ph_i models the number of individual resources of class H_i that are currently available and unassigned. For example, if the resource class "OR RN" represented by place ph_{ORRN} the HR subnet has five tokens, it means that 5 OR RNs are available (free) to be assigned to patients in the hospital at that time.

Every ph_i place uses two input transitions and one output transition that model HR arrival and exit from the hospital. Input transitions th_i and to_i represent the arrival of on-site and on-call HRs to the hospital. Output transition te_i represents the exit of either of these resources from the center. Thus, ph_i represents both on-site and on-call resources of a HR class H_i. The Petri net representation of HR arrival and exit is shown on the left-hand side of Figure 3. Tokens in place ph_i represent available instances of the resource class. Transitions th_i, to_i and te_i are linked to ph_i as per our previous discussion. Place ph_i has additional input and output transitions, not shown in the figure, that model assignment and release of HRs belonging to class H_i. Since the number of different resource classes of a hospital is fixed, the size of the HR subnet (modeling all HR classes) is constant.

As with HRs, every NHR in the hospital is associated with a resource class. Given NHR class NH_j with $j \in [1,n]$, where n is the total number of NHR classes, a place pn_j represents NH_j in the NHR component shown in Figure 2. At any given time, the number of tokens in pn_j indicates the number of resources of the corresponding class that are unassigned and available for tasks requiring those resources. Every place pn_j has an input transition and an output transition corresponding to NHR availability/unavailability.

The PN representation of NHR information is shown on the right-hand side of Figure 3. Tokens in place pn_j represent available instances of the resource class. Transitions tv_j, and tu_j model the availability or lack of availability of resource instances. Whenever a new NHR of this class is added to the available pool of NHR (such as by repairing an existing resource or buying anew), transition tv_j is fired and a token is added to pn_j. If an instance of the NHR becomes unavailable due to failure or the equipment becoming obsolete, transition tu_j is fired and a token is removed from pn_j. Place pn_j has additional in and out arcs modeling the assignment and release of NHR instances on behalf of hospital operations requiring those resources.

Finally, a function subnet represents the different functions performed in the hospital and the way human resources are assigned to tasks based on the function that those resources can perform. Figure 4 shows the subnet template for an HR class. Function place $pc_{i,x}$ is defined if HR class H_i can perform function F_x. For each such place $pc_{i,x}$ a transition $tg_{i,x}$ is defined such that $tg_{i,x} \in {}^\bullet pc_{i,x}$ and $tg_{i,x} \in ph_i{}^\bullet$, where ph_i is the place modeling HR class H_i. Also, for every place $pc_{i,x}$ a transition $tc_{i,x}$ is defined such that $tc_{i,x} \in pc_{i,x}{}^\bullet$ and $tc_{i,x} \in {}^\bullet ph_i$. In addition, place $pc_{i,x}$ will be connected to flow transitions contained in the subnets modeling patient

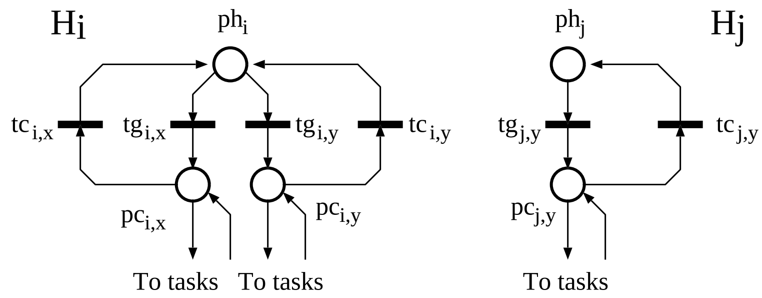

Fig. 4. Subnets for modeling functions performed by human resource classes.

workflows. Arcs connecting to flow transitions are shown by the dangling arcs to the tasks in Figure 4.

Figure 4 illustrates an example of a function net for two functions. In the figure, there are two HR classes, H_i and H_j. Suppose that two functions, F_x and F_y are performed in the system. HR class H_i, represented by place ph_i, can perform both functions, whereas H_j can perform only function F_y. Thus, when F_y is needed either of the two HR classes can be used. In this case either of the two assignment transitions $tg_{i,y}$ and $tg_{j,y}$ can be fired, assuming that instances of both HR classes are available. Such situations often happen in hospital workflows. Consider, for instance, the task of moving a patient within the hospital. This task could be performed by multiple HRs, such physician assistant, RN and transport. If multiple HRs are available, a decision must be made about which HR class can be assigned to the function. Since transitions *tg* represent resource assignment decisions, these transitions are called *decision* or *assignment* transitions.

An important feature of hospital resource modeling is resource proportions. In a hospital workflow, for example, a given task might require 0.5 OR RN while another task could require 0.75 OR RN. Such proportions can be modeled using Petri nets by taking the least common multiple for 0.5 and 0.75, which is 0.25. When this is necessary, in our models we use a single token to model the resource quantity corresponding to the least common multiple. In this example, a token would correspond to 0.25 of an OR RN time; the full assignment of an OR RN's time to a task would be modeled by an arc with weight 4 feeding into the transition modeling the task.

V. Integrated Model and Analysis Supported

The unified Petri net model of the UIH integrates subnets for the resources, functions and patient workflows in the center. The unified model is build around patient workflows. When a patient C_i arrives at the hospital with an initial symptom, a patient flow subnet is generated for the patient. Next, the specific resources and functions required for the patient flow subnet are identified. The resource classes are first assigned to their functions by the resource assignment transitions $tg_{i,x}$. The function subnet provides the required human resources to tasks of the patient flow by connecting arcs from the function places $pc_{i,x}$ to the patient's flow transitions. Once a task is complete, the corresponding task completion transition tf_x is enabled in the patient flow. Firing of this transition returns the resource to its function F_x, from which it is further returned to the appropriate HR class through transition $tc_{i,x}$. Patient flows return resources to NHR classes through transitions *tf*. Tokens are moved to reflect the progress of all patients in the hospital in their workflows. The size of the unified model is bounded by the total number of patients that can be in hospital at any point in time. The state in which the patient limit is reached in a given hospital is usually called the "by-pass state" of the hospital. In the UIH, it is estimated that the by-pass state is reached when the hospital has a 90% occupancy rate, or about 450 patients. Figure 5 shows a portion of the unified model of the UIH, for the case of a patient being admitted to the center through the ED while complaining of lower right quadrant abdominal pain.

An implementation of a prototype for maintaining and updating the unified model of the UIH is currently being planned. The prototype will interface with Millennium® and other systems at the UIH in order to keep the state of the unified model up-to-date with events occurring at the center. This kind of model will support various kinds of analyses, such as dynamic reconfiguration of hospital resources in response to sudden changes in service requirements or resource availability. We have developed an optimal technique for performing such a reconfiguration and we applied that technique to dynamic reconfiguration of healthcare facilities [7]. Additional potential applications include business planning, forecasting, and optimization of resource use within the hospital.

The model can also be used for simulation purposes. One application of simulation is to perform different stress tests of hospital response in extreme scenarios such as mass-casualty situations in the aftermath of major disasters. For example, appropriate timed extensions of our models can generate a large magnitude of virtual patients (victims) and measure the hospital's response time in serving these patients. Also, because the hospital resources are explicitly modeled in the PN, one can use the simulation of this PN as a decision-support tool in determining the best combination of resources that can provide a reasonable response to a simulated mass-casualty event. An additional application of the model is in hospital operation scheduling and resource assignment.

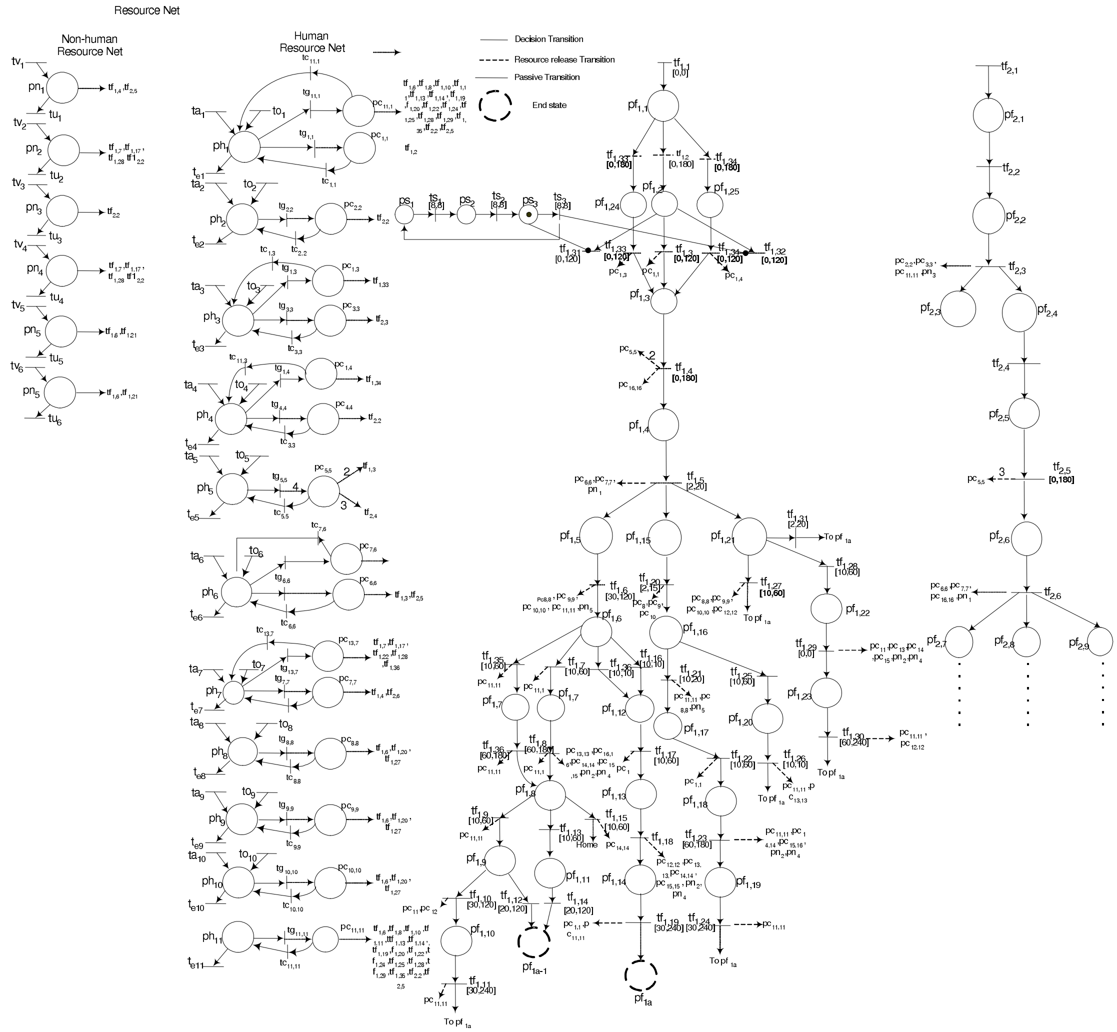

Fig. 5. Portion of unified Petri-net model of University of Illinois Medical Center at Chicago.

Existing methods can be used for this purpose [2], [3].

VI. Conclusions

We presented the key concepts underlying a new method for modeling complex healthcare delivery organizations, such as a large hospital. This work is part of a long-term project seeking to create one such model for the University of Illinois Hospital. Our goal is to monitor in real-time the status of each patient in the hospital, the availability of medical personnel, and the state of hospital equipment and facilities. Our integrated models will be amenable to a variety of analysis methods, including dynamic reconfiguration and simulation, which will provide improved decision-support and management tools than current practice.

References

[1] C. Girault and R. Valk. *Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications*. Springer-Verlag, Secaucus, NJ, 2001.

[2] M. Haji, L. Agostoni, P. Gullo, H. Darabi, and M. Mancini. An improved multi-mode resource allocation and project scheduling model. In *Proc. IEEE Int. Conf. on Automation Science and Engineering (CASE 2008)*, pages 365–370, Arlington, VA, 2008.

[3] M. Haji and H. Darabi. Petri net based supervisory conrol reconfiguration of project management systems. In *Proc. IEEE Int. Conf. on Automation Science and Engineering (CASE 2007)*, pages 460–465, Scottsdale, AZ, 2007.

[4] L. Maruster, W. van der Aalst, A. Weijters, A. van den Bosch, and W. Daelemans. Automated discovery of workflow models from hospital data. In B. Kröse and M. v. S. M. de Rijke, G. Schreiber, editors, *Proc. 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pages 183–190, 2001.

[5] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, Apr. 1989.

[6] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, and C. Mossa. Guideline-based careflow systems. *Artificial Intelligence in Medicine*, 20(1):5– 22, 2000.

[7] R. Sampath, H. Darabi, U. Buy, and J. Liu. Control reconfiguration of discrete event systems with dynamic control specifications. *IEEE Trans. on Automation Science and Engineering*, 5(1):84–100, 2008.

[8] W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[9] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. The ProM framework: A new era in process mining tool support. In *Application and Theory of Petri Nets*, Lecture Notes in Computer Science 3536, pages 444–454, Berlin, Germany, 2005. Springer-Verlag.

[10] L. Wen, W. M. P. van der Aalst, J. Wang, and J. Sun. Mining process models with non-free-choice constructs. *Data Mining and Knowledge Discovery*, 2007.