Methodology

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
1–20
© The Author(s) 2017
DOI: 10.1177/0037549717704314
journals.sagepub.com/home/sim

Reusing simulation experiment specifications in developing models by successive composition — a case study of the Wnt/β-catenin signaling pathway

Danhua Peng[1], Tom Warnke[1], Fiete Haack[1,2] and Adelinde M Uhrmacher[1]

Abstract

With the increasing size and complexity of models, developing models by composing existing ones becomes more important. We exploit the idea of reusing simulation experiments of individual models for composition to automatically generate experiments for the composed model. First, we illustrate the process of modeling based on composition and discuss the role simulation experiments can play in this process. Our focus is on semantic validation of the composed model. We explicitly specify simulation experiments in simulation experiment specification via a Scala layer, including the desired model behavioral properties and their required experiment set-ups. Models are annotated with experiment specifications, and upon composition, these specifications are adapted and automatically executed for the composed model. The approach is applied in a case study of developing a Wnt/β-catenin signaling pathway model by successively composing three individual models, where we exploit metric interval temporal logic to describe model behavioral properties and check averages of stochastic simulation results against these properties.

Keywords

Simulation experiments, model composition, experiment specification, semantic validation

1 Introduction

Model development is a costly and time-consuming process. Thus, with more models available, reusing existing models as a basis for developing new ones becomes more interesting. Apart from holding the promise of reducing the time and efforts required for model development, model composition and integrating models into larger ones help to develop more complex models that capture a wide range of phenomena. It is widely acknowledged that developing models by composing existing ones poses many challenges.[1] In particular, checking the semantic composability is a difficult task, which ensures the composed model is semantically valid.[2]

‘‘*Model validation* is substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with M&S objectives.’’[3] Thus, model validity relates to the domain of applicability, which is determined by the underlying assumptions of modeling, and to the objectives of the simulation study. When two models are reused for composition, the composed model might cater for new assumptions and objectives. However, some key observations in the behavior of the reused models are expected to be preserved or violated in the composed model, as pointed out by Stelling et al.: ‘‘Are the key behaviors of the original models still intact? Does it matter if they are not?’’[4]

As simulation experiments with the model facilitate producing this kind of observation of the model behavior, experiments of the individual models can be reused to automatically generate experiments for a composed model, to check whether the composed model exhibits the key behavior of the original models. Therefore, important information can be provided about the validity of the composition as well as the composed model. The general idea of the proposed approach, as presented in our prior work,[5] is to gain insights into semantic composability of models by exploiting their validation experiments. Our approach is not aimed at the entire process of developing models by

[1] University of Rostock, Institute of Computer Science, Germany
[2] Research Unit “Functional Genome Analysis”, Leibniz Institute for Farm Animal Biology (FBN), Germany

Corresponding author:
Adelinde M. Uhrmacher, University of Rostock, Institute of Computer Science, Albert-Einstein-Straße 22, 18059 Rostock, Germany
Email: adelinde.uhrmacher@uni-rostock.de

composition, which includes several steps such as selecting model components and composing them together; the goal is to support checking the semantic validity of composed models.

In this paper, we structure the process of building models by successive composition with a life cycle, to illustrate the step in this process our work aims to contribute to, i.e., semantic validation, and how our approach is related to and can be combined with other work on model composition. Based on this life cycle, we discuss the roles that simulation experiments can play in developing models by composition, such as for model documentation and selection, and address that in our work, simulation experiments are exploited to support semantic validation. Building on previous work of reusing simulation experiment specification to support model composition, we refine our approach by using a domain-specific language for simulation experiment description, execution, as well as generation and adaptation. For the first time, we demonstrate the approach with a case study of building a comprehensive realistic model by successive composition, i.e., a model that analyzes the spatio-temporal regulation of the Wnt/β-catenin pathway during the early cellular differentiation processes, and we extend our approach with new features that are required by this case study.

2 A life cycle of developing composed models

In traditional model development, different life cycles are developed to organize the modeling process, e.g., the work by Balci and Sargent.[6,7] This can serve as a basis to structure the process of developing models via composition and identify crucial steps. Likewise, several perspectives exist to structure the process of model development based on composition, e.g., see the work by Kaputis and Ng, and Szabo and Teo,[8,9] depending on how model composition is supported by the corresponding approaches.

To structure the model development process via composition (as depicted in Figure 1), we adopt the work presented by Balci and Sargent and start the life cycle with *problem formulation, requirements engineering* and *conceptual modeling*, which are merged into one phase denoted as Problem formulation, Requirements engineering and Conceptual modeling (*PRC*), as these are not the focus of this paper.[6,7] At the end of the PRC phase, the conceptual model is developed. Similar to other work for structuring the development of composed models, this process comprises of important phases such as *model selection, model composition, syntactic verification*, and *semantic validation*, which are essential for creating a meaningful composed model. Once a valid model is developed from composition, it can serve as a starting point for building larger models, i.e., models can be successively reused for composition. Therefore, the composed model needs to be documented as required in order to be reused. As such, this modeling life cycle includes an additional phase *model documentation and storage*, where the composed model is documented and stored into model repositories for further reuse. Although the life cycle is divided into six phases, the whole process and each phase of the process are iterative.

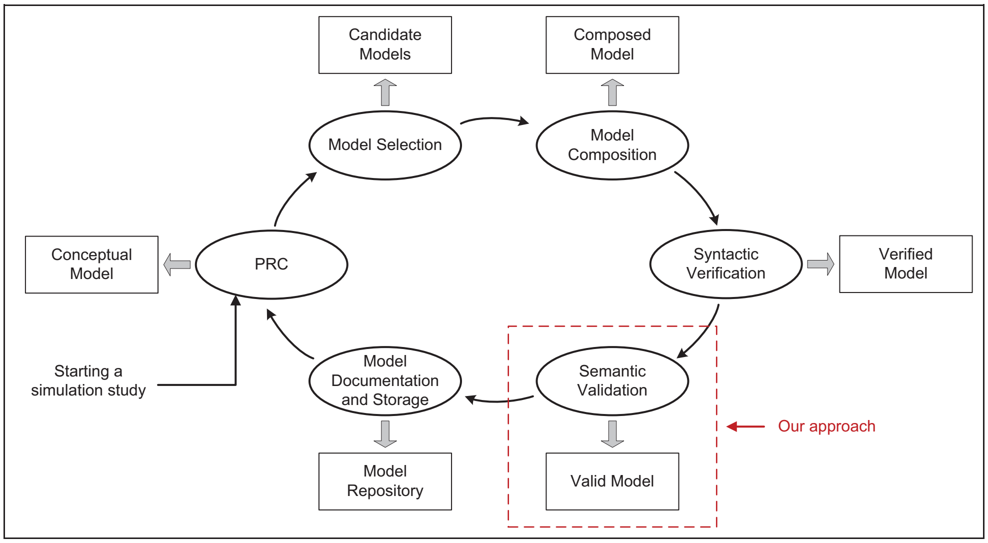


Figure 1. The life cycle of modeling based on composition. PRC stands for problem formulation, requirements engineering and conceptual modeling, which are merged into one phase in this life cycle with a conceptual model developed. Our approach aims to support semantic validation.

2.1 Problem formulation, requirements engineering and conceptual modeling

All model development life cycles start with analyzing the problem entity, defining the objectives and requirements of the modeling and simulation study, and developing the conceptual model, i.e., the PRC phase. In general, there is no wide agreement on conceptual modeling and the definition of a conceptual model.[10,11] In the work by Robinson,[12] the conceptual model is defined as *a non-software-specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions, and simplifications of the model.* Although different views and terms are used, it is generally agreed that the conceptual model is one of the first products in a modeling and simulation study and is independent from the simulation language or simulator.[11] In order to support model composition, specific efforts on conceptual modeling are needed.[13]

2.2 Model selection

Based on the defined conceptual model, modelers can search in the model repository and select suitable models for composition. Typically, model selection is based on matching between a user query and a model (component) repository. Therefore, the representation of user queries and models (components) becomes essential. Ontology and XML related techniques are often exploited to facilitate selecting models, e.g., the work by Teo and Szabo and Moradi et al.[14,15]

Model composition. Once selected, the candidate models are composed together. Different types of model composition are distinguished, e.g., black-box or white-box (see discussion in the work by Neal et al.).[16] In traditional component-based modeling approaches, typically model composition is realized by connecting the inputs and outputs of model components, which are usually defined as portable ‘‘building blocks’’ and can be combined for different contexts and needs.[2,17]

Meanwhile, in non-technical areas, such as computational biology, such building blocks with clear and static interfaces are more difficult to identify; accordingly, a different view of model composition is needed. On the one hand, instead of repositories of model components, extensive repositories of complete models can be found, e.g., the BioModels Database,[18] which facilitates model exchange and reuse. On the other hand, besides connecting inputs and outputs, an integration that merges variables, structures, and reactions of different models to be composed is required, e.g., *fusion*, which combines models into a single unified model without redundancies, and *composition*, which describes the ‘‘glue’’ that holds the models together, as presented by Randhawa et al.[19] In this case, the models used for composition cannot be treated as building blocks with inputs and outputs; instead, the composed models are created by merging rather than connecting components.

2.3 Syntactic verification

Regardless of whether the composed model is built from model components, or from complete models, through connection of model inputs and outputs or through merging model internal structures, the composition requires efforts to ensure a consistent, meaningful composed model. First of all, the consistency of the composed model on the syntactic level is required.

In the building block component-connection composition, syntactic composability is often ensured by model representation provided by the composition framework (e.g., in the work by Rohl and Uhrmacher),[20] predefined composition grammars (e.g., in the work by Szabo and Teo),[21] or matching rules (e.g., in the work by Moradi et al.).[15] When complete models are composed by fusion, the consistency of model descriptions is required, e.g., matching variables and compartments of different models that are composed to form a new one. In this case, possible conflicts are typically checked by the user, and therefore some tools provide additional assistance. For example, JigCell supports the different approaches presented in the work by Randhawa et al. to combine models.[19,22] It allows users to define models as spreadsheets, based on which models can be fused and composed via ‘‘mapping tables’’, or models can be defined as graphical box diagrams and then be aggregated by linking input and output ports.

2.4 Semantic validation

Syntactic verification refers to the coherence regarding the definition of the resulting composed model, which can be checked statically; the behavior of the composed model is of significant importance for the overall coherency as well. The compositions of models have to be valid; otherwise the models are not composable.[2] However, the validation of the composed model remains a grand challenge in modeling and simulation.[23] The situation is aggravated when the composed model is built from complete models. In contrast to model components that are developed to be used in an ‘‘unforeseen’’ context,[24] complete models are created to answer certain questions under specific contexts, and therefore are not able to be easily composed with other models in an arbitrary context.[17,25]

Although many efforts have been aimed at facilitating model composition, research on semantic validation of the composed model is limited and still under development. In the work by Petty et al.,[26,27] semantic composability is studied in a formal manner, where models are defined as mathematical computable functions, simulation is the sequential execution of a model represented by an labeled transition system (LTS), and model composition is viewed as mathematical function composition. The proposed approach evaluates semantic composability by comparing the simulation of the composed model and the simulation of a perfect model. A three-layered approach was proposed by Szabo and Teo to validate the composed model[28]: concurrent process validation as the first layer to ensure the logical correctness of component communication, metasimulation validation as the second layer to check the logical properties over time, and perfect model validation as the third layer to compare the composed model with a perfect model using a time-based formalism.[29] Other work on semantic composability exists, e.g., a formal method based on *Z* specification for Discrete Event System Specification (DEVS) models and semantic information comparison and state machine execution for composition of Base Object Models (BOMs).[15,30]

2.5 Model documentation and storage

In order for a model to be shared and reused, suitable documentation and accessible storage is important.[25] In recent years, model documentation and storage has received more and more attention, as it has a crucial impact on the effectiveness of the other phases in the life cycle, in particular model selection. In the work by Verbraeck and Valentin,[31] several guidelines for designing simulation building blocks were presented to improve reusability in component-based modeling from different aspects, e.g., self-containment and interoperability. In addition, some work exists on defining models in a way that supports model composition.[32–34]

To facilitate wide exchange and reuse of models, it is not sufficient to only provide the representation of the model itself, which is usually in a specific modeling formalism and may be difficult for third-party users to understand. Additional information about the model is necessary, such as the context in which the model is developed and the data used for the model development. Several guidelines have been proposed for model documentation, e.g., the MIRIAM (minimum information requested in the annotation of biochemical models) and the ODD (overview, design concepts, and details) protocol.[35–37] Ontologies are often used to annotate models with semantic information, and some tools have been developed to annotate and compose models based on ontologies such as SemGen.[38]

As more and more models are developed, more and more repositories exist, e.g., the BioModels database and the CD + + repository.[18,39] Thereby, work on the design and organization of model repositories can help model retrieval and selection, e.g., the work by Henkel et al. and Wang and Wainer.[40,41]

3 The role of simulation experiments in model composition

Exploiting simulation experiments can enable a more straightforward and reliable reuse of models, as shown in the work by Cooper et al.[42] However, the role of simulation experiments in developing models by successive composition has not been explored.

According to Cellier (pp.4),[43]‘‘an experiment is the process of extracting data from a system by exerting it through its inputs’’. Thereby, simulation experiments produce data from the model. Typically, model development is interleaved with executing simulation experiments.[44] For instance, in a model development process presented by Sargent,[7] the *operational validation* phase, which determines whether the model's output behavior has satisfactory accuracy for the intended purpose, relies on conducting simulation experiments on the *computerized model* to obtain data.

Effective reuse of models requires an understanding of model behavior, and simulation experiments can be used to gain information on the model behavior. Existing work on exploiting simulation experiments for model reuse mainly focuses on model documentation and storage. For example, in the work by Chreyh and Wainer[39], models are stored together with a number of experiments and their experimental frames (EF), which is ‘‘a specification of the conditions under which the system is observed or experimented with’’[45]. The documentation of models with EFs provides users with the context information which the models are developed and validated for. Each EF for a model contains some *experimental frame data* that are textual descriptions of objectives, assumptions and constraints, a set of *experiments* that contain descriptive data and can be applied on the model, and *experimental results information* for a given experiment on the model that includes a value indicating the success or failure of the experiment, description documentation, log file, and output file. Similarly, in the work by Henkel et al.,[40] models are stored in a graph database along with semantic annotations and associated experiments in simulation experiment description markup language (SED-ML) format.[46] While these approaches focus on reusing simulation experiments in model documentation and storage to further support model selection in the model development process, our idea is to reuse experiments to facilitate the semantic validation of the composed model.

Due to the abstraction nature, models are developed under a specific context, which can be reflected by experiment conditions, and therefore model validity is always related to the system and certain experiment conditions.[43] When models are reused for constructing larger models, the composed models could have new context; however, the new context may bear some similarities to that of the reused models. Therefore, the experiments conducted on the reused models for validation can be exploited to perform experiments on the composed model, providing information on the model behavior and further validating it. In addition, when models are developed by successively reusing existing models, the information of how this successive model development evolves is of interest. The history of the successive extension can help modelers to determine the contexts that the model has been applied for, and to gain insights into how the model can be reused for new contexts. Thereby, our focus is on reusing simulation experiments for semantic validation of the composed model and successive model development by composition, as shown in Figure 1.

4 Proposed approach

Figure 2 gives an overview about our approach. The approach depends on models being annotated with the explicit specifications of simulation experiments that have been executed with the model for validation, corresponding to step one in Figure 2. In the experiment specification, the expected behavioral properties of the model that are crucial to its validity are defined, as well as the information about how to generate this key behavior, e.g., in terms of model configuration and simulation configuration. Afterwards, individual models can be reused for composition, corresponding to step two in Figure 2. Based on how models are composed, the user can optionally specify information for experiment adaptation when necessary, such as renaming and reassignment of model parameters. Moreover, result expectations can be defined by the user to indicate, when checked on the composed model, which properties of the individual reused models are expected to hold and which are not. The associated experiment specifications of individual models are also reused and automatically adapted for the composed model, corresponding to step three in Figure 2. Based on adapted experiment specifications, experiments are executed with the composed model and results are checked, to test whether the specified behavioral properties, which have been successfully checked for the reuse model, still hold for the composed model, and whether this result meets the expectation, corresponding to step four in Figure 2. The goal, as presented in the work by Peng et al.,[5] is to automatically provide information about behavioral properties to the user during model composition, which supports validating the composed model.

To reuse simulation experiments of individual models for experimenting with the composed model, we need a method to unambiguously specify simulation experiments, a method to describe behavioral properties, a mechanism to automatically generate simulation experiments based on experiment specifications, and a method to assess the experiment results.

In our prior work,[5] we exploited SESSL (simulation experiment specification via a Scala layer) (http://sessl.org)[47] for specifying simulation experiments. To cope with stochastic models, LTL (linear temporal logic)[48] and CSL (continuous stochastic logic)[49] were used to express model behavioral properties for statistical model checking.

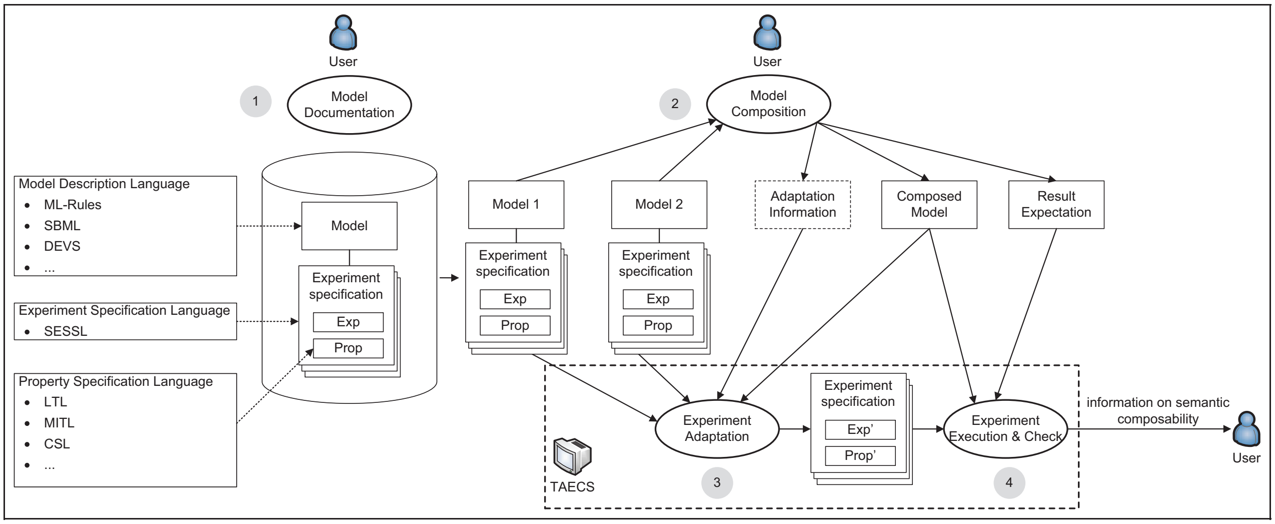


Figure 2. The overview of the approach. Models are annotated with explicit specifications of the simulation experiment executed for validation, and each experiment specification contains the expected behavioral property and the experiment conditions used to produce the property. When two models are composed, their experiment specifications are reused and adapted to automatically generate experiments for the composed model. The behavioral properties that hold for the individual reused model are checked against the experiment results to inspect whether the composed model exhibits the behavior or not and whether this is in agreement with user expectations, therefore providing insights on the validity of the composition.

SESSL was extended to support such property specification. With experiment specifications of individual models in SESSL, a mechanism was presented to automatically conduct simulation experiments on the composed model and perform statistical model checking to analyze the experiment results against specified properties.

In this paper, we aim to apply the approach in the case study of developing a realistic model of the Wnt/β-catenin pathway through model composition. In the following, we will describe the ingredients for this case study, and the required new developments compared to the previous work.[5]

4.1 Experiment specification

We rely on an explicit specification of experiments for model validation. A wide variety of approaches exists for model validation, and a detailed description of different approaches is presented in the work by Sargent.[7] Although each of these approaches checks whether a model's behavior satisfies a certain requirement, the accessibility of these requirements varies, from being elusive (e.g., face validity) and difficult to retrieve as they depend on interactive explorations (e.g., visual analytics),[50] through being implicitly part of the method (e.g., sensitivity analysis), up to being explicitly and declaratively specified as properties in a formal language, as in model checking (e.g., in the work by Fages and Rizk).[51] In modeling and simulation, experimental model validation evaluates model validity through experimenting with the model, in which the model context is reflected by experimental configuration.[52] If the simulation experiments that are executed for validating models shall not only be annotated with models, but also be reused, we need a declarative and accessible description of these experiments. Thus, our focus will be on the validation experiments that check the simulation output for properties, which are explicitly described to express the desired behavior of the model.

While the model configuration has an obvious effect on the experiment, configurations of the simulation run and result analysis also influence the experiment results. For example, results of simulation experiments can differ substantially depending on whether they use stochastic or deterministic algorithms to execute the model. It has also been acknowledged by some experiment description standards that simulation algorithms and their configuration can have a significant impact on simulation results, e.g., SED-ML,[46] which employs the KiSAO (kinetic simulation algorithm ontology)[53] to specify simulation algorithms. Other simulation parameters such as stopping conditions or replication numbers (in case of stochastic models), have an impact on simulation results as well. Furthermore, how the raw data produced in simulation are collected and processed is crucial for result analysis. Consequently, information about the model (e.g., what kind of model configurations have been tested) as well as information about the execution of the simulation experiment (e.g., what kind of execution algorithms with which set-up) need to be available to access or reproduce the experiments executed with a model.

As in our previous work,[5] SESSL is chosen for experiment specification in this paper. SESSL is an internal domain-specific language for specifying simulation experiments in a declarative style.[47] Specifications in SESSL are unambiguous and executable. Thus, it facilitates reusing and reproducing simulation experiments. Moreover, as an internal domain specific language, SESSL can be easily extended to support new features.

4.2 Property specification and checking

To allow automatic analysis of experiment results, an explicit specification of expected behavioral properties is necessary. Temporal logics and model checking techniques have been widely used for analyzing systems.[54,55] The properties of interest are formalized in temporal logics and then automatically verified with model checking algorithms. In our previous work,[5] we relied on LTL to express the behavioral properties of Lotka–Volterra models for individual simulation trajectories. As ML-Rules was used for model description, which is based on a continuous-time Markov chain,[56] CSL was exploited to allow the expression of probability regarding multiple replications of simulation.

LTL and CSL have been successfully used in many applications. However, the temporal operators in these languages do not allow for quantitative statements about time. A precise description of model outputs, however, often requires statements about the time a certain observation occurs. Therefore, we employ the metric interval temporal logic ($\mathrm{mitl}_{[a,b]}$)[57] rather than LTL or CSL in this work. $\mathrm{mitl}_{[a,b]}$ enriches the traditional modal operators of temporal logics with intervals, and defines that the operator is only evaluated in this interval.

Given a property specification, simulation results can be analyzed in different ways, depending on whether the model that generates those trajectories is deterministic or stochastic. For deterministic modeling formalisms such as ordinary differential equations (ODEs), finite time model-checking algorithms can be used to check the simulation trajectory against a defined property, e.g., in the work by Fages and Rizk.[51] For stochastic models, multiple replications are required and statistical model checking methods are usually employed to estimate the probability that an individual trajectory produced from a random simulation run satisfies the property,[58] as the approach used in our previous work.[5]

The validation of models in our case study relies on checking simulation results against properties that are derived from observations in wet-lab experiments with human neural progenitor cells.[59,60] Whereas we develop a model of a single cell, the available data refer mostly to aggregate observations made on heterogeneous populations of thousands up to a million of cells, and these aggregate observations are the basis for further discussions and explorations. Accordingly, we also base our property checking on averages, which here refer to averages of stochastic simulation runs executed with the same model rather than model populations discussed by Burrage et al.[61] A stochastic simulation is advisable due to the low amount of Axin, which is one of the key players involved in Wnt/β-catenin signaling.[62] Our methodology differs from statistical model checking in its handling of variance among simulations: aggregation instead of hypothesis testing on individual runs. The latter has proven to be less suitable in this case study, as most of the properties that are of interest could not be verified by statistical model checking due to the high variance among simulation runs.

To determine the number of simulation replications needed, we calculate confidence intervals at each relevant observation point and compare the width to a defined threshold. Once we have executed a sufficient number of runs, we calculate a point-wise average of all simulations to obtain the trajectory of our average cell.

Averaging across simulation runs considers the variance between simulation runs. However, simulation runs also fluctuate over time. These fluctuations of single runs are still evident in the average trajectory. However, specifications in $\text{mitl}_{[a,b]}$ and the algorithms to check them are not able to take account of such fluctuations. For example, an algorithm checking a trajectory for a strict increase might reject it because of some short, random episodes of decreasing. We avoid such errors by preprocessing trajectories with the local regression (LOESS) smoothing method.[63] The parameters band width and iteration number of the LOESS algorithm can be configured in the SESSL specification, and thus adapted to the model at hand. This method can be applied to individual runs as well as to averages.

In the previous work, SESSL was extended to support property specification in LTL and CSL, and statistical model checking. In this work, several new features are required, i.e., determining required simulation replications based on thresholds for point-wise confidence interval width, computation of the average trajectory, data preprocessing with LOESS trajectory smoothing, and specification and checking model behavioral property based on $\text{mitl}_{[a,b]}$. Thus, SESSL was extended to support these features. The implemented $\text{mitl}_{[a,b]}$ checking algorithm is described by Maler and Nickovic.[57] For the core functions of the other new features, we relied on the software library Apache Commons (commons.apache.org). Integration of the new features into the experiment specification interface of SESSL was straightforward. With SESSL being an internal domain-specific language, new experiment aspects like the ones introduced in this work can be implemented as Scala traits and functions (see the work by Ewald and Uhrmacher for details).[47]

4.3 Experiment generation and execution

While generating simulation experiments for the composed model by reusing experiment specifications of individual models, certain problems have to be considered. First, in the experiment specifications, the model location needs to be adapted from the original model to the composed model. In general, the composed model has a higher dimension with respect to model parameters, as it contains the model parameters of both reused models. Therefore, when generating simulation experiments for the composed model based on the experiment specification of one of the two reused models, a specific strategy to determine the configuration of the model parameters that only exists in the other model and those that exist in both models but are configured differently is required. Also, the modeler may make some changes in the model parameters during model composition, e.g., renaming, reassignment of existing model parameters and adding new parameters, which have to be taken into account as well. Thus, to generate experiments for the composed model, we firstly need to adapt the SESSL experiment specifications of the two reused models. The experiment adaptation is based on two algorithms presented in our previous work and we skip the technical details here.[5] Afterwards, the adapted SESSL experiment specifications are executed. Users can specify expectations on the result of checking properties on the composed model, i.e., which properties should hold and which should not. The experiment results of the composed model are automatically checked against the defined properties and further compared to the result expectations. Therefore, the user can ensure that certain behavioral properties are not violated and certain behavioral properties are not kept unexpectedly after the composition, and information on the semantic validity of the composed model is provided.

The first implementation of the concept was realized on top of the modeling and simulation framework JAMES II (http://jamesii.org),[64] as presented in our prior work,[5] where SESSL was only used as the user interface for specifying experiments, whereas experiment adaptation and generation was conducted within the experimental layer of JAMES II. Currently, the mechanism for adapting, executing, and checking experiment specification no longer relies on JAMES II; instead, our tool TAECS (a tool for adaptation, execution and checking of simulation experiments) has been developed based on SESSL. Simulation experiment specification, execution, adaptation, and generation are all realized based on SESSL, as shown in Figure 2. The adapted experiment specifications that are executed on the composed model are automatically documented and, if checked successively, can be annotated with the

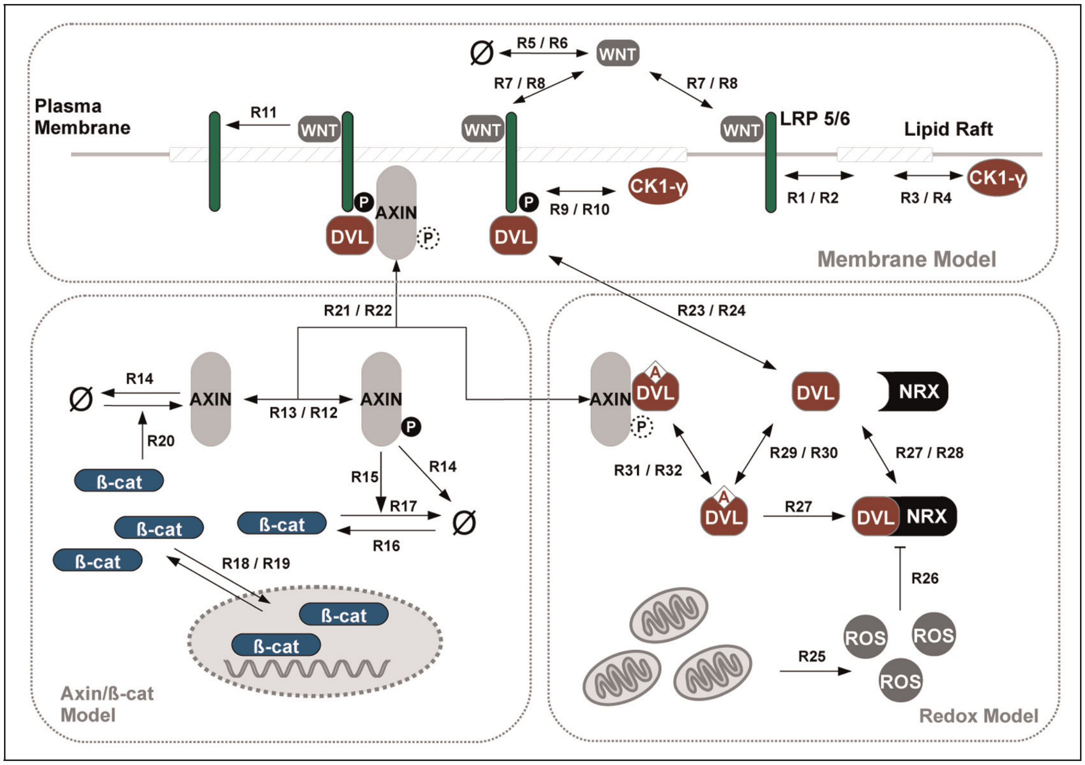


Figure 3. The Wnt pathway model that contains three individual models: the membrane model, the Axin/β-catenin model and the ROS model.[60]

composed model for further reuse. As a result, our refined approach supports successive model composition. SESSL is simulation system agnostic, and allows the integration of different simulation methods or other tools via bindings. Thus, different methods for model representation and property specification can be supported by our approach (see Figure 2).

5 Case study: Modeling the Wnt pathway

We apply our approach to the development of a model of a Wnt/β-catenin signaling pathway of human neural progenitor cells (hNPCs).[60] Wnt/β-catenin signaling is involved in central cellular processes, such as differentiation, proliferation, and migration of cells. Whereas its deregulated form leads to developmental disorders and various diseases, including several forms of cancer and Alzheimer's disease,[65] it also drives cell differentiation. A better understanding of this pathway and its regulation of human neural progenitor cells (hNPCs) differentiating into neurons, astrocytes, and oligodendrocytes cells will allow more effective replacement therapies of neurodegenerative diseases, such as Parkinson's or Huntington's disease.

In the following we will replay the development of the model and make the overall process of model development by composition and the role of reusing simulation experiments explicit.[60] Therefore, a computational stochastic model will be developed using the proposed approach through successively composing three individual models: a membrane model (M1), an Axin/β-catenin model (M2) and a reactive oxygen species (ROS) model (M3), as shown in Figure 3. First, the membrane model is composed with the Axin/β-catenin model, from which the first composed model (M12) is obtained. Then the resulting model is again composed with the ROS model (M3), which results in the second composed model (M123) comprising all three models. All model files can be found in the supplementary material, and also executables are provided to reproduce all experiments presented in this paper.

5.1 Problem formulation, requirements engineering and conceptual modeling

Several studies suggested an involvement of lipid rafts in Wnt/β-catenin signaling.[66] However, the spatio-temporal dynamics and the exact impact of lipid rafts on Wnt/β-catenin signaling remain unclear, which serves as motivation to develop a Wnt/β-catenin model that contains also membrane related dynamics. The aim is to compose a membrane model that comprises central key players of membrane related dynamics such as the receptor complex and its interaction with lipid rafts, and a core model of the Wnt/β-catenin that is able to predict the β-catenin accumulation within the nucleus depending on the activity of the destruction complex. Once this model is validated, our simulation experiments with this model will refer to disturbing lipid rafts (as the input), and observing the influence on β-catenin as the output.

We present our conceptual model in the following. According to the definition of conceptual model from Robinson,[12] a conceptual model typically refers to *objectives, inputs, outputs, content, assumptions, and simplifications*. The objective of our simulation study is to explore the role lipid rafts play in the Wnt/β-catenin signal transduction, with the main input variable being the presence/removal of lipid rafts and the output being the concentration of nuclear β-catenin.

The analysis of the Wnt/β-catenin signaling pathway has been the subject of a series of simulation studies and quantitative models.[67] The key component of Wnt/β-catenin signaling is a protein termed β-catenin. In the inactive state of Wnt/β-catenin signaling, β-catenin is constantly produced, but immediately targeted for degradation by a large protein complex termed a destruction complex. Upon Wnt stimulation, a reaction cascade is triggered, which leads to the inhibition of major components of the destruction complex, such as Axin. As a result, β-catenin accumulates inside the cytosol and subsequently shuttles into the nucleus. Once shuttled into the nucleus, β-catenin associates with the Lef/Tcf transcription factors and triggers a pathway-specific gene response relevant for the regulation of various physiological and developmental processes.[65] The relocation of β-catenin into the nucleus is therefore the main indicator for the pathway activation. The Wnt/β-catenin signaling starts with Wnt binding to the receptor LRP6 to form the Wnt receptor complex. The phosphorylation of the complex is restrained to be inside lipid rafts, and the phosphorylated complex recruits the key component of the destruction complex Axin, and therefore causes an accumulation of β-catenin inside the nucleus.

A simplification is that for the receptor complex only LRP6 and CK1γ are considered. Actually, the Wnt receptor complex also comprises a sub-variant of the Frizzled receptor as well as additional membrane-bound adapter proteins. This is reasonable for Wnt/β-catenin signaling, since all crucial events mainly depend on the dynamics of LRP6 and CK1γ. In addition, we also only consider Axin as the destruction complex, disregarding the remaining proteins such as GSK3β. Mathematical analyses have shown that such a reduced model is capable of reproducing the essential dynamics of Wnt-induced β-catenin signaling.[68]

In a previous study, it has already been shown that stochastic effects should not be ignored in Wnt/β-catenin signaling of human neural progenitor cells.[62] Thus, a modeling language that allows us to describe inter- and intra-compartmental dynamics and ships with a formal, stochastic semantics is required. Therefore, we decide to define our model within ML-Rules, which is a rule-based modeling language for cell biological systems.[69] ML-Rules supports hierarchical modeling, such as the cell including cytosole, nucleus, and membrane, which includes further structures termed lipid rafts, and compartmental dynamics. Also, it allows a flexible expression of reaction rate kinetics. As our data refers to the first 12 h (720 min) of differentiation, also our simulation experiments would be constrained to this time period.

5.2 Model selection: The membrane model (M1)

In the ideal case a suitable model would be stored in a model repository. However, more often than not, we will encounter the problem that an essential model is missing and needs to be developed, as is the case here. This individual model will describe the diffusion-driven shuttling of LRP6 and CK1γ between raft and non-raft membrane regions. Therefore, the model uses a reduced representation of the receptor complex. Lipid rafts are modeled as individual compartments within the membrane, similar to the nucleus being a single compartment within the cell. Accordingly, the main behavioral properties for this model component relate to the fraction of raft-associated LRP6 (LR[Lrp6]) and CK1γ (LR[CK1γ]) molecules according to Sakane et al.,[66] and we expect that after a warm up phase around one-quarter of the overall Lrp6 population and around three-quarters of CK1γ population to reside within lipid rafts. As an explicit and quantitative expression of properties over time is necessary, we use mitl$_{[a,b]}$ for property specification in this case study, instead of LTL as in the prior work.[5] Given the model $M1$ with its parameters $\tilde{p}$, and the simulation set-up $ses1$, we check the above property specified in mitl$_{[a,b]}$ as follows, where t_1 is the time point after the warm up phase:

$$\begin{aligned} M1_{\tilde{p}} \models_{ses1} G(t_1, 720, LRLrp6 > \ = nLrp6 * 0.25 \\ \wedge LRLrp6 < \ = nLrp6 * 03) \end{aligned} \tag{1}$$

$$\begin{aligned} M1_{\tilde{p}} \models_{ses1} G(t_1, 720, LRCK1\gamma > \ = nCK1\gamma * 0.7 \\ \wedge LRCK1\gamma < \ = nCK1\gamma * 075) \end{aligned} \tag{2}$$

To perform simulation experiments on this model, we configure the model parameters as: `nLR` = 5, `nLrp6` = 4000, `nCK1y` = 5000, where `nLR`, `nLrp6`, `nCK1y` denote the initial number of lipid rafts, LRP6, and CK1γ. In this as well as in the following experiment specifications, the values for parameters and initial concentrations were partly fitted, partly taken from literature, and partly directly measured in our wet-lab experiments (see Table 1 in the work by Haack et al. for more detail).[60] The simulation method presented by Warnke et al.[69] is used for

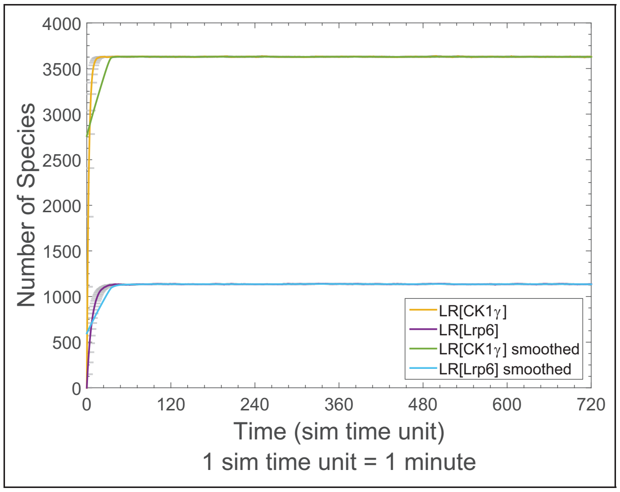


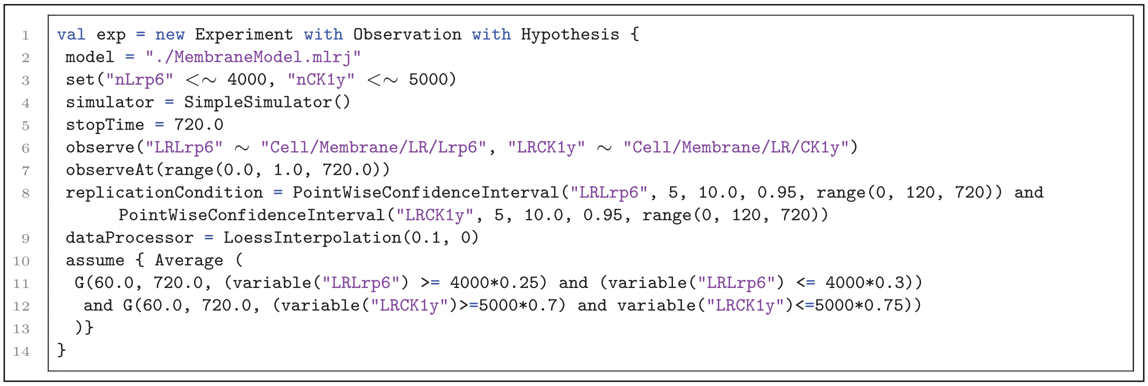
Figure 4. Simulation results of the membrane model (M1), with the average trajectories with 95% confidence interval (gray error bars) and the smoothed trajectories.

executing experiments, named as `SimpleSimulator` in SESSL. We set the simulation stop time as 720 simulation time units, with one time unit corresponding to 1 min in the wet-lab experiments. During the simulation, the number of species LR[Lrp6] and LR[CK1γ] are observed and recorded. As discussed in the previous section that in contrast to our prior work,[5] statistical model checking is not applicable in this case study, we check the average of multiple simulation runs and compute replications until all point-wise confidence intervals are smaller than a defined width threshold. We choose a 95% confidence level and the width threshold as 10, i.e., the 95% confidence interval is constructed with the average of observed species number plus/minus at most five. To alleviate the impact of stochasticity, the produced simulation raw data are processed using the LOESS smoother with a band width of 0.1 and no additional iterations, constituting a minimal smoothing. The simulation trajectories are shown in Figure 4.

We specify the experiments in SESSL, including the desired behavioral properties, as presented in Figure 5. The model to simulate is defined in file `MembraneModel.mlrj` (line 2), with the model parameter configuration in line 3. The simulator named `SimpleSimulator` is specified for experiment execution (line 4) and each simulation run stops when the simulation time is 720 (line 5). The amount of LRP6 and CK1γ located in lipid rafts are observed (line 6), which are accessible as `LRLrp6` and `LRCK1y` in SESSL, at every time unit from time 0 to time 720 (line 7) during the simulation. Line 8 denotes the specification of replication conditions that for both species `LR[ Lrp6]` and `LR[ CK1`γ`]` , the average value observed at every 120 time units from time 0 to 720 shall have 95% confidence interval with width being less than 10. Line 9 configures the data preprocessing method and the LOESS smoothing method is used. Next we specify the property regarding the fraction of lipid raft-associated LRP6 and CK1γ molecules based on $\text{mitl}_{[a,b]}$, according to Equations (1) and 2, indicated by the SESSL keyword *assume* (line 10). Based on the simulation trajectories shown in Figure 4, we set $t_1 = 60$. The corresponding property states that for the average of all simulated trajectories, from time 60 to time 720, the amount of `LRLrp6` is always between 25% and 30% of the initial amount of Lrp6 (i.e., 4000), and the amount of `LRCK1y` is always between 70% and 75% of the initial amount of CK1γ (i.e., 5000) (lines 10–13).

5.3 Model selection: The Axin/ β-catenin model (M2)

A series of models have been developed that describe the core β-catenin dynamics.[67] However, only one is of stochastic nature and has been validated for hNPCs.[62] Therefore, we select this model for composition. This intracellular model primarily describes the dynamics of β-catenin, i.e., its synthesis, its interaction with the



```
 1  val exp = new Experiment with Observation with Hypothesis {
 2   model = "./MembraneModel.mlrj"
 3   set("nLrp6" <~ 4000, "nCK1y" <~ 5000)
 4   simulator = SimpleSimulator()
 5   stopTime = 720.0
 6   observe("LRLrp6" ~ "Cell/Membrane/LR/Lrp6", "LRCK1y" ~ "Cell/Membrane/LR/CK1y")
 7   observeAt(range(0.0, 1.0, 720.0))
 8   replicationCondition = PointWiseConfidenceInterval("LRLrp6", 5, 10.0, 0.95, range(0, 120, 720)) and
          PointWiseConfidenceInterval("LRCK1y", 5, 10.0, 0.95, range(0, 120, 720))
 9   dataProcessor = LoessInterpolation(0.1, 0)
10   assume { Average (
11    G(60.0, 720.0, (variable("LRLrp6") >= 4000*0.25) and (variable("LRLrp6") <= 4000*0.3))
12     and G(60.0, 720.0, (variable("LRCK1y")>=5000*0.7) and variable("LRCK1y")<=5000*0.75))
13    )}
14  }
```

Figure 5. Experiment specification of the.embrane model (M1) in SESSL.

destruction complex and the resulting degradation process as well as its shuttling between nucleus and cytosol. These dynamics are triggered by an initial amount of Wnt. The model resembles a reduced version of the Lee model containing only two proteins (β-catenin and Axin).[70] Axin is further characterized by a phosphorylation site, which is important for the degradation of β-catenin. While phosphorylated Axin (AxinP) promotes β-catenin degradation, its unphosphorylated state (Axin) has no impact on β-catenin. The model has been validated based on in-vitro data of hNPC.[59] To reproduce the biphasic behavior observed in the wet-lab, a delayed autocrine mechanism proved to be essential and had been introduced into the model.[62]

This biphasic behavior of the nuclear β-catenin concentration is the main model property of interest. Within the first 2 h of differentiation a peak can be observed, after that β-catenin increases slightly again and stabilizes around a 1.4–1.7 fold increase. The properties can be expressed based on mitl$_{[a,b]}$ as follows:

$$M2_{\tilde{p}} \models_{ses2} (d(\beta cat)/dt >\ = 0) \qquad U(40, 100, F(0, 10, d(\beta cat)/dt <\ = 0)) \tag{3}$$

$$M2_{\tilde{p}} \models_{ses2} G(400, 720, \beta cat >\ = 1.4 * n\beta cat \wedge \beta cat >\ = 1.7 * n\beta cat) \tag{4}$$

It should be noted that no experiments have been stored together with the model – although a more recently introduced standard for describing simulation experiments encourages the storage of cell biological models along with experiments.[46] Thus, we reconstructed the experiments based on this publication by Mazemondet et al.[62] The simulation results are shown in Figure 6 and we describe the experiment in SESSL, as depicted in Figure 7. Experiment specification shown in Figure 7(a) corresponds to the property expressed in Equation (3), and that shown in Figure 7(b) corresponds to the property in Equation (4).

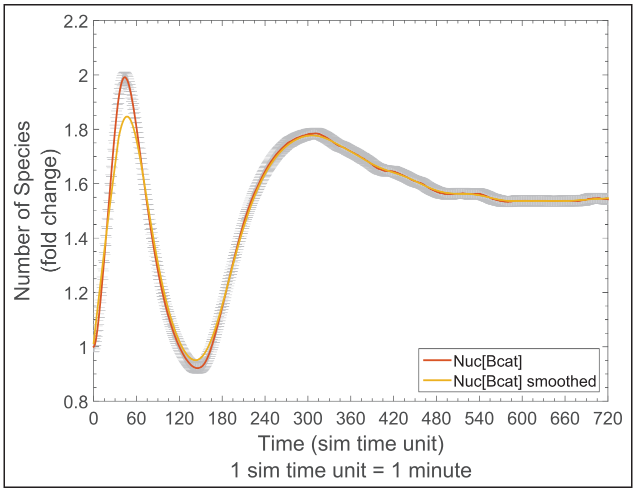


Figure 6. Simulation results of the Axin/β-catenin model (M2), with the average trajectories with 95% confidence interval (gray error bars) and the smoothed trajectories.

```
val exp = new Experiment with Observation with Hypothesis {
 model = "./AxinBcatModel.mlrj"
 set("nBcatNuc" <~ 5282)
 simulator = SimpleSimulator()
 stopTime = 720.0
 observe("NucBcat" ~ "Cell/Nuc/Bcat")
 observeAt(range(0.0, 1.0, 720.0))
 replicationCondition=PointWiseConfidenceInterval("NucBcat",
      5, 200.0, 0.95, range(0, 120, 720))
 dataProcessor = LoessInterpolation(0.1, 0)
 assume { Average (
  d("NucBcat") >= 0 U(40.0, 100.0, F(0.0, 10.0,
        d("NucBcat") <= 0))
  )}
}
```

(a)

```
assume { Average (
 G(400.0, 720.0, (variable("NucBcat") >= 1.4 * 5282) and
       (variable("NucBcat") <= 1.7 * 5282))
 )}
```

(b)

Figure 7. Experiment specifications of the Axin/β-catenin model (M2) in SESSL. (a) Experiment specification regarding the fold change of nuclear β-catenin concentration peaking within first 100 min. (b) Property specification regarding the fold change of nuclear β-catenin concentration staying in a certain range in the later hours. The experiment set-up is the same as that in (a).

5.4 Model composition: Composing M1 and M2 (M12)

ML-Rules models are reaction-based with hierarchically nested structures; however, ML-Rules models do not support a modular design of models. Therefore, and also given the nature of the systems to be modeled (see discussion on composition in the work by Randhawa et al.),[19] the black-box composition is not applicable. Instead, composing ML-Rules models asks for fusion, as presented in the work by Peng et al.,[71] where the variables, structures, and rules of the reused models are merged. Currently, the generation of a composed model from two ML-Rules models relies on the user, as it is rather difficult (if not impossible) to automatically compose two models based on fusion. In the composed model, the species Wnt no longer directly interacts with Axin but is mediated due to a membrane related processes, i.e., Wnt-bound and phosphorylated LRP6 (Lrp6PP) can recruit and bind Axin at the membrane. LRP6 thus interferes with the function of Axin in the degradation complex, hence disrupting the degradation of β-catenin. Thereby, the extracellular Wnt signal is eventually transduced into the cytosol, leading to the accumulation of β-catenin in cytosol and the nucleus. Also, in the composed model Wnt binds to Lrp6 and its phosphorylation is confined to lipid rafts.[72] Only phosphorylated Lrp6 recruits Axin from the cytosol, and due to its size, the resulting signalosome is unlikely to leave the lipid raft, thus having an impact on the diffusion patterns of Lrp6.

5.5 Syntactic verification: Composed model (M12)

A short inspection of the two models reveals that the two models M1 and M2 are independent, i.e., they neither share rules nor species nor compartments which could be checked for consistency. Instead, the rule relating Wnt and Axin has to be removed, and additional rules need to be added which describe the interaction between Wnt and Lrp6, Lrp6 phosphorylation, and the interaction between Lrp6 and Axin, respectively. The correct fusion of the two models relies on the user. However, some assistance can be provided to support the composition syntactically. For instance, an ML-Rules model editor was developed with syntax highlighting, which ensures the composed model is syntactically correct, and types and variables adhere to the constraints provided.

Whereas most parameters are inherited from the two composed models, parameters that relate to Wnt (as Wnt is no longer directly interacting with Axin) or to newly added rules, have to be (re-)calibrated based on recent wet-lab data or based on literature.[60]

5.6 Semantic validation: Composed model (M12)

To check whether the composition is semantically valid, we first inspect the two properties that have been defined for M1. As already stated, in the presence of Wnt (due to the initial amount of Wnt and the autocrine effect realized by a delayed production of Wnt within the model) and the forming of the signalosome, we no longer expect a stable Lrp6 fraction within lipid rafts. Instead, due to the size and thus reduced diffusion speed of the signalosome, an accumulation of Lrp6 within lipid rafts is expected.[73] In contrast, the previously observed steady fraction of nCK1γ should not be affected by the presence of Wnt. Moreover, we also reuse the experiment specifications of M2 (as shown in Figure 7) for experimentation with the composed model M12 to check the properties referring to β-catenin with the expectation that they also hold for M12.

During the composition there was no renaming of model parameters and the configuration of the newly introduced model parameters are provided as part of the model within the model file by the user. Consequently, in the reused experiment specifications only the location of the model for experimentation needs to be updated, from the reused model M1 or M2 to the composed model M12.

Based on the adapted experiment specifications, experiments are executed on the composed model and the behavioral properties are checked accordingly. The simulation trajectories are shown in Figure 8(a) and (b). The generated results show that all properties, except for the one regarding the steady LR[Lrp6] concentration, of the original models hold for the composed model, which is as expected and indicates the composed model M12 indeed works as intended with respect to the key behavioral properties of its two reused models.

Therefore, the composed model M12 is annotated with the three behavioral properties from the two reused models M1 and M2 under corresponding experiment set-ups, i.e., the properties regarding species CK1γ and nuclear β-catenin as shown in Equations (2), (3), and (4), and can be expressed based on mitl$_{[a,b]}$ as Equations (6), (7), and (8). As for the property regarding the species Lrp6, illustrated by the simulation results in Figure 8(a) and Figure 4, the composed model M12 behaves different from the original model M2, i.e., the amount of Lrp6 within lipid rafts accumulates instead of reaching a stable fraction as shown in Equation (1). The updated property regarding the species Lrp6 is described in Equation (5), while the experiment set-up stays the same as that in the reused model M1:

$$M12_{\tilde{p}} \models_{ses1} G(d(LRLrp6)/dt > \; = 0) \tag{5}$$

$$\begin{aligned} M12_{\tilde{p}} \models_{ses1} G(60, 720,\, LRCK1\gamma > \; = nCK1\gamma * 0.7 \\ \wedge LRCK1\gamma < \; = nCK1\gamma * 075) \end{aligned} \tag{6}$$

$$\begin{aligned} M12_{\tilde{p}} \models_{ses2} (d(\beta cat)/dt > \; = 0) gtU(40, 100, \\ F(0, 10, d(\beta cat)/dt < \; = 0)) \end{aligned} \tag{7}$$

$$\begin{aligned} M12_{\tilde{p}} \models_{ses2} G(400, 720, \beta cat > \; = 1.4 * n\beta \\ \wedge \beta cat > \; = 1.7 * n\beta cat) \end{aligned} \tag{8}$$

As our next step, we execute different experiments to further validate the model, e.g., to show whether depending on the number of initial Wnt the nuclear β-catenin levels of a different, independent study can be reproduced.[74] Afterwards, we turn to the question that motivated our simulation study, i.e., to analyze the impact of lipid rafts on β-catenin based on the calibrated and validated model. Within our wet-lab experiments we disrupted the lipid rafts of the hNPCs by applying methyl-beta-cyclodextrin. Whereas the Wnt-β-catenin signalling pathway appeared inactive in the later hours as expected (i.e., no β-catenin fold increase), we observed a peak of β-catenin within the first three to four hours. Accordingly, we remove the lipid rafts from the model M12, by setting nLR = 0 to reproduce the findings of our wet-lab experiments with lipid raft deficient hNPC. The two properties can be expressed based on mitl$_{[a,b]}$ as in Equations (10) and (9), while *ses*12 denotes the experiment set-up, which is similar to *ses*2 but nLR = 0. Simulation experiments as presented in

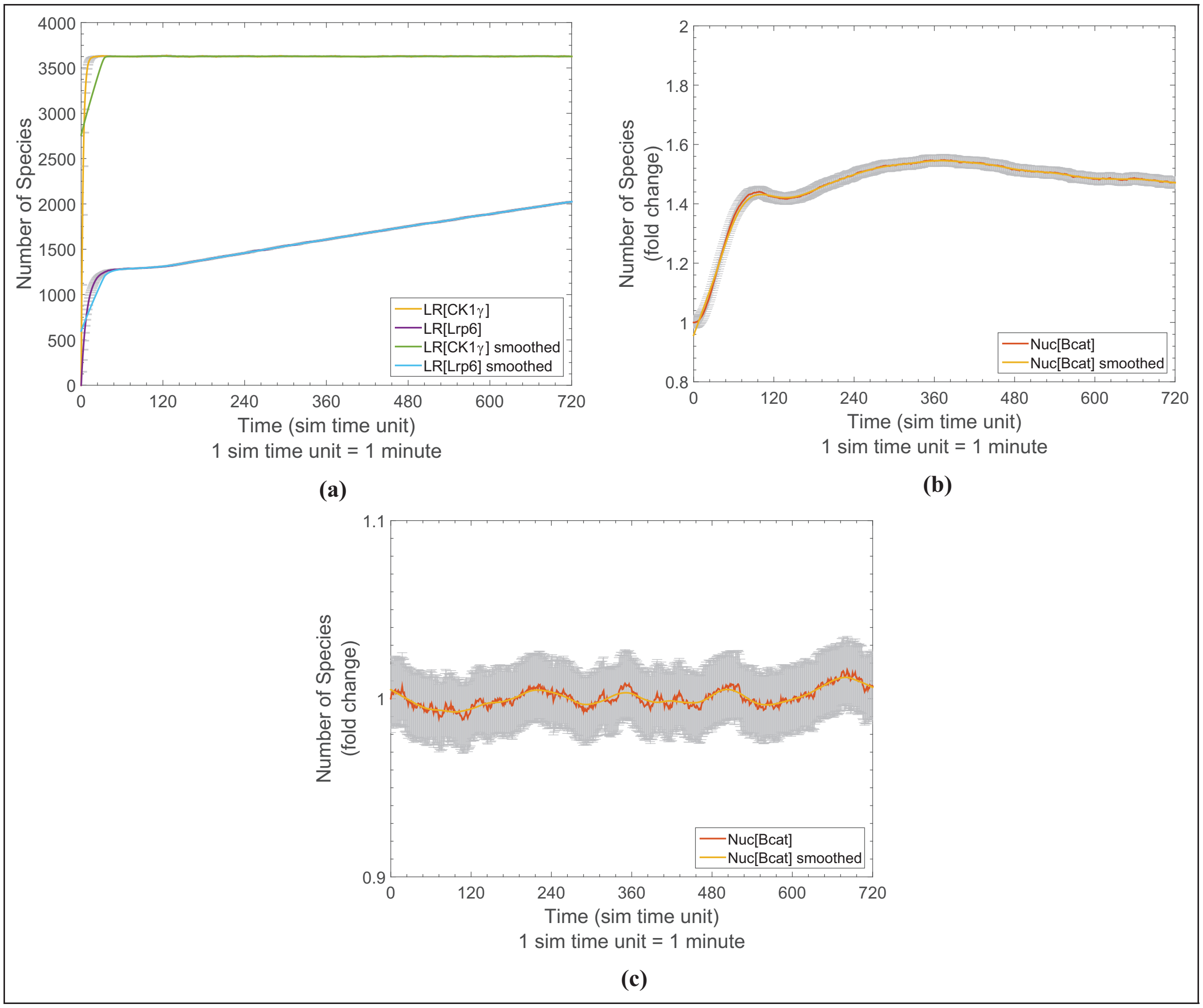


Figure 8. Simulation results of the first composed model (M12), with the average trajectories with 95% confidence interval (gray error bars) and the smoothed trajectories. (a) The simulation results regarding Lrp6 and CK1γ. (b) The simulation results regarding nuclear β-catenin with the default model configuration. (c) The simulation results regarding nuclear β-catenin when lipid rafts are removed from the model,i.e., nLR = 0.

Figure 8(c) showed that the second expectation can be met, however, not the first one, i.e., Equation (10) does NOT hold for the composed model M12, only Equation (9):

$$M12_{\tilde{p}} \models_{ses12(nLR\,=\,0)} G(400, 720, \beta cat> \; = 0.9 * n\beta cat \wedge \beta cat> \; = 1.1 * n\beta cat) \tag{9}$$

$$M12_{\tilde{p}} \models_{ses12(nLR\,=\,0)} (d(\beta cat)/dt> \; = 0 \; U(40, 200, F(0, 10, d(\beta cat)/dt< \; = 0)) \tag{10}$$

In M12, the Wnt influence on β-catenin entirely relies on lipid rafts. By completely removing those from the model, the raft-dependent LRP6 phosphorylation by CK1γ in response to a Wnt stimulus is prevented, and no Axin can be recruited, which results in no accumulation of β-catenin inside the nucleus. However, this is in contradiction to the observation in wet-lab experiments in the first few hours, where the amount of nuclear β-catenin increases. Therefore, an important mechanism is missing that signs responsible for the immediate increase of β-catenin and its peak in raft deficient cells within the first 3 h, and this mechanism appears to be Wnt independent.

5.7 Model documentation and storage

In the presence of lipid rafts our composed model appears to be fairly valid; in the absence of lipid rafts only its response referring to the last 8 h of our experiments is as expected, at which time autocrine mechanisms are already effective. Thus, the model underlines the role of lipid rafts, particularly in combination with autocrine mechanisms for the later hours of the early hNPCs' differentiation. We annotate our composed model M12 with the different

```
assume { Average (
 G(60.0, 720.0, variable("LRCK1y") >= 5000 * 0.7 and
       (variable("LRCK1y") <= 5000 * 0.75))
  and G(0.0, 720.0, d("LRLrp6") >= 0)
 )}
```

(a)

```
val exp = new Experiment with Observation with Hypothesis {
 model = "./FirstComposedModel.mlrj"
 set("nBcatNuc" <~ 5282, "nLR" <~ 0)
 simulator = SimpleSimulator()
 stopTime = 720.0
 observe("NucBcat" ~ "Cell/Nuc/Bcat")
 observeAt(range(0.0, 1.0, 720.0))
 replicationCondition=PointWiseConfidenceInterval("NucBcat",
       5, 200.0, 0.95, range(0, 120, 720))
 dataProcessor = LoessInterpolation(0.1, 0)
 assume { Average (
  G(400.0, 720.0, (variable("NucBcat") >= 0.9 * 5282) and
        (variable("NucBcat") <= 1.1 * 5282))
  )}
}
```

(b)

Figure 9. Experiment specifications of the first composed model (Mc1) in SESSL . (a) Property specification regarding fraction of LR[Lrp6] and LR[CK1γ]. The first order deviation is used to describe that a variable increases or decreases, denoted as d("variableName") in SESSL. This subfigure depicts the updated property specification regarding LR[Lrp6] and LR[CK1γ] while the experiment set-up is the same as the one in Figure 5 except the model location. (b) Experiment specification regarding the fold change of nuclear β-catenin concentration when lipid rafts are removed from the model by configuring `nLR` = 0. This listing describes the experiment specification regarding nuclear β-catenin under the condition that lipid rafts are removed. In addition, the first composed model is annotated with another two experiment specifications, i.e., the adapted experiment specifications from M2 as shown in Figure 7 by updating model location.

experiments that have been successfully executed and checked, i.e., the experiment with property regarding the species CK1γ that is from the reused model M1, the experiment with an updated property regarding the species Lrp6, the two experiments with properties regarding nuclear β-catenin from the reused model M2, and the experiment with property regarding nuclear β-catenin under the condition that lipid rafts are removed, as shown in Figure 9. In addition, we start a new round of modeling within our life cycle of model composition to explore possibilities that could explain the first peak. We will briefly introduce this study in the following subsections, with a focus on the reuse of simulation experiments for semantic validation of the composed model.

5.8 Further composition: Integrating the ROS model (M3)

Our previous experiments have revealed that the first increase of β-catenin appears to be endogenous and independent of Wnt. Thus, we need to search for a mechanism, which can be composed with our model M12 so that the resultant model can still exhibit the key properties of model M12 with corresponding experiments and yield the expected outcome, i.e., an increase of nuclear β-catenin can be observed within the first hours when lipid rafts are removed, which can be expressed based on mitl$_{[a,b]}$ as follows (corresponding to Equation (10)):

$$M123_{\tilde{p}} \models_{ses12(nLR\,=\,0)} (d(\beta cat)/dt> \;=0)gtU(40,200, F(0,10,d/dt(\beta cat)< \;=0)) \tag{11}$$

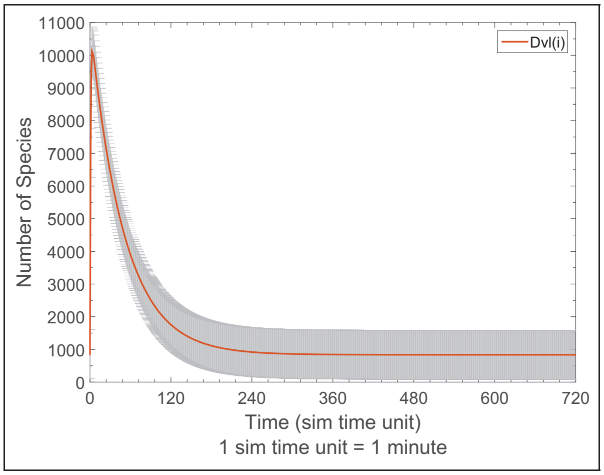


Figure 10. Simulation results of the ROS model (M3) with the average trajectory with 95% confidence interval (gray error bars).

Looking for such a mechanism we recall a recent study in which during the initiation phase of hNPCs' differentiation, an early spontaneous production of ROSs has been observed, which promotes a Dishevelled (DVL)-mediated downstream activation of canonical Wnt signaling.[75]

5.9 Model selection: The ROS model (M3)

We refer to the next reused model as the ROS/DVL model (M3), with which the first composed model M12 will be composed. A spontaneous release of ROSs from mitochondria activates DVL by releasing the redox-sensitive binding of NRX and DVL. Subsequently, cytoplasmic DVL binds to Axin and thereby leads to the activation of downstream β-catenin signal pathway, yielding a transient accumulation of β-catenin.[60,75] Only a few quantitative experimental data are available, and the model is based upon and calibrated against the findings of two publications and additional experimental data.[60,75,76]

We perform simulation experiments on the ROS model (M3) and observe the dynamics of DVL concentration, and the simulation results are presented in Figure 10. We

```
val exp = new Experiment with Observation and Hypothesis {
 model = "./RosModel.mlrj"
 simulator = SimpleSimulator()
 stopCondition = 720.0
 observe("Dvl" ~ "Cell/Dvl(i)")
 observeAt(range(0.0, 1.0, 720.0))
 replicationCondition = PointWiseConfidenceInterval("Dvl", 5, 1500.0, 0.95, range(0, 1, 5))
 assume { Average (
  F(0.0, 5.0, variable("Dvl") >= 10000 and (variable("Dvl") <= 12000))
  )}
}
```

Figure 11. Experiment specification of the ROS model (M3) in SESSL.

specify the experiment in SESSL, with the behavioral property that the DVL concentration reaches a value between 10,000 and 12,000 very quickly, directly at the beginning (we assumed within the first 5 min), mimicking the spontaneous increase in cytosolic DVL concentration observed in the wet-lab.[75] The property can be expressed based on $\mathrm{mitl}_{[a,b]}$ as:

$$M3_{\tilde{p}} \models_{ses3} F(00, 5.0, Dvl > \; = 10000gt\wedge \\ Dvl < \; = 12000) \tag{12}$$

As the fast peak of DVL concentration is the main property, an additional smoothing is not needed. The simulation results are presented in Figure 10 and the experiment specification is shown in Figure 11.

5.10 Model composition: Composing M12 and M3 (M123)

In the composed model M123, the model M3 interacts with model M12 via DVL binding to both LRP6 and Axin. Therefore, four additional rules are introduced. The kinetics of the new rules are defined based on wet-lab experiments and literature. As before, the syntax check of this white box composition does not cause any problems from a syntactical point of view. The more interesting question is whether the composition is semantically valid and whether it is able to reproduce the observations made in the wet-lab.

5.11 Semantic validation: Model (M123)

First we evaluate the composition of all models (M123) by reusing the experiment specifications of the ROS/DVL model (M3) and the first composed model (M12). In the model M12 and M2 we assumed that the peak of nuclear β-catenin within the first hours is Wnt-dependent, i.e., the property described in Equations (7) and (3), corresponding to the experiment depicted in Figure 7(a). However, this assumption does not appear to be true according to the inconsistency between wet-lab and dry-lab experiments on M12 when lipid rafts are removed. Therefore, we discard the experiment specification of M12 regarding this property (an adaptation of Figure 7(a) and Equation (7)), and reuse the remaining three experiment specifications, i.e., the one with the property regarding LRP6 and CK1γ within lipid rafts (Figure 9(a) and Equations (5) and (6)), the one with the property regarding nuclear β-catenin reaching a certain level in the later hours (an adaptation of Figure 7(b) and Equation (8)), and the one with the property regarding nuclear β-catenin under the condition lipid rafts are removed (Figure 9(b) and Equation (9)), assuming that none of the original dynamics are disrupted.

The simulation results of experimenting with M123 are presented in Figure 12, and all properties of M12 and M3 are checked successfully as expected, which shows the composition preserves the key behavior from its model components as intended.

Furthermore, we perform the experiment with the model M123 to check the property that model M12 fails, as shown in Equation (11), which has motivated us to build model M123. The simulation results are presented in Figure 12, showing that when lipid rafts are removed, the composed model M123 is indeed able to produce an early accumulation of nuclear β-catenin.

Based on M123, now the key observation from our wet-lab experiments can be reproduced, e.g., disrupting lipid rafts results in a peak of nuclear β-catenin concentration at around 2 h, without further increase.[60]

6 Discussion

Most existing approaches on semantic composability focus on composition methods where model components are connected with inputs and outputs. For example, approaches presented by Weisel et al. and Szabo and Teo depend on treating models as computable functions with inputs and outputs,[28,27] and therefore cannot be used for cases where identifying model inputs/outputs is difficult and models are composed by fusion, as in the presented case study. Moreover, semantic composability is checked based on analyzing the execution sequence of the composed model and comparing it to a reference (a perfect model or a pre-specified scenario),[15,29] either by

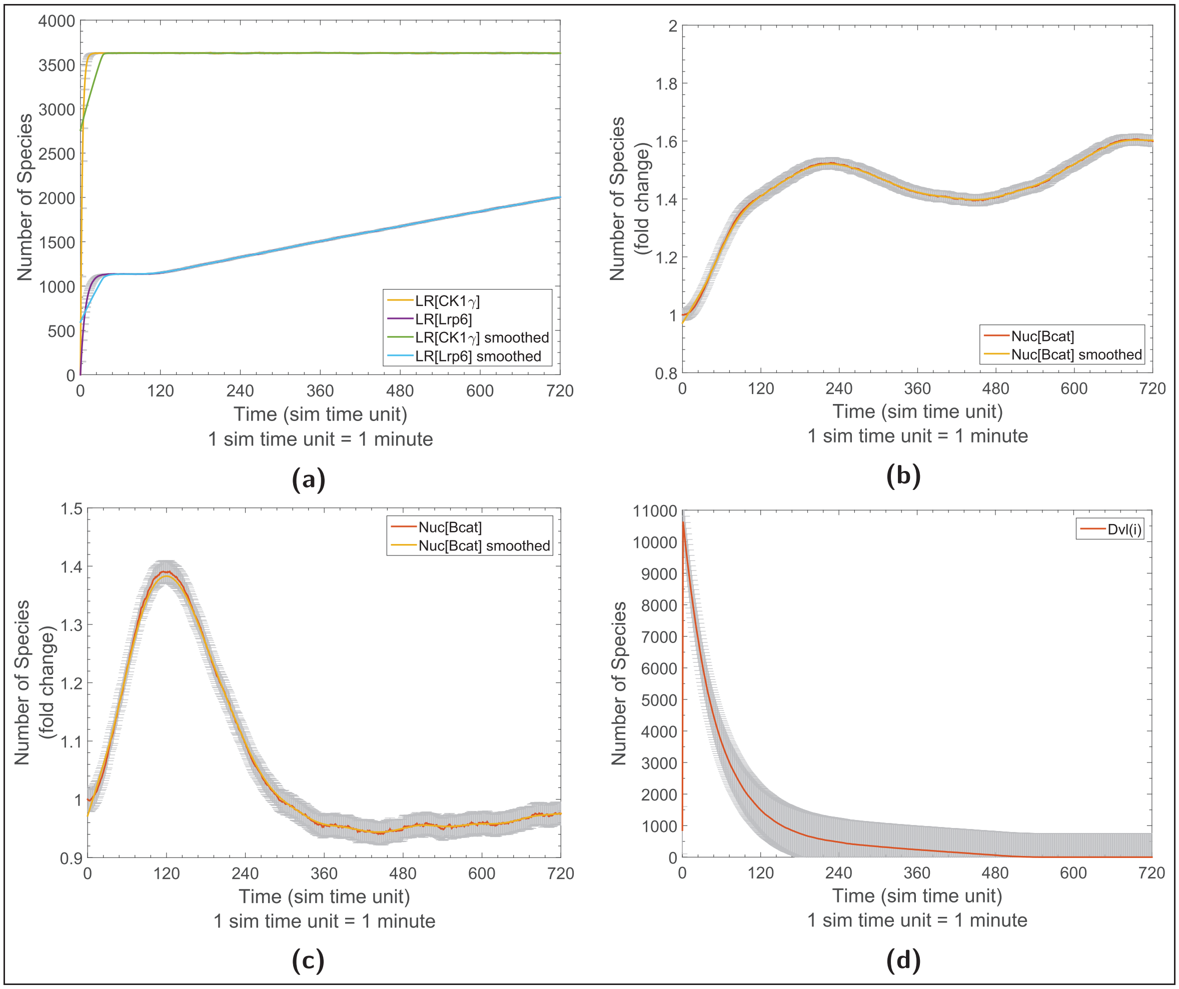


Figure 12. Simulation results of the second composed model (M123), with the average trajectories with 95% confidence interval (gray error bars) and smoothed trajectories. (a) The simulation results regarding `Lrp6` and `CK1`γ. (b) The simulation results regarding nuclear β-catenin with the default model configuration. (c) The simulation results regarding nuclear β-catenin when lipid rafts are removed from the model, i.e., `nLR` = 0. (d) The simulation results regarding DVL.

transforming model executions into labeled transition systems or through state-machine execution.[15,27,29]

Our approach takes another view and employs simulation experiments that have been performed with individual models to check model validity. Models are annotated with the specifications of their validation experiments. In the experiment specification, together with corresponding experiment conditions, the expected model behavioral properties are explicitly specified based on describing generated simulation trajectories with languages such as temporal logics. When models are composed, those experiment specifications are reused to perform experiments with the composed model and thus to provide insights on model semantic composability.

Therefore, the presented approach is not limited to a specific type of composition method. Furthermore, our approach is based on specifying and executing simulation experiments in SESSL, which can be extended to support different simulation systems. Hence, the approach is independent of concrete modeling formalisms and simulation algorithms. For illustration, an example based on Lotka–Volterra models is presented in the supplementary material, where the SBML (system biology markup language) is used for model description and the tool SBMLSim for model execution.[77,78]

As a proof of concept, the presented approach was firstly applied to a simple example of composing Lotka–Volterra models in our prior work,[5] where the often used temporal logic LTL and CSL were employed for specifying model behavioral properties and statistical model checking techniques were used for analyzing simulation results. However, to apply the approach in the case study of the realistic Wnt/β-signaling model, several additional requirements arose:

- a new language for specifying properties instead of LTL;
- a new mechanism to check simulation results other than traditional statistical model checking methods;
- a new method to deal with stochasticity within individual simulation trajectories.

Due to the extensibility of SESSL, those new requirements have been satisfied without much effort, and this also illustrates the advantage and importance of choosing SESSL as the experiment specification language. However, the possibility of adding new experimentation methods to SESSL ‘‘on-the-fly’’ also puts some responsibility on the user. If, as in our case study, existing methods do not suffice to conduct a simulation study, the choice and correct use of new methods pose a significant challenge. By documenting simulation experiments including the employed methods, SESSL contributes to the assessment, verification, and propagation of new experimentation methods. Thus, SESSL makes our approach applicable to a wide range of models and experimentation methods, as our case study exemplifies.

During the development of models via successive composition, our approach can provide assistance in checking semantic composability by automatically performing experiments with the composed model and checking simulation results against specified properties, to make sure the composition does not disrupt some key properties of model behavior. As such, with our approach, useful and fast feedback can be provided to the modeler, which facilitates the acceleration of the composition process and improvement in the quality of the composed model.

7 Conclusion

In this paper, we presented an approach to reuse simulation experiments in supporting model composition. Based on a discussion of the process of model development via composition and how simulation experiments can be involved in this process, we illustrated the specific step in this modeling process which our approach is aimed at, i.e., semantic validation of the composed model.

For demonstration, we applied the approach in a case study of developing a Wnt/β-signaling model. We relied on explicit specifications of simulation experiments in SESSL, behavioral properties expressed in $\text{mitl}_{[a,b]}$ and checked on averages of stochastic simulations, and alleviating fluctuation in simulation trajectories of stochastic models with a LOESS smoothing method. Simulation experiments of individual models to be composed were reused to generate and execute experiments on the composed model and to analyze the simulation results automatically.

As shown in the case study, the interplay of automatically generating experiments for the composed model, retrieving the user’s expected outcome of those experiments, and annotating and specifying new experiments on the resulting models by the user, can contribute to the coherency of the composition regarding model behaviors, and provide additional structure into developing and validating models by successive composition.

Funding

This work was supported by the CSC (China Scholarship Council) and the National Natural Science Foundation of China (grant number 61374185), the German Research Foundation (DFG) (grant numbers UH-66/15-1 and UH-66/14).

8 References

1. Page EH and Opper JM. Observations on the complexity of composable simulation. In: *Proceedings of the 31st winter simulation conference*, Phoenix, Arizona, USA, 5–8 December 1999, pp. 553–560. New York, NY: ACM.
2. Petty MD and Weisel EW. A composability lexicon. In: *Proceedings of the spring 2003 simulation interoperability workshop*, Orlando, Florida, USA, 30 March–4 April 2003, pp.181–187. Orlando, FL: Simulation Interoperability Standards Organization.
3. Balci O. Verification validation and accreditation of simulation models. In: *Proceedings of the 29th Winter Simulation Conference*, Atlanta, Georgia, USA, 7–10 December 1997, pp.135–141. Washington, DC: IEEE.
4. Stelling J, Mendes P, Klipp E, et al. Defining modeling strategies for Systems Biology. Report, FutureSysBio Workshop, Göteborg, Sweden, 2011.
5. Peng D, Ewald R and Uhrmacher AM. Towards semantic model composition via experiments. In: *Proceedings of the 2nd ACM SIGSIM conference on principles of advanced discrete simulation*, Denver, Colorado, USA, 18–21 May 2014, pp.151–162. New York, NY: ACM.
6. Balci O. A life cycle for modeling and simulation. *Simulation* 2012; 88: 870–883.
7. Sargent RG. Verification and validation of simulation models. *J Simul* 2013; 7: 12–24.
8. Kasputis S and Ng HC. Model composability: Formulating a research thrust: Composable simulations. In: *Proceedings of the 32nd winter simulation conference*, Orlando, Florida, USA, 10–13 December 2000, pp. 1577–1584. San Diego, CA: Society for Computer Simulation International.
9. Szabo C and Teo YM. An approach to semantic-based model discovery and selection. In: *Proceedings of the 43rd winter simulation conference*, Phoenix, Arizona, USA, 11–14 December 2011, pp.3054–3066. Washington, DC: IEEE.
10. Robinson S, Arbez G, Birta LG, et al. Conceptual modeling: Definition, purpose and benefits. In: *47th winter simulation conference*, Huntington Beach, California, USA, 6–9 December 2015, pp.2812–2826. Washington, DC, USA: IEEE.
11. Chwif L, Banks J, de Moura Filho JP, et al. A framework for specifying a discrete-event simulation conceptual model. *J Simul* 2013; 7: 50–60.

12. Robinson S. Conceptual modelling for simulation part i: Definition and requirements. *J Oper Res Soc* 2008; 59: 278–290.
13. Tolk A, Diallo SY, King RD, et al. Conceptual modeling for composition of model-based complex systems. In: *Conceptual modeling for discrete-event simulation*, 2010, pp.355–381. Boca Raton, FL: CRC.
14. Teo YM and Szabo C. Codes: An integrated approach to composable modeling and simulation. In: *Proceedings of the 41st annual simulation symposium*, Ottawa, Ontario, Canada, 13–16 April 2008, pp.103–110. Washington, DC, USA: IEEE.
15. Moradi F, Ayani R and Tan G. A rule-based approach to syntactic and semantic composition of BOMs. In: *Proceedings of the 11th IEEE international symposium on distributed simulation and real-time applications*, Chania, Crete Island, Greece, 22–24 October 2007, pp.145–155. Washington, DC: IEEE.
16. Neal ML, Cooling MT, Smith LP, et al. A reappraisal of how to build modular, reusable models of biological systems. *PLoS Computational Biology* 10(10): e1003849.
17. Davis PK and Anderson RH. Improving the composability of DoD models and simulations. *JDMS* 2004; 1: 5–17.
18. Li C, Donizelli M, Rodriguez N, et al. Biomodels database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst Biol* 2010; 4: 92.
19. Randhawa R, Shaffer CA and Tyson JJ. Model composition for macromolecular regulatory networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2010; 7: 278–287.
20. Röhl M and Uhrmacher AM. Composing simulations from XML-specified model components. In: *Proceedings of the 38th Winter Simulation Conference*, Monterey, California, USA, 3–6 December 2006, pp.1083–1090. Washington, DC, USA: IEEE.
21. Szabo C and Teo YM. On syntactic composability, model reuse. In: *Proceedings of the first Asia international conference on modelling & simulation*, Phuket, Thailand, 27–30 March 2007, pp.230–237. Washington, DC, USA: IEEE.
22. Vass MT, Shaffer CA, Ramakrishnan N, et al. The JigCell model builder: A spreadsheet interface for creating biochemical reaction network models. *IEEE/ACM Trans Comput Biol Bioinform* 2006; 3(2): 155–164.
23. Szabo C and Teo YM. An analysis of the cost of validating semantic composability. *J Simul* 2012; 6: 152–163.
24. Szyperski C, Gruntz D and Murer S. Component software: Beyond object-oriented programming. Boston, Massachusetts, USA: Addison-Wesley Longman Publishing Co., Inc, 2002.
25. Overstreet CM, Nance RE and Balci O. Issues in enhancing model reuse. In: *Proceedings of the first international conference on grand challenges for modeling and simulation*, San Antonio, Texas, USA, 27–31 January 2002, pp. 27–31. San Diego, CA: Society for Computer Simulation International.
26. Petty MD and Weisel EW. A formal basis for a theory of semantic composability. In: *Proceedings of the spring 2003 simulation interoperability workshop*, Orlando, Florida, USA, 30 March–4 April 2003, pp.416–423. Orlando, FL: Simulation Interoperability Standards Organization.
27. Weisel E, Petty M and Mielke R. Validity of models and classes of models in semantic composability. In: *Proceedings of the fall 2003 simulation interoperability workshop*, Orlando, Florida, USA, 14–19 September 2003. Orlando, FL: Simulation Interoperability Standards Organization.
28. Szabo C and Teo YM. An approach for validation of semantic composability in simulation models. In: *Proceedings of the 2009 ACM/IEEE/SCS 23rd workshop on principles of advanced and distributed simulation*, Lake Placid, New York, USA, 22–25 June 2003, pp.3–10. Washington, DC: IEEE.
29. Szabo C, Teo YM and See S. A time-based formalism for the validation of semantic composability. In: *Proceedings of the 41st winter simulation conference*, Austin, Texas, USA, 13–16 December 2009, pp.1411–1422. Washington, DC: IEEE.
30. Traore? MK. Analyzing static and temporal properties of simulation models. In: *Proceedings of the 38th winter simulation conference*, Monterey, California, USA, 3–6 December 2006, pp.897–904. Washington, DC, USA: IEEE.
31. Verbraeck A and Valentin EC. Design guidelines for simulation building blocks. In: *Proceedings of the 40th winter simulation conference*, Miami, Florida, USA, 7–10 December 2008, pp.923–932. Washington, DC: IEEE.
32. Pedersen M and Phillips A. Towards programming languages for genetic engineering of living cells. *J R Soc Interface* 2009; 6(Suppl 4): S437–S450.
33. Smith LP, Bergmann FT, Chandran D, et al. Antimony: A modular model definition language. *Bioinformatics* 2009; 25: 2452–2454.
34. Smith LP, Hucka M, Hoops S, et al. SBML level 3 package specification: Hierarchical model composition, 2013. Available at: http://sbml.org/Special/specifications/sbml-level-3/version-1/comp/sbml-comp-version-1-release-3.pdf
35. Novère NL, Finney A, Hucka M, et al. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat Biotechnol* 2005; 23: 1509–1515.
36. Grimm V, Berger U, DeAngelis DL, et al. The ODD protocol: A review and first update. *Ecol Modell* 2010; 221: 2760–2768.
37. Grimm V, Polhill P and Touza J. Documenting social simulation models: The ODD protocol as a standard. In: *Simulating social complexity - a handbook*. Berlin: Springer, 2013, pp.117–133.
38. Gennari JH, Neal ML, Galdzicki M, et al. Multiple ontologies in action: Composite annotations for biosimulation models. *J Biomed Inform* 2011; 44: 146–154.
39. Chreyh R and Wainer G. CD++ repository: an internet based searchable database of DEVS models and their experimental frames. In: *Proceedings of the 2009 spring simulation multiconference*, San Diego, California, USA, 22–27 March 2009. 159pp. San Diego, CA: Society for Computer Simulation International.
40. Henkel R, Wolkenhauer O and Waltemath D. Combining computational models, semantic annotations and simulation

experiments in a graph database. *Database* 2015; 2015: bau130.

41. Wang S and Wainer G. Semantic selection for model composition using SAMSaaS. In: *Proceedings of the symposium on theory of modeling & simulation: DEVS integrative M&S symposium*, Alexandra, Virginia, USA, 12–15 April 2015 pp.25–32. San Diego, CA: Society for Computer Simulation International.
42. Cooper J, Vik JO and Waltemath D. A call for virtual experiments: accelerating the scientific process. *Prog Biophys Mol Biol* 2015; 117: 99–106.
43. Cellier F. *Continuous system modeling*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. 1991.
44. Rybacki S, Haack F, Wolf K, et al. Developing simulation models-from conceptual to executable model and back-an artifact-based workflow approach. In: *Proceedings of the 7th international ICST conference on simulation tools and techniques*, Lisbon, Portugal, 17–19 March 2014, pp.21–30. Brussels, Belgium: ICST.
45. Zeigler B. *Theory of modelling and simulation*. New York: John Wiley, Wiley-Interscience Publication, 1976.
46. Waltemath D, Adams R, Bergmann FT, et al. Reproducible computational biology experiments with SED-ML-the simulation experiment description markup language. *BMC Syst Biol* 2011; 5: 198.
47. Ewald R and Uhrmacher AM. SESSL: A domain-specific language for simulation experiments. *ACM Trans Model Comput Simul* 2014; 24: 11.
48. Clarke E, Grumberg O and Peled D. *Model checking*. Cambridge, Massachusetts: MIT Press, 1999.
49. Sen K, Viswanathan M and Agha G. On statistical model checking of stochastic systems. In: *Proceedings of the 17th international conference on computer aided verification*, Edinburgh, Scotland, UK, 6–10 July 2005, pp.266–280. Berlin: Springer.
50. Luboschik M, Rybacki S, Haack F, et al. Supporting the integrated visual analysis of input parameters and simulation trajectories. *Comput Graph* 2014; 39: 37–47.
51. Fages F and Rizk A. On the analysis of numerical data time series in temporal logic. In: *Computational methods in systems biology*. Berlin: Springer, pp.48–63.
52. Leye S, Himmelspach J and Uhrmacher AM. A discussion on experimental model validation. In: *11th international conference on computer modelling and simulation*, Cambridge, UK, 25–27 March 2009, pp.161–167. Washington, DC, USA: IEEE.
53. Courtot M, Juty N, Knüpfer C, et al. Controlled vocabularies and semantics in systems biology. *Mol Syst Biol* 2011; 7: 543.
54. Carrillo M, Góngora PA and Rosenblueth DA. An overview of existing modeling tools making use of model checking in the analysis of biochemical networks. *Front Plant Sci* 2012; 3: 155.
55. Brim L, Češka M and Šafránek D. Model checking of biological systems. In: *Formal methods for dynamical systems*. Springer, 2013, pp.63–112.
56. Maus C, Rybacki S and Uhrmacher AM. Rule-based multilevel modeling of cell biological systems. *BMC Syst Biol* 2011; 5.
57. Maler O and Nickovic D. Monitoring temporal properties of continuous signals. In: *Formal techniques, modelling and analysis of timed and fault-tolerant systems*. Heidelberg: Springer, 2004, pp.152–166.
58. Legay A, Delahaye B and Bensalem S. Statistical model checking: An overview. In: *Runtime verification*. Berlin: Springer, pp.122–135.
59. Mazemondet O, Hubner R, Frahm J, et al. Quantitative and kinetic profile of Wnt/β-catenin signaling components during human neural progenitor cell differentiation. *Cell Mol Biol Lett* 2011; 16: 515–538.
60. Haack F, Lemcke H, Ewald R, et al. Spatio-temporal model of endogenous ros and raft-dependent Wnt/ beta-catenin signaling driving cell fate commitment in human neural progenitor cells. *PLoS Comput Biol* 2015; 11: 1–28.
61. Burrage K, Burrage P, Donovan D, et al. Populations of models, experimental designs and coverage of parameter space by Latin hypercube and orthogonal sampling. *Procedia Computer Science* 2015; 51: 1762–1771.
62. Mazemondet O, John M, Leye S, et al. Elucidating the sources of β-catenin dynamics in human neural progenitor cell. *Plos One* 2012; 7: e42792.
63. Cleveland WS. Robust locally weighted regression and smoothing scatterplots. *J Am Stat Assoc* 1979; 74: 829–836.
64. Himmelspach J and Uhrmacher AM. Plug'N simulate. In: *Proceedings of the 40th annual simulation symposium*, Washington, DC, USA, 26–28 March 2007, pp.137–143. Washington, DC: IEEE.
65. Clevers H and Nusse R. Wnt/β-catenin signaling and disease. *Cell* 2012; 149: 1192–1205.
66. Sakane H, Yamamoto H and Kikuchi A. LRP6 is internalized by Dkk1 to suppress its phosphorylation in the lipid raft and is recycled for reuse. *J Cell Sci* 2010; 123: 360–368.
67. Lloyd-Lewis B, Fletcher AG, Dale TC, et al. Toward a quantitative understanding of the Wnt/β-catenin pathway through simulation and experiment. *Wiley Interdiscip Rev Syst Biol Med* 2013; 5: 391–407.
68. Mirams GR, Byrne HM and King JR. A multiple timescale analysis of a mathematical model of the Wnt/ beta-catenin signalling pathway. *J Math Biol* 2010; 60: 131–160.
69. Warnke T, Helms T and Uhrmacher AM. Syntax and semantics of a multi-level modeling language. In: *Proceedings of the 3rd ACM SIGSIM conference on principles of advanced discrete simulation*, London, UK, 10–12 June 2015. pp.133–144. New York: ACM.
70. Lee E, Salic A, Krüger R, et al. The roles of apc and axin derived from experimental and theoretical analysis of the wnt pathway. *PLoS Biol* 2003; 1: e10.
71. Peng D, Steiniger A, Helms T, et al. Towards composing ML-rules models. In: *Proceedings of the 45th winter simulation conference*, Washington, DC, USA, 8–11 December 2013. pp.4010–4011. Washington, DC: IEEE.
72. Özhan G, Sezgin E, Wehner D, et al. Lypd6 enhances Wnt/ β-catenin signaling by promoting Lrp6 phosphorylation in raft plasma membrane domains. *Dev Cell* 2013; 26: 331–345.
73. Haack F, Burrage K, Redmer R, et al. Studying the role of lipid rafts on protein receptor bindings with cellular

automata. *IEEE/ACM Trans Comput Biol Bioinform* 2013; 10: 760–770.

74. Hannoush RN. Kinetics of Wnt-driven β-catenin stabilization revealed by quantitative and temporal imaging. *PloS one* 2008; 3: e3498.
75. Rharass T, Lemcke H, Lantow M, et al. Ca^{2+}-mediated mitochondrial ROS metabolism augments Wnt/β-catenin pathway activation to facilitate cell differentiation. *J Bio Chem* 2014; 289: 27937–27951.
76. Funato Y, Michiue T, Asashima M, et al. The thioredoxin-related redox-regulating protein nucleoredoxin inhibits Wnt–β-catenin signalling through dishevelled. *Nat Cell Biol* 2006; 8: 501–508.
77. Hucka M, Finney A, Sauro HM, et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 2003; 19: 524–531.
78. The Sytems Biology Institute. Simulation software for systems biology, http://systems-biology.org/software/simulation/sbmlsim.html, (2016, accessed 14 March 2017).

Author Biographies

Danhua Peng is a PhD student at the University of Rostock, Institute of Computer Science, Rostock, Germany. She received her master's degree from the National University of Defense Technology (China) in 2012. Her research interests include model development based on simulation experiments, model reuse, and semantic model composition.

Tom Warnke is working as a PhD student at the University of Rostock, Institute of Computer Science. He received his master's degree in Computer Science from the University of Rostock in 2014. His research interests include modeling language design, model verification, and computational demography.

Fiete Haack is working in Leibniz Institute for Farm Animal Biology (FBN), Research Unit ''Functional Genome Analysis'', Dummerstorf, Germany. He received his PhD degree from the University of Rostock in 2016 and his master's degree in Bioinformatics from the Free University Berlin in 2009. His research interests are spatial aspects of cellular signal transduction and specifically cell surface receptor dynamics and membrane organization.

Adelinde M Uhrmacher is a professor at the University of Rostock, Institute of Computer Science, and the head of the modeling and simulation group since 2000. She received her PhD in Computer Science in 1992 from the University of Koblenz. As a scholar of the Alexander of Humboldt Foundation she visited the University of Tuscon, Arizona, 1993. Afterwards she joined as Senior Researcher the Computer Science department of the University of Ulm. She has been on the editorial board of variety of journals (including editor in chief of SCS Simulation (2000-2006), editor in chief of ACM Transactions of Modeling and Computer Simulation (since 2013), associate editor at BMC Systems Biology (since 2008). Her research focuses on the development of modeling and simulation methods, and their application in diverse fields like demography, and cell biology. Of particular interest are spatial and stochastic multi-level modeling and simulation methods.