Visual Support
for the Modeling and Simulation of Cell Biological Processes

Dissertation

zur

Erlangung des akademischen Grades eines

Doktor-Ingenier (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik

der Universität Rostock

vorgelegt von

Dipl.-Ing. Andrea Unger

geb. am 10. Juli 1981 in Blankenburg/Harz

aus Rostock

Rostock, 6. April 2010

urn:nbn:de:gbv:28-diss2010-0137-8

Gutachter

Prof. Dr.-Ing. Heidrun Schumann, Universität Rostock, Deutschland (Betreuer)

Prof. Dr.-Ing. Bernhard Preim, Otto-von-Guericke-Universität Magdeburg, Deutschland

Prof. James J. Thomas, Pacific Northwest National Laboratory, Richland, Washington, USA

Tag der mündlichen Prüfung

14. Mai 2010

Abstract

The full potential of information visualization is often not tapped in the scientific reasoning process. To overcome this gap, this work aims at bringing visualization closer to demands of the analysis process for the specific domain of modeling and simulating cell biological systems. To this end, the need for visual support is systematically explored along the work flow in the application domain. Based on the identification of main segments of visual support that need to be provided, this work provides novel contributions for the visualization of stochastic simulation data. Two main visualization challenges are addressed. First challenge is the visual integration of data into the data generating context, which is assumed to facilitate the analysis process of complex data. A general approach for stochastic simulation data is introduced to capture the process of data generation. It comprises four process levels: model, experiment, multi-run simulation data, and single-run simulation data. Tailored visualization concepts are developed for all levels. Thus, the complete simulation process can be explored rather than single simulation data sets. Second challenge is the visualization of large and complex data sets. Within the application domain, simulation data incorporates heterogeneous data types and is given in temporal, spatial, and multivariate context. Such complexity is a challenging issue in current visualization research. New visualizations are developed based on a systematic evaluation of possible visual mappings. Multiple views, each revealing different aspects in the data, are combined and closely linked to support the exploration of the data in all its facets. The novel visualization concepts developed in this work have been made available in a visualization component library, which is integrated into the application domain.

Zusammenfassung

Das Potential von Informationsvisualisierung wird bei der Untersuchung wissenschaftlicher Fragestellungen häufig nicht ausgeschöpft. Diese Arbeit verfolgt das Ziel, diese Lücke zu verkleinern, indem die Anforderungen des Analyseprozesses stärker in der Visualisierung berücksichtigt werden. Da der Analyseprozess stark anwendungsabhängig ist, richtet diese Arbeit den Blick auf die Modellierung und Simulation zellbiologischer Systeme. In einem ersten Schritt wird der Bedarf für visuelle Unterstützung systematisch entlang des Arbeitsablaufs im Anwendungsfeld untersucht. Dadurch werden wesentliche Teilbereiche der visuellen Unterstützung identifiziert. Auf dieser Grundlage werden im Rahmen der Arbeit neue Beiträge für die Visualisierung von stochastischen Simulationsdaten entwickelt. Zwei wesentliche Visualisierungsherausforderungen stehen im Mittelpunkt. Zuerst ist die visuelle Integration der Daten in den datengenerierenden Kontext zu nennen, um auf diese Weise die Analyse von Simulationsdaten zu unterstützen. Eine neue Systematik erfasst den Prozess der Datengenerierung für stochastische Simulationsdaten mit vier Prozessebenen: Model, Experiment, Multi-Run-Simulationsdaten und Single-Run-Simulationsdaten. Davon ausgehend werden maßgeschneiderte Visualisierungskonzepte für alle Ebenen vorgestellt. Die dadurch ermöglichte Explorierung des gesamten Simulationsprozesses geht weit über die Untersuchung einzelner Simulationsdatensätze hinaus. Die zweite Herausforderung ist die Visualisierung großer und komplexer Datenmengen. Simulationsdaten im Anwendungsbereich sind heterogen, da sie Zustände und Ereignisse umfassen, und sind im zeitlichen, räumlichen und multivariaten Bezug gegeben. Die Arbeit begegnet dieser Herausforderung mit einer systematische Analyse möglicher Visualisierungen. Die daraus neuentwickelten Konzepte kombinieren verschiedene Sichten auf die Daten. Durch ihre enge Kopplung wird die Exploration aller Datenaspekte ermöglicht. Alle vorgestellten Visualisierungskonzepte sind in einer Visualisierungskomponentenbibliothek praktisch umgesetzt, die in die Anwendung eingebunden ist.

Danksagung

Ich möchte allen danken, die mich bei der Erstellung dieser Arbeit unterstützt haben. Mein Dank gilt insbesondere Heidi Schumann, die mir die Promotion an ihrem Lehrstuhl ermöglicht hat. Diese Arbeit verdanke ich wesentlich den vielen konstruktiven Gesprächen mit ihr, ihrem Vertrauen und ihrer stetigen, engagierten Begleitung in den vergangenen drei Jahren. Ich fühle mich sehr geehrt, Jim Thomas und Bernhard Preim als Gutachter gewonnen zu haben. Ich danke ihnen, dass sie die Begutachtung dieser Arbeit kurzfristig übernommen haben. Helmut Doleisch hat mir die Möglichkeit geboten, während eines zweimonatigen Forschungsaufenthalts am VRVis Wien wertvolle persönliche Kontakte zu knüpfen und neue Erfahrungen zu sammeln. Ihm und Philiipp Muigg danke ich für die freundschaftliche Zusammenarbeit. Gleiches gilt für die Kollegen an den Lehrstühlen für Computergraphik und Visual Computing, die mich so herzlich in Rostock aufgenommen haben.

Diese Arbeit hat wesentlich von der Kooperation mit anderen Anwendungsfeldern profitiert. In diesem Sinn danke ich den Kollegen im Graduiertenkolleg und vom Lehrstuhl für Modellierung und Simulation – besonders meinen lieben Bürokollegen, Hans-Jörg, Mathias und Matthias, Dagmar, Orianne, Géraldine, Ron und Yvonne. Sie haben mir in unzähligen Gesprächen wesentliche Anregungen gegeben, viele Konzepte dieser Arbeit sind in enger Zusammenarbeit erarbeitet und umgesetzt worden. Ich möchte mich auch bei den Studenten bedanken, mit denen ich zusammengearbeitet habe. Enrico Gutzeit, Ulrike Krüger und Thomas Gertz haben im Rahmen ihrer studentischen Arbeiten, Steffen Hadlak, Paul Bönisch und Stefan Friese mit ihren Implementierungsarbeiten zu dieser Dissertation beigetragen.

Ich möchte ganz herzlich meiner Familie und meinen Freunden danken, für ihr Verständnis und ihre Unterstützung. Tom, ohne dich hätte ich diese Arbeit nicht schreiben können.

Diese Arbeit wurde finanziell gefördert durch den Graduiertenkolleg **dIEM oSiRiS** der Deutschen Forschungsgemeinschaft.

Contents

List of Figures

List of Tables

List of Tables

Chapter 1

Introduction

Solving scientific problems involves the analysis of data. With ever increasing capabilities to generate and store data, methods used to gain insight into data have to advance accordingly. Visualization has been recognized as one important technology in this regard. The presentation of data in a visual form fits in very well with the human ability to perceive and comprehend visual information. In fact, by means of visualization, the strengths of computational processing and human cognition are brought together. Computers are able to store and process data as well as to generate visual representations in a very efficient manner. The human cognition is needed in the "sense making" of data. Moreover, by interactive features, it is the human's role to steer data processing and the adaptation of the visual representation to current needs.

Since visualization has been established as a research field of its own, its great potentials have been recognized and explored. This led to the development of many powerful techniques. However, still a gap exists between the potentials of these techniques and their application in the domains where data needs to be analyzed. Visualization is usually applied as a means to present the results of data analysis, rather than as a means to perform this analysis. To overcome these limitations in practical use, visualization in its role as an "enabling technology" has to be brought closer to the demands of the data analysis process and to the demands of users.

But the integration of visualization as a natural, reliable technology in the data analysis process cannot be carried out universally, as the demands are domain specific. This work investigates the integration of visualization within the specific domain of modeling and simulating cell biological systems. During the last years, the field saw a tremendous increase of interest. While the importance of cell biological systems in understanding the basics of life has long been recognized, new technologies to investigate cellular processes pushed the research activ-

ities in this area to a new level during the last 10 years. These developments comprise new experimental methods to observe biological systems in higher detail as well as the increasing capabilities to store and share data among researchers around the world. Moreover, the field of modeling and simulation has been established as an important additional technology. New approaches to model and simulate these systems in high detail, including stochastic effects and spatial context, are more and more utilized to investigate cellular processes.

All these developments give the opportunity to gain a better understanding of the multi faceted, complex processes in cells. But they also require new methods of data analysis, where visualization can provide a valuable contribution. The integration of visualization within the application field requires a close cooperation between developers of visualization and domain experts. For this work, this close cooperation is provided within the research training school **dIEM oSiRiS**. The project brings together researchers from biology and medicine with researchers from fields of computer science like data bases, visualization, and modeling and simulation. The conjoint research aims at the development of new computational methods, including modeling formalisms and simulation algorithms as well as the storage, analysis, and visualization of generated data. As a practical application within the research training school, the role of the Wnt protein family in the development of neurons from neural stem cells is investigated.

1.1 Challenges and Goals

Integrating visualization as a means of data analysis within an application requires to broaden the scope of visualization from handling data sets with specific characteristics to the process of problem solving. Although this process is application specific, the main challenges in this regard are universal.

In general, an analysis process is a sequence of individual steps. Throughout the process, a multitude of data sets with various characteristics needs to be evaluated under different objectives. Handling such heterogeneous data is still an open challenge for visualization research.

Further, providing comprehensive visual support goes beyond the visualization of data sets. Many of the existing visualization techniques are uncoupled from the process in which the data was derived, to provide a larger flexibility in their use within different application fields. Information visualization techniques perform an interactive post-processing of the data, a back-coupling with the process of data generation is usually not provided. But complementing the data visualization with information about the data generating process can significantly help

the user in understanding the data. Otherwise, analysts have to bring together their intrinsic knowledge about the process and the often complex visual information shown on the screen. By the integration of information about the generating process within the visualization, the analysts' efforts for gathering all the necessary information for decision making are reduced. Also, data analysts, who have not been involved in the process of data generation, get an easier access to the data. So far, the data generating process has been accounted in the design of very few visualization techniques and, thus, remains a challenge.

To integrate visualization within the process of data generation and data analysis, the specific application domain has to be taken into account. In this regard, the goal of this work is to investigate how comprehensive visual support can be provided for the modeling and simulation of cell biological systems. As an initial scientific question in this regard, it has to be answered what form of visual support is needed. Which data has to be analyzed throughout the process and what is the purpose of analyzing this data? This work aims at such a systematic approach by identifying the challenges for visualization on the basis of the work flow within the application domain.

Based on these observations, the focus of this work can be further refined to give attention to new challenges. In this regard, the focus of this work will be set on the development of new visualization techniques for the analysis of simulation data. Simulation data has to be analyzed at many stages of the work flow, including modeling, model validation, and the generation of new hypotheses from simulation results. The advent of discrete-event based approaches that consider stochastic effects and spatial context require new visualization methods that can handle such complex data. Moreover, the simulation data generated in a modeling and simulation project goes by far beyond the result of a single execution of the simulation. Usually, multiple experiments are executed, whose simulation output has to be compared with each other and with the experimental data from the real biological system. Further, analyzing simulation data from stochastic simulation requires the consideration of multiple executions of the same experiment, which produce different results due to these stochastic effects.

In this work, new visualization techniques are developed to support these different objectives of analysis, following the main idea to integrate data visualization into the data generating context in order to avoid an information loss in the visualization.

1.2 Results

Following the aim to provide visual support for the modeling and simulation of cell biological systems, this work presents genuine visualization approaches with respect to the visualization challenges identified in Section 1.1.
In this work, as a necessary first step for the **integration of visualization within the application domain**, first approaches are presented that aim specifically at the modeling and simulation of cell biological systems.

- To initiate a systematic research for open visualization challenges in the application domain, the visualization is conceptually integrated into the work flow of modeling and simulation, focusing specifically on cell biological systems. Along the work flow, the basic stages of the work flow along with their main data sources, characteristics of generated data, and analysis goals are pinpointed. Moreover, the interplay of the stages and, consequently, the produced data sets throughout the work flow become apparent. Based on this systematic approach, the integration of visualization at all stages of the data generating process in modeling and simulation is proposed, which has been published in [UBJ+07].

A second challenge is to handle **complex and heterogeneous data**. In this regard, the work proposes the following new visualization approaches.

- Visual support for complex simulation data from the Next Sub-Volume Method. A current research topic in modeling and simulation is to simulate cellular processes with detailed spatial context. The Next Sub-Volume Method [EE04, RKDB06] is one exemplary simulation algorithm. The visualization of the simulation data is demanding, as it is heterogeneous, comprising states and events, and has to be explored in its temporal, spatial, and multivariate context. From the systematic development of tailored views supporting the analysis of different facets of the data, a highly interactive multiple view framework is developed to explore the data in its entirety. Within the framework, the complexity of the spatio-temporal context is broken down into separate views. An overview on temporal developments is provided by the visualization of high level features over time. Identified time points of interest are shown with spatial details in additional views. The framework is unique in supporting the simultaneous visualization of events and states in their multi-dimensional context. Specifically the concepts to handle the heterogeneity and the spatial context of the data have been published in [UGJS09].

- Visual support for complex simulation data from the Attributed Π-Calculus. In alternative to an explicit integration of spatial context, the Attributed Π-Calculus [JLNU08, JLNUar] models spatial effects in an abstract way by the communication between processes. Visualizing the resulting simulation data – a time series of reaction networks – is an open challenge. Also here, the complexity of the data is broken into manageable chunks, by separating the visualization over time by high level features and the detailed visualization of single time points into different views, which are closely linked. Tailored to reflect the complex data with dynamically changing structural relationships, the high level features consists of graph complexity measures and attribute values. As part of joint research, this work specifically contributes to the visualization over time. These time points can then be analyzed in detail using a new table-based, highly interactive visualization technique for graphs, which is scalable and visualizes structural relationships as well as attributes of both nodes and edges. The graph visualization technique was published in [SJUS08]. The visualization concepts for the Attributed Π-Calculus have been submitted for publication [JSS+10].

- Visual support for statistical analysis of subsets from simulation data. An important analysis method for simulation data is the evaluation of statistical properties like mean, standard deviation, or extrema. One typical situation where data analysts make use of statistical values is the analysis of multi-run data. But the evaluation of global values is not sufficient. In large data sets, local variations can occur, which makes it useful to analyze and compare properties of subsets. Addressing this problem, a new visualization has been developed to analyze statistical properties of subsets in temporal context, which is an important aspect for simulation data. The visualization is integrated within the visualization tool SimVis [Dol04]. Subsets are generated interactively by the user in other coordinated views, by brushing of subsets with smooth transition from focus to context. Resulting degrees of interest associated with data items of the subset are addressed in both the computation and visualization of statistical properties. The new view provides analysis of statistical properties of subsets and their comparison with each other as well as with global values. In addition with interactive means to refine subsets on the fly in other views, the resulting visualization supports a user-driven evaluation of local characteristics of large and complex data sets, which has been published in [UMDS08].

In addition to handling specific data sets, the third challenge refers to the visualization of data in the **context of the data generating process**. The following visualization concepts

provide a new contribution within the field of visualization in this regard.

- Visual analysis of simulation data in the context of the data generating process. Simulation data is always derived in the context of an underlying model and an experiment. To capture this context of data generation, a general taxonomy is introduced for stochastic simulation data. It comprises the levels model, experiment, multi-run simulation data, and single-run simulation data, which are all linked to specific visualization goals. Including this context in the visualization can significantly help the user to gain insight into the simulation process. In a new visualization, the visual integration of model, experiment description, and resulting multi-run simulation data is provided. This view is the basis for the exploration of simulation data at the other levels of the data generating process, which enables the comparison of experiments, the comparison of multiple runs from one experiment, and the identification of single runs. Initially shown for the example of the stochastic, discrete-event based Gillespie algorithm, the approach can be generalized to analyze simulation data from modeling and simulation approaches based on chemical reaction networks. With the visual support at multiple levels of the simulation process, the visualization concepts enable the exploration of the complete simulation process. The results have been published in [US09].

Although the goal of this work is to provide new visualization methods to support the visual analysis within the application domain, the **presentation** of results remains an ongoing research topic in visualization. Presentation requires alternative approaches, which focus on the communication of findings and stated facts. This work proposes a new technique for presentation based on interactive images.

- Interactive images for presentation and communication using Illustration Watermarking. By providing descriptive information on demand, interactive images are a compact format for communication and presentation, which is easy to create, distribute, and access. As an appropriate technique to store the descriptive information and link it to the image content, techniques from information hiding are explored, which are called Illustration Watermarks in this context. In a user study, a new adaptive approach optimized for the application of interactive images outperformed a traditional proceeding in terms of quality. In addition, interaction procedures are designed to support the user in the exploration of descriptive information. The results of this work were published in [SUS08].

1.3 Structure

This work is structured as follows. In Chapter 2, the need for visual support within the modeling and simulation of cell biological systems is investigated. Therefore, the particular stages of the work flow are analyzed with respect to the involved data analysis. This leads to four main segments of visual support. In these segments, existing visualization techniques are reviewed that have been applied and specifically developed for the field of application. Based on this analysis, a main focus of this work is set on the development of visualization methods for particularly supporting the analysis of simulation data. Novel techniques in this regard are introduced in Chapter 3, which focuses on the visual integration of data into the data generating process to support the exploration of the simulation process, and in Chapter 4, which addresses the visualization of complex simulation data. In Chapter 5, a visualization component library is presented that realizes the integration of visualization into the data generating context of the application domain. The library comprises visualization tools that implement the proposed visualization concepts as well as tools that have been developed in the research training school. Chapter 6 concludes this work by summarizing the main contributions and a discussion on future research topics.

Chapter 2

Problem Analysis and Conceptual Approach

One goal of this work is to investigate how the capabilities of visual analysis can be made available in the application domain of modeling and simulating cell biological systems. To this end, a systematic approach is applied to identify gaps in existing visual support – being the starting point for the development of appropriate visualization concepts to complement the visualization toolbox in the application domain. The goals of this chapter are, first, to get a systematic view on the application domain in order to identify the need for visual support. In consequence of this analysis and a review of related work, open visualization challenges are identified. The focusing on the development of new visualization concepts for simulation data is motivated.

As a second goal, a conceptual approach to handle simulation data is introduced, which is the basis to systematically design appropriate visualization methods in this field. According to these goals, this chapter comprises three sections. In Section 2.1, the work flow of modeling and simulation is used as the basis to determine the need for visual support and a review of existing visual support in the application domain. As a result of this analysis, challenges and requirements for visualization are discussed in Section 2.2, which further narrows the focus of this work to one segment of visual support: the visualization of simulation data from discrete event based approaches. Section 2.3 introduces a general approach to cope with the challenges that arise for the visualization of stochastic simulation data. The results of this chapter are summarized in Section 2.4.

2.1 Problem Analysis

As a starting point to provide visual support for the modeling and simulation of cell biological systems, this section aims at identifying the potentials of visualization within the application domain. Basis is the process of modeling and simulation. Along this process, generated data and the purpose of data analysis can be identified. Thereby, areas of visual support become apparent. This proceeding offers the possibility to systematically reveal open visualization problems in the domain.

The following approach is pursued in the remainder of this section. In Section 2.1.1, the work flow in modeling and simulation is introduced. For visualization, it is of specific interest what data is generated and analyzed along the work flow. In this regard, four segments of visual support in the application domain are identified, which are integrated as a natural complement of the modeling and simulation process (Section 2.1.2). The existing visual methods are evaluated for each segment in Section 2.1.3.

2.1.1 The Work Flow of Modeling and Simulation

In the literature, a number of modeling and simulation work flows has been proposed [Sar00, Bal98], also specifically for biological applications [KHK+05, OD07]. Although slightly different in the sequence of steps, they generally comprise five main steps as they are listed in [Vel09]:

- Problem Definition
- Systems Analysis
- Modeling
- Simulation
- Validation

These steps are sequential in the sense that the results of one step are needed before a successive step is carried out. However, the work flow of modeling and simulation is highly iterative. Results of each step can lead back to any of the preceding steps. The model – in the focus of the modeling and simulation process – is usually not built by a single iteration of the work flow. Instead, the process leads from a first, simple model to more complex versions until the model fits the needs of the project. Specifically for systems biology, this loop like structure has been emphasized by [KHK+05]. In particular, findings from the modeling or

simulation phase may lead back to a refined systems analysis. With additional experiments, new hypotheses about the system are tested or available data is supplemented, initiating new iterations of the overall process. Hence, the iterative refinement of the model by going back and forth in the sequence of steps is an immanent feature of the modeling and simulation process.

In the following, a closer look is taken at the individual steps of the modeling and simulation process.

Problem Statement confines the problem to solve or the questions to answer with the modeling and simulation project. There are two starting points for any modeling and simulation project [Vel09]: First, the source system: a part of the real world, which is investigated. Second, a problem to be solved or questions to be answered about the system. This implicates that the goal of the modeling and simulation process is not to reproduce the real world system in all its complexity. Instead, the source system is investigated with specific objectives. As stated in [ZPK00, p.27], the "statement of objectives serves to focus model construction on particular issues." The problem statement determines the choice of methods and the acquisition of data throughout the work flow.

System Analysis refers to gathering and evaluating information about the source system that contributes to the stated problem. This information comprises all data that helps to gain a further understanding of the system under study. It includes knowledge accessible from the literature or data bases as well as own experiments conducted on the source system. The data can be subdivided into qualitative and quantitative parts. Qualitative data refers to information that can contribute to identify the relevant parts of the system and their relations. This information is necessary to build up a structural understanding of the process. Quantitative data, on the other hand, has an important function to describe the dynamics of the system.

Modeling comprises the development of a model based on results from systems analysis. The model includes the system components and their relations, which are considered to be relevant with respect to the problem definition. In order to gain a model that can be simulated computationally, it must be expressed in a formal language and fully quantified. The modeling stage follows two complementing objectives: the identification of qualitative, structural aspects of the model on one hand and its quantification on the other hand.

To derive the structure of the model, qualitative information from the systems analysis step is used to collect the basic components of the system and their relations. An informal model (sometimes also called conceptual model) is built. Based on requirements comprised by this

informal model, an appropriate modeling formalism has to be chosen that can reflect these requirements. Then, the informal model is transformed into a formal model.

The second objective, the quantification of the model, begins after the formal structure of the model is determined, because then the parameters of the model are known. These parameters are derived from quantitative data gained in systems analysis.

Simulation is the application of the model to the given problem or question [Vel09]. In computational modeling and simulation, simulation generally refers to the execution of the model under a certain experimental setup. The purpose of simulation is to observe the behavior of the model over time. If the model has the ability to mimic the source system's behavior with sufficient accuracy, it can be used to execute experiments, which are difficult, costly, or even impossible to carry out on the source system. In the best case, findings from simulation data lead to new hypotheses about the source system, which can be tested by new experiments.

Validation According to Balci [Bal98], "model validation is substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the M&S objectives." These modeling and simulation objectives are defined in the problem statement. Although listed as an individual step of the modeling and simulation work flow to highlight its importance, validation should be performed as a "continuous activity throughout the entire life cycle" [Bal98]. Targets of validation are both the structure and the behavior of the model. Many methods exist for validation, see [Bal98, Sar00]. They can be subdivided into objective methods, where statistical tests and procedures are used, and subjective methods, where experts evaluate the quality of the model or of the model behavior based on their knowledge. Subjective methods therefore highly involve visual inspections. Both objective and subjective tests are often based on a comparison of model behavior and source system behavior. Also, it may be reasonable to compare model structure and behavior to other models, especially if the source system is difficult to observe. Hence, data used for validation is gathered in systems analysis, modeling, and simulation.

2.1.2 Integration of Visualization into Work Flow

Given the basic steps of the work flow in the application domain, the ground is provided to integrate visualization as a complementary part of the modeling and simulation process. This will be subject of this section. Generally, all steps of the work flow – specifically systems analysis, modeling, simulation, and validation – involve the analysis of a multitude of data sets

and can therefore significantly benefit from methods of visualization. But the scope of this work comprises two main visualization challenges: The visual integration of data into the data generating context on the one hand and the visualization of heterogeneous, complex data sets on the other hand.

In this regard, the focus is set on those steps of the work flow that involve the generation of substantial amounts of data: systems analysis, modeling, and simulation. Supporting the specific analysis tasks of validation with methods of visual analysis remains an open research problem and will be addressed in the next phase of the research training school **dIEM oSiRiS**.

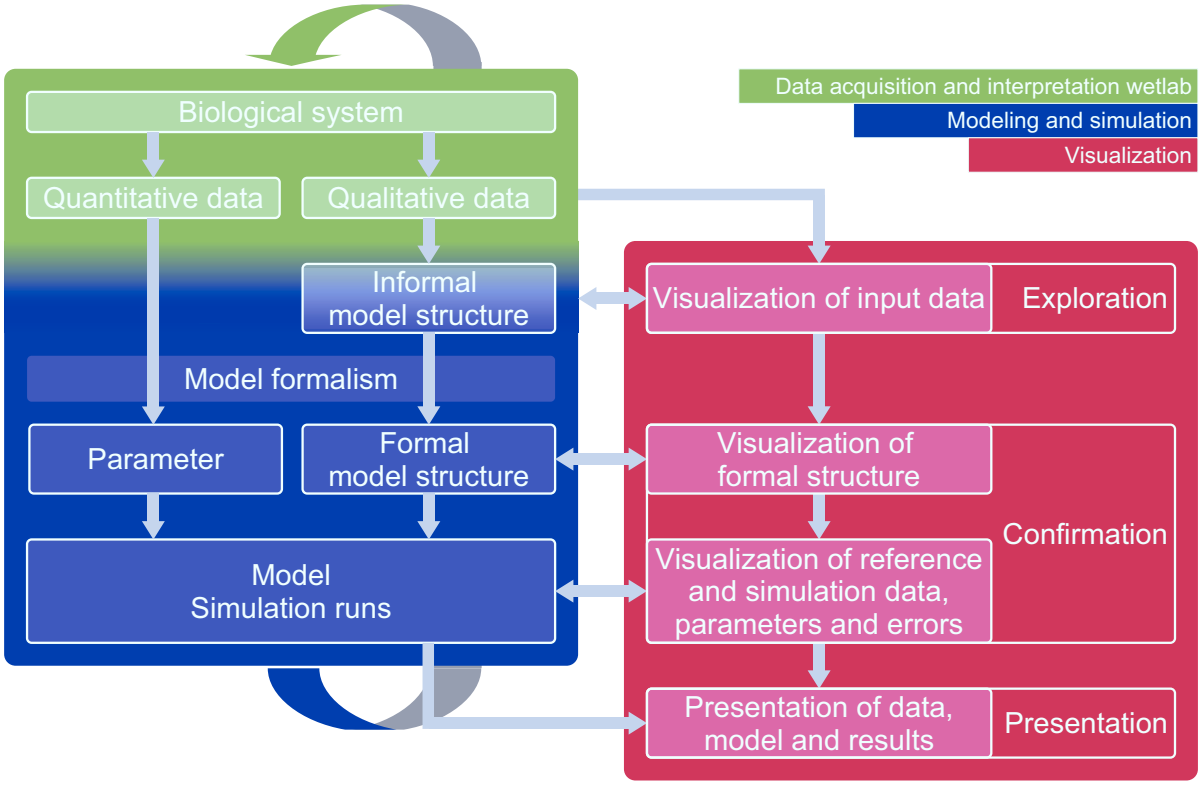


Figure 2.1: Schematic presentation of visual support within the modeling and simulation of cell biological systems. Based on the data that is generated along the process, shown on the left, four main segments of visual support are identified, as shown on the right.

In order to provide visual support for systems analysis, modeling, and simulation, a systematic approach is needed. With respect to the challenge to make the data generating process explicit in data visualization, the process of data generation along the work flow has been identified in joint work within the research training school. Along this process, important aspects of

visual support are conceptually integrated. The resulting integration of data visualization into the application domain and visualization, which has been published in [UBJ+07], is presented in a schematic overview in Figure 2.1. The box on the left side comprises the data that is produced along the work flow. The arrow from the end of the work flow back to its beginning indicates the loop-like structure. In Section 2.1.2.1, the flow of data is shortly summarized with respect to the steps of the work flow. In the box on the right side, the four main segments of visualization are depicted. The interplay of the analysis process with the visualization is indicated by arrows between both boxes. The four segments of visual support are inspected in Section 2.1.2.2.

2.1.2.1 Data Generation during Modeling and Simulation Work Flow

The left side of the scheme in Figure 2.1 provides an overview on data that is generated during the modeling and simulation work flow. **Systems Analysis** is the starting point of data generation. From observations of the *Biological System* – either experiments or information gathered from literature – *Qualitative Data* and *Quantitative Data* is collected. Going into the **Modeling** phase, qualitative data serves to identify structural relations of important system components, which contribute to an *Informal Model*. This informal model is transformed into a *Formal Model* by expressing it in a formal language, which has been chosen based on model requirements. Complemented by the *Parametrization* of the model, derived from quantitative data, the formal and quantified *Model* provides the basis for the phase of **Simulation**. Here, *Simulation Runs* are carried out to observe the model's behavior.

2.1.2.2 Visual Support throughout Work Flow

Complementing the flow of data through a modeling and simulation project, the right side in Figure 2.1 shows the needed visual support during the work flow. This comprises four main segments:

- **Visualization of Input Data**
  Input data comprises all data that is gathered during systems analysis. The term "input" is used as data from systems analysis provides the basis for the modeling and simulation project. Input data can originate from literature, data bases, or own experiments in the wet lab. Hence, input data can have many different forms. Specifically for biological systems, the availability of many different experimental methods, for example, microscopy, micro arrays, or blotting techniques, lead to data sets with very different characteristics.

As quantitative data is usually derived from the analysis of imaging data, resulting data often comes with uncertainty. Likewise, data taken from the literature or data bases, which has also been derived from experiments, has to be evaluated with respect to reliability and uncertainty. All these aspects have to be considered to provide appropriate visual support for the analysis of input data.

- **Visualization of Formal Structure**
  A model comprises important components of the system and their interrelations. It can be seen as a graph, with components as nodes and interrelations as edges. Hence, graph visualization is the method of choice to represent models visually. Depending on characteristics, complexity, and size of the model, different graph visualization techniques are appropriate. This also demands to reflect the specificity of modeling formalisms in the visualization. For larger models, the visualization becomes more challenging, but also more and more important as comprehending dependencies within a model require methods of visual analysis. With the advent of new modeling formalisms, new challenges arise for the model visualization. To reproduce the complexity of biological systems, new methods to build dynamic, multi level models consisting of multiple heterogeneous sub systems imply new visualization methods that can handle such complex structures.

- **Visualization of Reference and Simulation Data, Parameters and Errors**
  Simulation data results from the execution of a model under a certain parameterization. Specific data characteristics depend on the modeling formalism and the simulation algorithm. Given the diversity of existing approaches, a broad variety of data sets appears in this segment of visual support. During a modeling and simulation project, simulation is often executed to assess how well the model mimics the behavior of the source system. Consequently, multiple executions of the evolving model are performed under various parameterizations. Hence, the evaluation of many data sets derived under different conditions has to be supported as well as the comparison of simulation data with reference data from the source system. In addition to the visualization of the usually time dependent, multivariate data itself, the availability of additional information, such as the current model parameterization or errors, plays an important role for data evaluation. While these aspects form a set of open challenges in visualization, new modeling and simulation approaches that incorporate both spatial and stochastic effects further increase complexity and size of simulation data and lead to additional challenges such as visual multi-run analysis.

- **Presentation of Data, Model, and Results**
  Presentation of results and findings from a modeling and simulation project requires visual methods differing from those needed in previous steps of visual analysis. Instead of data analysis leading to new findings, the meaning of the data is now clear. Rather, it needs to be transformed into a representation that communicates results to third party users in an efficient manner. Data used for presentation can stem from all phases of the modeling and simulation project – systems analysis, modeling, simulation – or even be specifically generated for presentation purposes. Hence, while the purpose of visualization can be narrowed to the goal of presentation, the heterogeneity of possibly used data leads again to a broad diversity of necessary visualization methods.

The described visual support for modeling and simulation comprises all three basic goals of visualization: exploration, confirmation, and presentation [SM00]. The first three segments of visual support – visualization of input data, formal structure, and simulation data – are related to visual analysis, aiming both at exploration and confirmation. Exploration refers to an undirected search in data for inner structures and information, in order to generate new hypotheses. As visualization of input data is mainly performed to collect new information about the system under consideration, it can be mainly linked to the goal of exploration. Confirmation, on the other hand, aims at confirming or rejecting a given hypothesis by appropriate data visualization. Models, and also simulation data reflecting a model's behavior, are usually built from certain hypotheses. Hence, visual analysis of models and simulation data can be often linked to the goal of confirmation. However, this separation does not hold completely: confirmation is also performed by a visual analysis of input data and, likewise, modeling and simulation data is explored to gain new insights. The goal of presentation is reflected by the last of the described four segments of visual support. The focus of presentation is to communicate the results gathered by the previous steps of visual analysis to people who have not been involved in the process.

2.1.3 Related Work

The goal of this work is to complement existing visualization methods for the modeling and simulation of cell biological processes. In this regard, this section presents the range of currently available visualization methods in the application domain. The related work is presented with respect to the four segments of visual support that have been stated in the previous Section 2.1.2.2. The focus is set on visualization techniques that have been specifically developed for

the application domain as well as commonly used tools and visual representations.

Beyond visualization techniques reviewed in this section, visualization methods for biological or medical applications exist that investigate biological processes at different scales. Exemplary at a lower scale, molecular visualization aims at visualization of 3-D molecular structures of components of the cell such as genes and proteins [BRRB93, KMR+09, MPC+06, GSS01]. Abstracting from 3-D context, visualization for DNA sequencing has, for example, been addressed in [PFvdW+04]. On the other direction of scale, the broad field of medical visualization [PB07] aims at the visual analysis of tissues, organs, and organ systems, often derived from imaging data gained by CT, MRT and similar techniques. One recent example for volume visualization of the brain for research in neurobiology is presented in [BSG+09].

While this section focuses on the application domain, related work from the field of information visualization is presented in subsequent Chapters 3 and 4, which address the development of new visualization techniques. The presentation of related work in this section serves as the starting point for a subsequent discussion in Section 2.2. The discussion leads to the identification of open challenges for visual support, which need to be addressed by the development of new visualization techniques.

2.1.3.1 Visualization of Input Data

Two main sources for the collection of information about cellular processes exist: The evaluation of available knowledge from prior work, recorded in the literature and data bases, and the analysis of experimental data. The review of related work in systems analysis is sectioned accordingly. Knowledge from prior work is often captured in pathways or biological networks, which are described at first. Then related work to visualize experimental data is presented. At last, the integration of data from both types of sources is addressed, which has been subject of recent research.

Pathways and Biological Networks Understanding cellular processes relies on known information about functional relationships among cell components. Prominent representations to capture known functional relations among chemical compounds are pathways and reaction networks. In the following, the distinction of pathways and reaction networks is made based on their size, as it is usually found in the literature. Pathways contain biochemical reactions, which are necessary for a certain function of the cell and thus, can be seen as a functional unit. Usually, they comprise a few dozen proteins and reactions. Reactions networks contain a larger number of proteins and reactions, ranging from a couple of hundred to thousands. They

represent larger parts of the functionality of the cell.

Visual representations of pathways are commonly used in the application domain. A number of data bases exist that come with visual representations of pathways. They include the KEGG data base [KAG+08], BioCarta [1], GeneAssist Pathway Atlas [2], or WikiPathways [PKvI+08], which provide pathway information in manually created, static depictions. An example from the BioCarta website is shown in Figure 2.2(a). Algorithmically defined layouts of pathways are usually based on the force directed layout [BWCT09, LWSK04]. In [Sch03], the comparison of similar pathways is supported by arranging corresponding parts on the same horizontal layer.

Equipped with interactive functionality to explore relations among pathways, a multi-plane approach has been presented in [SKKS08]. Pathways are shown on 2-D planes in a 3-D space, with links between the planes to show relations. With highly interactive multiple linked views providing overview and detail, pathways can be explored for relevant interdependencies (Figure 2.2(b)).

For reaction networks with hundreds or thousands of proteins and reactions, several approaches have been proposed to reduce the size of the graph to a few dozen proteins and reactions, before this subset is visualized. An example is found in [HMT09]. By querying a specific biochemical entity from a data base containing a complete reaction network, a graph containing up to 50 nodes is extracted from the database and visualized. Interactively, the user can adjust the number of nodes concurrently shown by adapting number of nodes or the range of shown edge weights.

Numerous tools have been developed that support automatic node link layouts of the complete reaction network. However, node link layouts usually appear cluttered for thousands of nodes. To cope with clutter, visual manipulations such as zooming and basic analysis functions such as detection of sub-graphs and filtering are provided. Widely used examples are Cytoscape [SMO+03], Osprey [BST03], ProViz [INM+05], or Pajek [BM04], a general purpose network visualization tool. A screen shot from Cytoscape is shown in Figure 2.2(c). In [DMS+08], an overview and detail approach is presented. From a graph containing up to 1000 nodes, a focal node is chosen, whose surrounding is shown as a sub-graph in detail (up to 50 nodes). A constrained graph layout is used for interactive visualization, aiming at fast rendering of overview and higher quality detail visualization, while topology among both views is preserved. Aiming at the specific task of exploring the network for motifs, an interactive visualization technique in combination with algorithmic search for motifs has been presented in [SS05], which is shown

[1] http://www.biocarta.com/Default.aspx

[2] http://www4.appliedbiosystems.com/tools/pathway/all_pathway_list.php

in Figure 2.2(d). For the comparison of reaction networks, a multi-plane approach is used in [FHK+09], with the drawback that the 2-D network representations appear cluttered. While all these approaches use 2-D layouts of graphs, also 3-D layouts have been proposed for biological networks [Roj04, YWCND06].

The size of reaction networks in the application domain can go beyond the described thousands proteins and reactions, ranging to tens or hundreds of thousands. However, visualizations proposed explicitly for these large networks in the application domain have not been found in the literature. Here, new techniques in information visualization, where this is a recent research problem, need to be adapted to the requirements of the application domain.

Experimental Data A wide range of experimental methodologies exists that is suitable to investigate cellular processes. For the majority of methods, experiments on cell biological systems result in raw imaging data, which needs to be quantified in order to gain meaningful numbers. The analysis sequence involves steps like image processing, filtering, and statistical analysis before relevant conclusions can be drawn from the data, which is often accompanied by specific, often commercial tools to analyze imaging data. Results are scalar data about one or multiple genes or proteins, usually measured for a number of time steps. Exceptional in this regard is experimental data from microscopy, which provides structural information about cells in addition to quantitative data. Accordingly, microscopy data requires visualization methods from the field of volume visualization [WOCH09, dLVvL06]).

For many methods producing quantitative data, the size of generated data sets is rather small and therefore often not subject of advanced visualization techniques. This is very different for data derived from micro array experiments. Analysis of data from these high-throughput methods is, on one hand, demanding due to volume and complexity of produced data, but on the other hand also crucial to understand relations of chemical compounds in cells. Micro arrays measure expression values of several thousand genes simultaneously, which can then be compared under different experimental conditions and for different time points. Due to its high relevance, the need to analyze micro array data has initiated a lot of visualization research in the last years.

Known information visualization views like scatter plots, parallel coordinates, time value plots, and heat maps are frequently used for micro array analysis [Die09], and are integrated in powerful commercial software such as Spotfire and GeneSpring. An example for the visualization of micro array data for thousands of genes in a time value plot is shown in Figure 2.3(a). Especially heat maps have received much attention. Their similarity with the original data on

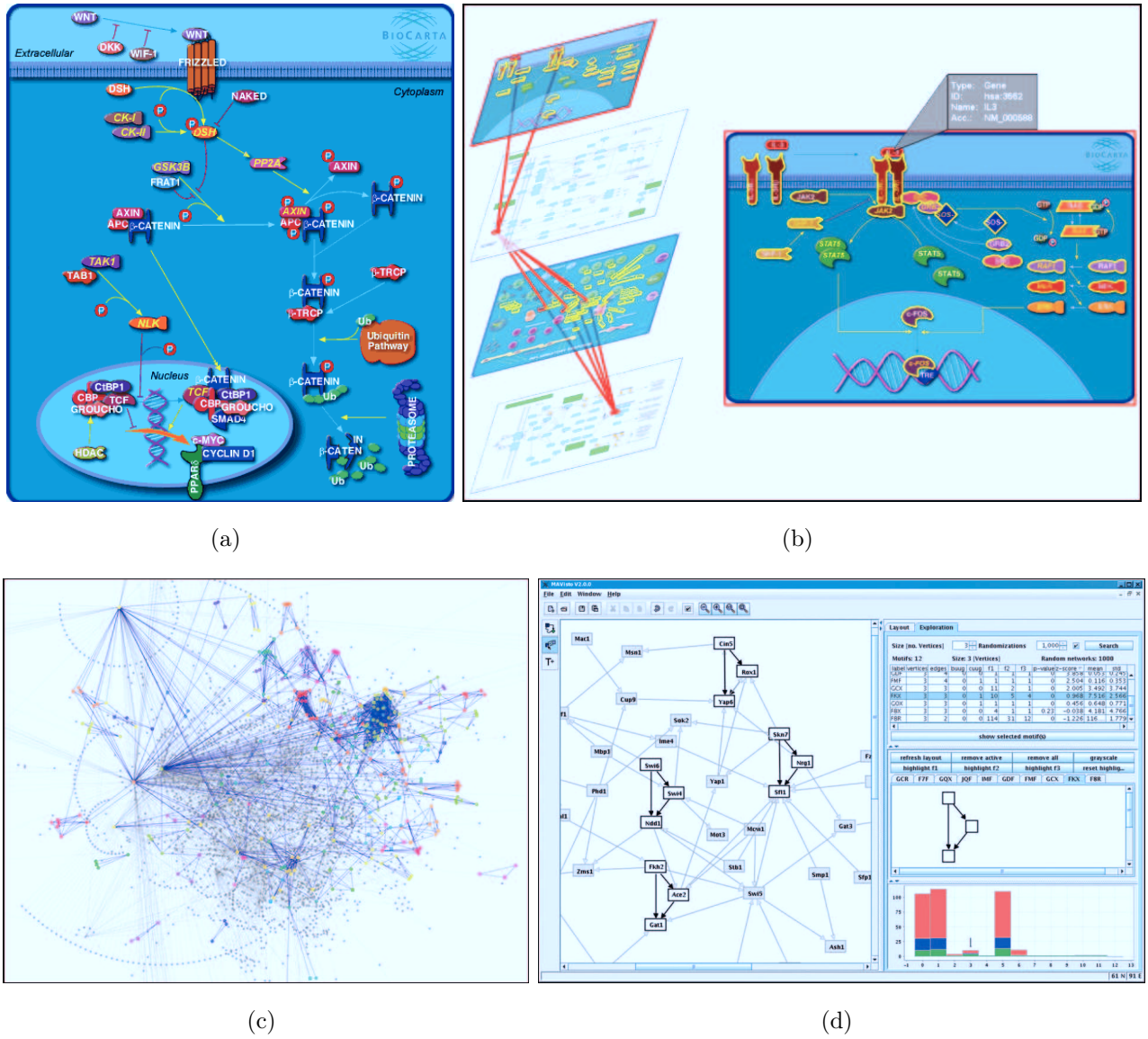


Figure 2.2: Exemplary visualizations of structural relations between components of the cell. (a) Static image of the Wnt signalling pathway, taken from the BioCarta website (http://www.biocarta.com/Default.aspx). (b) Exploration of multiple pathways by their interrelations in a multi-plane approach [SKKS08]. (c) Visualization of a reaction network with thousands of nodes in Cytoscape [SMO+03]. The image was taken from the Cytoscape website (http://www.cytoscape.org/). (d) Search for motifs in reaction networks in [SS05].

micro array chips has yielded a high acceptance in the biology domain. Also, animated scatter plots [CKC05] and 3-D scatter plots [SFHJ03, YISH06] have been employed for micro-array data.

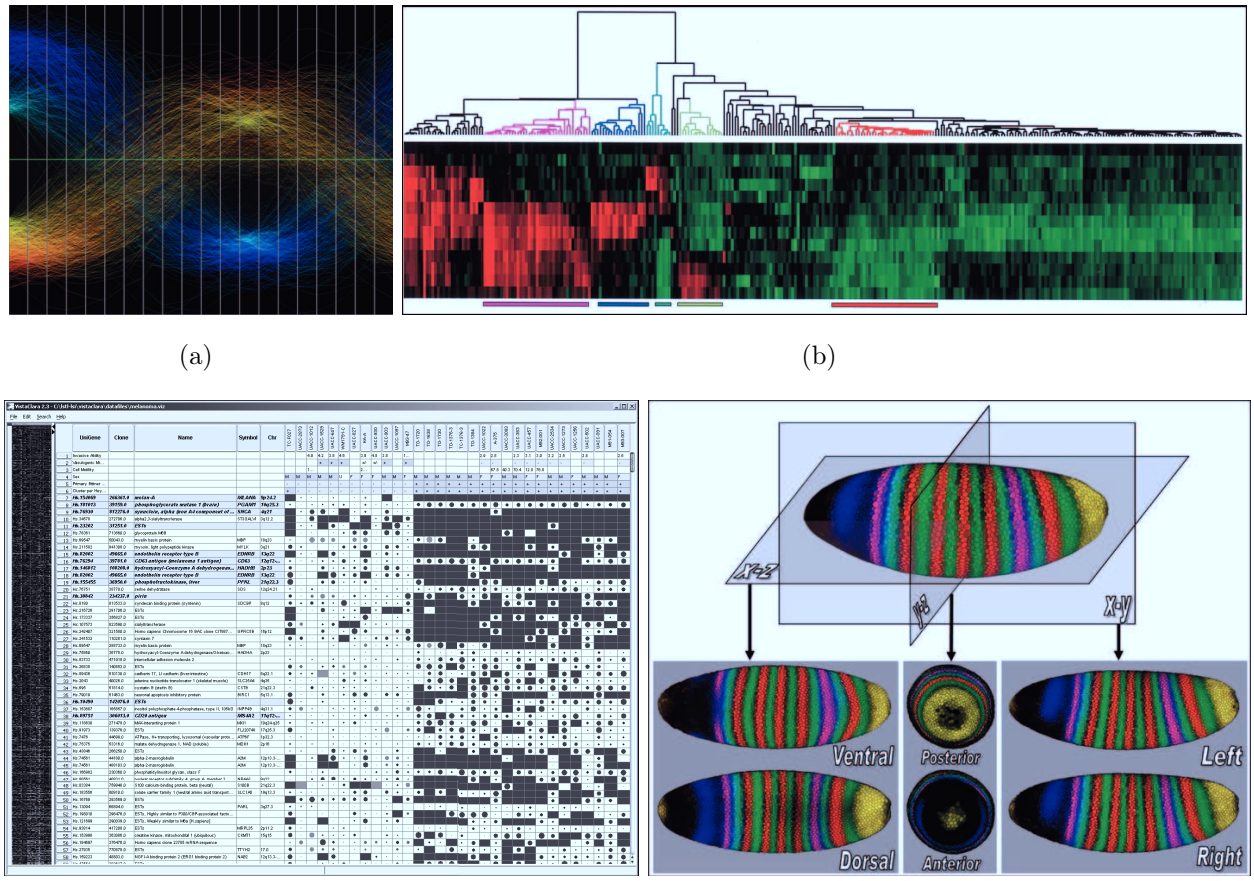
To cope with the complexity of the data, algorithmic methods such as clustering or self-organizing maps are frequently employed prior to visualization. Based on hierarchical clustering of similar expression profiles, the combination of dendrogram and heat map is commonly used to show clusters from micro array data [ESBB98, SS02a], as exemplified in Figure 2.3(b). Self-organizing maps have been proposed to group genes with similar expression profiles in [TSM+99]. In [SWC+02], a combination of k-means clustering and self-organizing maps is used as a basis to visually analyze expression profiles from different samples. Considering the uncertainty that comes when performing clustering, the inclusion of statistical information in the combined cluster dendrogram / heat map visualization has been described in [HDLT05]. Addressing similar objectives, [SGM+07] uses a heat map visualization to compare clustering results from two different methods. Further, graph based visualizations have been proposed for micro array data, for example in [BDBS04], where micro array data is shown in context of gene ontology data within tree maps. In [SSP+09], a node link layout is used to show cluster centroids as nodes and their similarity to other clusters by edges. For selected nodes, expression profiles of clustered genes are shown as time value plots.

In alternative to automatic pre-processing, another group of visualizations provides highly interactive interfaces to handle the size of micro array data. In this regard, an overview and detail concept was presented by [Kin04]. Samples are selected from an overview, which simultaneously shows expression profiles for several thousands of genes. The detail view uses a tabular approach that adapts the heat map metaphor (Figure 2.3(c)). A reordering of rows and columns is supported as well as automated table sorting. Focusing on temporal aspects, the highly interactive Time Searcher [HBMS03, HS04] has been used for micro array data.

An evaluation for a set of visualization tools for micro array data has been described in [SND04], including techniques described in [ESBB98], [SS02a], [HBMS03] and the commercial software tools Spotfire and GeneSpring. Given these tools, which combine clustering functionality with visualization concepts like Overview and Detail, linking of views, and interaction methods like brushing, domain experts were able to perform analysis tasks for different data sets ranging from 5000 to 15000 data items, although slight differences in usability of tools were encountered, which were partly due to different characteristics of data sets.

In addition to these information visualization techniques, gene expression data of cells in spatial context of cell tissues has been presented [WRH+09], thereby linking cell locations and gene expression. Supporting interactive functions like brushing of cells, the technique enables a visual analysis of gene expression in 2-D or 3-D spatial context in combination with abstract views like scatter plots (Figure 2.3(d)). In [HSPWR04], high-throughput data related

to proteins is shown in context of gene locations on the DNA. Bridging different levels of detail using a multiple view approach, the technique is able to handle several thousands of proteins.



(a) (b)

(c) (d)

Figure 2.3: Exemplary visualizations of micro array data. (a) In the time value plot [Die09], lines are colored according to values at the first time point. (b) Results of hierarchical clustering shown in a combined visualization of dendrogram and heat map with multiple time points per gene [ESBB98]. (c) Highly interactive overview (left) and detail visualization (right) [Kin04]. The detail visualization uses a table to show genes. Experimental data is shown in columns by adapting the heat map metaphor. (d) Micro array data shown in spatial context by mapping micro array data to the cells of the surface of an embryo [WRH+09].

Data Fusion Structural relationships among proteins and reactions and experimental data describing the dynamics of proteins comprise different facets of the same biological processes. To fully understand the underlying mechanisms, all known information about the biological entities has to be brought together in the visualization. The need to fuse data from different

sources is strongly emphasized in the literature [KHK+05, SND05].

In addition to structural relations as described by pathways and reaction networks as well as time dependent experimental data, descriptive information about proteins is important. Such annotation data provides additional information, which is biologically relevant, such as sub-cellular locations and can be retrieved from annotation data bases [3]. Two general alternatives can be used in visualization to bring the heterogeneous data together: The visualization of multiple aspects within one view, or the linking of multiple views showing different aspects.

The integrated visualization within one view is most often based on a graph visualization of the reaction network. Many visualization tools for pathways and reactions networks support the inclusion of additional data by mapping data to the visual attributes of nodes and edges [SMO+03]. Often, the color of nodes is used to encode gene annotation data [BWCT09, BST03]. However, only one value can be shown at a time.

To show multiple values over time or from different experimental conditions at once, the data is encoded in icons that represent the nodes. Usually, the number of time points is small and rarely exceeds ten. A common approach to show data in node icons are time value plots [BHK+05]. Additionally, direct value comparison of multiple time points is supported by showing stacked time value plots in each node icon (see Figure 2.4(a)). Also box whisker plots are proposed to show results of multiple experiments at once. Alternatively to time value plots, small heat maps at node positions have been used to encode experimental data [SKKS08, BCM+09, SHZ+07]. In [HLNZ05], micro array data is linked to edges by colored glyphs that are associated with edges.

The alternative to a single view, a visualization in multiple linked views, has also been employed numerous times. Several experimental conditions are shown simultaneously in small multiples in [BMGK08]. Sub-cellular locations of the proteins, shown as nodes, are considered by restricting their positions in the 2-D layout to certain locations. For each condition, a graph representation of the network is presented whose nodes are colored according to the value. Additional details are provided by parallel coordinates in a separate view (Figure 2.4(b)). BiologicalNetworks [BSRG06] uses, in addition to a view on the pathway, a heat map in a separate view to show multiple gene expressions for multiple time points. In Caleydo [MRS+09], multiple views showing data from different sources, such as patient data, pathways, and gene expression data, are combined and linked within a highly interactive bucket style visualization. The framework is intended for collaborative work of multiple experts from different domains and enabling the exploration of vast amounts of data. Also, the exploration of multiple heterogeneous

[3] http://proteinontology.org.au/

databases has recently been demonstrated with Semantic Substrates [LTG+10].

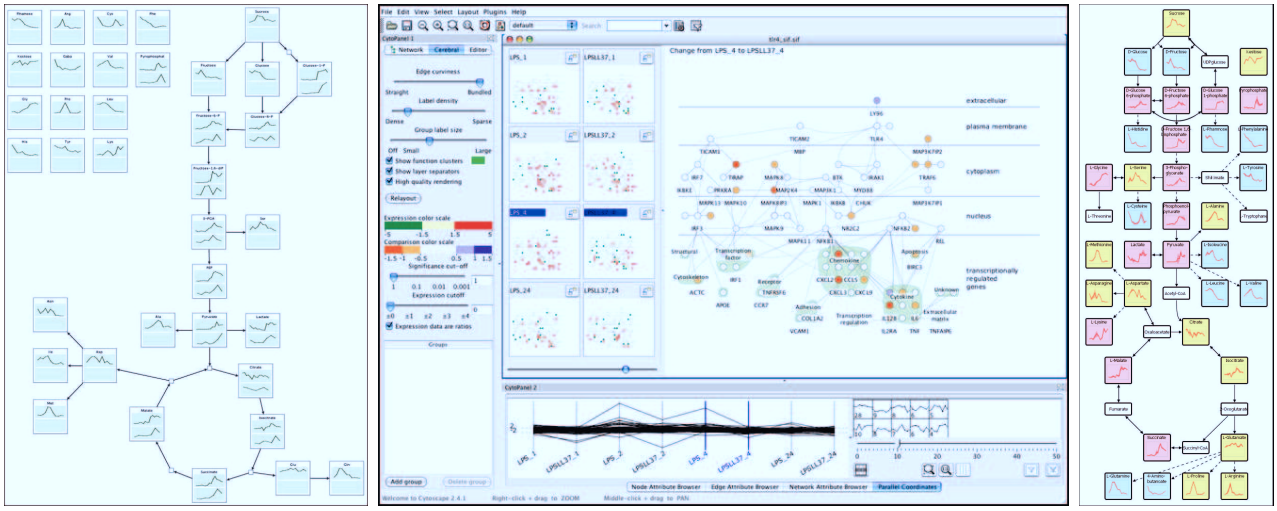
An evaluation of different approaches to show time series data within the graph representation has been presented in [Sar06]. The compared approaches include one view approaches with single time values per node and time series data in time value plots as well as multiple view approaches.

Beyond bringing together data from different sources in the visualization, the presence of multiple aspects of the data has been used in pre-processing steps to algorithmically identify relations in the data. In VANTED [JKS06], proteins are clustered by the similarity of their linked time series data (Figure 2.4(c)). In the visualization, nodes in the same cluster are encoded by the same color. Experimental data has been used to find sub-networks [BW09] (by evaluating if original and target node are both active), which are highlighted as active parts within the network. In VisANt [ZHaD04, HHW+09], gene annotation data is used to generate functional sub-graphs, which can be interactively collapsed or expanded in the visualization.

2.1.3.2 Visualization of Formal Structure

The complexity of models of cellular processes induces the need for visualization. This need has been acknowledged in the systems biology community. As formal structures can be interpreted as graphs, their visualization is highly related to the well established field of graph visualization. However, despite ongoing advances in graph visualization, most visual representations that are actually utilized within the application domain are particular developments embedded within larger tools that support modeling and simulation. Hence, the functionality of tools named in this section usually goes far beyond visual methods, but the focus of this review is set on the visualization of models in the application domain.

The development of these visualizations has been driven by requirements that researchers face in systems biology. The necessity for specific visual representations arises from particular features of biochemical models that are generally not comprised by universal graph drawing approaches. Most importantly, this includes the data representation. Biochemical reaction networks are often hyper graphs, where one edge connects more than two nodes. This is due to the fact that one reaction can affect more than two chemical compounds. As another point, biochemical reaction networks contain more than one type of chemical compounds and more than one type of relation among compounds. This led to the development of SBGN [NMS+08], the systems biology graphical notation. It defines standards for the graphical notation of model components as icons for both chemical compounds and their relations. Cell Designer [FMJ+08] supports the visualization of model structures with SBGN, as shown in Figure 2.5(a).



(a) (b) (c)

Figure 2.4: Exemplary visualization bringing together input data from multiple sources. (a) Single view approach to combine structural relations shown in the underlying graph and time-dependent experimental data in icons [BHK+05]. Within an icon, one time value plot is shown for each experimental condition. (b) Multiple view approach to combine structural relations between proteins with annotation data and experimental data [BMGK08]. Annotations like sub-cellular locations are considered in the graph layout. Experimental data is encoded in the color of nodes and edges. Small multiples are used for each time point and each experimental condition. Linked parallel coordinates provide further information. (c) Highlighting of algorithmically determined similarities in time series data by coloring node icons according to the cluster they belong to [JKS06]

Many visual representations of model structures adapt the biologist's understanding of cellular processes as genetic or biochemical reaction networks. To layout models, either editing functionality to build models is used – the layout is thus created by the user – or functionality for automatic layout is provided, usually done with the established force directed placement proposed by [FR91], which is for example the case for the on-line platform WebCell [LYC+06]. Examples that support model visualization in this sense are Cellware [DMS+05], SBMLSupportLayout and SBWAutoLayout [DBS06], aiming at complementing SBML models with layout information. Also specifically developed to meet requirements of the application is Narrator [MFPD07], which uses its own notation for models of cell biological systems. ProMoT [MSR+09] aims at visualization of modular models, as shown in Figure 2.5(b). Further, locations within the cell are an important factor in modeling, which is supported in the visualization for example

in Virtual Cell [SSM+03, MSS+08]. An example is given in Figure 2.5(c).

In addition, pathways described in the literature are directly adapted as the basis for modeling and simulation. Hence, graphical representations of pathways are also used to visualize model structures. In the majority of tools, models are shown by static representations, possibly supporting interactive manipulations of the layout, but not supporting an interactive visual analysis of the formal model structure. Providing more interactivity, Snazer [MIP10] supports several graph layout algorithms and basic analysis methods for graphs such as degree computation or shortest paths.

All these examples adapt the biologist's view on the formal model structure. The underlying formalism is not directly represented in visualization. Internally, the tools provide a mapping between the model structure given in a formalism and the visualization in a biological context and vice versa if editing is supported. The most prevalent formalisms are ordinary differential equations or partial differential equations. Stochastic approaches like the Gillespie algorithm are also supported by several tools.

To work directly with the formal notion of differential equations, many researchers use text editors embedded within modeling and simulation tools to edit the models. The JigCell model builder [VSR+06] supports a spreadsheet approach to define a set of ordinary differential equations forming the model structure. Bringing both representations together, a visual linking between model visualization in the biological context and its formal representation in a text editor is provided in PottersWheel [MT08] or in JDesigner[4].

All these works focus on rather small models containing up to a few dozen of chemical compounds, or nodes respectively in graph terms, which complies with current models sizes. Maintaining the biologist's view on models as biochemical reaction networks, many of the visualization tools introduced for pathways and reaction networks in the last section can be applied to analyze larger formal models, which are on the horizon in the application domain.

Alternative visual representations incorporating the underlying model formalism have been proposed within the application domain for Π-calculus [PCC06], state charts [EHC03, EHC05] that also include sub components (Figure 2.5(d)), and Petri nets [HR07], all providing static depictions of small model structures.

2.1.3.3 Visualization of Reference and Simulation Data, Parameters and Errors

For the analysis of simulation data, the visualization of the developments over time is essential. In this regard, tools that support the simulation of cell biological systems often provide capa-

[4] http://www.sys-bio.org/software/jdesigner.htm

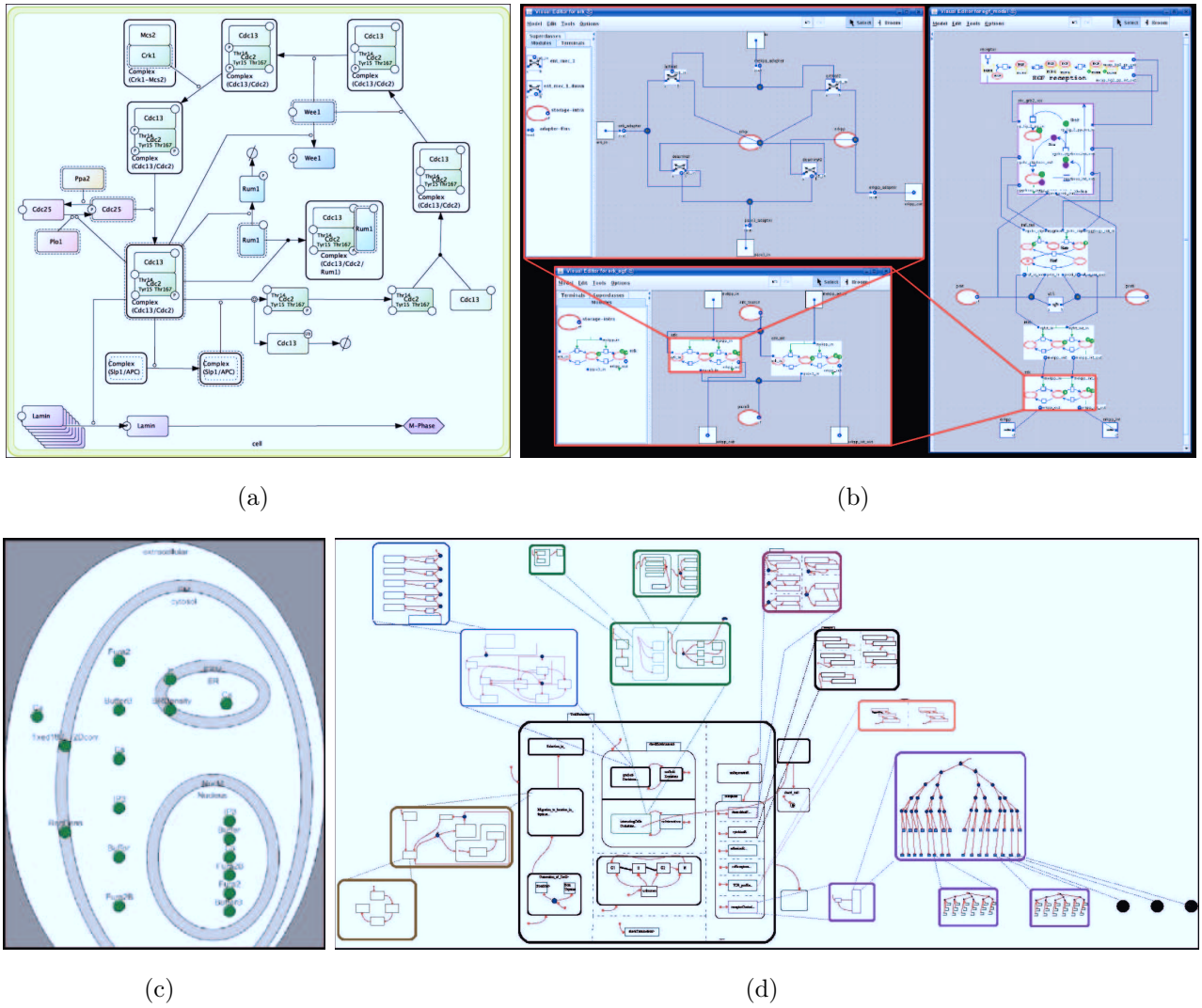


Figure 2.5: Exemplary visualizations for formal model structures. (a) Visualization in Cell Designer [FMJ+08] using the Systems Biology Graphical Notation . (b) Visualization of modular models [MSR+09]. (c) Visualization in context of cellular locations [SSM+03]. (d) Visualization with state charts [EHC05].

bilities to visualize the data. These capabilities are discussed in the following. A number of tools named in this section has already been mentioned in the context of visualizing the formal model structure. However, the visualization of the formal model structure and of the simulation data is often separated within the tools. Visualization methods that provide a linking of both are specifically discussed at the end.

In the application domain, simulation data is often derived from simulation based on ordinary differential equations. Usually, simulation is deterministic. Time value plots are predominantly

used in this regard. Among the multitude of tools that provide time value plots to present results from deterministic simulation are Copasi [HSG+06], Cell Designer [FMJ+08], JDesigner and SimDriver [BS06], E-Cell [THT+99], BioNessie [LJA+08], JSim[5], and Cyto-Sim [SM07]. The visualization of data from stochastic simulation, resulting in multiple time series, is also often done by time value plots that show either statistical properties like mean, variance or minimum and maximum values [EFZ08, MIP10] or all runs simultaneously in combination with histograms [CFF07]. The visualization of simulation data by statistical properties as in [MIP10] is shown in Figure 2.6(a). In general, time values plots represent time series data for a small set of variables or support the comparison of experimental and simulated data for single variables. Occasionally, other information visualization techniques are used, such as scatter plots [AMYS+09].

An increasingly important aspect in the simulation of cellular systems is the spatial context A visualization that combines a view on protein concentrations with a view on single proteins is presented in [BCPS03]. In a cellular 2-D context, temporal simulation data is mapped to time, resulting in an animation. Animated transitions between the views help the user to maintain a mental image. The views are further coupled with other visualizations for experimental data, such as a heat map showing the simulated expression of genes over time. In 3-D context, locations of individual proteins are visualized in [HFE05] and [FKRE09]. While the first example provides a rather simple animation and non-interactive presentation of simulation results, the latter approach comprises multiple visualization concepts to analyze cellular processes by movements of single proteins within the cell (Figure 2.6(b). Locations of proteins are visualized along with relevant structural components of the cell. In addition to an animation of movements over time, the visualization highlights reactions among proteins due to collisions and trajectories of interactively selected proteins as 3-D lines, showing multiple time points at once. An alternative rendering style resembling microscopic images aims at facilitating the comparison with experimental data.

Addressing a different scale of biological systems, visualizations of cell populations in spatial context are presented in [CHC+05] and [EHC05]. The latter one supports an on-line manipulation of the running simulation through an interactive visual interface showing current simulation results.

In all these approaches, visualization of simulation data is separated from visualization of the model. Approaches that put simulation data in context of the model commonly use animation to show values over time. As simulated variables are usually related to components of the

[5] http://physiome.org/jsim/

model, visual attributes of node icons are adapted to show the simulated data over time. These visual attributes include size, color, or length. Examples are SimWiz [RK04], which maps scalar values to the size of circles representing model components, and Narrator [MFPD07], where scalar values are shown as fill levels of rectangular icons (Figure 2.6(c)). Extending the 2-D representation in 3-D to include dynamic data, scalar values are mapped to the height of cuboids placed in 3-D over a 2-D model representation in [BS06] and [Qel07]. In Figure 2.6(d), a snapshot of the animation from [BS06] is shown. These approaches make use of the biologist's understanding of computational model as biochemical reaction networks.

In [HR07], animation is used to show simulation data in the context of a Petri net. Over time, data is mapped to the color of the Petri net places.

The majority of visual methods to show simulation data within the application domain are static and non-interactive. In this sense, they provide a presentation of data, rather than interactive visual analysis.

2.1.3.4 Presentation of Data, Model, and Results

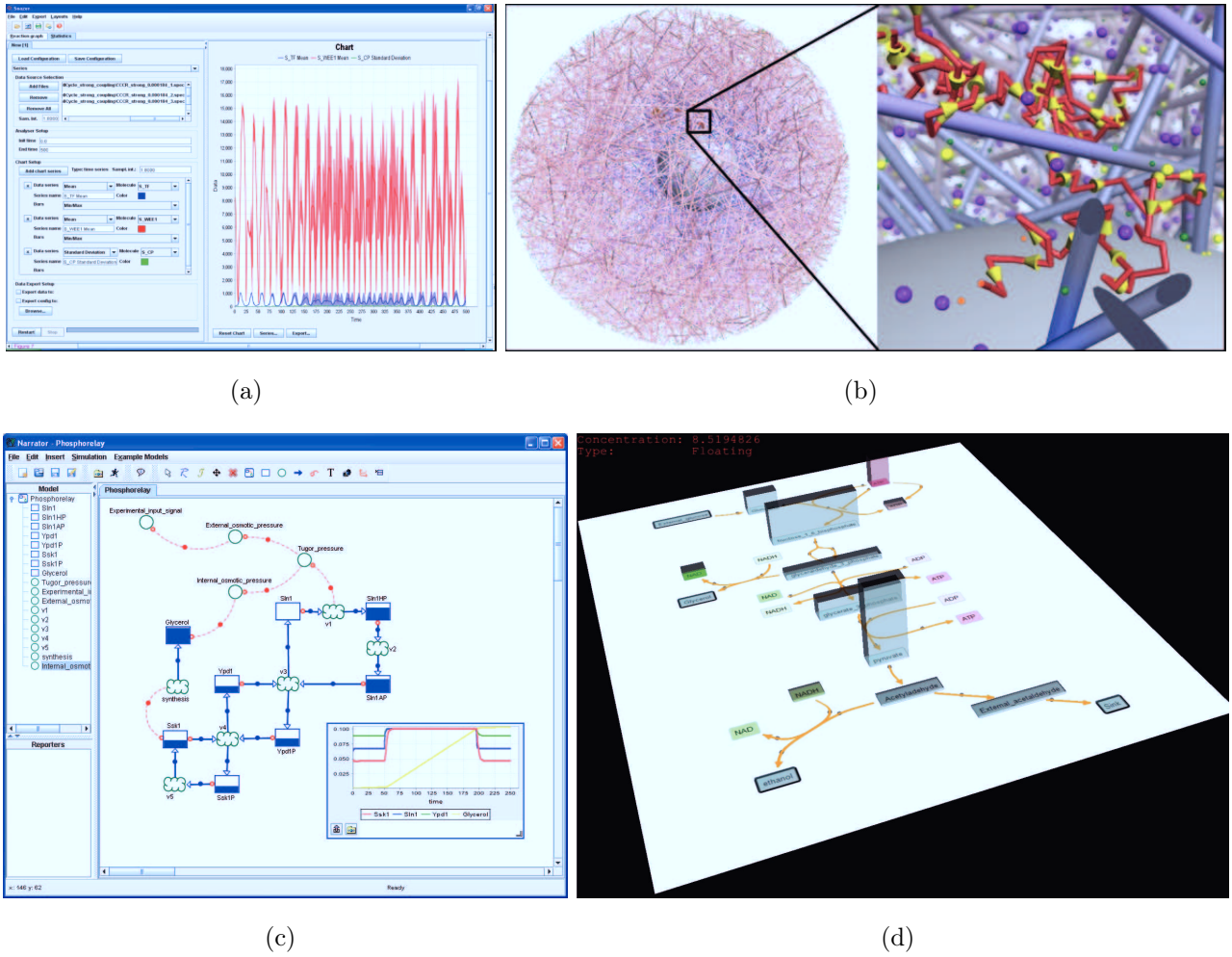
Almost any data that has been discussed in the previous sections can be subject to presentation, aiming at the communication of findings from a modeling and simulation project to external persons. As many of the visualization methods can be ranked as presentation techniques rather than methods of visual exploration and confirmation, they can be used to present results to third-party users. This is in particular true for representations of chemical reaction networks in combination with experimental data or for animations of simulation data in spatial or model context. Various tools support export of visual representations into image files, such as [BST03] and [INM+05], or to store animations as videos [MSS+08].

Moreover, some visual representations explaining functions of cellular processes have been developed for the purpose of presentation. This includes animation videos from Howard Hughes Medical Institute[6]. In [MJR+05], important aspects for building animations for learning and presentation purposes are discussed. However, creation of such exploratory animations is not facilitated by existing tools established in the application domain.

2.1.3.5 Summary

In summary, a multitude of visualizations has been employed in the application domain, comprising mainly the segments of visualization of input data, the formal model structure, and

[6] http://www.hhmi.org/biointeractive/animations/index.html



(a) (b)

(c) (d)

Figure 2.6: Exemplary visualizations of simulation data. (a) Stochastic simulation data, shown by statistical properties of selected variables in time value plots [MIP10]. (b) Trajectories of proteins shown in the 3-D context of the cell [FKRE09]. (c) and (d) Simulation data in context of the model. In (c), fill levels of the icons represent current values related to nodes [MFPD07] In (d), the model is shown in a 2-D plane, data values are mapped to the height of cuboids in 3-D [BS06]. In both approaches, the dynamics of the data are shown in an animation.

simulation data.

Regarding the **visualization of input data**, a large number of techniques has been developed to address the specific requirements of the application domain. Three main objectives appear in this segment:

- Visualization of structural relationships between chemical entities, ranging from very small graphs with up to 50 nodes to graphs with several thousands of nodes, is often con-

ducted with node link layouts. To cope with clutter that arises for hundreds or thousands of nodes, most techniques provide filtering methods to reduce the size of the graph.

- Visualization of experimental data, specifically micro array data, is addressed by powerful methods of visual analysis that are able to handle the complexity and size of the data.
- Visual fusion of data from different sources, bringing structural relationships, experimental data, and annotation data together in either one view or multiple linked views.

Regarding the **formal model structure**, most visualizations adopt the biologist's view of the model and display it as a graph of chemical compounds and reactions between them. According to the small size of currently used simulation models, visualizations are designed to cope with a few dozen components. Nevertheless, text editors are widespread to work with models.

To visualize **simulation data**, static time value plots comprising few variables are most often used. In addition, visualizations have been proposed that show simulation data in the context of the underlying model, representing the dynamics over time by animation.

For **presentation of results**, few techniques are found that have been specifically used in the application area. Methods comprise static images and videos, which are either extracted from simulation tools or specifically created with specialized imaging or animation software.

A significant disparity can be seen between both the number and the capabilities of techniques to visualize input data on the one hand and the other three segments: For the visualization of input data, especially for micro array data, a much higher number of techniques is available with sophisticated visual interfaces. This is due to the fact that the analysis of biological data is an important part of clinical research, while modeling and simulation represents a relatively young methodology to support the understanding of cell biological systems.

2.2 Discussion and Focus of this Work

Based on the work flow of modeling and simulation, four segments of visual support in the application domain have been identified: visualization of input data, model structure visualization, visualization of simulation data, and the presentation of results. Due to the appearance of a multitude of data sets with various characteristics in all segments, each of them is a challenging field for visualization research.

The broadest range of sophisticated visual analysis techniques has certainly been proposed for the analysis of **input data**. However, the majority of visualization methods for analysis

of input data have been presented independently from the goal of quantitative modeling and simulation. They primarily support qualitative analysis of data. New approaches are necessary with respect to the following developments.

- The tremendously increasing amount of data, which has to be analyzed, induces the need for further research, which has recently been moving in the direction of data fusion.
- More research will be necessary to support visual analysis of input data as it is required for the quantification of computational models.

Visualization methods in the next segment of visual support, the **visualization of formal model structure**, can be characterized as presentation techniques rather than supporting exploration or confirmation. To a large extent, they are focused on the well-established formalism of ordinary differential equations. The visualization often adopts the biologist's view on the model, showing the set of ordinary differential equations as a biochemical reaction network. However, new developments in the application domain require additional methods.

- With the establishment of new formalisms based on discrete events, which are able to incorporate more detailed spatial context, other representations have to be found to visualize formal structures.
- Models will increase in size and complexity in the near future and be built of multiple components at various scales, which require novel visualization methods.

So far, only first steps have been presented in this direction.

For the **visualization of simulation data**, new simulation approaches comprising stochasticity and spatial context lead to new challenges.

- Resulting data sets are large, as biochemical interactions are simulated in more detail, leading to a high number of time points.
- Replications of simulations as required for stochastic approaches and the incorporation of spatial context increase both volume and complexity of data.
- With larger models in the future, the number of variables measured over time will further increase.
- Simulation is often so time consuming that visualization of results has to be uncoupled from the simulation process.

These apparent problems can only be solved with new and highly interactive visual analysis methods.

At last, also the **presentation** of results as the last segment of visual support requires further research. So far, presentations are either static images, which are generated as a side product of existing tools, or elaborate videos generated by animation experts with specialized software. Despite well-designed exploratory presentation of cellular processes, videos do not leave room for user interaction beyond play/pause controls. In consequence, additional research has to come up with solutions for the following goals.

- New techniques are required that facilitate the generation of presentations.
- More interactive presentations are necessary that are able to respond to emerging questions during presentation.

In general, it can be stated that powerful visualization approaches exist in all segments. However, additional research in visualization is required to cope with new challenges in the application domain and complement the range of available visualization methods. The design of new tailored visualization methods is necessary at all segments of visualization in order to provide comprehensive visual support throughout the modeling and simulation work flow. As providing visualization techniques is very challenging for each segment, the remainder of this work will focus on one segment: the **visual analysis of simulation data**.

Setting the focus on this segment of visual support is motivated by the high relevance of understanding simulation data. Generation of simulation data can generally serve different purposes, comprising for example:

- Development and implementation of new modeling and simulation approaches.
- Analysis of simulation data as an intermediary step during the modeling and simulation process.
- Drawing conclusions and generating new hypotheses about the source system.

Within the research training school **dIEM oSiRiS**, all three aspects play an important role. The first goal arises from one specific focus of the research training school: the development of new modeling formalisms and simulation algorithms that are able to reflect spatial effects in cellular processes. Visualization of simulation data is needed to check the plausibility of new approaches as well as the quality of their implementation. Hence, visualization is needed in the debugging process of new modeling and simulation approaches. The latter two goals directly

follow from the work flow of modeling and simulation, which is exemplary executed for specific aspects of the Wnt signaling pathway in dIEM oSiRiS.

This leads to new challenges for visualization. Data sets with diverse characteristics are generated by simulation. As already indicated, complex data characteristics specifically arise from modeling and simulation approaches based on discrete event systems. In general, **heterogeneous simulation data** is generated from discrete event systems. In addition to the system state, which is given by values of state variables over time, also events that lead to changes in the system state provide valuable information about model behavior. Both states and events are time dependent and multivariate. A particular challenge is the large number of time points, as one time point is tracked for every occurring event. Also, approaches increasingly include spatial context, adding further dimensionality to generated data. Moreover, stochasticity, a common property of discrete event systems, requires simultaneous analysis of multiple simulation runs. All in all, simulation data from discrete event systems comprises a complexity that is rarely addressed by existing visualization techniques.

Further, uncoupling simulation process and visualization of simulation data, due to complex and time consuming simulation, results in an information loss during visual analysis. This challenge can be met if the visualization allows gaining insight into the context of data generation. Especially considering the multitude of data sets that is generated during a modeling and simulation project from different contexts, **integrating context information within data visualization** can support analysis significantly. Further, involving other experts in visual data analysis can be facilitated, even though they might not have participated in the process of data generation. Considering the data generating context during development of new visualization techniques offers a promising approach to support analysts in understanding their data and decision making.

In the remainder of this chapter, a conceptual approach is developed with respect to these challenges for the visualization of simulation data. It provides the ground for the systematic development of new visualization techniques for simulation data in Chapters 3 and 4.

2.3 General Taxonomy for the Visualization of Simulation Data

Integrating the data visualization into the context of data generation, as it has been stated as one main challenge for this work, is of specific interest to analyze large and complex simulation data. Consequently, the investigation of the data generating context for simulation data is the

starting point for a conceptual approach to visualize complex simulation data. This is subject of this section. The result of this analysis is the development of a hierarchy of process levels. It allows identifying which information has to be additionally included in data visualization to communicate the data generating process. But beyond, this hierarchy of process levels provides a solid foundation for new visualization techniques for simulation data, which will be introduced in the subsequent Chapter 3. Based on the visualization approach derived in the following, these new techniques can be developed to support analysis goals that commonly appear in the application domain, thus addressing open challenges in visualization research.

2.3.1 Levels of the Data Generating Process

The starting point for a visualization approach for complex simulation data is the identification and explicit description of the data generating context. In general, simulation data reflects the behavior of a model. This behavior is examined under certain experimental conditions. Thus, simulation data is always derived in the context of a model and an experiment. For stochastic simulation, which has been stated as one important challenge that comes with new modeling and simulation approaches, the same experiment has to be executed multiple times, as on run represents only one possible behavior of the system.

This data generating process can be captured by the introduction of process levels. Highest level is the *Model*, as all simulation data is derived from one model, which includes basic components of the system. Usually, multiple experiments are executed based on one model. Hence, the next level is the *Experiment*, describing one specific parameter setting for the model components. Resulting from these experiments, the simulation data is derived. As stochastic simulation requires the execution of multiple runs, the two levels *Multi-Run Simulation Data* and *Single-Run Simulation Data* are differentiated, the former level regarding the collective simulation output of one experiment and the latter regarding one simulation run of one experiment.

These process levels form a hierarchy, which is displayed in Figure 2.7.

The introduced process hierarchy provides a universal concept to explicitly describe the data generating process for stochastic simulation data. This systematic view has significant advantages for the visualization design:

- The explicit description of the context of data generation provides the ground for its tight coupling with data visualization.
- Relevant features of each process level can be identified to consider them in the visual-

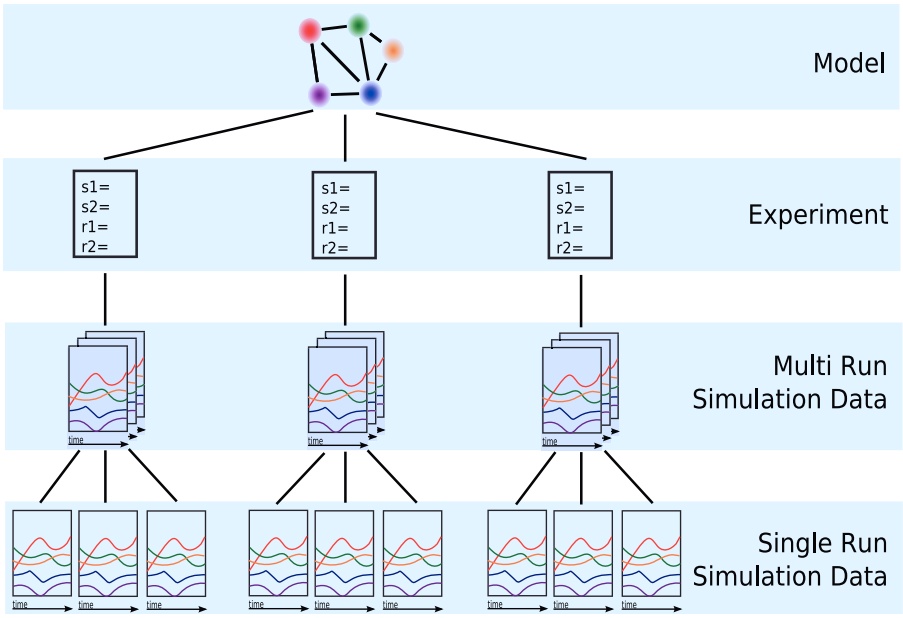


Figure 2.7: Overview on process levels in stochastic simulation.

ization.

Further, to understand the implications of a model or an experiment on the behavior of the simulated system, the simultaneous analysis of multiple sets of simulation data is demanded at different abstraction levels, which can now be assigned to the process levels as follows:

- at the model level: analysis of simulation output for one model, comprising multiple experiments
- at the experiment level: analysis of simulation output from one experiment
- at the multi-run level: analysis of multiple runs
- at the single-run level: detailed analysis of a single run, also considering additional aspects such as spatial context

Thereby, the process levels provide a valuable instrument to tailor visualization techniques to particular objectives.

2.3.2 Process Levels in the Application Context

Although the derived process level hierarchy applies for stochastic simulation in general, the concrete features that characterize each process level depend on the application domain. In the

following, the hierarchy of process levels is walked through with a focus on the intended application of modeling and simulating cell biological systems as stochastic, discrete-event systems. Discrete-event systems [BJSCNN01, Zim08] provide a formal approach to model and simulate these biological reaction networks. In general, the system state is defined by the values of state variables. These values are altered at discrete time steps by the occurrence of events, which happen randomly based on a stochastic distribution. The system states over time specify the system behavior.

Each process level is introduced with respect to the application. Both the particular data and the goals of visualization are subject of the visualization at each process level and therefore described in the following.

2.3.2.1 Model

In general, a model consists of important components of the investigated system and their interrelations.

An established approach to represent models of cell biological processes is the description as a biochemical reaction network. It describes biological processes as reactions of chemical compounds. In model terms, chemical compounds are the model components, their interrelations are given by reactions. Each reaction can have several reactants and reaction products. Another important feature of reaction networks are reverse reactions. Reactants of one reaction are reaction products of the oppose reaction and vice versa.

Describing a biochemical reaction network as a discrete-event system, the quantities of the chemical compounds are modeled as state variables; the reactions are modeled as events. The reaction rates are mapped onto stochastic distributions to model the occurrence of events. Each event changes the values of the related state variables in the system according to the related reaction. Table 2.1 gives a short overview on the mapping of components and interrelations of biochemical reaction networks to discrete-event systems.

	Model components	Model relations
Discrete-event systems	State variables	Event types
Reaction networks	Chemical compounds	Reactions

Table 2.1: Mapping of components and interrelations of biochemical reaction networks to discrete-event systems.

Data Characteristics At the model level, the relevant data to describe the data generating context stems from the model. It can be expressed as a graph $G(V, E)$ consisting of a set of vertices V (nodes) and a set of edges E that link nodes. The state variables are nodes and event types are edges. Specific for biochemical networks, an edge can link more than two nodes, resulting in a hyper graph. Deviant from standard graph definitions, biochemical reaction networks may include "production" or "degradation" edges, which are linked to only one node. Also, edges can be one or two-sided and, for each edge, in going and out going nodes have to be separated. Models currently inspected in modeling and simulation can be generally considered to be rather small, not having more than 30 nodes.

Visualization Goals Visual analysis at the model level requires the visualization of the model structure, including the model components and their interrelations. In addition, the model level is the highest level of the process level hierarchy. This implies that visualization at the model level should provide an overview of the model behavior. Thus, simulation data, distinguished for experiments, has to be shown at the model level. This leads to the following visualization goals:

- evaluation of the model structure
- linkage of data to model structure
- analysis of data in context of experiment description
- comparison of experiments

2.3.2.2 Experiment

An experiment describes the specific settings of a model for which the simulation is executed. Hence, an experiment can be regarded as an instance of the model under certain conditions. In the following, these conditions are referred to as the experiment description.

For the application at hand, the description of an experiment is related to both components and interrelations of the model structure. Related to model components, experiments of biochemical reaction networks are carried out for different initial concentrations of chemical compounds, which are equivalent to initial states of state variables. The experiment description for interrelations of the model components is related to the reactions. Each reaction is characterized by a reaction rate. In a discrete-event system, the reaction rate is expressed as a stochastic distribution for the occurrence of events. The duration of the simulation, or the simulation time, completes the experiment description.

Data Characteristics The data characterizing the experiment description are initial states of state variables and rates of reactions. Since state variables as nodes and reactions as edges form the model, the experiment description adds one attribute value to each node and each edge. For one experiment, these attributes have a fixed value, while they vary among experiments. In addition, each experiment is characterized by its duration over time.

Visualization Goals An important analysis goal related to the experiment level is to analyze dependencies among experiment description and resulting simulation data. Therefore, both experiment description and multi-run simulation have to be visualized. The experiment description is closely linked to the model structure. Hence, the visualization goals are:

- linkage of experiment description to model
- linkage of simulation data to experiment description

2.3.2.3 Multi-Run Simulation Data

For stochastic simulation, data from one experiment is not deterministic. To analyze the system behavior for an experiment, it is performed multiple times in order to gain an overview on possible simulation outputs. Due to stochastic distributions used to generate events, simulation data is different every time the simulation is performed – although simulation data is derived from the same model and the same experiment description. This experiment description is also constant with respect to the random number generator that is used to emulate the stochasticity. Only the seed value, the initial value of the random number generator, is changed in every simulation run. Only from the multitude of runs, conclusions about the likeliness of different outcomes can be made.

Data Characteristics Multi-run simulation data consists of a number of data sets, each as the result of a single run. The number of runs can range from few to thousands. Each run produces time dependent data with distinct sequences of time points, as changes to the system are induced by stochastic events. Time series are given for all state variables, which are contained in the model as its state variables. In addition, events occurring at each time point are linked to event types in the model.

Visualization Goals The dependencies of simulation data on the model structure and the experiment description need to be analyzed. The time dependent, multivariate simulation data

from different runs has to be compared in order to identify similarities and deviations. This leads to the following goals:

- linkage of data to experiment and model
- comparison of runs
- analysis of data over time
- analysis of dependencies among variables

2.3.2.4 Single-Run Simulation Data

The data that results from one simulation run represents one possible behavior of the model. The level of single run simulation data focuses on the characteristics of the resulting simulation data.

Data Characteristics Simulation data is time dependent, multivariate, and heterogeneous. It consists of a sequence of time points. At each time point, the state of the system is defined by current values of state variables. Each time point is further characterized by the occurrence of one event, which has changed the system state by altering the values of one or more state variables. Events can also be seen as an important characterization of simulation data. As state variables and events have different quality, the resulting data is heterogeneous.

Visualization Goals The analysis of a single run requires a detailed inspection of the temporal developments of state variables and occurrences of events. Rather than getting an overview on the behavior, explanations for observed behavior need to be found. What developments over time led to a specific state of the system? In this context, dependencies among state variables and also among events and state variables are of special interest, which is summarized by the following visualization goals.

- detailed visualization of states and events over time
- analysis of dependencies among events and states
- analysis of dependencies among multiple variables for both state variables and event types
- analysis of additional aspects such as spatial context

2.4 Summary

In this chapter, a necessary first step to provide comprehensive visual support for the modeling and simulation of cell biological processes has been made: the conceptual integration of visualization into the application domain, which has been published in [UBJ+07]. This integration has been based on the work flow in the domain, thus ensuring coverage of data analysis requirements. With respect to the data generated along this work flow, four segments of visual support have been identified: the visualization of input data, the visualization of the formal model structure, the visualization of simulation data, and the presentation of results. As the bottom line from reviewing existing visual methods in the application domain, all segments comprise open visualization problems and require further research.

For this work, the focus has been set on the visual analysis of simulation data from discrete event based systems, motivated by two main challenges for visualization: The visual integration of data into the data generating context and the visualization of large, heterogeneous, and complex data sets. Targeting these goals, the second part of this chapter introduced a conceptual approach to handle such simulation data, which was published in [US09]. The main contribution is the introduction of process levels that cover the data generating context for stochastic simulation data. Each level – model, experiment, multi-run data, and single-run data – comes with certain data characteristics and visualization requirements. The process levels serve as the basis to derive new visualization concepts that bring together data with its context of generation and further enable the analysis of complex data under different aspects. By deriving tailored visualization techniques for all process levels, visualization can focus on relevant aspects of the data at each level, thereby supporting multiple abstraction levels. These techniques will be subject of the subsequent Chapter 3.

Chapter 3

Visual Exploration of the Simulation Process

The goal of this chapter is to derive tailored visualizations for three of the process levels – model, experiment, and multi-run data – with respect to the visualization goals that have been discussed in Section 2.3.2. In contrast to the remaining process level of single run simulation data, which will be subject of the subsequent Chapter 4, those three process levels have in common that multiple sets of simulation data, which have been derived from one model but in different experiments or runs, have to be brought together visually to analyze and compare them. In this regard, the inclusion of the data generating context, given by model and experiment description, is of great importance to understand the data. Relating the different sets of simulation data to the context in which they have been derived, the causes for similarities and deviations among them can be instantly extracted from the visualization. Making the relations among data and the context of data generation explicit requires bridging the gap between heterogeneous pieces of information. In this regard, visualization concepts for these process levels need to go beyond mere data visualization, but rather expand the scope of visualization in the application domain towards the synthesis of a multitude of information for analytical reasoning, as it is the goal of the recently established field of Visual Analytics [TC05].

In Section 3.1, visualization concepts for the three process levels are developed with respect to the features of data sets that have been gathered within the research training school **dIEM oSiRiS**. The focus is set on a visual integration of simulation data and the context of data generation. This requires abstracting heterogeneous pieces of information to their most relevant aspects with respect to the current goals of visualization. However, to analyze individual pieces of the data in more detail, alternative views are necessary. Thus, this chapter presents two

additional visualization techniques.

First, this includes a visualization capable of exploring the underlying model in all its facets (Section 3.2). Instead of linking it to simulation data, the characteristics of the model, comprising structural relations as well as multiple attributes, are in the focus of the visualization. The visualization is able to cope with models consisting of tens of thousands of nodes and edges.

Second, additional concepts are discussed to analyze multi-run data (Section 3.3), based on the idea to group runs by their similarity and to visualize the resulting groups by their statistical properties. Based on a discussion of possible approaches, a new visualization is introduced to support the visual analysis of interactively generated subsets. The integration of a new view, showing statistical properties of subsets, within the multiple view framework SimVis [DMG+05, Dol04] provides the simultaneous visualization of different facets of the subsets.

3.1 Tailored Visualization Concepts for Process Levels

The ground for visualization at each process level is provided from the discussion of visualization goals in Section 2.3.2.1. Each process level comes with one main visualization goal:

- Model Level: analysis of simulation output for one model, comprising multiple experiments
- Experiment Level: analysis of multi-run simulation data from one experiment
- Multi-run Level: detailed analysis of individual runs

These different goals induce the need for tailored visualization concepts for each level. Nevertheless, all levels share the requirement to show the relevant simulation data in the context of the data generating process. This is a main goal of this work: the tight visual integration of both aspects to support the understanding of the underlying simulation process. In the following, the general concepts for all three process levels are developed to achieve this visual integration and support the related visualization goal.

3.1.1 General Concept

The basic idea is to integrate the visualization of simulation data within a graph visualization of the model, resulting in one single view. The model is the foundation of the whole simulation process. To understand the dependencies in the heterogeneous, multivariate simulation data,

the structural relationships among state variables and event types, which are expressed in the model, are extremely important. To visualize the simulation data in the model context, the corresponding part of the simulation data is shown along with the visual representations of the state variables and event types in the model. In addition, the individual parts of the experiment description are mapped to the state variables and event types in the model visualization.

This fundamental concept is used at all three process levels to visually integrate simulation data and the context of data generation. Despite its potential to support the understanding of the simulation process, the concept involves several challenges for the visualization. Essentially, visualizing simulation data within a visual graph representation limits the space that is available for the simulation data. This is critical with respect to the following characteristics of the simulation data, which apply to all three levels:

- The simulation data consists of multiple runs.
- Each simulation run may contain a high number of time points.
- Value ranges in the simulation data may be very heterogeneous, making a global comparison among state variables or event types difficult.

A strong abstraction of the simulation data is necessary to show it in the context of data generation.

Hence, the basic visualization concept at all levels is to bring together model structure, experiment description, and an overview on associated multi-run data, which is abstracted to most general trends, in one single view. Its adaptation to all three process levels in order to support the related visualization goals is described in the following.

The Model Level requires the comparison of experiments.

Hence, time dependent multi-run data has to be compared for different experiments along with the corresponding experiment description. In general, two concepts are applicable for comparison of data [PP95]: data based and image based comparison. Data based comparison refers to the fusion of multiple data sets into one data set, which is then visualized to show deviations among the data sets. Difference images are well known examples for data based comparison. The alternative, image based comparison, presents one image for each data set. Deviations among the images imply deviations in the data.

In order to apply data based comparison, data from experiments have to be combined so that the resulting data reveals disparities in the data and the context of data generation for

each experiment. This is not a trivial task. Considering the general concept of integrating multi-run simulation data within the data generating context, it has to be strongly aggregated. Generating such an abstract view from data that has been computed to reflect differences between experiments is likely to introduce uncertainty in the visualization. Further, data based comparison is usually limited to the comparison of two data sets, which is a severe restriction for the comparison of multiple experiments.

To circumvent these problems, an image based comparison of multiple experiments is used. As the model structure is equivalent for all experiments, multiple images can be used in two ways. Either, experiment description and multi-run simulation data are shown for multiple experiments within a single graph visualization of the model. Or, experiments are shown in separate views, each representing the experiment description and multi-run data of one experiment with respect to the model structure. The former primarily focuses on a direct comparison of values for individual state variables and event types among experiments. This is bought with the cost that the multivariate dependencies among both the multi-run data and the experiment description within one experiment are hard to evaluate. Thus, the second approach is chosen: Multiple images, each showing one experiment, are displayed simultaneously. Hence, the basic component at the model level is a view on one single experiment, which gives an overview on the relations between model, experiment description, and multi-run data. This basic component is exactly the visualization required to cover the Experiment Level, and will be discussed with respect to that level.

The relevant challenge at the Model Level is the visual linking of the views to enable a comparison of both the data and the data generating context among multiple experiments.

The Experiment Level demands for the visualization of the multi-run data from one experiment in its data generating context, given by the model structure and the experiment description. In this regard, the basic visualization concept covers the relevant visualization goals at the Experiment Level.

The Multi-Run Level requires a more detailed analysis of the runs of one experiment than the other two levels. The general visualization concept results in a strongly aggregated view on the simulation data, which is not sufficient to fulfill the visualization goal. Nevertheless, a view that communicates the context of data generation and relates it to the resulting data is demanded at the Multi-Run Level. In this regard, the basic visualization concept covering a single experiment is an important component of the visualization at this level. However, to

cover the remaining visualization goals, other views are needed to support the selection of runs as well as more detailed visualization of the time series data. Hence, a multiple view concept is used at the Multi-Run Level in which an overview, which matches the requirements of the visualization at the Experiment Level, is linked with additional views for run selection and detailed time series visualization.

As the bottom line of this discussion, all levels require a basic view that shows the simulation data of a single experiment in the context of its data generation. In the remainder of this section, this basic view is called Experiment View. In this view, various challenges need to be tackled.

These include

- appropriate graph layout for the model
- linkage of experiment description and multi-run simulation data to the nodes and edges of the graph
- aggregation of simulation data with respect to multiple runs and a potentially high number of time points, as the display space for simulation data is limited
- a concept for a global comparison of potentially heterogeneous local value ranges of individual state variables and event types

A visualization covering all these aspects is developed in Section 3.1.2. Thereafter, additional requirements to cover the Model Level and at the Multi-Run Level are considered. Specifically at the Model Level, the linking of multiple Experiment Views to compare experiments is discussed in Section 3.1.3. At the Multi-Run Level, an appropriate multiple view concept is derived in Section 3.1.4. Due to the high requirements of visualization at each level, multiple levels cannot be visualized simultaneously. In this regard, the reuse of the Experiment View at all levels has the main advantage that a visual linking among the process levels is provided.

To exemplify the applicability and scalability of the visualization to real-world data, the visualization concepts are shown throughout the section by two data sets from different sources. They are very different in scale. The first data set, which is referred to as *dry-lab data*, was generated from discrete-event based simulation. The underlying model has been used for testing purposes by modeling and simulation researchers in the research training school. Its data characteristics and the data generating context are equivalent to those described in the context of process levels discussed in Section 2.3. Since the visualization concepts have also been used to show data generated in the laboratory, the visualization will be exemplified on such an experimental data set. It is referred to as *wet-lab data*. As it was not developed in a simulation

process, an experiment description is not provided and the data does not contain information about events. Nevertheless, the visualization concepts are suitable to show experimental data in the context of a model. The two data sets are compared in Table 3.1.

	dry-lab data	wet-lab data
model level	15 proteins	8 proteins
	17 reactions	13 reactions
experiment level	4 experiments	3 experiments
multi-run level	10 runs	3 replications
single-run level	approx. $15,000$ time points	12 time points

Table 3.1: Comparison of example data sets *dry-lab data* and *wet-lab data*.

Further, the time series data associated to the state variables has different characteristics. In the *dry-lab data* set, values of state variables represent numbers of molecules in the system. Among the variables, these value ranges vary to a large extent, with a global value range from 0 to $98,000$. The *wet-lab data*, on the other hand, provides relative concentrations of molecules. Over time, all concentrations have been normalized against the concentration at the first time point. Hence, the initial concentration at the first time point always has a value of 1. Remaining concentrations have values from almost 0 to 3.81.

The visualization concepts introduced in this section have been published in [US09]. The main idea, the visual exploration of data in the context of data generation, has so far been rarely addressed in the literature. One of the few examples of such an application specific visualization was introduced by Matkovic et al. [MJJ+05].

3.1.2 Experiment View

The goal of the Experiment View is to visually combine model structure, experiment description, and resulting multi-run simulation data. Therefore, the following data has to be brought together.

1. The model structure
   can be expressed as a graph with state variables as nodes and event types as directed edges. Edges are in fact hyper edges, as they can link more than two nodes. Consequently, the model structure is a hyper graph.

2. The experiment description
   contains initial states of state variables and reaction rates of event types. As both state

variables and event types are represented in the model structure, the experiment description can be considered as attribute values of nodes and edges. The experiment description is given as one attribute value per node and edge of the model structure.

3. Multi-run simulation data
   consists, for each run, of the temporal development of the state variables, which are given by the model, and the occurrences of events over time, which can be related to the event types contained in the model.

The visualization of the **model structure** is the starting point to develop the Experiment View, as both experiment description and multi-run simulation data are linked to it. Current simulation models for cell biological systems are usually small, consisting of a few dozen components. Hence, a node-link-layout is a good choice for visualizing the model structure. It allows the representation of the model structure in a very intuitive way.

Experiment description and **multi-run simulation data** are visualized within this model visualization by iconic representations. These icons are placed at the positions of corresponding nodes and edges. Node icons include the initial state of the variable and provide an overview of the state variable's temporal development in a time value plot. Here, data from multiple runs has to be taken into account. Moreover, challenges arise as the number of time points can be very high. Also, value ranges of state variables may vary to a large extent during simulation. These challenges have to be considered in the design of node layouts. Edge icons, representing event types, are designed as stylized arrows to indicate the direction of the edge. The reaction rate is encoded in the length of the edge icon. To prevent an overload of the basic view, the occurrences of events are excluded from the Experiment View.

The visualization concept of the Experiment View is shown in Figure 3.1. The three main components are the 2-D layout of the model structure, node icons, and edge icons, which are now discussed in more detail.

3.1.2.1 2-D Layout of Model Structure

The node-link layout of the model structure has to fulfill three requirements: It needs to be applicable to hyper graphs whose edges connect more than two nodes. Positions of node and edge icons should be provided and distributed over the whole available display space to gain as much room as possible for the iconic representations. Third, in the biological context, models are often given with additional information about sub-cellular locations. The *wet-lab data* set

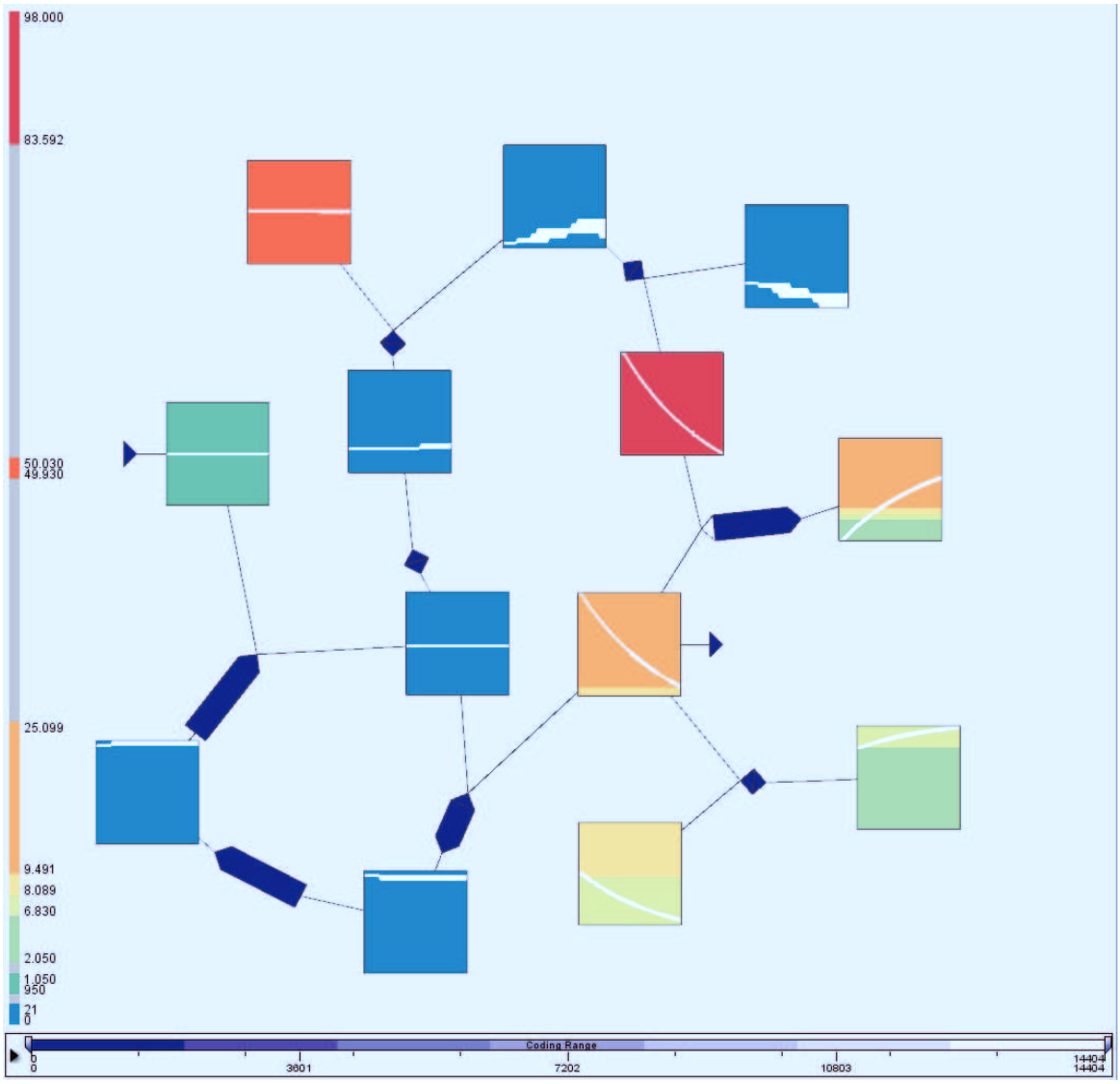


Figure 3.1: Experiment View, showing the *dry-lab data* set. Multi-run simulation data and experiment description are shown in icons that are positioned according to the 2-D layout of the model structure. The color scale for node icons is shown left, the color scale for edge icons is shown at the bottom.

contains such information. Each node is linked to a cell compartment such as the nucleus, the cytoplasm, or the membrane. These locations have to be communicated in the model layout.

Regarding the first two aspects, the force-directed layout [BETT99], a standard approach for relatively small graphs, is well suited. Hyper edges are considered during the computation of forces among nodes. The force-directed layout results in positions for nodes. Edge icons are positioned at the mid-point of linked nodes. In addition, special reactions exist that do

not fulfill the constraints of edges in a graph. They are only linked to either input or output nodes. To handle this aspect, one additional node is added for every edge to compute the layout. Enough space to place node and edge icons is ensured by adapting the forces used in the algorithm accordingly.

The third requirement, positioning of nodes based on sub-cellular locations, is not supported by the force-directed layouts. To this end, a second layout algorithm is adapted, which has been designed for models of cell biological systems and allows the inclusion of such constraints [BMGK08]. Node positions in the layout are restricted by related sub-cellular components of the chemical compounds, thus grouping nodes with similar locations in the cell in the resulting visualization. Similarly to the force-directed layout, the algorithm returns positions of nodes. Positions of edge icons are therefore again computed from the mid-points of linked nodes.

Both layout methods are heuristic methods that inherit randomness. Positions of nodes can therefore be manually adjusted by the user. Applying the two layouts to the data sets, the force directed layout has been used for the *dry-lab data* shown in Figure 3.1. As the *wet-lab data* provides sub-cellular locations of model components, the approach by [BMGK08] has been used for the layout (Figure 3.2).

3.1.2.2 Node Icons

In general, the information that is linked to a node comprises the experiment description and the resulting multi-run data. More specifically, for *dry-lab data*, an initial value of the associated state variable is given as the experiment description and the multi-run data consists of time series data for every run. This initial value is comprised in the time series data, as each run starts with the same initial value. For *wet-lab data*, an experiment description is not given. The multi-run data consists of one time series for every replication of the experiment. Main goals for visualization of multi-run data are, first, to show the state variable's **development over time** and, second, to enable a **global comparison of local values** among the state variables in the model.

For the given data, achieving both goals is challenging due to the limited size of the node icon on the one hand and the data characteristics on the other hand. With respect to the first goal of showing developments over time, this comprises the existence of multiple time series, which are given for both data sets. Further, the *dry-lab data* consists of a high number of time points. While the *wet-lab data* is less challenging due to few time points, it nevertheless requires tailored concepts to show its development for multiple runs.

Global comparison, the second goal, is made difficult for *dry-lab data* by very heterogeneous

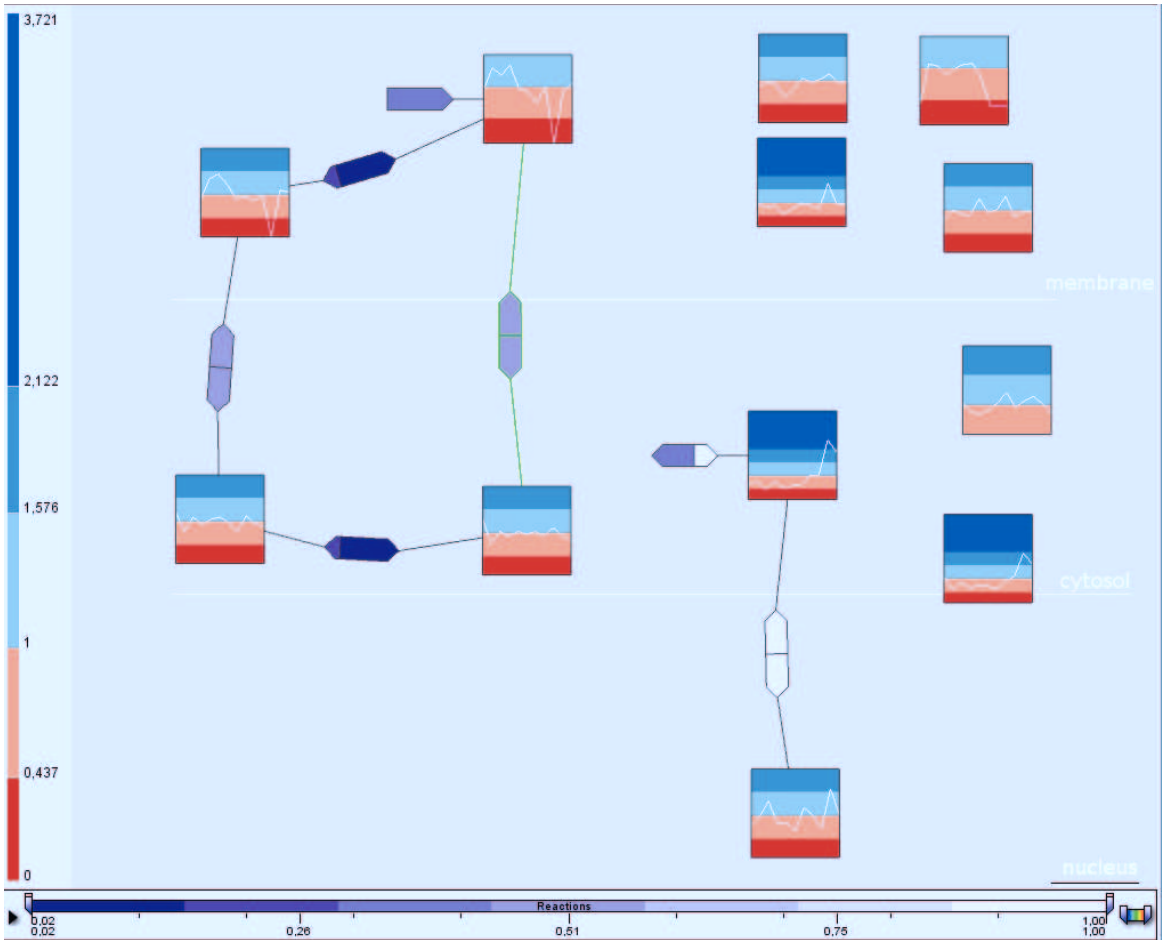


Figure 3.2: Experiment View, showing the *dry-lab data* set with sub-cellular compartments.

value ranges. Local value ranges of the variables cover differently sized parts of the global value range, which contains discrete values from 0 to 98,000. In contrast, the global value range in the *wet-lab data* comprises floating point values from almost 0 to 3.81. However, the nature of relative values has to be acknowledged in the visualization. All values have been normalized previously with respect to the first time point. For the visualization, it is therefore important to communicate whether values over time are below or above the initial value.

At first, it will be discussed how the **development over time** is visualized in the node icon.

The first main aspect is to cope with multiple runs, which cannot be shown simultaneously in the small node icon. Instead, data is aggregated over runs to give an impression of the variables' development over time. Possible aggregation measures include mean and standard deviation, median, or minimum and maximum. In general, the visualization of these aggregates over time demands for an explicit representation of the dimension time in the node icon. Hence, the dimension time is mapped on the x-axis of the node icon, which is a representation commonly

accepted in the application domain. However, the icon is limited in width and therefore in the number of time points that can be shown at once.

Referring to the high number of time points given for the *dry-lab data*, a down-sampling of the time series is necessary. Assuming a width of 50 pixels of the node icon, at most 50 values can be used in the visualization to represent the time series consisting of 15,000 time points. Sampling the data at certain time points results in information loss, as intermediate time points are neglected. Deriving sample data that comprises information about time intervals rather than single time points is an alternative. To compute mean, standard deviation, or median for time intervals, the aggregation of values in the time interval is required in addition to aggregation over runs, which, again, results in information loss about actually appearing values in the data. Value ranges, given by minimum and maximum values, are the only aggregates that provide a reliable overview on temporal trends without information loss even for multiple runs and a high number of time points. Moreover, value ranges communicate well the extent of deviation among runs and provide a general impression whether the variable shows a consistent increase or decrease, or a rather inconsistent behavior throughout runs.

For *wet-lab data*, the situation is different. All time points can be visualized directly in the node icon. Hence, mean, standard deviation, or median, derived for the time points, are suitable as measures to aggregate runs. However, the limited size of the icon permits to show multiple aggregates simultaneously. The user is thus given the ability to switch interactively among aggregation measures.

While the mapping of time on the horizontal axis of the node icon has already been motivated, a suitable representation of values over time is still needed. The visualization of value ranges over time, which are used to characterize *dry-lab data*, requires showing two data values, minimum and maximum, for each time point. In this regard, a time value plot is well suited, which maps the minimum and maximum over time on the vertical axis. This is a common representation in the application domain and can be used to easily identify whether attribute values decrease or increase over time.

However, the visualization of aggregated data in every node icon has to be set into a global context, according to the second main goal of **global comparison of local values**. As the local value range only covers a part of the global range, showing the global value range in the limited height of the icon would impede the identification of local temporal developments. To this end, a local value range is shown in the icon, which is set into a global context.

The communication of this global context is a general problem in visualization. To handle it, a new color mapping is introduced, based on local value ranges that appear in the data. It

results in a segmented global color scale. The local value range is set into a global context by coloring each node according to the color scale's segments that comprise the local value range. Aggregated data values are shown in a clearly distinguishable color, either showing value ranges of time intervals by a filled area between minimum and maximum over time, or by showing one of the other aggregates as lines over time. An example of a node icon is shown in Figure 3.3.

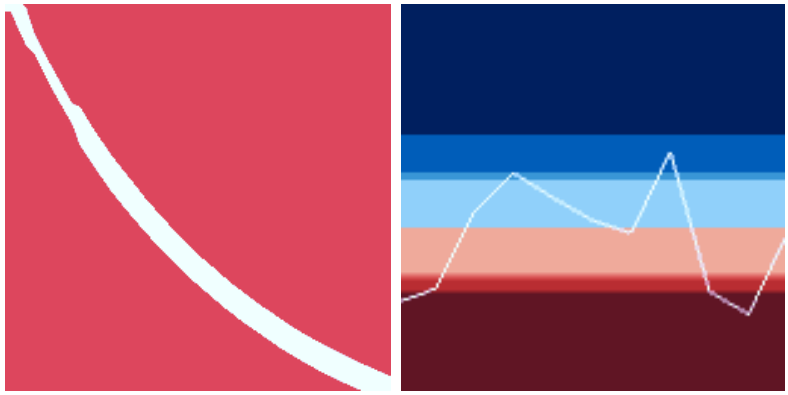
The color mapping goes beyond the communication of the global context of the local value range. It further eases the direct comparison of values over time, mapped onto vertical position, among different icons. Minima and maxima of the local data axes correspond to the values mapped to the borders of the color scale's segments. Hence, these minima and maxima are taken from a limited set of values. In the result, same values in different node icons are mapped to equal vertical position if the node icons comprise the same segments of the color scale.

Also, the color mapping accounts for both sequential and diverging color scales, which are both required with respect to the data (sequential for *dry-lab data* and diverging for *wet-lab data*). As an additional important advantage, it communicates uncovered value ranges (shown in gray on the left side in Figure 3.1). Further, the number of colors needed in the color scale does not depend on the number of value ranges in the data. Instead, the values ranges are mapped onto a predefined number of colors. Hence, a well-designed color scale can be used that contains clearly distinguishable colors. As the color mapping technique to take all aspects into account is quite complex and provides a new solution to a common problem of visualization, it is described in the separate Section 3.1.2.4, which follows the description of Edge Icons in Section 3.1.2.3.

3.1.2.3 Edge Icons

In addition to visualizing edge-related experiment description and multi-run simulation data, the edge icon needs to convey model information: the direction of the edge. Reactions, and thus edges, are usually directed. Moreover, reverse reactions can appear. They affect the same chemical compounds, but in opposite directions. In the model, these reverse reactions are commonly represented as two separate event types. To communicate the existence of such reverse reactions, they are visualized within one edge icon. Hence, an edge icon either includes a single reaction or two reverse reactions.

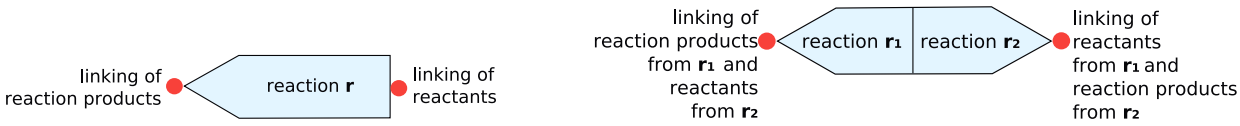
Consequently, an edge icon is designed as a stylized arrow to indicate the direction of the edge. Icons representing single-sided reactions have an arrow in one direction. Icons representing reverse reactions have arrows in both directions. Links to nodes that represent reactants and reaction products emanate from the ends of the arrows as shown in Figure 3.4. In addition,



(a) One node of *dry-lab data*. (b) One node of *wet-lab data*.

Figure 3.3: Design of node icon. An overview on the time series data is given. From the background color, the subrange of the global color scale can be identified (see Figure 3.1 and Figure 3.2). Due to different characteristics of the data, the two data sets require different concepts to give an impression of the temporal data as well as different color scales. For *dry-lab data* as shown in (a), simulation data, is shown by the range of values over time and a sequential color scale is used. For *wet-lab data*, as shown in (b), the experimental data is shown by mean over time. A diverging color sale is used, which clearly separates values below and above the normalized value at the initial time point.

the orientation of the icon is adapted according to the average positions of input and output nodes to head towards them.



(a) single-sided reaction (b) two-sided reaction

Figure 3.4: General design for icons of event types. The size of the arrows excluding the arrowhead encodes the reaction rate, the color encodes the average number of event occurrences for multiple runs.

Beyond this model-related information, edge icons should include the experiment description and an impression on the multi-run simulation data. From the experiment description, reaction rates are linked to events. To give an instant impression of reaction rates included in an experiment, they are encoded in the length of the stylized arrows. The visibility of all edges even for very small reaction rates is ensured by a fixed size for arrowheads, which is independent

from the reaction rate.

From multi-run simulation data, related data are occurrences of the event over time in every run. To avoid an overload of information in the Experiment View, the occurrences of events are not visualized over time, although this information can reveal inconsistencies in the data. As a general overview on the event-related data, the edge icons encode the average number of events occurring in the runs of one experiment. Therefore, a segmented global color scale is used. The color of the class representing the number of occurrences is encoded in the arrow representing the event. To visually distinguish between value ranges of nodes and edges, separate color scales are used.

3.1.2.4 New Color Mapping Concept for Global Comparison of Value Ranges

The global comparison of heterogeneous local value ranges is a known challenge in visualization. This challenge is faced when temporal developments of state variables are compared in a global context. In this section, the design of a global color scale is explained, which is used to compare local value ranges shown in the node icons as described in Section 3.1.2.2.

The concept is based on the color coding approach presented in [TFS08]. The authors address the problem of task-driven color coding and introduce an approach for the global comparison of value ranges. To this end, a segmented color scale is used. The idea of the approach is to map the segments of the color scale as closely as possible to the value ranges in the data. A necessary first step is to handle potentially overlapping intervals, as the same data values should be unambiguously mapped to one color. Hence, the value ranges in the data are transformed into non-overlapping, sequential intervals (compare to Figure 3.5, Step 1 and 2). Thus, data intervals are subdivided that share values with other data intervals.

To map these data intervals to segments in a color scale, the following approach is used in [TFS08]: The global color scale is derived from partial color scales with distinct hues, which are sequentially combined. Brightness and saturation are used to distinguish the intervals within one partial color scale. For the transition between two partial color scales, equal brightness and saturation are chosen for both intervals at the shared boundary.

With this encoding, value ranges in the data are exactly matched to segments in the color scale. However, the approach has some disadvantages for the application at hand:

- The global value range is subdivided with a very fine granularity. Every value range covers a unique range of color segments. For the intended use within node icons, where the comprised value range corresponds to the borders of the color scale's segments, similar

value ranges cannot be directly compared because they are not mapped to the same color segments.

- The number of color segments is equal to the number of non-overlapping data intervals. This can result in a high number of color segments. The problem arises to derive a color scale with an equally high number of colors that are visually distinguishable. This problem is not solved in [TFS08].

- The color scale covers the complete global value range. Parts of the value range that do not comprise data values are not communicated.

- The approach does not account for diverging color scales. In the *wet-lab data*, values are relative to a base value. For this data set, a diverging color scale is required that clearly separates values below and above the base value.

In the following, a new approach is introduced that can handle all these aspects. The basic idea is to merge non-overlapping data intervals into a predefined number of intervals, which can then be mapped to a predetermined, well-designed color scale with clearly distinguishable colors. The aggregation of similar value ranges into the same data interval results in a better comparison of similar values. While accounting for uncovered value ranges during the merging step, those are communicated in the resulting color scale. Further, if required by the underlying data, the approach is applicable to derive a diverging color scale for a given base value.

The approach complements the approach by [TFS08] by an additional step to merge non-overlapping intervals into a predefined number of intervals. Figure 3.5 illustrates the steps: the gathering of sequential intervals from the value ranges, the merging of these intervals, and the mapping of merged intervals to the segments of the color scale.

The suitability of the approach strongly depends on the appropriate merging of the data intervals (Step 3 in Figure 3.5). A heuristic approach to solve this problem is introduced in the following. At first, the general approach is introduced. Then, it is described how it is adapted to account for uncovered value ranges and a base value that implicates a diverging color scale.

The input to apply the heuristic comprises the intended number of intervals, which can be mapped to the segments of a color scale, and the sequential, non-overlapping data intervals. The goal of applying the heuristic is to get intervals that comprise similar numbers of input intervals and similar sizes of the value ranges as the second criterion.

A step-wise bottom-up approach is used to merge intervals until the number of intervals remains that matches the number of color segments in the predefined color scale. In every

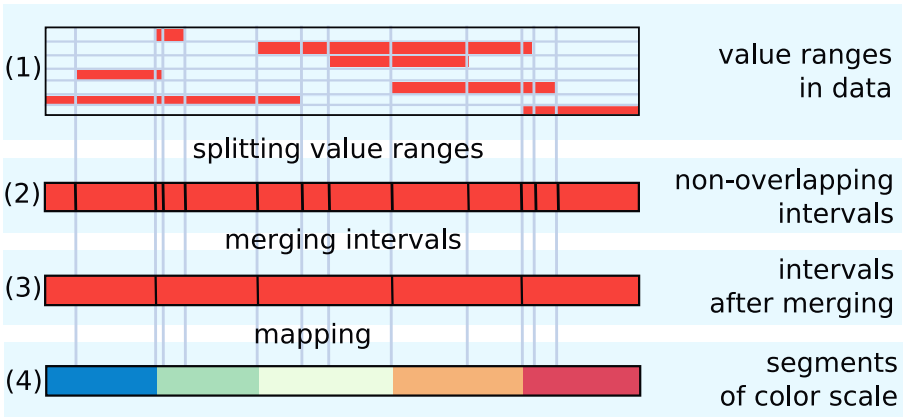


Figure 3.5: Steps to create the segmented color scale from the value ranges of the data. The possibly overlapping value ranges in the data (1) are split into sequential intervals (2). These intervals are merged until a predefined number of intervals remains (3), which can be mapped to the segments of the predefined color scale (4).

step, two neighboring data intervals I_i and I_{i+1} are merged into one new interval, based on the number of intervals aggregated into I_i and I_{i+1} in preceding steps and the value range enclosed by I_i and I_{i+1}.

To select the pair of neighboring intervals I_i and I_{i+1} that are merged, a list is used that contains all pairs of neighboring intervals as well as the current values of the two merging characteristics. It should be noted that each interval may be contained in at most two pairs of intervals. In every step, the interval pairs are sorted in the list according to the following two criteria:

1. minor sum of aggregated intervals within interval pair
2. smaller value range enclosed by interval pair

Then, the first pair of the list is merged and the list is updated. To this end, the newly merged interval is removed from the beginning of the list. The interval pairs that contain either I_i or I_{i+1} are updated by replacing the old interval with the newly merged interval (I_i, I_{i+1}). Both characteristics important for the merging step are recomputed and the interval pairs are repositioned in the list. The process ends when the number of intervals is reduced to the intended number of intervals.

With this approach, the size of the resulting data intervals solely depends on the above described heuristic. Intervals are primarily merged based on the number of intervals aggregated into them. Multiple local value ranges covering the same part of the global value range result in

a fine granularity of the color scale in this area. Segments of the color scale might cover small data intervals, in comparison to the global value range. In this case, segments of the color scale are devoted to distinguish small value ranges, which might not reflect significant disparities in the data. For an overview on value ranges appearing in the data, it may be appropriate to merge very small intervals to have more segments available to communicate larger value ranges more precisely. The decision about appropriate minimum sizes of data intervals cannot be done automatically. As the user will often have a feeling about such minimum sizes, an optional parameter is introduced, which can be interactively set. Throughout the merging process, it serves as a threshold to aggregate very small value ranges regardless of the number of merged intervals. This allows for a better adaptation of the resulting intervals to the characteristics of the data.

This basic concept is extended to account for two requirements:

1. Uncovered value ranges
   Parts of the global value range, which are not covered by local value ranges

2. Base value
   The comparison of local value ranges is done with respect to a base value, which separates the global value range in values below and above

Uncovered value ranges So far, the segments of the color scale have been determined from non-overlapping data intervals, regardless whether value ranges are actually contained in the data. That way, parts that cover value ranges actually existing in the data have been equally treated as parts that are not covered by existing data values. But considering such uncovered value ranges in the computation of the segments has main advantages. First, all segments of the color scale can be used to cover value ranges that appear in the data. No colors have to be reserved to cover value ranges not represented in the data. Further, the value range covered by one segment is not unnecessarily broadened to cover empty parts of the global value range. More precise segments are available for those parts of the global value range that actually include data. Moreover, the explicit communication of uncovered parts in the global value range helps the user to recognize which data values are present in the data. To account for uncovered value ranges, the idea is to maintain large uncovered value ranges during the merging of intervals. However, maintaining all uncovered value ranges, even if they are small, might lead to an inappropriate high number of small intervals, which cannot be merged because a small interval between them does not cover data values. To this end, an additional condition is introduced that applies if the first pair of intervals (I_i, I_{i+1}) in the list is empty:

- (I_i, I_{i+1}) are merged if the enclosed value range is smaller than the value range of the next pair of non-empty intervals (I_j, I_{j+1}) in the list
- (I_j, I_{j+1}) are merged otherwise

The condition provides a compromise to maintain large uncovered value ranges, but to merge small uncovered value ranges with data intervals.

Base value Depending on the current goal of analysis, it might be desired to discriminate whether values lie above or below a value of special interest. Besides the choice of an appropriate diverging color scale, accounting for this base value in the visualization requires adapting the computation of the data segments to this condition. To this end, the base value is used to divide the list of interval pairs into two: one list contains all interval pairs whose data values are below the base value, the other list contains all interval pairs with higher values. The interval pair (I_i, I_{i+1}) that encloses the base value is split into $(I_i, I_{basevalue})$, assigned to the first list, and $(I_{basevalue}, I_{i+1})$, assigned to the second list. For each of the two lists, the merging of intervals is performed until in each list, the number of segments is reached which can be mapped to the according part of the diverging color scale.

This concludes the new approach to derive a color scale for global comparison of local value ranges. At last, the application of this approach to the two data sets is discussed.

For *dry-lab data*, a sequential color scale is used that has been designed to show the segments with similar brightness to avoid unintended accentuations in the data (Figure 3.1). Considering the large global value range, data intervals with a size less than 25 should not appear in the color scale as they are not meaningful for a first overview on the data. For the mapping of *dry-lab data*, with its large global value range, to the segments of the color scale, a minimum interval size of 25 has been used. Within the global value range, large parts are not covered by data values. The effect of accounting for uncovered value ranges as described can be seen in Figure 3.1, where the existence of large uncovered value ranges, shown in gray, within the global value range become immediately apparent from the color scale shown at the left. Hence, actually occurring data ranges are encoded more precisely. With the resulting color scale, similarities among value ranges in the data can be assessed independent from similarities of colors.

For the *wet-lab data*, a diverging color scale is used, which has been computed with the constraint of a base value. The two colors separate the value range very clearly in value below and above the base value, as shown in Figure 3.2.

3.1.3 Visualization at Model Level

The main visualization goal at the Model Level is the comparison of multiple experiments. To this end, the use of an image based comparison of multiple experiments has been motivated in the introduction of this section. Hence, multiple Experiment Views are shown at once, one Experiment View for one experiment.

To fulfill the requirements of an image based comparison, the Experiment Views have to be visually linked: Similar aspects of the experiments have to shown equally in all images, and deviations in experiments have to be represented by deviations among the images accordingly. As the underlying model structure is equivalent in all experiments, the same layout for the model is used in all Experiment Views. Node and edge icons are placed at the same positions in all Experiment Views. Further, similar values in the experiment description and in the multi-run data need to be shown similarly.

Referring to the experiment description, the length of arrows in edge icons, which indicate reaction rates, have to be computed on a global basis over all experiments. The experiment description that is linked to nodes is comprised in the visualization of multi-run simulation data in the node icon. Hence, this problem is handled if the global comparison of multi-run data is achieved.

To this end, the two color scales used in the Experiment View have to be expanded to cover multi-run data of all experiments. One color scale shows the local value ranges of multi-run data linked to node icons. It is derived based on the approach introduced in Section 3.1.2.4, which computes the segments of the pre-defined color scale for an arbitrary number of local value ranges. Thus, the approach is applicable to derive a color scale that inherits all value ranges in all experiments. The other color scale encodes the average number of event occurrences for each edge, which can be easily expanded to cover the value range spanning all experiments.

In the result, all aspects of one Experiment View – the model structure, the experiment description, and the multi-run data – are visually linked among multiple Experiment Views. Deviations among the views thus indicate, as intended, deviations in the experiment description and the simulation data. An example is shown in Figure 3.6.

The approach supports the comparison of experiments by multiple facets, given by model structure, experiment description, and the multi-run data. Specifically, a global comparison of simulation data is supported among state variables and experiments. However, the quantity of simultaneously presented information makes a comparison of specific aspects tedious. For example, the comparison of values for a single variable among multiple experiments requires the identification of the variable in each Experiment View. Another example is the identification

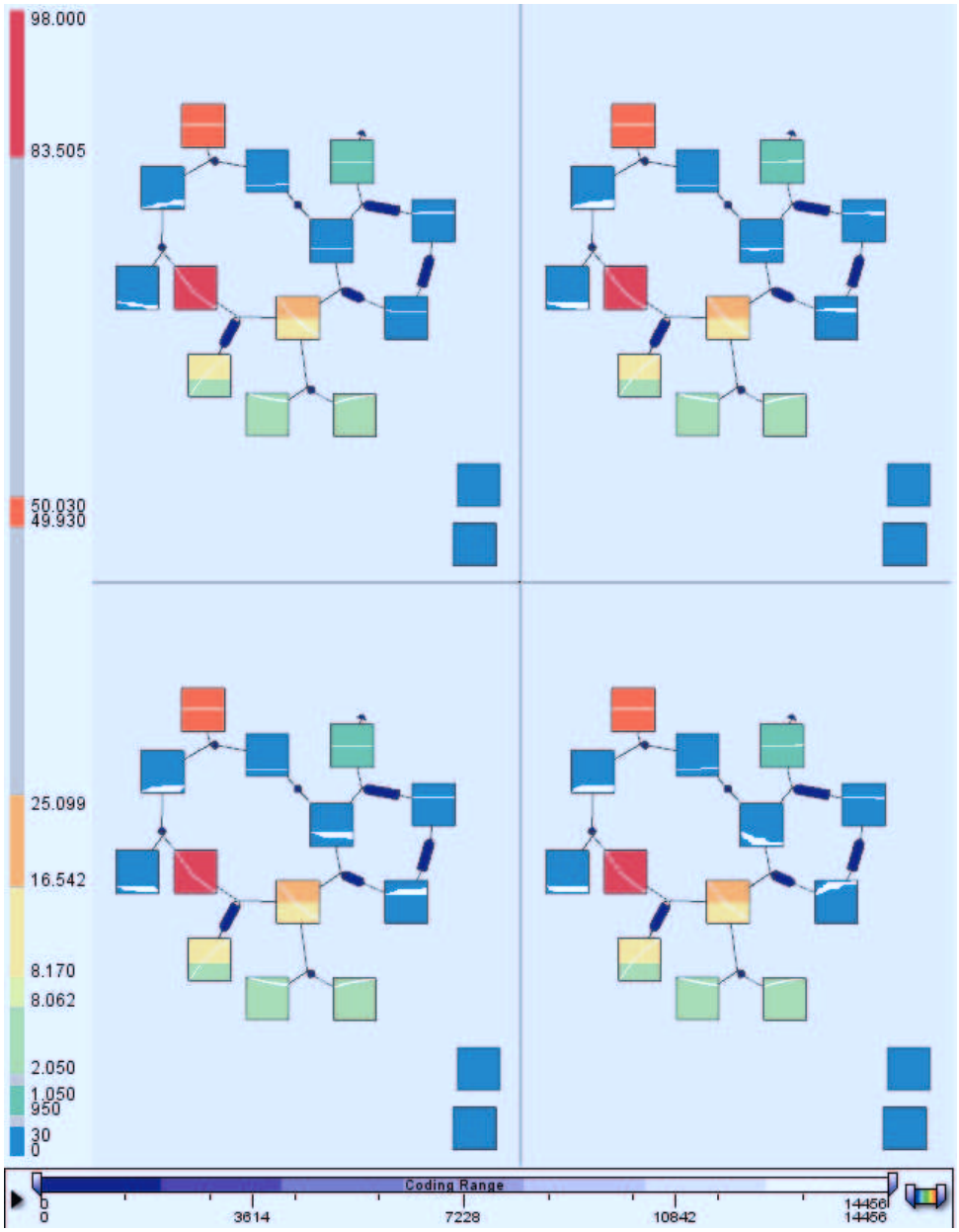


Figure 3.6: Visualization for comparison of experiments, exemplified for *dry-lab data*. For every experiment, one Experiment View is shown. Components of Experiment Views are visually linked. The layout of the model structure is reused. The mapping of reaction rates to the length of edge icons is done in a global context. Multi-run data linked to node icons is shown in the same global color scale derived over all variables in all experiments (at the left). Similarly for edge-related multi-run data, one second global scale (at the bottom) encodes the average number of event occurrences.

of experiments that are based on the same initial value of one state variable.

To provide better support for these tasks in the visualization, a visual highlighting of related information on demand is discussed in the following. In general, relations in the data are given by the following aspects:

- model structure: similar or structurally related state variables or event types
- experiment description: similar initial values of state variables or reaction rates of event types
- multi-run data: similar multi-run data, which can be characterized by similar value ranges or similar developments over time

All these similarities are explicitly represented in either the data or the data generating context and can therefore be automatically extracted. The visual highlighting of related information needs to be done with respect to the current objectives of the user. This includes the selection of a piece of information from the visualization as a point of interest and the selection of a similarity criterion to determine the information related to this point of interest. Regardless of the applied similarity criterion, the point of interest can always be selected from the set of state variables and event types, as all information is related to the components of the model. As the last step, the resulting region of interest, containing state variables and event types, needs to be highlighted in all views.

Thus, the visual highlighting of related information comprises three steps:

1. The user interactively selects a point of interest from one Experiment View
2. A region of interest is automatically defined, based on the point of interest and a similarity criterion.
3. The visualization is adapted to visually emphasize the region of interest.

In the following, the automated definition of a region of interest and its subsequent visual highlighting are described for one very important relation in the data: structural relations in the model structure, which result in a region of interest that comprises the same structurally related model components from all experiments.

Automatic selection of region of interest For a given a point of interest – a state variable or an event type –, the region of interest is defined based on a similarity criterion. In the following, the definition of the region of interest is derived for the example of structural relatedness, which is given by the model structure.

Relations given by the model structure cannot only be used to select the same model component in all experiments, but to further extract structurally related components. For the user, this can be important as it does not only allow the comparison of one state variable among experiments, but, for example, of all state variables that take part in the same reaction.

To this end, the region of interest is defined by two relations: For a selected state variable or event type as the point of interest, all equivalent state variables or event types from other experiments are included in the region of interest. Further, components of the model structure are also added to the region of interest according to their structural proximity to the point of interest.

This structural proximity is a measure whose value is user-defined. It is defined at a granularity that allows distinguishing between state variables and event types. For the minimum structural distance 0, only the point of interest and the same structural component in other Experiments Views are selected. Given a state variable as a point of interest, a structural distance of 1 adds all event types linked with the state variable to the region of interest.

Visual highlighting of region of interest The visualization needs to be adapted to highlight the region of interest, which can comprise one or multiple state variables or event types among experiments. In general, the state variables and event types that are contained in the region of interest are visually highlighted by coloring the border of the icons and underlying them with a shadow.

For the specific example of using structural relations as the similarity criterion, an additional visual distortion supports the magnification of the region of interest for a more detailed visualization of the related multi-run data, while the context is preserved. To maintain the visual linking among Experiment Views, a coordinated distortion is used, resulting in the same distorted layout of the model structure in all experiments. The distortion is driven by the characteristics of the visualized data. As the information to be analyzed is carried by the icons, it is important to magnify them without distortion. Also, structural dependencies among magnified icons have to be maintained. To comply with these object-related requirements, an object-based approach is used as described in [SS02b]. The region of interest is uniformly magnified, while remaining parts of the display are reduced in size and distorted. In Figure 3.7, an example of

the local distortion applied to all Experiment Views is shown. Changes in the layout induced by the interactive selection are communicated by animations. This enables the user to maintain the overview on structural relations in the data when comparing experiments.

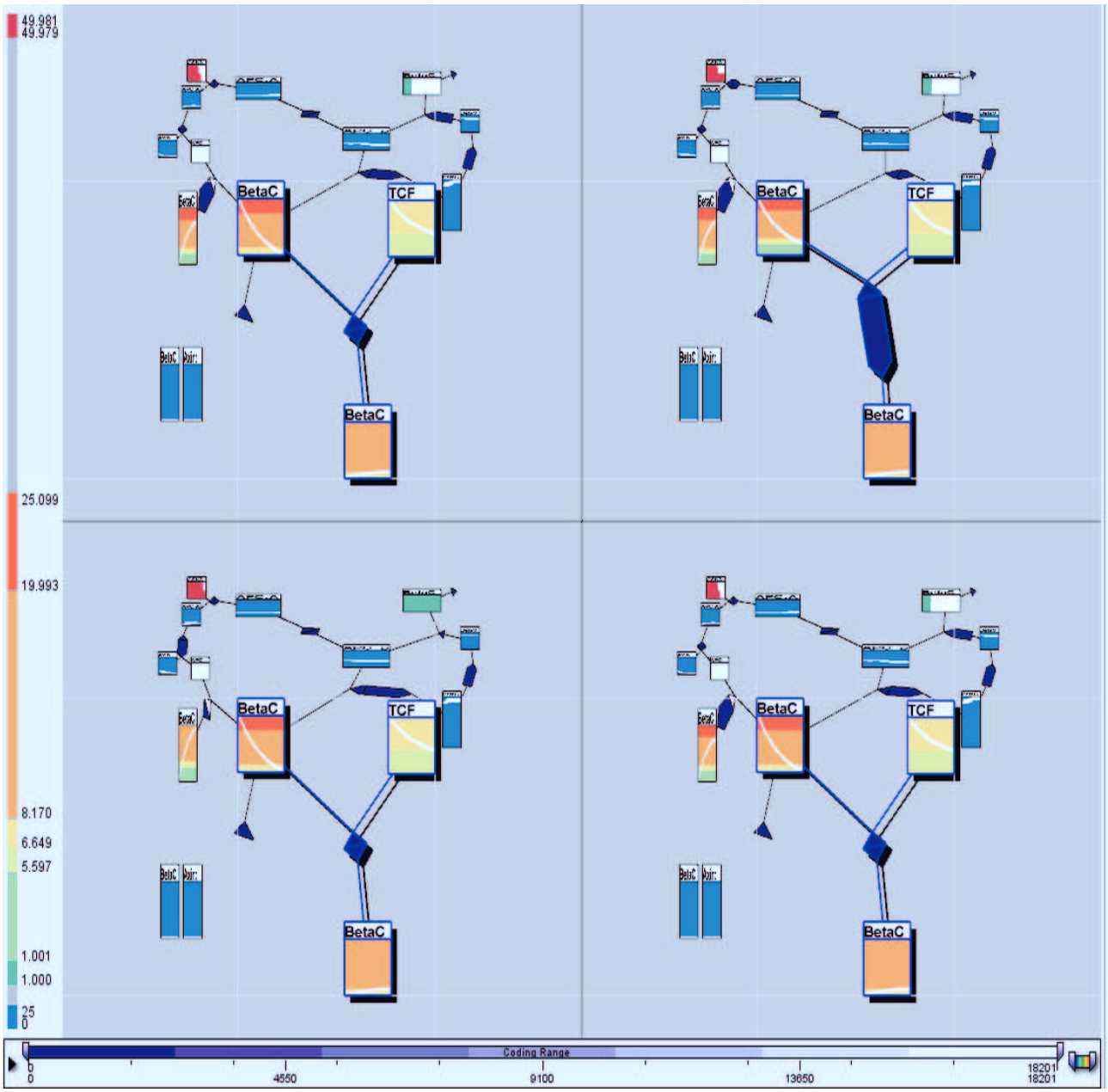


Figure 3.7: Visualization for comparison of experiments, exemplified for *dry-lab data*. For every experiment, one Experiment View is shown. The screen shot includes the object-based distortion based on a region of interest. (The screen shot has been taken with a comparable simulation data set, which explains deviating value ranges compared to Figure 3.6)

It should be noted that the distortion of the visualization is not suitable for arbitrary regions of interest. In order to be consistent with the requirement of a visual linking among Experiment Views, it is only applicable if the same model components are selected among experiments.

3.1.4 Visualization at Multi-Run Level

The Experiment View provides a strongly aggregated view on the multi-run data of an experiment. At the third process level, the multi-run level, a more detailed evaluation of the simulation data is required. In the analysis, especially the temporal developments of the data are of interest and are therefore in the focus of this section. For one experiment, time series data is given for multiple runs, and within each run for each state variable and each event type. Due to this complexity, the whole time series data cannot be adequately visualized at once.

This results in two basic requirements for the visualization at the multi-run level: First, appropriate visual representations for the interactive selection of individual time series are necessary. Second, these selected time series need to be visualized in detail. To this end, a multiple view approach is used to comprise these requirements.

With respect to the first requirement, the Experiment View is valuable as it contains all state variables and event types in the context of the simulation process and provides an overview on the multi-run data – two aspects that can be important to identify state variables and event types of interest. However, functionality to select individual runs is not provided. To this end, a separate view is needed.

Regarding the second requirement, the visualization of time series data, data of state variables and event types needs to be distinguished. For state variables, a scalar value is given for each time point during a run that represents the number or concentration of proteins at that time point. Simulation data related to event types, on the other hand, captures the occurrences of this event type over time. Hence, separate views to visualize the data over time are used for state variables and event types.

All in all, four views are required to provide the required selection of time series data and the detailed visualization:

- Experiment View
  Overview on experiment and selection of state variables and event types

- Multi-Run View
  Selection of single runs of the experiment

- State View
  For the selected run, analysis of the time dependent simulation data related to state variables

- Event View
  For the selected run, analysis of the time dependent simulation data related to event types

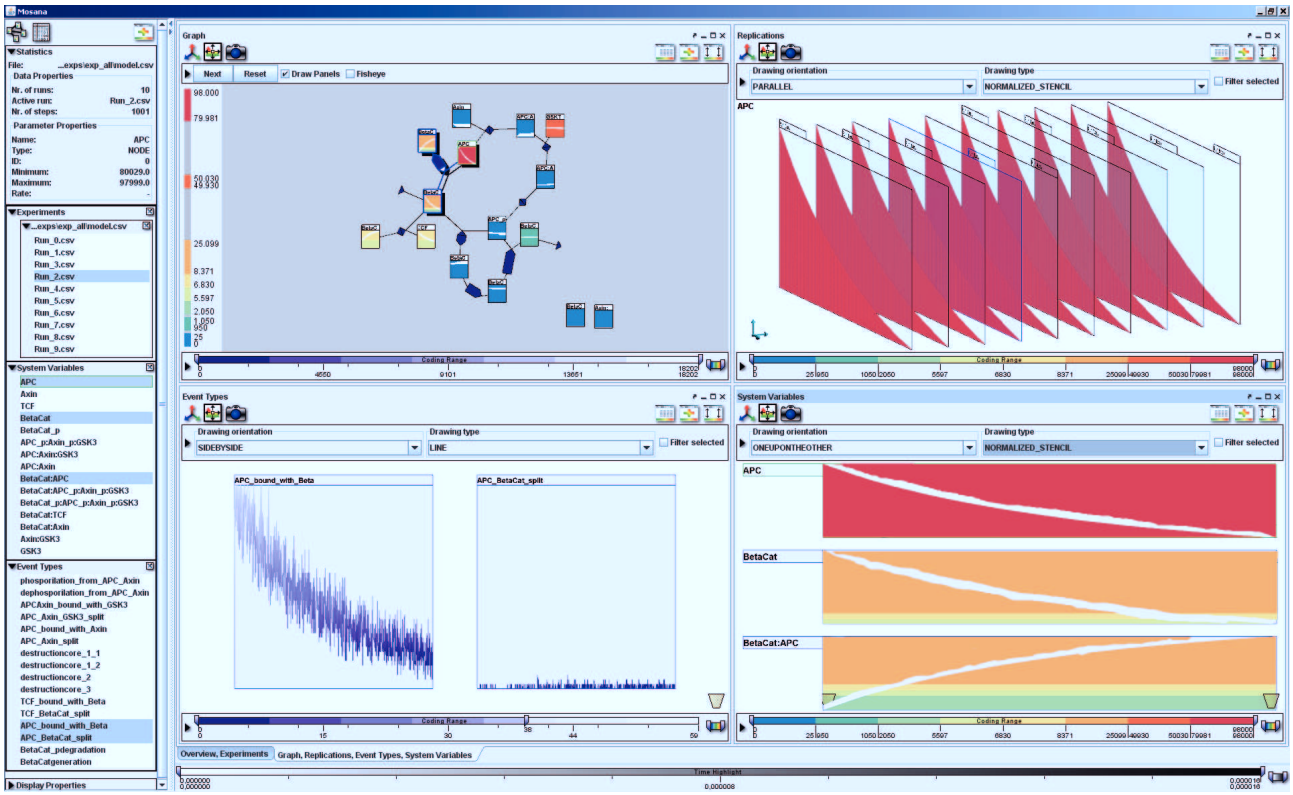


Figure 3.8: Visualization for the detailed exploration of one experiment, shown for the example of *dry-lab data*. The *Experiment View* (top left) is coordinated with the *Multi-Run View* (top right) for the selection of one run, the *Event View* (bottom left) that shows the single-run data connected to event types, and the *State View* (bottom right) for the detailed analysis of single-run data of state variables. The data of the icons highlighted by a shadow are shown in the State and Event Views. The icon highlighted by green borders is the state variable shown in the Multi-Run View.

The Experiment View serves as the visual component from which state variables and event types are chosen. Selecting one node icon results in one state variable, selecting one edge icon results in one or two event types, as reverse reaction are encoded in the same edge

icon. To enable the comparison of state variables, up to three node icons can be selected, which corresponds to the maximum number of nodes that are linked to one edge. Thus, all simulation data for a reaction can be analyzed in detail, which is considered as a typical goal of the analysis. For event types, only one edge icon can be selected, as it already involves the comparison of two event types in most cases. The limited number of possible selections restricts the number of time series, which are compared in either State View or Event View.

The Multi-Run View provides the functionality to select one run from the set of runs for detailed analysis. Thus, one visual representation is needed for each run in this view. In order to avoid the introduction of another visualization technique in addition to Experiment View and time visualizations in State and Event View, the visual representation in the Multi-Run View is reused from other views. To this end, the time series visualization in State and Event View is used. While these views visualize time series data for one or multiple state variables or event types, the Multi-Run View has to include all runs. Hence, univariate, or at most bivariate, time series data of every run is shown in the view. One or two variables – either a state variable or one or two event types (depending on the edge icon) – is selected from the Experiment View. The data is shown differently for states and events, according to the encoding used in the State View and Event View, respectively.

For the detailed visualization of state variables or events over time, the large number of time points per run has to be acknowledged, as it appears in the *dry-lab data* (approximately 15.000 time points). This has to be considered for both the State and Event View. They are discussed separately in the following, as the underlying data has different characteristics.

The State View has to visualize scalar values of state variables over time. A time value plot [Wil05] is well suited to convey temporal developments of scalar values and is widely accepted in the application domain. However, the time value plot view cannot visualize all time points in detail, because the number of time points, as provided with the *dry-lab data*, is too high. The maximum number of time points that can be shown is equivalent to the number of pixels available for the temporal axis. This problem similarly arose for the visualization of temporal data in node icons. However, only a single run needs to be considered in the State View. With respect to the discussion in Section 3.1.2, the visualization of temporal data in the State View is supported as follows. The time points that are mapped to one pixel on the temporal axis are aggregated in one interval and presented by both the minimum and maximum values. To compare the up to three state variables that are selected from the Experiment View,

heterogeneous value ranges have to be considered. Thus, each variable is shown in its own time value plot. The local value range is mapped on the vertical axis of the time value plot. To communicate the local value ranges in a global context, the approach derived for node icons in the Experiment View is reused: The value range mapped to the vertical axis corresponds to the segments of the global color scale. By reusing the color scale from the Experiment View, a visual linking of time value plots in the State View with the node icons in the Experiment View is given. The time value plots are arranged on top of each other, to reserve as much space as possible for the temporal axis.

The Edge View displays the occurrences of events over time for a single run and one or two event types. As the number of time points is too high to be shown at once, the number of occurrences is computed for equidistant time intervals. Hence, scalar values are obtained from the occurrences of events over time. To visualize them, time value plots are also used in the Event View. The number of intervals depends on the pixels that are available to show data over time. For each event type, one time value plot is used. The values comprised at the vertical axis of the time value plot range from 0 to the maximum number of occurrences from the up to two event types. Arranging the time value plots side by side enables the direct comparison of occurrences over time. To visualize the event data, the mapping of values to color from the Experiment View cannot be reused, because a different aggregation of the data is used. Instead, a local color scale is used according to the value ranges in the Event View. To provide a visual linking, the basic hue of the color scale in the Experiment View is also used for time series data in the Event View.

As described, the time points cannot be shown in full detail in State View and Event View. To provide a more detailed visualization of the temporal aspects, the visible range of time points shown in time value plots can be adjusted interactively. This is done globally for Multi-Run View, State View, and Event View, to explore the temporal aspects of the data in multiple facets.

The resulting multiple view framework is highly interactive, which demands for visual feedback about current selections. To this end, selected state variables and event types have to be highlighted in the Experiment View. The state variable or the event types, which are displayed in the Multi-Run View, need to be distinguishable from states in the State View and events in the Event View. Further, the selected run has to be highlighted in the Multi-Run View. In Figure 3.8, selected icons in the Experiment View are highlighted by adding a shadow to the icons. The icon selected for the Multi-Run View is additionally highlighted by applying a

different color to its borders. In the Multi-Run View, the time value plot that corresponds to the selected run is highlighted by its border color.

3.1.5 Discussion

Focusing on the visualization of simulation data as provided by the concepts from Section 3.1, the concepts enable that the temporal developments of multiple variables are visualized simultaneously and that the global comparison of their heterogeneous value ranges is supported by a new global color scale. Hence, the visualization concepts provide application specific solutions for open challenges in visualization. Moreover, the analysis of the data is supported at different abstraction levels, which arise from the process levels given by the application context. Multivariate, time dependent data containing multiple runs is compared for different experiments at the Model Level or analyzed for one experiment at the Experiment Level. At the Multi-Run Level, the data is analyzed for the individual runs of an experiment. Thus, the visualization concepts provide significant contributions for the visualization of multiple sets of multivariate, time dependent data with heterogeneous value ranges.

Nevertheless, the main value of the visualization concepts derived in Section 3.1 goes far beyond the capabilities of visualizing simulation data: They enable the exploration of the whole simulation process for one model. The explicit visual representation of both the data generating context and the simulation data allows the exploration of the process in two main directions: First, the data generating context covering the whole simulation process can be visually explored to analyze its impact on the resulting model behavior, which is captured by the simulation data. Second, the simulation data can be visually explored for notable characteristics, which can then lead back to the context of data generation. The tight visual integration of both aspects allows the user to switch the direction of the exploration at any time during the analysis sequence.

In addition to the behavior of the model, also the factors that cause the behavior are immediately apparent in the visualization. Moreover, with specific visualization concepts for each of the three process levels, multiple visualization goals are supported. Preserving the visual linkage among the process levels by the reuse of a common view, a seamless transition among the process levels is provided, although the visualizations are too complex to be shown at once.

These visualization concepts have been developed with respect to data sets that appeared in the research training school **dIEM oSiRiS**. The underlying models are relatively small, and limited numbers of both experiments and runs are present. Further, the focus on the integration of data generating context and data led to the disregard of certain aspects in the underlying

portions of information, which have not been relevant for the visualization goals at the process levels.

To visualize individual aspects comprised in the simulation process by all their facets and for larger quantities of information, supplementary visualization concepts are necessary. In the remainder of this chapter, two aspects are regarded: the visualization of a larger model, which is the foundation of the simulation process, and the detailed visualization of multiple runs of one experiment.

Focusing on model characteristics rather than associated simulation data, a new visualization is introduced in Section 3.2 that supports the exploration of models comprising thousands of proteins and reactions by multiple facets, including structural relations and dozens of attributes.

First concepts towards a visual multi-run analysis, which are scalable to high numbers of runs, are presented in Section 3.3.

3.2 Visual Exploration of Large Models

Under the perspective of exploring the facets of the model instead of exploring the simulation process, the characteristics of the model have to be visually explored rather than associated simulation data. These characteristics are given by relations among proteins and reactions and by a multitude of heterogeneous attributes that describe the proteins and reactions of a model. Models that have been regarded in the context of the simulation process allow the exploration of a functional unit of the cell, comprising a few dozens of proteins and reactions. However, a model of the cell can easily comprise thousands of proteins and reactions. Node-link layouts, as they have been proposed in Section 3.1 for small models, cannot be used to cope with such high numbers of nodes and edges. Moreover, they do not handle multiple heterogeneous attributes.

In this section, a new visualization is presented for large networks with thousands of nodes, edges, and dozens of attributes of nodes. It has been developed in joint work within the research training school *dIEM oSiRiS* and was published in [SJUS08]. Further details about underlying concepts of the technique are found in the doctoral thesis by Hans-Jörg Schulz [Sch10].

The basic idea is to represent the model as a bipartite graph and to visualize it with a table-based approach, which has the advantage of being familiar to data analysts. Bipartite graphs comprise two independent node sets – here proteins and reactions – with no adjacent nodes. In the literature, few visualizations have been proposed for bipartite graphs, including a node-link-visualization called *Anchored Maps* [Mis06] as well as solutions for movie-actor-networks in the InfoVis 2007 contest [KJKC07]. Table-based approaches have been used in domains like

visual analytics [SGL08] or for the exploration of transition graphs [PvW08].

In the visualization, proteins and reactions are shown in two individual tables. Each row in one table shows one protein and each row in the other table shows a reaction. In each table, attributes are shown in columns. Lines between the two tables show structural relations between proteins and reactions. Edge weights are mapped onto the width of the individual edges. Additionally, 1-mode projections are computed and depicted as arcs at the sides of the tables. A projection is established between two nodes of one set if they are both linked to the same node in the other set. Thus, projections are basically additional edge sets that connect only nodes from one node set and can be seen as shortcuts to show dependencies between nodes of the same set. The overall layout of the described tables, edges, and arcs is shown in Figure 3.9.

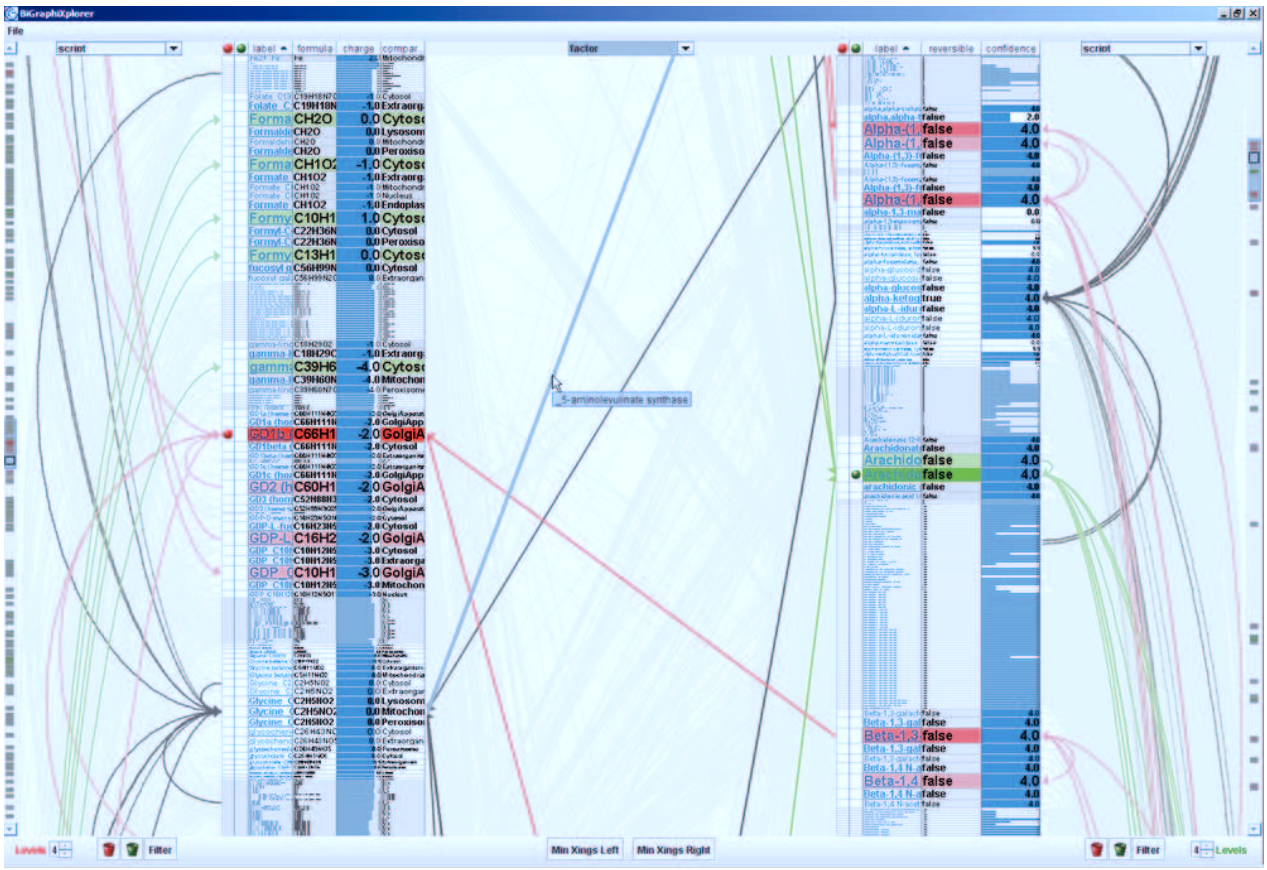


Figure 3.9: The basic visualization concept with the two node sets as separate tables with integrated focus and context, connecting edges in between, and 1-mode projections at the sides. Further, concepts for node selections are shown: Each table contains columns for the selection of focus nodes. Selected rows, traversed edges, and 1-mode projections are highlighted. Selection markers in fish eye scrollbars at the left and right show locations of selected nodes.

Although being scalable to thousands of proteins and reactions and to numerous attributes, large and dense models demand additional features to minimize visual cluttering and enhance the accessibility of the representation. Large node sets result in very long tables, which do not fit the screen height. Large edge sets result in a lot of lines running in between the two tables, producing a lot of edge crossings. Dozens of attributes, which might consist of textual descriptions or even images and figures, result in very wide tables with dozens of columns. In large models, the ability to derive subsets from the entire data set is important to identify functional units and further analyze them.

To handle large node sets, a focus+context technique like the table lens [RC94] allows larger parts of the table to be shown at once and therefore reduces efforts for browsing. To cope with a cluttered visualization of lines between the tables, two methods are integrated in the visualization. The first is a barycentric crossing minimization heuristic [JM97], which reduces the number of edge crossings. It can be applied to reorder either of the two tables. The second is the highlighting of hovered edges, combined with interactive functionality to carry the user directly to connected nodes outside the screen. To cope with dozens of attributes, a separation is carried out between primary, relevant attributes shown within the table, and secondary, supplementary attributes. These secondary attributes are outsourced to HTML pages, which can be retrieved on demand.

At last, the exploration by selections is regarded. In the model, relations among its components can result from various aspects, including structural relatedness given by edges, projections, paths through multiple nodes, or relations in attribute space, for example by sub-cellular locations or derived structural attributes like in-degree or out-degree. To enable a simple and user-controlled interaction while accounting for all these aspects, the selection process is split into an interactive and an automatic part. In a first step, a set of focus nodes is interactively selected by the user, which is passed, in the second step, to a selection script for automatic selection.

The integration of the proposed selection concept into the table-based visualization requires two steps: the interactive selection of nodes as well as the visual highlighting of selections. For the interactive selection of nodes, single nodes or entire intervals are interactively chosen. To highlight selected nodes, the respective rows are enlarged in height and colored with respect to a degree of interest, which has been assigned to nodes during selection. Further, highlighting paths along which the script traverses by coloring the edges is very helpful for understanding the inner workings of selection scripts. Resulting selections can span over both tables and scattered all over them. To easily find regions with selected rows in large tables, selection

markers are added within the scrollbars at the sides of the visualization. They carry directly to the respective row. In addition, tables can be reduced to show only selected nodes for a detailed exploration of these functional units.

The presented visualization technique supports the interactive exploration of models of biochemical reactions that contain thousands of proteins and reactions and dozens of attributes. Displaying proteins and reactions in separate tables offers the combination of structural information and multiple attributes in one display. The combination of interactive and automated selection of nodes enables the identification of functional subunits in the model.

3.3 Visual Multi-Run Analysis

For the development of visualization concepts in Section 3.1, the importance of multi-run analysis has been acknowledged, but was not in the focus of visualization. The presented visualization supports the representation of statistical properties of all runs of an experiment. This is one strategy to cope with multi-run data. For a more elaborate analysis, especially if larger numbers of runs are generated, the identification and comparison of possible behaviors of the simulated model is required. In this regard, it is useful to group runs with similar behavior, which can then be analyzed and compared.

3.3.1 General Considerations

Grouping runs with similar behavior involves two challenges. First, runs need to be grouped based on their similarity, resulting in subsets of the data. Second, these subsets need to analyzed and compared visually. Both challenges – generation and analysis of subsets – frequently appear in data analysis and are not limited to the analysis of multi-run simulation data. In the following, they are discussed with specific regard on this application.

Regarding the first challenge, the generation of subsets, two alternative approaches can be followed. Either, subsets are automatically generated using an underlying computational method, or the subsets are interactively generated by the user. Automatic methods require an explicit definition of similarity among runs. Here, especially the temporal developments are of interest. In a diploma thesis [Krü08], first approaches towards a visual multi-run analysis have been presented. Various definitions of similarity among runs in a univariate context have been discussed, including local measures that compare values at certain time points, and global measures, which take the complete time series into account. The criteria for similarity can be manifold. For example, the average distance between time series can be used, which is a

commonly used similarity measure for time series data [VWVS99], but also in other applications. In addition, similarity can be defined from the increase or decrease of values over time or from the number of values that fall into a certain geometric area. Alternatives further include the classification of runs based on example curves over time. In the diploma thesis, a hierarchical clustering approach was proposed as the basis to explore subsets of runs on multiple abstraction levels.

An important result of the thesis is that an appropriate similarity measure strongly depend on the user objectives, which need to be described explicitly. Here, a close cooperation with users is necessary to find measures that suit their understanding of similarity in the data. This might include similarity measures that take more than one variable into account. As the results generated by clustering are hard to predict from the choice of a similarity measure, this will also require the comparison of clustering results based on different similarity measures. Further research in this direction is required to find solutions that fully match the user's requirements.

The interactive generation of subsets within a visualization system provides a valuable alternative. Similarity of data objects is usually represented by the spatial proximity of their representation in the visualization. By brushing in the visualization, similarity among data objects can be defined very intuitively. Multiple criteria of similarity can be combined, which might be difficult to express by an explicit similarity measure. Different meanings of similarity can be explored by interactive redefinition of subsets in various views.

The second challenge is the visualization of subsets, which should enable the user to analyze and compare simulation runs. To this end, the temporal context of the data is of interest. The data dimension time is therefore a main point of visualization concepts for visual multi-run analysis. From discussions with users it can be stated that they demand the analysis of subsets by statistical properties. These statistical properties include *mean*, *standard deviation*, *median*, and the range of values, given by *minimum* and *maximum* values. While mean or median encapsulate information about the general behavior of a data set, standard deviation and the range of values communicate the distribution of values. However, the concrete visualization concepts depend on the method of subset generation, which influences the resulting representation of data subsets. Further, providing information about how subsets were generated complies with the goal to visualize data in the context of data generation.

In the following, an approach for interactive subset generation is introduced in Section 3.3.2. The concept, which has been derived in cooperation with the VRVis Vienna in Austria, has been published in [UMDS08]. Specific challenges arise from the ability to interactively define subsets with smooth brushing [DH02]. This introduces a continuous transition from focus to

context in the definition of the subset. This feature has to be taken into account for both the computation of statistical properties of the subsets as well as in their visualization. The visualization of statistical properties of subsets is developed within the multiple view concept of the SimVis system, which provides views for the interactive selection and refinement subsets and, thereby, the context on how the subsets are obtained.

3.3.2 Visualizing Statistical Properties of Smoothly Brushed Subsets

The interactive generation and analysis of data subsets requires powerful visualization tools, which support an interactive selection of portions of the data according to relationships in the data the user is currently interested in. The SimVis system [DMG+05, Dol04] is one example for such a visualization tool. It provides multiple coordinated views for the analysis of large multivariate and time dependent data sets, with multiple data items for each time point. The information visualization views that are integrated in SimVis enable the selection of subsets by a multitude of relationships, including similarities in time series data [MKO+08], which is specifically relevant for the generation of subsets of runs.

However, with the given views in the system, the visual analysis of subsets is not adequately supported. The high number of values per time point hampers the identification of the subsets' properties. Hence, the analysis of subsets by their statistical properties is demanded by the user.

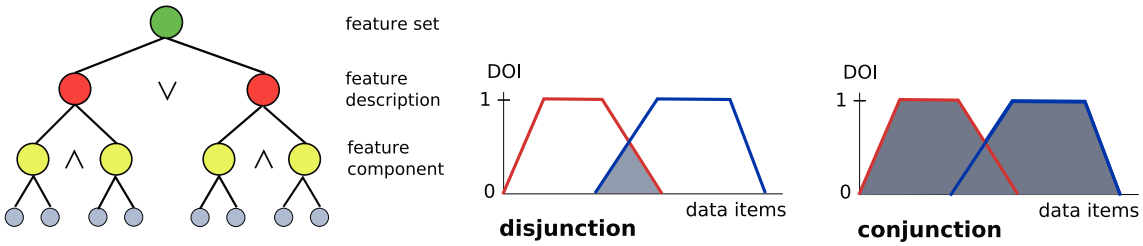
To this end, a new view to analyze statistical properties of interactively generated data subsets has been developed as part of the multiple view system SimVis. The linkage to other views has several benefits. The other views provide the exploration of the data in various contexts and brushing to interactively select subsets, which are then visualized in the new statistical view. After interactive refinement of subsets in other views, the effects of these refinements on statistical properties can be directly analyzed in the statistical view. Further, the statistical view shows an abstraction of the subset; the other views visualize detailed information about the items of the subset. By linking the view showing statistical properties with other views for detailed visualization and interactive definition of the subsets, the context from which the subsets have been derived is provided alongside the resulting statistical properties.

For the definition and visualization of statistical properties, the concept of smooth brushing [DH02] as part of SimVis plays an important role. Smooth brushing is a focus + context concept for interactive selection, which introduces a continuous transition from data in focus to context data. Taking this continuous transition into account is beneficial because the visualization of statistical properties can lead to additional insights about the local behavior within a subset.

Smooth brushing affects the formal description of data subsets, as described in Section 3.3.2.1. Thus, its impact has to be considered for both the specification (Section 3.3.2.2) and visualization (Section 3.3.2.3) of statistical properties. The visualization concept comprises the visual representation of single subsets with an emphasis on the parameter time and the interactive visual analysis of multiple subsets, in order to analyze variances among subsets.

3.3.2.1 Definition of Smoothly Brushed Data Subsets

The description of data subsets in the SimVis system is based on the feature definition language (FDL) tree [DMG+05]. Every node in the FDL tree is specified by a time dependent degree of interest (DoI) function that is assigned to the data set. The leaves of the tree are user-defined by smooth brushing, one leaf is generated for every brush. DOI values in the continuous range $[0, 1]$ indicate the smooth transition from data in the focus (with $DoI = 1$) to context data ($DoI = 0$). Nodes on higher levels in the FDL tree (Figure 3.10(a)) are composed by either conjunction or disjunction (Figure 3.10(b)) of their children's DoI functions. All brushes within one view are composed into one node on the level of feature components. DoI functions defined in different views are logically combined in feature descriptions (by conjunction) and feature descriptions (disjunction).



(a) Hierarchy levels of FDL tree

(b) Logical combination of two DoI functions (red and blue), assigned to a list of data items, into one DoI function

Figure 3.10: Feature Definition Language tree (a) and composition of one Degree of Interest function from two functions (b).

Data subsets are closely connected to the nodes of the FDL tree. For each node, the corresponding data subset consists of all data items whose DoI value is non-zero. Since the DoI function is time dependent, the data subset may be variable over time.

3.3.2.2 Specification of Statistical Properties

This section is related to the derivation of the statistical measures mean, standard deviation, and the range of values for data subsets with respect to the effects of smooth brushing. Due to smooth brushing, variable degrees of interest in the continuous interval [0, 1] occur within a subset. Therefore, a modified derivation of the statistical measures is proposed in the following.

For mean and standard deviation, the individual data items are weighted by their degrees of interest. That way, the impact of a data item on the resulting statistical value corresponds to the interest the user has assigned to the item. For a data subset consisting of the data items i, mean m and standard deviation d can be derived from weighting the values $value_i$ of a variable by their degrees of interest doi_i, as in the following equations (compare to [BPC07]).

$$m = \frac{1}{sum_{doi_i} * sum_i} \sum^{i} (doi_i * value_i)$$

$$d = \sqrt{\frac{1}{sum_{doi_i} * sum_i} \sum^{i} (doi_i * value_i - m)^2}$$

Since the goal is to communicate the temporal developments of statistical properties, mean and standard deviation are derived for every point in time by considering all data items currently belonging to the data subset. Hence, for every time point, mean and standard deviation are expressed by one value.

To determine minimum and maximum values of a data subset, a different approach is needed since these values represent limiters of a data subset. They are not composed from a number of values. Hence, the range of values is derived for different degrees of interest. To this end, the data subset is further subdivided based on the data items' degrees of interest. Given a degree of interest doi, the subset S_{doi} consists of all items whose degree of interest is greater than or equal to doi. Minimum and maximum values for one attribute are then specified for every subset S_{doi}. Again, to account for temporal developments, these values are derived for every point in time separately.

3.3.2.3 Visualization of Statistical Properties

A suitable visual representation of the statistical properties, which have been derived in the previous Section 3.3.2.2, is subject of this section, with an emphasis on temporal developments. Further, approaches for visual comparison of subsets are discussed, in order to analyze variations among data subsets. With regard to multi-run analysis, comparison of subsets is an important analysis goal to identify possible outcomes of the simulation.

Visual Representation of Statistical Properties of One Subset In this section, the visualization for the statistical properties of one smoothly brushed subset is introduced with respect to the modified derivation of statistical values described in Section 3.3.2.2.

To communicate the temporal developments of statistical properties, the visualization is based on the parameter time. Relations among multiple variables are neglected to reduce visual clutter. The general visualization approach thus shows the time dependent statistical values for one attribute. Based on time value plots, time is mapped to the horizontal axis and the currently selected variable to the vertical axis. An axis scaling independent from the analyzed subset is achieved by scaling the axes according to the global minimum and maximum values of time and the current variable.

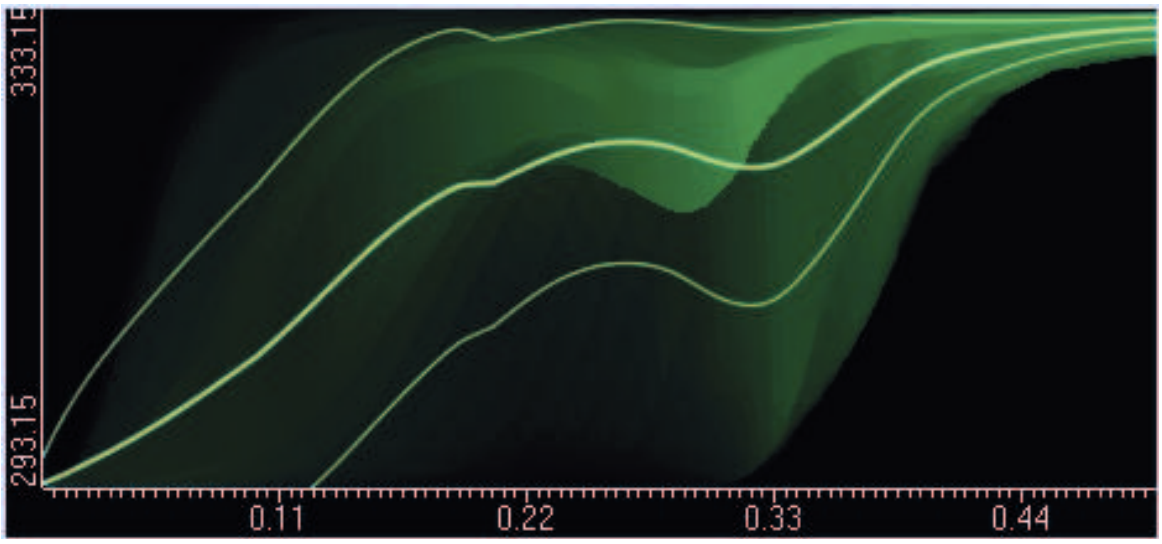


Figure 3.11: A subset's time dependent statistical properties of one variable. Mean and standard deviation are weighted by degree of interest, value ranges are shown for different levels of interest, encoded by the opacity of the surface.

An example of the visualization is shown in Figure 3.11. Mean and standard deviation, which are weighted by degree of interest, are encoded as lines by connecting the values of the time points in order to show temporal developments. The location of standard deviation below and above mean is adapted from error bars. To distinguish both characteristics, different line widths are used. While mean and standard deviation are represented by one value per time point, the range of values is present for different degrees of interest. Visualizing the value range as a continuous area over time offers an intuitive visual representation to encode range values for different degrees of interest. The degree of interest of a range of values is encoded by the

opacity of the area.

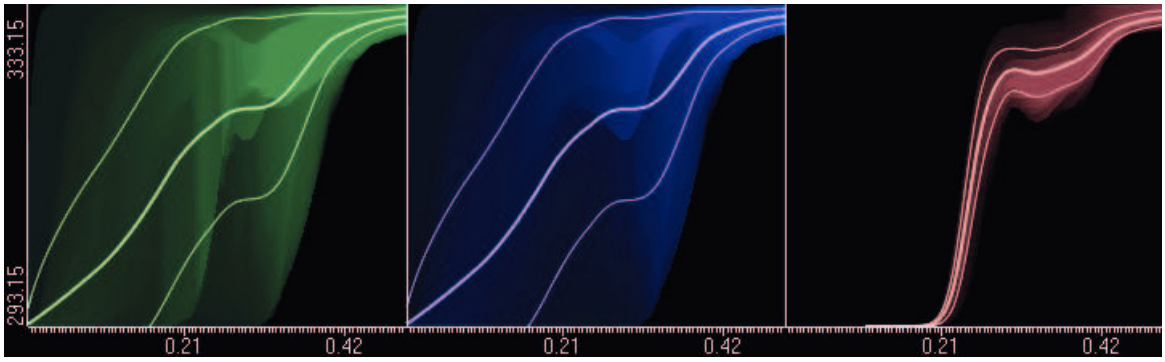
In Figure 3.11, the continuous DoI range [0,1] has been subdivided into discrete, equidistant levels. Due to implementation reasons, 256 levels are generated, resulting in 256 subsets and the same number of minimum and maximum values for each time point. This exceeds the number of opacity values the human eye can separate. Thus, the impression of a continuous transition between the value ranges for different levels of interest is created.

As additional features, the minimum DoI value for the visualization of statistical properties can be adjusted to analyze the subset for different degrees of interest in detail. Also, the visualization of the individual statistical characteristics can be switched on and off depending on the analysis task.

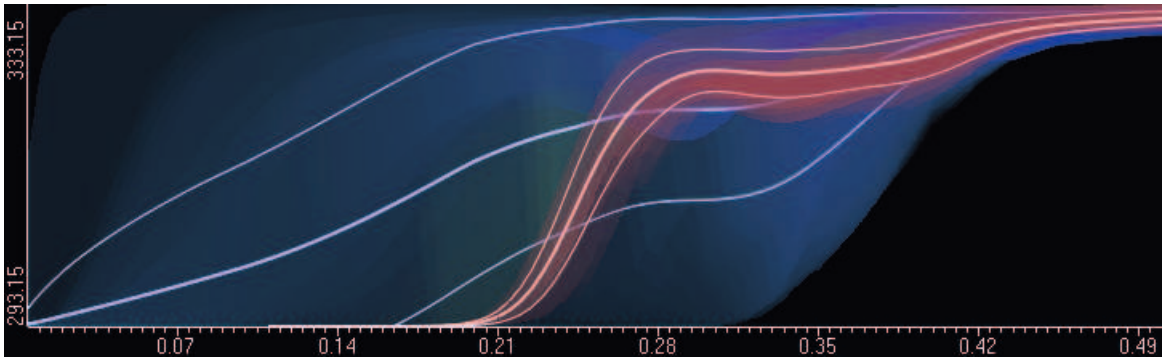
Interactive Visual Analysis of Multiple Subsets The visual representation of a single subset provides limited capabilities to analyze the variations within a data set. Thus, the visual comparison of multiple subsets is considered as an important task, which can be performed for a broad range of analysis goals. For example, it may be useful to compare subsets defined by single or combined brushes or to evaluate the influence of single subsets in a combination. In addition, the characteristics of subsets can also be compared to those of the whole data set. The unified description of all data subsets as nodes in the FDL tree (Section 3.3.2.1) allows the consideration of this broad range of analysis goals.

Given this flexibility for the definition of subsets, a data level approach for comparison [PP95], which combines multiple data subsets into one data representation, is not suitable. In addition, the subsets do not necessarily share the same data items. Thus, the idea of image level comparison is followed. Based on the visual representation introduced in Section 3.3.2.1, three general concepts are supported, which are related to different analysis goals. To give an overview on the general behavior, the data subsets can be arranged in separate images aligned side by side. Second, small deviations between subsets can be analyzed by overlaying the statistics of subsets. To cope with occlusions introduced by overlaying, an interactive lens is an alternative.

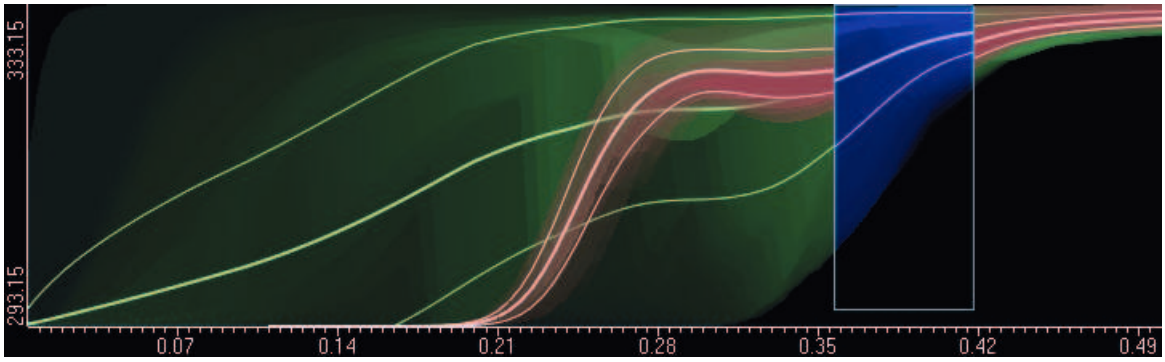
To compare data subsets arranged in separate images, all images are equally sized and based on the same axes scaling to provide an intuitive and reliable comparison among images. Avoiding occlusions, separate images provide a good overview on the general behavior, as shown in Figure 3.12(a). However, small deviations between subsets can hardly be discriminated and the limited display size of the statistical view has to be fractioned by the number of data subsets.



(a) One image for each data subset, aligned beside each other.



(b) Three data subsets (red, green and blue) superimposed.



(c) Two data subsets superimposed (green and red), the third subset (blue) shown in an interactive lens.

Figure 3.12: Visual representations to compare statistical properties of multiple subsets.

For a detailed comparison of subsets in the statistical view, the visual representations of subsets can be overlaid (Figure 3.12(b)). Compared to separate images, the approach enables to span the visual representations of all subsets over the available display size. However, the overlay of subsets is problematic if the visualization includes value ranges, encoded by closed surfaces. Using transparent value ranges, colors of overlaid subsets are mixed. Opaque value

ranges, on the other hand, result in occlusions of subsets in the background. An alternative representation of value ranges by lines produces visual clutter. By interactively adjusting the transparency of the value ranges and by a reordering of subsets from back to front, a mixing of colors becomes traceable for the user and occlusions are made visible. Further, subsets and single data characteristics can be interactively switched on and off to support the comparison of details in the data.

As an alternative for the analysis of small deviations, an interactive lens concept [BSP+93, SFB94] is proposed. As the general idea, one of the subsets to be compared is shown in a rectangular lens whose size and position can be interactively adjusted (Figure 3.12(c)). In the lens, only a fraction of the visual representation of the subset's statistical characteristics is shown, according to the position and size of the lens. The background of the lens is opaque to avoid a mixing of colors, which overcomes one drawback of the overlaid display. The interactive adjustment of the lens dimensions is an integral feature to perform the visual comparison: The user can sequentially explore the differences between data subsets by moving and resizing the lens interactively.

These concepts of visual comparison can be adapted to compare the statistics of multiple variables in the subsets, by comparing the resulting images or overlaying the visual representations in one view.

3.4 Summary

This chapter focused on the development of visualization concepts for the process levels model, experiment, and multi-run, with the goal to support the understanding of the simulation process by the inclusion of the data generating context along with the data. In this regard, visualization concepts for all three process levels have been introduced in Section 3.1.

As the foundation, a tight integration of model structure, experiment description, and multi-run simulation data within a single view has been introduced. It handles the challenges of multiple runs with high numbers of time points as well as very heterogeneous local value ranges in the simulation data. The multi-run data, given for many time points, is highly abstracted to communicate the general trends over time. For the global comparison of heterogeneous value ranges, a novel approach has been introduced to derive a global color scale based on value ranges that appear in the data. Moreover, it communicates large uncovered value ranges and accounts for both sequential and diverging color scales, depending on the data characteristics.

The fundamental concept has been adapted to all process levels, in order to cover the related

visualization goals. To compare multiple experiments at the Model Level, multiple views, each showing a single experiment, are visually linked. The global comparison of value ranges among variables and experiments, supported by this concept, is accompanied with additional interactive functionality for local comparison. Local regions of interest are derived by the structural similarity given in the model structure. At the multi-run level, a detailed analysis of the runs of one experiment is required while maintaining the context of data generation in the visualization. To this end, the basic view of one experiment is embedded within a multiple view concept. The data generating context as well as the selection of model components is provided by the basic view. Additional views support the selection of individual runs and detailed visualizations of temporal developments for an interactively selected small set of state variables or event types.

These concepts have been developed with respect to data sets that were generated within the research training school **dIEM oSiRiS**. The focus has been set on the integration of data and the context of data generation in order to explore the simulation process related to one model. As a consequence of necessary abstractions made about the individual aspects of information, a detailed exploration of certain parts of the simulation process requires additional visualizations.

On the one hand, this includes the model, the basis of the simulation process. The visualization in Section 3.1 omits many of its facets to visualize it as part of the data generating context. To explore the model in all its facets, comprising both structural aspects and multiple attributes, a table-based visualization has been presented in Section 3.2 as the result of joint research within the research training school. It is scalable to large models containing tens of thousands of proteins and reactions.

The second individual aspect of the simulation process is the visual multi-run analysis. It has been discussed in Section 3.3 based on the idea to generate and subsets of similar runs. A main challenge is the definition of similarity that is used to group runs. As a flexible concept to define similarity among runs according to current objectives of the user, an approach is presented that makes use of the interactive generation of subsets provided in the multiple view framework SimVis. Accounting for user demands, statistical properties are used to characterize these subsets in the visualization. The challenge that was solved is to compute and visualize the statistical properties with respect to smooth brushing, which introduces a continuous transition from focus to context.

In summary, the tight integration of visualization and the data generating context at all three levels allows the exploration of the complete simulation process rather than solely of the simulation data that results from it. To this end, the information comprised in the visualization

is abstracted according to the most prevalent visualization goals that are linked to the process levels. In addition, supplementary visualizations focusing on specific aspects, such as model analysis and multi-run analysis, are provided.

In contrast, at the level of single-run data, only a single data set is analyzed, which leaves room to visually depict additional details of the simulation data, such as heterogeneity and spatial context. Exemplary visualization techniques for complex single-run simulation data, which results from specific modeling and simulation approaches used in the application domain, are subject of the next Chapter 4.

Chapter 4

Visual Analysis of Complex Simulation Data

The second main visualization challenge in this work, in addition to the visual integration of data into the context of data generation, is the visual analysis of heterogeneous and complex data sets. This is the subject of this chapter.

In the application domain, simulation approaches based on discrete-event systems lead to heterogeneous data sets with event and states, given in a temporal and multivariate context. The consideration of spatial context in the simulation – the biological processes in cells depend on locations of proteins – adds additional complexity to the simulation data. Although visualization concepts have been described in the literature that deal with certain facets of the data, the complexity that appears in the application context is usually not handled. Hence, new visualization concepts are necessary to support the specific characteristics of simulation data. In the context of process levels, which have been introduced in Section 2.3, these visualization approaches complement the level of *single-run simulation data*.

Specifically, new visualization approaches are developed for data from two different modeling and simulation approaches, which have been employed in the research training school **dIEM oSiRiS**: *Next Sub-Volume Method* and *Attributed* Π-*Calculus*. The two approaches handle spatial context very differently. The Next Sub-Volume Method [EE04, RKDB06] subdivides the cell into sub-volumes by imposing a grid on the cell, which leads to reactions of proteins within sub-volumes and movements of proteins between them. The Attributed Π-Calculus [JLNU08, JLNUar], an extension of the established Π-Calculus [Pri95, PRSS01], is a new development within the research training school by Mathias John. Applied to the application domain, the ability of proteins to react with each other is restricted based on their attributes,

expressing, for example, location within the cell. The resulting simulation data consists of a time series of reaction networks.

Both visualizations have been developed in joint work within the research training school, in close cooperation between the fields of visualization and modeling and simulation. The visual analysis concept for the Next Sub-Volume Method, which is subject of Section 4.1, has been developed within a diploma thesis [Gut08] and published in [UGJS09]. The visualization concept for the Attributed Π-Calculus is introduced in Section 4.2.

4.1 Visual Analysis of the Next Sub-Volume Method

In this section, the visualization of simulation data from the Next Sub-Volume Method is in the focus. The Next Sub-Volume Method is a simulation algorithm for biochemical reaction networks. It incorporates spatial context by subdividing the cell into a grid of sub-volumes. Since the simulation algorithm produces heterogeneous and complex data sets, visualization plays a significant role in data analysis. A very important characteristic of the data is the spatial context. Here, many techniques from the field of volume visualization are available, most of them focusing on static and univariate volume data. However, simulation data from the Next Sub-Volume Method is time dependent and multivariate. Although some of the existing approaches deal with multivariate and time dependent volume data, visualizing data with these characteristics remains a challenging task. In addition, the simulation output is composed of two heterogeneous data types. Besides analyzing the simulation data by its states over time, the simulation makes it possible to track the process that leads to changes in the system state. This process is characterized by events, whose characteristics are different from states.

The goal of this section is to develop a visualization technique that supports the analysis of data derived from the Next Sub-Volume Method with its heterogeneous, time dependent, multivariate, and spatial characteristics. A single view cannot provide a solution for all those aspects. Thus, a visualization concept comprising multiple coordinated views is proposed, each covering certain parts of the data. A highly interactive user interface allows the adaptation of the visual representation according to the user's current questions about the data.

In order to find an appropriate combination of visualization concepts, a systematic approach is applied. First, a short overview on the Next Sub-Volume Method and the resulting simulation data is given in Section 4.1.1. To cope with the multi-faceted characteristics of the data, a classification of the data characteristics is presented in Section 4.1.2, which serves as the basis

to identify potential visualization concepts with respect to techniques described in the literature. From this classification, the visualization design is derived (Section 4.1.3). The multiple view framework that integrates the results of the design is introduced in Section 4.1.4.

4.1.1 Simulation Data from the Next Sub-Volume Method

Spatial stochastic simulation algorithms become increasingly important in the application domain. *Particle algorithms* [vZtW05, AB04] can trace the position of each single molecule and execute reactions with a given probability whenever two particles collide. In contrast, *lattice-based algorithms* subdivide space into volume elements with a homogeneous distribution of particles. The *Next Sub-volume Method* [EE04, RKDB06] is an example for lattice-based algorithms.

To simulate the interactions of biochemical species in a spatial and temporal context, a grid is imposed on the volume of the cell. The grid cells are called sub-volumes, to distinguish them from the biological term cell. Interactions of proteins within sub-volumes and movements between sub-volumes are described by events. These events take place with specific rates that depend on the state of the sub-volume, i.e. the current number of particles. The estimated time between two events is an exponentially distributed random number with mean equal to the reciprocal of the sum of reaction and diffusion rates. With high rate or diffusion constants or a large number of particles inside the system, the inter-event times can become very small, i.e. the time scale of the system might drop to the microsecond or nanosecond domain.

In the resulting simulation data, two data types can be distinguished: *state data* and *event data*. While state data describes the system state for each time point, event data refers to events that change the system state over time.

The state data is given by the number of particles in each sub-volume for each time point. As different species exist in the system, the number of particles has to be differentiated for each species. Thus, the data in each sub-volume is multivariate.

The event data consists of two types: reactions and diffusions. Reactions describe the interaction of multiple particles within one sub-volume. As a consequence, the number of particles is altered within the sub-volume. Reactions are characterized by the particles they involve, the 3-D location in the grid, and the time point when they occur. Diffusions describe the movement of a particle from one sub-volume to a neighboring sub-volume. Hence, diffusions are associated with a direction. Diffusions can involve any of the particles in the system whose diffusion constant is non-zero. They are defined by the involved particle, the two neighboring sub-volumes that are affected, and the time point. As both event types are related to a specific

subset of the multivariate particles, the events are multivariate as well.

For visual analysis of the simulation data, multivariate events and states need to be visualized in their spatial and temporal context.

4.1.2 Classification of Data Characteristics

A relevant aspect of the data is the spatial context. Thus, visualization methods for data in 3-D context are the starting point for a systematic identification of suitable visualization concepts. However, these approaches do not account for the heterogeneity and complexity of the data that is derived from the Next Sub-Volume Method.

During the visual analysis process, these aspects have to be supported as well. This includes the visualization of both states and events. They have very different characteristics, but are semantically related to each other. While state data represents the overall state of the system, event data is important to directly identify changes in the data. An important aspect of simulation data is always the temporal development. Nevertheless, also the data at a single time point may be relevant for visual analysis. Thus, the visualization needs to show data at single time points as well as developments over time. Further, both states and events are multivariate. Referring to state data, both the spatial value distribution of a single variable and the relations among multiple variables have to be explored. For event data, it may be of interest to evaluate general occurrences of events in spatial and temporal context. In addition, a distinction of events by related state variables may be demanded by the user.

This leads to six different facets of the spatial data. They can be categorized as three opposite pairs, which are summarized in Figure 4.1: states vs. events, static vs. dynamic context, and univariate vs. multivariate context. To apply a systematic approach for the identification of possible mappings of the data to a visual representation, each pair is discussed in addition to general concepts for spatial data, which are presented in Section 4.1.2.1. In Section 4.1.2.2, specific concepts for states and events are discussed. Section 4.1.2.3 deals with visualization of data in static and dynamic context. Finally, visualization in univariate and multivariate context is subject of Section 4.1.2.4. For all three pairs, the spatial context is specifically regarded.

4.1.2.1 3-D spatial data

The visualization of spatial data, given as one scalar value per point on a 3-D grid, has been extensively addressed in volume visualization (see [PB07] and [EHK+04] for overviews). Two main concepts are used for rendering: direct volume rendering and surface extraction. In

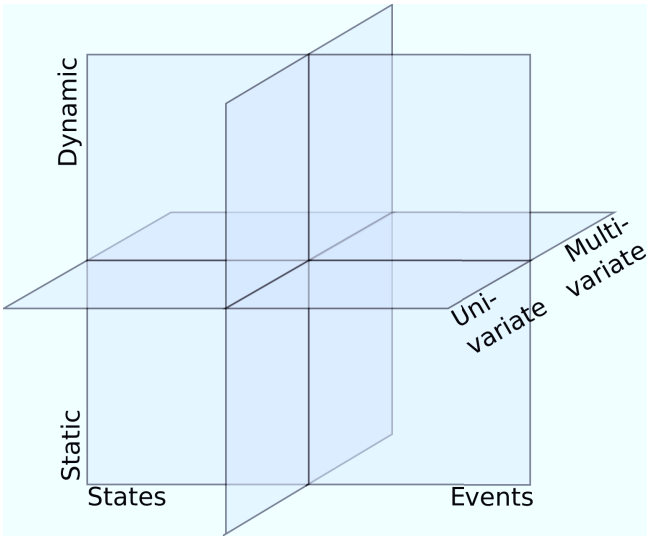


Figure 4.1: Classification of 3-D spatial data.

direct volume rendering, the visual representation is directly derived from the volume data, for example by ray casting, splatting, or texture slicing. A transfer function is used, which maps the scalar values in the grid to graphical attributes. For example, the transfer function can affect the color and transparency of the visual representation of the volume element. Different transfer functions are appropriate depending on the data, the application background, and the rendering method. Alternatively, surface extraction techniques use a threshold to derive an iso-surface, representing a constant value throughout the volume. These iso-surfaces are visualized. To improve the effectiveness of the visualization, illustrative approaches [Bru06] have been explored recently for volume visualization.

In general, basic volume rendering approaches are well-suited for univariate and static data. Visualizing scalar values given in a 3-D grid, they are designed to handle state data, but do not account for event data.

4.1.2.2 States vs. Events

The distinct characteristics of events and states need to be considered to find appropriate visualization concepts. State data consists of multivariate scalar values for each grid point over time. Events, on the other hand, are defined as occurrences at concrete time points and 3-D locations.

To visualize state data in 3-D space, volume visualization methods are well suited. They additionally need to include the dynamic and multivariate aspects, which are subject of discussion in the next two Sections 4.1.2.3 and 4.1.2.4.

Visualizing event data requires the highlighting of discrete points in time and space. A general taxonomy and survey of visualization approaches for events can be found in [CCC+05]. Focusing on spatial context of events, approaches are basically comparable to 3-D scatter plots (see, for example, [KSH04]), where axes represent spatial dimensions.

4.1.2.3 Static vs. Dynamic data

Visualization concepts for both static and dynamic data are required. Visualizing the static data at a single time point, the value distribution of state variables and the current event need to be shown in spatial, univariate, and multivariate context. The visualization of a single time point is therefore derived from appropriate concepts as discussed in Sections 4.1.2.2 and 4.1.2.4.

To show dynamics of the data, additional concepts are required. Aigner et al. [ABM+07] speak of two possible mappings for time dependent data:

- Mapping time on time
- Mapping time on space

Mapping of time on time refers to a visualization changing over time. When mapping time on space, multiple time points are visualized simultaneously.

The first approach, *mapping time on time*, is usually referred to as animation. The visualization changes over time according to the consecutive time points given in the data. The speed of animation does not necessarily comply with the time domain of the data, as the scale of the time domain might not be appropriate for the animation. In general, the user is able to sequentially explore the data in an intuitive way. Nevertheless, the data is not directly comparable for multiple time points.

Mapping time on space has the advantage that the development over time becomes visible at a glance. Time can be included either explicitly as a spatial dimension or implicitly by using visual comparison techniques. When time is mapped on a spatial dimension in the visual representation, at most two other spatial dimensions are available to display other aspects of the data. Hence, 3-D data cannot be shown in combination with all time points.

Maintaining 3-D context, visual comparison techniques allow the simultaneous representation of data from multiple discrete time points. They are subdivided into image based comparison and data based comparison [PP95]. Using image based comparison, multiple visual representations of the same type are arranged on the display, each including a different time point. It therefore uses up more space on the display than the visualization of a single time point.

Data based comparison techniques combine data from multiple time points into a single visual representation. One example implementation is a difference image [POM+09] from two time points, where only the values that have changed would be non-zero. Overall, only a limited number of time points can be simultaneously shown with visual comparison concepts.

The explicit mapping of time on space is well established for non-spatial data. For scalar values over time, as they are given by state data, time value plots are widely used. According to [CCC+05], event data over time is typically encoded by marks on horizontal lines, as for example in [PMS+98]. In addition, also time value plots or variants of them like the ThemeRiver [HHWN02] are used. Here, time points of events are shown on the horizontal axis, while at the vertical axis, additional scalar information such as the number of events is encoded. For periodic event data, alternative mappings have been proposed for time on space such as the spiral metaphor [WAM01].

4.1.2.4 Univariate vs. Multivariate data

Both data types, event and state data, have to be visualized in univariate and multivariate context. When regarding univariate event data, the events are discerned by the time point when they occur, 3-D location, and general event type (diffusion or reaction). In a multivariate context, event data is additionally distinguished by the state variables the events are related to. Looking at state data, the univariate data includes the value distribution over space and time for a single state variable. This data given for all state variables forms the multivariate state data. Thus, for each point in space and time, a scalar value is given for every state variable.

To visualize event data, similar concepts can be used to locate univariate and multivariate event data in time and space, as univariate and multivariate events refer to the same data entities. Multivariate event data, however, requires additional visual cues to discern events by the state variables they are related to. To visualize multivariate event data in space, visual attributes such as shape and color are commonly used.

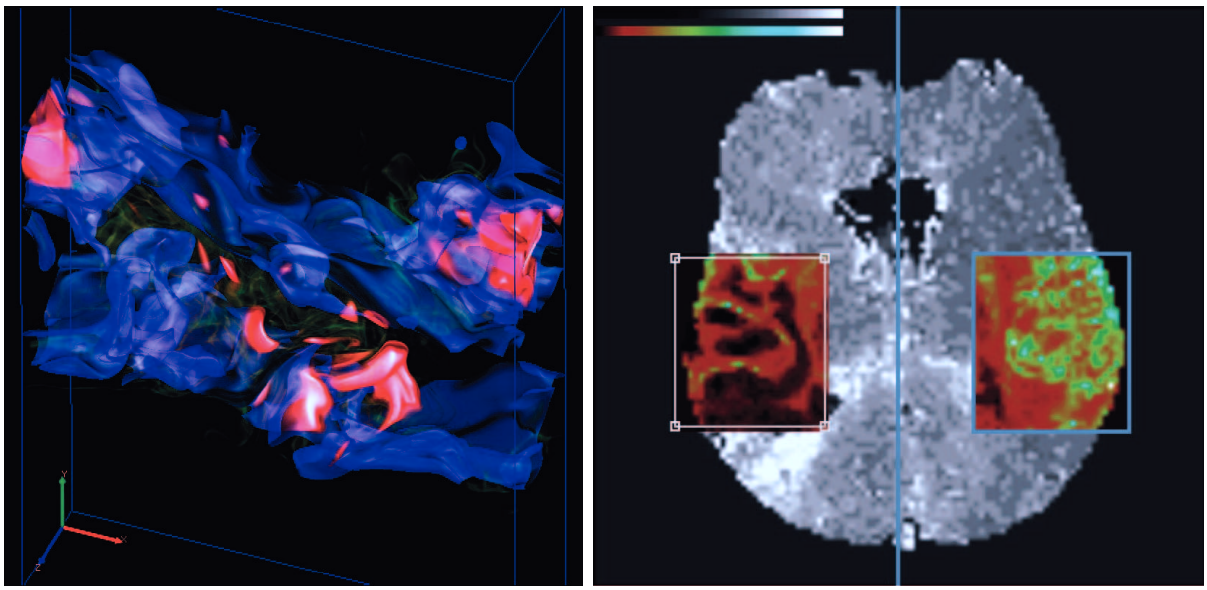
To visualize the spatial and temporal context of univariate state data, concepts can be applied that have been discussed in Sections 4.1.2.1 and 4.1.2.3. The visualization of multivariate state data in space and over time is a more challenging task. To handle multivariate data, a number of specific visualization approaches exist. Scatter plots and parallel coordinates are examples, but they are not designed to communicate a 3-D context of the data.

For the visualization of multivariate volume data, two general approaches are distinguished in the literature:

- Multivariate direct volume rendering

- Visualization of multivariate data on 2-D slices

Using *multivariate direct volume rendering*, the multivariate values per grid point are mapped to graphic attributes of the grid point's visual representation. Then, volume visualization methods (see Section 4.1.2.3) can be used for rendering. Akiba et al. [AMCH07] list the following three mapping techniques in this regard: The first technique maps multiple variables to the individual color components of the voxel color. This approach allows the visualization of at most three variables, as the common color spaces like HSV and RGB compose colors from three components. Secondly, one variable value can be selected as the representative value of the grid point. Therefore, a suitable rule needs to be found to select the representative variable. The third approach refers to composing multiple variable values into one value. An example is shown in Figure 4.2(a). Here, a proper weighting of values needs to be derived.



(a) (b)

Figure 4.2: Example visualization techniques for multivariate spatial data. In (a), the color of each grid point is composed from multiple weighted parameter values [AMCH07]. In (b), the visualization of multiple variables is supported by using an interactive lens (in the example, a synchronized lens is used for comparison of symmetric regions) [POM+09]. (Image (b) is courtesy of Christian Bendicks, University of Magdeburg. Data are courtesy of Jonathan Wiener, Boca Raton Community Hospital.)

An alternative to the multivariate visualization within one volume is to visualize one volume for every variable. Such an image based comparison requires more screen space than a com-

pact visualization. In addition, a direct comparison of values at the same 3-D location is not supported by this approach.

To this end, several visualization concepts have been proposed that make use of a *visualization of multivariate data on 2-D slices*. Preim et al. [POM+09] propose the use of color icons or lenses. Color icons are composed of multiple color values, each derived from the mapping of one scalar value of the voxel. The icon for each voxel is displayed at the corresponding position on the 2-D slice. An interactively located lens allows the display of additional variables in the region covered by the lens. An example is provided in Figure 4.2(b).

All these approaches are designed to visualize at most 3 or 4 variables. This is not only due to technical limitations. More importantly, the limitation arises from the observation that visualizing multiple variables in a spatial context is highly demanding for the user.

4.1.3 Visualization Design

To find an appropriate visualization design, the design choices are based on the data classification in the previous Section 4.1.2. As discussed in Section 4.1.2, three opposite pairs of data characteristics need to be considered for the visualization design in addition to spatial context: states vs. events, static vs. dynamic data, and univariate vs. multivariate data. The goal is to derive a visualization design that supports the visual analysis of the data by all these aspects.

4.1.3.1 States vs. Events

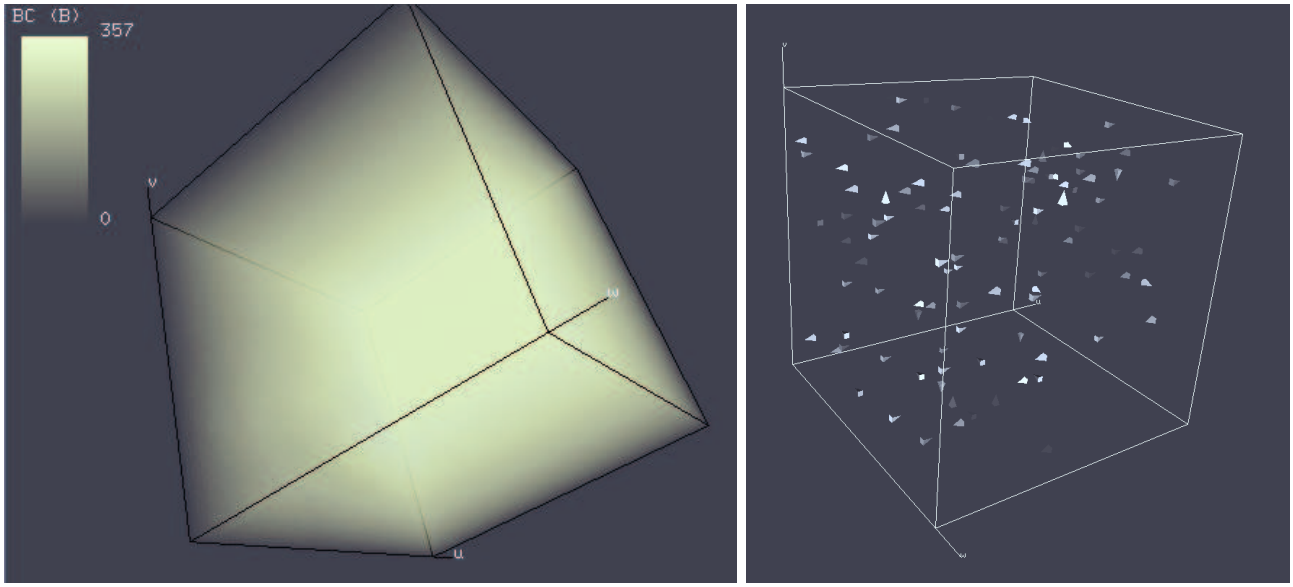
As the first pair of data characteristics, the visualization of states and events is discussed. Both require different concepts in order to be shown in spatial context.

State data can be visualized by volume visualization methods. Two general rendering methods have been identified in Section 4.1.2.2.

- surface extraction
- direct volume rendering

As the data at hand is not continuous in space, meaningful iso-surfaces cannot be extracted from the volume data. Thus, **direct volume rendering** [LCN98] is used to visually analyze the value distribution. To achieve interactive frame rates, texture slicing with object aligned slices [EHK+04] is applied, as it is one of the fastest approaches for volume rendering (Figure 4.3(a)), and the accuracy of the approach is sufficient.

Event data in spatial context is characterized by 3-D locations. To this end, the common and intuitive visual representation of placing **3-D icons within 3-D space** is adapted (Figure 4.3(b)). The shape icon is used to visually discern the two event types, reactions and diffusions. A reaction induces changes within one sub-volume. To show a reaction, it is sufficient to indicate the position of the sub-volume. Hence, reactions are mapped to cubic icons covering the sub-volume. Diffusion describes the movement of a particle between two neighboring sub-volumes, which requires the display of source and destination of the moving particle. Icons for diffusions cover both the source and destination sub-volume. To convey the direction, the icon is shaped as an arrow pointing to the destination.



(a) States (b) Events

Figure 4.3: Basic visualization of states (for a single time point) and events (for multiple time points) in spatial context.

4.1.3.2 Static vs. Dynamic

The visual analysis of the simulation data incorporates both the exploration of value distributions at a single time point as well as exploring trends over time. For one time point, the visualization needs to include the multivariate state and event data in spatial context. A visualization of the dynamic data should convey the state changes over time and the sequence of events that leads to these changes.

Static data is basically visualized by direct volume rendering for state data and a 3-D scatter plot to show the current event. The type of the event is distinguished from the shape of the icon included in the scatter plot.

Dynamic data is challenging to be visualized, considering the 3-D context of the data. But temporal developments are a main aspect of simulation data. Therefore, appropriate concepts to visualize event and state data in spatial and temporal context are discussed in detail. In Section 4.1.2.3, four general concepts have been identified to visualize temporal aspects: Animation, image based comparison, data based comparison, and time as a spatial dimension. With respect to the application context, they all have inherent strengths and weaknesses, which are summarized in Table 4.1. To show all aspects of the data, a combination of these concepts is necessary.

	Advantages	*Drawbacks*
Animation	intuitive mapping of all time points	sequential display of time points
Image based comparison	simultaneous display of time points spatial context	subset of time points additional screen space required
Data based comparison	simultaneous display of time points spatial context	subset of time points
Time as spatial dimension	all time points simultaneously	no spatial context

Table 4.1: Potential visualization concepts to show dynamics of spatial simulation data. Each concept has advantages and drawbacks with respect to the application field.

Animation is well suited to gain a general impression of temporal developments in simulation data. For state data, animation is straightforward, as dynamics in state data become apparent from changes in the volume visualization over time. The characteristics of events, however, require special consideration. Also, the high number of time points in the simulation data is challenging for animation, because showing all time points takes up a reasonable amount of time.

Event data is not well visualized in animation if only the current event of the time point is shown. Each event would flash for a very short time, which is visually hard to follow and reveals little information about the temporal context. In the visual analysis, the sequence of events is of interest. Thus, instead of only one event, the preceding sequence of events is visualized in the animation by including the respective events within the 3-D scatter plot. While the

animation progresses, new events are added to the visualization. To convey the temporal order in the sequence of events and to avoid a cluttering of the visualization due to a high number of events, events are slowly faded out during animation. After the event icon appears opaque in the visualization of the time point when the event occurs, its transparency increases as the animation progresses until it becomes invisible.

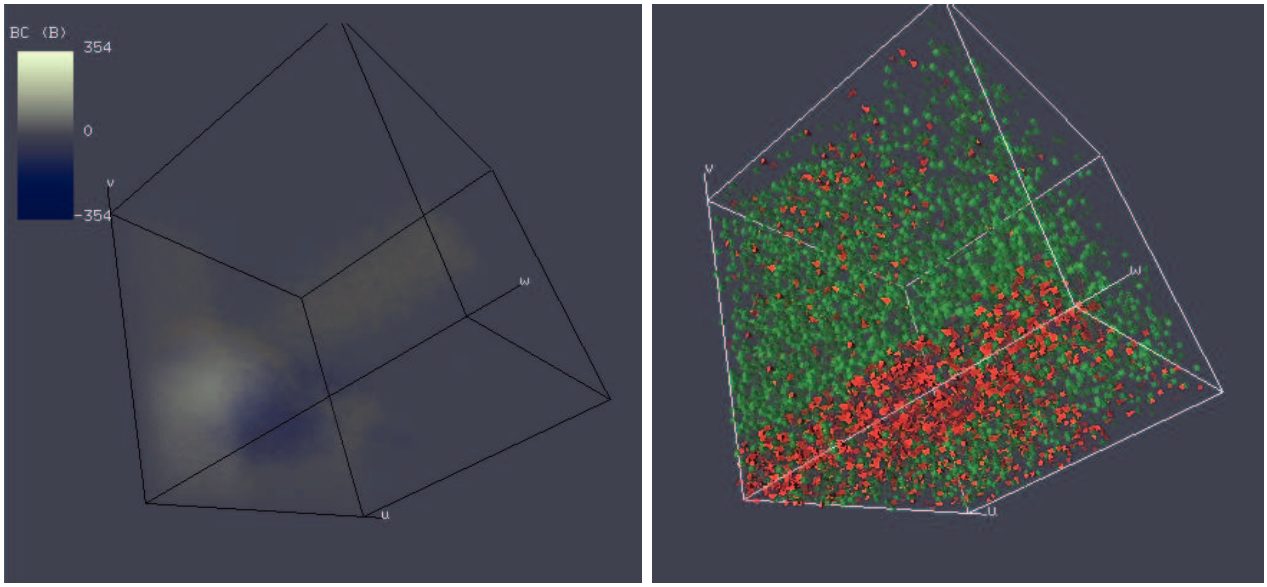
The second problem for animation is the high number of time points in the data. To reduce the time needed to explore temporal developments in the data, an interactive adjustment of the speed of animation is necessary. In an accelerated animation, time points have to be skipped and only key time points are shown. This works fine for state data, because the visualization of state data at key time points still reflects the overall changes in the data. However, skipping events leads to an information loss. This can be avoided by also showing events from skipped time points in the visual representation. Hence, not a single event is added to the 3-D scatter plot from one animated time point to the next, but multiple events that reflect the changes. To avoid a flickering of the visualization, the concept of fading out older events is also used if time points are skipped.

With these concepts, animation together with interactive control about the speed of animation is used to gain an overview over temporal developments of state and event data in spatial context.

Animation provides a sequential exploration of temporal developments. To compare the complex data of time points in all facets, the simultaneous visualization of time points is valuable. This is supported by visual comparison techniques.

Image based comparison of state data is suited to visualize the spatial value distribution at multiple time points. For event data, image based comparison results in multiple 3-D scatter plots, each showing a single event. Adapting the approach of showing the preceding sequence of events, as already described for animation, image based comparison of event data allows the comparison of sequences of time points. Image based comparison has the general drawback that additional screen space is required to show multiple images.

Data based comparison, on the other hand, has the advantage that data from multiple time points is brought together in one visual representation. It does not require additional display space compared to visualizing static data. With respect to event data, data based comparison is suitable to combine events from multiple time points in one 3-D scatter plot (Figure 4.4(b)). This approach has already been described for the visualization of sequences of events. For state data, a different concept is required. Here, data based comparison can well be used to emphasize deviations between time points by computing a difference image computed



(a) State data (b) Event data

Figure 4.4: Data based comparison of state data and event data. For state data, the changes between two time points are visualized in a difference image. The approach is limited to the comparison of two time points. For event data, numerous time points are shown within one volume. In addition to the shape of the node icon, the two types of events, reactions and diffusions, are separated by color. Reactions are shown in green, diffusions in red.

from the values at the time points (Figure 4.4(a)). It is limited, however, to the comparison of two time points.

The detailed exploration of time points requires the identification of relevant time points. This can be further supported with a simultaneous visualization of all time points by the explicit mapping of **time to a spatial dimension**, which requires leaving out spatial information. The goal is to gain an overview on the complete time series at a glance.

To this end, the idea is followed to derive *high level features* from the data at each time point to characterize the data independent from spatial context. High level features are derived from the original state and event data in spatial context, denoted as *low level features* in this regard. Low level state data is given by scalar values in space. Abstracting from spatial information, the sum of particles in all sub-volumes is used to characterize the time point by high level state data. The high level representation of events, only the occurrence at a time point is considered, not the location in 3-D.

The visualization of high level features of state data can be achieved with established representation like a time value plot or a heat map over time. For high level event data over time,

markers on a time line are proposed in the literature. But due to the very high number of time points, multiple time points are mapped on a single pixel in the display. This problem has to be considered for high level events and states. With respect to high level events, every time point corresponds to one event. The visualization has to convey not only if a high level event is mapped to a pixel, but how many. For this purpose, a novel concept is introduced. Similar to a heat map over time, color is used to encode how many high level events are mapped to each pixel on the time line. In Figure 4.5, the visualization of high level event data over time is shown. Regarding high level state data, one scalar value is given for each time point and variable. The high number of time points leads to the mapping of multiple scalar values for numerous time points to one pixel. A heat map is not feasible to visualize multiple scalar values per time point. A time value plot is better suited. All scalar values that are mapped to one pixel can be displayed by their position on the vertical axis.

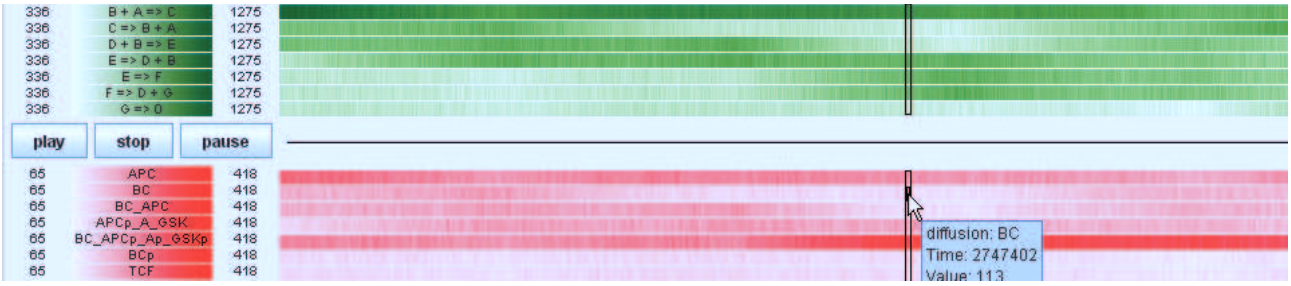


Figure 4.5: Explicit representation of time on a spatial axis. Occurrences of events over time are shown in a heat map. The two types of events, reactions and diffusions, are separated by color. Reactions are shown in green, diffusions in red.

In summary, appropriate visualization concepts to convey temporal developments along with spatial context include animation, image based comparison, and data based comparison. In addition, a simultaneous visualization of all time points, based on high level features that discard spatial context, supports a general overview on temporal developments and allows the identification of time points of interest for detailed inspection. All these concepts are applied to both state and event data, resulting in different visual representations.

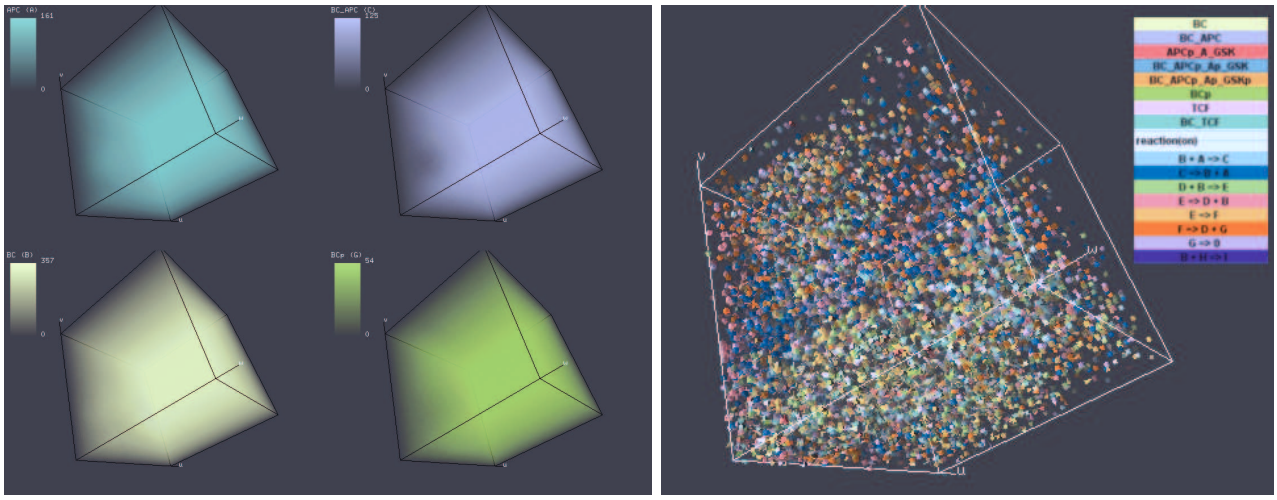
4.1.3.3 Univariate vs. Multivariate

Due to distinct characteristics of state and event data, univariate and multivariate data have to be handled separately for the two data types.

Univariate data To visualize univariate state data in spatial context, standard approaches of direct volume rendering can be applied. Univariate event data is shown by icons located in 3-D space, where the shape of the icon indicates type of event.

Additionally, the need for the visualization of non-spatial data has been motivated in order to cope with dynamic aspects of the data. Visualizing abstract state data in univariate context comprises the visualization of one state variable over time, which can be done by one time value plot. Univariate event data, on the other hand, includes all events that appear over time, separated by the two basic event types diffusion and reaction.

Multivariate data The visualization of multivariate state data is possible by combining multiple variables in one volume or by a combination of multiple univariate volumes. The combination of multiple variables in one volume is not suited for the data at hand, as different scales exist in the data, which cannot be integrated into a multivariate visualization. Hence, multiple univariate volumes are used to gain a general comparison of value distributions for multiple variables. An example for the visualization including one volume for each variable is shown in Figure 4.6(a).



(a) Multivariate state data

(b) Multivariate event data

Figure 4.6: Visualization of multivariate data in spatial context. Left: Combining multiple univariate state volumes. Right: Using colors to encode multivariate event data.

To visualize multivariate events in spatial context, the visual attributes of event icons are adapted. As the shape is already determined by the event type, the icon color is used. The color is encoded according to the state variables that are related to the event. Diffusions are defined for one state variable. Thus, the color of the state variable can be used to encode the

event icon. Reactions involve multiple variables. The icon color of a reaction cannot be mapped to those of the corresponding state variables. Instead, one user-defined color is assigned to all reactions that are related to the same state variables. Figure 4.6(b) shows multivariate events in the volume.

For an overview of developments over time, high level state data and event data have to be visualized in multivariate context. In general, values of multivariate high level state and event data can be visualized in a local or global context. Global context enables the comparison of variables; local context supports a more precise visualization of the values of each variable, as different scales exist in the data. Multivariate high level state data can be visualized in one time value plot to show variables in global context. Local context of variables is shown by one plot for each variable, each comprising the local value range of the high level feature. For high level events, multivariate context leads to multiple rows in the heat map, each comprising the occurrences of one variable over time. In a local context, the color scale of the heat map is based on the local value range of a variable. In a global context, the mapping of data to color is performed with respect to the global value range. In the application context, the appropriate mapping depends on the intention of the user. The visualization of multivariate event data with a local mapping is shown in Figure 4.7.

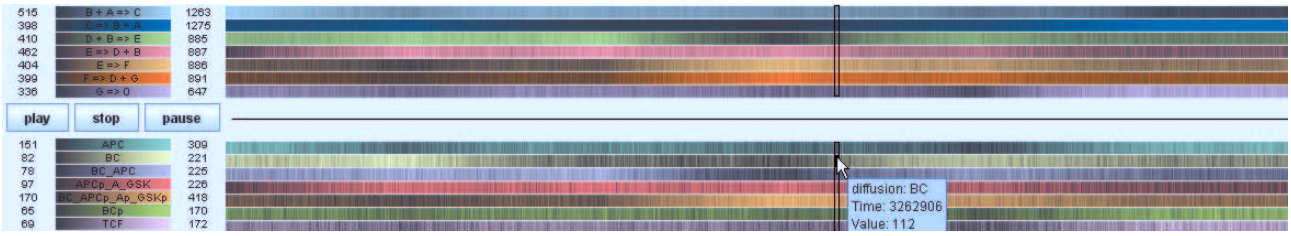


Figure 4.7: Visualization of multivariate event data over time with color mapping based on local value ranges.

4.1.4 Multiple View Framework

To visualize the data by all the aspects that have been discussed, a single view cannot provide a comprehensive solution. Therefore, multiple coordinated views are used. Accounting for the guidelines for the use of multiple coordinated views in [BWK00], a minimum number of views should be derived from the proposed visualization concepts.

The main challenge for the visual analysis of simulation data is the co-existence of spatial and temporal context. The demand to include both aspects explicitly in the visualization has

induced the development of an overview and detail concept to handle the data. The overview provides an abstract view on temporal developments and discards spatial context. It further serves as a means to identify time points of interest. The visualization of abstract data over time is subsumed in one view, the **Time View**. Data at time points of interest is analyzed in detail by additional views that focus on the spatial context. As it has been shown in Section 4.1.3, the visualization of event data and state data in spatial context requires different concepts due to the different characteristics of the two data types. Therefore, separate views are implemented for event and state data. **Event View** and **State View** subsume concepts that show spatial data for either one or multiple time points and in either univariate or multivariate context.

These views need to be visually linked in one multiple view concept. In the following, the three views are described as well as their linking and necessary interaction methods to adapt the visualization to the user's needs.

The framework is flexible to account for additional views. This includes further time views that provide temporal development in different granularities, which might be necessary to cope with high numbers of time points, or further detail views to analyze multivariate state data on 2-D slices.

4.1.4.1 Time View

The objective of the Time View is to give an overview on developments in the simulation data over time. To this end, event and state data are abstracted from spatial context. For univariate event data, the derived high level features comprise the occurrences of events over time, which are distinguished for the two event types reaction and diffusion. Additionally, the multivariate context of events is considered by computing the occurrences separately depending on affected state variables.

For event data, a heat map has been identified as a useful visual representation. This is a very compact visualization, also in multivariate context. State data is visualized by one time value plot per state variable. As the visualization of multivariate state data requires additional screen space, the number of state variables is limited. With respect to the State View described in the following, a maximum of four variables is shown simultaneously.

Accounting for the goal of identifying time points of interest in the overview of temporal developments, additional functionality is necessary to visually separate time points. To this end, an interactive lens is used. It allows focusing on single time points while maintaining the context of the whole time series. Controlled via mouse movements on top of the time value plot, the visualization is distorted in horizontal direction to widen the screen space within

which the currently focused time interval is shown. At the same time, the time points outside this interval are visually compressed. This supports a very fast visual separation of close time points without losing the overview over a larger time range.

4.1.4.2 Event View

The Event View integrates the visualization concepts for event data in spatial context considering all aspects discussed above. The event for one time point is visualized as a 3-D icon at the corresponding 3-D location in the volume. The shape of the icon encodes the event type, while the color discerns events by related state variables. This approach is easily extended to visualize events from multiple time points. They can be included within one volume in the sense of a data based comparison. Considering a sequence of events over time, temporal order is conveyed by the transparency of the icons. Although image based comparison is generally applicable, it is not supported due to the limited available screen space in the multiple view framework.

To distinguish events by related state variables, color coding is used. However, the color coding of multivariate events has limitations. The user can only discern a limited number of colors, especially as icons representing the events are rather small in the visualization. Therefore, the user has the ability to interactively select a subset of events, based on related state variables. The events are then explicitly highlighted in the visualization by the use of color, while all other events are visually de-emphasized by either hiding them or applying a uniform color.

4.1.4.3 State View

For a visualization of state data in spatial context, the State View has to provide data in a static and univariate context, but also to support the comparison of multiple time points and for the comparison of multiple variables. In general, univariate volume visualization shows the data for one variable at one point in time. Including multiple univariate volume visualizations for different variables enables the comparison of variables.

The data presented at once has to be limited to an amount that can be perceived at once. Compliant with findings from the literature, the number of simultaneously shown univariate volumes is limited to four. The color of a univariate volume is determined by the corresponding state variable, the scalar value of the voxel is mapped to transparency. The limitation to four volumes allows for a very efficient rendering, as the data can be stored within a single

3-D texture in $RGBA$ format and loaded into the texture memory. Every element in $RGBA$ contains a transparency value that encodes the data value at that point.

For comparison of time points, it would be possible to show multiple univariate volumes of one variable for different time points. However, in the limited screen space, either multiple time points or multiple variables can be shown, but not both. To apply the same concept for comparison of time points in both univariate and multivariate context, only data based comparison is supported in the State View, image based comparison of time points is not supported. A difference image is built from the volumes of one variable at two different time points. Here, diverging color scales are used to show either increase or decrease of values in the volume.

4.1.4.4 Visual Interface

To visualize the data in all facets, three basic views have been introduced. The Time View serves as an overview on temporal developments to identify time points of interest, the two spatial views State View and Event View provide details at one or multiple time points.

The integration of the basic views within one visual interface gives rise to a number of challenges, which are addressed in this section. One challenge is the **arrangement of views**. This mainly includes bridging the gap between overview and detail, but also the arrangement of the detail views, which visualize different facets of the data. The second challenge is to develop a coherent **interaction concept** throughout the views. Breaking up the data set into multiple views also requires, as the third challenge, the visual communication of relations in the data across views by a **visual linking**. These three aspects are discussed in the remainder of this section.

The **arrangement of views**, the first of the three challenges, is driven by the overview and detail concept: the overview on temporal developments, provided by the Time View, serves for the identification of time points, the detail views support their further investigation. But the relation among views is not solely top-down from overview to detail. The visual exploration involves a constant back and forth between them. Also, the overview can help to integrate the data of a single time point into the overall context. Thus, it is reasonable to visualize overview and detail views simultaneously. A vertical split of the screen space is used to gain as much width as available for the Time View, where the identification of time points is carried out. The detail views, State View and Event View, represent complementary facets of the data and need to be shown simultaneously. As the volume data shown in both views usually is best fit into a squared display space, they are positioned beside each other, placed above the Time View.

The second challenge that has to be addressed is a coherent **interaction concept**. This comprises local interactions within one view and global interactions affecting all views. In the following, local interactions are described for each view, before interactions with global effect are discussed.

The Time View is equipped with local interactions to control the visual representation of high level state and event data. This includes the selection of event variables, separated by reactions and diffusions, and state variables, whose high level features are shown. Additionally, it can be selected whether high level state and event data are shown in local or in global context.

Local interactions in the State View include the control over the spatial context by adjusting the applied volume transformation. Further, the user can switch between univariate and multivariate context. In multivariate context, up to four variables can be selected, as the number of simultaneously shown variables is limited. In the Event View, local interactions are similar to those in State View. The volume transformation can be interactively adjusted; the user can switch between univariate context and multivariate context for both reactions and diffusions. In multivariate context, up to four reactions and diffusions can be shown.

The most important aspect of the global interaction concept is the interactive control of the temporal context. In general, static visualization of one time point, animation, and comparison of time points have to be supported. Both static visualization and animation require the interactive control over one current time point. The comparison of time points requires the selection of two time points, which is the maximum number of time points that can be compared in the State View.

The control over the visualization in temporal context is integrated into the Time View, which has been introduced to supplement the selection of time points. In general, the selection of time points is provided with a horizontal time slider. Time points in the time slider and in the visual representation of high level features are mapped to the same horizontal position, to facilitate the selection of time points of interest. The user can interactively switch between two modes: the selection of one time point or of two time points. With respect to the first mode, the control over the currently visualized time point is important for static visualization and animation. A natural linking of visualization of static data and animation is provided by interactions like pausing, adjusting the animation speed, and jumping to time points of interest. To select two time points for comparison, the time slider is transformed into a range slider, which allows selection of two time points. Animation is not applicable in this mode. These local interactions in the Time View affect the visualization globally: The visualization concepts in State View and Event View are adapted, depending whether a single time point, an animation,

or the comparison of two time points need to be visualized.

In addition, the globally applied color scheme to distinguish multivariate states and events can be interactively adjusted by the user. All interactions that affect the display of variables are provided in an additional panel in the framework, distinguished by global options for assignment of colors, and local interactions for Time View, State View, and Event View.

The last challenge is the **visual linking**, in order to communicate relations in the data among views. This mainly comprises the following aspects:

- temporal context: Selected time points must be visually communicated in the high level visualization of the Time View, to provide a visual linking to the low level visualization of these time points in State View and Event View.

- spatial context: A visual linking is necessary between the spatial visualizations in State View and Event View, as the two views present complementary facets of the data.

- multivariate context: All views may comprise the visualization of multiple variables. The relation between high level representations of state variables in the Time View and their low level representations in the State View need to be conveyed as well as relations between high level event data in the Time View and low level event data in the Event View. Further, relations among state variables and events affecting these states have to be conveyed among the views.

The visual linking of views to convey temporal context is given by the interaction concept: Selected time points in the Time View, shown in the time slider, are visualized in detail in both the Event View and State View.

The linking among states and events by spatial context is provided by applying the same volume transformation to the volumes in State View and Event View. An interactive manipulation of the volume transformation in one view affects both views simultaneously.

The linking of data in multivariate context is supported by color coding. Among the views, the same colors are reused for visual representations of the same state variables or the same events throughout all views. To link states with correlated events, the same color is assigned to diffusion events and the affected state variable. Linking multivariate events among views, the same color is used if the events affect the same state variables.

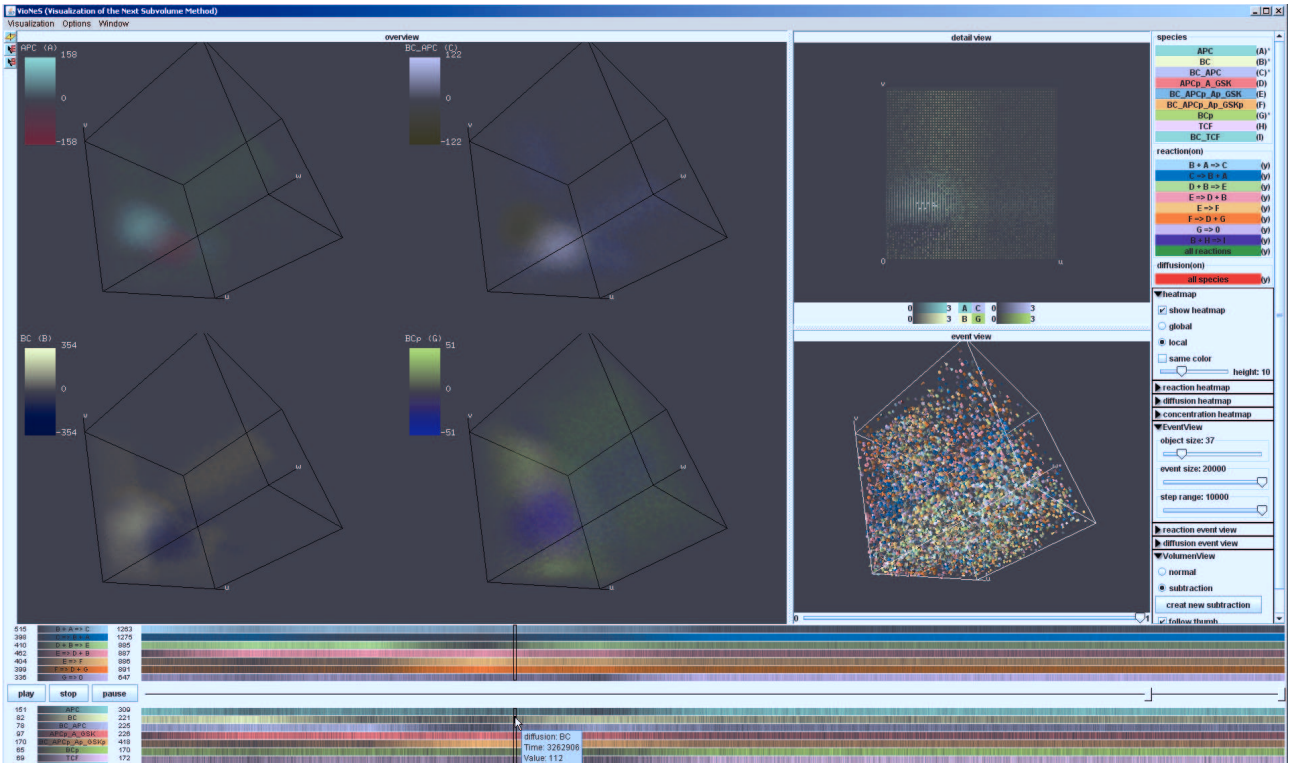


Figure 4.8: The multiple view framework, visualizing data from two time points. The Time View (bottom) provides an overview on the occurrences of reactions and diffusions over time. In the State View (top left), multiple difference images derived from state data at the two time points are shown, each for one state variable. The Event View (bottom right) includes the sequence of events between the two time points in multivariate context. Additionally, a 2-D slice view is included at the top right.

4.1.5 Discussion

The presented interactive multiple view framework allows the visual analysis of complex simulation data from the Next Sub-Volume Method in its entirety with three basic views. Each view subsumes concepts for various analysis aspects. Thus, the user needs to understand few visual representations, which helps to get an easier access to the data. Specifically, breaking down the spatio-temporal aspects of the data into separate views supports the visual exploration of large and complex simulation data sets. A Time View gives an overview on the dynamics and points to relevant developments over time, which can be inspected in detail in a State View, visualizing univariate or multivariate state data for one or multiple time points in spatial context, and an Event View, which displays sequences of univariate or multivariate events in spatial context. A close linking among the views reveals relations among different facets of the data.

The framework is flexible in the sense that additional views can easily be added. For example, a detail view to analyze multivariate state data on 2-D slices is shown in the screen shot of the framework in Figure 4.8.

The visual data exploration is supported by a highly interactive user interface, to intuitively adapt the visualization to current needs of the user. Time points can be interactively selected and explored in either static context or in dynamic context in an animation or by data based comparison of time points. Visualized variables of event data and state data are chosen in an additional panel, thus allowing the manipulation of the advanced color coding scheme for variables and combinations of variables.

The framework is scalable to large data sets. Using the visualization, data sets with up to a million time points have been analyzed. This is enabled by reducing the large size of the data to high level features at all time points in a pre-process. An increase or decrease of particle numbers over time, which is reflected in high level state data, and accumulations of certain events, as represented by high level event data, lead to time points of interest. During the visual analysis process, only low level data related to current time points of interest has to be loaded into the framework. Using a GPU based rendering, the visualization of data sets with a resolution of 64x64x64 sub-volumes has been conducted at interactive frame rates.

The visualization framework has been used by simulation researchers within the research training school for visual debugging of algorithms. The visualization technique supports checking for unexpected distributions of particles or an accumulation of diffusion events in a specific area, which can hint to errors in the algorithm implementation. The presented methods thus help the user to first get an overview of the data and then zoom in to analyze the specific states of single time points, and, further, single sub-volumes.

4.2 Visual Analysis of the Attributed Π-Calculus

In Section 4.1, the systematic analysis of visualization concepts for complex simulation data leads to a multiple view framework that closely links and coordinates specific views. In this section, the results are adapted to visualize complex simulation data from another modeling and simulation approach, which has been used in the research training school: the Attributed Π-Calculus [JLNU08, JLNUar]. Data generated by this approach has other characteristics than the data regarded in the previous section. Instead of an explicit spatial context, the data consists of a time series of reaction networks with multiple attributes. Handling such time-varying structures is an open challenge in the field of visualization.

Similarly to Section 4.1, the visualization technique is based on the concept to break down the complexity of the data along the temporal axis: High level features, which characterize the data over time, give an overview on the dynamics of the data. The high level features lead to time points of interest, which are analyzed and compared in detail by additional views. These views are combined in a framework of multiple linked and coordinated views.

Three main aspects were discussed in Section 4.1.5 that are relevant to derive the visualization concept for complex simulation data:

- The development of appropriate views that support the exploration of the different facets of the data.
- The development of a suitable visual interface to provide view arrangement, an interaction concept, and a visual linking of views.
- Appropriate high level features have to be derived to characterize the dynamics of the data.

In the following, these three aspects are regarded with respect to the simulation data from the Attributed Π-Calculus. First, the simulation data is described in Section 4.2.1, along with the derivation of appropriate high level features. Then, the necessary views as well as their integration within one visual interface are developed in Section 4.2.2, before the results are discussed in Section 4.2.3.

4.2.1 Simulation Data from Attributed Π-Calculus

The Attributed Π-Calculus [JLNU08, JLNUar] extends the Π-Calculus [Pri95, PRSS01], a modeling formalism that has been applied to model cell biological systems. The Π-Calculus represents reactions between proteins as communications between processes. The Attributed Π-Calculus introduces attributes of processes. Additional constraints that are based on attribute values are used to restrict the ability of processes to communicate with each other. This can be used, for example, to constrain the ability of proteins to react with each other depending on spatial locations within the cell.

The result of one run of the stochastic, discrete event based simulation is a time series of reaction networks, one for every time point at which a reaction occurs. Depending on the number of proteins to start with and the duration of the simulation, the amount of data gathered by this process grows large very quickly. The data to be visualized consists of a set of proteins, a set of reactions, and a set of directed links connecting proteins with reactions – for

each time step. In the following, a graph-based notion is used for the data for each time point: the reaction network consists of the two distinct node sets proteins and reactions and directed links between them as the edge set.

Each node has a name, a set of *attributes*, and a set of *properties*. Attributes and properties are separated because of their distinct characteristics. Attributes are attributes in the sense of the Attributed Π-Calculus, which identify and parametrize the proteins. They are static values that do not change over time. The number and the meaning of attributes depend on the implementation. The property, on the other hand, is dynamic as the value changes over time. In the exemplary data that has been used for this work, two attributes and one property are given for the nodes of each node set.

For the derivation of high level features that characterize the reaction network at each time point, the dynamic aspects of the data are considered: *graph structure* and *node properties*. To numerically express the graph structure, so called *complexity measures* are employed. Besides several simple graph characteristics like the relative number of edges (also called *density*), the number of components and their average graph theoretical diameter, or the average node degree, other measures have been developed that (implicitly) take into account many structural facets like the branching factor and the number of cycles and condense them into one single value. An overview of a number of such measures is given in [BB05]. In the application context, different complexity measures are made available, as the choice of an appropriate measure depends on the current analysis goal.

Referring to node properties, a straight forward approach is to use them directly as high level features. However, as one node may not exist at all time points due to structural changes, its property is not defined at every time point. A specific data characteristic, nevertheless, allows the definition of high level properties for all time points: if a node has an impact on the reaction network at a time point, the value of its node property is greater than 0. Hence, a node property value of 0 is assumed at all time points where the node is not part of the reaction network, as this is equal to a concentration of 0 for proteins or a reaction rate of 0 for reactions.

Consequently, the high level data comprises:

- a set of complexity measures of which each measure describes the dynamic graph structure by one time series

- a set of node properties of which each property is described by multiple time series, one for every node that has this property

4.2.2 Multiple View Concept

The visual analysis of the time series of reaction networks aims at gaining insight into changes in structure and in property values over time. Applying the overview and detail concept in this regard, an **Overview** visualizes general temporal developments with respect to changes in structure and property values. The Overview incorporates high level features of the data, characterizing the dynamics in the data and allowing the identification and selection of time points of interest. The visualization of low level features for specific time points is conducted in a **Detail View**. The low level features comprise one reaction network for each time point with two node sets, edges between them as well as node attributes and properties.

In the following, the visualization concepts are developed for the **Overview** (Section 4.2.2.1) and **Detail View** (Section 4.2.2.2). The visual interface is presented in Section 4.2.2.3.

4.2.2.1 Overview

The goal of the Overview is to provide a general impression of the dynamics in the data and to let the user identify time points of interests. These dynamics are described by the high level features that have been derived in Section 4.2.1: a set of complexity measures and a set of node properties, each property containing one time series for every node with that property. This data has to be visualized in the Overview. Two problems have to be addressed in this regard. First, multiple data scales exist in the high level data. Each complexity measure and each node property has its own data scale. Second, a high number of time points is given in the data. Further, these time points are unevenly distributed due to stochastic events in the simulation. This complicates the identification of time points of interest, which is one goal of this view. In the following, it is discussed how these two problems are addressed in the visualization.

In Section 4.1.3.2, time value plots have been proposed for scalar high level features over time. Due to multiple scales in the data, multiple node properties and complexity measures cannot be combined in one view. Hence, one time value plot is necessary for each node property and each complexity measure.

These time plots have to be arranged within the Overview, so that the user can explore the high level features in all facets. Two approaches can be applied in general. The first is to show only one plot at a time. The user can sequentially explore the data by visualizing the different high level features subsequently. The second approach is to visualize multiple time value plots at once and arrange them simultaneously within the view. The data is presented in parallel to the user.

Both approaches have advantages and drawbacks. The sequential exploration of multiple time plots step by step over time supports a detailed visual inspection of temporal developments, because the axes of the time value plots can be spanned over the whole available screen space. As a major drawback, the sequential exploration leads to high cognitive demands for the user to comprise the data in its whole. The parallel exploration is less demanding because multiple time value plots are simultaneously shown within the view. But arranging the views side by side comes with the cost of diminished visual representations of the high level features, which makes the exploration of temporal developments difficult.

To balance the advantages and drawbacks of both approaches, a combination is chosen as the visualization concept for high level features. In general, multiple time value plots are shown in parallel within the Overview. But instead of arranging them side by side, they are overlaid onto the same screen space. Hence, the axes of the time value plots are spanned over the whole display space available for the Overview. Also, the overlay of plots has the advantage that the common time scale of the high level features is conveyed by the visualization. However, the difficulty for the user to relate the shown time series to the respective time value plot grows quickly with the number of overlaid time value plots. To find a reasonable compromise with respect to the numerous data scales in the high level features, the number of time value plots shown at once is limited to two. These two value plots are chosen from the two basic parts of the high level features: One from the set of complexity measures and one from the set of node properties. Visualizing both aspects provides the simultaneous display of the complementary aspects of the high level features. Other node properties and complexity measures can be sequentially explored, based on user interaction. It should be noted that showing one node property in a time value plot involves multiple time series, while one complexity measure corresponds to one time series.

The combination of parallel and sequential exploration of time value plots forms the basic concept of the Overview. It is accompanied by color coding for a clear visual distinction of the overlaid time value plots. Separate colors are applied to the node property and the complexity measure. The concrete color scheme is described in Section 4.2.2.3, as it is derived within the multiple view concept.

The second main problem is the high number of time points in combination with the uneven distribution of time points. This problem is very similar to the discussed appearance of a high number of events in Section 4.1.3.2. To cope with this problem, a heat map was introduced to visualize occurrences of events over time. The approach is adapted to support the identification of time points. It is located directly below the time axis of the time value plots (see Figure 4.11),

to provide a linking with the visual representation of the high level features. The information encoded by the heat map is comparable to a histogram: for each pixel along the time axis, it indicates how many time points are mapped onto that position, with a color scale ranging from white (high number of time points) to black (no time point at that pixel). Covering only a small part of the screen space, the heat map supports the task of identifying time points. As an additional concept to separate close time points, an interactive lens has been proposed in Section 4.1.3.2 to separate close time points from each other. This concept is also used for the visualization of high level features from the Attributed Π-Calculus. The resulting visualization concept for the Overview is shown in Figure 4.9.

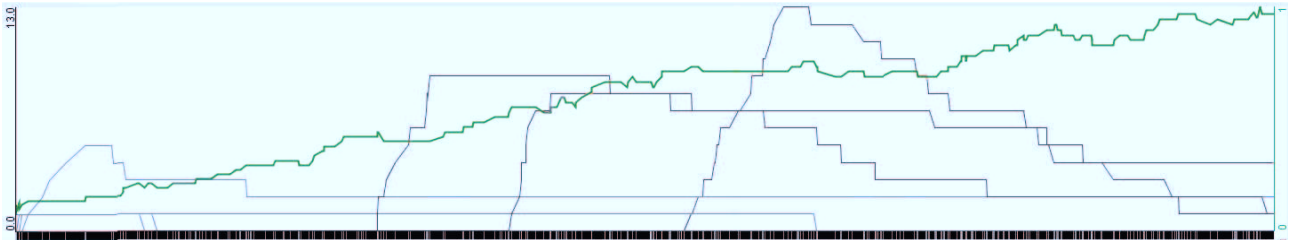


Figure 4.9: The time value plot of the Overview shows one node property, comprising multiple time series, in grey, with the data range shown on the left and one complexity measure in green, with the data range shown on the right. On the bottom, the heat map is visualized to show the uneven distribution of time points.

4.2.2.2 Detail View

The Detail View needs to support the analysis of a single time point and the comparison of multiple time points. The data at one time point is a reaction network that consists of two node sets, their attributes and properties, and the links between the node sets. To visualize a single time point, the scalable and highly interactive visualization technique for reaction networks is used, which has been presented in Section 3.2 and published in [SJUS08].

For the comparison of data from multiple time points, two approaches have been identified in Section 4.1.3.2: data based comparison and image based comparison. Data based comparison means that differences between time points have to be computationally determined before visualizing them within one image. As building a difference image that reflects the structural changes in the data is non-trivial and would introduce a new view within the already complex visualization, an image based comparison is preferred for the data at hand. Hence, multiple instances of the table-based visualization, each showing a different time point, are simultaneously

displayed. Due to the limited screen space and the cognitive demands to compare multiple reaction networks, the number of concurrently shown time points is limited to two. The visual representations of both time points are arranged beside each other as shown in Figure 4.10.

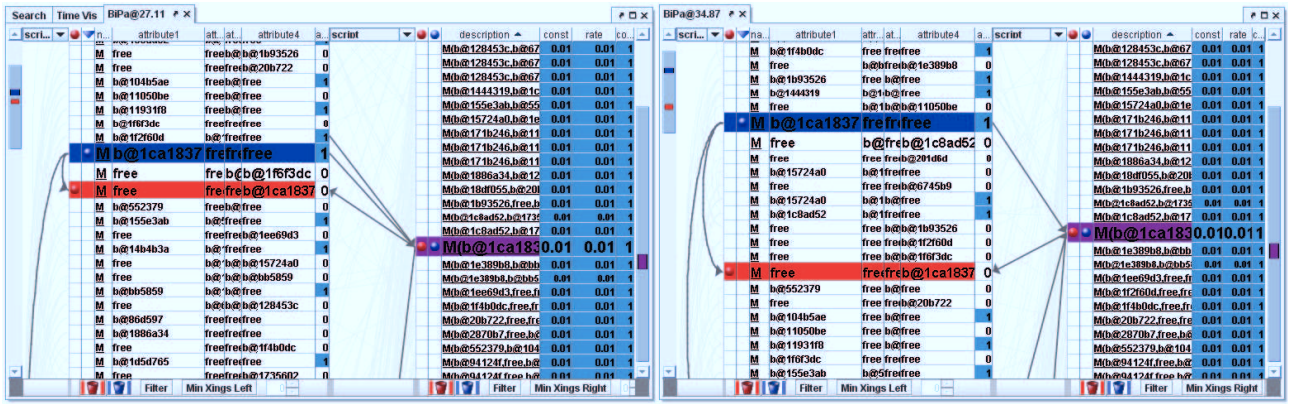


Figure 4.10: The Detail View, showing the reaction networks of two time points beside each other, each displayed using the table-based visualization introduced in Section 3.2.

4.2.2.3 Visual Interface

For the linking of views in a multiple view framework, three challenges have been identified in Section 4.1.4.4, which are now discussed with respect to simulation data from the Attributed Π-Calculus:

- arrangement of views
- interaction concept
- visual linking

The **arrangement of views** is very similar to the concept used for simulation data from the Next Sub-Volume Method, because both concepts incorporate overview on temporal developments and detail views for specific time points: The views are simultaneously visualized, with a vertical split of the screen space to acquire maximum width for the Overview with its numerous time points. For the comparison of multiple time points, a Detail View showing two time points in parallel is used. Also, views can be hidden and restored if a larger display is temporarily required for one view.

The **interaction concept** has to account for two main aspects: In the Overview, interactive facilities need to be provided to select either one or two time points, which are visualized in the

Detail View. Second, the selection of nodes of interest from the reaction network is necessary to analyze structural relations in detail.

For the selection of a single time point in the Overview, brushing of time points on top of the visualization is provided. The Detail View is adapted to the currently selected time point, providing an easy mechanism to explore single time points. The selection of two time points for detailed comparison requires a more elaborate interaction design. To provide a clear guidance of the user through the analysis process, the concept is tailored to comparison tasks usually carried out by the user. Two typical goals are identified:

- the comparison of one identified time point to its neighboring time points
- the comparison of one identified time point to other, non-neighboring time points

To address both goals, two mechanisms are provided that aim at an easy selection of time points. The first method allows the simultaneous visualization of two subsequent time points. This requires only the definition of a single time point. The successive time point is implicitly known. By going back and forth in time, deviations from one time point to the next can be directly investigated.

The second method aims at the more advanced comparison of two non-neighboring time points. Here, a 2-step selection is provided. First, the user identifies a time point of interest that he wants to investigate in detail. Upon selection of this time point, it is "fixed" – and constantly shown within the Detail View. Afterwards, the second time point for the Detail View can be chosen in the Overview and changed over time to compare it to the primarily selected time point.

The second aspect of interaction, the selection of individual nodes, has been described in Section 3.2, where the visualization technique was introduced that is used as the Detail View: One or more nodes from the reaction network can be selected manually or via selection scripts. In the multiple view concept, the selections can be used in a broader context than for the inspection of a single time point. Highlighting the related high level feature (their node properties) in the Overview instantly reveals their time dependent behavior. In turn, this linking opens up for an additional selection mechanism: The Overview can also serve to adjust node selections, now by brushing their node properties directly in the Overview.

The arrangement of one Overview and table-based representations of two time points in the Detail View integrates well with the concept to share node selections among all views. In addition to the Overview, node selections are highlighted and made adjustable throughout the

visual representations of time points in the Detail View to support the in-depth inspection of the simulation data, which typically involves following one or multiple nodes over time.

The **visual linking** comprises highlighting current time points of interest and node selections throughout multiple views. Node selections in the Overview have to be visually linked to the Detail View. In addition, complexity measures and node properties need to be discriminated in the Overview. To this end, one of the strongest visual cues is used: color. The color scheme assigns specific colors for the two possible node selections, unselected node properties, and complexity measures. By default, red and blue are used for the node selections, unselected node properties in black and complexity measures in green. The colors are reused throughout the visual interface. Further, selected time points are highlighted by underlying yellow boxes in the Overview. The use of colors is exemplified in the image of the overall visualization concept in Figure 4.11.

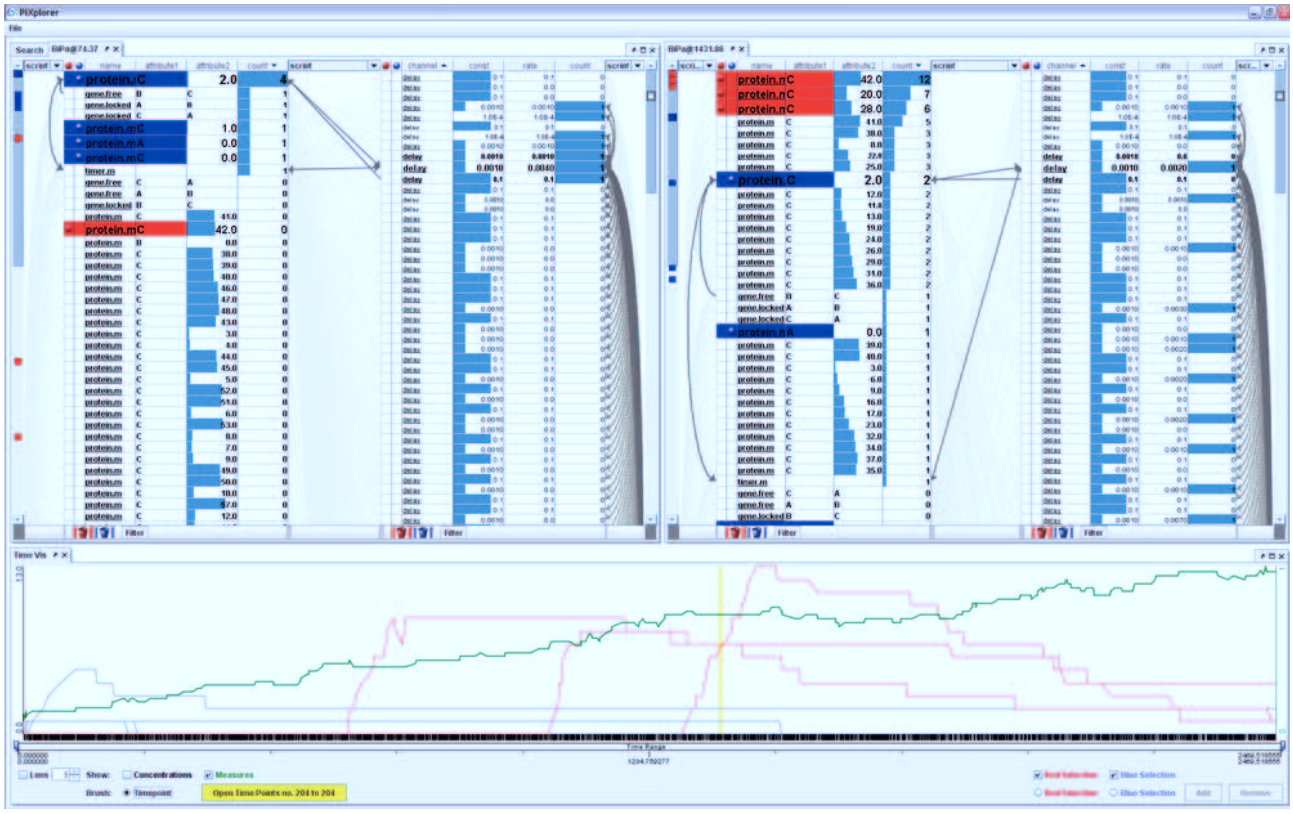


Figure 4.11: The multiple view visualization for simulation data from the Attributed Π-Calculus. The time value plot of the Overview is shown at the bottom and the detailed reaction networks of two time points are shown on the top.

4.2.3 Discussion

The derived multiple view framework concept enables the visual analysis of single-run simulation data from the Attributed Π-Calculus. The multiple view concept is based on the findings from the systematic discussion of visualization concepts in Section 4.1. Generally, the idea is adapted to give a general overview on temporal developments by derived high level features and to analyze identified time points of interest in detail views.

By tailoring important aspects of the visualization concept to the data at hand – including the derivation of appropriate high level features, the design of overview and detail view, and the visual interface to combine these views –, it becomes possible to analyze the complex data, comprising a time series of reaction networks with multiple attributes. The changes over time in structure and attribute values are characterized by appropriate high level features such as complexity measures, which reflect structural changes, and node properties, which capture dynamics in attribute values. In addition, the exploration and comparison of structural relations at specific time points is supported by a graph visualization approach, which is adapted from Section 3.2, as the data characteristics are similar. The highly interactive multiple view concept further enables the investigation of individual nodes or node selections of the reaction networks in all its aspects – developments over time as well as their role in the reaction networks at individual time points. In all views, the user can interactively construct and adjust a selection of nodes, which is then highlighted in the overview in its temporal development and in the detail view as individual entities.

The visualization technique has been practically used to debug the implementation of a simulator of the Attributed Π-Calculus and to check the plausibility of simulation results for cell biological models that were represented with the modeling approach.

4.3 Summary

In this chapter, the challenge to handle large and complex simulation data sets of a single run has been addressed. In the research training school, two approaches have been investigated that generate such data: Next Sub-Volume Method and Attributed Π-Calculus. Based on simulation data from the Next Sub-Volume Method, a systematic discussion aimed at identifying potential visualization concepts to handle all facets of the data: Space, heterogeneity due to states and events, static and dynamic context as well as univariate and multivariate context. Appropriate concepts have been integrated within a multiple view framework for the visual analysis of data from the Next Sub-Volume Method. It comprises three basic views,

each providing the visualization of certain aspects of the data. In the time view, an overview on temporal developments is given by high level features, which have been derived from the original data. From the overview, time points of interest can be identified. State view and event view support the detailed visual analysis and comparison of time points of interest in spatial context. By a close linking and coordination of views, the multiple view framework supports the exploration of the data in its entirety.

The systematic development of the multiple view framework also provides the basis for the visual analysis of complex simulation data from the Attributed Π-Calculus. The resulting data, a time series of reaction networks, is different from the output of the Next Sub-Volume Method. But the idea to combine an overview of temporal developments and details views for selected time points is a valuable approach to handle also this complex simulation data. The resulting multiple view framework accounts for the specific characteristics of the data by deriving appropriate high level features to characterize the temporal developments. Further, overview and detail view as well as the visual interface to coordinate them is tailored to the data at hand.

The visualization concepts represent novel contributions with respect to the complexity of the data that is considered in both approaches. The visual analysis of all facets of the data is supported by the introduced methods. Regarding the application domain, the presented visualization concepts provide a method to access the overwhelming data that results from simulation and, thus, to gain insight into the black box of simulation. By this, the analyst is given the possibility to understand the inner workings of the investigated cellular processes.

Chapter 5

Realization of Visual Support in the Application Domain

This work aims at bridging the gap between the potentials of visualization and their practical application in the data analysis process. In previous chapters, this integration of visualization into the data generating context has been discussed conceptually. The goal of this chapter is to present how the introduced visualization concepts have been realized and integrated into the application of modeling and simulation of cell biological systems, with respect to the process of data generation in the research training school **dIEM oSiRiS**.

In Section 5.1, a visualization component library is developed, which is closely linked to the process of data generation in the research training school. The library realizes the conceptual integration of visualization into the process of data generation. It comprises software tools that implement the visualization concepts developed in this work. These tools are introduced in Section 5.2. Thus, mainly the visual analysis of simulation data is addressed. Further, examples are presented how these tools have been applied in practice. In addition, the library contains visualization tools that have also been developed in the research training school. To support presentation of results, one segment of visual support that has not been addressed so far, a presentation technique based on Illustration Watermarking is introduced in Section 5.3. The technique has the potential to facilitate the communication of findings and results via interactive images. The results of this chapter are summarized in Section 5.4.

5.1 Design of a Visualization Component Library

The conceptual integration of visualization into the process of data generation in the application field has been presented in Section 2.1.2 and summarized in Figure 2.1. It has been realized in the research training school as shown in Figure 5.1. On the left side, the implementation of the data generating process is displayed, comprising multiple data bases. It is described in more detail in Section 5.1.1. The implemented integration of visualization is driven by this realization of the data generating process in the research training school. The design decisions are discussed in Section 5.1.2, which lead to a visualization component library comprising multiple visualization tools. The components of this library, which are shown on the right side in Figure 5.1, implement the visualization concepts derived in this work and in related visualization research in the research training school. They are summarized with respect to the corresponding segment of visual support in Section 5.2. Further, examples for the practical application of these tools are presented.

5.1.1 Process of Data Generation

In order to design the software-based realization of visual support, the existing realization of the process of data generation in the research training school is regarded.

- **Qualitative and quantitative data about the biological system** is collected from the literature or from biological experiments in the laboratory. As biological experiments are elaborate and resulting data requires post-processing before it can be analyzed, the access to the data via a data base is demanded. In the research training school, the experiment data base **eDB** has been developed as a central repository of experiments conducted by experts form biology and biomedicine. It is constantly extended to cover a broader range of experimental methodologies. Further, information about the biological system is contained in commonly accessible **on line data bases**. This knowledge about pathways (e.g., provided by the KEGG data base [KAG+08]), proteins (e.g., [NAB+07]), and genes (e.g., [Gen00]), is an essential part of the modeling process.

- **Formal model structures** are currently provided by two main sources. The model data base **mDB** [KMHK09] mainly focuses on storing models with an underlying biological context. In addition, large models are provided by the **JAMES II** modeling and simulation framework, which is generally suitable for modeling and simulation and not limited to a specific application. It is described more detailed in the context of simulation runs.

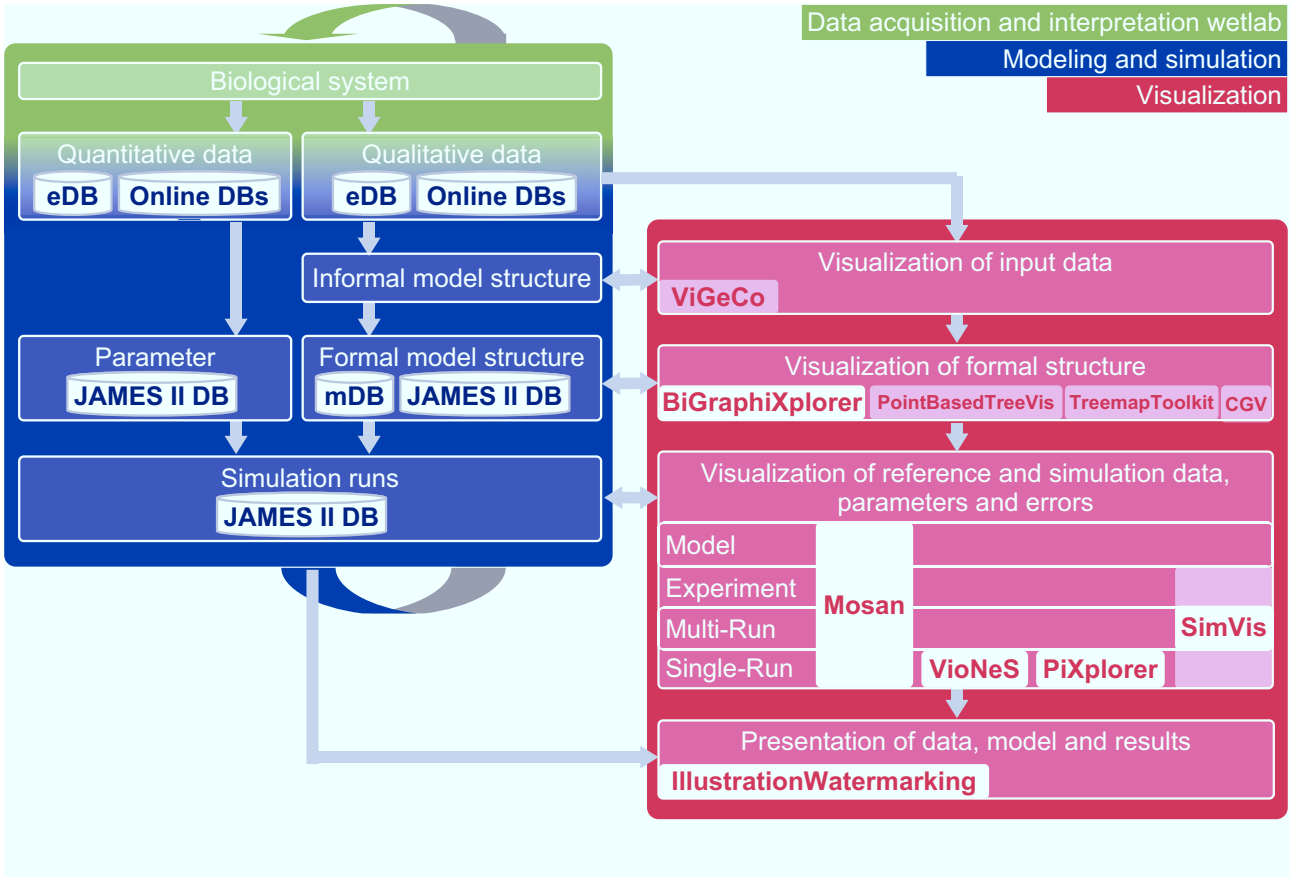


Figure 5.1: Practical integration of data visualization into the context of data generation within the research training school **dIEM oSiRiS**. On the left side, the given realization of the data generating context is presented. It is basically provided by data bases used to store the results of data generation. On the right side, visualization tools are named that implement visualization concepts developed in this work (shown in white boxes) and related visualization research (in transparent boxes), with respect to the segments of visual support and, for the focused segment of visualizing simulation data, to the process levels. The tools are closely coupled to the data bases shown on the left.

- **Simulation runs** are provided by the modeling and simulation framework **JAMES II** [Him07], an internal development of the modeling and simulation group in Rostock. All simulation data, which has been described and used throughout this work, has been generated with JAMES II and made available for analysis by using the facilities of the framework to store data in data bases.

These sources of data in the research training school are summarized on the left side in Figure 5.1, shown in the context of the conceptual process of data generation presented in

Section 2.1.2.

5.1.2 Design Decisions for Integration of Visualization

The results of the data generating process are provided by various sources. This includes data bases and the modeling and simulation framework JAMES II. The integration of visual support in JAMES II is of main interest, as the focus of this work has been set on the visualization of simulation data, which is generated by this framework.

Conceptually, given visualization concepts that follow the idea to tightly integrate data visualization into the data generating process, an on-line visualization of simulation data is supported. "On-line" refers to the instant visual analysis after the data has been generated. It should be noted that "on-line" is distinguished from "on-the-fly" visualization, the latter related to the visualization of streaming data, which is not supported by the developed visualization techniques.

However, a pure on-line visualization limits the potentials of visual analysis in two ways: First, the visual analysis is only available to users that are involved in the process of data generation. But the application domain requires joint efforts of researchers from different fields, which also demands for a data analysis performed by multiple users. Second, the performance of simulation algorithms is a recent research issue in the modeling and simulation community, especially for discrete-event based approaches incorporating stochasticity and spatial resolution, as they are investigated in the application domain. Currently, the execution of simulation is often time-consuming. In these cases, it is useful to separate the execution of simulation from the visual analysis of the simulation data.

To this end, visualization generally accesses simulation data that has been stored in data bases after generation. This circumvents both limitations of the on-line visual analysis: as the simulation does not have to be executed every time it is analyzed, multiple users can access results from the data bases and the temporal efforts for data generation is reduced. Nevertheless, the strengths of on-line visualization are still provided, because the data can be retrieved immediately after results have been generated and stored. But most of all, the developed visualization concepts, with their focus on integrating data visually in the data generating context, provide all necessary information about the data generation even if the visual analysis is carried out independent from the execution of the simulation. Facilities to store simulation data along with the context of data generation such as models and experiment descriptions are provided as part of the JAMES II framework. These capabilities, denoted as **JAMES-DB** in Figure 5.1, are used to provide the linkage between the generation of simulation

data and visualization.

In consequence, resulting data is provided by various data bases in the application domain. The visualization concepts necessary to deal with the diversity of data cannot be integrated within a single software tool. Thus, the visualization concepts are implemented as independent software tools, each supporting specific data sets, which appear at the specific stages of the modeling and simulation work flow. The visualization tools, further described in Section 5.2, are organized with respect to the segments of visual support from Section 2.1.2. According to the data characteristics, each visualization tool has to be linked to the corresponding data bases that provide the data. The tools form a component library for visual support in the application domain, which is shown on the right side in Figure 5.1.

This framework design is suited to cope with the challenges of visual support in the application domain: a multitude of data sets appears, provided by various data sources, and requiring a broad range of visualization concepts. In the component library, tailored implementations are subsumed into one framework. Further, it provides the necessary flexibility to add new visualization methods and to link them to data bases as required. In this regard, the visualization component library has the potential to provide comprehensive visual support in the application domain.

5.2 Visualization Tools

The visualization component library comprises JAVA-based implementations of the visualization concepts developed in this work. In addition, complementary visualization research, especially related to graph visualization [Sch10], has been carried out in the research training school, also resulting in several visualization tools. In the following, the resulting set of available visualization tools is presented with respect to the segments of visual support, which have been identified in Section 2.1.2: the visualization of input data, of formal model structures, and of simulation and reference data. A summary of tools in the library is given in Figure 5.1.

5.2.1 Visualization of Input Data

For the visualization of input data, comprising information about the biological system, various visualization techniques are available from the literature, which have been implemented in specific tools. It has not been the aim of this work to integrate them into the visualization component library. An exemplary visualization of experimental data, which has been developed in the context of the research training school, is the interactive visualization tool **ViGeCo**

[THS07, TS08]. It supports an interactive exploration of similarities among genes determined from micro array data by visualizing combinations of genes (Figure 5.2).

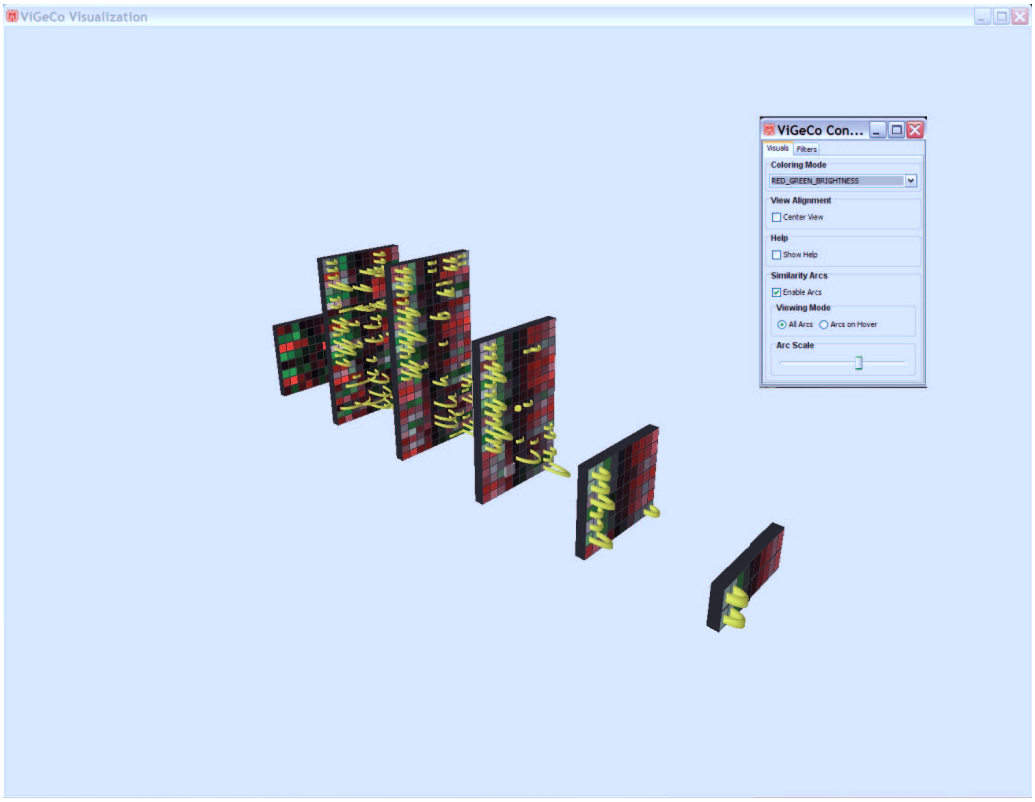


Figure 5.2: Visualization of micro array data with ViGeCo [THS07, TS08], a tool for the visualization of gene combinations, in order to explore similarities among genes. The visualization is based on heat maps, which are commonly accepted in the application domain. Each panel shows gene combinations containing a specific number of genes. Similarity arcs, shown in yellow, indicate gene combinations with similar values.

5.2.2 Visualization of Formal Model Structure

The visualization of formal model structures is highly related to graph visualization. In the course of this work, a visualization technique for large models of reaction networks has been presented as joint work within the research training school. The concepts have been realized in the **BiGraphiXplorer** tool. Beyond large models, the tool is applicable to visualize other large graphs that appear in the biological domain, as demonstrated in the following example.

Considering the thousands of biological entities of a cell, the visualization of known information about all these entities is challenging. As one example, the human metabolic network

[DBJ+07], as it is stored in the BIGG data base [SBRG], comprises 2764 chemical compounds and 3311 chemical reactions. 17518 directed links between chemical compounds and chemical reactions exist. Further, various attributes are given for compounds and reactions, ranging from charges to bibliographical references. The BiGraphiXplorer does not only support the visualization of the reaction network including attributes. It further enables, with its interactive selection functionality, the identification of functional subunits in the network. A screen shot of the BiGraphXplorer visualizing the BIGG data base (SBML snapshot from 20-DEC-2007) has been shown in Figure 3.9.

More details about the practical application of the BiGraphiXplorer are given in [Sch10].

Further, this field of visualization, especially with consideration of large graphs, is in the focus of the dissertation by Hans-Jörg Schulz [Sch10], which has been developed complementary to this thesis within the research training school. Tools that arose from this work, e.g., **Point-Based Visualization** [SHS09] and the **Treemap Toolkit**, are also available in the component library. In this regard, also **CGV** (Coordinated Graph Visualization) [TAS09], a highly interactive tool for visualization of large graphs developed in the computer graphics group in Rostock, is part of the library.

5.2.3 Visualization of Simulation Data

The visualization of simulation data has been in the focus of this work. Four different process levels have been identified in Section 2.3 – model, experiment, multi-run, and single-run –, each requiring specific visualization concepts. Within the research training school, the developed visualization techniques are available as software tools. The following tools provide complete implementations of the visualization concepts as they have been described in this work.

In addition, applied practical use of the visualization tools in the application domain is presented.

5.2.3.1 Mosan

The tool implements the tailored visualization concepts for the process levels of stochastic simulation data – *Model*, *Experiment* and *Multi-Run Simulation Data* – as described in Section 3.1. The integration within one tool supports the linkage among process levels, thus enabling a transition from one process level to the next during the visual analysis sequence.

The tool has already been applied in several application scenarios, including:

- **Visual analysis and presentation of biological experiments**

The visualization tool has been practically applied for the visual analysis and comparison of biological experiments conducted in the laboratory. Especially the close coupling of experimental results with a presumed model structure has been considered as very helpful to confirm or reject hypotheses about the model behavior. The provided comparison of multiple experiments has been used to communicate findings gained in the laboratory to other researchers, as the basis for discussions about future research. The screen shot in Figure 5.3 shows the visual analysis of multiple biological experiments with Mosan.

- **Face validation**
  The process of validation, which accompanies the modeling and simulation process, requires the application of both computational and visual methods. For face validation, which refers to visual inspections of obtained results, the visualization tool has been used. Specifically, it was used to validate a simulator for Π-Calculus models, by checking whether generated simulation data corresponds to simulation data from a reference simulator.

In addition to these application examples that have already been carried out in practice, a future application scenario is to tightly couple the visualization with ongoing data base research to retrieve models and experimental data. In this regard, the visualization concepts will be used to **present results of data base retrieval**. This way, powerful functionality to extract information about cell biological systems, either as simulation models or as experimental data, is supplemented by an intuitive and easily accessible visual presentation of search results. Current data base research aims at retrieving and ranking models based on annotations of models, to identify model components with similar biological meaning [KMHK09]. In **dIEM oSiRiS**, such functionality is implemented in the model data base mDB. Further, the goal is to extract relevant experimental data for a model by searching for experimental data that has been collected for one or multiple components of the model. This functionality is integrated into the experiment data base eDB.

With its tailored concepts for the process levels identified in the application domain, Mosan is able to:

- compare models by their structure
  Ranked models, which are returned as the result of a data base query, can be compared to find the best match with the user's demands.

- visualization of experimental data in the context of the model

Figure 5.3: Mosan visualizing experimental data from three biological experiments in the context of a presumed model structure.

> For a given model, related experimental data comprising one or multiple experiments is visually linked to the model.

The implementation of this integration is currently carried out in the research training school by coupling Mosan, the model data base mDB, and the experiment data base eDB.

5.2.3.2 VioNeS

VioNeS implements the multiple view framework for the visual analysis of the Next Sub-Volume Method as introduced in Section 4.1. Thus, the tool supports the visualization of complex single-run simulation data with spatial context. The tool has been employed for debugging

purposes, to check plausibility of results and correctness of implementations with respect to the Next Sub-Volume Method. A screen shot of the tool was shown in Figure 4.8.

5.2.3.3 PiXplorer

The PiXplorer is the realization of visualization concepts for the analysis of simulation data from the Attributed Π-Calculus, which have been presented in Section 4.2. Similar to VioNes, it operates on the process level of single run simulation data. It has been applied to debug a simulator of the Attributed Π-Calculus and to check results for plausibility, which was exemplified in Figure 4.11.

5.2.3.4 SimVis

SimVis [DMG+05, Dol04] is an external, powerful visualization tool, which has been made available in the research training school for visual analysis purposes. It supports the interactive exploration of large time dependent and multivariate data sets. Originally, it has been designed for spatial data, but can also be applied to other data that comprises multiple data items per time point.

Resulting form the concepts developed in Section 3.3, the tool has been complemented by a new statistical view. The view supports the analysis of statistical properties of interactively generated subsets. Beyond, the tool is applicable for visual analysis at different process levels, ranging from comparison of multiple experiments, over the proposed application for multi-run analysis to the analysis of spatial, single-run simulation data.

The proposed visualization of statistical properties of interactively generated subsets has been used to evaluate the dependence of the simulated behavior of a cellular process on an unknown parameter of the model.

The unknown reaction rate arose from the transformation of a model given as ordinary differential equations (ODE) into a model for discrete stochastic simulation, which requires additional parameters. Simulation experiments were conducted for nine different values of the reaction rate, ten runs were performed for each experiment. As a first step, interactive brushing was applied to partition the simulation data in subsets comprising data generated for low and high values of the reaction rate. Brushing was performed in scatter plots, which explicitly display the unknown reaction rate on one axis, as shown in Figure 5.4. The statistical view of these two subsets reveals significant differences in temporal developments between each of the two subsets and the whole data set. In general, higher reaction rates lead to a higher number of bound molecules over the complete simulated time. A more detailed comparison of the influence

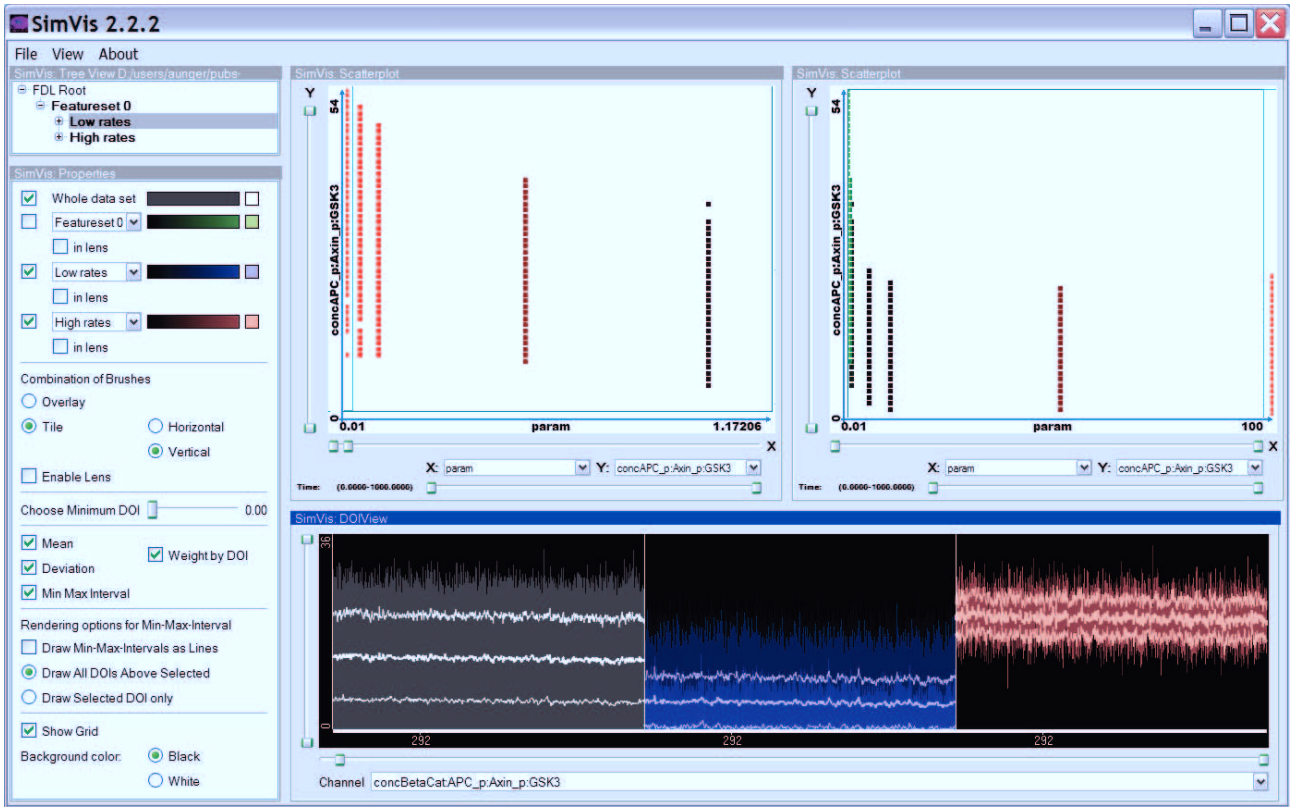
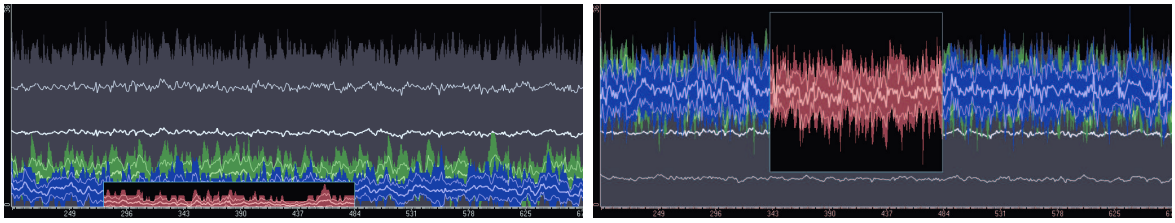


Figure 5.4: Screen shot of the SimVis system. The two scatter plots on top have been used to generate two subsets of the data according to the value of the unknown reaction rate. The statistical view at the bottom visualizes the impact of the reaction rate on the number of molecules over the simulated time. Statistics for the whole data set (left) are shown as gray, the subset derived from simulation runs with low reaction rates (middle) as blue and the subset derived from simulation runs with high reaction rates (right) as red curves.

of reaction rates by comparing statistical properties of different low rates and different high rates further revealed that the system is expected to be sensitive to low reaction rates (Figure 5.5(a)), while high reaction rates do not cause strong variations of the system behavior (Figure 5.5(b)). This valuable insight for the further modeling of the system exemplifies that the visual analysis is not only helpful to understand validated models, but can help in the modeling of incomplete models.



(a) View for lower reaction rates. Red curves (in lens): reaction rate 0.01, blue curves: reaction rate 0.05, green curves: reaction rate 0.1. In general, lower reaction rates cause lower quantities of the reaction product B

(b) View for higher reaction rates. Red curves (in lens): reaction rate 10, blue curves: reaction rate 50, green curves: reaction rate 100. No dependence between the reaction rate and the number of resulting molecules B can be seen.

Figure 5.5: Statistical view for the number of B molecules for different reaction rates. The statistics for the complete data set are shown in gray shades.

5.3 Presentation with Interactive Images based on Illustration Watermarking

In this section, a concept is described to realize interactive images, which are proposed as a presentation technique in the application domain. Based on techniques from digital watermarking, the concept is called **Illustration Watermarking** [SIDS03]. Interactive images supplement the communicative power of static images, the preferred medium for presentation in the application domain, by additional descriptive information, which is shown on user's demand. Since the auxiliary information is only displayed upon request, a permanent occlusion of parts of the image content is avoided. Hence, static images are turned into media, which can be interactively explored.

The concept to augment images with additional data is not new. Specific software such as Macromedia Flash [RD04], Tool-Book [SS04], or MS PowerPoint [Wem03] provide integrated data storage of complex interactive illustrations within a single data file - but the display of those illustrations relies on the use of the software. More generally applicable, several graphics format specifications provide storage of annotation data. This includes JPX files [JTC04, JW04], TIFF [Ass92], PNG [SW04], SVG [Gro03] or X3D [Che04]. However, these formats do not directly link image objects and their descriptions. In addition to integrated data storage within one file, examples for separated storage of image and auxiliary data are Web pages, including data with links pointing to separate files, or data base management systems, treating auxiliary data as a collection of related data items.

The concept of Illustration Watermarking combines several advantages of related technologies from the literature: A compact storage of image and auxiliary data is enabled within a single file that can be easily created, distributed, and accessed. Common file formats can be used and the file size does not depend on the volume of auxiliary data, because it is integrated into the image content. Thus, the image can be accessed with common image viewers, although special software is required to explore auxiliary data. As another advantage, auxiliary data is not limited to text. Any binary data can be embedded, including for example images, animation, or audio. Most importantly, auxiliary data can be linked to specific regions of the image, thus providing a region based annotation of the image content. Two aspects must be considered to realize the concept of interactive images:

- storage of image and auxiliary data
- interaction techniques to explore interactive images.

In the following paragraph, a data storage technique based on Illustration Watermarking is described, which adapts the amount of embedded data to the image content. Results of a user study are summarized, in which it could be shown that the novel technique outperforms a standard technique in terms of image quality. Aiming at an easy exploration of embedded auxiliary, concepts to localize annotated image regions and retrieve properties of auxiliary data are introduced in the succeeding paragraph, before the usability for the application domain is discussed. The presented results have be published in [SUS08].

Illustration Watermarking for 2-D digital images The idea of Illustration Watermarking is to use techniques from digital watermarking to provide local annotations of the image content. Digital watermarking is "the practice of hiding a message about an image, audio clip, video clip, or other work of media within that work itself" [CMB02]. It is often related to security aspects, which does not apply for the intended use of Illustration Watermarking. In interactive images, the auxiliary information is commonly accessible and provides additional value for both the author and the user. Hence, other aspects then security and robustness are important. Illustration Watermarking focuses, instead, on techniques to embed a significant amount of data (*high capacity*), while perceivable modifications of the image content should be avoided (*transparency*). Further, for an easy distribution of the illustration, watermark recovery should be carried out without the original image (*blind detection*). Also, to annotate specific objects of the image, the watermark message is embedded locally (*content-based*).

A simple approach (introduced in [KM92] and [TRvS+93]) is to replace the least significant bits in the spatial domain. It allows embedding comparably large amounts of data without perceivable image modifications. For increased robustness, various techniques embed data in the frequency domain [CW01] or in the wavelet domain [MU01]. Similar to the intended application, techniques that embed data by replacing parts of the image content have been used for the annotation of image objects before, for example in [BKM+02] or [RVEP04]. In both approaches, the auxiliary data contained within the image is very limited, restricted to indices pointing to a data base and short text labels, respectively.

As none of the approaches supports the criteria of high capacity, transparency, blind detection, and content-based embedding at the same time, a new adaptive technique is developed in the following. The focus is set on lossless color images in the RGB color space with 24 bits (8 per color component) per pixel. The approach is in general based on data embedding in the pixel's least significant bits (LSB). To ensure blind detection, it must be known which bits of the image content have been replaced and what kind of data is encoded. With the standard LSB approach, bits containing annotation data can be retrieved if the annotated region and the size of the encoded bit stream are known. To identify the annotated region, its outline pixels are marked in the least significant bit of the blue color component. Bit stream size and data type are stored at a prominent location within the region.

The basic LSB approach assumes a constant capacity among colors and pixels and distributes the replaced bits equally among pixels and color components. However, prior tests have shown that the capacity depends on image features, mainly on texture and color. To maximize the capacity, the number of bits replaced in the color components of pixels has to be adapted to the image features. In consequence, additional information about the distribution of embedded bits is required in both the encoding and decoding process to ensure blind detection. This additional information is provided in a newly introduced *Capacity Map*. To build the exact same capacity map in both the encoding and decoding process, image data used to build the capacity map has to be separated from image content data modified in the encoding process. Tests have shown that it is a good choice to restrict data encoding to the four least significant bits of each color component. Thus, the four most significant bits maintain the original values and are suited for generating the capacity map. The remaining four bits of every color component are available to encode the watermark, leading to a maximum capacity of 12 bits per pixel, four in every color component.

In the following, it is described how the image features texture and color are used to build an image's capacity map.

- **Texture** The observation can be made that the more homogeneous the texture, the easier content modifications are visible and the less is the capacity. To account for the homogeneity of the texture, an estimate of the capacity based on entropy values [DD02] of small image blocks has been adapted for color images. Further, only the four most significant bits can be used for capacity computation due to the requirement of blind detection. Regarding every color component separately, the computed entropy values are used to assign capacities of either four, three, or two bits of the color component to all pixels in the image block. Thresholds, chosen as $\frac{1}{2}$ of the maximum entropy for 4 bits capacity and $\frac{1}{4}$ of maximum entropy for 3 bits capacity, are based on informal test results.

- **Color** Colors are stored in the RGB color space. The embedding of data is conducted in this color space. Hence, it has to be ensured that the modifications of the color in RGB color space, which result from the embedding of data, cannot be perceived. But distances in RGB color space do not correspond to distances perceived by the human eye. These distances have to be determined in uniform color spaces, for example, CIELAB and CIELUV [Sto03]. These color spaces represent perceived distances for similar colors with Euclidean distance values.

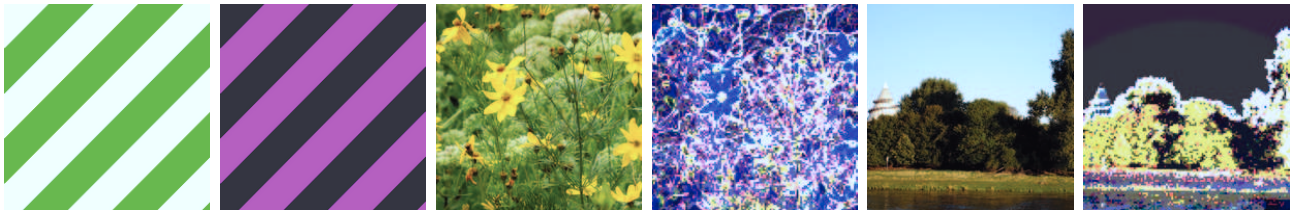
  A color's capacity can be described as the maximum distance to another color in RGB, which is still equally perceived, according to the distance in the uniform color space. To define a given color's capacity, the perceived distances to other colors are computed that might result from the embedding of data. Starting with close colors, the distance in RGB is increased as long as the resulting colors are still equally perceived. As only the four most significant bits of the color components can be used to compute the capacity, the same capacity is assigned to all colors sharing the four most significant bits. Here, the smallest capacity of all colors is assigned as the capacity.

To combine texture and color analysis, both capacity values are computed for every pixel's color components and the higher value is chosen as the capacity.

The adaptive technique has been evaluated in a user study with two main intentions. On the one hand, the reliability of computed image capacities was tested. To this end, computed image capacities and empirically determined image capacities were compared. On the other hand, it was tested whether the adaptive technique leads to higher capacities than the standard technique. For that purpose, empirically determined capacities of both approaches were compared.

During the study, 112 participants were asked to detect watermarked regions in images.

Images differed in image type, encoding technique, and size of encoded bit stream, in order to determine the maximum capacity for constant image quality. Three image types were included in the study. *GreenPattern* (Figure 5.6(a)) consists of large homogeneous regions with only two colors. The computed capacity (Figure 5.6(b)) results from color information. In contrast, the texture in *Flowers* (Figure 5.6(c)) is very inhomogeneous. The computed capacity (Figure 5.6(d)) mainly results from texture information. *Landscape* (Figure 5.6(e), capacity map in Figure 5.6(f)) was included to evaluate whether adapting the number of replaced bits among pixels increases the overall capacity. Thus, watermarked regions spanned both homogeneous and inhomogeneous image regions.

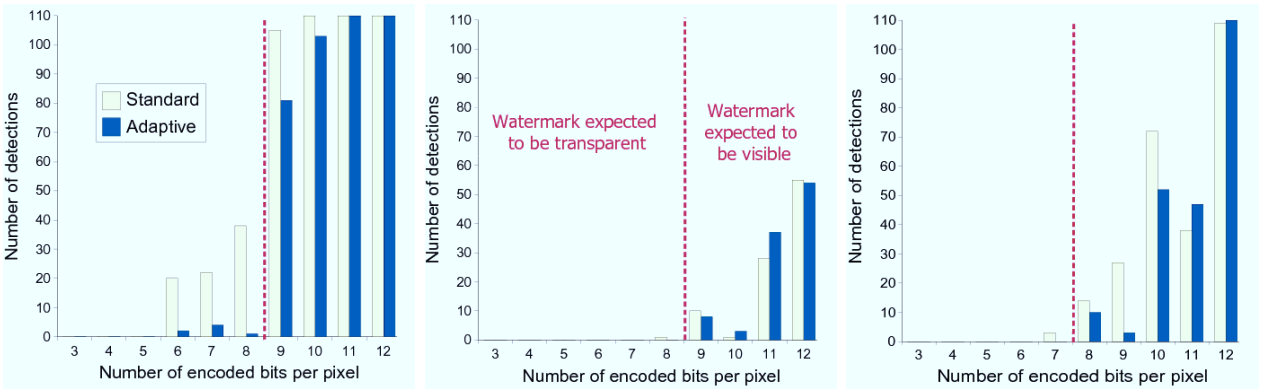


(a) (b) (c) (d) (e) (f)

Figure 5.6: Image types in the study and computed capacity maps. (a) image type *GreenPattern*, (b) capacity map for *GreenPattern*, (c) image type *Flowers*, (d) capacity map for *Flowers*, (e) image type *Landscape*, (f) capacity map for *Landscape*. For better representation, values in the capacity maps, originally one to four bits per color component and pixel, have been scaled to a range between 0 and 255, brighter values indicating higher capacities.

From the results of the study (Figure 5.7), it could be shown that the capacity map is a good estimate of the capacity of an image. Only for one of the three image types (*GreenPattern*), few participants detected watermarks, which were expected to be transparent according to the computed capacity. Comparing standard and adaptive approach, a statistically significant lower number of watermark detections was observed for the adaptive approach than for the standard LSB method for *GreenPattern* (standard: 520, adaptive: 415) and *Landscape* (standard: 263, adaptive: 222). Thus, the adaptive distribution of the bit stream among the pixel's color components leads to a higher capacity especially in homogeneous regions, resulting in a capacity of 7 to 8 bits per pixel.

Exploration of Auxiliary Data Another important aspect of interactive images is the exploration of the interactive content by the user. First, the user needs hints about the - per



(a) Green Pattern (b) Flowers (c) Landscape

Figure 5.7: Results of the user study: For every image type, the number of participants who detected the watermark is shown for different volumes of encoded data (in bits per pixel). The standard LSB technique is shown in yellow bars, the adaptive approach in blue bars. The red dashed line indicates the expected visibility threshold.

default hidden - auxiliary data contained in the image. Upon the user's request of auxiliary data, an appropriate layout has to be applied.

In the following, the former of these two aspects is addressed: interaction methods that lead the user to annotation data contained in the image. This includes the need for visual cues to communicate existence and properties of auxiliary data. Annotation data should be easily detectable and a clear correlation between cue and augmented region is necessary. Important properties of the auxiliary data are location and size of regions being annotated, kind of auxiliary data (for example, text, image, or sound) as well as amount of data and accordance to user interest. While most of these properties can be retrieved from the encoded data, accordance to user interest is assumed to be externally defined. Two approaches to support the viewer in the exploration of interactive images are presented, *Informative Cursor* and *Meta Previewer*.

The prototype of an Informative Cursor is the known mouse cursor, which changes its style to indicate that a certain action can be performed. In addition to the current position, the Informative Cursor utilizes arrows to lead the user to regions with associated data. Graphical attributes encode properties of annotation data, as shown in Figure 5.8(a). The type of data is indicated by letters (I for image, T for text, A for combinations) in the arrowhead. The lengths of the arrows indicate the distance to the annotated region. The amount of associated data corresponds to the thickness of the arrows. A darker color of the arrow stands for a higher correlation with the user's interest. Further, the closest annotated image region is emphasized

by a highlighted contour, so the user can quickly reach the region and demand auxiliary data.

The *Meta Previewer* displays all annotated regions of an image by small markers and shows information about the auxiliary data in a focused region. It follows the idea of Magic Lenses (e.g., [BSP+93]). The Meta Previewer is placed in front of the image to display additional information associated with the covered image region. It consists of a small rectangular shape which includes a down-scaled version of the original image with markers for all annotated image regions. The focused region is magnified by distortion (compare with Fish Eye Views, e.g., [LA94, CM01, GS03] or non-linear magnification [KR97]). In addition, a frame provides further information about the meta data associated with the focused region. An example is given in Figure 5.8(b). Within the Meta Previewer located at the cursor position, the blossom is magnified, its contour highlighted. The right bar stands for the accordance to the user's interest. The small bars on top show the amount of auxiliary data differentiated by type. Further, small spheres indicate positions of other annotated image regions.

(a) Informative Cursor (b) Meta Previewer

Figure 5.8: Exploration of annotated images. The Informative Cursor (a) points to annotated image regions with arrows, encoding further properties of the annotation data by graphical attributes of the arrow. The Meta Previewer (b) magnifies the focused region and displays properties of related auxiliary data, while also indicating positions of other annotated regions. Both Informative Cursor and Meta Previewer are permanently updated while being moved across the image.

Discussion With the concept of region-based *Illustration Watermarking*, 2-D raster images can be turned into interactive media, providing additional information about objects in the image on the user's demand. This is a valuable approach for the application domain. To present the results of a modeling and simulation project, complex findings need to be communicated. As the amount of information that can be contained in a single static image is limited, images only contain the most important facts to keep the presentation as clear as possible. Terms are abbreviated; complex concepts are strongly abstracted and shown by graphical icons. To understand these presentations, often further information about the single entities in the image is needed. This can comprise textual descriptions as well as images providing additional detail. Further, a major goal in the application is to understand the dynamics of cell biological systems. These are difficult to capture within a static image. With interactive images, changes over time can be conveyed by describing them in additional data, directly related to the entities where change happens.

Such annotation data cannot be shipped within a few encoded bits. Hence, the proposed approach in Section 5.3 is valuable as it maximizes the capacity of the image by distributing the encoded data according to image features. Further, a reasonable estimate of the image capacity is given. This is useful in the encoding process, because the user is informed about the amount of data that can be embedded without degrading the image quality. In combination with the presented techniques to explore auxiliary data, Illustration Watermarking has the potential to help users in the communication of their ideas. This idea has to be further investigated by future research.

5.4 Summary

In this chapter, the implementation of visual support in the research training school has been described. Following the conceptual integration of data visualization into the data generating context as presented in Section 2.1.2, a visualization component library has been designed that covers visual support at all four segments of necessary visual support: the visualization of input data, of the formal model structure, of simulation and reference data, and presentation. The component library is linked to the data generating context by interfaces to data bases, which store results of data generation in the research training school. The library is flexible in the sense that it can be extended by additional methods that will be developed in the future.

In its current state, the library contains tools that realize the visualization concepts introduced in this work. With respect to presentation of results, a methodology has been introduced

to store annotation data within an image, thus turning an image into an interactive medium. Beyond, the library provides visualization tools that were developed in related visualization research.

By various practical examples, the usefulness of the derived visualization concepts for the intended application domain could be exemplified.

Chapter 6

Conclusion

In practical applications, visualization is often used for presentation purposes, rather than as a means of data analysis. Considering the overwhelming and increasing volumes of data that appear in many domains, the process of analytical problem solving can significantly benefit from visual support, which is the aim of the field of visual analytics. In this regard, it has been a main motivation of this work to bring visualization closer to the demands of the application. These demands are highly application specific.

In this work, it is investigated how visual support can be provided for the specific application domain of modeling and simulating cell biological systems. One major contribution is the conceptual integration of visual support into the process of modeling and simulation. Further, novel visualizations are derived to support the visual analysis of stochastic simulation data with respect to the underlying simulation process. This chapter summarizes the contributions of this work and gives a perspective on open visualization problems.

The research has been conducted within the research training school **dIEM oSiRiS**, which provides the environment for interdisciplinary research.

6.1 Summary

This dissertation presents four main contributions: First, visualization is conceptually integrated as a means of data analysis into the application domain of modeling and simulating cell biological systems. Second, a general taxonomy for stochastic simulation data allows the development of visualization concepts that support the exploration of the simulation process rather than single data sets. Further, novel concepts for the visual analysis of complex simulation data are systematically developed. At last, a visualization component library is derived that

realizes the integration of the proposed visualizations in the application context and is flexible to contain future developments.

Conceptual integration of visualization in the application domain In a systematic approach, the need for visual support in the application domain is analyzed. Using the work flow of modeling and simulation as the basis, the process of data generation is identified, which leads to four main segments of visual support:

- visualization of input data
- visualization of formal model structure
- visualization of simulation and reference data
- presentation of findings

As the result of an analysis of existing visual support on the one hand and visualization requirements on the other hand for each segment, the focus of this work is set on the visualization of data from discrete event based simulation. The visual analysis of the resulting data is essential in a modeling and simulation project. From a visualization viewpoint, the visualization is challenging because understanding the data requires its visual integration into the context of data generation. Further, the complexity of the data is seldom considered in existing visualization methods.

Exploration of the simulation process based on tailored visualizations for process levels The general taxonomy of the data generating process for stochastic simulation data comprises the four levels *Model*, *Experiment*, *Multi-Run Simulation Data* and *Single-Run Simulation Data*. Tailored visualization concepts for all process levels, explicitly visualizing data in the context of data generation, lead to visual support at various abstraction levels: the comparison of experiments at the *Model Level*, the evaluation of one experiment comprising multi-run simulation data at the *Experiment Level*, the exploration of multiple runs at the *Multi-Run Level*, and the detailed analysis of a one run at the *Single-Run Level*.

Basis for the visualization is a view on one experiment that integrates model structure, experiment description and multi-run simulation data. Due to its size, the multi-run data is highly abstracted to the most general trends over time. The global comparison of heterogeneous local value ranges in the multivariate data and among experiments is supported by a novel color scale, which is derived on a heuristic applied to the value ranges in the data. It is suitable for

sequential and diverging data scales, thus being applicable to a broad range of data sets. With its ability to convey data ranges as well as uncovered value ranges within the global range, it communicates valuable information about the data values at a glance. The view on one experiment serves as the basis on the model, experiment, and multi-run level. At the model level, a visual linking of multiple experiment views allows for the comparison of experiments. To address the analysis of multiple runs, an overview and detail concept links basic view and one view showing the single runs as the overview with detail views on single run data.

In these concepts, the model representation is reduced to the most important aspects to understand the simulation data. For a detailed analysis, a table-based visualization supports the evaluation of large models in all their facets, including multiple variables. Further, aiming at more elaborate visual multi-run analysis, a novel view for the analysis of statistical properties of interactively generated subsets is developed and integrated into a multiple view framework.

These visualization concepts are novel contributions in two ways: First, data is visually integrated within the context of data generation, with the necessary consideration of process information identified from the process levels. Second, tailored visualizations for each process level support the exploration of the complete simulation process, which goes beyond the visualization of single simulation data sets. This is expected to facilitate the visual analysis of simulation data derived in a modeling and simulation project: all necessary information to comprehend simulation data is presented in the visualization and multiple data sets can be analyzed at different levels of detail. Further, the analysis process is supported for users that have not been involved in data generation, as they can reconstruct how the data was generated from the visualization.

Visual analysis of complex simulation data Further visualizations at the single-run level aim at handling the complexity of simulation data in the application domain, with its heterogeneous, time dependent, multivariate, and spatial context. The systematic evaluation of potential visualization concepts for each of these aspects leads to a multiple view framework. To explore data over time, it comprises an overview on time dependent high level features. These high level features are abstracted from spatial context to support a general impression on the development of the simulation data over time. Low level features, comprising multivariate data in spatial context, are conveyed in specific views for event data and state data. In these views, single time points are analyzed or multiple time points are compared. The coordination and linking of these views, integrated within a highly interactive multiple view framework, supports the exploration of the data in all facets.

These visualization concepts, developed with respect to data generated from the simulation algorithm Next Sub-Volume Method, have been adapted to simulation data from a novel modeling and simulation approach, the Attributed Π-Calculus, which results in a time series of reaction networks. Again, time dependent high level features, in this case the structural complexity and values of variables over time, are shown in an overview. For a single time point, details about the reaction network are shown with a table-based visualization technique. The comparison of time points is provided by the linking of multiple detail views. The interactive framework of multiple coordinated views accounts for the specific views required for the simulation data.

Realization as a visualization component library The visualization concepts are realized as software tools within a visualization component library. The library is closely linked to the process of data generation in the research training school. Interfaces to available data bases enable the access to the results of the data generating context.

The software based realization closely follows the conceptual integration of visualization into the application domain. In this regard, the library allows for interactive exploration at all four identified segments of visual support. According to the focus of this work, mainly visualization tools for simulation data are currently provided. At the other segments, exemplary visualizations are integrated that have been derived in the context of the research training school. For presentation purposes, interactive images are proposed. Enabling local annotations of images, a technique based on digital watermarking supports the embedding of reasonable amounts of annotation data within an image and the exploration of these annotations.

First examples for the practical use of the proposed visualization concepts in the research training school indicate that the application based development of visualization methods is beneficial for data analysts.

6.2 Future Work

Developed in the first phase of the research training school **dIEM oSiRiS**, this work serves as a starting point to develop visual support for the modeling and simulation of cell biological systems. Numerous challenges that arise in the still emerging application field have not been addressed in this work, but need to be considered to provide comprehensive visual support.

With respect to the work flow of modeling and simulation, which has been the basis of this work, the highly relevant validation step has not been considered in this work. Here, a further

tight integration of computational and visual methods in the sense of visual analytics has to be provided, as validation relies on both statistical tests and visual inspections.

The focus of this work has been set on the visual analysis of simulation data. Analogue, the development of visual methods is necessary at the other three segments of visual support that have been identified from the process of data generation in the application. Although approaches have been presented at all segments, the range of visualization techniques is by far not complete.

Also within the focused segment of the visualization of simulation data, open visualization challenges appear. New multi-level modeling and simulation approaches require novel visualization concepts. These new approaches in the application field are able to couple multiple heterogeneous models, given at different abstraction levels. Further, the comparison of simulation data and reference data generated by real-world experiments as well as the comparison of simulation data from different modeling and simulation approaches are subject of future research, as they help to evaluate the quality and accuracy of different methods. The ground for these techniques is provided with the taxonomy of process levels for stochastic simulation data, but additional research is required to bridge the gaps between heterogeneous data sets from different sources.

The introduced visualization component library provides a single-sided linkage to the process of data generation: Data generated in the process of modeling and simulation is handled by the visualization techniques. Considering the generation of a multitude of complex data sets, each requiring elaborate analysis, and further considering multiple users involved in the decision making process, it will become more and more important to record the results of visual analysis and make them accessible. Visual analysis usually aims at the identification of relevant pieces of information contained in the data. In the focused segment of visualization of simulation data, process levels provide a natural way to record such results of visual analysis as a path through the process hierarchy. In Figure 6.1, such a path through the hierarchy is indicated by highlighting the respective elements at each process level and the links between process levels in red. Starting at the model level comprising stochastic simulation data, visualization concepts supporting all process levels enables the identification of experiments of interest. From the simulation data of an experiment, representative runs are identified from a visual multi-run analysis. At the single run level, time points of interest are identified, indicated by dashed lines in the time value plot at the single run level in Figure 6.1. The storage of such paths along with corresponding visual representations and additional information provided by the users is left to future research. In the research training school, the investigation of paths through the

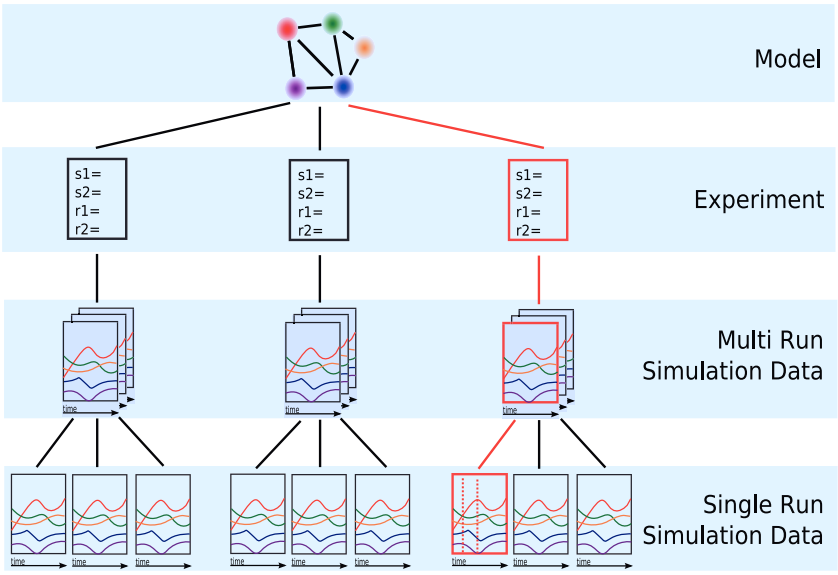


Figure 6.1: Exemplification of recording the results of visual analysis as a path through the hierarchy of process levels. At the single run level, the path is prolonged by the identification of time points of interest

visualization work flow is a recent issue, mainly addressed by Marc Streit. First ideas are also presented in [Sch10].

As another aspect, the appearance of uncertainty in the visualization domain has not been regarded in this work. Uncertainty occurs at all stages of data generation in the application context. The available information about the cell biological system is often incomplete. Experimental data is the result of highly susceptible processes in the laboratory, followed by an analysis sequence to transform raw imaging data into meaningful numbers. As experimental data is the basis of the modeling and simulation process, associated uncertainty may proceed through the work flow. In fact, one main reason for the modeling and simulation of cell biological systems is that the underlying system is not directly observable in all its facets. Hence, uncertainty is an integral part of the application domain. This has to be communicated by visual methods in order to enable informed decisions.

All these open problems, appearing in the context of the application domain, represent known challenges in the field of visualization. They are summarized in the following.

- combination of computational and visual methods in the sense of visual analytics
- visual integration and comparison of data from heterogeneous data sources
- visualization concepts to handle complex data from multi-level modeling and simulation

- appropriate recording of results of visual analysis
- visualization of uncertainty

Specifically the visualization of data from multi-level modeling and simulation and the visualization of uncertainty are investigated in the second phase of the research training school.

Bibliography

[AB04] Stephen S. Andrews and Dennis Bray. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys Biol*, 1(3-4):137–151, 2004.

[ABM+07] Wolfgang Aigner, Alessio Bertone, Silvia Miksch, Christian Tominski, and Heidrun Schumann. Towards a conceptual framework for visual analytics of time and time-oriented data. In *WSC '07: Proceedings of the 39th conference on Winter simulation*, pages 721–729, Piscataway, NJ, USA, 2007. IEEE Press.

[AMCH07] Hiroshi Akiba, Kwan-Liu Ma, Jacqueline H. Chen, and Evatt R. Hawkes. Visualizing multivariate volume data from turbulent combustion simulations. *Computing in Science and Engg.*, 9(2):76–83, 2007.

[AMYS+09] Boanerges Aleman-Meza, Yihai Yu, Heinz-Bernd Schüttler, Jonathan Arnold, and Thiab R. Taha. Kinsolver: A simulator for computing large ensembles of biochemical and gene regulatory networks. *Comput. Math. Appl.*, 57(3):420–435, 2009.

[Ass92] Adobe Developers Association. *TIFF 6.0 Specification.* 1992.

[Bal98] Osman Balci. Verification, validation, and accreditation. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 41–4, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[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.

[BCM+09] Robert Byrnes, Dawn Cotter, Andreia Maer, Joshua Li, David Nadeau, and Shankar Subramaniam. An editor for pathway drawing and data visualization in the biopathways workbench. *BMC Systems Biology*, 3(1):99, 2009.

[BCPS03] Charles Baker, Sheelagh Carpendale, Przemyslaw Prusinkiewicz, and Michael Surette. Genevis: simulation and visualization of genetic networks. *Information Visualization*, 2(4):201–217, 2003.

[BDBS04] Eric Baehrecke, Niem Dang, Ketan Babaria, and Ben Shneiderman. Visualization and analysis of microarray and gene ontology data with treemaps. *BMC Bioinformatics*, 5(1):84, 2004.

[BETT99] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing - Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[BHK+05] Ljudmilla Borisjuk, Mohammad-Reza Hajirezaei, Christian Klukas, Hardy Rolletschek, and Falk Schreiber. Integrating data from biological experiments into metabolic networks with the dbe information system. *In Silico Biology*, 5:93–102, 2005.

[BJSCNN01] Jerry Banks, II. John S. Carson, Barry L. Nelson, and David M. Nichol. *Discrete-Event System Simulation*. Prentice Hall, 3rd edition, 2001.

[BKM+02] Nikolaos V. Boulgouris, Ioannis Kompatsiaris, Vasileios Mezaris, Dimitrios Simitopoulos, and Michael G. Strintzis. Segmentation and content-based watermarking for color image and image region indexing and retrieval. *EURASIP J. Appl. Signal Process.*, 2002(1):418–431, 2002.

[BM04] Vladimir Batagelj and Andrej Mrvar. Pajek - analysis and visualization of large networks. *Graph Drawing Software*, pages 77–103, 2004.

[BMGK08] Aaron Barsky, Tamara Munzner, Jennifer Gardy, and Robert Kincaid. Celebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Transactions on Visualization and Computer Graphics*, 14:1253–1260, 2008.

[BPC07] Gleb Beliakov, Ana Pradera, and Tomaso Calvo. *Aggregation Functions: A Guide for Practitioners*, volume 221 of *Studies in Fuzziness and Soft Computing*. Springer, 2007.

[BRRB93] Lawrence D. Bergman, Jane S. Richardson, David C. Richardson, and Frederick P. Brooks, Jr. View: an exploratory molecular visualization system with user-definable interaction sequences. In *SIGGRAPH '93: Proceedings of the 20th*

annual conference on Computer graphics and interactive techniques, pages 117–126, New York, NY, USA, 1993. ACM.

[Bru06] Stefan Bruckner. Interactive illustrative volume visualization techniques for exploration and communication. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 6, New York, NY, USA, 2006. ACM.

[BS06] Frank T. Bergmann and Herbert M. Sauro. SBW - a modular framework for systems biology. In *WSC '06: Proceedings of the 37th conference on Winter simulation*, pages 1637–1645. Winter Simulation Conference, 2006.

[BSG+09] S. Bruckner, V. Solteszova, M.E. Groller, J. Hladuvka, K. Buhler, J.Y. Yu, and B.J. Dickson. Braingazer - visual queries for neurobiology research. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1497–1504, Nov.-Dec. 2009.

[BSP+93] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, New York, NY, USA, 1993. ACM.

[BSRG06] Michael Baitaluk, Mayya Sedova, Animesh Ray, and Amarnath Gupta. Biologicalnetworks: visualization and analysis tool for systems biology. *Nucleid Acids Research*, 34:466–471, 2006.

[BST03] Bobby-Joe Breitkreutz, Chris Stark, and Mike Tyers. Osprey: a network visualization system. *Genome Biology*, 4:R22, 2003.

[BW09] Romain Bourqui and Michel A. Westenberg. Visualizing temporal dynamics at the genomic and metabolic level. *Information Visualisation, International Conference on*, 0:317–322, 2009.

[BWCT09] Evan P. Boswell, John T. Wessler, Urska Cvek, and Marjan Trutschl. The storage, retrieval, and visualization of biological pathway data. *Information Visualisation, International Conference on*, 0:301–306, 2009.

[BWK00] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *AVI '00: Proceedings of*

the working conference on Advanced visual interfaces, pages 110–119, New York, NY, USA, 2000. ACM Press.

[CCC+05] Wingyan Chung, Hsinchun Chen, Luis G. Chaboya, Christopher D. O'Toole, and Homa Atabakhsh. Evaluating event visualization: a usability study of coplink spatio-temporal visualizer. *International Journal of Human-Computer Studies*, 62(1):127–157, 2005.

[CFF07] Yaroslav Chushak, Brent Foy, and John Frazier. Biomolecular network simulator: software for stochastic simulation of cellular biological processes. In *SpringSim '07: Proceedings of the 2007 spring simulation multiconference*, pages 345–349, San Diego, CA, USA, 2007. Society for Computer Simulation International.

[CHC+05] Trevor M. Cickovski, Chengbang Huang, Rajiv Chaturvedi, Tilmann Glimm, H. George E. Hentschel, Mark S. Alber, James A. Glazier, Stuart A. Newman, and Jesus A. Izaguirre. A framework for three-dimensional simulation of morphogenesis. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(4):273–288, 2005.

[Che04] Vladimir Geroimenkoand Chaomei Chen. *Visualizing Information Using SVG and X3D: XML-Based Technologies for the XML-Based Web.* Springer-Verlag, 2004.

[CKC05] Paul Craig, Jessie Kennedy, and Andrew Cumming. Animated interval scatterplot views for the exploratory analysis of large-scale microarray time-course data. *Information Visualization*, 4(3):149–163, 2005.

[CM01] M. S. T. Carpendale and Catherine Montagnese. A framework for unifying presentation space. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 61–70, New York, NY, USA, 2001. ACM.

[CMB02] Ingemar Cox, Matthew Miller, and Jeffrey Bloom. *Digital Watermarking: Principles & Practice.* Academic Press, 2002.

[CW01] Brian Chen and Gregory W. Wornell. Quantization index modulation methods for digital watermarking and information embedding of multimedia. *J. VLSI Signal Process. Syst.*, 27(1/2):7–33, 2001.

[DBJ+07] Natalie C. Duarte, Scott A. Becker, Neema Jamshidi, Ines Thiele, Monica L. Mo, Thuy D. Vo, Rohith Srivas, and Bernhard Ø. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proc. of the National Academy of Sciences of the USA*, 104(6):1777–1782, 2007.

[DBS06] Anastasia Deckard, Frank T. Bergmann, and Herbert M. Sauro. Supporting the SBML layout extension. *Bioinformatics*, page btl520, 2006.

[DD02] Marc Van Droogenbroeck and Jérôme Delvaux. An entropy based technique for information embedding in images. In *IEEE Signal Processing Symposium*, pages 81–84, Leuven, Belgium, 2002.

[DH02] Helmut Doleisch and Helwig Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. *Journal of WSCG*, 10(1):147–154, 2002.

[Die09] Janko Dietzsch. *Verfahren zur Genexpressionsanalyse*. PhD thesis, Universität Tübingen, 2009.

[dLVvL06] Wim de Leeuw, Pernette Verschure, and Robert van Liere. Visualization and analysis of large data collections: a case study applied to confocal microscopy data. *IEEE Transactions on Visualization and Computer Graphics*, 12:1251–1258, 2006.

[DMG+05] Helmut Doleisch, Michael Mayer, Martin Gasser, Peter Priesching, and Helwig Hauser. Interactive feature specification for simulation data on time-varying grids. In Thomas Schulze, Graham Horton, Bernhard Preim, and Stefan Schlechtweg, editors, *SimVis*, pages 291–304. SCS Publishing House e.V., 2005.

[DMS+05] Pawan K. Dhar, Tan Chee Meng, Sandeep Somani, Li Ye, Kishore Sakharkar, Arun Krishnan, Azmi B.M. Ridwan, Sebastian Ho Kok Wah, Mandar Chitre, and Zhu Hao. Grid Cellware: the first grid-enabled tool for modelling and simulating cellular processes. *Bioinformatics*, 21(7):1284–1287, 2005.

[DMS+08] Tim Dwyer, Kim Marriott, Falk Schreiber, Peter Stuckey, Michael Woodward, and Michael Wybrow. Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1293–1300, 2008.

[Dol04] Helmut Doleisch. *Visual Analysis of Complex Simulation Data using Multiple Heterogenous Views*. PhD thesis, Technical University of Vienna, 2004.

[EE04] J. Elf and M. Ehrenberg. Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Systems Biology*, 1(2):230–236, 2004.

[EFZ08] Florian Erhard, Caroline Friedel, and Ralf Zimmer. Fern - a java framework for stochastic simulation and evaluation of reaction networks. *BMC Bioinformatics*, 9(1):356, 2008.

[EHC03] Sol Efroni, David Harel, and Irun R. Cohen. Toward Rigorous Comprehension of Biological Complexity: Modeling, Execution, and Visualization of Thymic T-Cell Maturation. *Genome Research*, 13(11):2485–2497, 2003.

[EHC05] Sol Efroni, David Harel, and Irun R. Cohen. Reactive animation: Realistic modeling of complex dynamic systems. *Computer*, 38:38–47, 2005.

[EHK+04] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Aaron E. Lefohn, Christof Rezk Salama, and Daniel Weiskopf. Real-time volume graphics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 29, New York, NY, USA, 2004. ACM.

[ESBB98] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868, 1998.

[FHK+09] David C. Y. Fung, Seok-Hee Hong, Dirk Koschützki, Falk Schreiber, and Kai Xu. Visual analysis of overlapping biological networks. *Information Visualisation, International Conference on*, 0:337–342, 2009.

[FKRE09] Martin Falk, Michael Klann, Matthias Reuss, and Thomas Ertl. Visualization of signal transduction processes in the crowded environment of the cell. In *PACIFICVIS '09: Proceedings of the 2009 IEEE Pacific Visualization Symposium*, pages 169–176, Washington, DC, USA, 2009. IEEE Computer Society.

[FMJ+08] A. Funahashi, Y. Matsuoka, A. Jouraku, M. Morohashi, N. Kikuchi, and H. Kitano. Celldesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, 96(8):1254–1265, 2008.

[FR91] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21:1129–1164, 1991.

[Gen00] Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[Gro03] World Wide Web Consortium (W3C): SVG Working Group. *Scalable Vector Graphics (SVG) 1.1 Specification*. 2003.

[GS03] Carl Gutwin and Amy Skopik. Fisheyes are good for large steering tasks. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 201–208, New York, NY, USA, 2003. ACM.

[GSS01] Donna Gresh, Frank Suits, and Yuk Yin Sham. Case study: an environment for understanding protein simulations using game graphics. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 445–448, Washington, DC, USA, 2001. IEEE Computer Society.

[Gut08] Enrico Gutzeit. Effektive Visualisierung der Next Subvolume Methode. Master's thesis, University of Rostock, 2008.

[HBMS03] Harry Hochheiser, Eric H. Baehrecke, Stephen M. Mount, and Ben Shneiderman. Dynamic querying for pattern identification in microarray and genomic data. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03)*, pages 453–456, Washington, DC, USA, 2003. IEEE Computer Society.

[HDLT05] Matthew Hibbs, Nathaniel Dirksen, Kai Li, and Olga Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics*, 6(1):115, 2005.

[HFE05] Johan Hattne, David Fange, and Johan Elf. Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, 21(12):2923–2924, 2005.

[HHW+09] Zhenjun Hu, Jui-Hung Hung, Yan Wang, Yi-Chien Chang, Chia-Ling Huang, Matt Huyck, , and Charles DeLisi. Visant 3.5: multi-scale network visualization, analysis and inference based on the gene ontology. *Nucleid Acids Research*, 37:W115–W121, 2009.

[HHWN02] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[Him07] Jan Himmelspach. *Konzeption, Realisierung und Verwendung eines allgemeinen Modellierungs-, Simulations- und Experimentiersystems - Entwicklung und Evaluation effizienter Simulationsalgorithmen*. PhD thesis, Universität Rostock, 2007.

[HLNZ05] Matthew Holford, Naixin Li, Prakash Nadkarni, and Hongyu Zhao. Vitapad: visualization tools for the analysis of pathway data. *Bioinformatics*, 21:1596–1602, 2005.

[HMT09] Curtis Huttenhower, Sajid Mehmood, and Olga Troyanskaya. Graphle: Interactive exploration of large, dense graphs. *BMC Bioinformatics*, 10(1):417, 2009.

[HR07] Simon Hardy and Pierre N. Robillard. Visualization of the simulation data of biochemical network models: a painted petri net approach. In *SCSC: Proceedings of the 2007 summer computer simulation conference*, pages 802–808, San Diego, CA, USA, 2007. Society for Computer Simulation International.

[HS04] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[HSG+06] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jurgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. COPASI–a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[HSPWR04] Susan L. Havre, Mudita Singhal, Deborah A. Payne, and Bobbie-Jo M. Webb-Robertson. Pquad: Visualization of predicted peptides and proteins. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 473–480, Washington, DC, USA, 2004. IEEE Computer Society.

[INM+05] Florian Iragne, Macha Nikolski, Bertrand Mathieu, David Auber, and David Sherman. ProViz: protein interaction visualization and exploration. *Bioinformatics*, 21(2):272–274, 2005.

[JKS06] Bjorn Junker, Christian Klukas, and Falk Schreiber. Vanted: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, 7(1):109, 2006.

[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.

[JLNUar] Mathias John, Cédric Lhoussaine, Joachim Niehren, and Adelinde M. Uhrmacher. The Attributed Pi Calculus with Priorities. *Transactions in Computational Systems Biology*, 2010 (to appear).

[JM97] Michael Jünger and Petra Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1(1):1–25, 1997.

[JSS+10] Mathias John, Hans-Jörg Schulz, Heidrun Schumann, Adelinde M. Uhrmacher, and Andrea Unger. Constructing and visualizing reaction networks from pi-calculus models. *Theor. Comput. Sci.*, 2010.

[JTC04] Joint Technical Committee ISO/IEC JTC1, editor. *Information Technolog - JPEG 2000 Image Coding System: Extensions*. International Standard ISO/IEC 15444-2, 2004.

[JW04] James S. Janosky and Rutherford W. Witthus. Using JPEG2000 for Enhanced Preservation and Web Access of Digital Archives - Case Study. In *IS&T's 2004 Archiving Conference*, volume 1, pages 145–149. IS&T - The Society for Imaging Science and Technology, 2004.

[KAG+08] Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi. Kegg for linking genomes to life and the environment. *Nucl. Acids Res.*, 36(suppl_1):D480–484, 2008.

[KHK+05] Edda Klipp, Ralf Herwig, Axel Kowald, Christoph Wierling, and Hans Lehrach. *Systems Biology in Practice - Concepts, Implementation and Application*. WILEY-VCH Verlag, 2005.

[Kin04] Robert Kincaid. Vistaclara: an interactive visualization for exploratory analysis of dna microarrays. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 167–174, New York, NY, USA, 2004. ACM.

[KJKC07] Robert Kosara, T.J. Jankun-Kelly, and Eleanor Chlan. Information visualization contest 2007. In *Proc. of IEEE InfoVis*, 2007.

[KM92] Charles Kurak and John McHugh. A Cautionary Note on Image Downgrading. In *Proceedings of the Eighth Annual Computer Security Applications Conference*, San Antonio, TX, USA, 1992.

[KMHK09] Dagmar Köhn, Carsten Maus, Ron Henkel, and Martin Kolbe. *Data Integration in the Life Sciences*, volume 5647 of *Lecture Notes in Computer Science*, chapter Towards Enhanced Retrieval of Biological Models through Annotation-Based Ranking, pages 204–219. Springer Berlin / Heidelberg, 2009.

[KMR+09] Matthias Keil, Richard J. Marhofer, Andreas Rohwer, Paul M. Selzer, Jürgen Brickmann, Oliver Korb, and Thomas E. Exner. Molecular visualization in the rational drug design process. *Publ. in: Frontiers in Bioscience 14 (2009), pp. 2559-2583*, 2009.

[KR97] T. Alan Keahey and Edward L. Robertson. Nonlinear magnification fields. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, pages 51–58, Washington, DC, USA, 1997. IEEE Computer Society.

[Krü08] Ulrike Krüger. Visualisierung zeitabhängiger Simulationsdaten in verschiedenen Aggregationsstufen. Master's thesis, University of Rostock, 2008.

[KSH04] Robert Kosara, Gerald N. Sahling, and Helwig Hauser. Linking scientific and information visualization with interactive 3d scatterplots. In *In Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG*, pages 133–140, 2004.

[LA94] Ying K. Leung and M. D. Apperley. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, June 1994.

[LCN98] Barthold Lichtenbelt, Randy Crane, and Shaz Naqvi. *Introduction to volume rendering.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[LJA+08] Xuan Liu, Jipu Jiang, Oluwafemi Ajayi, Xu Gu, David Gilbert, and Richard Sinnott. Bionessie - a grid enabled biochemical networks simulation environment. *Studies in Health Technology and Informatics*, 138:147–157, 2008.

[LTG+10] Michael D. Lieberman, Sima Taheri, Huimin Guo, Fatemeh Mirrashed, Inbal Yahav, Aleks Aris, and Ben Shneiderman. Visual exploration across biomedical databases. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(PrePrints), 2010.

[LWSK04] Alexander Lüdemann, Daniel Weicht, Joachim Selbig, and Joachim Kopka. PaVESy: pathway visualization and editing system. *Bioinformatics*, 20:2841–2844, 2004.

[LYC+06] Dong-Yup Lee, Choamun Yun, Ayoun Cho, Bo Kyeng Hou, Sunwon Park, and Sang Yup Lee. WebCell: a web-based environment for kinetic modeling and dynamic simulation of cellular networks. *Bioinformatics*, 22(9):1150–1151, 2006.

[MFPD07] Johannes Mandel, Hendrik FuSZ, Niall Palfreyman, and Werner Dubitzky. Modeling biochemical transformation processes and information processing with narrator. *BMC Bioinformatics*, 8(1):103, 2007.

[MIP10] Tommaso Mazza, Gennaro Iaccarino, and Corrado Priami. Snazer: the simulations and networks analyzer. *BMC Systems Biology*, 4(1):1, 2010.

[Mis06] Kazuo Misue. Drawing bipartite graphs as anchored maps. In *APVis '06: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 169–177, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[MJJ+05] Kresimir Matkovic, Mario Jelovic, Josip Juric, Zoltan Konyha, and Denis Gracanin. Interactive visual analysis and exploration of injection systems simulations. In *Visualization, 2005. VIS 05. IEEE*, pages 391–398, 2005.

[MJR+05] Phillip McClean, Christina Johnson, Roxanne Rogers, Lisa Daniels, John Reber, Brian M. Slator, Jeff Terpstra, and Alan White. Molecular and Cellular Biol-

ogy Animations: Development and Impact on Student Learning. *Cell Biology Education*, 4(2):169–179, 2005.

[MKO+08] Philipp Muigg, Johannes Kehrer, Steffen Oeltze, Harald Piringer, Helmut Doleisch, Bernhard Preim, and Helwig Hauser. A Four-level Focus+Context Approach to Interactive Visual Analysis of Temporal Features in Large Scientific Data. *Computer Graphics Forum*, 27(3):775–782, 2008.

[MPC+06] Elaine Meng, Eric Pettersen, Gregory Couch, Conrad Huang, and Thomas Ferrin. Tools for integrated sequence-structure analysis with UCSF Chimera. *BMC Bioinformatics*, 7(1):339, 2006.

[MRS+09] Heimo Müller, Robert Reihs, Stefan Sauer, Kurt Zatloukal, Marc Streit, Alexander Lex, Bernhard Schlegl, and Dieter Schmalstieg. Connecting genes with diseases. *Information Visualisation, International Conference on*, 0:323–330, 2009.

[MSR+09] Sebastian Mirschel, Katrin Steinmetz, Michael Rempel, Martin Ginkel, and Ernst Dieter Gilles. PROMOT: modular modeling for systems biology. *Bioinformatics*, 25(5):687–689, 2009.

[MSS+08] Ion I. Moraru, James C. Schaff, Boris M. Slepchenko, Michael Blinov, Frank Morgan, Anuradha Lakshminarayana, Fei Gao, Ye Li, and Leslie M. Loew. The virtual cell modeling and simulation software environment. *IET Syst Biol*, 2(5):352–362, 2208.

[MT08] Thomas Maiwald and Jens Timmer. Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics*, 24(18):2037–2043, 2008.

[MU01] P. Meerwald and A. Uhl. A survey of wavelet-domain watermarking algorithms. In *SPIE Symposium, Electronic Imaging, Conference on Security and Watermarking of Multimedia Contents, San Jose, USA*, 2001.

[NAB+07] Darren Natale, Cecilia Arighi, Winona Barker, Judith Blake, Ti-Cheng Chang, Zhangzhi Hu, Hongfang Liu, Barry Smith, and Cathy Wu. Framework for a protein ontology. *BMC Bioinformatics*, 8(Suppl 9):S1, 2007.

[NMS+08] Nicolas Le Novère, Stuart Moodie, Anatoly Sorokin, Michael Hucka, Falk Schreiber, Emek Demir, Huaiyu Mi, Yukiko Matsuoka, Katja Wegner, and Hi-

roaki Kitano. Systems biology graphical notation: Process diagram level 1. available at sbgn.org, 2008.

[OD07] Sarah P. Otto and Troy Day. *A Biologist's Guide to Mathematical Modeling in Ecology and Evolution*. Princeton University Press, 2007.

[PB07] Bernhard Preim and Dirk Bartz. *Visualization in Medicine: Theory, Algorithms, and Applications (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 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.

[PFvdW+04] Tim Peeters, Mark Fiers, Huub van de Wetering, Jan-Peter Nap, and Jarke J. van Wijk. Case Study: Visualization of annotated DNA sequences. In *VisSym 2004, symposium on Visualization*, pages 109–114, Konstanz, Germany, 2004.

[PKvI+08] Alexander R. Pico, Thomas Kelder, Martijn P. van Iersel, Kristina Hanspers, Bruce R. Conklin, and Chris Evelo. Wikipathways: Pathway editing for the people. *PLoS Biology*, 6(7):e184, July 2008.

[PMS+98] Catherine Plaisant, Richard Mushlin, Aaron Snyder, Jia Li, Dan Heller, and Ben Shneiderman. Lifelines: using visualization to enhance navigation and analysis of patient records. *Proceedings of the AMIA Symposium*, pages 76–80, 1998.

[POM+09] Bernhard Preim, Steffen Oeltze, Matej Mlejnek, Eduard Groeller, Anja Hennemuth, and Sarah Behre. Survey of the Visual Exploration and Analysis of Perfusion Data. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):205–220, March-April 2009.

[PP95] Hans-Georg Pagendarm and Frits H. Post. Comparative visualization - approaches and examples. In *Visualization in Scientific Computing*. Springer, 1995.

[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. *Inf. Process. Lett.*, 80(1):25–31, 2001.

[PvW08] A. Johannes Pretorius and Jarke J. van Wijk. Visual Inspection of Multivariate Graphs. *Computer Graphics Forum*, 27(3):967–974, 2008.

[Qel07] Ermir Qeli. *Information Visualization Techniques for Metabolic Engineering*. PhD thesis, Universität Marburg, 2007.

[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.

[RD04] Robert Reinhardt and Snow Dowd. *Macromedia Flash MX 2004 Bible*. John Wiley & Sons, Ltd., 2004.

[RK04] Ursula Rost and Ursula Kummer. Visualisation of biochemical network simulations with SimWiz. *Systems Biology, IEE Proceedings*, 1(1):184–189, 2004.

[RKDB06] Vidal J. Rodriguez, Jaap A. Kaandorp, Maciej Dobrzynski, and Joke G. Blom. Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in Escherichia coli. *Bioinformatics*, 22(15):1895–1901, 2006.

[Roj04] Igor Rojdestvenski. Virtual reality and knowledge spaces: examples and applications in molecular biology. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 49–56, New York, NY, USA, 2004. ACM.

[RVEP04] Yuriy Rytsar, Slava Voloshynovskiy, Frederic Ehrler, and Thierry Pun. Interactive Segmentation with Hidden Object-Based Annotations: Toward Smart Media. In *Proceedings of the SPIE, Storage and Retrieval Methods and Applications for Multimedia*, volume 5307, pages 29–37, 2004.

[Sar00] Robert G. Sargent. Verification, validation, and accreditation: verification, validation, and accreditation of simulation models. In *WSC '00: Proceedings of the 32nd conference on Winter simulation*, pages 50–59, San Diego, CA, USA, 2000. Society for Computer Simulation International.

[Sar06] Purviben Bharatkumar Saraiya. *Insight-based studies for pathway and microarray visualization tools*. PhD thesis, Virginia Polytechnic Institute and State University, 2006.

[SBRG] San Diego Systems Biology Research Group, University of California. BiGG: Database of biochemically, genetically and genomically structured genome-scale metabolic network reconstructions. http://bigg.ucsd.edu.

[Sch03] Falk Schreiber. Comparison of metabolic pathways using constraint graph drawing. In *APBC '03: Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003*, pages 105–110, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[Sch10] Hans-Jörg Schulz. *Explorative Graph Visualization*. PhD thesis, University of Rostock, 2010.

[SFB94] Maureen C. Stone, Ken Fishkin, and Eric A. Bier. The Movable Filter as a User Interface Tool. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 306–312, 1994.

[SFHJ03] Nameeta Shah, Vladimir Filkov, Bernd Hamann, and Kenneth I. Joy. Genebox: Interactive visualization of microarray data sets. In *Proc. of the International Conference in Mathematics and Engineering Techniques in Medicine and Biological Scienes (METMBS)*, 2003.

[SGL08] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.

[SGM+07] John Sharko, Georges G. Grinstein, Kenneth A. Marx, Jianping Zhou, Chia-Ho Cheng, Shannon Odelberg, and Hans-Georg Simon. Heat map visualizations allow comparison of multiple clustering results and evaluation of dataset quality: Application to microarray data. In *IV '07: Proceedings of the 11th International Conference Information Visualization*, pages 521–526, Washington, DC, USA, 2007. IEEE Computer Society.

[SHS09] Hans-Jörg Schulz, Steffen Hadlak, and Heidrun Schumann. Point-Based Tree Representation: A new Approach for Large Hierarchies. In *PacificVis'09: Proceedings of the IEEE Pacific Visualization Symposium*, pages 81–88, 2009.

[SHZ+07] Nathan Salomonis, Kristina Hanspers, Alexander Zambon, Karen Vranizan, Steven Lawlor, Kam Dahlquist, Scott Doniger, Josh Stuart, Bruce Conklin, and Alexander Pico. Genmapp 2: new features and resources for pathway analysis. *BMC Bioinformatics*, 8(1):217, 2007.

[SIDS03] Henry Sonnet, Tobias Isenberg, Jana Dittmann, and Thomas Strothotte. Illustration watermarks for vector graphics. *Computer Graphics and Applications, Pacific Conference on*, 0:73, 2003.

[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.

[SKKS08] Marc Streit, Michael Kalkusch, Karl Kashofer, and Dieter Schmalstieg. Navigation and exploration of interconnected pathways. *Computer Graphics Forum (EuroVis 2008)*, 27(3):951–958(8), 2008.

[SM00] Heidrun Schumann and Wolfgang Müller. *Visualisierung. Grundlagen und allgemeine Methoden*. Springer, 2000.

[SM07] Sean Sedwards and Tommaso Mazza. Cyto-Sim: a formal language model and stochastic simulator of membrane-enclosed biochemical processes. *Bioinformatics*, 23(20):2800–2802, 2007.

[SMO+03] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13:2498–2504, 2003.

[SND04] Purvi Saraiya, Chris North, and Karen Duca. An Evaluation of Microarray Visualization Tools for Biological Insight. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pages 1–8, Washington, DC, USA, 2004. IEEE Computer Society.

[SND05] Purvi Saraiya, Chris North, and Karen Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, 4(3):191–205, 2005.

[SS02a] Jinwook Seo and Ben Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, 35(7):80–86, 2002.

[SS02b] Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics*. Morgan Kaufman, 2002.

[SS04] Inc. SumTotal Systems. *ToolBook Instructor 2004 - User Guide*. 2004.

[SS05] Falk Schreiber and Henning Schwobbermeyer. MAVisto: a tool for the exploration of network motifs. *Bioinformatics*, 21(17):3572–3574, 2005.

[SSM+03] Boris M. Slepchenko, James C. Schaff, Ian Macara, , and Leslie M. Loew. Quantitative cell biology with the virtual cell. *Trends in Cell Biology*, 13(11):570–576, November 2003.

[SSP+09] Theresa Scharl, Gerald Striedner, Florentina Potschacher, Friedrich Leisch, and Karl Bayer. Interactive visualization of clusters in microarray data: an efficient tool for improved metabolic analysis of E. coli. *Microbial Cell Factories*, 8(1):37, 2009.

[Sto03] Maureen C. Stone. *A field guide to digital color*. A K Peters Ltd., 2003.

[SUS08] Henry Sonnet, Andrea Unger, and Heidrun Schumann. Interactive images using illustration watermarks: Techniques, studies, and applications. In *WSCG'2008, The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media, Plzen, Czech Republic*, 2008.

[SW04] Joint Technical Committee ISO/IEC JTC1 SC24 and PNG Group (W3C). *Portable Network Graphics (PNG) Specification*. International Standard ISO/IEC 15948:2003, 2nd edition, 2004.

[SWC+02] M. Sultan, D.A. Wigle, C.A. Cumbaa, M. Maziarz, J. Glasgow, M.S. Tsao, and I. Jurisica. Binary tree-structured vector quantization approach to clustering and visualizing microarray data. *Bioinformatics*, 18(suppl_1):S111–119, 2002.

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

[TC05] James J. Thomas and Kristin A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.

[TFS08] Christian Tominski, Georg Fuchs, and Heidrun Schumann. Task-Driven Color Coding. In *IV '08: Proceedings of the 2008 12th International Conference Information Visualisation*, pages 373–380, Washington, DC, USA, 2008. IEEE Computer Society.

[THS07] Christian Tominski, Clemens Holzhüter, and Heidrun Schumann. Interactive poster: Visualization of gene combinations. Poster at IEEE Conference on Information Visualization in Sacramento, USA, 2007.

[THT+99] Masaru Tomita, Kenta Hashimoto, Kouichi Takahashi, Tom Shimizu, Yuri Matsuzaki, Fumihiko Miyoshi, Kanako Saito, Sakura Tanida, Katsuyuki Yugi, J. Craig Venter, and Clyde A. Hutchison. E-cell: software environment for whole-cell simulation. *Bioinformatics*, 15:72–84, 1999.

[TRvS+93] A. Z. Tirkel, G. A. Rankin, R. M. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne. Electronic watermark. In *Digital Image Computing, Technology and Applications (DICTA'93)*, 1993.

[TS08] Christian Tominski and Heidrun Schumann. Visualization of Gene Combinations. In *IV '08: Proceedings of the 2008 12th International Conference Information Visualisation*, pages 120–126, Washington, DC, USA, 2008. IEEE Computer Society.

[TSM+99] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S. Lander, and Todd R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6):2907–2912, 1999.

[UBJ+07] Andrea Unger, Susanne Biermann, Mathias John, Adelinde M. Uhrmacher, and Heidrun Schumann. Visual support for modeling and simulation of cell biological systems (poster). Winter Simulation Conference, WSC'07, Washington, D.C., USA, December 2007, 2007.

[UGJS09] Andrea Unger, Enrico Gutzeit, Matthias Jeschke, and Heidrun Schumann. VioNeS - Visual Support for the Analysis of the Next Sub-volume Method. In

Information Visualisation, 2009 13th International Conference, volume 0, pages 10–17, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[UMDS08] Andrea Unger, Phillip Muigg, Helmut Doleisch, and Heidrun Schumann. Visualizing Statistical Properties of Smoothly Brushed Data Subsets. In *IV '08: Proceedings of the 2008 12th International Conference Information Visualisation*, pages 233–239, Washington, DC, USA, 2008. IEEE Computer Society.

[US09] Andrea Unger and Heidrun Schumann. Visual support for the understanding of simulation processes. In *PACIFICVIS '09: Proceedings of the 2009 IEEE Pacific Visualization Symposium*, pages 57–64, Washington, DC, USA, 2009. IEEE Computer Society.

[Vel09] Kai Velten. *Mathematical Modeling and Simulation - Introduction for Scientists and Engineers*. WILEY-VCH Verlag, 2009.

[VSR+06] Marc T. Vass, Clifford A. Shaffer, Naren Ramakrishnan, Layne T. Watson, and John J. Tyson. The JigCell Model Builder: A Spreadsheet Interface for Creating Biochemical Reaction Network Models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(2):155–164, 2006.

[VWVS99] Jarke J. Van Wijk and Edward R. Van Selow. Cluster and Calendar Based Visualization of Time Series Data. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 4, Washington, DC, USA, 1999. IEEE Computer Society.

[vZtW05] Jeroen S. van Zon and Pieter Rein ten Wolde. Green's-function reaction dynamics: A particle-based approach for simulating biochemical networks in time and space. *J Chem Phys*, 123(23), 2005.

[WAM01] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 7, Washington, DC, USA, 2001. IEEE Computer Society.

[Wem03] Faithe Wempen. *Microsoft Office - PowerPoint 2003 Bible*. John Wiley & Sons, Ltd., 2003.

[Wil05] Leland Wilkinson. *The Grammar of Graphics*. Springer, 2nd edition, 2005.

[WOCH09] Yong Wan, Hideo Otsuna, Chi-Bin Chien, and Charles Hansen. An interactive visualization tool for multi-channel confocal microscopy data in neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1489–1496, 2009.

[WRH+09] Gunther H. Weber, Oliver Rubel, Min-Yu Huang, Angela H. DePace, Charless C. Fowlkes, Soile V. E. Keranen, Cris L. Luengo Hendriks, Hans Hagen, David W. Knowles, Jitendra Malik, Mark D. Biggin, and Bernd Hamann. Visual Exploration of Three-Dimensional Gene Expression Using Physical Views and Linked Abstract Views. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 6(2):296–309, 2009.

[YISH06] Kentaro Yano, Kazuhide Imai, Akifumi Shimizu, and Takao Hanashita. A new method for gene discovery in large-scale microarray data. *Nucleic Acids Research*, 34(5):1532–1539, 2006.

[YWCND06] Yuting Yang, Eve Syrkin Wurtele, Carolina Cruz-Neira, and Julie A. Dickerson. Hierarchical visualization of metabolic networks using virtual reality. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 377–381, New York, NY, USA, 2006. ACM.

[ZHaD04] Jie Wu Zhenjun Hua and, Joseph Mellor and and Charles DeLisi. Visant: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, 5:17, 2004.

[Zim08] Armin Zimmermann. *Stochastic Discrete Event Systems - Modeling, Evaluation, Applications*. Springer, 2008.

[ZPK00] Bernhard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of Modeling and Simulation - Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2nd edition, 2000.

Selbstständigkeitserklärung

Ich erkläre, dass ich die eingereichte Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Andrea Unger

Rostock, 6. April 2010

Résumé

Personal

Name	Andrea Unger
Place of Living	Rostock
Date of Birth	July 10, 1981
Place of Birth	Blankenburg/Harz, Germany
Nationality	German

Education

since 02/2007	Ph.D. student at the Department of Computer Graphics, University of Rostock, member of research training school **dIEM oSiRiS**
04/2006	Diplom-Ingenieur (Dipl.-Ing.) in Computervisualistik (Computational Visualistics)
10/2000 - 04/2006	Studies at the Faculty of Computer Science, Otto-von-Guericke-Universität Magdeburg
07/2000	Abitur (high school diploma)
08/1992 - 07/2000	Europagymnasium Richard von Weizsäcker in Thale

Thesis Statements

1. Information visualization provides powerful interactive techniques to support data analysis. But in practical applications, visualization is often used for presentation of analysis results, not as an integral means to conduct data analysis. To overcome this gap, visual analytics requires broadening the scope from specific data sets to the process of problem solving, which is dependent on the application domain.

2. For the application of modeling and simulating cell biological systems, a systematic analysis of the application work flow allows the identification of the demands for visual support. At all steps of the work flow, data analysis can significantly benefit from visualization. Four segments of visual support are identified: the visualization of input data, of the formal model structure, of simulation data as well as the presentation of findings and results. To conceptually integrate visualization into the data generating process, the process of data generation and the segments of interactive visual support are closely linked.

3. For each segment of visual support, numerous visualization concepts have been introduced in the literature. Nevertheless, the emerging application field demands the development of complementary visualization techniques throughout the process of problem solving to handle novel challenges in each segment.

4. A specific challenge in the application domain is the visual analysis of simulation data from discrete event based approaches. Often, the data generation is so time consuming that the data analysis has to be uncoupled from the data generation. To make the visualization of simulation data comprehensible, the process of data generation has to be communicated as well. Further, the simulation data is large and complex. Stochasticity requires numerous runs of the simulation. The resulting simulation data contains large numbers of time points, multiple state variables and events as well as spatial context.

5. The necessity to account for the process of data generation in the visualization is considered by a general taxonomy for stochastic simulation data. It captures the data generating

context by four process levels: model, experiment, multi-run data, and single-run data. Each process level provides relevant information on how data has been derived. Further, visualization goals are linked to the levels that aim at different abstractions of the data.

6. The identification of visualization goals at each process level provides the ground to develop tailored visualization concepts. A basic view tightly integrates model structure, experiment description, and highly abstracted multi-run simulation data, which conveys the general trends over time. Adapting this concept at all process levels, multiple visualization goals are supported, ranging from comparison of experiments, over analysis of multiple runs from one experiment to the identification of runs of interest. With additional detail concepts for the visualization of large models and for visual multi-run analysis, the derived concepts support the exploration of the complete simulation process, which goes beyond the analysis of simulation data.

7. Discrete-event based simulation of cell biological system results in large and complex data sets. Data is heterogeneous due to the duality of states and events and further incorporates temporal, spatial, and multivariate context. A systematic analysis of potential visualization concepts leads to the development of a multiple view framework, which can handle the complex data in all facets. Specifically, the spatio-temporal context is broken down into separate views. The applied concept to visualize temporal trends by high level features in one view and details for time points of interest in other views can be adapted for simulation data with other characteristics. In this regard, the approach has been used for simulation data that comprises a time series of reaction networks.

8. Numerous visualization concepts are necessary to fulfill the requirements of visual support in the application domain. To this end, a visualization component library realizes the integration of visual support in the application. The visualization components are linked to the data generating context via interfaces to data bases.

9. The applicability of the novel visualizations for the modeling and simulation of cell biological systems is indicated by various practical examples. The techniques have been used to debug implementations of modeling and simulation approaches, to check the plausibility of results, for face validation, visual analysis and presentation of reference data generated in the laboratory, and for the exploration of large biochemical reaction networks. Future application scenarios include the tight integration of visualization and data base research, thus providing a visual front end for data base retrieval.