A Pathway Editor for Literature-based Knowledge Curation

Ken Ichiro Fukuda **Toshihisa Takagi**

Computational Biology Research Center (CBRC)
National Institute of Advanced Industrial Science and Technology (AIST)
Aomi Frontier B/D 17F, 2-43 Aomi, Koutou-ku, Tokyo 135-0064 JAPAN
Email: fukuda@cbrc.jp

Graduate School of Frontier Sciences, University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8562, Japan

Abstract

As functional-genomics data become available in an ever increasing rate, proper accumulation and dissemination of background knowledge about biological functions of higher order, such as signal transduction pathways, become indispensable. While a graph-analogical representation is typically observed in biological literatures, due to the diversity of topics that the term "signal" covers, the kind of biological entities that constitutes pathways are highly diverse and mixed-up even in a single diagram. As signal transduction pathway describes the underlying molecular mechanisms of phenotypes, biological processes, its component are hierarchically structured and can be decomposed to sub-processes of arbitral granularities. To support the accumulation processes of this type of knowledge, a pathway editor that displays and manipulates pathway knowledge in a strongly structured manner, such as hierarchical recursive structures is necessary.

This paper describes a fully operational implementation of a pathway editor for signal transduction pathways. The system is designed to support curation task for signal transduction pathway knowledge that is buried in biological literatures. The system supports manipulation of attributed compound graphs. An XML format for recursive pathway representation is also presented. The software should be available to the open public from *www.ontology.jp/GEST/*.

Keywords: Complex Data Input, Biological Pathway Modeling, Literature Based Knowledge, Knowledge Representation, XML for biology

1 Introduction

Signal transduction is the mechanism of cellular response against outer stimuli. In general, it describes the actual biological implementation of a biological process, such as the underlying basis of cancer and apoptosis.

Typically, this knowledge is shared through scientific literatures. Rather than storing bunches of lists of relations between molecules, biologists have developed their own "information visualization" method to understand and share the mechanisms of cell behaviors. And that is drawing of diagrams.

In diagrams, each biological pathway is commonly represented as a graph like structure using circles and arrows with some annotations. Figure 1 is an example of mating signal transduction in yeast. (A pheromone binds to a seven transmembrane helix receptor, i.e., GPCR, G-protein coupled receptor, and triggers the downstream processes.) However, due to the diversity of topics that the term "signal" covers, the kind of biological entities that constitutes pathways are highly diverse and mixed-up even in a single biological context. The domain ranges from metal ion, DNA sequence, and protein to phenotype or biological concepts such as cellular response to outer stimuli. Interactions that relate these biological entities are likewise diverse and refer to causality relations of different phenomena, physico-chemical interactions, translocation or secretion of molecules.

Since biological process is an ensemble of molecules, interactions and other processes, each process can be decomposed into an arbitrary level, which refers to a phenomenon or sub-process of different granularities. In literatures, these entities of different granularities may interact directly such as "protein A activates biological process B".

To make this complex knowledge accessible, we need a formal representation model to handle decomposable pathways and support-systems to extract pathway information from articles since curation is very expensive. Therefore, studies of human-computer interfaces to support efficient and precise knowledge accumulation and dissemination have increased its importance. Especially, to extract pathway knowledge from literature, an ontology-aware tool with a graphical user interface (GUI) that provides visualization, editing and processing of pathways is necessary.

In this paper, a fully operational pathway editor called GEST (Graph Editor for Signal Transduction) is described, which is designed to cope with the above mentioned issues. The software handles the intrinsic complexity of signal transduction pathways to support efficient accumulation of pathway knowledge described in literatures and is written in Java so as to provide platform independence. An XML format for decomposable pathway representation to store and share pathway data among researchers and application programs is also briefly introduced.

2 System Overview

As described before, signal transduction pathways are not free-standing entities but an ensemble of multiple entities. Even a pathway itself can be a part of a larger pathway. Hence the interface of a pathway editing tool should particularly focus on how to represent

Copyright ©2004, Australian Computer Society, Inc. This paper appeared at The Second Asia-Pacific Bioinformatics Conference (APBC2004), Dunedin, New Zealand . Conferences in Research and Practice in Information Technology, Vol. 29. Yi-Ping Phoebe Chen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

This work was partly supported by BIRD of Japan Science and Technology Agency (JST) , and a Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

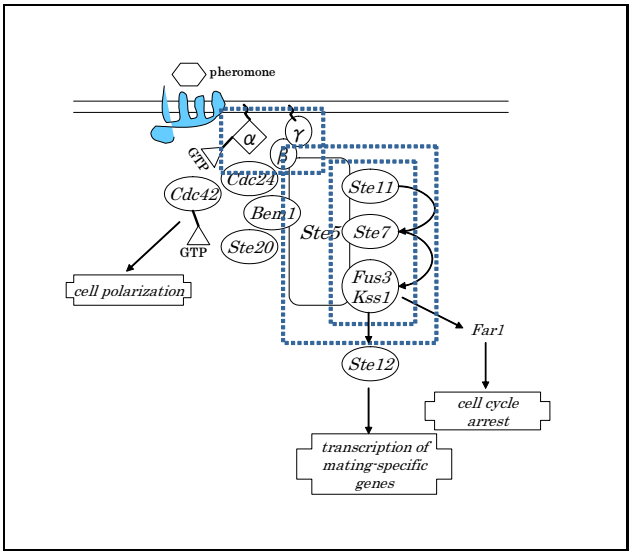

Figure 1: The cell membrane is represented by two lines and ovals represent proteins. By placing these objects intuitively in a diagram, the mechanism of yeast mating signal is represented. This arises at least two problems, (1)implicit mentioning of sub-structures or subprocesses, (2)irregular description of granularity. (1) Biologists know that the diamond and two small circles, α, $\beta\gamma$ are the subunits of a G-protein. Thus they understand that after the binding of a pheromone, the $\beta\gamma$ subunits are released from the α subunit and bind to protein *ste5*. *Ste11*, *Ste7*, *Fus3/Kss1* compose a famous "MAPK cascade" where *Fus3/Kss1*, a MAP Kinase, is phosphorylated by *Ste7*, a MAP Kinase Kinase, which is phosphorylated by a MAP Kinase Kinase Kinase (MAPKKK) *Ste11*. The signal is relayed from *Ste11* to *Fus3* by these phosphorylation reactions. It is also known that *Ste5* serves literally as a scaffold for this MAPK cascade. The rectangles with dotted line show these implicit substructures. (2) The G-protein is represented with three individual objects while other proteins are represented by single objects. The arrow from *Ste11* represents an enzymatic reaction (phosphorylation) between bio-molecules and the arrow from *Ste12* represents causal relation between an activated protein and an abstract phenomena.

and manipulate hierarchical and recursive structures naturally.

The proposed software is based on the compound graph structure(Fukuda 2001), which allows us to describe pathways in a recursively decomposable way.

A compound graph is an extension of a graph definition in which each node can contain a graph inside itself. While a *graph* $G = (V, E)$ is defined by a finite set V of *nodes* and a finite set E of *edges*, a *compound graph* $CG = (G, T)$ is defined as the pairing of an *interaction graph* $G = (V, E^G)$ and a *decomposition tree* $T = (V, E^T, r)$ that share the same set of nodes V. T holds the information whether a node contains a graph inside itself. r is the root of this hierarchy.

The capability of encoding sub-structures that partition a graph into hierarchically organized fragments defined by a decomposition tree T and an interaction graph G provides a formal data structure for the model, which is necessary to represent the intrinsic pathway feature naturally not only for human beings but also for computers. Each node and edge is an object and each object has a set of attributes.

3 Conventional Graph Editor Systems

Although many people explicitly or implicitly use a hierarchical representation for biological systems(Nedeliec 2002), only few address the importance of a formal definition of multi-level abstraction of pathway knowledge(Fukuda 2001),(Demir, Babur, Dogrusoz, Gursoy, Nisanci, Cetin-Atalay, Ozturk 2002).

For example, several graph visualization systems provide recursive folding of subgraphs (McCreary 1999, Salamonsen 1999), which yield a clustered graph [1]. This grouping functionality is a facility to represent a series of nodes and edges within a pathway by a single node and enables pathways to be built as a series of information abstraction levels. However this feature is basically utilized to hide a part of the information for visualization purpose and does not support the manipulation of this graph structure explicitly. It does not define the data structure of this hierarchy of abstraction levels and fails to provide a well-defined file format for these pathways.

GenMAPP(Dahlquist 2002) is a tool to visualize gene expression data on hand-drawn maps representing biological pathways and groupings of genes. It provides a good visual presentation of pathways, but does not have an internal data model. As GenMAPP is a drawing tool, a gap remains between knowledge in the form of drawings and knowledge in a computer accessible form.

Attributed graphs typically get only weak supports from general graph drawing tools. For example, Graphlet's(Himsolt 1996) graph editor cannot edit nor visualize the labels even though its file format, GML, allows storing arbitrary number of labels.

To extract pathway knowledge from scientific articles and to perform biologically important inferences on the knowledge, proper support for attributed compound graphs with a well-defined file format is necessary, which allows our user to provide formal semantics to the edited graph.

4 General Graph Editor Features

Similar to those of many other graph drawing tools (McCreary 1999, Lisitsyn 1999, Himsolt 1996, Mehlhorn 1995), the system interface is designed to be intuitive and simple as shown in Figure 2. The main screen consists of eight main components: a menubar, a toolbar, three palettes each for nodes and edges and experimental results, an editor window, an abstraction-level explorer, and a property window.

4.1 User Interface and Visualization

The editor window is responsible for creating and managing the compound graph interactively and supports basic graph editing operations such as insertion, deletion, copy & paste, node movements, size or color changes, etc. Through the menubar and toolbar, the system provides a mechanism to access various functionalities of the system. Two buttons beside the node palette provide facility to zoom in/out graph image to arbitrary size.

Hierarchy Manipulation One of the most interesting features of GEST is the ability to manipulate hierarchies of compound graph nodes. The abstraction-level explorer shows the decomposition

[1] A *clustered graph* is a compound graph where edges in the graph are allowed only between leaves of the decomposition tree. Another graph model related to compound graphs is a *nested graph* in which edges are only between children of the same parent.

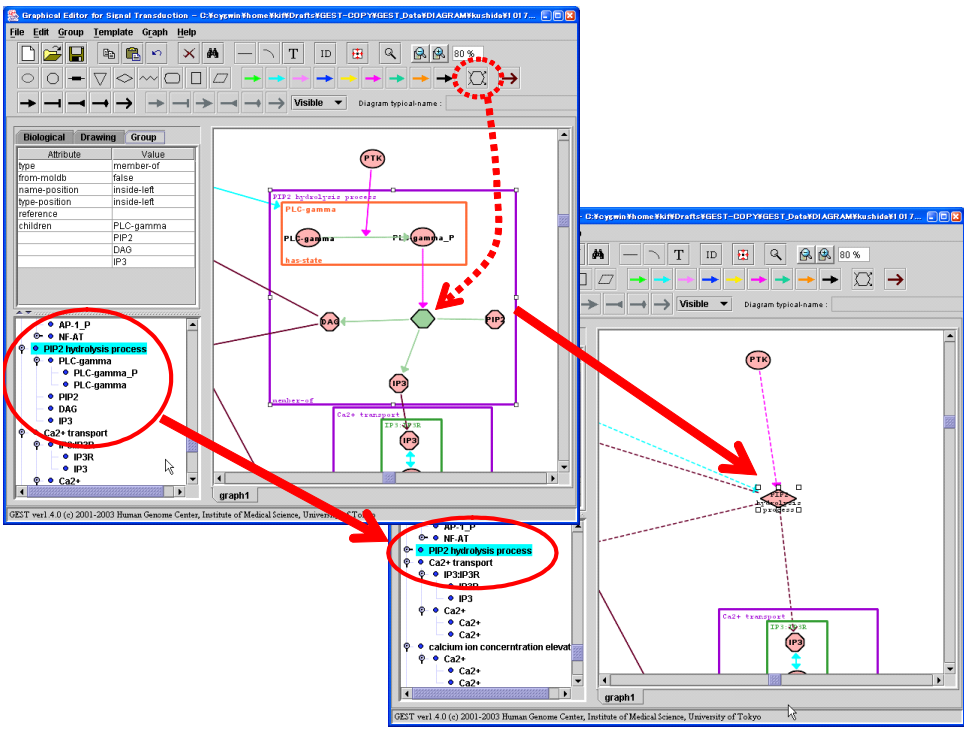

Figure 2: A screen capture of the main screen. "open", "close" operation of compound graphs. By collapsing the tree in the explorer, the associated node collapses in the editor window as shown by the thick arrow. Dotted line shows a many to many edge example (Hydrolysis of PIP2 to IP3 and DAG).

tree of the compound graph presently displayed on the editor window. By opening or closing the nodes in the explorer, users can open or close each sub-structure or sub-process(Figure 2), and can view the pathway through multiple abstraction levels.

Node and Edge Buttons By selecting an appropriate object type from the palettes, the user can create new objects in the editor window. The types of nodes and edges are shown in Figure 3.

To be independent from ontology development, GEST itself does not have a complete ontology to describe biological pathways. Each node and edge has a set of attributes defined by its type, i.e., different node types have different attribute sets, and these attribute slots are "entry points" to the ontologies. Figure 4 is a list of the external ontologies used to annotate each object.

Each edge type corresponds to a concept in the reaction ontology. The "reaction" slot in each edge provides a menu which shows the corresponding lower reaction concepts in the reaction ontology. By selecting the exact concept from the menu, one can specify the appropriate meaning of each edge.

The edge type **Modulate** is an edge object that points to an edge.

For example, to describe phosphorylation of a protein, the user creates a **Metabolysis** edge, which is pointed by a **Modulate** edge. In the reaction ontology, "*metabolysis*" has a child concept "*protein metabolisis*" that has a child "*protein modification process*" that has a child "*state-change phosphorylation*". The concept "*modify phosphorylation*" is under "*modulate reaction*"-"*modulate metabolysis*"-"*modulate protein modification*".

Each node type specifies its meaning by an associated ontology. For example, a Sig-Passing-Node has to fill-in a biological-process ontology concept and a

Shape	Name of Node Type
⬭	Protein-Node
⬡	Chemical-Node
▬	DNA-Node
▽	Pointing-Node
◇	Sig-Passing-Node
⩛	RNA-Node
⬭	Cell-Struct-Node
□	Cell-Node
▱	Phenotype-Node

Color	Interaction Edge Name
Yellow-green	Release
Cyan	Binding
Pink	Gene-expression
Blue	Cleavage
Yellow	Transport
Magenta	Modulate
Green	Metabolysis
Orange	Conformation Change
Black	Default

Color	Decomposition Edge Name
Magenta	Member-of
Green	Composed-of
Orange	Has-state
Blue	On-membrane
Pink	In-organelle
Black	Others

Figure 3: Types of nodes and edges supported in the software.

- [1] Localization
- [2] Signal passing process (bio-process) Ontology
  - Biological functions & molecular interactions
- [3] Signaling molecule Family (signal-family) Ontology
  - Families of proteins classified by functional similarities
    - Smad2,3[regulatory], Smad6,7[inhibitory]
    - ligand, antigen, scaffold protein, transporter
- [4] Species
- [5] Phenotype [MeSH/disease]
- [6] Tissue and Organs [TissueDB, http://tissuedb.ontology.ims.u-tokyo.ac.jp/]
- [7] Molecule DB : protein [SWISS-PROT], protein complex
- [8] Chemical DB [KEGG]
- [9] DNA DB [ATCC]
- [10] Reaction DB
- [11] Cell-line DB
- [12] Reference DB [PubMed]

Figure 4: List of Ontologies for pathway annotation.

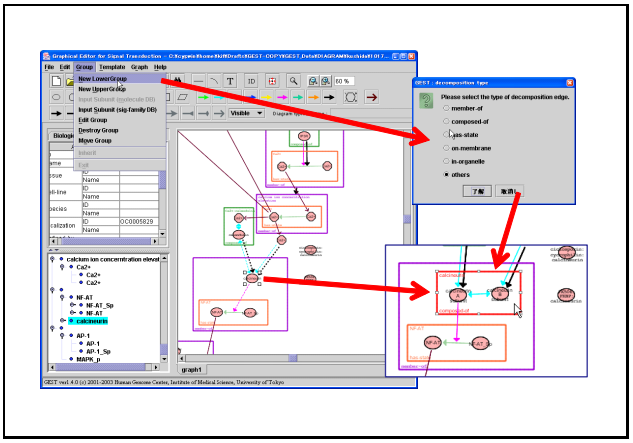

Figure 5: Creating a new tree under "calcineurin". A dialog pops up to specify what decomposition relation should be used to decompose the node "calcineurin".

Cell-Struct-Node requires an ontology concept of a cellular-structure ontology.

Pointing-Node is a node that has a link to another pathway.

Readers should notice that every node may have an underlying compound graph structure if it is an internal node in the decomposition tree. The rectangles ("opened nodes") have different colors according to the type of decomposition edges (Figure 3).

Many to Many Edges To represent a relation such as "hydrolysis of PIP2", a many to many edge is required. This type of edges can be created by clicking the button "intermediate". An intermediate node will appear to create a many to many edge by connecting the source nodes and target nodes with the intermediate node by the associating edge type (Figure 2).

4.2 Manipulation of Data

The system uses three modes to edit three different types of knowledge, i.e., interaction edges, multi-level abstraction and experimental results.

Default Mode "Default mode" is the mode to create/select/delete interaction edges and nodes. Each object has its own attribute values such as name, synonym, localization information, Pubmed ID (PMID)s, structural information, and drawing conditions. A dialog pops up by double clicking an object and the user can edit the values. These values are also editable from the spread-sheet above the explorer.

Group Mode To create an inner node of the decomposition tree or a node that has an underlying sub-structure, ("group node") the user have to enter another mode called "edit group mode". The mode is specially designed for editing the decomposition tree structure. To enter the mode, the user selects **[Group]** menu from the menu bar. If a set of nodes is selected, **[Create UpperGroup]** in the **[Group]** menu creates a new inner node that is the parent of the selected nodes. By selecting **[Create LowerGroup]**, a new tree can be inserted into a leaf of the current decomposition tree (Figure 5).

[Destroy Group] will delete an inner node **a** from the decomposition tree and inserts its children to **a**'s parent node.

Experiment Result Mode Results of biological experiments described in literature are important sources of knowledge, e.g., *"treatment with IFN-γ in Jak1 deficient cell line U4A did not inhibit the activation caused by TGF-β."*

In GEST, this information can be represented separately from the pathway diagram by creating a new tab. To do so, the user selects **[New Graph]** from **[Graph]** in the menu-bar. This creates a new tab in the graph editor window (Figure 6).

Each tab inherits the original compound graph, which is edited in the first tab. This original pathway is editable only in the first tab. In other tabs, the user can add experiment results using intermediate nodes and edges for experiment results. In other words, by creating new tabs, the user is able to describe several sets of experiments.

Experiment results typically include the following information: information about genetic modifications of the cell-lines, a list of chemicals that each cell was treated with, and the corresponding results. For example, in Figure 6, treatment with "**Chem1**" decrease the amount of "**c**", but if combined with over expression of "**a**" (++) then "**Chem1**"does not inhibit "**c**". Numerical information can be added by right clicking each edge.

5 Other Functionalities of the System

5.1 Defining preceding-edges

There are cases where the user needs to define dependencies between interactions(e.g., diagram with loops, or parallel processes). To enable this, each edge in GEST has a slot "preced-edge" to specify the edges that should occur beforehand. As a result, edges in a pathway can be ordered in a semi order according to their dependencies. This information is expected to be useful as constraints in inference of pathways.

5.2 Attribute value propagation

From a practical point of view, filling up attributes for every node and edge is a laborious task when editing a hierarchical graph. To ease this task, GEST can propagate attribute values of a node to its children. If the attribute set of the parent node differs from its child node's attribute set (e.g. when a Sig-Passing-Node has a Protein-Node and a Chemical-Node as its child), only the values of the intersection of the parent node type's and each child node type's attribute sets are propagated.

5.3 Templates

A decomposition tree represents all the elements of a pathway in a hierarchy that reflects how a biologist cognized the pathway. Thus, each pathway fragment represents a biological concept, e.g., a receptor complex, a transcription factor, or a sub-pathway, and so on. Assembling this information in a reusable form will be of great value, since these pathway fragments can be considered as building blocks of signal transduction pathways.

The GEST's template function is a functionality to save these pathway fragments. Template format is defined by a subset of the proposed XML format.

It is expected that by providing adequate names and specifications for the collections of frequently used templates, "controlled vocabularies" or "canonical forms" of signal transduction pathways become available. Each vocabulary will not only provide names but also the structural organization of its components. Using templates as building blocks will support the user to build canonical pathways.

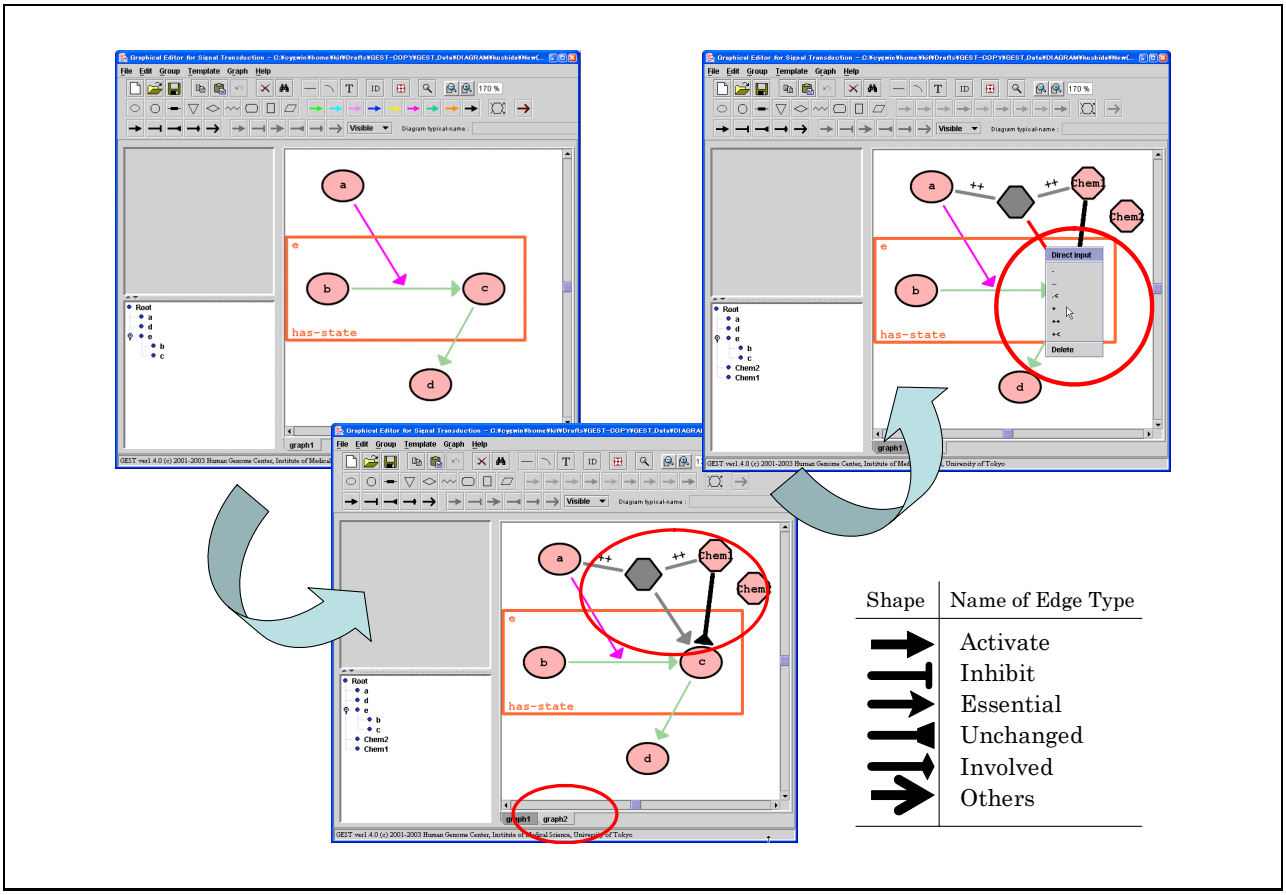

Figure 6: Adding experimental results and the types of edges. The compound graph in tab "graph1" is the original pathway. "graph2" inherits a copy of the original pathway and can add nodes and edges to describe experimental results.

6 File Format

A format for signal transduction pathway data must handle two kinds of information: topological information of diagrams and biological information of pathways.

Graph drawing softwares are continually being developed and almost all of them have their own file format. While there are efforts to solve this problem(Himsolt GML), there are no widely accepted formats for attributed compound graphs.

Thus a new graph representation format based on XML was developed, which offers features suitable for data sharing of pathways and their sub-pathways. XML makes the data application independent and makes it simple to parse and to write with standard editors and other software. The actual specification of the XML format for signal transduction pathways can be found in (GEST 2003).

Basically, the structure resembles to that of GML(Himsolt GML) but has been extended for compound graphs and to cope with biological objects. All attribute values can be edited from the GUI system presented in this paper.

6.1 Representation of Compound Graphs

The format consists from three major parts. Firstly, the root element holds information such as version number, date of creation, name of author, name of the pathway, etc., and the *graph* element.

Secondly, the graph element lists *node* elements, *interaction edge* elements, and *decomposition edge* elements. Each node and edge has an object ID, which is specified as an attribute in each tag.

Thirdly, a node element contains elements that represent biological information of the node such as name, protein family name, localization, modification sites, cell-line, etc. and a *graphics* element that specifies graphical information and coordinates. An edge element holds information about its source node, target node or target edge, the type of edges, and so on.

7 Discussion and Future Directions

GEST was tested in a pathway database project to curate diagrams from review articles. The diagrams can be queried by a system called FREX(FREX 2003). The curated diagrams typically contain around 70 nodes and their decomposition depths are about 5. Compared to a conventional editor based curation process, the cost was reduced a lot as the hierarchy information is managed through the GUI.

GEST is based on a qualitative model and is designed for literature-based pathway knowledge curation. Pathway models can be characterized by at least two criteria. The first one is whether the model is qualitative or quantitative. The second one is the resolution of system specification. Simulation models are specified in high resolution, usually in the level of chemical reactions with numerical parameters. CellML(CellML 2003) and SBML(Hucka 2003) provide a specification language for these models.

The counter part to these simulation models is higher-order knowledge and background knowledge shared through literatures. A simulation model has to describe a working system, but knowledge in literatures does not. Therefore, even though higher-order knowledge contains certain structures to be called

pathways, it does not necessarily specify a system in the same resolution as simulation models. For example, there are interactions that can only be specified in low resolution, such as "protein A was essential for process B".

The size of pathways reported in literatures may vary a lot. Some pathways are small fragmented pathways and others can be very rich. Some pathways may conflict with other pathways. One good feature of the compound graph model is its recursive structure, i.e., each node of a compound graph is also a compound graph that represents a concept of different granularity. In other words, a formal definition of multi-abstraction level provides a methodology to split big pathways into meaningful chunks and to construct a big pathway from smaller sub-processes. This will help the user to manage their own "reference pathway set". Extending further the functionality of template management and implementing rules that detect conflicts between processes will provide strong support for pathway database users.

Unfortunately the current GEST is designed to represent pathway knowledge "as is" described in literatures and inference mechanisms such as knowledge update remain to be implemented. As GEST data are strongly structured with ontological annotations one possible direction is to apply logic based inferences(Fukuda 2001).

Although GEST has an explicit definition about its hierarchical representation model, this definition does not define a canonical unique representation of pathways. As mentioned before, we expect that templates will become a kind of "extensional" definition of canonical process representations.

Considering the huge variety of topics discussed in pathway informatics, there is currently no unified standard for pathway representation that suffices each specific pathway project. For example, designing a data exchange format between these pathway representations addresses open questions such as how to map qualitative pathway representation to quantitative pathway representation. These issues should be discussed in a community based consortium such as BioPaX(BioPAX 2002).

Other future directions include implementation of graph embedding algorithms specially designed for compound graphs(Giuseppe 1994, Bertault 1999). This would be beneficial because with an embedder the system can import XML files without coordinate data. Currently the system supposes that every node in the input file has an appropriate coordinate that was defined by the system through its GUI.

8 Conclusions

In this paper, a fully operational implementation of an attributed compound graph editor is presented. The system focuses on visualization and data manipulation of pathway knowledge and incorporates several novel functionalities to support users in extracting pathway knowledge from literatures. The file format is defined by XML and handles multi-level abstraction of signal transduction pathway description.

Acknowledgments

We would like to thank Shiho Tada of Hitachi ULSI Systems Co., Ltd. Advanced Technology Development Dept. Life Science System Development Gr.

References

Ken ichiro Fukuda & Toshihisa Takagi (2001), 'Knowledge Representation of Signal Transduction Pathways', *Bioinformatics*, **17**(9), 829–837.

Claire Nedellec (2002), 'Bibliographical Information Extraction in Genomics', *IEEE Intelligent Systems*, MAY/JUNE, 76–80.

E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, G. Nisanci, R. Cetin-Atalay & M. Ozturk. (2002), 'PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways', *Bioinformatics*, **18**(7), 996–1003.

C. McCreary and L. Barowski. (1999), 'VGJ: Visualizing graphs through java', *in* 'Proc. of Graph Drawing '98', Springer-Verlag', 454–455.

Wayne Salamonsen, Kevin Yee Chuen Mok, and Prasanna Kolatkar. (1999), 'BioJAKE: A Tool for the Creation, Visualization and Manipulation of Metabolic Pathways', *in* 'Pacific Symposium on Biocomputing 4', 392–400.

Dahlquist, K.D., Salomonis, N., Vranizan, K., Lawlor, S.C., Conklin, B.R. (2002), 'GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways, *Nat Genet*, May;**31**(1), 19–20.

M. Himsolt. (1996), 'The Graphlet System (system demonstration)', *in* 'Proc. of Graph Drawing '96', Berline Heidelberg, Springer-Verlag, 233–240.

Ivan A. Lisitsyn and Victor N. Kasyanov. (1999), 'Higres – Visualization System for Clustered Graphs and Graph Algorithms', *in* 'Proc. of Graph Drawing '99', Berline Heidelberg, Springer-Verlag, 82–89.

L. Mehlhorn and S. Naher. (1995), 'LEDA: a platform for combinatorial and geometric computing', *ACM*, **38**, 96–102.

M. Himsolt. (1997), Manuscript, University Passau, Germany, Available at http://infosun.fmi.uni-passau.de/Graphlet/GML/.

GEST, http://www.ontology.jp/GEST/index.html.

FREX, http://www.ontology.jp/FREX/index.html.

CellML, http://www.cellml.org/.

M. Hucka, et al. (2003), 'The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models' *Bioinformatics*, **19**(4), 524-531.

Ken ichiro Fukuda and Toshihisa Takagi, (2001), 'Signal Transduction Pathways and Logical Inferences', *in* 'Proc. of The 2001 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences, METMBS '2001', 297–303.

BioPAX, http://www.biopax.org/.

Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis (1994), 'Algorithms for drawing graphs: an annotated bibliography' *Computational Geometry: theory and applications*, *4*(5), 197–204.

Francois Bertault and Mirka Miller, (1999), 'An Algorithm for Drawing Compound Graph', *in* 'Graph Drawing, 7th International Symposium,GD'99', Springer, September, 197–204.