Modeling and Analyzing Biomedical Processes using Workflow/Petri Net Models and Tools

Mor Peleg[a], Samson Tu[b], Abhijit Manindroo[c], and Russ B. Altman[b]

[a]Department of Management Information Systems, University of Haifa, Haifa, Israel
[b]Stanford Medical Informatics, and [c]Department of Electrical Engineering, Stanford University, Stanford, CA, USA

Abstract

Computer simulation enables system developers to execute a model of an actual or theoretical system on a computer and analyze the execution output. We have been exploring the use of Petri Net (PN) tools to study the behavior of systems that are represented using three kinds of biomedical models: a biological workflow model used to represent biological processes, and two different computer-interpretable models of health care processes that are derived from clinical guidelines. We developed and implemented software that maps the three models into a single underlying process model (workflow), which is then converted into PNs in formats that are readable by several PN simulation and analysis tools. We show how these analysis tools enabled us to simulate and study the behavior of two biomedical systems: a Malaria parasite invading a host cell, and patients undergoing management of chronic cough.

Keywords:

Workflow, Petri Net, simulation, Process model, Clinical guidelines, biological systems

Introduction

Modeling is an important step in system development. It involves the creation of a simplified, explicit representation of a complex system. Creating a model facilitates communication between system developers and users, increases the understanding of the system under development, and enables simulation. Simulation involves execution of a model under hypothetical conditions on a computer and analysis of the execution output. Simulation of biomedical systems is potentially very useful. For example, the behavior of safety-critical medical systems can be analyzed and tested before they are actually put into use. Analysis of medical decision-support systems can identify (1) bottlenecks that impede patient management, (2) underutilization of resources, and (3) erroneous flow control. While a full understanding of biological systems at a molecular level is often beyond reach, a system can be modeled, using incomplete data, to an extent that it produces behavior that can be compared with experimental measurements.

Many approaches exist for modeling biomedical systems. Some of the approaches can be used to represent any domain of interest, whereas other approaches have constructs that are specifically tailored for modeling biological or medical domains (e.g., the domain of clinical guidelines) [1, 2]. Few of these approaches have tools facilitating the study and analysis of the dynamic behavior of the system. This paper reports the preliminary results in translating biological and guideline-based care processes to PN formalisms and using PN-based tools to study the behavior of these apparently dissimilar processes.

Models for studying system dynamics

Petri Net [3] is one of the most widely used methodologies for representing concurrent, discrete-event dynamic systems in a way that enables simulation. A *PN* is a directed, bipartite graph in which nodes are either places or transitions, where *places* represent Boolean conditions (e.g., parasite in blood stream) and *transitions* represent activities (e.g., invasion of host cell). *Tokens* that are placed in places signify that the condition that represents the place holds. The placement of tokens in the net, called marking, defines the net’s state. A PN can be simulated by moving tokens according to a firing rule; when all the places with arcs leading to a transition have a token, the transition is enabled, and may fire, by removing a token from each input place and adding a token to each output place. The results of the simulation can be plotted as graphs, or otherwise analyzed.

PNs can also be structurally examined to verify desired system properties, such as boundedness and liveness. *Boundedness* guarantees that in every place of the net, the number of tokens is always less than some finite number (e.g., no toxic accumulation of metabolites, where each metabolite molecule is represented by one token). *Liveness* guarantees that all transitions (biomedical processes) can be enabled. Another type of analysis involves determining whether we can move from one state of the system (e.g., parasite inside host liver cell) to another state (e.g., parasite cleared).

PNs have been applied mostly in manufacturing and safety-critical systems. In the biomedical domain, PNs and their extensions have been used for analysis of pathways [4] and gene regulatory networks [5]. We have developed a methodology for representing the structural, functional, and dynamic aspects of biological systems [1, 6]. We refer to this methodology as *Bio-Workflow (BioWf).* BioWf is based on a combination of two models: the Workflow (Wf) model of the Wf Management Coalition (WfMC) and a biological concept model. The *Wf model* [7] consists of a Wf Process, (WfP) representing nesting and ordering of processes (activities), and an Organizational Model, representing the structural components that participate in the processes (e.g., enzyme, substrate), and the functional roles that they play (e.g., phosporylation). A *WfP* is a network of activities that are connected to each other using *Transitions*. Each activity has *Transition Restrictions* defining whether its preceding and subsequent activities are executed in parallel (*AND split or join*) or are mutually exclusive (*XOR split or join*). The Wf model distinguishes between different kinds of activities: *Applications*, which are atomic processes, *Subflows*, which are high-level processes that are decomposed into other processes, and *Route Activities*, which are used for branching and synchronizing control flow. The biological concept model provides a framework for describing biological entities. It incorporates parts of the Transparent Access to Multiple Biological Information Sources [8] and The Unified Medical Language System [9]. The WfP maps to PNs, allowing for verification of formal properties and qualitative simulation [10]. In prior work, we have used manual mapping to study the behavior of normal and abnormal protein translation [6].

Another domain in which process modeling is very useful is guideline-based clinical decision support. Many computer-interpretable guideline formats organize guidelines as care plans that unfold over time [2]. Typically, the care-plan components include clinical actions, decisions, patient-state steps, subguidelines, and component used for allowing parallel execution of other components. The guideline-formalisms link care-plan components in sequence, in parallel, and in iterative and cyclic structures, thus defining control-flow. In addition, all the models support nesting of plans, as well as expression of temporal constraints on plan components. Quaglini and colleagues had described a methodology where high-level representations of guideline-based care processes are translated directly into PN for simulation purpose [11]. As part of the Health Level 7 Clinical Decision Support Technical Committee, we have been evaluating the Wf model as a common control-flow model to which the various guideline formalisms could be mapped [7]. As well as being a suitable process model, the Wf model is a tested standard of the WfMC and has mathematical formal foundations, based on PNs [10].

In this paper, we present the algorithms that we developed and implemented to (1) map the BioWf model, and two guideline models (GLIF [12], SAGE [13]) to the WfP model, and (2) to translate the generic WfP model to a PN model. We discuss the biomedical questions that can be answered by simulating and analyzing PNs, and present the results of simulation and analysis of biological systems and medical systems.

Methods

We created the source ontologies of the BioWf and the two guideline-models and their instances using Protégé-2000 [14]. We created the generic WfP and the PN ontologies in Protégé.

Mapping process models to WfP models

In order to convert instances of the biomedical models (ontologies) into instances of WfP classes, we used mapping tools of Protege-2000 that mediate knowledge and data between any two knowledge-based application components [15]. We used the Mapping Editor to create instances of an ontology of mapping relations between the GLIF ontology (source) and the WfP model (target). The mapping relations that we defined for the GLIF ontology are shown in Table 1. Since the BioWf model and the process model in SAGE were derived from the WfP model, the mappings that we used to translate them to the WfP model are trivial, and are hence not shown. We used the Mapping Interpreter, which processes a set of mapping relations defined for two knowledge models and migrates instances from one (source) model to the other (WfP).

Table 1. Mapping relations used to map instances of the GLIF ontology to the WfP model. Sync - Synchronization

Source class	Target class	Condition
Algorithm	Workflow Process	
Action Step or Decision Step	Application	No value for *detail* slot of source class
Action Step or Decision Step	Subflow	Source class has value for *detail* slot
Tasks of an Action Step	Workflow Applications	
Branch Step	Route Activity is AND-split	
Synch Step (SS)	Route Activity	
SS.continuation	AND-join	continuation includes ‘AND’ or is of the form >=n
SS.continuation	XOR-join	continuation includes ‘XOR’ or is of the form >=1
Patient State Step	Route Activity	
Connector	Regular Transition	
Connector.next_step. description.specification	Regular Transition.condition	Connector.next_ step is-a Patient_ State_ Step

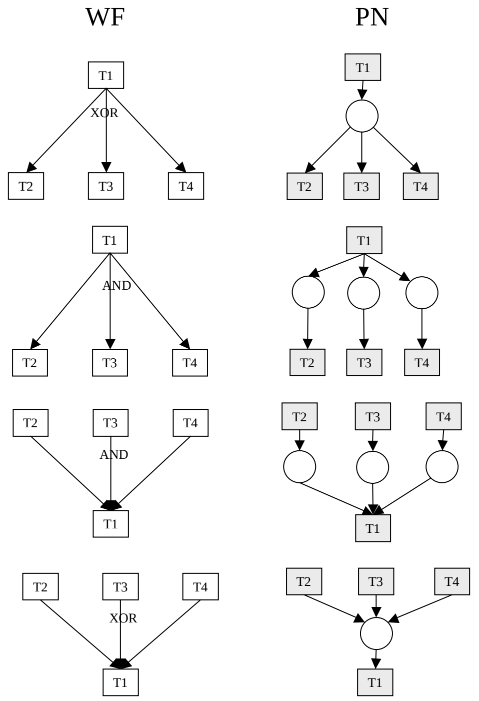
Converting Workflows to Petri Nets

We developed Java programs that convert generic WfPs into PNs. One program converts WfP instances in Protégé format into a generic PN in Protégé format. Its algorithm, adapted from previous work [10], is shown in Figure 1. Since the WfP model is hierarchical, we developed a second program that collapses the hierarchical net into a single PN by expanding hierarchical transitions. A third set of programs convert PNs from Protégé format to the formats of two PN tools: Woflan [16] and TimeNET [17].

Results

To evaluate our algorithms and tools, we converted several source models (3 in BioWf, 1 in GLIF, and 1 in SAGE), containing 1-7 nested Process Diagrams each, into WfPs and into PNs in Woflan and TimeNET format. Figure 2 shows a BioWf Process Model showing a Malaria parasite invading a host cell, and Figure 3 shows the resulting PN, in TimeNET format. Using these tools, we checked the models' structural properties and studied its behavior.

1. Translate WF activities to PN transitions
2. Translate WF Processes to PNs and populate PN transitions
3. Create places in PNs; connect them to transitions; add *i* places between PN transitions that correspond to WF activities that are linked (*i* depends on WF transition restrictions):



4. For top-level net: add first and last places
5. Link PN transitions that correspond to Subflows to their lower-level PNs

Figure 1. Algorithm for converting a WfP into a PN. Circles - PN transitions; White rectangles - WF activities; gray rectangles - PN transitions.

Verification of structural properties

- Is the model's process definition correct?

Woflan identified an error in one of our BioWf models, where an activity was connected to the wrong successive activity, leading to an incorrect process definition with two start places.

- Can all the modeled processes be executed?

Confirmed by Woflan (live net)

- Can there be an unbounded accumulation of items (patients, cells, metabolites)?

Woflan confirmed that the model is bounded.

- How many states can the system be in?

This is a measure of system complexity. TimeNET identified 3003 states in the PN of Figure 3.

- Can we identify possible pathways by discovering preservation rules for amounts of tokens in sets of system stages (places)?

This method could be used to discover pathways in which a key metabolite is modified or transported. The set of places that correspond to the key metabolite is a *place invariant* (the sum of tokens in the set of places is constant). Only trivial results were obtained for our examples (by TimeNET)

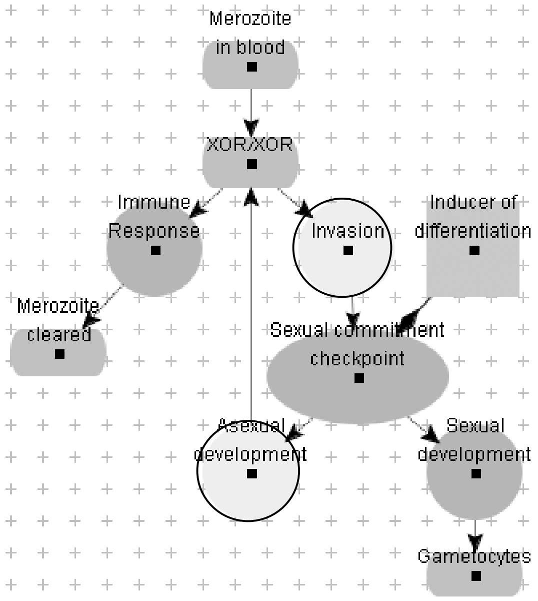


Figure 2. A BioWf showing Malaria parasites (Merozoites) invading host erythrocytes. The parasites can be cleared from the blood stream by an immune response, or invade a Red Blood Cell (RBC). Following invasion, they follow sexual development to produce gametocytes, or undergo asexual development that results in merozoites in the blood stream. Round Rectangles - Route Activities; ovals with a bold contour - High-Level Processes; ovals - Low-level Processes, Square - participant.

Studying System Behavior

TimeNET helped us to answer the following questions about the dynamic behavior of systems:

- Are there any bottlenecks?

Bottlenecks are resources or transitions that slow down an entire process. Examples include patients waiting for tests to be performed, or metabolites that accumulate before a rate-limiting step in a pathway. Figure 4 shows two plots that are characteristic of bottlenecks: R2 and R3, which measure parasites that invaded a Red Blood Cell (RBC), committed to asexual development (P5) and or to sexual development (R7). In these examples, the bottlenecks were caused simply by the fact that the transitions that follow the bottleneck places take time. Other reasons for bottlenecks include limited resources that transitions must wait to become available, and waiting to synchronize with concurrent paths. Bottlenecks can be shortened by changing parameters that affect transition rate or token availability, or by changing PN's structure.

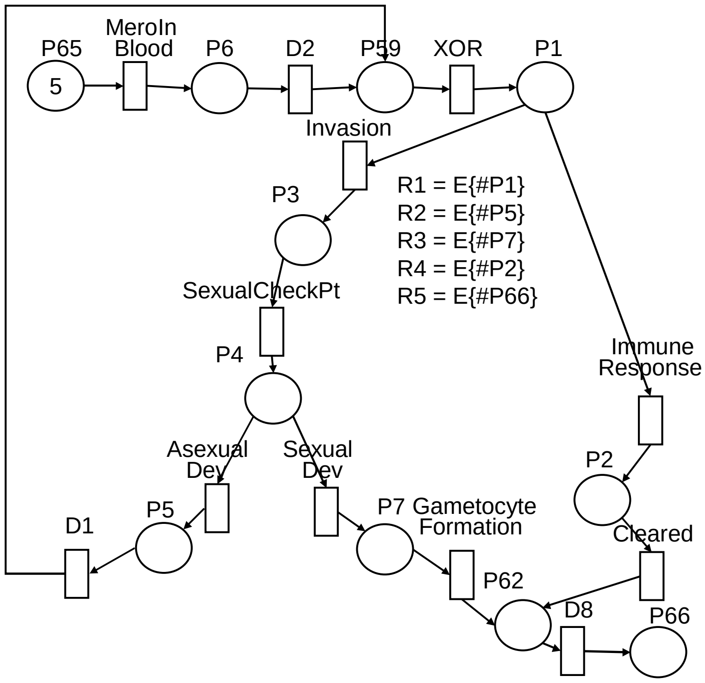


Figure 3. The PN resulting from the conversion algorithm, in TimeNET format. Rectangles - transitions and circles - places. Dummy transitions (D) and places are created by the algorithm and do not have correspondent nodes in the Wf. Therefore, their names are meaningless.'5' written in the start place represents initial marking. R – performance measure

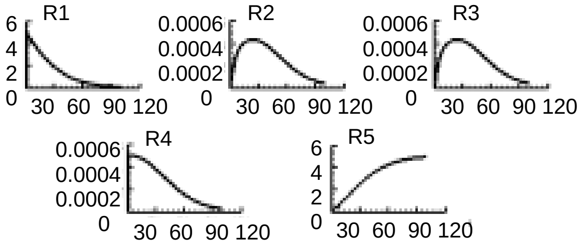


Figure 4. Graphs displaying the kinetics of the performance measures defined in Figure 3. The y-axis shows the values of the performance measures (average number of tokens at certain places) from time zero to a given time on the x-axis.

- How long does it take half of the tokens (e.g., patient, parasite) in the starting place to reach a specified end state?

Plot R5 in Figure 4 shows that 50% of the tokens reach the end state within 40 time units.

- What is the kinetic profile of the system: how many tokens (e.g., parasites, patients) are there at a certain stage (place) of the process, over time?

We can observe the behavior of one stage or a set of stages (summing the tokens in a set of places). Figure 4 shows different kinetic profiles:

R5 – a hyperbolic curve showing a sink place (P66) that accumulates all of the tokens in the system

R2, R3 – parabolic curves showing transient stages. Tokens gradually enter these stages, stay there for a while and then transition to the next stage

R1, R4 – curves showing stages that starts with many tokens and loose them over time

- At some time after the start of the "experiment" (execution of the net), what is the proportion of tokens (e.g., patient, parasite) in each state?

By comparing the kinetic profiles of different places over time, we can see the proportion and number of tokens that reach different places. This ratio depends on the length of the paths to each place, the durations of transitions on each path, and the relative probabilities of conflicting transitions.

- How does system behavior depend on parameter values?

In medical systems, we can change parameters in order to optimize performance measures, such as utilization of resources or average length of patient treatment. In biological systems, the purpose of modeling is not to improve a man-made system, but to reproduce the behavior of an actual system, thereby gaining confidence that the model is complete. Changing parameters in a biological model is similar to changing conditions in in-vitro systems (e.g., pH, number of parasites released from a cell). Parameters that can be changed include: (1) the initial state of the system (marking), (2) the delay distribution functions of the timed transitions, (3) the relative probability of simultaneously enabled conflicting transitions, and (4) blocking a transition that corresponds to some biomedical process.

- Can we validate that control flow is correctly specified?

By playing the Token Game, a user interactively steps through the processes that are enabled and sequences of processes that are actually executed. This often reveals errors in control flow modeling.

- Does the system reach steady state, and what is the proportion of items at steady state?

Our PN examples are derived from WF processes, which have one final place. Correct WF definitions are those of terminating processes, where all tokens end in the final place. However, systems that exhibit continuous operation that achieve a steady state are common in biology. Although they are not sound Wfs they may be modeled by Wfs and PNs.

Discussion

Our work demonstrates the possibility of applying a common methodology and tools to further understanding of the dynamics of biological processes and clinical care processes. Our results are preliminary in so far as we are still in the process of evaluating different PN formalisms and tools. The Woflan and TimeNET tools were complementary in their functionality and enabled us to cover many interesting biomedical analysis. However, we encountered several shortcomings with the analysis that these tools provided. First, in the Woflan tool, only one token can be passed on an arc that connects transitions and places. This limits our ability to represent transitions that require or produce more than one token in a certain place. Examples include cell division and release of many parasites from an infected cell, which produce more than one token at a place, and metabolic reactions involving stoichiometric coefficients greater than 1.

TimeNET measures the number of tokens at a place, averaged from time zero till some time stamp. This measure is a smoothing function. The tool does not give the number of tokens at a particular time point. Another useful measure that cannot be assessed directly is the number of transition completions per time interval. This measure could be useful for assessing the utilization of a medical activity that is represented by a transition.

Reachability is very useful for answering questions about states (markings) that are reachable from a given initial state of the system. It is especially interesting to find conditions under which a state is not reachable (e.g., when a certain reaction is blocked). Although TimeNET has a function for checking reachability, this function did not work on our PN examples, not even smaller nets.

The mapping that we used to convert WfP to PNs disregards participants or resources that participate in activities (e.g., the number of technicians, amount of equipment, molecules involved in reactions). Instead, it concentrates only on the control flow among activities. This has shortcomings in modeling various biomedical systems. For example, the mapping is not suitable for quantitative analysis of metabolic pathways or the resource requirements of implementing guidelines. Other mappings can be developed to map participants into places of PNs. For example, the Careflow process developed at University of Pavia [10] maps clinical workflow processes into Colored PNs [18], where parameterized tokens are used to differentiate types of resources and participants. This PN formalism allows simulation of the effects of resource allocation in the clinical setting.

Acknowledgements

Supported in part by: (1) Grant LM06594 with support from the Department of the Army, Agency for Healthcare Research and Quality, and the National Library of Medicine (NLM), (2). NLKM grant LM05708 from National Library of Medicine, and (3) the Burroughs-Wellcome Fund.

References

1. Peleg M, Yeh I, Altman RB. Modeling biological processes using Workflow and Petri Net models. *Bioinformatics* 2002;18(6):825-837.

2. Peleg M, Tu SW, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *J Am Med Inf Assoc* 2003;10(1):52-68.

3. Girault C, Valk R. (eds). *Petri Nets for Systems Engineering*. Berlin Heidelberg: Springer-Verlag; 2003.

4. Reddy VN, Liebman MN, Mavrovouniotis ML. Qualitative Analysis of Biochemical Reaction Systems. *Comput Biol Med* 1996;26(1):9-24.

5. Matsuno H, Doi A. Hybrid Petri Net Representation of Gene Regulatory Network. In: Pacific Symposium on Biocomputing; 2000: 338-349.

6. Peleg M, Gabashvili IS, Altman RB. Qualitative models of molecular function: linking genetic polymorphisms of tRNA to their functional sequelae. *Proc IEEE* 2002;90(12):1875-1886.

7. Layna Fischer (Ed). Workflow Handbook: Future Strategies Inc., Lighthouse Point, FL, USA.; 2002.

8. Baker PG, Goble CA, Bechhofer S, Paton NW, Stevens R, Brass A. An ontology for bioinformatics applications. *Bioinformatics* 1999;15(6):510-520.

9. Lindberg C. The Unified Medical Language System (UMLS) of the National Library of Medicine. *J Am Med Rec Assoc* 1990;61(5):40-42.

10. van der Aalst WMP. The application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers* 1998;8(1):21-66.

11. Quaglini S, Stefanelli M, Cavallini A, Micieli G, Fassino C, Mossa C. Guideline-based careflow systems. *Artif Intell Med* 2000;20(1):5-22.

12. Peleg M, Boxwala A, Ogunyemi O, Zeng Q, Tu S, Lacson R, et al. GLIF3: The Evolution of a Guideline Representation Format. Proc. AMIA Symp; 2000: 645-649.

13. Tu SW, Campbell J, Musen MA. The structure of guideline recommendations: A synthesis. Proc AMIA Symp, in press. 2003.

14. Gennari J, Musen MA, Fergerson RW, Grosso WE, Crubezy M, Eriksson H, et al. The Evolution of Protege: An Environment for Knowledge-Based Systems Development. *International Journal of Human-Computer Interaction* 2002;58(1):89-123.

15. Crubezy M. Chapter 17. Ontologies and Problem-Solving Methods. In: S. Staab RSe, editor. *Handbook on Ontologies in Information Systems*: Springer; 2003.

16. Verbeek HMW, Basten T, van der Aalst WMP. Diagnosing Workflow Processes using Woflan. *The Computer Journal* 2001;44(4):246-279.

17. Zimmermann A. *TimeNET 3.0 User Manual*. Berlin: Institut fur Technische Informatik of the Technische Universitat Berlin; 2001.

18. Jensen K. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*: Springer Verlag; 1995.

Address for correspondence

Mor Peleg, Department of Management Information Systems, University of Haifa, Haifa, Israel, 31905