CONVERSION FROM BIOPAX TO CSO FOR SYSTEM DYNAMICS AND VISUALIZATION OF BIOLOGICAL PATHWAY

EUNA JEONG
eajeong@ims.u-tokyo.ac.jp

MASAO NAGASAKI
masao@ims.u-tokyo.ac.jp

SATORU MIYANO
miyano@ims.u-tokyo.ac.jp

Human Genome Center, Institute of Medical Science, University of Tokyo, Tokyo 108-8639, JAPAN

The vast accumulation of biological pathway data scattered in various sources presents challenges in the exchange and integration of these data. Major new standards for representation of pathway data and the ability to check inconsistency in pathways are inevitable for the development of a reliable pathway data repository. Within the context of biological pathways, the cell system ontology (CSO) had been developed as a general framework to model system dynamics and visualization of diverse biological pathways. CSO provides an excellent environment for modeling, visualizing, and simulating complex molecular mechanisms at different levels of details. This paper examines whether CSO addresses the integration capability of pathway data with system dynamics. We present a conversion tool for converting BioPAX to CSO. Transforming the data from BioPAX to CSO not only allows an analysis of the dynamic behaviors in molecular interactions but also allows the results to be stored for further biological investigations, which is not possible in BioPAX. The conversion is done using simple inference algorithms with the addition of view- and simulation-related properties. We demonstrate how CSO can be used to build a complete and consistent pathway repository and enhance the interoperability among applications.

Keywords: Cell System Ontology (CSO); biological pathway integration; ontology mapping; system dynamics; BioPAX.

1. Introduction

In the current post-genomic era, the interactions among biological entities and networks are being uncovered by molecular biologists at an accelerating pace. Understanding individual biological entities and networks is not sufficient to explain how a cell works. There is a growing need for developing environments that enable us to describe complex and dynamic biological pathways at the system level. In order to address this requirement, we had developed a new system-dynamics-centered ontology called the cell system ontology (CSO) [3]. The three main features of CSO are as follows:

First, CSO allows the manipulation of different levels of granularities and abstraction of pathways, e.g., metabolic pathways, regulatory pathways, signal transduction pathways, and cell-cell interaction. CSO has a hierarchical structure to explicitly define the classes and relationships among those classes. It ensures that the relations

between the classes are treated in a correct and consistent way.
Secondly, CSO can capture both quantitative and qualitative models by using the hybrid functional Petri net with extension (HFPNe) [6]. CSO can explain not only the qualitative aspects of a model such as the biological functions and behavior of the networks but also the quantitative features such as reaction priority and kinetics.
Thirdly, CSO can encode information related to visualization and simulation of biological pathways. A well-designed representation will reduce the development time of special applications and enhance communication between software tools. In addition, CSO provides mature core vocabularies for annotating biological properties and standard icons for easy modeling and accelerating the exchangeability among applications.
Recently, a large amount of biological pathway data has been generated. This data is available in several formats such as BioPAX [1] and SBML [2]. Unfortunately, these formats neglect the system dynamics behavior or lack the formal definitions of each term. In order to facilitate data integration, we have made an effort to convert the existing pathway representations, particularly BioPAX, to CSO. BioPAX is based on a formal ontology, and many pathway databases export their data to the BioPAX format. BioPAX level 2 represents only metabolic pathway and molecular interactions. Additional types of pathway data such as signal transduction pathways and genetic regulatory networks are yet to be captured. Furthermore, the BioPAX format does not support dynamic models for simulation.
In this paper, we investigate the capability of CSO with regard to integration of pathway data with system dynamics. Transforming the data from BioPAX to CSO not only allows an analysis of the dynamic behaviors along with a visualization of the molecular interactions but it also allows the results to be stored for further biological investigations, which is not possible in BioPAX. The conversion allows other pathway data represented in the BioPAX format to benefit from CSO tools such as Cell Illustrator for visualization and simulation [5] and BioGraphLayout for automatic layout [4].

2. Comparison of Two Biological Pathway Representations – CSO and BioPAX

This section compares two representations–CSO and BioPAX–for biological pathways in the Web Ontology Language (OWL) [7]. In order to avoid confusion among terms used in the different formats, we use a namespace prefix `cso:` for CSO and `bp:` for BioPAX. CSO is a comprehensive representation for dynamic cell systems based on HFPNe [6] and consists of 195 classes. For the complete description and specifications of CSO, refer [3] and [10], respectively. BioPAX is a data exchange format for metabolic and molecular interactions, consisting of 41 classes. For the specifications of BioPAX, refer [1]. In this section, we compare only the two formats of the main data model.

Biological pathway model CSO is a general framework to understand the behavior of cell systems in an integrated manner. Therefore, the ontology has to represent not only a biological model itself but also a complete environment of the model, such as the results of model simulation, graphical representation of a model, and literature citations. All data in CSO is structured around `cso:Project` that represents the comprehensive environment of a pathway model. A project has one `cso:Model`, which describes the pathways via a set of simulatable biological processes based on HFPNe and biological facts. The class `cso:Fact` is designed to represent information that is not related to dynamic simulation but important for understanding the pathway functionalities such as the effect of a drug's efficacy in terms of the degree to which it binds to plasma protein.
BioPAX defines `bp:pathway` that consists of a set of interactions. Some interactions can be grouped to convey any meaning as pathway steps. A pathway consists of subpathways and interactions; alternatively, a pathway can be defined without specifying the interactions within it. In a pathway, the pathway steps can be listed in `bp:pathwayStep` and order relationships between pathway steps may be established to describe the overall flow of a pathway by using NEXT-STEP and STEP-INTERACTIONS slots. However, the temporal order may not be significant for specific steps.
Since the BioPAX classes `bp:pathway` and `bp:pathwayStep` provide insights into the underlying pathways, this information is mapped to `cso:Fact`. If the slot PARTICIPANTS of `bp:interaction` includes `bp:pathway`, i.e. a pathway catalyzed by an enzyme, this `bp:interaction` is converted to `cso:Fact`.

Biological interaction CSO has a process-centered structure to represent biological pathways. The class `cso:Process` represents the simulatable interactions among physical entities via `cso:Connector`. The same entity may play different roles along with the involved process as an activator, an inhibitor, or a reactant. Depending on its role, a different connector is needed.
BioPAX describes a catalyzed interaction by using `bp:control` and `bp:conversion` as subclasses of `bp:interaction`. The `bp:control` interaction must have one controller as a physical entity and one controlled interaction as a conversion. If a biochemical reaction is catalyzed by multiple enzymes, a separate catalysis interaction is required for each enzyme. If uncatalyzed, the `bp:conversion` interaction can be defined without `bp:control`.
Figure 1 shows how an enzyme-mediated trimerization of a protein complex is represented in (a) BioPAX and (b) CSO. In the figure, the boxes depict instances of classes and the arrows indicate the relationships between instances, i.e., slot names. BioPAX uses two classes `bp:catalysis` and `bp:biochemicalReaction` for the catalyzed interaction, while CSO requires only one–`cso:ProcessBiological`. The participants of `bp:catalysis` are an enzyme and an interaction described via CONTROLLER and CONTROLLED slots, respectively. In turn, `bp:biochemicalReaction` has two participants, as shown in the LEFT and RIGHT

slots. On the other hand, in CSO, three entities are involved in trimerization, and each of them is linked to trimerization via different connectors. The stoichiometric coefficient included in `bp:physicalEntityParticipant` is represented as the simulation property of `cso:Connector` rather than that of the physical entity.

Biological entity BioPAX defines several slots such as CONTROLLER, COFACTOR, RIGHT, and LEFT for describing the participants in interactions. The range of these slots are all `bp:physicalEntityParticipant`, which holds a physical entity as `bp:physicalEntity` and other information such as cellular location, stoichiometric coefficient, and sequence features.
In CSO, `cso:Entity` contains properties for location and sequence features related to an entity as well as the entity itself; it is not separated into two classes. Therefore, the combination of `bp:physicalEntityParticipant` and `bp:physicalEntity` corresponds to a single `cso:Entity`.
Figure 1 also shows how the participants involved in the trimerization of a protein complex are represented in (a) BioPAX and (b) CSO. In BioPAX, each LEFT and RIGHT is `bp:physicalEntityParticipant` that wraps `bp:complex`. In turn, the COMPONENTS slot of complex_R3 has another physical entity participant whose physical entity is complex_R1. In (b), each connector links a trimerization process to each entity. The three ENTITY slots of complex_R3 indicate that complex_R3 consists of three complex_R1s. The stoichiometric coefficient is represented as the connector's concentration to transfer from the entity to the process in CSO. The same `cso:Entity` can participate in two different processes via different connectors, which may have different kinetics.

Core vocabulary BioPAX refers to external controlled vocabulary to annotate biological pathway data such as cell types, cellular locations, evidence, and experimental forms. CSO also defines a class called `cso:ControlledVocabulary` (`CV`) for annotating biological properties.
However, there are several differences between BioPAX and CSO. First, in CSO, `cso:CV` is further divided into several subclasses for distinctive usage and rapid parsing, for example, `cso:BiologicalEvent` for a biological process and `cso:BiologicalRole` for the entity's primitive role, e.g. cofactor and enzyme. Second, `cso:CV` provides a predefined common vocabulary for annotation as instances of each subclass, rather than refers external sources. The terms are selected from freely available sources and reorganized to meet the objective of CSO. Lastly, in order to avoid losing the relationship already defined in the external sources, we place this information in CSO. It will reduce the time to parse and query external sources.

How to represent visualization and simulation The functionality of modeling system dynamics and visualization is unique to CSO. For the mathematical simulation of biological pathways, `cso:Entity`, `cso:Process`, and `cso:Connector`

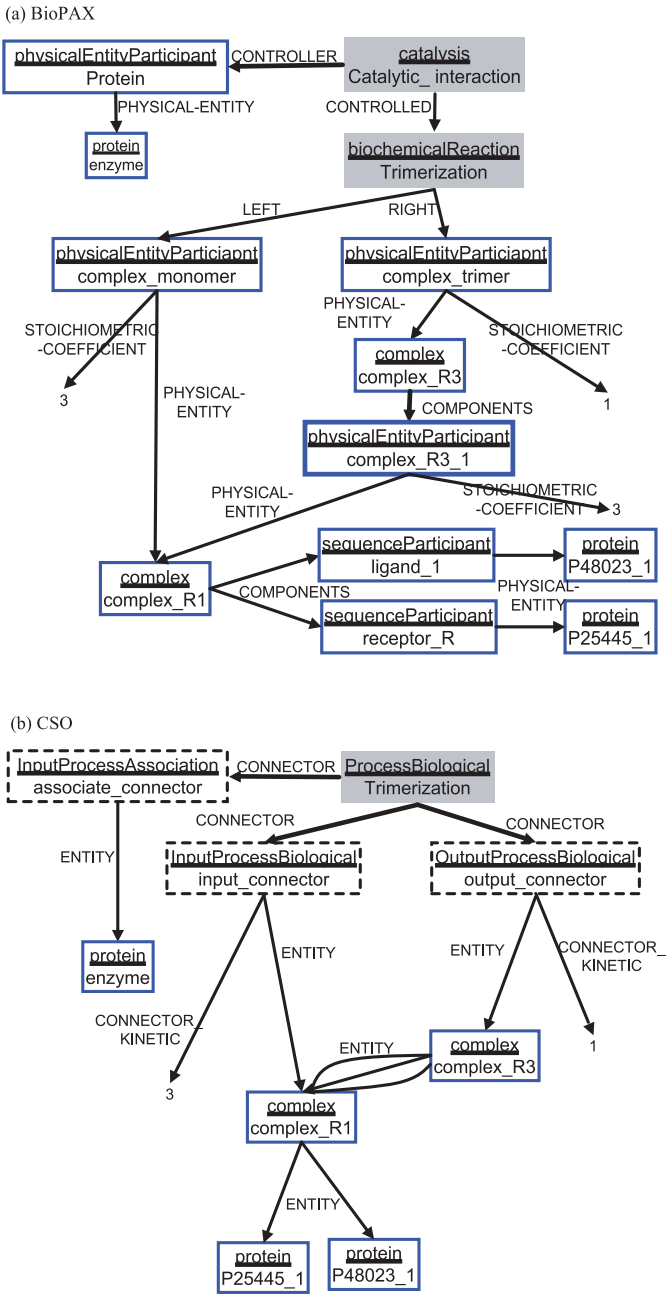

Fig. 1. Graphical view of (a) BioPAX and (b) CSO classes representing trimerization of a protein complex. The boxes show classes: grey for interaction, dotted for `cso:Connector`, and others for participants. The arrows connect two instances (classes) via slots.

all have different slots to describe simulation properties such as initial value, evaluatable script, kinetics, and delay (for details, see [3]). The conversion procedure assigns default values to the simulation-related properties. The simulation tools will use these values to enable the simulation of a time-interval-based discrete event, continuous events performed by differential equations, and more complicated events by using object-like programming language. CSO defines several graphical properties such as geometric position, graphical shape, and image-file-related properties for visualization. The layout of the biological components in networks and the corresponding image files can be stored in CSO, following which they will be used by the pathway visualization tools. In addition, the corresponding icons for all core terms of biological events and cell components are embedded in CSO. As an image file, both non-scalable and scalable coding are acceptable. For example, Portable Network Graphics (PNG) [12] and Scalable Vector Graphics (SVG) [14] are recommended.

3. Inference for Transforming BioPAX to CSO

For ontology mapping between BioPAX and CSO, the main focus is on the inference of `cso:Connector`, which defines the role of the participating entities and the simulation conditions. The comprehensive rules for mapping between BioPAX and CSO are given in Supplementary material.

In the first step, we identify `cso:Entity` from BioPAX data. From the definition of BioPAX, the physical entity participating in a process is stored in PHYSICAL-ENTITY of `bp:physicalEntityParticipant`. The same physical entity can be used in multiple `bp:physicalEntityParticipant`s. Hence, a pair of `bp:physicalEntityParticipant` and its `bp:physicalEntity` is mapped to one `cso:Entity`.

In the next step, we identify `cso:Process` from BioPAX data. The pair of `bp:control` and `bp:conversion` is mapped to `cso:Process`. If a control interaction controls a pathway, it will be mapped to `cso:Fact` and not to `cso:Process`. If `bp:conversion` is uncatalyzed, it is mapped to `cso:Process`. The algorithms for mapping `cso:Entity` and `cso:Process` are given in Supplementary material.

Here, we need to generate appropriate connectors for participant properties, namely, CONTROLLER, LEFT, and RIGHT. BioPAX contains five slots to influence the direction of a reaction and the regulation role of an enzyme: CONTROL-TYPE, DIRECTION, SPONTANEOUS, LEFT, and RIGHT. The LEFT and RIGHT slots denote the reaction direction as well as the participants. Algorithm 3.1 shows the inference procedure for the connectors.

The inference of CSO connector is simply divided into two parts based on whether or not a conversion interaction is controlled by a control interaction. The priority is given in the following order: CONTROL-TYPE, DIRECTION, and SPONTANEOUS. The CONTROL-TYPE property of `bp:control` defines the control relationship, which is described in lines 2 to 6 in the algorithm. Depending on the value, i.e., activation or inhibition, the controller entity in BioPAX is connected to

Algorithm 3.1 inferCSOConnector

```
1: if bp:control has bp:conversion then
2:   if (CONTROL-TYPE eq "activation") then
3:     create cso:InputAssociation for CONTROLLER
4:   else if (CONTROL-TYPE eq "inhibition") then
5:     create cso:InputInhibitor for CONTROLLER
6:   end if
7:   if (c_i is bp:catalysis and defines DIRECTION) then
8:     if (DIRECTION eq "reversible") then
9:       create cso:InputProcess for LEFT and cso:OutputProcess for
         RIGHT {for one process P1}
10:      create cso:OutputProcess for LEFT and cso:InputProcess for
         RIGHT {for another process P2}
11:    else if (DIRECTION eq "physiol-left-to-right" or "irreversible-left-to-
       right") then
12:      create cso:InputProcess for LEFT
13:      create cso:OutputProcess for RIGHT
14:    else if (DIRECTION eq "physiol-right-to-left" or "irreversible-right-to-
       left") then
15:      create cso:OutputProcess for LEFT
16:      create cso:InputProcess for RIGHT
17:    end if
18:  end if
19: end if
20: if bp:conversion is uncatalyzed then
21:   if (SPONTANEOUS eq "L-R" or empty) then
22:     create cso:InputProcess for LEFT
23:     create cso:OutputProcess for RIGHT
24:   else if (SPONTANEOUS eq "R-L") then
25:     create cso:OutputProcess for LEFT
26:     create cso:InputProcess for RIGHT
27:   end if
28: end if
```

a process via an association connector or an inhibitor connector, respectively. The "create" function in the algorithm is accompanied by a referral to an entity already defined in the first step, thereby identifying `cso:Entity`.

BioPAX uses the DIRECTION property to indicate the directionality and reversibility of the reaction, using values such as reversible, irreversible-left-to-right, irreversible-right-to-left, physiol-left-to-right, and physiol-right-to-left. If the DIRECTION slot of `bp:catalysis` is defined, then the entities included in the LEFT and the RIGHT slots will be mapped to different connectors in CSO, as shown in

lines 7 to 18. The reversibility of the reaction is specified, for example, "reversible" for the interaction occurring in both directions, and "irreversible-left-to-right" and "irreversible-right-to-left" for the interactions occurring only in the specified direction. In the "reversible" case, the participants in the interaction may be either reactants or products. In CSO, two processes are required for a reversible interaction. For each direction, every participant is linked to one process as an input and to another process as an output, as shown in lines 8 to 10. We consider that "physiol-left-to-right" and "irreversible-left-to-right" imply the same direction from left to right regardless of the reversibility shown by lines 11 to 13. The same assumption is used for the opposite direction, "physiol-right-to-left" and "irreversible-right-to-left," in lines 14 to 16. The class `bp:catalysis` has a slot for the cofactor (not shown in the algorithm). If the cofactor is defined, it will be mapped to an association connector whose biological role is annotated as a cofactor in CSO.

The SPONTANEOUS slot is used to represent a process if it occurs without any external intervention in BioPAX. The lines from 20 to 28 show the case where the conversion interaction is uncatalyzed and the SPONTANEOUS slot is defined. The possible values indicate the direction of the interaction: "L-R" for left-to-right, "R-L" for right-to-left, and "not-spontaneous" for not at all. If the slot value is left empty, the spontaneity is not known. In this case, we assume that the direction is left-to-right. If a conversion is catalyzed, "not-spontaneous" can be used to confirm whether the interaction is controlled by a control interaction.

In BioPAX, LEFT and RIGHT are not used to indicate the direction of a conversion. As shown above, depending on the directionality, LEFT may constitute either substrates or products of a conversion. If the directionality is not specified, we consider that LEFT stores substrates and RIGHT stores products, following the conventions for LEFT and RIGHT of BioPAX [1].

4. Experimental Results

This section describes the preliminary experiments based on the inference algorithms described in Section 3. We have selected the G1/S DNA damage checkpoints pathway in BioPAX from Reactome (ID=69615). For the purpose of reasoning, the Pellet OWL reasoner [8] is used. For obtaining a detailed mapping table between classes in two ontologies, see Supplementary material.

Since the OWL format is intended to be machine-readable, an intuitive, graphical visualization of the interaction is desirable to analyze the given model. In the following figures, we have used simplified notations for brevity. For example, the components of 26S_proteasome, which consists of 41 proteins, are omitted. If the Reactome model uses a complete text-based description for the name, the name is abbreviated; for example, "ubiquitanation" instead of "Ubiquitination_of_phosphorylated_Cdc25A." The layout is modified manually to improve an understanding of the figures.

Figure 2 shows the Reactome BioPAX model loaded in Cytoscape [11]. Cytoscape shows the binary relation between two nodes along with the direction.

The nodes represent instances of `bp:catalysis`, `bp:biochemicalInteraction`, and `bp:physicalEntity`. The edges indicate the slots connecting two instances such as CONTROLLED and LEFT. Since Cytoscape shows the participants at the `bp:physicalEntity` level, the information about cellular location and post-transformation stored in `bp:physicalEntityParticipant` is lost in this view. For example, Unitprot:P30304-1, a boxed protein shown in the top left of Figure 2, is related to three participants: P30304 located in the cytoplasm, P30304 located in the nucleoplasm, and P30304 phosphorylated located in the nucleoplasm. Each participant plays a different role: LEFT of ubiquitination, LEFT of phosphorylation, and RIGHT of phosphorylation, respectively. Similar situations are shown for UniProt:P04637-1 and UniPort:Q00987-1 in the bottom left of the figure.

The converted model in CSO is visualized via the Cell Illustrator [9] in Figure 3. Since CSO encodes physical entities and their related information simultaneously, Unitprot:P30304-1 is correctly represented as three entities depending on their cellular location and modified state in the Cell Illustrator view. Cell Illustrator uses the cellular locations to correctly locate the entities and processes in subcellular compartments. The used icons for the processes are the default images provided by CSO.

In addition, the initial values for simulation are assigned for each process and entity because they are not available in the BioPAX format. The changes in the concentration of the interesting entities along with time are charted. The results of the simulation with charts can be stored in CSO. Any changes such as replacement of images with user-defined images and adjustments to parameters are recorded in CSO. The information encoded in CSO can be reused by any associated tool for pathway analysis, visualization, and simulation with less effort.

During conversion, we detected several problems such as ambiguous and missing information in the Reactome BioPAX. BioPAX has a restriction in that some classes are defined for organizational purposes and should not have instances. Unfortunately, when pathway databases are exported to BioPAX, some concepts could not be converted into BioPAX. Furthermore, there are no proper guidelines for incorporating external databases to BioPAX. In the Reactome example, ubiquitin ligase and cyclin E are defined as `bp:physicalEntity` and not `bp:protein`. Although CSO provides a more detailed hierarchy of classes, it is not easy to identify the entity type, e.g., protein or DNA, without some kind of human intervention.

We detected another ambiguity in a catalyzed interaction. In the lower right region of Figure 2, p21/p27 plays two different roles: CONTROLLER of catalysis and LEFT of inactivation. This case cannot be simulatable because an enzyme does not consume its concentration during an interaction, but a substrate does it. In CSO, the catalyzed inactivation is mapped to one process, and p21/p27 is involved in inactivation via two connectors: an association connector and an input connector. In this case, the interaction can be mapped to a biological fact in CSO because it is not clear which one is inactivated. On the other hand, we interpreted this interaction as two processes by adding one more binding process. In the first process, p21/p27

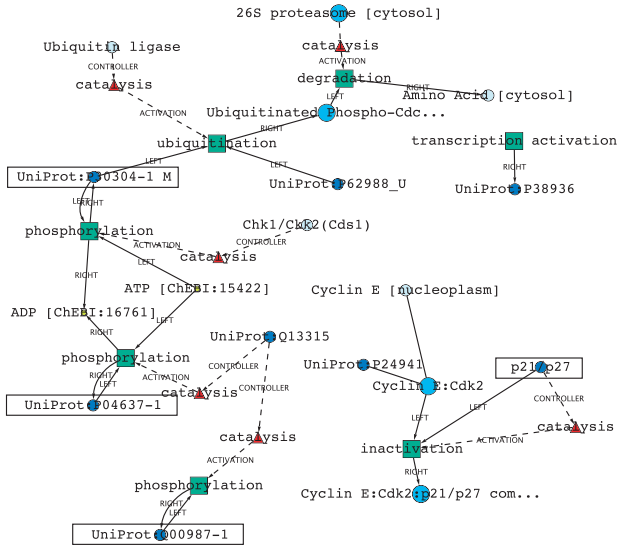

Fig. 2. Cytoscape view. The names of nodes are abbreviated after modification. The visual styles are set manually. Catalysis is denoted by the triangles; conversion, by the rectangles; and physicalEntity, by the circles.

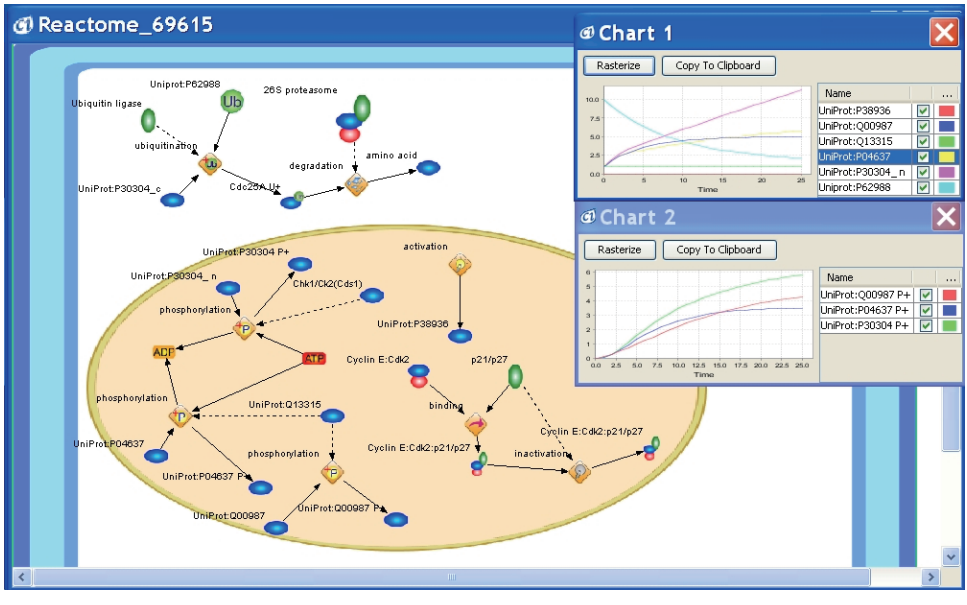

Fig. 3. Cell Illustrator view with simulation results. Graphical visualization is automatically done using the given biological properties and the corresponding standard icons in CSO.

and Cyclin E:Cdk2 bind together and a complex is generated, while in the second process, the resulting complex Cyclin E:Cdk2:p21/p27 is inactivated by p21/p27, as shown in Figure 3.
Another problem is that INTERACTION-TYPE is not defined in the Reactome BioPAX. This property is important to decide the type of process that occurs. In particular, CSO provides the corresponding image for each process and supports its visualization. Generally, if the type of entity or process is not given, an unknown type is assigned. For this experiment, we assign the interaction type from the given instance ID that describes the detailed interaction.

5. Discussion and Conclusion

In this study, we have explored the potential of the cell system ontology (CSO) to integrate biological pathway data with system dynamics and visualization. CSO captures the quantitative aspects of biological functions and equips mature core vocabularies and the corresponding standard icons. These functionalities are important to enhance semantic validation for a given model. The experimental results show how CSO can be used to generate a complete and consistent pathway repository. The conversion generates dynamic models with improved visualization from static ones. The simulations allows to explore the possible dynamic behavior of pathway components and these results might be useful for further investigation of biological systems.
During conversion to CSO, we faced several problems caused by incomplete BioPAX data. First, some important data for visualization, such as the interaction type, is often missing. Therefore, human intervention becomes necessary in order to obtain the correct value, which is time consuming, as the comments have to be read. Secondly, the used terms that have no external references may cause problems. BioPAX uses PSI-MI [13] as controlled vocabulary for the interaction type. For example, "trimerization" is used to describe the interaction that binds three identical molecules in CSO, which is not defined in PSI-MI. However, it is used to describe a binding of three nonidentical molecules in BioPAX. As visualized with the CSO standard icon, this makes a model invalid. Thirdly, as described in Section 3, some concepts could not be converted into BioPAX because BioPAX only supports metabolic pathways and molecular interactions. In the conversion from BioPAX to CSO, it might be necessary to recover the primitive, original meaning used in the external sources.
These abovementioned problems are the challenges we face in the next step of conversion. The ability to check inconsistency and incompleteness in pathways is inevitable for the development of a reliable pathway data repository. In a future work, we will develop more comprehensive inference methods for missing or ambiguous data to reduce human intervention as much as possible.

References

[1] Bader, G. and Cary, M., BioPAX – biological pathways exchange language level 2, version 1.0 documentation, 2005.
[2] Finney, A. and Hucka, M., Systems biology markup language (SBML) Level 2: structures and facilities for model definitions, 2003.
[3] Jeong, E., Nagasaki, M., Saito, A., and Miyano, S., Cell system ontology: representation for modeling, visualizing, and simulating biological pathways, accepted in *In Silico Biology*, 2007.
[4] Kojima, K., Nagasaki, M., Jeong, E., Kato, M., and Miyano, S., An efficient grid layout algorithm for biological networks utilizing various biological attributes, *BMC Bioinformatics*, 8, 76, 2007.
[5] Nagasaki, M., Doi, A., Matsuno, H., and Miyano, S.,Genomic Object Net: I. A platform for modeling and simulating biopathways, *Applied Bioinformatics*, 2:181-184, 2003.
[6] Nagasaki, M., Doi, A., Matsuno, H., and Miyano, S., A versatile Petri net based architecture for modeling and simulation of complex biological processes, *Genome Inform.*, 15(1):180–197, 2004.
[7] Smith, M., Welty, C., and McGuinness, D., OWL Web Ontology Language Guide, 2004
[8] `http://pellet.owldl.com/`, Pellet: The open source OWL DL reasoner.
[9] `http://www.cellillustrator.com/`, Cell Illustrator 3.0.
[10] `http://www.csml.org/`, Cell System Markup Language (CSML).
[11] `http://www.cytoscape.org/`, Cytoscape: analyzing and visualizing network data.
[12] `http://www.libpng.org/pub/png/`, PNG (Portable Network Graphics).
[13] `http://www.psidev.info/index.php?q=node/60`, Molecular interaction XML format documentation.
[14] `http://www.w3.org/TR/SVG11/`, Scalable vector graphics (SVG) 1.1 specification.