



ELSEVIER

Available online at www.sciencedirect.com



ScienceDirect

Electronic Notes in
Theoretical Computer
Science

Electronic Notes in Theoretical Computer Science 194 (2008) 149–164

www.elsevier.com/locate/entcs

Signal Transduction Pathways as Concurrent Reactive Systems: A Modeling and Simulation Approach Using LSCs and the Play-Engine

Claudia Täubner,^{1,2} Silke Eckstein³

*Institute of Information Systems
Technical University of Braunschweig
Germany*

Abstract

Signal transduction pathways describe how cells respond to extracellular signals which are received by receptors at the cell membrane and usually transferred into the nucleus. In this paper we present our approach to model these signal transduction pathways as concurrent reactive systems by means of Life Sequence Charts and to simulate them using the Play-Engine tool. This aspect is part of a bigger approach, where we provide an extendable system to generate models of signal transduction pathways in different modeling languages and to simulate these models with the corresponding simulation tools.

Keywords: Concurrent Reactive System, Signal Transduction Pathway, Life Sequence Chart, Play-Engine, Colored Petri Net

1 Introduction

Signal transduction pathways describe how signals received by receptors at the cell membrane are processed inside the cell via biochemical reactions. Usually, the signal is transferred into the nucleus, where it causes a change in the currently active genetic program of the cell. Understanding this flow of information inside a cell is fundamental for an in-depth understanding of the functioning of a cell as a whole. Modeling and simulating this information flow is beneficial, because it helps to understand the flow of signals in a complex network, to test hypotheses *in silico* before validating them with experiments and to validate the data collected about a certain pathway.

¹ partially supported by the German BMBF, “Intergenomics” (031U210C).

² Email: taeubner@ifis.cs.tu-bs.de

³ Email: eckstein@ifis.cs.tu-bs.de

The fact that a flow of information in a complex network is to be described has led to the idea of applying languages for the description of concurrent reactive systems in this area, even if these were originally developed to assist the construction or engineering of systems and not the description of already existing systems [19,11]. A couple of specification languages, e.g. Petri Nets, Live Sequence Charts etc. qualify for this task, all bringing different advantages and disadvantages with them. In the same way the corresponding simulation tools have different strengths and shortcomings.

In systems biology several modeling and simulation approaches are being applied [23]. Most of them can be classified according to the following three dimensions [38]: discrete and continuous, qualitative and quantitative as well as deterministic and stochastic. When examining biological pathways, the considered events are discrete but the other two dimensions are not determined. Especially various kinds of Petri Nets have been applied in this area [10,30], ranging from Colored Petri Nets describing metabolic pathways qualitatively [14] to Stochastic Petri Nets, which allow a quantitative modeling, and Hybrid Functional Petri Nets quantitatively representing metabolic pathways [27]. In [34] an approach called Pathway Logic is presented, where rewriting logic is used to model signal transduction. The π -calculus is being to model signal transduction pathways and gene regulatory networks [33,32]. Live Sequence Charts are being applied to model a certain part in the development of the worm *C.elegans* [20] and UML-Statecharts to model the activation of T-cells [5,6] and the signaling pathway of *Arabidopsis thaliana* [36].

Considering the different approaches discussed above, it would certainly be beneficial to develop different models for one signal transduction pathway using different specification languages and to be thereby able to apply different simulation tools. However, this is only possible if the models do not have to be developed manually but can be generated out of a source for pathway data.

There exist a number of pathway databases compiling the current knowledge about biological pathways. Thus, we developed an extendable system to generate different models, expressed with different modeling languages, out of the data stored in pathway databases in order to simulate these models with the corresponding simulation tools. What most of the approaches discussed above have in common is, that the models are constructed manually and not generated out of a database, and in those approaches, where the models are derived out of one or more databases (e.g. [27]), only one kind of models is generated. This is in contrast to our approach, where we offer the possibility to generate different representations for a certain pathway. At the moment we operate on one of the major pathway databases, TRANSPATH[®] [24], and use Live Sequence Charts [2] and Colored Petri Nets [16] as specification languages. The belonging simulation tools are Play-Engine [12] and CPN Tools [17], respectively. The application of UML-Statecharts [29] together with the Rhapsody tool [37] is in preparation.

In this paper we are going to focus Live Sequence Charts and the Play-Engine tool for specification and simulation of signal transduction pathways. Whereas our general architecture and the CPN modeling aspects are introduced only very briefly

and are explained in more detail in [4]. We will discuss what distinguishes LSCs and the Play-Engine tool from other languages and tools and why we chose them for our pathway modeling approach. Furthermore, we will explain how we connected the Play-Engine tool to a pathway database and what kind of tools we developed for this purpose. Furthermore, we present some first simulation results.

The paper is organized as follows: In section 2 we present some basic information on signal transduction pathways and discuss a pathway exemplar. Then we introduce our modeling and simulation approach for signal transduction pathways, thereby concentrating on LSCs and the Play-Engine. In the fourth section we give technical details of our approach and discuss the tools we developed in this context. In section five we present first simulation results. Finally, we draw some conclusions and an give outlook on our future work.

2 Signal Transduction Pathways

In organisms proteins have a wide variety of functions and they interact with each other in likewise multifaceted ways. These interactions of proteins are being described by means of pathways or networks, which are typically represented as certain kinds of maps. A distinction is drawn between metabolic and regulatory pathways, whereas metabolic pathways describe the conversion of classes of substances into other classes of substances, regulatory pathways describe how the function of something is regulated. Here, not the conversion of classes of substances into other classes but the transduction of signals is significant (cf. Fig. 1). This is the reason why they are also called signal transduction pathways.

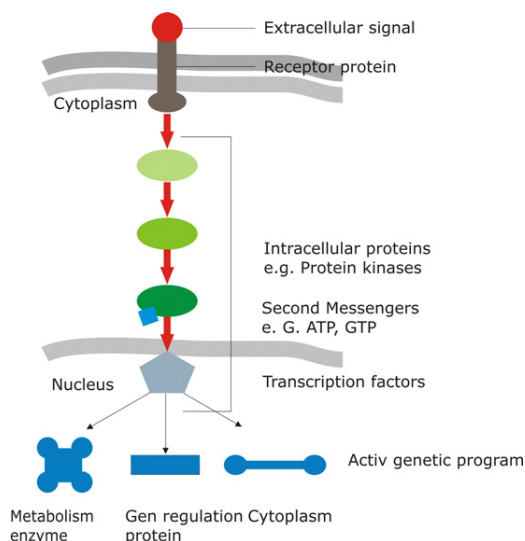


Fig. 1. Signal Transduction (cf. [1])

Expressed differently: a signal transduction pathway describes how a cell responds to an extracellular signal, e.g. a signaling molecule excreted by a bacterium. The signaling molecule is received by a receptor protein and then transferred via

Examples for such signal transduction pathway databases are TRANSPATH[®] [24], KEGG [21], Reactome [18] and BioCyc [22]. They usually provide a web interface for interactive searches and also make their data available as text files in flat file or XML format. Some of the databases already use a more or less standardized exchange format on XML basis, e.g. SBML (Systems Biology Markup Language, [15]).

The database we chose to start with was TRANSPATH[®], which provides the accession methods described above but uses its own XML format. In January 2007 TRANSPATH[®] contained entries about 60,000 molecules, 100,000 chemical reactions, 20,000 genes and 57 pathways. The information was based on 30,000 publications. The web interface provides access to all these entries and in addition contains interactive maps, which give an overview over a certain pathway (cf. Fig. 2). The XML version of the database is divided into 6 files containing data about molecules, genes, reactions, pathways, annotations and references, respectively. They are accompanied by a DTD describing the structure of the files.

In order to give a better understanding of what signal transduction pathways are about, we discuss the TLR4 pathway as an example:

Sepsis is the systemic immune response to a severe bacterial infection [26]. We are born with a functioning innate immunity system which recognizes bacterial and viral products. In sepsis, when a bacterium attacks an endothelial cell, different kinds of mechanisms are activated. Receptors of the innate immune system are activated by microbial components such as LPS (endotoxin, lipopolysaccharide), which is a signaling molecule involved in the initiation of the sepsis syndrom. Receptors, which recognize such LPS molecules, are a family of transmembran receptors known as Toll-like receptors (TLRs). To date, there are 12 TLRs identified in mice and 10 TLRs identified in humans. TLR4 is one of them and is identified as a significant receptor in mice strain experiments. TLR4 is also annotated in TRANSPATH[®] and is our example for signal transduction pathway explanations (see Fig. 2).

The TLR4 signal transduction pathway is subdivided into the MyD88-dependent and MyD88-independent pathway and consist of four chains. The whole pathway is specified in [3]. In order to give an overview over the general flow of information, it is sufficient to explain one part in detail.

One chain of the MyD88-dependent pathway in endothelial cells starts with the LPS binding to the TLR4 receptor complex consisting of CD14 and MD2. This molecule complex is leading to the recruitment of the adaptor molecules MyD88 and TIRAP. Following, IRAK and IRAK-4 are recruited to the receptor complex via interaction of special parts of their spatial arrangements. IRAK recruits and activates TRAF6 which is one part of a molecule complex consists of ECSIT and MEKK1. This recruitment also leads to the activation of IKK α and IKK β which are molecules of a complex with two IKK γ molecules. The activation of the IKK-complex leads to the degradation of I κ b. This inhibition of I κ b facilitates the translocation of NF- κ B in the nucleus. NF- κ B is a transcription factor which connects with its special promoter region. This results in the expression of proinflammatory molecules. Furthermore, normal physiological functions of the endothelial cells are severely

damaged. This bacterial sepsis and its associated expression of proinflammatory molecules causes death with the utmost probability.

3 Modeling Signal Transduction Pathways with Live Sequence Charts

Signal transduction pathways as presented in the previous section sum up the current knowledge about the flow of information inside a cell that is provoked by an extracellular signal. Languages developed for the engineering of concurrent reactive systems are notably suitable to describe this flow of information. Live Sequence Charts (LSCs) is a scenario-based visual formalism to capture the behavioral requirements of a system. Together with the supporting tool this language offers the possibility to exploit and analyze the available information in an intuitive way. The language is powerful although due to its graphical representation it is easy to comprehend, at least in its basic structure. This even holds for non-experts.

Thus, this approach is particularly appropriate for biologists, who want to gain first experiences with modeling and simulation of biological systems. Other approaches like e.g. Petri Nets require a much longer training period, even though they also have graphical representations. Even more sophisticated are formalisms, which only have a textual representation, as e.g. π -calculus [31]. Various kinds of Petri Nets as well as the π -calculus, have proven suitable for modeling biological pathways.

Besides, LSCs are particular suitable for modeling the cascade-like events in signal transduction pathways as we will see in section 3.2 after introducing LSCs and the Play-Engine.

3.1 Live Sequence Charts and the Play-Engine Tool

The Live Sequence Charts formalism [2] was developed as an extension of Message Sequence Charts, also known as UML sequence diagrams. In all of these languages objects are represented as rectangles with the object name inside and a vertical line beneath, also called live line. On these live lines sequences of events and message passing activities occur and the messages between the objects are depicted as horizontal arrows from the sending to the receiving object.

In LSCs a distinction is made between universal and existential charts, whereas the former describes behavior that may happen, the latter describe behavior that must happen. Universal charts contain precharts, which describe the condition under which the behavior described in the main chart has to occur. Furthermore, forbidden behavior can also be expressed.

In [12,13] a tool and a methodology to exploit and analyze the available information are introduced, the Play-Engine and the play-in/play-out approach, respectively. In order to apply the tool and the development process, the reactive system to be described must be represented via a graphical user interface which the developer has to construct or at least to prototype at first. This interface is used in

the play-in part of the process to specify desired, obligatory or forbidden scenarios as well as the desired reactions of the system and the conditions under which certain things may happen. The Play-Engine constructs Live Sequence Charts out of these scenarios. In our approach we will not play-in the behavior of the pathways but generate the corresponding LSCs out of the pathway data available e.g. in the TRANSPATH[®] database. However, when modifying the pathways in order to do in-silico experiments, we use the play-in mode.

During play-out the user can access the GUI in the way he would access the system itself and the behavior specified during play-in is executed. Simultaneously, the Play-Engine uses the GUI to reflect the system states. In our context the play-out mode is interesting in two ways: Firstly, simulating the pathways as they are stored in the TRANSPATH[®] database - i.e. simulating the generated LSCs - helps the user understand the information flow of the pathway and furthermore makes it possible to find inconsistencies in the TRANSPATH[®] data (see Sec. 5). Secondly, modifying a pathway during play-in and analyzing the modified behavior during play-out offers the possibility to do some experiments in-silico and to decide, depending on the results, which of them shall be undertaken in-vitro, i.e. in the lab.

Live Sequence Charts, the Play-Engine and the play-in/play-out approach have already been applied to model a biological system: In [20] the process of cell fate acquisition during *C. elegans* vulva development was modeled and simulated.

3.2 Mapping Signal Transduction Pathways to LSCs

As described above, in the Play-Engine every system that is developed is represented by a Graphical User Interface. Our idea here is to use the pathway maps provided by TRANSPATH[®] (cf. Fig. 2) as GUIs. Consequently, we will be able to animate these maps according to the simulation of the pathway. This means that the biologists will be able to use a simulation tool, which displays the simulation run in their own graphical language. In Fig. 3 you can see a fragment of the TLR4 pathway in LSC representation together with its corresponding map functioning as a GUI. Below we are going to explain how we map the components of signal transduction pathways to LSC concepts.

In Fig. 4 a class diagram is shown, which represents pathways, as they are stored in TRANSPATH[®], in an abstract way. Each pathway consists of a number of chains of reactions, which are concurrent to each other. Molecules participate in reactions as reactants (or educts), enzymes, inhibitors and products. One can also say that the presence of the reactants of a chemical reaction activates the product. Some reactions can only take place if a certain enzyme is available, and others can not take place, if a certain inhibitor is available. Molecules can be members of molecule complexes which are treated as molecules themselves. Molecules can also be in different states, e.g. ubiquitinated or not. In TRANSPATH[®] these different states are represented with two entries in class molecule and a reference (state of) between the two instances.

When mapping a pathway into LSCs the pathway chains are natural points for decomposition into several charts, one for each chain. The molecules are the objects

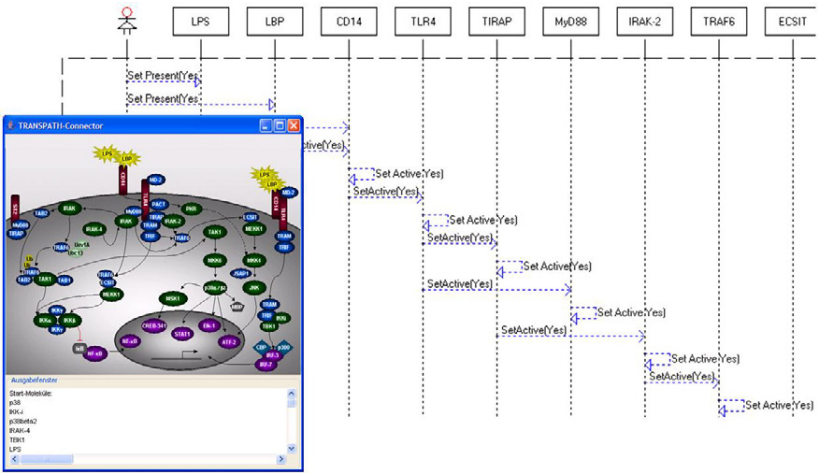


Fig. 3. Fragment of the TLR4 pathway in LSC representation.

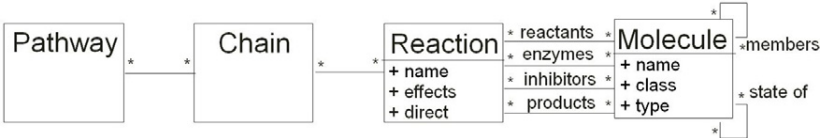


Fig. 4. Representation of pathways as a class diagram.

our system consists of and their reactions are captured through messages and events.

Figure 3 demonstrates that LSCs are particularly suitable to model the cascade-like events in signal transduction pathways. It shows the beginning of a chain of the TLR4 pathway, where the signaling molecule LPS binds with aid of LBP to the receptor complex of TLR4 consisting of CD14 and MD2. Afterwards the TLR4 molecule is activated and in turn activates TIRAP and MyD88 et cetera.

To discuss the mapping in a more detailed way, we will exemplarily look at a reaction between two molecules, which is inhibited, if a third molecule is present (cf. Fig. 5): If the presence of a molecule causes the activation of another molecule, a message is sent from the activating to the activated molecule and then an activation event happens in the receiver molecule. In the example in Fig. 5 $I\kappa B$ activates $NF-\kappa B$. This reaction can only occur, if the inhibitor $IKK\beta$ is not present. In the corresponding LSC this is reflected by $IKK\beta$ not being active. The inhibition is implemented through a forbidden element – $IKK\beta.active = Yes$ in our example – of this universal chart, which prevents the chart from being executed if the condition is true.

3.3 Alternative Approach: Modeling Signal Transduction Pathways as CPNs

In this section we give a brief overview over our representation of signal transduction pathways with Colored Petri Nets (CPNs). More details can be found in [4]. In contrast to regular Petri Nets [9] in Colored Petri Nets the tokens are distinguishable [16]. Each token has a certain value belonging to a certain data type. The values of the tokens are called colors and the data types color sets. Each place must

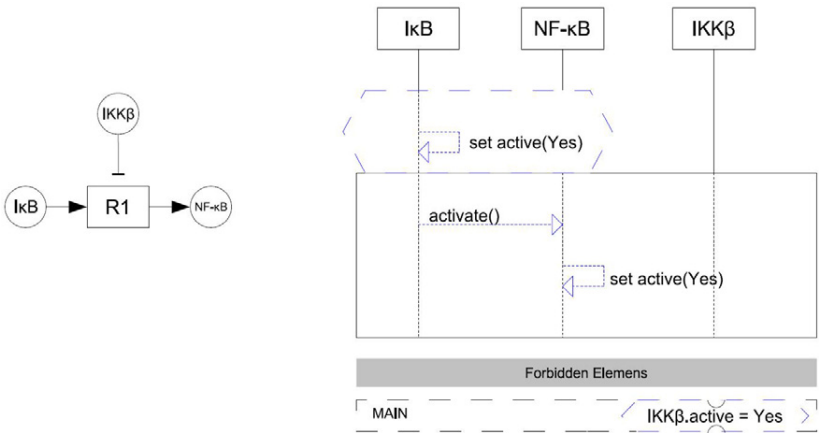


Fig. 5. Mapping a reaction with inhibition to LSC concepts.

carry the information, which kind of tokens it accepts, i. e. tokens of which color (data type).

In our approach, the tokens represent molecules and molecule complexes. Each place can therefore carry certain molecules or molecule complexes. Transitions represent chemical reactions. The incoming and outgoing arcs of a transition specify together with their weights, how many molecules of which kind function as educts, enzymes, inhibitors or products, respectively. Figure 6 contains a subnet of the Petri Net representation of the TLR4 pathway, which was shown in Fig. 2.

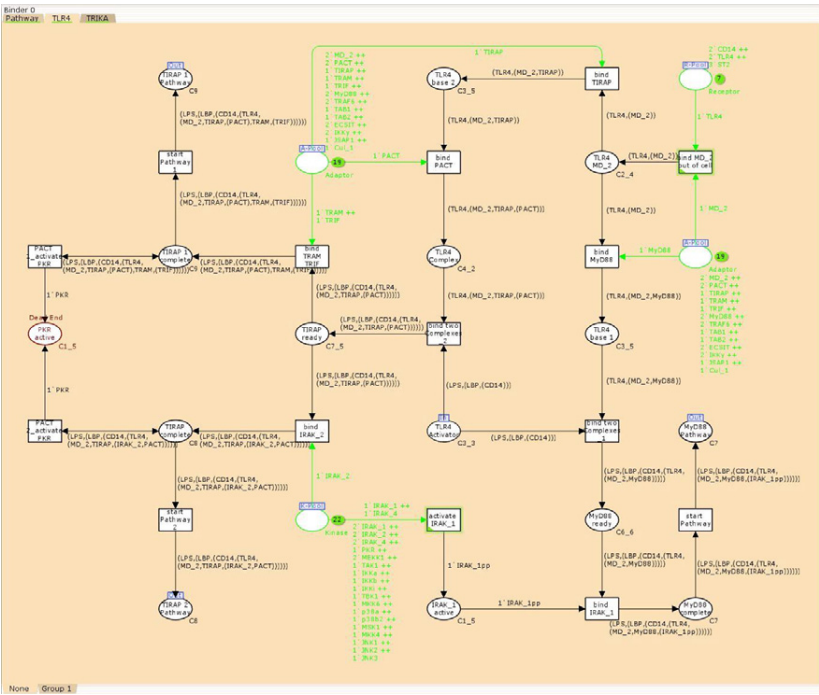


Fig. 6. Subnet of the TLR4 CPN.

To gain some experiences in how to model a signal transduction pathway with

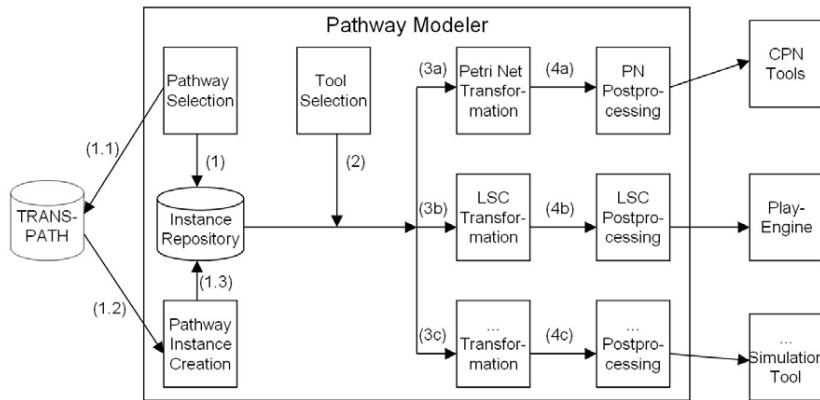


Fig. 7. Architecture and main workflow

Colored Petri Nets we began with a manual transformation [35,8] and out of that we derived the assignment of concepts from signal transduction pathways to CPN concepts. That transformation was based on numerous discussions with biologists, among them one member of the TRANSPATH[®] team.

4 Simulating Signal Transduction Pathways with the Play-Engine

After having introduced our approach from a conceptual point of view in the previous section, we are now going to present the tools developed in this context and to discuss some more technical aspects.

4.1 Architecture and Workflow

The main architecture of our system “Pathway Modeler” is presented in Fig. 7. We will explain it by discussing the main workflow.

Initially, the user has to select the pathway he wants to simulate. The system checks whether it is already available in the Instance Repository (1). If not, it collects all data belonging to the pathway out of the TRANSPATH[®] XML files and creates a pathway instance (1.1 - 1.3). Afterwards, the user selects the simulation tool he wants to generate a pathway model for (2). Presently, CPN Tools and the Play-Engine are available. Depending on the choice, a Petri Net representation or a Live Sequence Chart representation of the selected pathway are constructed (3a/3b). The system is prepared for further simulation tools to be integrated (3c). After the transformation is done, some postprocessing steps follow. Finally, an input file for the selected simulation tool is the result of the transformation process.

As we are concentrating on the LSC aspects in this paper we are now zooming down into this part of the system. Figure 8 represents the tools used and developed in this context as well as their connection through import and export relationships.

On the left hand side of the figure there is the TRANSPATH[®] database which is our source for pathway data. Besides the actual pathway data represented in XML

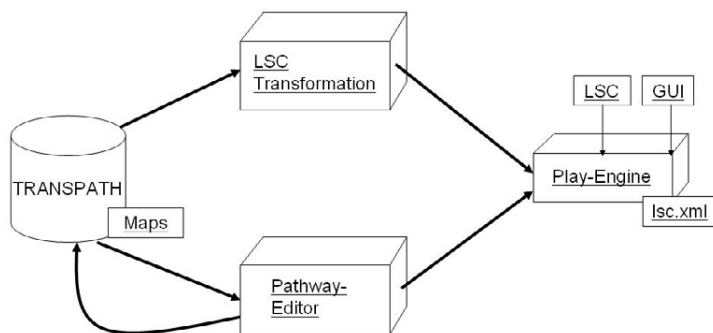


Fig. 8. Pathway-Editor and LSC Transformation module

files it contains pathway maps in GIF format (cf. Sec. 2). On the right hand side there is the Play-Engine tool, which needs a GUI for the system being described, capturing the systems behavior in LSCs and storing these LSCs in XML files. The database and the simulation tool are connected via two tools or modules: the LSC Transformation module and the Pathway-Editor.

4.2 LSC Transformation Module

As input the LSC Transformation module gets a pathway instance corresponding to the class diagram in Fig. 4. It constructs out of the pathway instances LSC instances represented in the XML format the Play-Engine uses to store its LSCs in. In order to enable this, we analyzed various LSCs as they were stored by the Play-Engine and generated out of this data a DTD describing the LSC storage format of the Play-Engine. Thus, we are now able to generate our own LSCs independently from the play-in process, store them in the place in the file tree where the Play-Engine stores its LSCs and to simulate them using the Play-Engine.

The construction of the LSC is done according to the mapping described in the previous section.

4.3 Pathway-Editor

The Pathway-Editor [36] assists the user in a number of ways to build GUIs for the Play-Engine. Export in other file formats is also supported. There are three main tasks the Pathway-Editor has to fulfill: (1) Construction of pathway maps as GUIs for the Play-Engine. (2) Export of these maps in GIF format for use in the TRANSPATH® database. (3) Automatic generation of such a pathway GUI out of the pathway data stored in TRANSPATH®.

For the first task the Pathway-Editor provides menus and tool bars, which contain all functionalities to draw signal transduction pathways and to attach further information to the elements and components of the pathways. When creating the library for the graphical elements provided by the Pathway-Editor we stayed as close as possible to the TRANSPATH® legend for the graphical elements used in the pathway maps. In Figure 9 a screenshot of the Pathway-Editor is shown. By positioning the mouse over an object of interest the attached information is shown

in the menu on the right hand side.

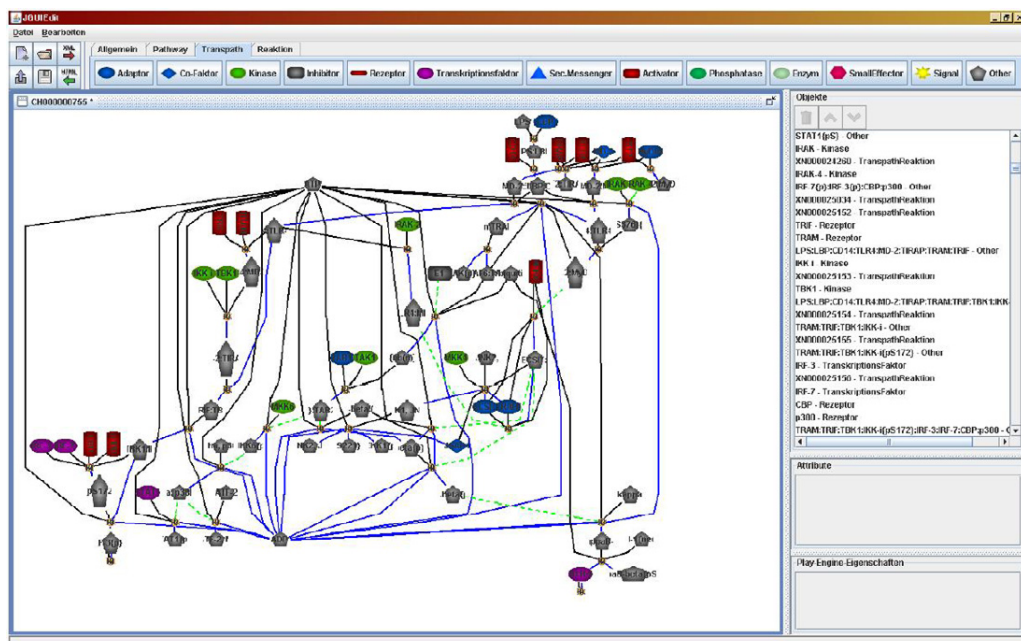


Fig. 9. Pathway-Editor displaying a generated pathway map.

The second task was easy to implement as only the graphical part (the pathway map) had to be exported in GIF file format. So far, the database annotators of Biobase have used a standard graphic tool to create the pathway maps without a library assisting their task. When using the Pathway-Editor they have all graphical elements a pathway map consists of readily available and the creation of maps will thus be much more comfortable.

The third task is based on the following idea: When generating LSCs automatically out of the pathway data, we also need the accompanying GUI to be automatically generated. The first idea of importing the pathway maps from the TRANSPATH[®] database was abandoned because of technical difficulties regarding the recognition of the arrows representing the reactions between molecules. We thus decided to generate the maps directly out of the pathway data. This implies, that the elements of the map have to be positioned automatically, for which we are using a placing algorithm out of the Graphviz package [7]. TRANSPATH[®] provides data about molecules and their biochemical reactions but no information about compartments of the cell like nucleus, cell membrane and mitochondria. Hence, our algorithm can not draw them automatically, but we provide the according graphical elements in order to enable the user to complete the maps. Afterwards, the positioning can of course be further enhanced manually. According to points (1) and (2) these generated maps can be exported as GUIs for the Play-Engine as well as GIF files for the TRANSPATH[®] database.

A benefit of this procedure is, ability to generate pathway maps which reflect the current level of knowledge about the pathway at hand, whereas the maps contained

in the TRANSPATH[®] are adjusted to the current level of knowledge only once in a while.

5 Results

There are two kinds of results we are going to report on in this section: Firstly, we are going to discuss our experiences with using LSCs and the Play-Engine for modeling and simulating signal transduction pathways. Secondly, we are going to present first simulation results.

Live Sequence Charts is a formalism that was developed to describe concurrent reactive systems, a field of application, which perfectly suits signal transduction pathways. The graphical formalism, though powerful, is in its basic form easy to learn and understand, even for non-experts.

The fact, that the Play-Engine needs a GUI representing the system to be described, has led to the effect, that we can provide an access for the biologists to this simulation tool, which uses their own language: the graphical pathway maps. The Play-Engine thus enables us to animate these maps and led the biologists watch their pathways being "executed". Hence, the system reflects the way biologists represent pathways. It is intuitive for anyone who works with TRANSPATH[®] signal transduction pathways. The Pathway-Editor developed in this context enables us to generate pathway maps which reflect the current level of knowledge about pathway at hand directly out of the data stored in the TRANSPATH[®] database.

The play-out process of the Play-Engine can be used for checking the generated LSCs for inconsistencies or logical conflicts, which would give rise to the assumption, that there are inconsistencies in the data source. There also exists a smart play-out process, which offers the possibility to use verification tools to analyze the LSCs. In this context questions like "Is it possible, that ..." can be asked and answered.

When simulating a number of pathways, we already revealed some inconsistencies in the source data. They are summarized in Table 1.

Pathway	Name	Missing Molecule Ref.	Evid. Chain	no Stateof	others
CH000000711	TGFbeta-Pathway	16	-	-	-
CH000000741	Epo-Pathway	7	3	1	-
CH000000746	IL-1-Pathway	3	-	1	3
CH000000749	RhoA-Pathway	5	-	-	-
CH000000755	TLR4-Pathway	4	-	1	3
CH000000759	EDAR-Pathway	-	2	-	-
CH000000768	APP-Pathway	-	-	-	-
CH000000773	TPO-Pathway	-	-	1	-
Total		35	5	4	6

Table 1
Evaluation of Pathways

The columns "Pathway" and "Name" contain the accession number (the key) of the pathway and its name in the TRANSPATH[®] database. The following columns contain different kinds of seemingly wrong (column 3 and 4) or missing (column 5

and 6) references, respectively. In column "Missing Molecule Ref." e.g. the number of wrong references to molecule IDs in the XML files is denoted. Whereas the references mentioned in column 3 are wrong for sure, the inconsistencies reported in column 4 to 6 are contradictions between the documentation of the TRANSPATH[®] database and the data contained in it. Here, consultations with the TRANSPATH[®] team have to reveal, whether the data has to be corrected or the documentation of the database has to be improved.

6 Conclusion

In this paper we presented our approach to treat signal transduction pathways as concurrent reactive systems and to model and simulate them with languages and tools originally developed to assist the construction or engineering of reactive and concurrent systems. As a source for the pathways to be simulated we use the TRANSPATH[®] database and as simulation tools the Play-Engine and CPN Tools for Life Sequence Charts and Colored Petri Nets, respectively.

In the future we plan to support other modeling languages and simulation tools, as e.g. Statecharts and the Rhapsody tool, for which first preparations already have begun. Furthermore, we are going to adapt our system to accept CSML (Cell System Markup Language, [28]) as an input format for signal transduction pathways. CSML is an XML based exchange format for biological pathways. The developers of CSML provide tools to translate pathways described with other XML-based languages, as e.g. CellML [25] and SBML [15] (Systems Biology Markup Language), into CSML. Since most of the major pathway databases make their data available using one of these exchange formats, supporting CSML will enable us to process pathways not only from TRANSPATH[®] but from other major databases as well.

Acknowledgement

We would like to thank C. Choi, E. Shelest, B. Störmann and C. Rio Bartulos for providing us with insights in signal transduction pathways, R. Dalkiran and A. Kolossa for grammatical improvements, our colleagues from the Intergenomics project (especially R. Münch) for fruitful discussions and last but not least all the students that contributed – directly or indirectly – through their diploma theses to this project: N. Fleischer, H. Langhorst, M. Pirsch, T. Springmann, M. Ziegler, O. Witthöft, J.-C. Treusch and F. Rudolph.

References

- [1] B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Lehrbuch der Molekularen Zellbiologie*. Wiley-VCH Verlag GmbH / Co. KGaA, Weinheim, 3 edition, 2005.
- [2] W. Damm and D. Harel. LSCs: Breathing Life into Message Sequence Charts. *Formal Methods in System Design*, 19(1):45–80, 2001.
- [3] S. M. Dauphinee and A. Karsan. Lipopolysaccharide signaling in endothelial cells. *Lab Invest*, 86(1):9–22, 2006.

- [4] S. Eckstein and C. Täubner. An extendable system for conceptual modeling and simulation of signal transduction pathways. In *First International Workshop on Conceptual Modelling for Life Sciences Applications (CMLSA 2007)*, Auckland, New Zealand, November 5-9 2007. Submitted.
- [5] S. Efroni, D. Harel, and I. R. Cohen. Toward rigorous comprehension of biological complexity: modeling, execution, and visualization of thymic T-cell maturation. *Genome Research*, 13(11):2485–2497, 2003.
- [6] S. Efroni, D. Harel, and I.R. Cohen. Emergent Dynamics of Thymocyte Development and Lineage Determination. *PLoS Computational Biology*, 3(1):127–136, 2007.
- [7] J. Ellson and S. North. <http://www.graphviz.org/>.
- [8] N. Fleischer. Modellierung und Simulation der P. aeruginosa Infektion mit Petri-Netzen. Master's thesis, TU Braunschweig, 2005.
- [9] C. Girault and R. Valk. *Petri Nets for Systems Engineering - A Guide to Modeling, Verification, and Applications*. Springer, Berlin, 2003.
- [10] S. Hardy and P. N. Robillard. Modeling and Simulation of Molecular Biology Systems Using Petri Nets: Modeling Goals of Various Approaches. *J Bioinform Comput Biol*, 2(4):595–613, 2004.
- [11] D. Harel. A Grand Challenge for Computing: Towards Full Reactive Modeling of a Multi-Cellular Animal. *Theoretical Computer Science (Bull. EATCS)*, 81:226–235, 2003.
- [12] D. Harel and R. Marelly. *Come, Let's Play - Scenario Based Programming Using LSCs and the Play-Engine*. Springer, Berlin, 2003.
- [13] D. E. Harel and R. E. Marelly. Specifying and executing behavioral requirements: the play-in/play-out approach. *Software and Systems Modeling*, 2(2):82–107, 2003.
- [14] M. Heiner and I. Koch. Petri Net Based Model Validation in Systems Biology. *Proc. Int. Conf. on Application and Theory of Petri Nets*, pages 216–237, June 2004.
- [15] M. Hucka, A. Finney, B. J. Bornstein, S. M. Keating, B. E. Shapiro, J. Matthews, B. L. Kovitz, M. J. Schilstra, A. Funahashi, J. C. Doyle, and H. Kitano. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. *Systems Biology*, 1(1):41–53, June 2004.
- [16] K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use*. Springer, Berlin, 2nd edition, 1997.
- [17] K. Jensen, L. M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. on Software Tools for Technology Transfer (STTT)*, Sp. Sec. CPN 04/05, 2007.
- [18] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(suppl.1):D428–432, 2005.
- [19] N. Kam, D. Harel and I. Cohen, Modeling Biological Reactivity: Statecharts vs. Boolean Logic, Proc. International Conference on Systems Biology, November, 2001, <http://citeseer.ist.psu.edu/kam02modeling.html>
- [20] N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, E. J. A. Hubbert, and M. J. Stern. Formal Modelling of C. elegans Development - A Scenario-Based Approach. In G. Ciobanu and G. Rozenberg, editors, *Modelling in Molecular Biology*, pages 151–173. Springer, Berlin, 2004.
- [21] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(suppl.1):D354–357, 2006.
- [22] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahren, S. Tsoka, N. Darzentas, V. Kunin, and N. Lopez-Bigas. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research*, 33(19):6083–6089, 2005.
- [23] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice - Concepts, Implementation and Application*. Wiley-VCH, Weinheim, 2005.
- [24] M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, O. Kel-Margoulis, and E. Wingender. TRANSPATH(R): an information resource for storing and visualizing signaling pathways and their pathological aberrations. *Nucleic Acids Research*, 34(suppl.1):D546–551, 2006.

- [25] C. M. Lloyd, M. D. B. Halstead, and Poul F. Nielsen. CellML: its future, present and past. *Progress in Biophysics and Molecular Biology*, 85(2-3):433–450, Jun-Jul 2004.
- [26] S. Motta and V. Brusici. Mathematical Modelling of the Immune System. *Modelling in Molecular Biology*, pages 193–218. Springer-Verlag Berlin, 2004.
- [27] M. Nagasaki, A. Doi, H. Matsuno, and S. and Miyano. Petri Net Based Description and Modeling of Biological Pathways. *Algebraic Biology - Computer Algebra in Biology*, pages 19–31, 2005.
- [28] M. Nagasaki, E. Jeong, A. Doi, A. Saito, and S. Miyano. *CSML 3.0 - Cell System Markup Language - Basic Concept and Specification*, 2006.
- [29] OMG. *UML 2.1.1 Superstructure Specification v.2.1.1.*, February 2007.
- [30] M. Peleg, D. Rubin, and R. B. Altman. Using Petri Net tools to study properties and dynamics of biological systems. *J Am Med Inform Assoc*, 12(2):181–99, 2005.
- [31] Milner R. *Communicating and Mobile Systems. Pi-calculus*. Cambridge University Press, 1999.
- [32] A. Regev and E. Shapiro. The pi-calculus as an Abstraction for Biomolecular Systems. In Gabriel Ciobanu and Grzegorz Rozenberg, editors, *Modelling in Molecular Biology*, pages 219–266. Springer, Berlin, 2004.
- [33] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the pi- calculus process algebra. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 6, pages 459–470, Singapore, 2001. World Scientific Press.
- [34] C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic: Modeling of protein functional domains in signal transduction. In R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, editors, *Proc. of the Pacific Symposium on Biocomputing*, pages 568–580, Singapore, 2004. World Scientific.
- [35] C. Täubner, B. Mathiak, A. Kupfer, N. Fleischer, and S. Eckstein. Modelling and Simulation of the TLR4 Pathway with Coloured Petri Nets. In A. P. Dhawan, F. A. Laine, M. Akay, and K. H. Chon, editors, *28th Annual Int. Conf. of IEEE Engineering in Medicine and Biology Society*, page 221, New York, 2006.
- [36] C. Täubner and T. Merker. Modelling of the Signaling Pathway of *Arabidopsis thaliana*. In N. Guimaras and P. Isaias, editors, *In IADIS International Conference Applied Computing*, volume 2 of *IADIS Press*, pages 295–300, Algarve, Portugal, February 22 - 25 2005. IADIS Press.
- [37] Telelogic AB. Telelogic Rhapsody. <http://www.telelogic.com/products/rhapsody>, 2007.
- [38] A.M. Uhrmacher, D. Degenring, and B. Zeigler. Discrete Event Multi-level Models for Systems Biology. In *Transactions on Computational Systems Biology*, volume 3380 of *LNBI*, pages 66–89. Springer, 2005.