Modeling biological processes using Workflow and Petri Net models

Mor Peleg, Iwei Yeh, and Russ B. Altman[*]

Stanford Medical Informatics, Stanford University, Stanford, CA, 94305, USA

peleg@smi.stanford.edu, **yeh@smi.stanford.edu**, **russ.altman@stanford.edu**

Running title: *Modeling Biological Processes using Workflows*

Keywords: *ontology, knowledge representation, biological process, Workflow, Petri Net*

[*] To whom correspondence should be addressed

Abstract

Motivation: *Biological processes can be considered at many levels of detail, ranging from atomic mechanism to general processes such as cell division, cell adhesion or cell invasion. The experimental study of protein function and gene regulation typically provides information at many levels. The representation of hierarchical process knowledge in biology is therefore a major challenge for bioinformatics. To represent high-level processes in the context of their component functions, we have developed a graphical knowledge model for biological processes that supports methods for qualitative reasoning.*

Results: *We assessed eleven diverse models that were developed in the fields of software engineering, business, and biology, to evaluate their suitability for representing and simulating biological processes. Based on this assessment, we combined the best aspects of two models: Workflow/Petri Net and a biological concept model. The Workflow model can represent nesting and ordering of processes, the structural components that participate in the processes, and the roles that they play. It also maps to Petri Nets, which allow verification of formal properties and qualitative simulation. The biological concept model, TAMBIS, provides a framework for describing biological entities that can be mapped to the workflow model. We tested our model by representing malaria parasites invading host erythrocytes, and composed queries, in five general classes, to discover relationships among processes and structural components. We used reachability analysis to answer queries about the dynamic aspects of the model.*

Availability: *The model is available at* http://smi.stanford.edu/projects/helix/pubs/process-model/

Contact: *altman@smi.stanford.edu*

Introduction

With the increasing volume of genomic data available, it has become clear that biologists need computational methods for data organization and analysis. To develop computer applications that aid in this task, we first need a knowledge model that can represent biological systems. Such a model should represent the high-level physiological processes and connect them to molecular-level functions. The emergence of structured terminologies, such

as the Gene Ontology (The Gene Ontology Consortium, 2000) is an important first step in creating the infrastructure required for a biological knowledge model.

Our purpose was to develop a biological process model that could (1) consistently represent the dynamic knowledge about high-level biological processes, in the context of their component molecular-level sub-processes, and (2) be amenable to inference, verification of dynamic (control-flow) properties, and qualitative simulation. We hereby propose a set of requirements that are desirable for a biological process model that fulfills this purpose. First, the model should represent the following three views of a biological system.

(1) The **Static-structural** view of bio-molecular complexes, chemical, and biopolymers that participate in the system, their properties, and the relationships among them;
(2) The **Dynamic** view that shows how processes are ordered over time (control flow) and how a process is recursively broken down to component processes and reactions (atomic processes). The dynamic model should support sequential, parallel, conditional and iterative processes; and
(3) The **Functional** view that shows the actors (e.g., enzyme) that perform each function, the substrates (input) of each function, and the products of the function (outputs). The cellular location of the substrates and products should be specified.

Second, the model should include a biological ontology that will define biological concepts and arrange them in classification hierarchies. Ontologies provide consistent definitions and interpretations of biological concepts, and enable software applications to share and reuse the knowledge consistently (Gruber, 1995). Ontologies can be used to perform logical inference over the set of concepts to provide for generalization and explanation facilities (Schulze-Kremer, 1998).

Third, the representation should be intuitive. Biologists should find it easy to create and comprehend a system that is represented using the model.

Fourth, the model should be hierarchical to manage the complexity of the represented system.

Fifth, the model should be mathematically based to allow verification of properties that are desirable in biological systems, and simulation of system behavior. For example, we could verify that there is no toxic accumulation of metabolites in pathways (boundedness property). For a high-level biological process composed of lower-level processes, could verify that all

the component processes can participate (liveness property). We can also search for processes that are continuously operating (violate the fairness property) and may therefore be good targets for affecting system behavior. A formal model should also enable us to check whether we can move from one state of the system (e.g., parasite inside host liver cell) to another state (e.g., parasite cleared).

And sixth, the model should allow important inference capabilities. For example, proposing the consequences of knockout experiments – substrates that may accumulate, other reactions that might process these substrates, etc. Other examples include identifying processes that are triggered in response to environmental effects (e.g., heat shock), or finding processes that are active during a certain developmental phase. The reasoning mechanism should use the biological concept model to form abstractions of concepts (e.g., *tight-junction formation* is a kind of *adhesion* process).

The rest of the paper is organized as follows. First we present our analysis of related models, according to the set of desired properties that we defined. Then we present our model, which combines the best elements of two of the models that we compared. We demonstrate our model using the process of Malaria parasite invasion of host erythrocyte cells.

Analysis of related models

We evaluated eleven existing models that were developed in biology, business, and software engineering, in terms of the desired properties that we defined above. It is important to stress that properties that we defined are desirable with respect to our goal. The models that we analyzed were developed for different purposes, and therefore do not necessarily fulfill our goals. Table 1 shows a summary of the evaluation. We highlight some of our conclusions below.

Substantial work has been done to create **controlled vocabularies** that provide classification hierarchies of structural and functional biological-role categories (The Gene Ontology Consortium, 2000) (Frishman et al., 2001; Serres and Riley, 2000). These controlled vocabularies provide a categorization of biological structures, processes, and functions. Researchers can use these vocabularies to classify processes (*is-a* relationships) and list their sub-processes (*part-of* relationships). Researchers can also use the controlled vocabularies to

categorize gene products based on the cellular location in which they reside, the biological processes in which they participate, and the functional roles that they fulfill.

Transparent Access to Multiple Biological Information Sources (**TAMBIS**) (Baker et al., 1999) is an ontology for describing data to be obtained from bioinformatics sources. TAMBIS goes beyond taxonomic models. It represents biological concepts and forms a semantic network of concept relationships that can be used to make inferences from biological data. TAMBIS is defined using Description Logics (Bordiga, 1995) – a logic-based knowledge representation formalism that defines concepts in terms of their properties and uses reasoning to classify the concepts based upon those descriptions.

Controlled vocabularies and TAMBIS do not aim to fully model the functional aspect of a process, as they concentrate on the gene products that carry out a reaction but not on other substances that are involved in the process, such as substrates, products, and inhibitors. Likewise, these models chose not to represent dynamic aspects of processes (i.e., the ordering of sub-processes within higher-level processes, temporal relationships, the events that trigger a process occurrence, and the conditions that are necessary for the process to occur). This is because dynamic aspects were not relevant for classifying gene products and for accessing biological knowledge sources. Although order and temporal aspects could be represented in Description Logics, inference and computation on these aspects would be difficult.

EcoCyc (Karp, 2000) is a bio-ontology that represents a functional model for metabolic-reactions in E.Coli, using a frame-based formalism. It provides annotations and sequences of all *E. coli* genes. EcoCyc describes all known pathways of *E. coli* small-molecule metabolism. EcoCyc represents reactions by specifying their reactants and products. An enzymatic reaction represents a reaction that is carried out by a certain catalyst, and may have reaction modulators (i.e., activators and inhibitors) and enzyme cofactors. Pathways are represented as ordered lists of reactions with branch points. EcoCyc concentrates on representing metabolic reactions. It can also represent transport reactions, signal transduction, and control/regulation. EcoCyc does not attempt to represent all types of non-metabolic reactions (e.g., cellular movement) and high-level processes (e.g., signal transduction). Moreover, EcoCyc does not have a dynamic model that enables representing parallel processes, process triggering, and temporal constraints. EcoCyc's static-structural model (i.e., the ontology of chemicals) is not adequate for modeling all eukaryotic structures.

Rzhetsky and colleagues (Rzhetsky et al., 2000) presented a knowledge model for **regulatory networks**. As in EcoCyc, this model can be used to specify static-structural and functional aspects of regulatory networks. High-level processes can be categorized into a number of biological processes and can contain one or more actions. Each action is defined as having both a biochemical (e.g., phosphorylation) and logical (e.g., activation) definition that specify the catalyst, upstream, downstream, and side action-agents. Action agents can be substances or effects (e.g., heat shock, radiation). Therefore, certain triggers of processes can be expressed. However, there is no underlying dynamic model, and high-level processes simply contain ordered lists of their constituting actions.

Modeling of system dynamics, function, and structure has been extensively studied in the fields of software engineering and business management. Many commonalities exist between biological systems and man-made systems. Therefore we investigated the possibility of using the models developed in these disciplines to represent biological systems. Following is an analysis of some of the system modeling methods that were developed in those fields.

Statecharts (Harel and Gery, 1997; Harel et al., 1990) provide a formal, executable representation formalism that describes dynamics of reactive systems. **Statecharts** are state machines with hierarchy, orthogonality (i.e., parallel states), and broadcast communication. They can represent concurrent behavior, by splitting the system into structural components (e.g., enzyme1, substrate1), and for each component, defining its own Statechart. For example, 37 proteins and 12 biomolecular complexes participate in the process of invasion of host erythrocytes by Malaria parasites. Each one of these components would have its own Statechart, and a biologist who would want to understand the overall system behavior would have to integrate the information contained in the different diagrams. Tools would be very helpful in this task In terms of expressiveness, Statecharts is a formalism that concentrates on representing system dynamics. It does not represent the functional roles of structural components.

The Object Modeling Technique (OMT) (Rumbaugh et al., 1991), and the Unified Modeling Language (UML) (Booch et al., 1998) combine Statecharts with other models that represent system structure and function. However, the large number of models hinders the human comprehension of integrated system structure and behavior (Peleg and Dori, 1999).

The Workflow model of the Workflow Management Coalition

The Workflow Management Coalition defined a Workflow model for business processes (Workgroup Management Coalition, 1999). We found that this model can be mapped to biological systems. In the following two definitions, the biological analogous entities are given in parentheses. A *Business Process* is a set of linked procedures or activities (component processes) that collectively realize a business objective or policy goal (biological behavior), normally within the context of an organizational structure (e.g., cell, organism) defining functional roles (e.g., receptor) and relationships (e.g., members of a bimolecular complex). A *Workflow model / Process Definition* is a representation of a business (high-level biological) process in a form that supports automatic manipulation. The process definition provides a dynamic and functional model that consists of a network of activities (logical steps in the process) and their relationships, criteria to indicate the start and termination of the process, and information about the participants of individual activities. Activities are connected to each other using transitions, which may contain conditions (conditional transition). Four types of activities can be expressed: (1) activities that are used for **routing** and do not contain process definitions (route activity), (2) activities that are **hierarchical** and can be nested into sub-flows (subflow activity), (3) **loop** activities, and (4) **generic** activities, that do not contain sub-flows and loops. The process definition is often graphically displayed, as a flowchart of activities, which eases human comprehension. An example of a Workflow model is shown in Figure 1 and Figure 2. Workflow models can be mapped to Petri Nets (Aalst, 1998), which can be analyzed for correctness**.**

Petri Nets

A **Petri Net** (Peterson, 1981) is a formal model that is used to model concurrent systems. A Petri Net is represented by a directed, bipartite graph in which nodes are either places or transitions, where places represent conditions (e.g., parasite in blood stream) and transitions represent activities (e.g., invasion of host erythrocytes). Tokens that are placed on places define the state of the Petri Net (marking). A token that resides in a place signifies that the condition that the place represents is true. A Petri Net can be executed in the following way. When all the places with arcs 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 place pointed to

by the transition. **High-level Petri Nets** include extensions that allow modeling of time, data, and hierarchies

There are many benefits to using Petri Nets. First, Petri Nets have a firm mathematical foundation that allows analysis of performance measures and analysis of properties. Following are examples of properties that are relevant for biological systems.

(1) Liveness: All transitions (biological processes and reactions) can be enabled;

(2) Boundedness: In every place, the number of tokens is always less than n (e.g., no toxic accumulation of metabolites, where each metabolite molecule is represented by one token);

(3) Soundness: a combination of liveness and boundedness that ensures proper termination. If we add one *source* place with one token, and one *sink* place, then, the procedure will terminate eventually; the moment the procedure terminates there will be a token in the sink place and all other places will be empty. In addition, there should be no dead tasks (i.e., activities that never happen). In terms of biological systems, soundness ensures that all biological processes and reaction could be carried out and while the system executes, there will be no infinite accumulation of one species of biomolecule or cell type (relevant for examples of Malaria parasites); and

(4) Reachability: given a certain state (marking) M of a Petri net, is another state, M', reachable from state M? (e.g., if we block the immune system, can we still reach a state where the parasite is cleared from the blood system?).

A second benefit of Petri Nets is that they explicitly represent states, which allows for the modeling of milestones and implicit choices. A third benefit is that Time Petri nets (Berthomieu and Diaz, 1991) can express temporal constraints on the earliest and latest time of a transition. Thus, Petri Nets can be used to represent minimum and maximum duration of a process (that occurs on a transition). Structural analysis techniques and enumerative methods can be used to analyze time Petri Nets. Another benefit is that Hierarchical Petri Nets can control the complexity of the representation of biological systems. And last, Colored Petri Nets can define attributes of tokens, states, and transitions. This can be used to represent the different participants that perform the tasks.

Petri Nets, and other graph-theoretical models, have been used by several groups to model biological pathways. In these models, biological processes are represented as a collection of

concurrent processes. However, these Petri-Net-based models, described below, do not include a static-structural model or ontology of biological concepts. Petri Nets are an ontology of places and transitions. It is not an ontology that can reason about biological concepts such as bio-molecules, genes, and functional roles. In (Reddy et al., 1996), Petri Nets are used to generate a qualitative model of metabolic pathways. Places represent compounds and transitions represent enzymatic reactions. This model was used to identify reactions that can operate continuously (violate the fairness property) and may therefore be good targets for affecting system behavior. Kufner and colleagues used a similar approach to create Metabolic Displays (MDs) (Kufner et al., 2000) of all pathways between *source* and *sink* metabolites, based on data from metabolic and sequence databases. Differential Metabolic Displays (DMDs) display the differences between two or more MDs of distinct biological states (e.g., different tissue types, different organisms). DMDs can identify gaps in specific pathways and enable prediction of existence or absence of specific proteins and protein functions in certain systems.

Self-Modified Petri Nets have been used to represent a quantitative model of biochemical networks (Hofestadt and Thelen, 1998). Hybrid Petri Nets were used to model regulatory networks by taking into account concentrations of proteins and RNA (Matsuno and Doi, 2000).

Stochastic activity networks (SANs), an extension of Petri Nets, were used for representing biological pathways and simulating their kinetics (Mounts and Liebman, 1997).

MetaNets (Kohn and Lemieux, 1991) are a graph theoretical model of metabolic gene-expression networks. Nodes represent metabolites, enzymes, and nucleic acid, while arcs represent relationships between pairs of nodes (e.g., substrate, inhibitor). The method can identify regulatory properties of metabolic networks.

Object-Process Methodology

The **Object-Process Methodology (OPM)** (Dori, 1995; Peleg and Dori, 1999) was developed in the field of information systems engineering. It represents structure, function, and behavior of systems in a single, scaleable, graphical model. The static-structural model is more elaborate than the one used in workflows. It can represent not only *part-of* and particular *general* structural relationships, but also specialization (*is-a*), and *characterization* relationships that specify attributes of objects. OPM is very flexible, so a modeler who uses

OPM can add any attribute that he wishes to an object that participates in a process (e.g., cellular location). However, OPM does not guide the user as to what should be the structural components of objects that participate in processes. Functional aspects (i.e., participants, inputs, and outputs of the processes) are represented graphically in OPM, unlike the Workflow model. The behavioral model of OPM supports the notion of events, exceptions, and temporal constraints. The graphical specification of OPM has an equivalent text-based specification in the form of English-like sentences that aid domain experts in validating OPM models.

PSL and PIF

The Process Specification Language (PSL) (Schlenoff et al., 2000), and the Process Interchange Format (PIF) (Lee et al., 1998), which has been merged with PSL, can represent the temporal relationships among activities and the participants in the activities. However, these models cannot represent the structural aspects of the participants, do not designate roles (functions) to the participants, and do not describe the data (substances) that is used as input and output of activities.

Business Process Modeling Language

The Business Process Modeling Language (BPML) (Arkin and Agrawal, 2001) is an XML-based markup language designed to model business processes deployed over the Internet. BPML specifies transactions, data flow, messages and scheduled events, business rules, security roles, and exceptions. It supports both synchronous and asynchronous distributed transactions. Each process includes a definition of all messages communicated between the process and its participants, which is similar to Statecharts's broadcast communication mechanism.

Our model

We found that by combining the best aspect of two of the models: (1) the Workflow model that is mappable to Petri Nets, and (2) TAMBIS, which serves as a biological concept model, we can fulfill all of the requirements for a dynamic model for high-level biological processes. In the context of biological systems, a Biological Workflow Model is a representation of a high-level biological process in a form that supports automatic manipulation. A *high-level*

biological process is a set of linked component processes that collectively realize biological behavior, normally within the context of a cell or organism, defining functional roles and relationships among cellular substances. The organization of component processes as a network provides a dynamic model. The subflow activities represent high-level processes are hierarchically nested into lower-level component processes. The input and output parameters (substrates and products) of low-level processes are specified, thus providing a functional definition. The process participants are specified in the organizational model, which provides the static-structural aspect of the workflow. A participant may be an individual (bio-molecule, in analogy to a human participant, who is also an individual), a group of individual molecules that are assembled into a unit (organizational unit in the original Workflow definition and a biomolecular complex in a biological system), or a functional role (e.g., protease). The process definition is often graphically displayed, as a flowchart of activities, which eases human comprehension. Workflow models can be mapped to Petri Nets, which can be analyzed for correctness**.**

The Workflow model does not satisfy two important requirements: inclusion of a biological concept model and reasoning that is dependant on biological concepts. By representing biological entities with a biological controlled vocabulary of concepts (TAMBIS) and integrating the Workflow model with TAMBIS, we were able to support these requirements as well.

Mapping the workflow model constructs to biological entities

We used the workflow model as a biological process model by mapping

(1) Activities to biological processes;
(2) Data inputs and outputs to substrates and products, respectively;
(3) Organizational units to biomolecular complexes;
(4) Humans (individuals) to biopolymers;
(5) Workflow participants to biopolymers and biomolecular complexes; and
(6) Roles to biological processes and functions.

We extended the Workflow model to support biological systems by including the class hierarchy of the TAMBIS ontology, version 0.96, which serves for our framework as a biological controlled vocabulary. The biopolymers, biomolecular complexes, and the different

types of biological processes and functions that we use in our workflow model all refer to classes in the TAMBIS ontology.

In our representation, we link the biopolymers and biomolecular complexes to (1) biomolecular complexes to which they belong, (2) roles that they fulfill, (3) their inhibitors, and (4) database entries of corresponding protein and gene sequences.

Adding other elements that are relevant to biological systems

We added a categorization of evidence that we have for facts in the knowledge-base, and the cellular location of the participating elements, As well as Malaria specific elements. We classified the evidence into five categories that we found useful when modeling processes that occur in *Plasmodium* species. The evidence types are: "speculative", "in-vivo", "in-vitro", "in-culture", and "inferred from other *Plasmodium* species". Our evidence classification stresses the likelihood that the fact that is represented is true. This is different from the evidence code system used by the Gene Ontology (The Gene Ontology Consortium, 2001)that stresses the method by which the information was retrieved. For example, we found it useful to distinguish between evidence based on in-vivo versus in-vitro or in-culture studies. The distinction is needed because in-vitro studies are always a simplification of the processes that take place in-vivo. We use the code "inferred from other *Plasmodium* species" to represent genes or gene products that are present in one *Plasmodium* species but are not identified yet in the *Plasmodium* species that is modeled. Since *Plasmodium* species are closely related, facts that come from a related species may well have corresponding counterparts. The levels of evidence can be considered in pointing out weak points in the model, where more experiments should be carried out to collect more data. Level of evidence should also be considered when the model is inconsistent with new results for which experimental evidence is strong.

Graphical representation of the relationships between a process and its participants

We augmented the Workflow model with elements taken from OPM, to create a graphical representation of the relationships between a process and the static components that participate in it, as shown in Figure 2. We used different connectors to connect a process to its input sources, output sources, and participants that do not serve as substrates or products (e.g., catalysts). We added a fourth type of connector that links a process to a chemical that inhibits the process. This inhibitor arc is also present in an extension of Petri Nets discussed in (Peterson, 1981).

System Architecture

We used the Protégé-2000 knowledge-base-editing tool (Noy et al., 2000) to create the ontologies of our framework, and configured it to graphically display the structural and behavioral aspects of the model. Figure 3 shows the system architecture, including mappings among parts of the model, and data flows.

Reasoning capabilities

Using Protégé's Axiom Language (PAL) we composed first-order logic queries that can aid in discovering relationships among processes and structural components. We also used Protégé's query tool to form simple queries that return all instances of a class where one of the class's slot values contains a specified value (e.g., all the biomolecular processes that exhibit the function of Adhesion).

A biological Workflow Example

The example is based on the invasion of human host erythrocytes by the Malaria parasite *Plasmodium falciparum* (Barnwell and Galinski, 1998; Blackman 2000). The full model includes ten Workflow diagrams, and is available at http://smi.stanford.edu/projects/helix/pubs/process-model/**.** Due to lack of space we only show two of the Workflow diagrams. At the start of the invasion process, the Malaria parasite is at the developmental stage called merozoite. The invasion process has three phases. In the first phase, the merozoite recognizes an erythrocyte and attaches to it. In the second phase, the merozoite enters the erythrocyte and is positioned inside a vacuole within the cytoplasm of the erythrocyte. In the final phase, the membrane of the parasitophorous vacuole (PV) and the plasma membrane of the erythrocyte close. This is followed by the transformation of the merozoite to trophozoite (not shown in this paper). Figure 1 shows the workflow model of the invasion process. The invasion process starts with "Merozoite in blood", followed by the processes "Merozoite encountering an erythrocyte" and by "Merozoite recognizing an erythrocyte that is appropriate for invasion". The next stage in the invasion process involves "attachment, reorientation, and tight-junction formation". In the following stage, three activities are done in parallel: "Entry of merozoite into erythrocyte", "Formation of PV", and "Localization of Rhop complex". The figure shows details of the later activity; the Rhop complex localized to the rhoptries serves as a substrate to the activity. The activity results in

the Rhop complex in the location of the PV membrane. At the next stage, three more activities are done: "Closure of erythrocyte membrane" and "Closure of PV membrane around the merozoite", which are done in parallel, and "Fusion of microspheres with merozoite plasma membrane", done before the two other activities, as indicated by its position above the those activities. The figure shows that the "Fusion of microspheres with merozoite plasma membrane" process dislocates the ring-infected erythrocyte surface antigen (RESA) and ring-membrane antigen (RIMA) from the microspheres. On the left side of the diagram, the activity "MSP-1 (merozoite surface protein 1) secondary processing" is shown. The timing of this activity is not certain, but it is known that it correlates with the time of either the last stage or the stage before last of the invasion process. Therefore, this activity is specified in parallel to both of these stages.

An example of hierarchy is shown in Figure 2. It represents the expansion (nesting) of the activity "attachment, reorientation, and tight-junction formation", presented in Figure 1. The first stage of this process involves one of three activities: the most prevalent activity, shown in the middle, "Initial attachment to erythrocyte involving Glycophorin A", or one of the alternative pathways for initial attachment. The initial attachment process is followed by "Reorientation of merozoite". Next, the activities "Processing of AMA-1" and "Formation of tight junction" are performed. The way in which AMA-1 is involved in invasion is not clear. Also shown, are the details of the "tight-junction formation" activity. MCP-1 (Merozoite capping protein 1) is thought to be involved in this activity. The merozoite's EBA-175 and the erythrocytic Glycophorin-A act as substrates. The product of this activity is a complex of EBA-175 and Glycophorin A. The figure also shows several inhibitors (Neuraminidase, trypsin, chymotrypsin, and chymostatin) that inhibit the tight-junction formation activity. Double clicking on any of the participants, substrate, or products, shown in a workflow, reveals their details, as shown in the insert of Figure 2. These details include attributes that are defined by the biological ontology: (1) synonyms, (2) the name of the biomolecular complex to which the biopolymer belongs, (3) the roles that the biopolymer fulfills, (4) alternative biopolymers that can fulfill its roles, (5) the inhibitors of the biopolymer, (6) a database entry that stores the sequence of the biopolymer, and (7) the sequence component that codes for the biopolymer (e.g., gene, list of exons).

Figure 4 shows the static-structural view, which depicts the participants, substrates, and products of activities, and the structural relationship among them. One relationship links a biopolymer to the biomolecular complex of which the biopolymer is a member. Another type of relationship links a biopolymer or biomolecular-complex to its biological function. Other types of relationships link a complex to a higher-level complex that includes it, and link a biopolymer to other biopolymers that can fulfill its role (not shown).

Mapping the Workflow model to Petri Nets

We wanted to support qualitative, rather than quantitative simulation, because many of the quantitative biological data is missing, unreliable, or imprecise. We also wanted the simulation model to be hierarchical, to control complexity of the modeled systems. We therefore chose to use hierarchical Petri Nets that answer these requirements. Van der Aalst defined the mapping from the Workflow Model of the Workflow Management Coalition to Petri Nets in (Aalst, 1998). Only the control-flow aspect of the workflow model (i.e., the Process Model) is mapped. The mapping abstracts from the static-structural and the functional parts of the model. In a nutshell, activities are represented by transitions, and places (conditions) are introduced in between every two activities that are connected to each other. Branching is mapped by using Petri Net building blocks for representing AND splits and joins, and XOR splits and joins. In addition, the time interval during which an enabled transition may fire can be specified. This enables modeling of constraints on the duration of an activity. Figure 5 shows the Petri Net that corresponds to the Workflow of Figure 2. We are planning to support, in the future, automatic generation of Petri Nets from workflows. Once the Workflows have been translated to Petri Nets, we could use different commercially available tools, such as Woflan (Aalst, 1998) or DesignCPN (http://www.daimi.au.dk/designCPN/) to verify the Petri Nets and perform simulation. These tools can handle Petri Nets generated from Workflows that have up to hundreds of processes.

Reasoning about the model

We identified five types of queries that can be used to generate biological information and aid in prediction: (1) functional roles, (2) biological reactions, (3) biological processes, (4)

reachability, and (5) temporal/dynamic aspects. Table 2 shows the query types, some motivating biological examples, and the answer that was derived from our system.

We used Protégé's first-order axiom language to define the first three kinds of queries. The two other query types were manually computed, at this stage. As shown in Table 1, queries that relate to system structure and functionality (categories 1,2, and 3) can be composed for the other biological models that we reviewed.

Queries that concern the dynamic aspects of the system (categories 4 and 5) can be answered by using our model, because it has a mapping to hierarchical Petri Nets. For example, we answered the question: "Without the main process of initial attachment *Initial attachment to erythrocyte involving Glycophorin A* (t_3 of Figure 5), can we get to *Merozoite permanently attached* (P_8)?, By constructing a reachability tree or alternatively, a firing sequence, we showed that P_8 is reachable through one of the two activities that represent alternative pathways of invasion. For example, a firing sequence that uses the alternative pathway for invasion that uses Glycophorin B, starts with one token in place P_1 and ends with one token in place P_8 is $t_1t_2t_5t_6t_7t_8$. Although the query that we show relies only on the Petri Net shown in Figure 5, to ease comprehension, we could also ask a query that relies on information that is contained in the two Petri Nets that correspond to figures 1 and 2 (i.e., if we inhibit the process "Initial attachment to erythrocyte involving Glycophorin A", shown in Figure 2, can we get from the state of "Merozoite in blood", shown in Figure 1, to the state "enveloped merozoite in erythrocyte", shown in Figure 1). In this case, reachability is harder to notice by simply looking at the set of workflows.

Discussion

Our work differs from that of the pathway modeling methods that have used Petri Nets in several respects: (1) we represent substrates, products, and catalysts as parts of a functional model of the workflow. When we translate Workflows to Petri Nets, these components are not represented as places, (2) our model includes a biological controlled vocabulary, (3) We have a static-structural model that organizes the participants of a system in subsumption hierarchies, and represents relationships such as part-of, inhibitor, and relationships between a gene to its product, and to its sequence database entry, and (4) we defined queries that answer different types of questions about biological structure, function, and dynamics.

The model that we developed is a qualitative model of biological processes; it does not represent cellular concentration of reactants, or kinetic coefficients that are required to formalize quantitative models. It might be possible to extend our model to represent such information by using colored Petri Nets, where the attributes of tokens can account for the cellular concentration of reactants. Alternatively, we can use the Hybrid Petri Nets approach, used in (Matsuno and Doi, 2000).

Although it is not our focus at this time, our model can represent uncertainty in the temporal relationships among processes, as was done in composing the "MSP-1 secondary processing" activity in parallel to both the last and next to last phase of the invasion process, shown in Figure 1. Another form of uncertainty involves branching into alternative paths. Biologists often do not know what controls the branching, but do know the *probabilities* of cases that follow each path. Probabilities are not modeled in Petri Nets. However, they can be represented as conditional flows using explicit XOR splits (e.g., the condition of the XOR may be "P < 0.2"), thus representing a static deterministic model. Bayesian Networks support probabilistic reasoning using dependent probabilities. Bayesian reasoning might be added to the current model. Coarse granularity can be used when representing a highly uncertain situation. In addition, learning methods could be deployed to calculate probabilities. Alternatively, probabilities may be simulated. We would like to implement a simulation tool that will allow us to "override" the static probabilities of a transition by replacing it with other probabilities. This can be useful to model the probabilities for different strains of Plasmodium or for different hosts.

We have added process duration constraints to our model. We need to implement quantitative temporal analysis, such as that described in (Berthomieu and Diaz, 1991). We can then be able to answer queries such as predicting the time interval during which a host that was bitten by an infective mosquito will be infective.

We need to develop a method for representing the differences in the process models of several species and strains. The representation should take into account both genotypic and phenotypic differences (e.g., different organelles, different protein concentrations). This could be extended to compare processes of the host versus parasite, for targeting processes that are unique or different in the parasite.

Conclusion

We developed a model for representing biological processes. Unlike other biological models, our model represents the dynamic, functional, and static aspects of biological processes, and uses controlled biological terminology. The dynamic representation uses a formal mathematical model that enables verifying boundedness and soundness as well as answering reachability questions, which in the context of biological systems, may aid in predicting system behavior in the presence of dysfunctional processes, functions, or structural components. Our system also enables answering queries that concern structural and functional aspect of biological systems.

Acknowledgements

The work was funded by the Burroughs Wellcome Fund and grant GM07365-26 from the National Institute of General Medical Sciences. We thank Dr. T. Hanekamp for his help in validating the biological correctness of our models and for his insights about representing inhibitors and data from related Plasmodium species. We thank Dr. H. Ginsburg for helping us validate the correctness of the Malaria example and for his helpful suggestions.

References

Aalst, W. M. P. v. d. (1998). The application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers* **8**, 21-66.

Arkin, A., and Agrawal, A. (2001). Business Process Modeling Language, Working draft 0.4: Business Process MAnagement Initiative). http://www.bpmi.org/bpml-spec.esp

Baker, P. G., Goble, C. A., Bechhofer, S., Paton, N. W., Stevens, R., and Brass, A. (1999). An ontology for bioinformatics applications. *Bioinformatics* **15**, 510-520.

Barnwell J.W., and Galinsky M.M. (1998). *Chapter 7: Invasion of Vertebrate Cells: Erythrocytes*. In Sherman, I.W. (ed), *Malaria: parasite biology, pathogenesis and Protection*. ASM Press, Washington DC, pp. 93-113.

Berthomieu, B., and Diaz, M. (1991). Modeling and verification of time dependent systems using time petri nets. *IEEE Transactions on Software Engineering* **17**, 259-273.

Blackman M.J., (2000). Proteases involved in erythrocyte invasion by the malaria parasite: function and potential as chemotherapeutic targets. *Curr Drug Targets.* **1**, 59-83.

Booch, G., Rumbaugh, J., and Jacobson, I. (1998). *The Unified Modeling Language User Guide*: Addison-Wesley Longman, Inc.).

Bordiga, A. (1995). Description Logics in Data Management. *IEEE Transactions on Knowledge and Data Engineering* **7**, 671-682.

Dori, D. (1995). Object-Process Analysis: Maintaining the Balance Between System Structure and Behavior. *Journal of Logic and Computation* **5**, 227-249.

Frishman, D., Albermann, K., Hani, J., Heumann, K., Metanomski, A., Zollner, A., and Mewes, H. W. (2001).

Functional and structural genomics using PEDANT. *Bioinformatics* **17**, 44-57.

Gruber, T. R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Int. Journal of Human-Computer Studies* **43**.

Harel, D., and Gery, E. (1997). Executable Object Modeling with Statecharts. *IEEE Computer* **30**, 31-42.

Harel, D., Lachover, H., Naamad, A., Pnueli, A., Politi, M., Sherman, R., Shtull-Trauring, A., and Trakhtenbrot, M. (1990). STATEMATE: A Working Environment for the Development of Complex Reactive Systems. *IEEE Transactions on Software Engineering* **16**, 403-414.

Hofestadt, R., and Thelen, S. (1998). Quantitative Modeling of Biochemical Networks. *In Silico Biology* **1**.

Karp, P. D. (2000). An ontology for biological function based on molecular interactions. *Bioinformatics* **16**, 269-285.

Kohn, M. C., and Lemieux, D. R. (1991). Identification of Regulatory Properties of Metabolic Networks by Graph Theoretical Modeling. *Journal of Theoretical Biology* **150**, 3-25.

Kufner, R., Zimmer, R., and Lengauer, T. (2000). Pathway analysis in metabolic databases via differential metabolic display (DMD). *Bioinformatics* **16**, 825-836.

Lee, J., Grunninger, M., Jin, Y., Malone, T., Tate, A., and Yost, G. (1998). The PIF Process Interchange Format and Framework v. 1.2. *The Knowledge Engineering Review* **13**, 91-120.

Matsuno, H., and Doi, A. (2000). Hybrid Petri Net Representation of Gene Regulatory Network. In Pacific Symposium on Biocomputing, pp. 338-349.

Mounts, W. M., and Liebman, M. N. (1997). Qualitative modeling of normal blood coagulation and its pathological states using stochastic activity networks. *International Journal of Biological Macromolecules* **20**, 265-281.

Noy, N. F., Fergerson, R. W., and Musen, M. A. (2000). The knowledge model of Protege-2000: Combining interoperability and flexibility. In International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000) (Juan-les-Pins, France.

Peleg, M., and Dori, D. (1999). The Model Multiplicity Problem: Experimenting with Real-Time Specification Methods. *IEEE Transactions on Software Engineering* **26**, 742-759.

Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems* (Englewood Clifs, NJ: Prentice-Hall).

Reddy, V. N., Liebman, M. N, and Mavrovouniotis, M. L. (1996). Qualitative Analysis of Biochemical Reaction Systems. *Comput Biol Med* **26**, 9-24.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenson, W. (1991). *Object-Oriented Modeling and Design* (Englewood Clifs, NJ: Prentice-Hall).

Rzhetsky, A., Koike, T., Kalachikov, S., Gomez, S. M., Krauthammer, M., Kaplan, S. H., Kra, P., Russo, J. J., and Friedman, C. (2000). A knowledge model for analysis and simulation of regulatory networks. *Bioinformatics* **16**, 1120-1128.

Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., Lubell, J., and Lee, J. (2000). The Process Specification Language (PSL): Overview and Version 1.0 Specification (Gaithersburg, MD: National Institute of Standards and Technology).

Schulze-Kremer, S. (1998). Ontologies for Molecular Biology. In Proceedings of the Third Pacific Symposium on Biocomputing, pp. 693-704.

Serres, M. H., and Riley, M. (2000). MultiFun, a multifunctional classification scheme for Escherichia coli K-12 gene products. *Microb Comp Genomics* **5**, 205-22.

The Gene Ontology Consortium. (2001). Gene Ontology Evidence Codes. http://www.geneontology.org/GO.evidence.html

The Gene Ontology Consortium (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics* **25**, 25-29.

Workgroup Management Coalition. (1999). Interface 1: Process Definition Interchange. http://www.wfmc.org/standards/docs/

Table 1. Properties of process modeling methods indicated with '+' if present, unless otherwise noted. The Graphics column denotes those methods that have a graphical representation. The Nesting column denotes those methods that enable nesting, to manage complexity. Methods that represent static-structural, function, and dynamic aspects of systems, are noted in the next column by S, F, and N, respectively. The letter C represents models whose dynamic aspects are component-based, and the letter I represents methods whose dynamic model is integrative. The "Bio-specific information" column indicates methods that have biological-specific ontologies, or that distinguish among components, based on a biological understanding (e.g., substrates, products). The next two columns indicate methods that support verification of model properties (e.g., soundness), and methods that have supporting simulation or validation tools, respectively. "Computational model" notes the formal model on which the model upon which the method rests. The biological query column lists the query types that are supported by each model. It is based on the five query types that are defined in Table 2.

Model	Graphics	Nesting	Struct. Funct. dynamic	Bio-sp. info	Verification	Simulation/ Validation tools	Computational model	Biological query*
Control. Vocabul.			S	+			-	1,3
TAMBIS			S	+			Description Logic	1,3
EcoCyc	+		S, F	+			Frames	1,2,3
Rzhetsky		+	S, F	+			Frames	1,2,3
Statechart	+	+	D (C)			+	Statecharts	
OMT/ UML	+	+	S, F, D (C)			+ for Statecharts	Semi-formal models+ Statecharts	
OPM	+	+	S, F, D (I)			+ of static aspects	Semi-formal	
Workflow	+	+	S, F, D (I)		+	+	Petri Net	
Petri Net models	+	+	D (I)	In some	+	+	Petri Net	DMD: 1,2,4,5 Meta Net: 2,3
PIF/PSL		+	D (I)				Data model In KIF	
BPML		+	D (C)				XML	
Our model	+	+	S, F, D (I)	+	+	+ for Petri Nets	hierarchical PN, Frames	1,2,3,4,5

* The biological query types are defined in Table 2. For the axis of biological queries, we only considered models that can represent biological-specific information.

Table 2. Types of biological queries and motivating biological examples

Query type	**Example**	**Derived answer from the model**
1. Roles		
1.1 Biomolecular complexes or Biopolymer that have the same role - Scoped to cellular location, same substrates and products, same biological process (participation), or to the same (or different) inhibitor	Biopolymers that have the same set of roles and are not inhibited by the same inhibitor	Cysteine protease is inhibited by Leupeptin. The following proteins have the same roles but are not inhibited by Leupeptin: Serine Protease, Erythrocytic uPA, Calpain I and II
2. Reaction (functional model)		
2.1 All atomic activities that share the same substrates (products, inhibitors, participants)	- What atomic activities have the same participants?	"Triggering timely-release of the microneme proteins" and "Binding to reticulocytes" have the participants NBP1 and NBP2.
3. Biological Process		
3.1 All activities that are classified to be a kind of biological process, according to the TAMBIS classification hierarchy (scoped to cellular location)	- All activities that are a kind of adhesion - All activities that are a kind of adhesion and occur in the erythrocytic plasma membrane	Formation of tight junction, Binding to Reticulocytes, Initial attachment to erythrocyte Initial attachment to erythrocyte, Binding to Reticulocytes, Formation of tight junction
3.2 All activities that are inhibited by inhibitor x (see reachability)	All activities inhibited by cytochalasin All activities inhibited by neuraminidase	Reorientation, Actin Polymerization Formation of tight junction
4. Reachability		
4.1 If an activity is inhibited what other activities can take place? Is it a deadlock? XOR: use other activities (transitions), parallel: use other activities (transitions)	If we knock out, or Initial attachment to erythrocyte involving Glycophorin A" activity, what other activities will take place (directly in XOR, directly in parallel)	Directly in XOR: "Alternative pathway for invasion involving Glycophorin B", and "Alternative pathway for invasion involving no Sialic acid"
4.2 If an activity is inhibited, can we still get to a specified state?	If we inhibit "Initial attachment to erythrocyte involving Glycophorin A" , can we get to a state of "merozoite permanently attached"?	Yes. For example, $t_1t_2t_5t_6t_7t_8$
4.3 Does an inhibitor inhibit an entire high-level process (If inhibitor X inhibits a reaction then check that there isn't another path in the subflow activities that will go around that reaction).	Does neuraminidase inhibit the "Attachment, Reorientation, and tight junction formation" process?	Yes. It inhibits the sub process "Tight junction formations" which is essential
4.4 Establish a marking, find reachability	MSP is a substrate. What metabolic paths will be taken? What products will form?	Proteolytic cleavage followed by secondary processing. The products will be: 83 kDa, 38 kDa, 30 kDa, 19 kDa, and 33 kDa
5. Temporal/dynamic aspects		
5.1 What other processes occur in parallel to processX?	What processes occur in parallel (AND-ed) to "Entry of merozoite into erythrocyte"	Formation of PV, Localization of Rhop complex, MSP-1 Secondary Processing
5.2 How long does processX last?		30– 60 seconds

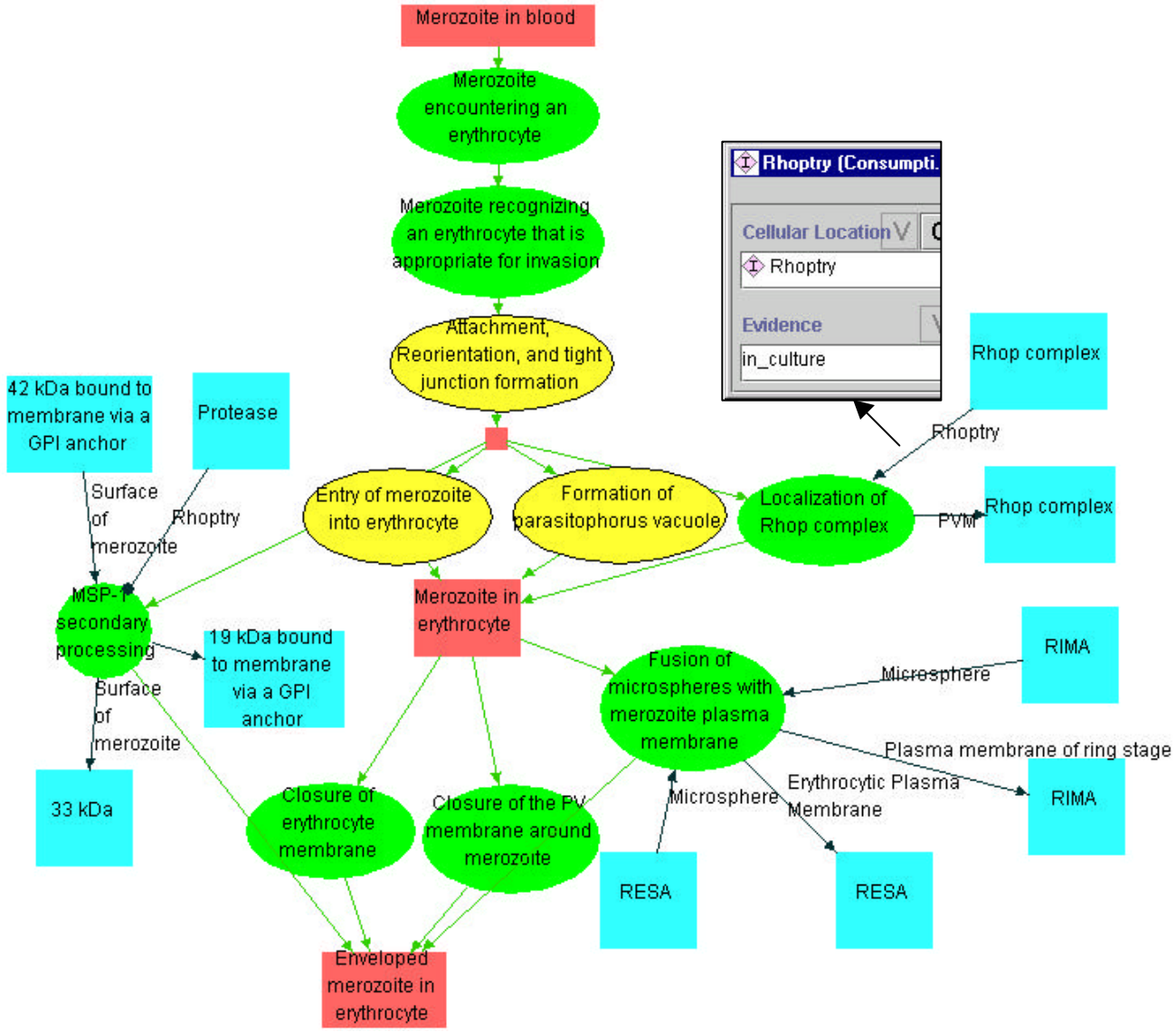

Figure 1. Process model of the invasion of erythrocyte by P. falciparum merozoites. Small squares represent routing activities. Complex activities that are nested into subflows are shown as ovals with a bold contour. Activities that do not contain subflows are shown as ovals. Arrows that connect two activities signify the flow of control. Branch points that are not marked by "XOR" signify parallel flows. Substrates and products are shown as squares that are connected to activities with arrows. Process participants that do not serve as substrates or products (e.g., Protease) are connected to the process with a connector that has a round end. The display can be configured so that the connectors display the cellular location of the products and substrates (shown) or the evidence for their involvement in the activity.

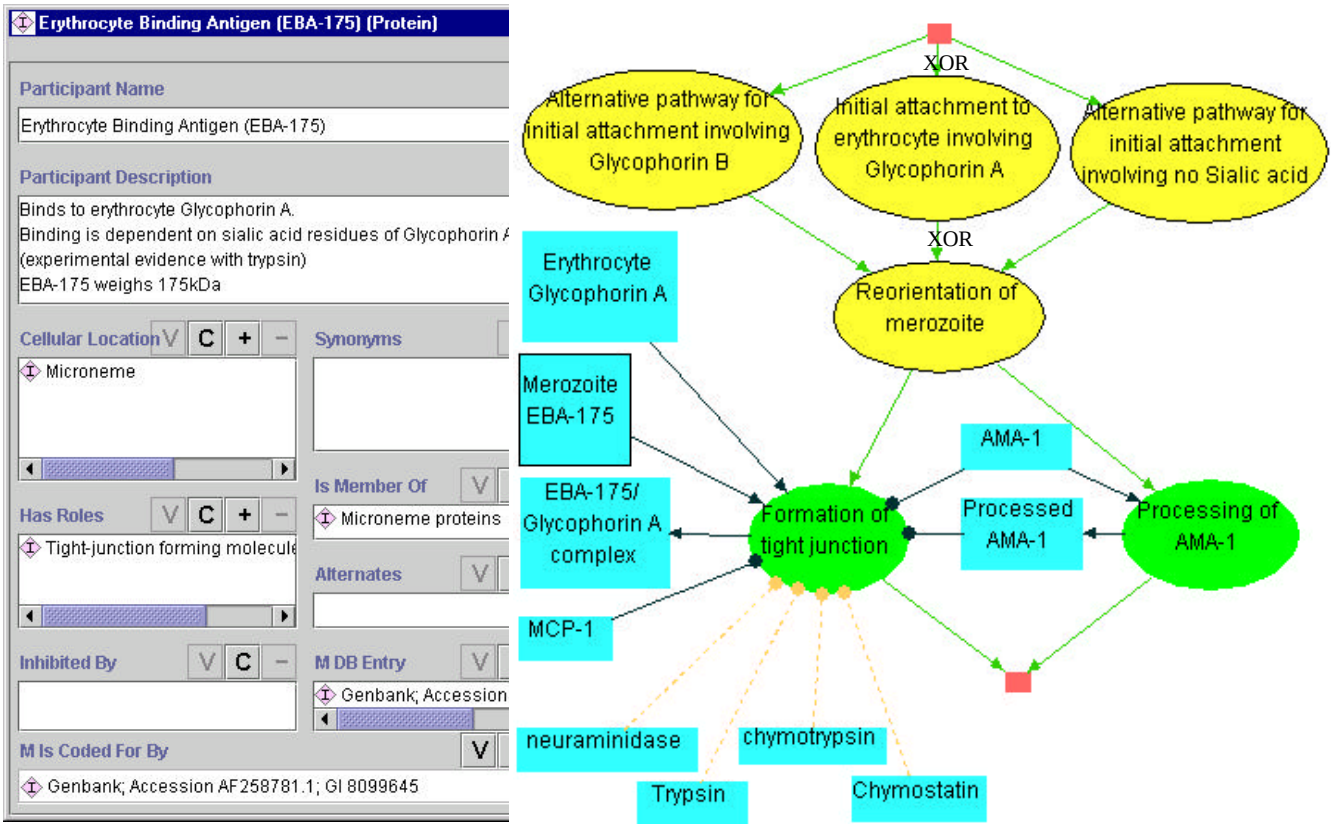

Figure 2. The detailed view (nesting) of Process "Attachment, reorientation, and tight-junction formation" of Figure 1 (third process from top). This figure uses the same symbols as Figure 1. In addition, inhibitors (e.g., neuraminidase) are shown as squares that are connected to an activity through a dashed line. The insert shows the details of one of the proteins (EBA-175) involved in the formation of tight-junction.

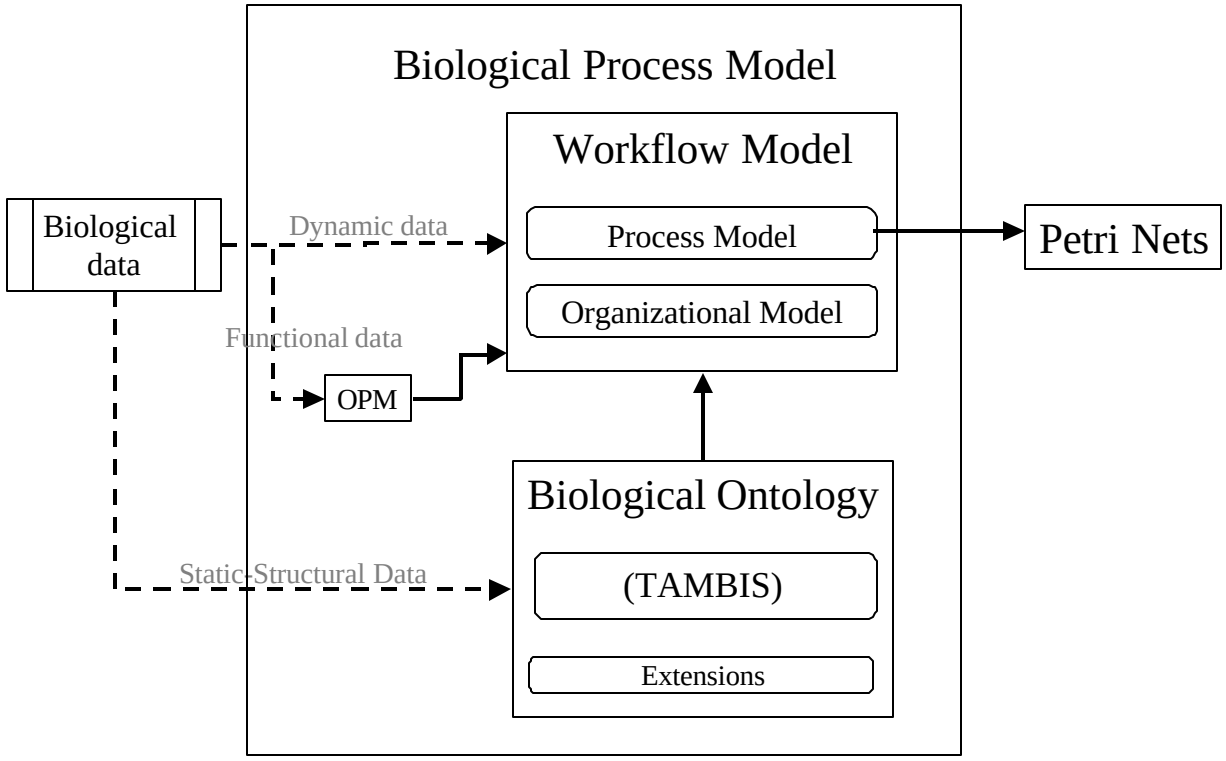

Figure 3. The architecture of our biological process model and its mapping to Petri Nets. The Biological Process Model consists of three parts: the Workflow model, the Biological Ontology, and OPM Relationships. The Biological Ontology consists of TAMBIS and our extensions to it. TAMBIS includes definitions of biological process, function, structure, and substance. The extensions that we made to TAMBIS consist of definitions of sub-cellular compartments, evidence types, and Malaria-specific elements. There are two types of arrows in the diagram. Solid arrows represent mappings between the different parts of our framework. As shown, TAMBIS is mapped to the Workflow Model; OPM relationships are mapped to the Workflow Model, and the control flow of the Process Model is mapped to Petri Nets. The dashed lines represent biological data that flows to the different parts of the Biological Process Model. Dynamic data (process organization) flows to Workflow Model. Functional data (process participants, substrates, products, and inhibitors) flow to the OPM relationships. Static-structural data flows to the Biological Ontology. It includes Sequence data, protein functional annotation, and process & function classification.

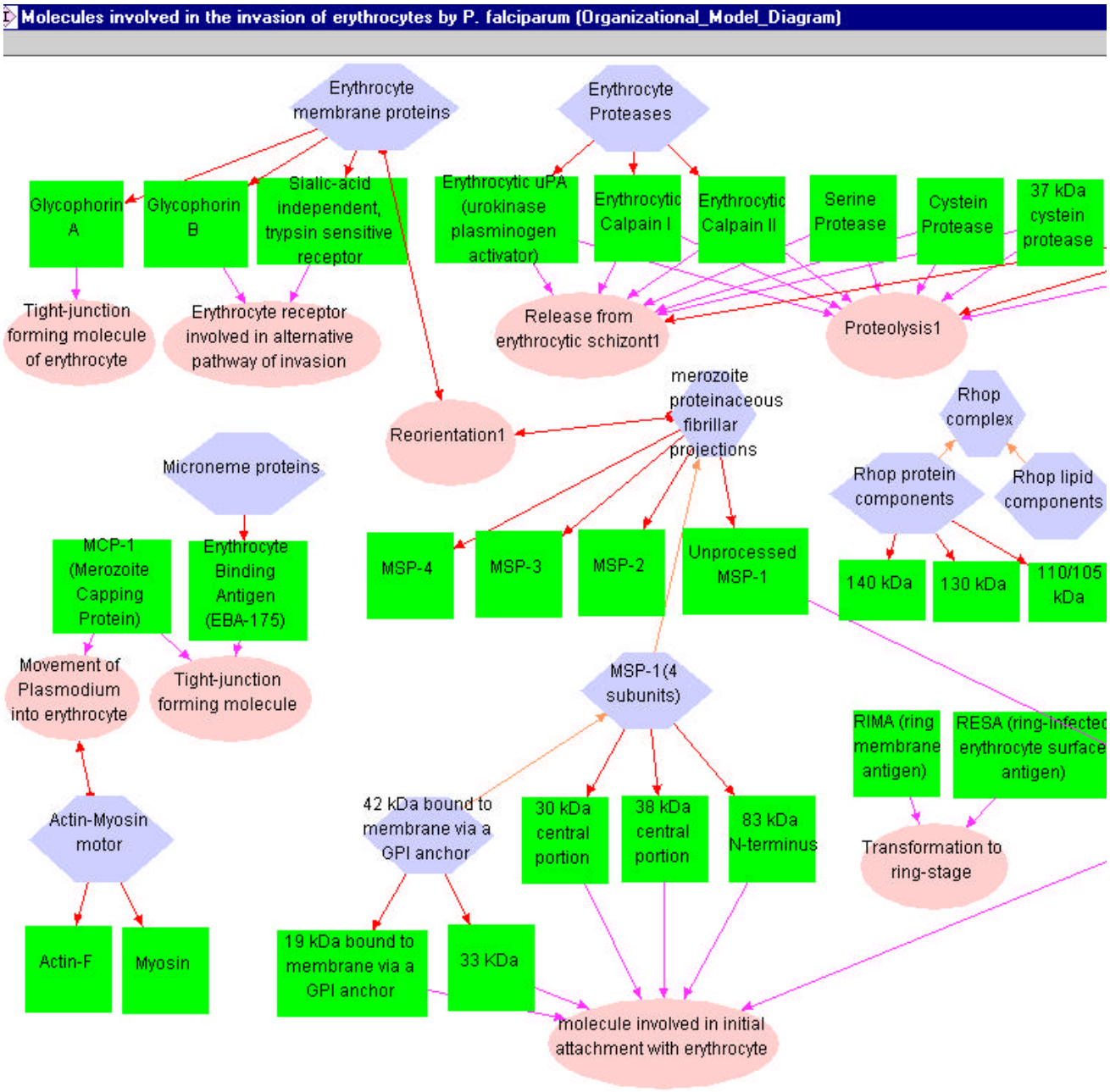

Figure 4. Part of the static model that shows biomolecular complexes (hexa gons), their members (which are biopolymers, shown as squares), and roles that the biopolymers or biomolecular complexes play (elipses).

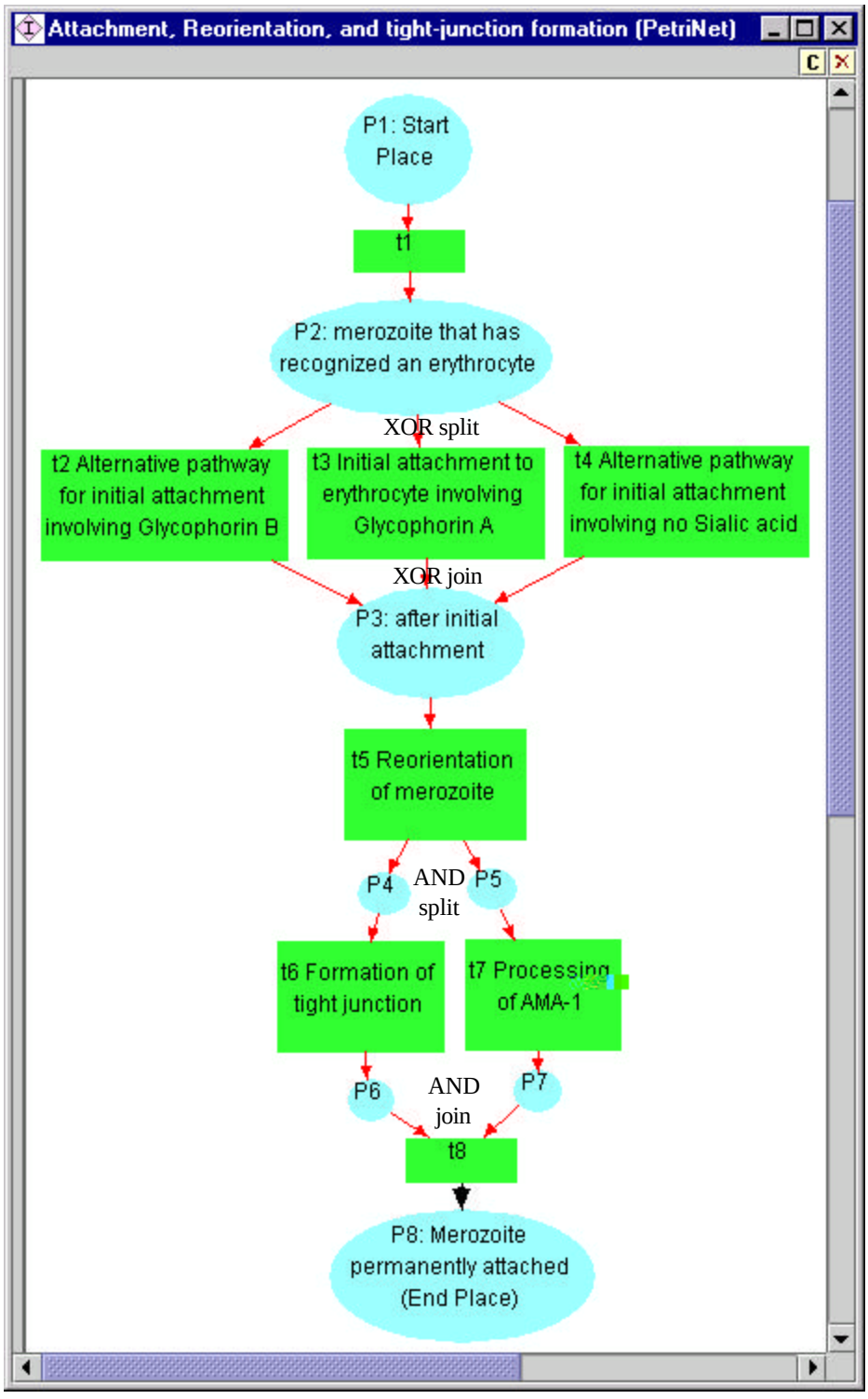

Figure 5. A Petri Net model of the invasion process, corresponding to the Workflow model shown in Figure 2. Places are shown as circles, and transitions as rectangles, and are labeled as $t_1..t_8$. The first and last places in the Petri Net are also labeled (as P_1 and P_8). Implicit XOR split and joins are marked as "XOR split" and "XOR join", respectively. AND split and joins are also marked.