Constructing and Visualizing Chemical Reaction Networks from Pi-Calculus Models

Mathias John[1], Hans-Jörg Schulz[2], Heidrun Schumann[3], Adelinde M. Uhrmacher[3], and Andrea Unger[4]

[1]BioComputing, Lifl (Cnrs Umr8022) & Iri (Cnrs Usr3078) & University of Lille 1, France
[2]Graz University of Technology, Austria
[3]University of Rostock, Germany
[4]Helmholtz Centre Potsdam, Germany

Abstract. The π-calculus, in particular its stochastic version the stochastic π-calculus, is a common modeling formalism to concisely describe the chemical reactions occurring in biochemical systems. However, it remains largely unexplored how to transform a biochemical model expressed in the stochastic π-calculus back into a set of meaningful reactions. To this end, we present a two step approach of first translating model states to reaction sets and then visualizing sequences of reaction sets, which are obtained from state trajectories, in terms of reaction networks. Our translation from model states to reaction sets is formally defined and shown to be correct, in the sense that it reflects the states and transitions as they are derived from the continuous time Markov chain-semantics of the stochastic π-calculus. Our visualization concept combines high level measures of network complexity with interactive, table-based network visualizations. It directly reflects the structures introduced in the first step and allows modelers to explore the resulting simulation traces by providing both: an overview of a network's evolution and a detail inspection on demand.

Keywords: pi-calculus, stochastic modeling, reaction networks, graph visualization

Correspondence and offprint requests to:
Mathias John,
Interdisciplinary Research Institute,
Parc de la Haute Borne, 50 avenue de Halley, BP 70478
59658 Villeneuve d'Ascq Cédex, FRANCE.
e-mail: mathias.john@lifl.fr

1. Introduction

Over the last years, computational modeling became a valuable method to study the dynamics of complex cell-biological systems. The idea is to provide an abstract representation of the system under study, expressed in a modeling formalism, and to analyze such with the help of computers. By this, computational modeling helps to structure knowledge, to evaluate existing theories, and to suggest new set-ups for wet-lab experiments.

The π-calculus [Mil99] is considered to be a well-suited formalism for the modeling of cell-biological processes [Reg03, PRSS01]. Its stochastic version, called the stochastic π-calculus [Pri95], comes with a stochastic semantics in terms of continuous time Markov chains (CTMC) that allows to account for the dynamics of cell-biological processes. The stochastic π-calculus has been applied in several modeling studies, see for example, [Kut06, TK08, CCG$^+$09, MJM$^+$09]. However, due to the origin of the π-calculus in the field of concurrency theory, the basic paradigm of the stochastic π-calculus is the one of process communication, whereas cell-biological systems are rather described in terms of chemical reactions and solutions. Therefore, efforts have been made to find ways of transforming reaction networks into process communication, see e.g., [PRSS01, Kut06].

The way back from stochastic π-calculus models to chemical reactions, however, is largely unexplored. But it is this half of the modeling cycle that actually allows modelers to review whether the transformation from reactions to process communication yields the desired results and whether the model is a meaningful interpretation in the biochemical context. To this end, we propose a two-step approach of first extracting reaction sets from stochastic π-calculus models and then visualizing the obtained results in terms of reaction networks. Thereby, the size of resulting reaction sets forms the major challenge. Each of a stochastic π-calculus model's CTMC states defines a set of chemical reactions that may possibly occur. Existing approaches try to create reaction sets accounting for all states of a model at once [Car08a, Car08b, MG09]. In practical cases, this, however, mostly leads to reaction sets of large, up to infinite size (a detailed example and also an introduction to the stochastic π-calculus can be found in Section 2). Therefore, we propose an approach that considers each model state individually. Timed traces of model states are obtained based on stochastic simulation in terms of [Gil77]. This first step of our approach is presented in Section 3.

To support model analysis across multiple consecutive states of a simulation, we propose the second step of visualization. Reaction sets of individual states are represented in a time-dependent manner, such that the dynamics of the transformed model are easily accessible. By making use of a representation as reaction networks, our visualization is capable of handling individual reaction sets of large size. Without a systematic visual approach for their representation and exploration, experts may theoretically be able to understand the results of the transformation, but practically drown in them. Previous attempts of visualizing stochastic π-calculus models are either mere static graphical notations [PCC06] or do not scale well to large models because of the chosen visualization approach of a 3-dimensional spatial layout [Phi]. In both cases, the visualizations do not reflect the nature of a reaction network and stick closely to the notion of communicating automata. Hence, in Section 4, we propose a visualization setup that is more scalable by employing more space-efficient visualization methods, such as an overview, which summarizes the distinct states of the solution using measures of network complexity, and table-based representations for a detailed view of individual states. By this, the exploration of these large data sets is visually supported and the modeler can finally draw the needed conclusions from the transformation results – be it during the actual modeling phase in which the model must be debugged and validated, or while working with the model to gain an understanding of the inner workings of the modeled cell-biological system.

It is notable that our approach is not only applicable to the stochastic π-calculus, but also to most of its extensions. We provide a brief discussion on this topic in Section 5, before concluding this paper by giving an outlook on directions for future work in Section 6.

2. The Stochastic π-Calculus

We base our investigation on the biochemical form of the stochastic π-calculus. This is a reformulation of the stochastic π-calculus that is closer to the field of modeling biochemical systems, in the sense that it provides notions of molecules and chemical solutions rather than just accounting for concurrently acting processes (achieved by syntactic restrictions). The two existing up-to-date stochastic π-calculus tools for modeling
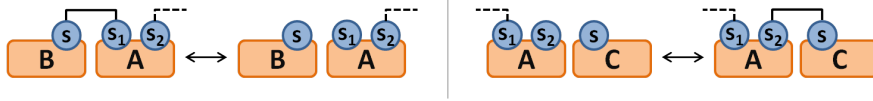
Fig. 1. Binding reaction between three different sorts of molecules. Molecule A may perform an un-/binding reaction with B by creating and removing links (black line) between binding sites s_1 and s. Similarly, molecules A and C may bind using respective binding sites s_2 and s. Dashed lines represent links that may be present but do not have to be. That is, reactions on site s_1 may happen independently of the state of site s_2 and vv. This means the following chemical reactions can take place: A,B $\longleftrightarrow$ AB; AC,B $\longleftrightarrow$ ACB; A,C $\longleftrightarrow$ AC; AB,C $\longleftrightarrow$ ABC.

biochemical systems base on the biochemical form [PC07, LJU10]. In the following, when mentioning the stochastic π-calculus, we actually refer to its biochemical form.

The idea of modeling chemical reactions in the stochastic π-calculus is, instead of using rather monolithic species for complexes like ABC, to describe a complex of three molecules A, B, and C, to actually consider molecules with their individual binding sites and complexes by establishing links between them, see Figure 1. As shown in [HFB$^+$03, FHR$^+$03], this view may significantly increase the conciseness of model descriptions. For this, the stochastic π-calculus requires modelers to shift their focus from reactions to molecules and their interactions. Consider, e.g., the complex formations depicted in Figure 1. These can be modeled in the stochastic π-calculus in the following way:

```
A(s1,s2,s1N,s2N)  =:  s1!(s1N).A(s1N,s2,s1,s2N)
                    + s2!(s2N).A(s1,s2N,s1N,s2)
B(s)  =:  s?(sN).B(sN)
C(s)  =:  s?(sN).C(sN)
```

Molecule definitions are introduced for A, B, C with parameters s1,s2,s,s1N,s2N. The only values in the stochastic π-calculus are names, also called channels. Interaction capabilities between molecules, i.e., the reactions that may possibly occur, depend on name scoping. For example, a chemical solution with one A and two B, all not bound, can be denoted as the parallel composition of molecules $S = $ A(bAB,bAC,uAB,uAC) $\mid$ B(bAB) $\mid$ B(bAB). The idea is that molecule A(bAB,bAC,uAB,uAC) shares the name bAB as the value of its first parameter with all B(bAB), allowing for free interaction between all A and B. By contrast, a solution $S' = $ A(uAB,bAC,bAB,uAC) $\mid$ B(uAB) $\mid$ B(bAB) is to denote a complex of A with the first B, since only these two may interact on channel uAB. In the very same way, interaction capabilities between A and C molecules can be controlled, using names bAC and uAC. Whereas interactions on channels bAB, bAC shall model binding reactions, those on channels uAB,uAC are considered to denote unbinding reactions. This is specified in the definitions of the respective molecules. The only notion of interaction in the stochastic π-calculus is the one of communication, with one sending and one receiving partner. Molecules A(bAB,bAC,uAB,uAC) and B(bAB) can send (!) and receive (?) on channel bAB, respectively. As denoted in the parentheses after the !-operator, A(bAB,bAC,uAB,uAC) sends channel bAB to B(bAB) that addresses it with formal parameter bNext. After their interaction, A(bAB,bAC,uAB,uAC) is to be replaced by A(uAB,bAC,bAB,uAC) and B(bAB) by B(uAB). Thus, interactions on channel B(bAB) denote a binding between A and B, as they lead from solution S above to solution S'. Subsequently, A(uAB,bAC,bAB,uAC) and B(uAB) may communicate on channel uAB in the very same way to be replaced again by A(bAB,bAC,uAB,uAC) and B(bAB), thus denoting the unbinding by closing the cycle from S' to S. Alternatively, as denoted by the choice operator (+), A(bAB,bAC,uAB,uAC) can perform the very same interactions, just on channels bAC and uAC, with C(bAC). Notice, that this happens entirely independently of the values of parameters s1 and s1N. Thus, a solution $S = $ A(bAB,bAC,uAB,uAC) $\mid$ B(bAB) $\mid$ C(bAC) represents by the interaction capabilities of its contained molecules the following eight chemical reactions that in fact reflect those depicted in Figure 1:

$$
\begin{array}{ccc}
\text{A(bAB,bAC,uAB,uAC), B(bAB)} & \longleftrightarrow & \text{A(uAB,bAC,bAB,uAC), B(uAB)} \\
\text{A(bAB,uAC,uAB,bAC), B(bAB)} & \longleftrightarrow & \text{A(uAB,uAC,bAB,bAC), B(uAB)} \\
\text{A(bAB,bAC,uAB,uAC), C(bAC)} & \longleftrightarrow & \text{A(bAB,uAC,uAB,bAC), C(uAC)} \\
\text{A(uAB,bAC,bAB,uAC), C(bAC)} & \longleftrightarrow & \text{A(uAB,uAC,bAB,bAC), C(uAC)}
\end{array}
$$

Notice, that we actually need to distinguish reactions based on the parameter values of their reactants, since, e.g., A(bAB,bAC,uAB,uAC) has different interaction capabilities then A(uAB,bAC,bAB,uAC). In fact, one can consider A(bAB,bAC,uAB,uAC) and A(uAB,bAC,bAB,uAC) to be of a different chemical species, just like A and AB. Notice further that our model so far only works in the presence of a single A molecule. Consider, e.g., a solution $S = $ A(uAB,bAC,bAB,uAC) $\mid$ A(uAB,bAC,bAB,uAC) $\mid$ B(uAB) $\mid$ B(uAB). Here the scope of uAB is not

Molecule	M	$::=$	$A(\tilde{x})$	instantiation
Solution	S	$::=$	$\mathbf{0}$	empty solution
		$\mid$	M	single molecule
		$\mid$	$S \mid S$	parallel composition
		$\mid$	$(\nu x)S$	solution with link
Prefixes	π	$::=$	$x?(\tilde{x})$	receiver
		$\mid$	$x!(\tilde{x})$	sender
Choice	C	$::=$	$\pi.S$	interaction
		$\mid$	$C + C$	or
Definition	D	$::=$	$A(\tilde{x}) =: C$	parametric molecule definition

Fig. 2. Syntax of the stochastic π-calculus (biochemical form), where $x, \tilde{x} \in \textit{Vars}$.

restricted enough, since A(uAB,bAC,bAB,uAC) may communicate with two different B(uAB). That is, solution S can be considered to represent complex $AABB$, since all molecules are connected, which was not intended. This problem is solved by assigning distinct names to represent the links of each molecule A, e.g., $S' =$ A(uAB1,bAC,bAB,uAC1) | A(uAB2,bAC,bAB,uAC2) | B(uAB1) | B(uAB2). As a direct consequence, the number of reactions represented by solution S' doubles to 16, and in general yields $n * 8$, where n is the number of A molecules in the solution. The stochastic π-calculus also provides a ν-operator by which channel names can be created dynamically when an interaction occurs. Thus, models in the stochastic π-calculus often yield up to infinitely large reaction sets when considering all possible solutions a model may produce starting from a given initial solution, i.e., a model's entire state space. Thus, enumerating a model's entire set of possible reactions for all states at once is not feasible, such that we decided to do it for each state individually.

In the following, we present the syntax and the operational semantics of the stochastic π-calculus. We omit a description of a stochastic simulator for the stochastic π-calculus here, as our approach is mostly independent of how state trajectories are generated.

2.1. Syntax

The stochastic π-calculus bases on a set of process names $A \in \textit{Proc}$ and an infinite set of channel names $x, y, z \in \textit{Vars}$. We make largely use of tuple notation, i.e., $\tilde{x} = (x_1, \ldots, x_n)$, where $|\tilde{x}| = n$ defines the length of $\tilde{x}$ to be n. As usual in the π-calculus, channel names denote both values and formal parameters. To channel values, rate constants in $\mathbb{R}_+$ are assigned. The rate constant of a channel can be obtained by using function $\kappa : \textit{Vars} \to \mathbb{R}_+$, mapping channels to rate constants. For the sake of simplicity we do not account for prioritized interactions, i.e., for channels with infinite rate constants on which interactions happen immediately. However, we believe that our results in Section 3 can be extended in a straightforward way.

The syntax of the stochastic π-calculus (biochemical form) is presented in Figure 2. It considers five syntactic categories: molecules, solutions, prefixes, choices, and definitions. Solutions are parallel composition of molecules M. Given a solution where more than two molecules can interact, the order of the occurrence of interactions is considered to be undetermined. Instead of $M_1 \mid \cdots \mid M_n$, we also write $\prod_{i=1}^{n} M_i$, where $\prod_{i=1}^{0} M_i = \mathbf{0}$ (empty solution). The ν-operator $(\nu x)S$ allows us to create a new channel name x with scope to S. We write $(\nu \tilde{x})$ to denote sequences of ν-operators including the empty sequence. A molecule is considered to be an instance $A(\tilde{y})$ of a parametric molecule definition $A(\tilde{x})$ that replaces the formal parameters $\tilde{x}$ by actual parameters $\tilde{y}$. We assume that in sets of process definitions no name A occurs twice and that the numbers of actual and formal parameters are always equal. Definitions $A(\tilde{x}) \triangleq C$ specify the interaction capabilities of molecules in terms of a choice C. We assume that all channel names in $\tilde{x}$ are pairwise distinct, since they are distinct formal parameters. Choices denote exclusive options $\pi.S$ with prefixes π denoting a send or a receive action. Once the action π of option $\pi.S$ is performed, all other options are discarded and the molecule is replaced by the solution S. We denote choices $\pi_1.S_1 + \cdots + \pi_n.S_n$ by $\sum_{i=1}^{n} M_i$ and leave $\sum_{i=1}^{0} M_i$ undefined, since a molecule needs to be able to perform at least one action. A receiver $x?(\tilde{x})$ is supposed to receive a tuple of channels $\tilde{x}$ on channel x, and a sender $x!(\tilde{x})$ to send a tuple of channels $\tilde{x}$ on

$$
\begin{aligned}
fv(C_1 + C_2) &= fv(C_1) + fv(C_2) & fv(\mathbf{0}) &= \emptyset \\
fv(x?(\tilde{x}).S) &= \{x\} \cup (fv(S) \setminus \{\tilde{y}\}) & fv(S_1 \mid S_2) &= fv(S_1) \cup fv(S_2) \\
fv(x!(\tilde{y}).S) &= \{x\} \cup fv(\{\tilde{y}\}) \cup fv(S) & fv(A(\tilde{x})) &= \{\tilde{x}\} \cup fv(A(\tilde{x}) \triangleq C) \\
fv((\nu x : k)S) &= fv(S) \setminus \{x\} & fv(A(\tilde{x}) \triangleq C) &= fv(C) \setminus \{\tilde{x}\}
\end{aligned}
$$

Fig. 3. Free channel names of the π-calculus with priority.

$$
\begin{aligned}
S_1 \mid S_2 &\equiv S_2 \mid S_1 & (S_1 \mid S_2) \mid S_3 &\equiv S_1 \mid (S_2 \mid S_3) \\
S \mid \mathbf{0} &\equiv S & (\nu x)(S_1 \mid S_2) &\equiv (\nu x)S_1 \mid S_2 \text{ if } x \notin fv(S_2) \\
S_1 \equiv_\alpha S_2 &\Rightarrow S_1 \equiv S_2 & (\nu x_1)(\nu x_2)S &\equiv (\nu x_2)(\nu x_1)S
\end{aligned}
$$

Fig. 4. Axioms of the structural congruence of the π-calculus with priority.

channel x. We again assume that all channel names in $\tilde{y}$ are pairwise distinct, since they are distinct formal parameters.

We introduce a structural congruence of solutions which is the least relation satisfying the rules in Figure 4. It identifies all those solutions that satisfy associativity and commutativity. Furthermore, it allows us omitting empty solutions in parallel compositions. Moreover, it identifies solutions by consistently renaming bound names, i.e., using α-conversion. For this, it strongly relies on the function fv to determine free names in solutions, see Figure 3. We consider all those names to be free which are not in the scope of ν-binders or the formal parameters of molecule definitions and receive actions. Renaming operations also extend to rate constant function κ. The structural congruence lets us change the order of ν-operators and extend their scope to solutions whose free names do not equal the name bound by the ν-operator.

We say a solution S is in *prenex normal form* if and only if $S = (\nu\tilde{x}) \prod_{i=1}^{n} M_i$. We introduce a *multiset representation* of a prenex normal form by $(\nu\tilde{x}) \prod_{i=1}^{n} M_i^{m_i}$, where molecules M_i occur m_i times. Notice, that every solution is structurally congruent to a prenex normal form.

2.2. Operational semantics

The semantics of the stochastic π-calculus maps a solution S to a state of a continuous time Markov chain (CTMC) including its transitions. The main idea is to regard all distinct solutions which are reachable from S as successor states. Thereby, two states are considered distinct if their solutions are not structurally congruent. The transitions of S are derived from its possible communications, transition weights are obtained by summing up the rate constants of all possible communications leading to the same successor state. As shown in, e.g., [KLN07], stochastic simulators can be derived directly from such stochastic π-calculus semantics.

In order to account for all possible communications in a solution, the semantics strongly relies on determining the redexes of prenex normal forms. Thereby, a redex of a solution in prenex normal form $S = (\nu\tilde{x}) \prod_{i=1}^{n} M_i$ is defined as a tuple (i_1, j_1, i_2, j_2) that captures with $i_1, i_2 \in \{1, \ldots, n\}$ the indices of two molecules forming an interaction pair, with M_{i_1} being the sender and M_{i_2} the receiver. By j_1, j_2 then, the indices of the actions in the choices defining M_{i_1}, M_{i_2} that let M_{i_1}, M_{i_2} communicate are reflected. Consider, e.g., solution $S = \mathsf{A(b,b,u1,u2)} \mid \mathsf{B(b)} \mid \mathsf{B(b)}$, with definitions as in our introducing example (also recalled in Figure 6). We obtain four redexes $(1, 1, 2, 1)$, $(1, 2, 2, 1)$, $(1, 1, 3, 1)$, and $(1, 2, 3, 1)$, accounting for all possible communications in S.

The rules of the operational semantics of the stochastic π-calculus are presented in Figure 5. We denote the substitution of tuple $\tilde{x}$ by tuple $\tilde{y}$ as $[\tilde{y}/\tilde{x}]$. Substitution only applies to tuples of the same length. By rule (APP) molecules are replaced by their definitions with the formal parameters being substituted by the actual ones. Rule (RED) formulates the reduction of a pair of communicating molecules M_1, M_2. Therefore, it captures the indices i_1, i_2 of the actions that let M_1, M_2 communicate and the channel x on which they communicate. In order to ensure a proper substitution, it further checks whether M_2 is actually ready to receive the amount of values that M_1 sends. Rule (SEL) selects two molecules to communicate. Only those pairs qualify to which rule (RED) applies and that refer to distinct molecules, since one molecule may not

Application

$$\text{(APP)} \quad \frac{A(\tilde{x}) =: C}{A(\tilde{y}) \xrightarrow{app} C[\tilde{y}/\tilde{x}]}$$

Communication

$$\text{(RED)} \quad \frac{M_1 \xrightarrow{app} \sum_{i=1}^n \pi_i.S_i \quad M_2 \xrightarrow{app} \sum_{i=1}^{n'} \pi_i'.S_i' \quad \pi_{i_1} = x!(\tilde{x}) \quad \pi_{i_2}' = x?(\tilde{y}) \quad |\tilde{x}| = |\tilde{y}|}{M_1, M_2 \xrightarrow[(i_1,i_2)]{x} S_{i_1} \mid S_{i_2}}$$

$$\text{(SEL)} \quad \frac{M_{i_1}, M_{i_2} \xrightarrow[(j_1,j_2)]{x} S \quad i_1 \neq i_2}{(\nu\tilde{x}:\tilde{k}) \prod_{i=1}^n M_i \xrightarrow[(i_1,j_1,i_2,j_2)]{x} (\nu\tilde{x}:\tilde{k})(\prod_{i=1, i \neq i_1, i_2}^n M_i) \mid S}$$

Markov chain

$$\text{(SUM)} \quad \frac{S \equiv S_1 \quad \sum_{\{(k,\ell)|S_1 \xrightarrow{x}_\ell S_2, S_2 \equiv S', k = \kappa(x)\}} k = r \neq 0}{S \xrightarrow{r} S'}$$

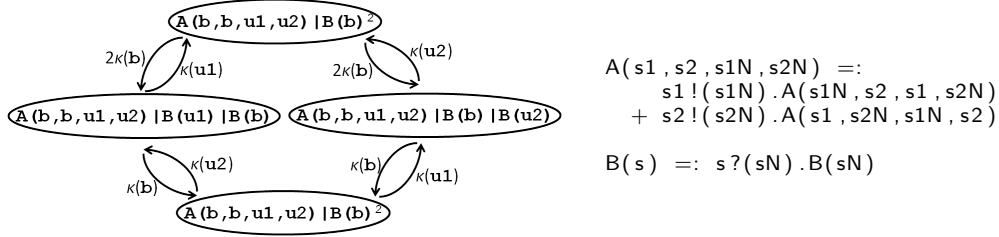**Fig. 5.** Rules of the stochastic semantics of the stochastic π-calculus (biochemical form).


```
A(s1 ,s2 ,s1N ,s2N )  =:
    s1 !( s1N ).A( s1N ,s2 ,s1 ,s2N )
  + s2 !( s2N ).A( s1 ,s2N ,s1N ,s2 )

B( s )  =:  s ?( sN ).B( sN )
```

Fig. 6. The CTMC of solution $S = \mathsf{A(b,b,u1,u2)} \mid \mathsf{B(b)}^2$, obtained following the rules of the operational semantics of the stochastic π-calculus as presented in Figure 5.

communicate with itself. Communication transitions are annotated with communication channel x and redex ℓ. This information is used to obtain state transitions in rule (SUM), where state transitions are defined by combining all communications that lead to the same successor and summing up corresponding rate constants.

For illustration, Figure 6 depicts the CTMC of solution $S = \mathsf{A(b,b,u1,u2)} \mid \mathsf{B(b)}^2$, with molecule definitions as in our introducing example (also recalled in the figure).

3. Extracting reaction sets from processes

In this section, we show how to extract reaction sets from solutions. We fix a notion of reactions that directly follows the ideas given in Section 2. That is, an interaction $M_1 \mid M_2 \xrightarrow[(j_1,j_2)]{x} S$, with $S \equiv (\nu\tilde{x}) \prod_{i=1}^n M_i$ yields reaction $M_1, M_2 \xrightarrow{x} (\nu\tilde{x})(M_1', \ldots, M_n')$. It may seem surprising that reactions are with a channel upon the arrow and not with a rate constant as usual. However, the idea is that if modelers denote interactions to happen on different channels, they also intend them to denote different reactions, even if they show the same rate constants.

In order to identify communications in a given solution, we introduce communication labels that comprise a communication's channel, rate constant, and redex. More precisely, a communication label of a solution S is a tuple $\ell = (x, i_1, j_1, i_2, j_2)$, where $S' \xrightarrow[(i_1,j_1,i_2,j_2)]{x} S''$, with $S' \equiv S$. The set $\Lambda(S)$ contains all communication

labels of a solution S in prenex normal form and is defined as:

$$\Lambda(S) = \{(x, i_1, j_1, i_2, j_2) \mid S \xrightarrow[(i_1,j_1,i_2,j_2)]{x} S'\}, \text{ with } S = (\nu\tilde{x})\prod_{i=1}^{n} M_i$$

Since two communications labels may refer to the same reaction, we need to ensure that we do not extract reaction duplicates. Therefore, we introduce an equivalence of communication labels $\sim$ that identifies two communication labels that refer to the same reaction. It is defined as the least relation fulfilling the following rule:

$$(x, i_1, j_1, i_2, j_2) \sim (x', i_1', j_1', i_2', j_2')$$
$$\text{if and only if}$$
$$x = x', M_{i_1} \mid M_{i_2} \equiv M_{i_1'} \mid M_{i_2'}, M_{i_1} \mid M_{i_2} \xrightarrow[(j_1,j_2)]{x} S, M_{i_1'} \mid M_{i_2'} \xrightarrow[(j_1',j_2')]{x'} S' \equiv S$$

We define the quotient set of communication labels that contains the equivalent classes of the communication labels of solution S in the following way:

$$\Lambda^\sim(S) = \{[\ell] \mid \ell \in \Lambda(S)\}, \text{ where } [\ell] = \{\ell' \in \Lambda(S) \mid \ell' \sim \ell\}$$

By definition, each equivalent class in $\Lambda(S)$ represents a single reaction. The set of reactions of S can thus be obtained by picking a representative of each equivalent class in $\Lambda^\sim(S)$. Consider, e.g., solution $S = $ A(b,b,u1,u2) | B(b) | B(b), with definitions as in our introducing example (also recalled in Figure 6). We obtain $\Lambda^\sim(S) = \{\{(b,1,1,2,1), (b,1,1,3,1)\}, \{(b,1,2,2,1), (b,1,2,3,1)\}\}$, containing the equivalent classes representing the two reactions:

$$\text{A(b,b,u1,u2), B(b)} \xrightarrow{b} \text{A(u1,b,b,u2), B(u1)}$$
$$\text{A(b,b,u1,u2), B(b)} \xrightarrow{b} \text{A(b,u2,u1,b), B(u2)}$$

Constructing $\Lambda^\sim(S)$ from a given solution S requires a pairwise comparison of communication labels, yielding quadratic complexity in the number of communication labels in S. An increased performance can be achieved by making use of the fact that most π-calculus simulators, e.g., [PC07, LJU10], work with species set representations of solutions $S = (\nu\tilde{x}) \prod_{i=1}^{n} M_i^{m_i}$. The idea is that a molecule pair M_1, M_2 shows the same interaction capabilities as all other molecule pairs M_1', M_2', where $M_1 = M_1'$ and $M_2 = M_2'$, i.e., where M_1 and M_2 are of the species as M_1' and M_2', respectively. Thus, instead of considering S, one can extract the set of reactions directly from solution $S' = (\nu\tilde{x}) \prod_{i=1}^{n} M_i$. In most practical cases the number of species will be considerably smaller then the number of molecules. We introduce the quotient set $\Lambda_\bullet^\sim$ that allows us to extract reactions directly from species set representations as follows:

$$\Lambda_\bullet^\sim(S) = \Lambda^\sim((\nu\tilde{x})\prod_{i=1}^{n} M_i), \text{ with } S \equiv (\nu\tilde{x})\prod_{i=1}^{n} M_i^{m_i}$$

For illustration consider the solution $S = $ A(b,b,u1,u2) | B(b)^2 as in our preceding example. We obtain $\Lambda_\bullet^\sim(S) = \{\{(b,1,1,2,1)\}, \{(b,1,2,2,1)\}\}$, i.e., two equivalence classes representing the same reactions as before. The following lemma states that in general this way of extracting reaction sets is correct.

Lemma 1. For all solutions S in prenex normal form it holds that $\sharp\Lambda^\sim(S) \subseteq \sharp\Lambda_\bullet^\sim(S)$ and for all $[\ell] \in \Lambda^\sim(S)$, there exists exactly one $[\ell'] \in \Lambda_\bullet^\sim(S)$, such that $\ell \sim \ell'$.

Proof. Let $S \equiv (\nu\tilde{x}) \prod_{i=1}^{n} M_i^{m_i}$, $S' = (\nu\tilde{x}) \prod_{i=1}^{n} M_i$ and $S = (\nu\tilde{x}) \prod_{i=1}^{n'} M_i'$. By definition, $\Lambda_\bullet^\sim(S) = \Lambda^\sim(S')$. Thus, $\sharp\Lambda^\sim(S) \subseteq \sharp\Lambda_\bullet^\sim(S))$ clearly holds. Furthermore, assume that there exists no $[\ell'] \in \Lambda_\bullet^\sim(S)$, such that $[\ell] \sim [\ell']$. Let $\ell = (x, i_1, j_1, i_2, j_2)$ and $M_{i_1}' \mid M_{i_2}' \xrightarrow[x]{(i_1,j_1,i_2,j_2)} S_1$. By definition, there exists i_1', i_2', such that $M_{i_1}' = M_{i_1'}$ and $M_{i_2}' = M_{i_2'}$. This implies that $M_{i_1'} \mid M_{i_2'} \xrightarrow[x]{(i_1',j_1,i_2',j_2)} S_1$, such that $\ell \sim (x, i_1', j_1, i_2', j_2)$. Since $\sim$ is an equivalence relation, the definition of $\Lambda^\sim$ provides that there exists $[\ell'] \in \Lambda^\sim(S')$ with $[\ell'] \sim (x, i_1', j_1, i_2', j_2)$. By transitivity it also holds that $\ell' \sim \ell$, which contradicts our initial assumption. Moreover, transitivity provides that there cannot exist $[\ell''] \in \Lambda(S')$, with $\ell \sim \ell''$ and $\ell' \neq \ell''$. $\square$

Given a solution S, the rates of reactions in $\Lambda^\sim(S)$ can be easily determined by summing up the rate constants of the communication labels belonging to the same equivalence class. However, when using $\Lambda^\sim_\bullet(S)$, where not all communication labels are individually considered, this is not feasible (e.g., $\Lambda^\sim(S) = \{\{(b,1,1,2,1),(b,1,1,3,1)\},\{(b,1,2,2,1),(b,1,2,3,1)\}\}$ vs. $\Lambda^\sim_\bullet(S) = \{\{(b,1,1,2,1)\},\{(b,1,2,2,1)\}\}$, with S as in the preceding examples). In this case reaction rates need to be determined more directly in the following way: consider communication label $\ell = (x,i_1,j_1,i_2,j_2)$ and solution $S \equiv (\nu\tilde{x})\prod_{i=1}^n M_i^{m_i}$. By the definition of $\sim$, all the communication labels belonging to $[\ell]$ show the same rate constant. Thus, it suffices to compute the number of communication labels $\ell' \in \Lambda(S)$, with $\ell' \sim \ell$. We start by counting the number of communication labels $\ell' = (x,i_1',j_1,i_2',j_2)$ in $\Lambda(S)$, with $M_{i_1} = M_{i_1'}$ and $M_{i_2} = M_{i_2'}$. Following the rules of combinatorics, this yields $m_{i_1} * m_{i_2}$, if $M_{i_1} \neq M_{i_2}$ and $m_{i_1} * (m_{i_2} - 1) = m_{i_1} * (m_{i_1} - 1)$, otherwise. However, molecules M_{i_1}, M_{i_2} may also be defined to perform the same reaction several times. For example, a pair of molecules $\mathsf{D}(), \mathsf{E}()$, with definitions $\mathsf{D}() =: \mathsf{x!().E()} + \mathsf{x?().D()}$ and $\mathsf{E}() =: \mathsf{x!().E()} + \mathsf{x?().D()}$. By definition of $\Lambda^\sim_\bullet$, all the communication labels involving M_{i_1}, M_{i_2}, also those where M_{i_1}, M_{i_2} swap the sending and receiving role, are considered by $[\ell] \in \Lambda^\sim_\bullet(S)$. Again, it is clear that all copies of M_{i_1}, M_{i_2} behave the same way. Thus, we define the reaction rate of a communication label of solution S in the following way:

$$\rho(\ell, S) = \kappa(x) * \sharp[\ell] * \tau, \text{ where}$$
$$\tau = \begin{cases} m_{i_1} * (m_{i_2} - 1), \text{ if } M_{i_1} = M_{i_2} \\ m_{i_2} * m_{i_1}, \text{ else} \end{cases}$$
$$\text{and } \ell = (x,i_1,j_1,i_2,j_2), S \equiv (\nu\tilde{x})\prod_{i=1}^n M_i^{m_i}, [\ell] \in \Lambda^\sim_\bullet(S)$$

Consider, e.g., solution $S = \mathsf{A(b,b,u1,u2)} \mid \mathsf{B(b)}^2$ as in our preceding examples, where $\Lambda^\sim(S) = \{\{(b,1,1,2,1),(b,1,1,3,1)\},\{(b,1,2,2,1),(b,1,2,3,1)\}\}$ and $\Lambda^\sim_\bullet(S) = \{\{(b,1,1,2,1)\},\{(b,1,2,2,1)\}\}$. We obtain that $\kappa((b,1,1,2,1),S) = \kappa((b,1,2,2,1),S) = 2\kappa(b)$. It remains to ensure that really all communication labels $\ell' \in \Lambda(S)$, with $\ell' \sim \ell$, are take into account. The following lemma states that our way of computing rates based $\Lambda^\sim_\bullet$ is correct in general.

Lemma 2. Let S be a solution in prenex normal form. For all $[\ell] \in \Lambda^\sim_\bullet(S)$, it holds that $\rho(\ell,S) = \sum_{\{(x,l)\mid(x,l)\in\Lambda(S),(x,l)\sim\ell\}} \kappa(x)$.

Proof. Let $\ell = (x,i_1,j_1,i_2,j_2)$ and $\Delta = \{\ell' \mid \ell' \in \Lambda(S), \ell' \sim \ell\}$. By the definition of $\sim$ and distributivity, it suffices to show that $\tau * \sharp[\ell] = \sharp\Delta$, with τ as defined above. By definition, it holds that $\tau * \sharp[\ell] \leq \sharp\Delta$. For the case $\tau * \sharp[\ell] \geq \sharp\Delta$, suppose that there exists $\ell' = (x',i_1',j_1',i_2',j_2')$, with $\ell' \sim \ell$, such that ℓ' is not considered by $\tau * \sharp[\ell]$. Since $\ell' \sim \ell$, it holds that $x = x'$ and either (1) $M_{i_1} = M_{i_1'}$ and $M_{i_2} = M_{i_2'}$ or (2) $M_{i_1} = M_{i_2'}$ and $M_{i_2} = M_{i_1'}$. In case (1), there exists $\ell'' = (x,i_1,j_1',i_2,j_2')$, which by definition is considered in $\sharp[\ell]$. Thus, by the definition of τ, also (x',i_1',j_1',i_2',j_2') is considered in $\sharp[\ell] * \tau$. Similarly, in case (2), we obtain that there exists $\ell'' = (x,i_2,j_1',i_1,j_2')$, which is considered in $\sharp[\ell]$. Again, the definition of τ provides that also (x',i_2',j_1',i_1',j_2') is considered in $\sharp[\ell] * \tau$. Thus, in both cases, we obtain a contradiction with our initial assumption. $\square$

Notice that, as also pointed out in [CCG$^+$09, JLNV11], the special case of modeling unary reactions in the stochastic π-calculus violates the kinetic law of Mass action that is usually subsumed to provide the rates of reactions. The reason is a difference in the way the possible interactions between reactants are counted. The kinetic law of Mass action draws this number following a combination without repetition. For example when applying a reaction $A, A \xrightarrow{x} AA$ to a solution $S = \{A^3\}$, Mass action counts $\binom{3}{2} = 3$ interactions. By contrast, when defining A in the stochastic π-calculus to perform the very same reaction, i.e., $\mathsf{A(s,sN1,sN2)} =: \mathsf{s!(sN1).A(sN1,s,sN2)} + \mathsf{s?(sN2).A(s,sN1,sN2)}$, the stochastic π-calculus follows a variation without repetition. That is, the set of communication labels yields $\Lambda(S) = \{(i,1,j,2) \mid i,j \in \{1,2,3\}, i \neq j\}$, such that $\sharp\Lambda(S) = \binom{3}{2} * 2! = 6$. As a direct consequence of this, corresponding rate constants in the stochastic π-calculus have to be divided by 2. This is sufficient, since in the stochastic π-calculus reactions may have at most two reactants in general and thus also at most two reactants of the same kind in particular.

From the two lemmas above, it directly follows that the set of reactions extracted from a solution S correctly reflects the CTMC of S and thus the dynamics of stochastic π-calculus models, see corollary below. This is because the transitions of S dissect $\Lambda^\sim(S)$ again into a set of equivalent classes, each class holding those reactions leading to the same successor state. Considering, e.g., solution $S = \mathsf{A(b,b,u1,u2)} \mid \mathsf{B(b)}^2$ as in our preceding examples, we, in fact, obtain for each of the two reactions a distinct transition, since the

reactions differ in their products. In general, one can, however, think of reactions that only differ in their channel, or that transform a solution S into the same solution S', although their reactants and products differ. These would then relate to the same state transition.

Corollary 3. For all solutions S in prenex normal form it is true that the quotient set of communication labels $\Lambda_\bullet^\sim(S)$ correctly reflects the stochastic dynamics of S, i.e., for all transitions $S \xrightarrow{r} S'$ it holds that the maximal subset of $\Delta \subseteq \Lambda_\bullet^\sim(S)$, with $\Delta = \{(x,l) \mid (x,l) \in \Lambda_\bullet^\sim(S), S \xrightarrow[l]{x} S'\}$, fulfills the following:

1. (timed correctness)

$$\sum_{\{(x,l)\mid(x,l)\in\Lambda(S),S\xrightarrow[l]{x}S'\}} \kappa(x) = \sum_{\{\ell\mid\ell\in\Delta,\}} \rho(\ell,S)$$

2. (probabilistic correctness)

$$\frac{\sum_{\{(x,l)\mid(x,l)\in\Lambda(S),S\xrightarrow[l]{x}S'\}} \kappa(x)}{\sum_{\{(x,l)\mid(x,l)\in\Lambda(S)} \kappa(x)} = \frac{\sum_{\{\ell\mid\ell\in\Delta,\}} \rho(\ell,S)}{\sum_{\{\ell\mid\ell\in\Lambda_\bullet^\sim(S),\}} \rho(\ell,S)}$$

To summarize, our way to extract reactions from species sets $S = (\nu\tilde{x}) \prod_{i=1}^n M_i^{m_i}$ is to consider the interaction capabilities of each pair M_{i_1}, M_{i_2} and to count the numbers of interaction capabilities that correspond to the same reaction. Our second of visualization, as introduced in the following section, subsumes for each state a species set that contains beyond the current solution also the products of all possible reactions. Molecule numbers are, however, assigned dependent on the occurrence in the current state. That is, in particular the number of a molecule only occurring as a product of a reaction and not in the current solution is 0. We obtained an implementation by extending the simulator presented in [LJU10]. The results of applying it to an example model based, including their visualization, are given in Section 4.4.

4. Visualization

The result of the transformation in Section 3 is a set of reactions for each time point of the simulation. This representation describes the simulation entirely. Nevertheless, it is not well suited to explore and understand the simulation. The consumed and produced species are not given directly, although they are as important as the reactions to the analyst. We therefore use a representation that explicitly contains both reactions and species, and further the relations among them. These reaction networks, one for each time point, better fit the demands of the modeler.

Each reaction network contains:

- a set of reactions, each coming with a description and a static *parameter* called reaction rate constant that identify the reaction, and a time dependent *property* representing the current reaction rate
- a set of species, which are given by a name and a set of static *parameters*, and a *property* value representing the amount of the species, which changes over time
- a set of directed links between species and reactions to indicate which reactions consume or produce which species

Depending on the number of species to start with and the duration of the simulation, the amount of data amassed by this process grows large very quickly. To validate the simulation results and by this the model, interactive visualization can help to explore the simulation data. However, the whole time series of reaction networks with all its individual changes is too complex to be shown in one visual representation. In the sequence of networks, structural changes (species/reactions appearing and disappearing), and changes of property values may occur. Even if a comprehensive visual representation was found, it would confront the user with an amount of information that is very challenging to conceive in its entirety. Because of this, it is sensible to support the analyst with a tailored visualization and interaction methods that naturally guide her through the typical analysis sequence. Providing tailor-made visualizations for each step of the analysis sequence leads to a more efficient analysis, as only the currently necessary parts of the data are shown in the visual representation. This reduces the cognitive load and eases interaction, such as selecting an individual data item, compared to displaying everything at once.

Hence, we tailor our concept to the typical task sequence that the analyst carries out during the analysis of a simulation. It usually comprises of the following three steps:

1. **identification** of a time point of interest that for some, still to be determined reason deviates from the expected model behavior
2. **inspection** of this time point to pinpoint the cause of the deviation to a part of the reaction network
3. **comparison** of this time point to other (usually earlier) time points, in order to identify the cause of the reaction network evolving into this unexpected state

The implications for a visualization supporting these tasks are discussed in the following section.

4.1. Visualization Concept

In order to transform the analysis sequence given above into a tailored visualization and interaction methods, the three steps imply an emphasis on the temporal changes of the system state and on the network structure at each time point. Following this general scheme, we construct our visualization based on two views: for the *identification* of time points of interest, an **overview** of the dynamics in the simulation data is provided. This overview effectively characterizes each time point by several numerical data, thus allowing the modeler to make an informed choice when identifying and selecting a time point of interest. Upon selection, a **detail** view gives insight into the actual structure of the respective reaction network – hence allowing for the *inspection* of single time points. This includes information about parameters and properties, so that the time point can be inspected in all its facets.

The third step of the analysis sequence is the *comparison* of a time point of interest to other, usually prior time points. To this end, our concept allows to identify and select a second time point from the overview, in addition to the priorly inspected time point. The visual comparison is supported by linking and coordinating one detail view for each of the two time points. This is reasonable as the underlying data (reaction networks) is the same, just multiple instances of it have to be handled. The limitation that only two time points are concurrently visualized is made due to high cognitive demands for comparing reaction networks. Nevertheless, the highly interactive interface provides the flexibility to successively inspect and compare numerous time points.

The 2nd and 3rd step of the analysis sequence, in-depth inspection and comparison, usually require to investigate the role of certain species and reactions over time and within the network structure. Our multiple view concept accounts for these analysis requirements. In all views, the user can interactively construct and adjust selections of reaction and species. They are then highlighted in the overview in their temporal development and in the detail views as individual entities. This way, a rich analysis of species and reactions of interest – taking into account dynamics as well as structural dependencies – is supported. To compare the behavior among subsets of network elements, two selections can be instantiated, denoted as the *red* and *blue* selection in the following. These two colors will be used to highlight all data representations and navigation elements that are related to the selections.

Figure 7 shows the realization of the overall visualization concept. It comprises the overview over all time points and the detail views of two single time points. The horizontal split maximizes the available width of the overview to facilitate the identification of time points, while positioning the two detail views side by side supports their parallel exploration. It is sensible to display overview and detail simultaneously as the exploration is a constant back and forth between them. The time points that are inspected in the detail views are visually highlighted in the overview (vertical orange lines). This helps the user to maintain a complete mental model of the data and to integrate the currently investigated parts of the data into the overall context.

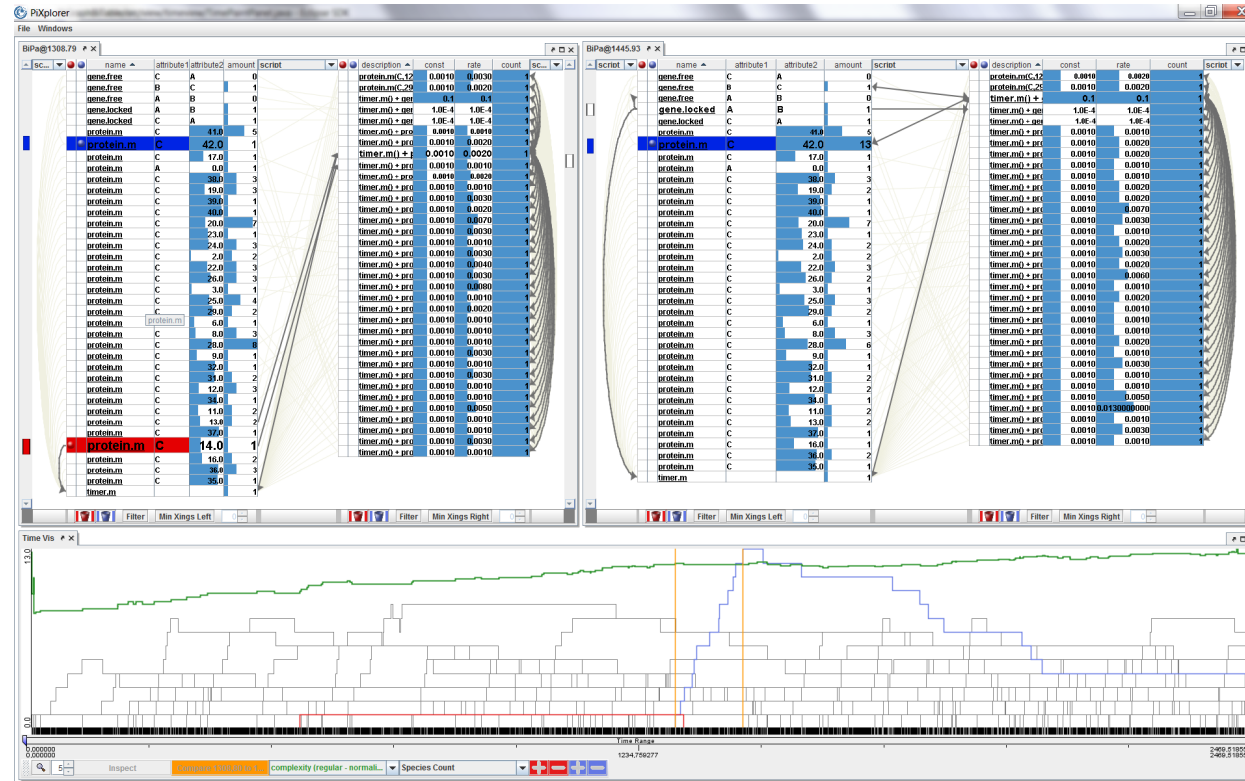The following sections discuss the two different types of views, overview and detail view, in more detail.

Fig. 7. Screenshot of the visualization tool to give an impression of the overall visualization scheme. The overview is shown at the bottom. It displays numerical data that characterizes the temporal developments. Here, individual time points can be identified and selected (highlighted in orange), whose reaction networks are then shown in all their facets in the table-based detail views above. The structural aspect of the network is encoded in the links between the tables, the parameters and properties of species and reactions are displayed with a light blue gauge-like representation in the table cells. The simultaneous visualization of two time points in parallel side by side supports their detailed comparison. Further, subsets of reactions and species, highlighted in *red* and *blue*, can be constructed and adjusted throughout all views to follow both their dynamics as well as their structural dependencies in different time points. In the screenshot, two reaction networks are compared to investigate the role of a species (*blue* subset), whose amount changes significantly between the two time points. The one species that appears only in the former reaction network is highlighted in *red*. The structural interrelations in the detail views provide the necessary information to the analyst to evaluate whether the increase of the species amount is plausible.

4.2. Overview

The goal of the overview is to explore the dynamics of the reaction network and its components, so that the user can **identify** time points of interests. This section describes how we derive appropriate numerical data to characterize the dynamics, how we visualize them, and which interactive facilities are provided by the overview.

To express the general temporal development of the reaction network, we employ so-called *complexity measures*, interpreting the reaction network as a graph structure with the species and reactions as nodes and the links between them as edges. Besides simple graph characteristics, like the relative number of edges compared to all possible edges or the average node degree, more subtle measures condense several structural facets, such as the branching factor and the number of cycles, into one single value. An overview of such measures is given in [BB05]. One example is the information content I_{vd} of the distribution of vertex degrees $deg(v_i)$ of a graph:

$$I_{vd}(V) = \sum_{i=1}^{V} deg(v_i) \log_2 deg(v_i)$$

The other part of the data that reflects changes over time, the dynamics of individual species and reactions, can be effectively characterized by their properties, describing their amount (or rate, respectively) for each time point at which the species (or reaction) is part of the reaction network. If species or reactions are not constantly part of the reaction network, their property value at those time points is 0. Consequently, the numerical data comprises a set of complexity measures, a set of species properties, and a set of reaction properties. Each complexity measure is represented by one time series. The set of species properties consists of one time series for each species occurring during simulation, the set of reaction properties one for each reaction that appears over time.

To visualize the time dependent numerical data, time value plots are well suited. Plotting data values on a vertical data axis over a horizontal time axis is a visualization familiar to most users. However, numerous data scales make it necessary to have one time value plot for each complexity measure, one for the set of species properties, and one for the set of reaction properties - but only the combined numerical data characterizes a time point. To make the different dynamic aspects visible at a glance, multiple time value plots are overlaid. A major advantage is that the common time scale is conveyed, which facilitates the identification of time points. At the same time, each of the individual plots covers maximum screen space and does not become illegibly small, as it might happen with a side by side arrangement. But the user's effort to relate the plots to the corresponding numerical data grows quickly with the number of plots. To limit the number of concurrently shown time value plots while still providing a good characterization of time points, we bring together one time value plot from each of two complementary parts (complexity measures and property sets). Color is used to distinguish the two time value plots (properties in black and complexity measure in green). The other property set as well as other complexity measures can be sequentially explored, based on user interaction. In Figure 8, the data scale of the species or reaction property is shown at the left, the data scale of the complexity measure at the right.

A second challenge, in addition to handling numerous data scales, is the uneven distribution of time points, which follows from the stochastic nature of the simulation. To communicate the irregular distribution of time points and facilitate the identification of time points in dense intervals, the overview is accompanied with several visual add-ons (see Figure 9):

- A 1-dimensional **heat map** reflects the uneven distribution of time points. Located below the time axis, the heat map indicates how many time points are mapped onto that pixel, ranging from white (multiple time points) to black (no time point at that pixel).

- With the **range slider** (below the time value plot), the user can narrow the time range shown in the visualization to an interval of interest, thus broadening the available space for these time points.

- The **interactive lens** supports a fast visual separation of narrow time points without losing the overview over a larger time range. Controlled via mouse movements over the time value plot, a focus+context mechanism stretches the area within the lens horizontally to discern close time points, while the outside area is compressed.

Visualizing the numerical data over time in combination with interaction techniques to separate close
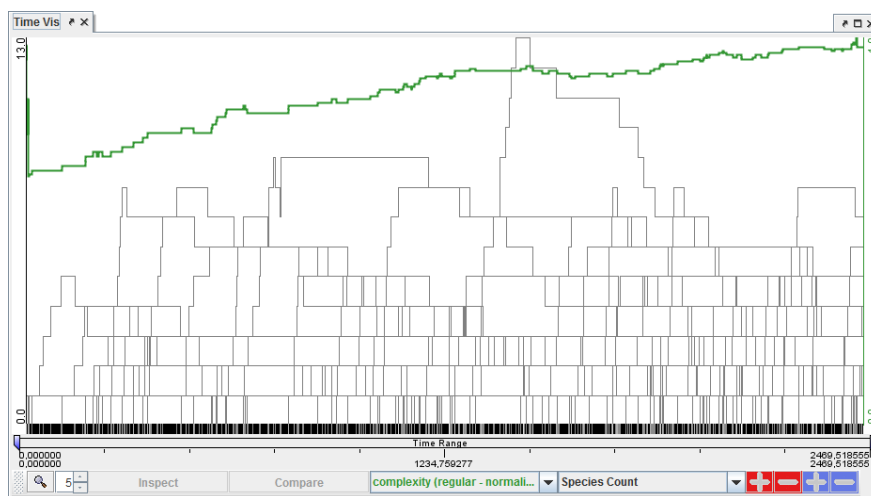
Fig. 8. The overview visualizes numerical data over time to characterize the dynamics of the simulation data. To bring together the two main dynamic aspects of network complexity and individual properties, multiple time value plots are overlaid. One time value shows the currently selected complexity measure (in green, with its axis annotation on the right) to given an impression of the development of the network complexity. It is apparent that, after a drop-off at the beginning, the complexity of the network generally grows. The other time value plot comprises multiple time series, one for every element of the currently selected property set (in black, axis annotation on the left). Currently, species properties are selected. It can be seen that the amount of the species significantly vary among individual species as well as over time.
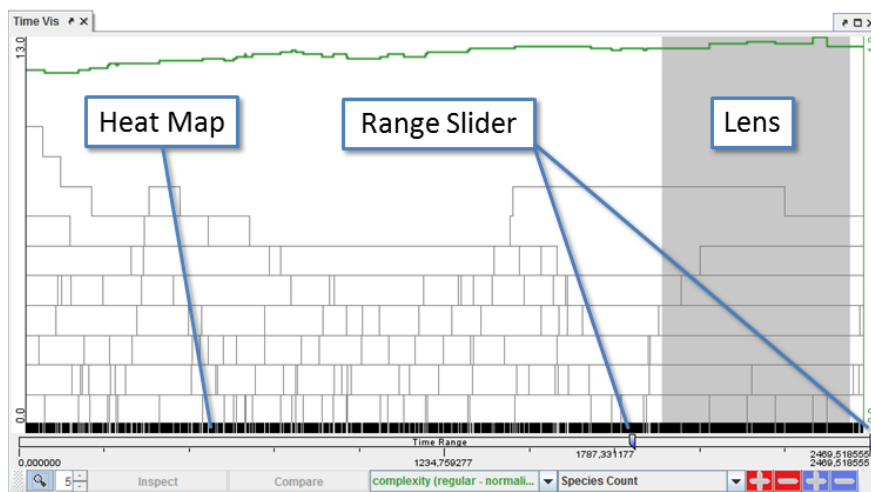

Fig. 9. Visual add-ons in the overview to support the identification of time points. The heat map shows the uneven distribution of time points over the temporal range. From Figure 8, one can see that many time points occur towards the end of the simulation, which are hard to discern. Hence, a range slider allows to narrow the visible time range to an interval of interest to better discriminate time points. This is further supported by an interactive lens, which enlarges a temporal interval on demand, while the overall temporal context is maintained. Applying these interactive features gives the analyst the opportunity to identify individual time points from the overall simulation even in dense time intervals, as it can be seen in the screenshot.

time points supports an effective inspection of the data dynamics and, by this, the identification of time points of interest.

As part of the multiple view concept, the overview provides interaction features to select time points as well as subsets of species and reactions. In accordance to the analysis sequence, time points are selected for either inspection or comparison, between which the user can interactively switch in the visual interface. Both objectives demand the selection of a single time point. Either, one time point is selected for inspection, or another time point is compared to the previously inspected time point. Thus, a unified selection mechanism is applied for both objectives: The time point of interest is brushed in the heat map, because it is the visual

representation of the time axis. Another important interactive task is the construction and adjustment of subsets of species and reactions, which are represented in the overview by their dynamic property values. To carry it out, brushing is used here as well: The user directly brushes time series of interest in the time value plot and then chooses whether the *blue* or *red* subset is adapted. Using designated brushing areas for the selection of time points and for the selection of subsets leads to clear and, at the same time, very flexible interaction methods, which allow the user to adapt the currently investigated parts of the simulation data with few mouse clicks.

4.3. Detail Views

The detailed view allows the user to **inspect** the changing reaction network at individual time points.

As we have presented a very scalable and highly interactive visualization technique for networks before [SJUS08], we extend this approach to serve as detail views in our visualization. The said approach offers a table-based view on the network by using two interlinked tables, one for the species and one for the reactions. This enables us to display the parameters and properties (species' amount and reaction rate) alongside the name in the table. Their relations, indicating which species partake in which reaction, are shown as connecting lines in between both tables. Their width indicates their weight, which is in this case the stoichiometric factor with which the species participate in a reaction. So-called 1-mode-projections are displayed on the outer sides of the tables. 1-mode-projections are basically shortcuts that enable a user to analyze the reaction network without always jumping back and forth between the two tables. They are constructed by contracting paths of length 2 into a 1-mode-projection. E.g., if there is a path from one species to another species via some reaction, both species are connected by such a shortcut, directly connecting reactants and products. The same goes for the reactions, where the projections connect subsequent reactions – a first reaction which produces a species that is consumed by the second reaction.

In addition to this overall visualization layout, a number of interaction features are incorporated with this basic setup (see Figure 10):

- A **table lens** [RC94] is provided for big tables, applying a focus+context mechanism to the tables. It shows details in the focus region by enlarging the rows around the mouse cursor and compressing all other rows into a context visualization. This reduces the overall height of the tables, which in turn reduces cognitive load and vertical scrolling.

- **Edge-based traveling** [TAS09] is provided to aid in rapid navigation between the two tables of a detail view. Its effect is basically that a link becomes clickable if one of its incident rows lies outside of the current view. Usually this would require the user to scroll the table that contains the off-screen row until the row appears on the screen. But with edge-based traveling, clicking the link takes care of the scrolling and brings the off-screen row automatically into view. In order to know in beforehand if the off-screen row is of interest, hovering over the clickable link will display the label of this row.

- An extensive **selection concept** provides all necessary functions for interactive and predefined, automated analysis of the shown reaction network. To this end, it features manual selection to highlight rows of interest, as well as a script-based selection that performs a propagation of manual selections along the network according to script-defined procedures. The scripting language uses a set-based notation that is carried out in a breadth-first manner. To enable comparison between different selections, two selection states are available, highlighted in *red* and *blue* to account for the global color scheme that links selections in the detail view to selected properties in the overview.

- **Fisheye scrollbars** with selection markers [Byr99] allow the user to get a visual overview of currently selected rows in both tables. This is extremely helpful, as selections can span the entire table and a sorting of the table according to the selection status of its rows is not always desirable. Hence each scrollbar features visual markers of the current selection, which can also be clicked on to take the user directly to the corresponding rows without scrolling. As the selection markers can be quite crowded which makes them hard to click individually, a local fisheye distortion is applied to the scrollbars to counter this effect.

- A **filtering** can be applied to the tables, generating a condensed view of selected rows only. Hence, the filtering is directly coupled with the selection scripts, allowing for complex filtering operations based on structural and attribute conditions.

Together with the table-based visualization approach, this allows the user to investigate the reaction
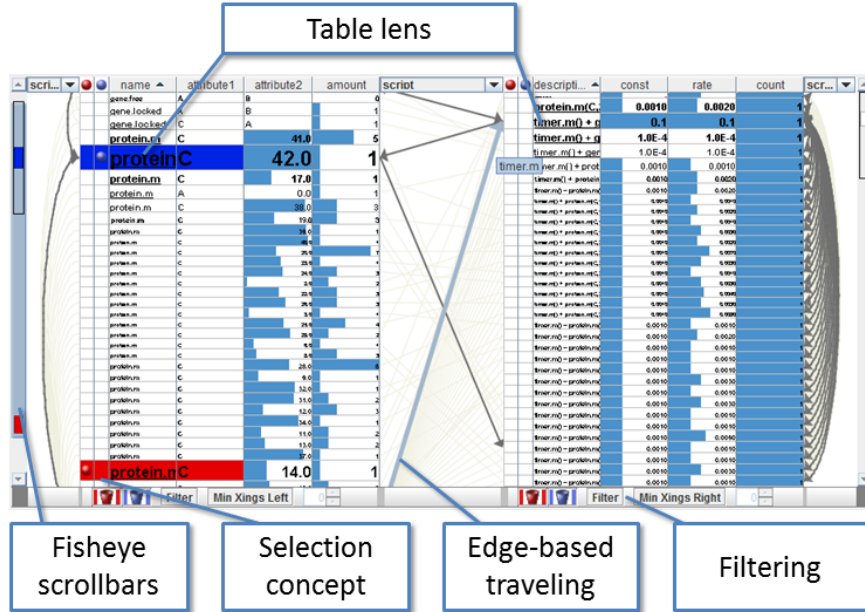
Fig. 10. Table-based detail view of a reaction network at one time point. The left table lists all species that are currently part of the reaction network. Each row represents one species, the columns provide descriptions, parameters, and property values. Numerical values are visually encoded in light-blue within the cells. The visual representation of reactions in the right table is analogue. Relations between species and reactions are shown by lines between the two tables. On the outer side of the tables, 1-mode projections indicate "shortcuts" from one species to another, or from one reaction to another. The screenshot shows the first selected time point from Figure 7 including the introduced *red* and *blue* species selections. The selections are highlighted within the table and in the fisheye scrollbar in terms of color. The *blue* species is currently in the focus of the table lens in the left table and therefore enlarged to spot its details. Also, its links to reactions and its 1-mode projections to other species are accentuated, to let the analyst explore its structural interrelatedness. One link leads to the reaction in which the species was produced (in the lens focus in the right table). The links of this reaction lead to species currently outside the screen. The analyst can now easily navigate there by one click on the edge (edge-based traveling). To solely inspect selections, disregarding all non-selected parts, a filtering mechanism can be applied to each table. It would result in the two selected rows remaining in the left table and in an empty right table, as it contains no selected rows.

networks at individual time points in detail. The different interaction methods and their tight integration with one another provide various analysis paths that lead to quick results even for large reaction networks.

4.4. Applying the Visualization to an Example Model

This section illustrates the presented visualization tool by investigating an example model of polymerization, where molecules of the same sort bind to a single complex.

The example model works as follows: a molecule M (monomer) has three binding sites, to each of which another M may bind. Natural restrictions, e.g., that due to geometric constraints two molecules may not form rings, are not considered here. Following the schema as presented in the initial example of Section 2, molecule M can be modeled in the stochastic π-calculus as follows:

```
M(s1,s2,s3,s1N)  =: s1!(s1N).M(s1N,s2,s3,s1)  // link s1
                  + s2?(s2N).M(s1,s2N,s3,s1N)  // link s2
                  + s3?(s3N).M(s1,s2,s3N,s1N)  // link s3
```

Entirely free monomers are represented by instances M(free,free,free,b), where all monomers are considered to share the name free but to be equipped with a unique name b. Consider, e.g., a solution $S = $ M(free,free,free,b1) |
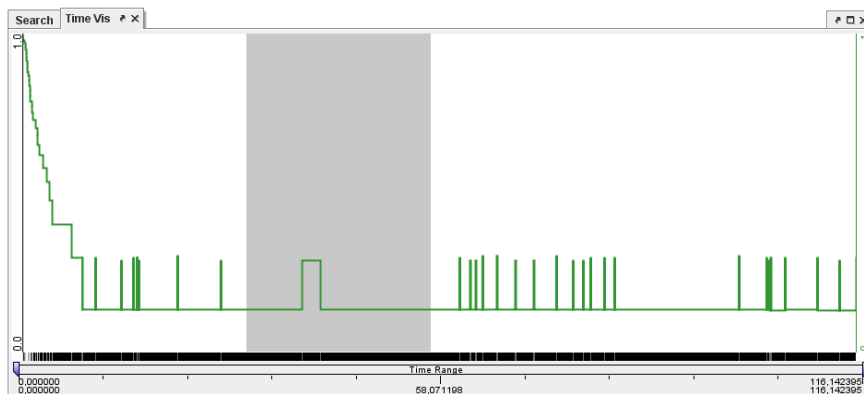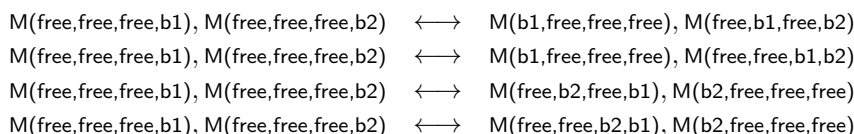
15

Fig. 11. Overview showing the structural complexity of the evolving reaction network over time. The gray region is the lens region with the horizontal distortion, so that release and binding events which are otherwise condensed into spikes become visible as separate time points.

M(free,free,free,b2). The following eight different reactions may happen in S:

$$\text{M(free,free,free,b1)}, \text{M(free,free,free,b2)} \quad \longleftrightarrow \quad \text{M(b1,free,free,free)}, \text{M(free,b1,free,b2)}$$
$$\text{M(free,free,free,b1)}, \text{M(free,free,free,b2)} \quad \longleftrightarrow \quad \text{M(b1,free,free,free)}, \text{M(free,free,b1,b2)}$$
$$\text{M(free,free,free,b1)}, \text{M(free,free,free,b2)} \quad \longleftrightarrow \quad \text{M(free,b2,free,b1)}, \text{M(b2,free,free,free)}$$
$$\text{M(free,free,free,b1)}, \text{M(free,free,free,b2)} \quad \longleftrightarrow \quad \text{M(free,free,b2,b1)}, \text{M(b2,free,free,free)}$$

We consider a single simulation run of the model, starting with solution $S = \prod_{i=1}^{30} \text{M(free,free,free,b}i)$ and assuming rate constants $\kappa(\text{free}) = 1.0$ and $\kappa(\text{b}i) = 0.01$ for the binding and unbinding reactions, respectively. An overview is given in Figure 11 showing the structural complexity of the evolving reaction network over time. Two observations can be made from this curve: that it is rapidly decreasing in the beginning and then oscillates roughly between two values. It is important to notice, that the more monomers are bound, the less complex the reaction network is, because less binding reactions are possible. Hence the plot is plausible, showing the initial solution with all monomers being free and thus making many reactions possible. The more of them bind together (which is probable as the rate constant of the release reaction is $100\times$ lower than the one of the binding reaction), the fewer reactions are still possible and hence the lower the structural complexity of the reaction network gets.

The second observation of the oscillation can be explained along the same line: because the rate constant for binding is much higher than for releasing a monomer, the polymer (and thus the solution it is in) is quite stable. Only every now and then, a monomer gets released, indicated by the increase in structural complexity because of the many now possible binding reactions. The (re-)binding follows usually shortly after the release and the complexity drops off to a lower value, again. This results in the spiked nature of the oscillation and can be confirmed by looking at the detail view for a time point with such a sudden increase of complexity. After **identifying** such a time point of interest in the overview (e.g., the left one within the lens region in Figure 11), the structure of the reaction network is displayed as shown in Figure 12. Upon **inspection** of the network, the monomer just having been released is easy to identify, because it is the only present monomer (amount > 1) with its first binding site not being allocated (attribute1 = free). One can see the large number of outgoing edges to possible binding reactions, which are the reason for the increased structural complexity.

As the inspection has now yielded the concrete free monomer, it is possible to **compare** its state with the time point before and after this one. To do this, both of these time points can be selected and opened up as detail views from the overview. Both are shown side-by-side in Figure 13. The time point before the peak clearly shows the still bound monomer (blue) and its soon to happen release reaction (violet). As it can be seen, the product of this reaction are two monomers, one of them (the red one) the unbound monomer from Figure 12, at this time step still with amount = 0, as the monomer is not yet released. It is easy to see, that while a monomer has many binding reactions, it has only a single release reaction – hence the difference in structural complexity. The same goes for the time point after the release: the free monomer (now having amount = 0) has bound to the polymer again and now permits only one release reaction instead of
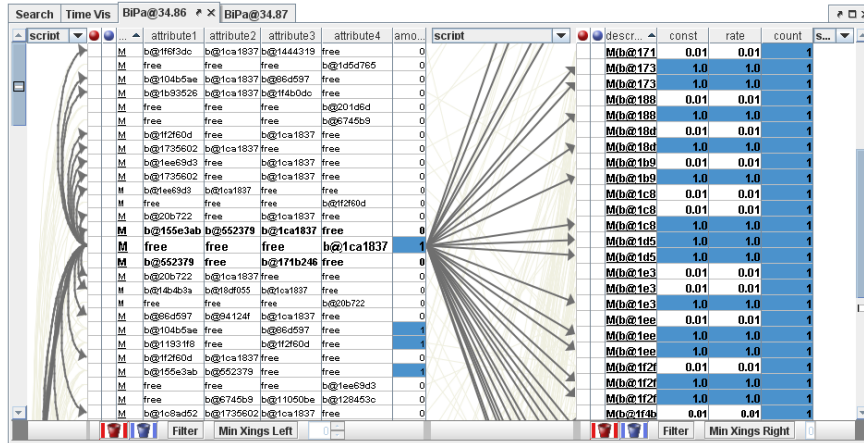
Fig. 12. Detailed view showing the peak highlighted in Figure 11.

multiple binding reactions as before, effectively causing the drop-off in structural complexity. This confirms the assumption that alternating release and binding reactions cause the observed oscillatory pattern in the complexity plot.

4.5. Discussion

All in all, the presented method to visually explore dynamic biochemical reaction networks scales well to the size as well as to the heterogeneous character of the data, with its structural and numerical facets. This is noteworthy in the sense, that dynamic reaction networks can be interpreted as time-varying graphs, which are notoriously hard to visualize, especially if it gets large and encompasses many time steps. Traditionally, in the area of Graph Drawing, the visualization of time-varying graphs is associated with the problem of providing a sequence of (stacked) views that maintain the mental model of the user. This means that a graph layout must be found that minimizes unnecessary positional changes across all time points in order to ease comparison and traceability of the changes. Besides specifically tailored layout methods (e.g., the Foresighted Layout with Tolerance [DG02]) also special visual metaphors (e.g., the Worm Metaphor [DE02]) have been developed to cope with this challenge. Yet, these standard solutions do neither scale-up as much as needed to cope with the amount of data produced in this scenario, nor are they able to allow the visual exploration of structural changes and changes in the node properties at the same time. This is, why a whole new visualization concept was needed, which copes with the possibly huge number of time steps by condensing and plotting the structural information into complexity measures, and which deals with possibly large networks by utilizing a table-based visualization approach, which has been shown to scale up to about 100,000 nodes [EK02]. By breaking the data down into its temporal and structural dimension and showing these two facets in multiple linked views, the visual interface gives the user intuitive access to the complex data. In this combination, the presented concept stands as a novel visualization solution in the area of interactive visualization.

5. Applicability to extensions of the π-calculus

Different extensions to the π-calculus for the stochastic modeling of cell-biological processes have been proposed. SPiCO [KLN07] associates sets of functions to channels. In [Kut06] an encoding from SPiCO to the stochastic π-calculus is presented that only increases the amount of used channel names, thus yielding larger reaction sets. Since our approach is explicitly designed to deal with large reaction sets it is applicable to SPiCO as well.

Other extensions cannot be translated back into the stochastic π-calculus. π@ [Ver07] and its stochastic version Sπ@ [VB08] allow for communication on tuples of channels and thus increase the size of reaction sets similar to SPiCO. The attributed π-calculus [JLNU10] extends the idea of process parameters to molecule
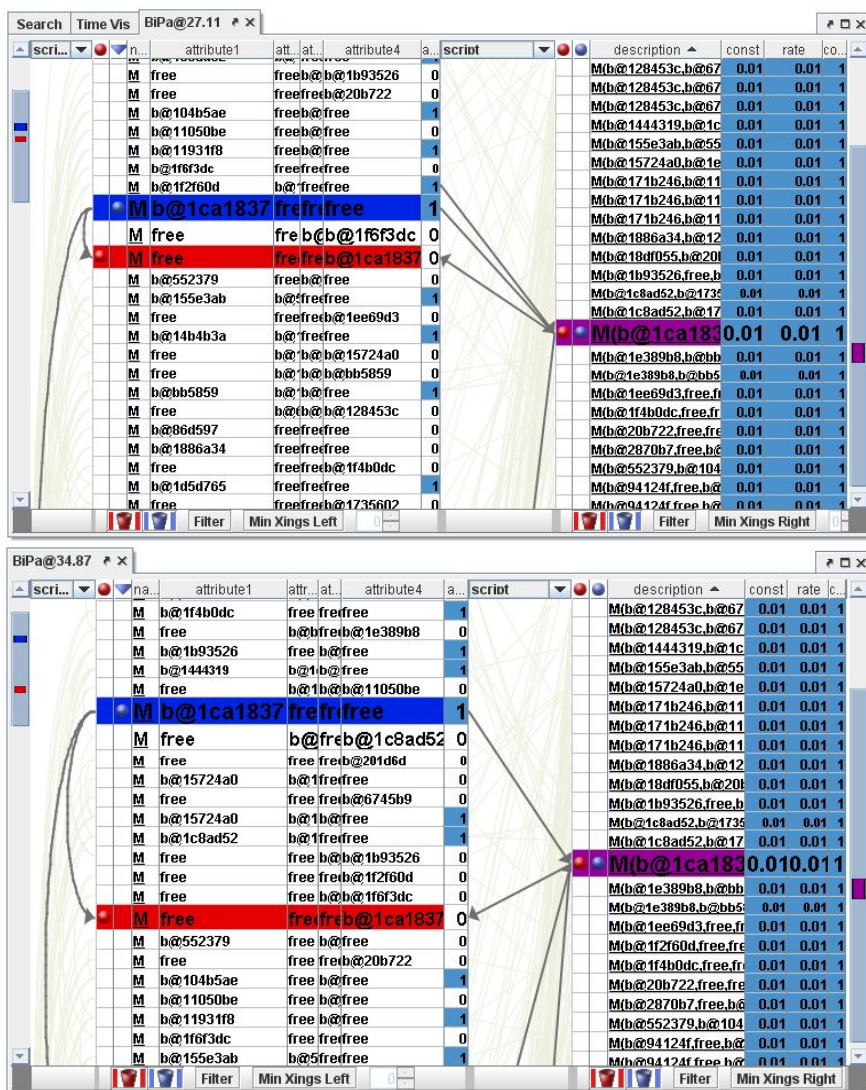
17

Fig. 13. Detailed views showing the states of the reaction network before (*top*) and after (*bottom*) the peak from Figure 12.

attributes which can be of any type not just channels. Communication constraints are introduced that allow to make the rate of a communication dependent on the attribute values of interaction partners. To all of these extensions our approach can be well applied since it does not depend on how communication labels are obtained and a biochemical form can be formulated for them in the same way as for the stochastic π-calculus, see e.g., [JLNU08]. In fact, our visualization is in particular designed for the attributed π-calculus considering molecules with attribute values. Notice that SPiCO, $\pi@$, S$\pi@$, and also the attributed π-calculus allow for prioritized interactions.

BioAmbients [RPS$^+$04], Brane Calculi [Car04], and BlenX [DPR08] associate locations with processes. This information may be integrated into our concept in two different ways. On one hand, one can try to reflect the location of processes as additional species attributes. In fact, the same idea is used for an encoding of BioAmbients into the attributed π-calculus, see [JLNU10]. On the other hand, spatial information could also be associated with reactions. Although the latter approach requires to extend the notion of reaction labels accordingly, it does not impact the visualization since reaction nodes are considered to have properties which are represented in the same way as the attributes of species nodes, see below. Yet, an extension of the

visualization that explicitly aims at the exploration of spatial information would be desirable in any case but is subject to future work, see Section 6.

The imperative π-calculus [JLN09] extends on the attributed π-calculus by allowing processes to read and change values in a global imperative store. Effectively, this means that states are not entirely described by processes but by pairs of processes and stores. In order to apply our approach to the imperative π-calculus both the construction and the visualization of reaction networks needs to be extended. For the construction, communication and reaction labels need to capture possible effects on the global imperative store and their relations have to be adapted accordingly. These changes are then to be reflected in the components of the reaction network and the visualization.

6. Conclusion and Future Work

We conclude, that our two step approach of first constructing and then visualizing reaction networks from the states of stochastic π-calculus-models succeeds in bridging the gap between the notion of communicating processes and models of biochemical systems. Furthermore, in our example, we observe that the synergy of formal and visual methods allows the modeler to gain additional confidence in the obtained results. The clear formal definition of our transformation, its proof of correctness, and the fact the concepts defined in the formal step build the direct foundation of the visualization add to this point. The high interactivity of the presented visualization concepts engages modelers more than a static plot and the combination of overview and detail contributes to the overall understanding and detailed insights into the dynamics of biochemical systems.

In future work, we aim to provide even more direct ways of a back-and-forth transformation between abstract models and biochemical reaction networks. A first step in this direction has already been taken by proposing a new rule-based language [JLNV11], which can be shown to be as expressive as the π-calculus while at the same time being much closer to the cell-biological domain. Another challenging aspect for visualization as a whole, as well as our visualization concept in particular is the inclusion of multiple simulation runs. This would introduce another, even higher layer of overview, as the set of all simulation runs would rank above the individual run shown in the overview so far. This is not only an issue in terms of screen real estate, but also in terms of providing linking and interaction between all different views. Lastly, the meaningful inclusion and depiction of spatial information, i.e., compartments, beyond being just another table column, is of great interest for more advanced models involving transport processes, such as shuttling. In general, all these possible extensions of our visualization concept would enhance the exploration possibilities of the network structures that result from stochastic simulation and thus broaden the application of our concept to other domains.

References

[BB05]   Danail Bonchev and Gregory A. Buck. Quantitative measures of network complexity. In Danail Bonchev and Dennis H. Rouvray, editors, *Complexity in Chemistry, Biology, and Ecology*, pages 191–235. Springer, 2005.

[Byr99]   Donald Byrd. A scrollbar-based visualization for document navigation. In *DL'99: Proceedings of the fourth ACM conference on Digital libraries*, pages 122–129, 1999.

[Car04]   Luca Cardelli. Brane Calculi - Interactions of Biological Membranes. In *Computational Methods in Systems Biology, International Conference, CMSB'04*, LNCS, pages 257–278, 2004.

[Car08a]   Luca Cardelli. From processes to ODEs by chemistry. In *IFIP Theoretical Computer Science*, pages 261–281, 2008.

[Car08b]   Luca Cardelli. On process rate semantics. *Theoretical Computer Science*, 391(3):190–215, 2008.

[CCG+09]   Luca Cardelli, Emmanuelle Caron, Philippa Gardner, Ozan Kahramanogullari, and Andrew Phillips. A Process Model of Rho GTP-binding Proteins. *Theoretical Computer Science*, 410(33-34):3166–3185, 2009.

[DE02]   Tim Dwyer and Peter Eades. Visualising a fund manager flow graph with columns and worms. In *IV'02: Proceedings of the 6th International Conference on Information Visualisation*, pages 147–152, 2002.

[DG02]   Stephan Diehl and Carsten Görg. Graphs, they are changing – dynamic graph drawing for a sequence of graphs. In *GD'02: Proceedings of the 10th International Symposium on Graph Drawing*, pages 23–31, 2002.

[DPR08]   Lorenzo Dematté, Corrado Priami, and Alessandro Romanel. Modelling and Simulation of Biological Processes in BlenX. *SIGMETRICS Performance Evaluation Review*, 35(4):32–39, 2008.

[EK02]   Stephen G. Eick and Alan F. Karr. Visual scalability. *Journal of Computational and Graphical Statistics*, 11(1):22–43, March 2002.

[FHR⁺03] James R. Faeder, William S. Hlavacek, Ilona Reischl, Michael L. Blinov, Henry Metzger, Antonio Redondo, Carla Wofsy, and Byron Goldstein. Investigation of early events in fcεri-mediated signaling using a detailed mathematical model. *Journal of Immunology*, 170(7):3769–3781, 2003.

[Gil77]   Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[HFB⁺03] William S. Hlavacek, James R. Faeder, Michael L. Blinov, Alan S. Perelson, and Byron Goldstein. The complexity of complexes in signal transduction. *Biotechnology and Bioengineering*, 84(7):783–794, 2003.

[JLN09]   Mathias John, Cédric Lhoussaine, and Joachim Niehren. Dynamic compartments in the imperative pi-calculus. In Pierpaolo Degano and Roberto Gorrieri, editors, *Computational Methods in Systems Biology, International Conference, CMSB'09*, volume 5688 of *Lecture Notes in Computer Sience*, pages 235–250. Springer Verlag, 2009.

[JLNU08]   Mathias John, Cédric Lhoussaine, Joachim Niehren, and Adelinde M. Uhrmacher. The attributed pi calculus. In *Computational Methods in Systems Biology, International Conference, CMSB'08*, volume 5307 of *Lecture Notes in Computer Science*, pages 83–102. Springer Verlag, 2008.

[JLNU10]   Mathias John, Cédric Lhoussaine, Joachim Niehren, and Adelinde M. Uhrmacher. The attributed pi-calculus with priorities. *Transactions on Computaional Systems Biology XII. Special Issue on Modeling Methodologies*, 5945:13–76, February 2010. LNCS (Lecture Notes in Bioinformatics), Springer Berlin/Heidelberg.

[JLNV11]   Mathias John, Cédric Lhoussaine, Joachim Niehren, and Cristian Versari. Biochemical reaction rules with constraints. In *Proceedings of the European Symposium on Programming*, pages 338–357, 2011.

[KLN07]   Céline Kuttler, Cédric Lhoussaine, and Joachim Niehren. A stochastic pi calculus for concurrent objects. In Hirokazu Anai, Katsuhisa Horimoto, and Temur Kutsia, editors, *Second International Conference on Algebraic Biology*, number 4545 in Lecture Notes in Computer Science, pages 232–246. Springer Verlag, 2007.

[Kut06]   Céline Kuttler. Simulating bacterial transcription and translation in a stochastic pi-calculus. *Transactions on Computational Systems Biology*, 4220/2006:113–149, 2006.

[LJU10]   Stefan Leye, Mathias John, and Adelinde M. Uhrmacher. A flexible architecture for performance experiments with the pi-calculus and its extensions. In Barry Lawson, editor, *3rd International ICST Conference on Simulation Tools and Techniques, ICST, Malaga, Spain*. ICST/IEEE, 2010.

[MG09]   Roland Meyer and Roberto Gorrieri. On the relationship between π-calculus and finite place/transition petri nets. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR - Concurrency Theory, 20th International Conference*, number 5710 in Lecture Notes in Computer Science, pages 463–480. Springer Verlag, 2009.

[Mil99]   Robin Milner. *Communicating and Mobile Systems: the π-calculus*. Cambridge University Press, 1999.

[MJM⁺09] Orianne Mazemondet, Mathias John, Carsten Maus, Adelinde M Uhrmacher, and Arndt Rolfs. Integrating diverse reaction types into stochastic models - a signaling pathway case study in the imperative pi-calculus. In M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, editors, *Winter Simulation Conference*, pages 932–943. Institute of Electrical and Electronics Engineers, Inc., 2009.

[PC07]   Andrew Phillips and Luca Cardelli. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In Muffy Calder and Stephen Gilmore, editors, *Computational Methods in Systems Biology, International Conference, CMSB'07*, volume 4695 of *Lecture Notes in Computer Science*, pages 184–199. Springer Verlag, 2007.

[PCC06]   Andrew Phillips, Luca Cardelli, and Giuseppe Castagna. A graphical representation for biological processes in the stochastic pi-calculus. *Transactions on Computational Systems Biology*, 7:123–152, 2006.

[Phi]   Andrew Phillips. Some 3D videos of SPiM simulations. `http://research.microsoft.com/en-us/projects/spim/default.aspx`. retrieved 05-OCT-2009.

[Pri95]   Corrado Priami. Stochastic π-Calculus. *Computer Journal*, 6:578–589, 1995.

[PRSS01]   Corrado Priami, Aviv Regev, Ehud Y. Shapiro, and William Silverman. Application of a Stochastic Name-Passing Calculus to Representation and Simulation of Molecular Processes. *Information Processing Letters*, 80(1):25–31, 2001.

[RC94]   Ramana Rao and Stuart K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *ACM SIGCHI'94: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 111–117, 1994.

[Reg03]   Aviv Regev. *Computational Systems Biology: A Calculus for Biomolecular Knowledge*. PhD thesis, Tel Aviv University, 2003.

[RPS⁺04] Aviv Regev, Ekaterina M. Panina, William Silverman, Luca Cardelli, and Ehud Shapiro. BioAmbients: An Abstraction for Biological Compartments. *Theoretical Computer Science*, 325(1):141–167, 2004.

[SJUS08]   Hans-Jörg Schulz, Mathias John, Andrea Unger, and Heidrun Schumann. Visual analysis of bipartite biological networks. In *VCBM'08: Proceedings of the Eurographics Workshop on Visual Computing for Biomedicine*, pages 135–142, 2008.

[TAS09]   Christian Tominski, James Abello, and Heidrun Schumann. CGV – an interactive graph visualization system. *Computers and Graphics*, 33(6):660–678, December 2009.

[TK08]   Oksana Tymchyshyn and Marta Z. Kwiatkowska. Combining Intra- and Inter-cellular Dynamics to Investigate Intestinal Homeostasis. In *Formal Methods in Systems Biology, First International Workshop, FMSB 2008*, pages 63–76, 2008.

[VB08]   Cristian Versari and Nadia Busi. Efficient stochastic simulation of biological systems with multiple variable volumes. *Electronic Notes in Theoretical Computer Science*, 194(3):165–180, 2008.

[Ver07]   Cristian Versari. A core calculus for a comparative analysis of bio-inspired calculi. In *European Symposium on Programming (ESOP'07)*, pages 411–425, 2007.