

Developing simulation models - from conceptual to executable model and back - an artifact-based workflow approach

Stefan Rybacki, Fiete Haack, Karsten Wolf, and Adelinde M. Uhrmacher
Institute of Computer Science
University of Rostock
Rostock, Germany
{firstname.lastname}@uni-rostock.de

ABSTRACT

Developing a model for simulation is an art and a science. The question is how this process can be suitably supported. Integrating workflows into simulation systems promises user guidance, documentation and reproducibility of this process. However, the highly interactive, partly concurrent, partly optional nature of the modeling process challenges traditional activity-based workflow approaches. This is intensified by its multiple inter-dependencies and the need for an easy extension. We will illuminate this based on a modeling example from cell biology. To support the required flexibility, we propose an artifact-based workflow approach instead. Conceptual model, formal model and different data and information sources are specified declaratively as artifacts. The life cycle of an artifact is defined by stages, guards, milestones, and sentries, following the Guard-Stages-Milestone (GSM) approach. It is shown that the declarative specification provides a better fit for the process of developing a model.

Categories and Subject Descriptors

I.6.7 [Simulation and Modeling]: Simulation Support Systems—*Environments*; D.2.7 [Software Engineering]: Distribution, Maintenance, and Enhancement—*Extensibility, Enhancement*

Keywords

Modeling, Conceptual Modeling, Workflow, Artifact

General Terms

Design, Human Factors, Reliability, Documentation

1. INTRODUCTION

Independently of interpreting modeling as a science [33] or rather as an art [34], it is undisputed that developing the

model is the most difficult part of a simulation study. “Model building requires an ability to analyze a problem, abstract from its essential features, select and modify basic assumptions that characterize the system, and then enrich and elaborate the model until a useful approximation results.” [34, p.20]. In this process, a vast amount of heterogeneous information needs to be collected, screened, and analyzed. To do so effectively requires that the modeler has an intimate relationship to the system and phenomena to be modeled, because abstraction and selective enrichment of models are a key issue [35]. In addition, experiences in executing simulation experiments are required as well, as those typically are needed to answer the question whether a useful approximation has been achieved. Thus, developing a model is an intricate process requiring diverse skills and experiences. Consequently, this process has been described in process models (modeling and simulation life-cycle models), in which system analysis, developing the conceptual model, developing the formal model, and executing experiments with the former are typically iteratively interwoven [1, 31, 8, 23]. Quests to realize more structured user guidance based on these process models, e.g., [39], are caught in between providing meaningful support and at the same time allowing for sufficient flexibility. The introduction of workflow management systems into the modeling and simulation life-cycle promises the possibility to balance between user support and flexibility. In addition, by exploiting workflow systems in the modeling process, documentation and provenance of models is supported, which are central requirements for improving current practice in modeling and simulation [20, 21].

2. BACKGROUND

In the following we briefly discuss modeling and simulation life-cycles. We then dedicate a section to the main product of this process, i.e., the model, before we come up with a modeling and simulation life-cycle that will be used as reference throughout the paper. Finally, we conclude with discussing different workflow approaches, and the use of workflows in the context of modeling and simulation.

2.1 Modeling and simulation life-cycle

A couple of different life-cycle models have been proposed for modeling and simulation [18, 14, 30] to structure the process of model creation into different phases. A salient feature of all life-cycles is their iterative nature, in which the model is successively refined by elaboration and enrichment

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Simutools 2014, March 17-19, Lisbon, Portugal

Copyright © 2014 ICST 978-1-63190-007-5

DOI 10.4108/icst.simutools.2014.254650

[34, 35]. Concrete methods are not part of the life-cycle but are reserved for the concrete instantiation of a life-cycle, as those are application dependent. Intelligent techniques can help selecting concrete methods for the problem at hand, e.g., selecting suitable methods for executing models [6, 15]. An intrinsic and important part of each modeling and simulation life-cycle are validation and verification (V&V) processes [31]. As argued by [23], this should imply specific tests for the results of each phase in the modeling and simulation life-cycle. For instance, conceptual model, formal model, executable model, simulation results, raw data, and prepared data should all be subject to validation and verification. Some analysis might refer to solely one result, more often however the inter-dependencies between the results of the different phases have to be taken into account. Also, if based on the analysis a result of a phase is revised than all depending phases need to be revisited.

Thus, the modeling process presents itself as a highly interactive, intricate, and application dependent process. Therefore, it exists a consensus of experts that rigid models, e.g., water fall models, do not appear suitable to describe and structure this process [32].

2.2 Conceptual model and formal model

As argued before the model is the primary product of the modeling and simulation life-cycle. However when referring to model, we find terms like abstract model, conceptual model, communicative model, formal model, programmed model, computational model, executable model, operational model, and experimental model [25, 18, 3, 23]. Not all approaches distinguish between models at the same granularity, e.g., in [23] we find conceptual model, formal model, and executable model. In addition, meaning of those terms varies. Sometimes abstract or conceptual model means the yet not expressed perception that a person has about a system. Sometimes the term conceptual or communicative model is used to denote a model, which describes the system in terms of variables and inter-dependencies between those variables. In addition, the model might be annotated with other multi-media information. Diverse methods are proposed to support conceptual modeling. Among them some authors see also formal approaches like Petri Nets and DEVS [11, 3]. Thereby, conceptual and formal model are no longer distinguishable. Programmed model, computational model, or executable model all refer to an implementation on a computer. The operational or experimental model denotes the validated model, ready to use for experiments.

Despite the diversity of the terms used, most of the approaches distinguish between conceptual, formal, and executable model. Thereby, a clear distinction between conceptual and formal model is of central importance. The conceptual model is one of the first products and it shall be independent from the simulation language or the simulator that will be used later on. This perception is widely agreed upon [3]. An explicit phase for creating an executable model can be neglected in cases where the formal model is automatically transformed into an executable one.

2.3 A Process Model of the Modeling and Simulation Life Cycle

The modeling and simulation life-cycle Figure 1 is based on the life-cycle proposed in [23]. It conforms with other established life-cycle models emphasizing the interactive nature of this process [18, 14, 30]. It starts with an explicit specification. The question to be pursued very much decides what level of abstraction is needed. The conceptual model can be inspected and also first conclusions about behavioral implications and thus its credibility can be drawn. Typically variables and their scale and inter-dependencies are identified, flow charts, or UML diagrams can be used for further structuring, additional annotations with text or data help to check where something might be missing. However, one essential result of the conceptual model should be the clarification of what kind of modeling formalism to use for the next step, e.g., does stochasticity or space play a role, is a discrete time step or a discrete event approach more suitable, are variable structures required, should multiple organization levels including up and downward causation be considered etc. The next step is then to transform the conceptual model into the formal model. Often the formal model is directly executable which renders the next phase from the point of view of the user superfluous. Based on the formal model, we can already perform first computer-supported validation and verification steps. The collected data/information are central in the validation and verification of the models, as they relate conceptual and formal model to the system to be modeled and analyzed by simulation. These experiments comprise again many different steps, require considerable time, and also deserve some workflow support - however due to their rather fixed nature the degree of required flexibility is different and thus also the workflow approach best pursued, see e.g., [27]. While the simulation experiments are running, new information can be collected. This is by itself a rather complex process as diverse sources for information need to be inspected, analyzed, possibly processed, and documented, e.g., do the data come from a different model, have they been measured by oneself or others[17], again this process requires some support which is the subject of traditional scientific workflows [12]. From this it should be clear that to support the process of modeling by a workflow system a high degree of flexibility is required.

2.4 Different workflow approaches

Most approaches model workflows [38] as a set of elementary activities that are arranged using various control flow patterns that permit sequential, parallel, or alternative execution and can be structured using complex activities that represent subprocesses. Several languages support this view, for instance the Business Process Model and Notation (BPMN) [10] as the currently most popular one. These languages provide a concise, elegant, and well readable representation of well structured routine procedures but have well-known limitations when processes with a large amount of flexibility need to be expressed. There are attempts to inject flexibility into activity-based workflow models such as the ADEPT approach [5] but they mainly focus on punctual modifications such as insertion and deletion of activities rather than major rearrangements in the control logic.

As an alternative to activity-based workflows, declarative

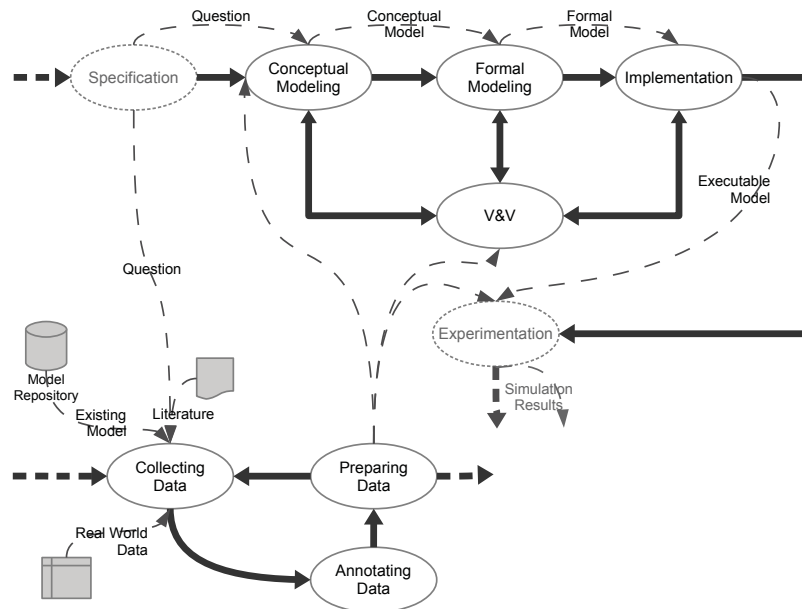


Figure 1: Modeling and simulation life-cycle process with essential steps and their dependencies, as well as part of the information flow between the different steps (dashed arrows); bold dashed arrows indicated process connections outside of this life-cycle

workflows have been introduced, for instance the DECLARE approach [22]. A declarative workflow is a collection of constraints for the execution of a workflow. A built-in reasoner of a workflow management system can infer activities to be executed next. Flexibility is achieved through a modification of the constraint base and subsequent re-evaluation. An inserted or deleted constraint may have severe effects on the resulting execution order which means that major modifications can be achieved more easily than with activity-based workflows.

As a class of declarative workflows, artifact centric processes [19, 4] have been introduced. In an artifact centric process, constraints are derived from business relevant *artifacts*, e.g., documents or manufactured items. An artifact has an inherent life-cycle. Activities cause progress in that life-cycle of one or several artifacts. In this sense, the life-cycle of an artifact imposes constraints on the execution order of activities. Consequently, an artifact centric process is mainly controlled by the artifacts. A reasoning (or planning) component in a process engine derives execution orders of activities from the artifacts. Again, flexibility is achieved by adding, deleting, or modifying artifacts. Compared to other declarative workflow approaches, artifacts are more easily understood by non-experts than, e.g., temporal logic specifications of constraints [2].

2.5 Workflows in simulation

In modeling and simulation we find predominantly scientific workflows, which are data driven and activity-based [36, 37, 9]. Scientific workflows are concerned with describing, documenting, and synchronizing the inter-dependent activities and computations that belong to scientific problem-solving, such as simulation studies. Accordingly, different

approaches exist that exploit workflows for modeling and simulation [28, 27, 24].

For instance, the experimentation process includes the configuration of the experiment, executing one configuration, which in turn might imply multiple replications (in case of dealing with stochastic simulations). How many replications are needed has to be determined by analyzing the simulation output. Each replication represents one simulation run, with a length that might be calculated, e.g., based on steady state analysis. Depending on the result of a configuration, an optimization algorithm might determine the next configuration. Thus, one experiment might comprise multiple configurations. In [29] we showed how workflows can be used to support the execution and automatic documentation of rather complex experiments based on a specification, e.g., based on a specification language like SESSL [7]. However, a simulation study aimed at developing a model requires many of these experiments and user interactions to successively elaborate and enrich the model, specify the experiment, execute and interpret it. The process of modeling could of course be added as further step in experimentation, however, as argued above (Section 2.2), this would not do justice to this process.

There exists approaches that take a more fine-grained workflow view on modeling [9, 24]. For instance, with a focus on Finite Element Methods (FEM), the pre-processing step for executing a model (i.e., developing a model and configuring the experiment) has been designed as a workflow in which the respective tasks are pipelined: define geometry data, define material parameter, create FEM parameters, adjust boundary conditions, adjust initial conditions, and chose matrix solver [24]. The solving phase, i.e., executing

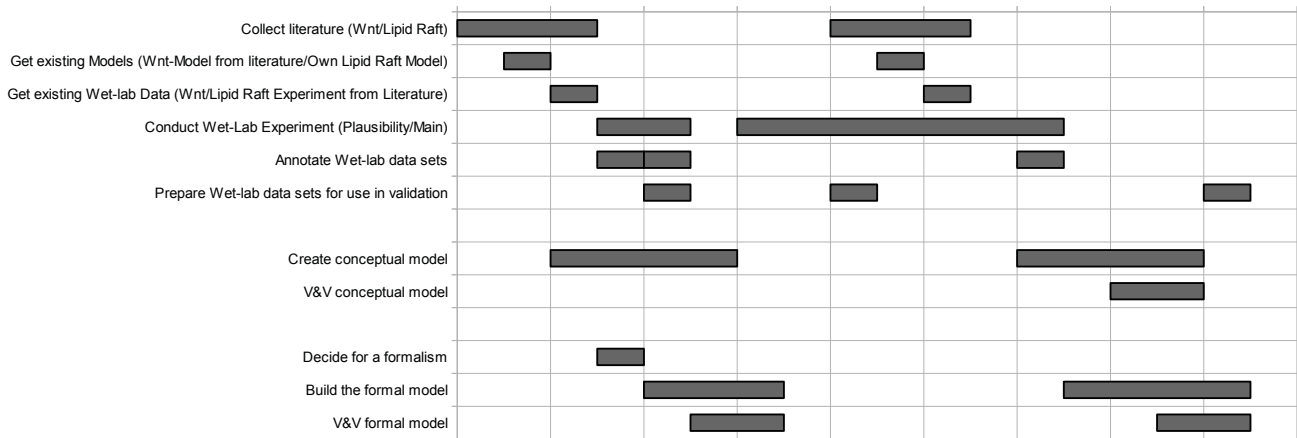


Figure 2: Overview of activities performed in the modeling process presented in the case study. Activities are shown in a gantt chart. It illustrates the order of activities executed (sequential/parallel)

the experiment (in this case one simulation run), and the post-processing phase, i.e., where the results are visualized and interpreted, complete the workflow. This pipeline focus on the executable model and stands in the tradition of scientific workflows (see e.g., cases studies in MyExperiment [26]).

3. CASE STUDY

To evaluate the suitability of a specific workflow approach the modeling life-cycle presented in Section 2.3 is used. This is combined with a modeling example from the real world which serves as test case to identify problems describing the process as workflow.

3.1 The Modeling Process

The chosen example originates in the domain of cellular biology. It deals with the creation of a simulation model that investigates the influence of cell membrane-related processes (lipid rafts) on a specific, intra-cellular signaling pathway (Wnt-pathway). The model is based on already existing mathematical models of the pathway under study, experimental measurements and additional qualitative and quantitative data from literature. Figure 2 shows the activities that were carried out to create the simulation model (note that this was done, without workflow support). The entire process can be separated into three main parts. Firstly, there are activities that deal with the collection and gathering of data that form the basis of the modeling process and of the simulation study later conducted. Secondly, there are activities that deal with the creation of a conceptual model and its verification and validation. Thirdly, activities for the creation of a formal model based on the conceptual model and its verification and validation were performed. Overall this matches with the general modeling life-cycle, except the absence of activities explicitly involving an executable model. This is due to the fact, that in our example the executable model can be automatically derived from the formal model.

In detail, the following steps were undertaken to produce the formal model. Note that the presented modeling process aims

at extending existing models of the Wnt-pathway, instead of creating a new model from scratch.

Therefore information about already existing models and additional in-vitro and in-silico data were gathered and evaluated. Based on the literature and the data available the conceptual model is developed. During this process, one of the existing models was selected as reference model, namely the one that fitted the requirements (for our simulation study) best. In a next step experimental data suitable for subsequent model parametrization were extracted and the formalisms for the implementation of the formal model was chosen.

Note, that in our use case, the formalism chosen for the (new) model was different to the formalism of the reference model. Therefore the reference model had to be translated and validated, before the formal model could be extended. Concurrently a first wet-lab experiment was conducted to check whether the given conditions allow an evaluation of the scientific question by further in-vitro studies (plausibility). In our specific use case, it had to be evaluated, whether lipid rafts are actually present in the cell line designated for the main in-vitro studies.

Having the formal model of wnt pathway implemented and validated, the actual evaluation of the scientific question could then be pursued. Therefore, further literature about lipid rafts had to be gathered and evaluated, or literally spoken, one had to step back and start from the beginning. Subsequently the gathered information about lipid rafts was incorporated in the conceptual model, that contained only the Wnt pathway so far. At the same time the formal model was extended by membrane-related processes (i.e., lipid rafts dynamics) accordingly. Concurrently parametrization and validation information from literature (in-silico and in-vitro data) had to be annotated and prepared for the validation as well as for the simulation study. Meanwhile the main wet-lab experiments were already set up and running. Once the in-vitro study was finished, V&V was applied to both, conceptual (face validation) and formal model (face validation, fitting and cross validation). Having the formal model

validated thoroughly, the simulation study could now be executed.

The modeling process proved to be highly iterative going through the same phases multiple times. It is important to notice that some activities took longer to perform than others, e.g., conducting a wet-lab experiment compared to gathering information for revising the model from literature. However, even long-term activities, like wet lab experiments or calculation intensive in-silico experiments, typically do not occupy the modeler the entire time. This time is used to perform other activities, e.g., starting to work on the conceptual model, or to annotate, process and compile data previously gathered from the literature. At the same time, working on the conceptual model is alternated with working on the formal model and vice versa. Overall, it seems that the modeling process comprises a variety of closely intertwined and concurrent activities, like literature study, model creation/extension and simulation experiments and wet-lab measurements; i.e. many tasks the modeler starts performing are not necessarily finished first before performing another one, but the modeler rather switches back and forth between them.

3.2 Mapping to a Workflow Net

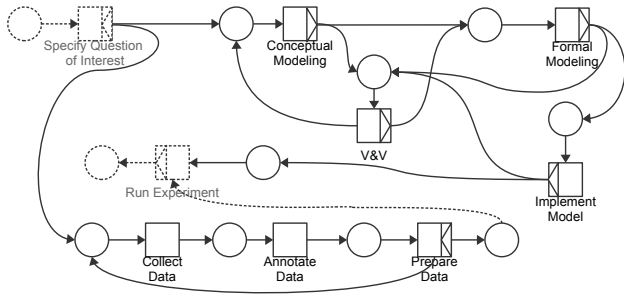


Figure 3: The basic model and simulation life-cycle as workflow net

The question is how to map this modeling process into workflows. Workflow nets are a description language for activity-based workflows [38]. A description of the modeling and simulation life-cycle (see Figure 1) with workflow nets is shown in Figure 3. Whereas the structure of the workflow appears as a straightforward translation, the resulting workflow is not sufficiently flexible. For instance, it is not possible to go back from formal to conceptual modeling without going through V&V of the formal model. However, one could argue that also in the life-cycle model this is not foreseen either, but the case study showed that this seems to be desired. Figure 4 extends the existing process model so that it is possible to go back from formal modeling to conceptual modeling without going through V&V. Another problem is caused by the concurrency of activities. Whereas those can be handled by the activity-based workflow in principle, particularly the optional interaction of activities in gathering data and developing the model requires a rather careful workflow design. It becomes more complex when adding constraints over multiple activities, e.g., it might be desired to invalidate a successful V&V result for the formal model if the V&V for the conceptual model failed.

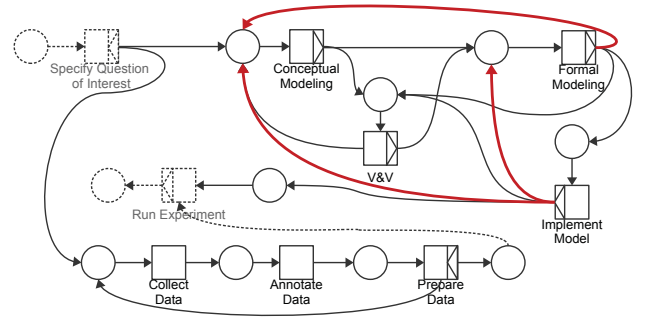


Figure 4: The extended workflow net adding the ability to go back in the modeling process without the need to go through V&V every time

This is even elevated, when there is a new activity to be added (Figure 5). For instance, if the model shall be accredited. The workflow modeler needs to be aware where to add the new activity and how to connect it to the existing activities, without changing the behaviour of the previous workflow model. This requires again a very careful workflow design. In addition, this general scheme makes it rather difficult to react to application specific strategies. Sometimes no conceptual model is needed, e.g., when the question is whether an existing formal model is also valid for a new set of data. Please note that we already assume that the workflow net uses templates so that, e.g., for formal modeling a set of techniques are available. Still for any changes of the structure, i.e., new interaction capabilities or new nodes, the entire workflow net has to be checked for consistency and semantic soundness.

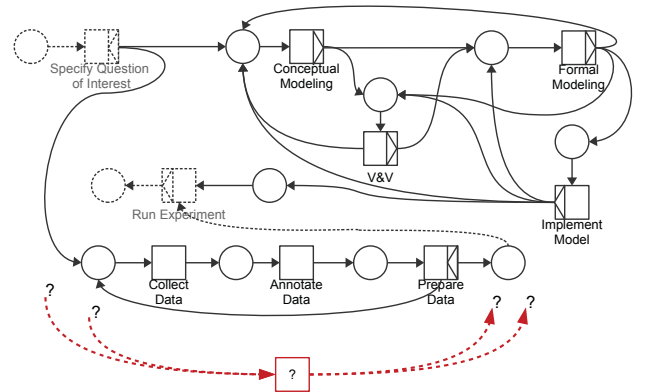


Figure 5: Adding additional accrediting for V&V, can be complex

4. AN ARTIFACT-BASED APPROACH FOR MODEL DEVELOPMENT

An artifact-based workflow directs the focus from the entire workflow net to the involved artifacts and the constraints defined by those. The constraints influence activities which work on the artifacts. This allows to define constraints in a declarative manner, locally. The workflow is generated auto-

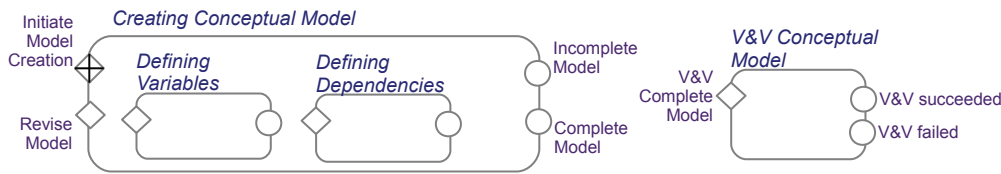


Figure 6: The stages of the artifact Conceptual Model

matically. Therefore the constraints are interpreted, e.g., by a forward planning algorithms. Clearly artifacts with their constraints can easily be added, deleted, or modified. It is up to the engine to evaluate this description and to derive a sound workflow.

In the following, we develop an artifact-based workflow meta-model for describing the modeling process (Section 2.3). To describe the artifacts the notation of Guard, Stage and Milestone (GSM)[13] is used. The life-cycle of each artifact comprises *stages*. A *guard* (represented as diamond) is attached to a stage (represented as rounded rectangle) and is a condition under which the activation of the associated stage is allowed. Stages themselves can be nested and stages on the same level of nesting can be activated in parallel (as long as their guards allow it). *Milestones* (represented as circles) are used to summarize the result of a stage. Another important aspect of the GSM notion is a sentry. A *sentry*, is used as a “global” guard, which is able to control when stages can be activated and milestones can be achieved. In addition they can directly manipulate the milestones. Sentries can be triggered internally or externally, due to changes within one or more artifact instances.

4.1 Artifacts, Guards, Stages and Milestones

The meta-model of the modeling process includes three artifacts, the CONCEPTUAL MODEL, the FORMAL MODEL, and the DATA.

Conceptual Model The CONCEPTUAL MODEL artifact is responsible for managing instances of conceptual models during their creation, change, Verification and Validation (V&V). As shown in Figure 6, it comprises different stages, namely a creation and a V&V stage. The creation stage has multiple sub-stages, *Defining Variables* and *Defining Dependencies*. On creation the artifact automatically enters the *Creating Conceptual Model* stage, which either ends in the milestone *Incomplete Model* or *Complete Model*. Additionally, the stage also allows revising a model. A complete model is a model suitable for V&V, i.e., all variables used in dependencies have also been defined. Thus, a complete conceptual model does by no means imply that the model is complete with respect to the question of interest and the system under study.

The V&V stage has two different milestones, namely V&V successful or not successful (failed). Its guard requests a complete conceptual model before this stage can be activated.

Formal Model The process of creating, verifying, and validating a formal model is captured in the FORMAL

MODEL artifact shown in Figure 7. The artifact consists of two top-level stages, *Creating Formal Model* and *V&V Formal Model*. *Creating Formal Model* comprises two sub-stages, i.e., *Selecting Formalism* and *Building Model*. *Creating Formal Model* is also the stage that is activated when this artifact is created, activating the *Selecting Formalism* as well. The guard of *Building Model* ensures that a formalism must be selected before this stage can be activated. Similar to the CONCEPTUAL MODEL artifact the stage representing the creation of a formal model has two possible milestones, *Complete* and *Incomplete* model. Analogously to the conceptual model, a model is named “complete” when it is suitable for V&V, i.e., no syntactic errors exist. Again, the V&V stage has two different milestones, namely V&V successful or not successful (failed) and its guard requests a complete formal model before this stage can be activated.

Data The DATA artifact, shown in Figure 8, can refer to rather heterogeneous information sources. A non-exhausting list of such information is:

- specific literature,
- model (conceptual or formal) either self created, retrieved from literature or model repositories, e.g., the BioModels Database[16],
- experiment description either retrieved from literature, self created or from experiment repositories, e.g., the MyExperiment Project[26],
- experiment data collected in wet-lab or simulation experiment,
- notes and media such as images and video

The artifact itself comprises three stages. The *Collecting Data*, *Annotating Data* and *Analyzing/Preparing Data* stage. On initialization of the DATA artifact, it automatically enters the collecting stage. This stage is meant for retrieving and collecting data, by e.g., reading a paper, querying a model repository, extracting data from literature, or measuring data. Once the data is collected it can be annotated with meta-data (e.g., type) and/or further processed. The data is analyzed and possibly prepared for further use by other stages of other artifacts. For instance wet-lab data can be used in the V&V stage of the FORMAL MODEL artifact, depending on the V&V method a specific preparation of the data might be required. Annotating, Analyzing and Preparing can take place in parallel and iteratively.

4.2 Artifacts and their interaction

In the previous section the artifacts and their stages were presented. Artifacts influence and interact with each other. Figure 9 illustrates some of that interaction. Firstly, brown

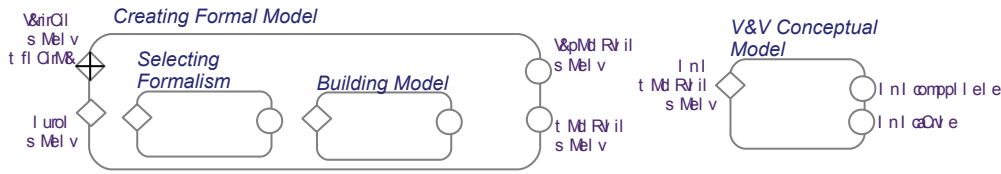


Figure 7: The stages of the artifact Formal Model

arrows indicate a local interaction between stages of one artifact. Secondly, green arrows indicate an interaction between stages of different artifacts, here the influence of the CONCEPTUAL MODEL artifact on the FORMAL MODEL artifact. Thirdly, blue arrows indicate the possible use of an artifact having reached a specific milestone by other artifacts, here Prepared Data as input to the V&V stages of the CONCEPTUAL MODEL and FORMAL MODEL artifact.

Firstly, we want to discuss the local interactions of stages (guards and milestones) (brown arrows). In the CONCEPTUAL MODEL artifact the Creating Conceptual Model stage is activated on initialization/creation. It allows activities that are able to define a conceptual model, e.g., interactively via a graph based editor. This might activate the two sub-stages and will lead to an incomplete or complete model. As long as the Incomplete Model milestone is reached the V&V stage can not be activated as its guard does not permit this. The only option for the artifact is to reenter the Creating Conceptual Model stage via the Revise Model guard. Once the Complete Model milestone is reached the V&V stage can be activated. Multiple activities might be available for the V&V stage, depending on the model and additional DATA artifacts. Each of those activities can be executed on the completed model (possibly in parallel). The stage will then lead to either a successful or failed milestone. Whereby the failed milestone will lead to triggering the Incomplete Model milestone of the previous stage which in return only leaves the option of model revision. This dependency between different milestones is defined using sentries.

The Creating Formal Model stage of the FORMAL MODEL artifact is activated on initialization/creation. Additionally, the Building Model stage has a guard that allows activation only if Selecting Formalism was finished before, which leaves it as the only stage available in the beginning. Once this stage has been completed, the model can be built. The Building Model stage can be activated e.g., by an editor for the selected formalism. If the Building Model stage is finished an Incomplete or Complete Model milestone is reached. As long as the Incomplete Model milestone is reached the V&V stage can not be activated as its guard does not permit this. The only option for the artifact is to reenter the Creating Formal Model stage via the Revise Model guard. Once the Complete Model milestone is reached the V&V stage is able to be activated and can produce a successful or failed V&V milestone. In case of a failed V&V milestone the Complete Model milestone will be set to Incomplete Model.

Additionally to these local artifact interactions, global interactions between artifacts exist (green arrows). For instance,

the FORMAL MODEL artifact is influenced substantially by the CONCEPTUAL MODEL artifact. If the conceptual model is revised (so has been newly completed), also the formal model should be revised, thus the state of the formal model is set to incomplete. Similarly, reaching a failed V&V milestone in the CONCEPTUAL MODEL artifact will automatically set the FORMAL MODEL artifact to incomplete. This also means in return, that the V&V Formal Model stage of the FORMAL MODEL artifact depends on the successful finishing of the V&V stage of the CONCEPTUAL MODEL before it can be activated.

As already mentioned blue arrows indicate the use of a specific artifact by another. In Figure 1 this can happen for the DATA artifact, which is or can possibly be used, e.g., by the V&V stages of the FORMAL MODEL and CONCEPTUAL MODEL artifacts. For instance a DATA artifact represents some data gathered in a wet-lab experiment conducted in parallel to the creation of the conceptual and formal model. This artifact could now be used by the FORMAL MODEL artifact in its V&V stage as comparison data set, when validating the model. However, this is only one possibility of using a DATA artifact, e.g., since such an artifact can represent a formal model described in literature, it could be used as starting point for the building of an extended model captured by a FORMAL MODEL artifact.

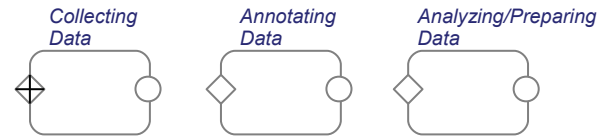


Figure 8: Stages of the artifact Data

4.3 Activities

The meta-model is described via the GSM-notion, without actual activities that take place during stages and lead to milestones. Activities drive the actual development of the model and their order of execution is controlled by the described meta-model. Activities can be distinguished between “local” and “global”. Local activities are only executed in the context of a specific artifact, while global activities affect more than one artifact. There can be any number of activities available during executing a modeling process. Each of which also has conditions under which it can be executed (sentries can be used here). Those conditions are:

- a specific stage is active or can be activated
- a specific artifact with a specific milestone is available

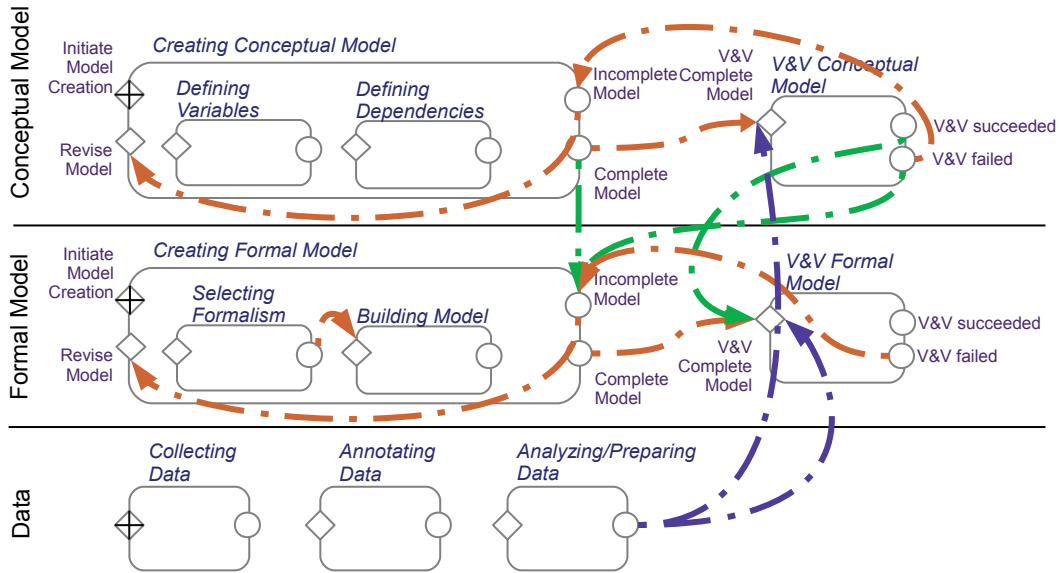


Figure 9: Inter-dependencies between artifacts (not all shown)

- a combination of the above (e.g., multiple stages in one or multiple artifacts are active and a specific artifact has reached a specific milestone)

An activity that is selected for execution (e.g., by the user or the system) can then activate a stage (if local) or multiple stages (if global). Furthermore it can once finished set a milestone (if local) or multiple milestones (if global). It can also trigger sentries (see above). Examples of activities are:

- a graph editor/mind map editor for building the conceptual model
- Editor for building the formal model
- V&V techniques as presented in[1]
- V&V techniques, e.g., statistical model checking, face-validation and cross-validation for the formal model
- analysis techniques for data preparation
- model retrieval techniques to access model repositories

4.4 Applicability to Case Study

Section 3.2 discussed some properties of the real world modeling process (Section 3) which are difficult to describe using an activity-based workflow description. The artifact-based approach simplifies this. For instance, the concurrency of activities in the modeling process combined with additional constraints is easily described by the proposed (nested) stages (which can be activated at the same time if on the same level), proper guards, and sentries. As shown, formal model, conceptual model and data are each modeled by their own artifact with their own independent life-cycle. This allows in principle for a concurrent independent progression of artifacts. To deal with constraints between artifacts, a simple sentry can be used, e.g., a failed validation of the conceptual model can automatically set the formal model to incomplete. So that the formal model needs to be revised before it can enter the V&V stage. Adding more constraints is simply achieved by defining more sentries or adapting guards of

stages.

The artifact-based approach allows an easier integration of new activities or functionality, as well. For instance, trying to add additional accreditation tests for each of the products of the modeling and simulation life cycle [23], a new artifact can be added, introducing an accreditation stage with two milestones (successful and failed accreditation) and a sentry that specifies that for the modeling process to complete, for all products the accreditation stage must have been completed successfully.

In contrast, the same changes would require a careful adaptation of the existing workflow structure in the activity-based approach.

To support the applicability Figure 10 illustrates how the modeling activities from the case study would map to artifacts. Here, a FORMAL MODEL, a CONCEPTUAL MODEL and four DATA artifacts are created, which are, one for literature, one for the model extracted from literature and one for the data generated by the self conducted wet-lab experiments. Additionally two snapshots of the artifacts during the modeling process are shown. Active, inactive and finished stages are colored green, white and blue respectively.

5. SUMMARY

We introduced an artifact based description of the modeling process in modeling and simulation and showed its suitability for the presented case study. In particular, its strength lie in its flexibility compared to the more traditional activity-based workflow approach. The declarative nature of artifacts allows to define constraints and leaves the consistency of the entire workflow up to the reasoning mechanism which becomes part of the workflow engine.

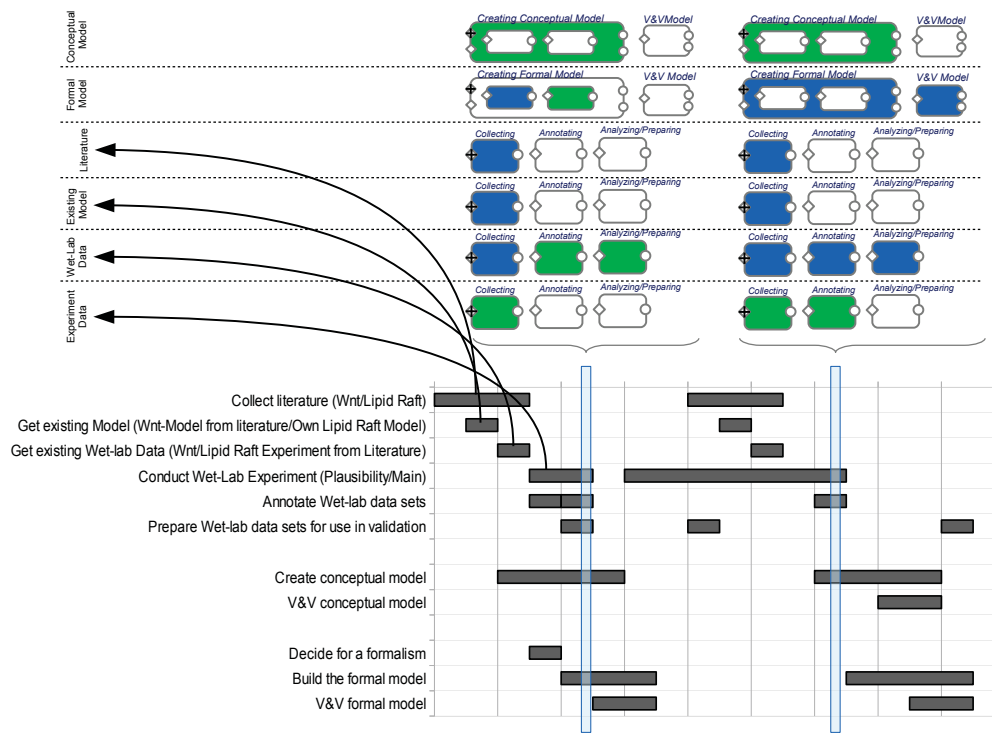


Figure 10: Application of artifact-based model to case study; top: states of the artifacts at two selected points in time (Active, inactive and finished stages are colored green, white and blue respectively); bottom: activities from case study over time

6. ACKNOWLEDGMENT

This work is funded by the DFG via the projects CoSA and VASSiB (SPP 1335).

7. REFERENCES

- [1] O. Balci. Verification, validation, and accreditation. In *Proceedings of the 30th conference on Winter simulation, WSC '98*, pages 41–4, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
- [2] K. Bhattacharya, N. S. Caswell, S. Kumaran, A. Nigam, and F. Y. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems Journal*, 46(4):703–721, 2007.
- [3] L. Chwif, J. Banks, J. P. de Moura Filho, and B. Santini. A framework for specifying a discrete-event simulation conceptual model. *J. Simulation*, 7(1):50–60, 2013.
- [4] D. Cohn and R. Hull. Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.*, 32(3):3–9, 2009.
- [5] P. Dadam and M. Reichert. The adept project: a decade of research and development for robust and flexible process support. *Computer Science - R&D*, 23(2):81–97, 2009.
- [6] R. Ewald, R. Schulz, and A. M. Uhrmacher. Selecting simulation algorithm portfolios by genetic algorithms. In *24th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation, PADS 2010, Atlanta, Georgia, USA, May 17-19, 2010*, pages 48–56. IEEE, 2010.
- [7] R. Ewald and A. M. Uhrmacher. Sessl: A domain-specific language for simulation experiments. *ACM Transactions on Modeling and Computer Simulation*, 2013 (to appear).
- [8] N. Gilbert and K. G. Troitzsch. *Simulation for the social scientist*. Open University Press, 1999.
- [9] K. Görlach, M. Sonntag, D. Karastoyanova, F. Leymann, and M. Reiter. Conventional Workflow Technology for Scientific Simulation. In X. Yang, L. Wang, and W. Jie, editors, *Guide to e-Science*, Computer Communications and Networks, pages 323–352–352. Springer London, London, 2011.
- [10] O. M. Group. *Business Process Model and Notation*, 2009.
- [11] C. Heavey and J. Ryan. Process modelling support for the conceptual modelling phase of a simulation project. In L. F. Perrone, B. Lawson, J. Liu, and F. P. Wieland, editors, *Winter Simulation Conference*, pages 801–808. ACM, 2006.
- [12] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(Web Server issue):729–732, 2006.
- [13] R. Hull, E. Damaggio, R. De Masellis, F. Fournier, M. Gupta, F. T. Heath, S. Hobson, M. H. Linehan, S. Maradugu, A. Nigam, et al. Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In *DEBS*,

- pages 51–62, 2011.
- [14] W. Kreutzer. *System simulation programming styles and languages*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.
- [15] A. D. Lattner. Towards automation of simulation studies - artificial intelligence methodologies for the control and analysis of simulation experiments. *KI*, 27(3):287–290, 2013.
- [16] N. Le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, et al. Biomodels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic acids research*, 34(suppl 1):D689–D691, 2006.
- [17] O. Mazumondet, M. John, C. Maus, A. M. Uhrmacher, and A. Rolfs. Integrating diverse reaction types into stochastic models: a signaling pathway case study in the imperative π -calculus. In *Winter Simulation Conference*, pages 932–943. Winter Simulation Conference, 2009.
- [18] R. Nance. The conical methodology: A framework for simulation model development. Technical Report SRC-87-002, Virginia Institute of Technology, 1986. accessed 25 October 2013.
- [19] A. Nigam and N. S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428–445, 2003.
- [20] K. Pawlikowski, H.-D. Jeong, and J.-S. Lee. On credibility of simulation studies of telecommunication networks. *Communications Magazine, IEEE*, 40(1):132–139, Jan 2002.
- [21] L. F. Perrone, C. S. Main, and B. C. Ward. Safe: simulation automation framework for experiments. In O. Rose and A. M. Uhrmacher, editors, *Winter Simulation Conference, WSC '12, Berlin, Germany, December 9-12, 2012*, page 249, 2012.
- [22] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst. Declare: Full support for loosely-structured processes. In *EDOC*, pages 287–300. IEEE Computer Society, 2007.
- [23] M. Rabe, S. Spieckermann, and S. Wenzel. Verification and validation activities within a new procedure model for v&v in production and logistics simulation. In A. Dunkin, R. G. Ingalls, E. Yücesan, M. D. Rossetti, R. Hill, and B. Johansson, editors, *Winter Simulation Conference*, pages 2509–2519. ACM, 2009.
- [24] M. Reiter, U. Breitenbücher, O. Kopp, and D. Karastoyanova. Quality of data driven simulation workflows. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–8, 2012.
- [25] S. Robinson. Choosing the right model: conceptual modeling for simulation. In S. Jain, R. R. C. Jr., J. Himmelspach, K. P. White, and M. C. Fu, editors, *Winter Simulation Conference*, pages 1428–1440. ACM, 2011.
- [26] D. D. Roure, C. Goble, J. Bhagat, D. Cruickshank, A. Goderis, D. Michaelides, and D. Newman. myExperiment: Defining the Social Virtual Research Environment. In *4th IEEE International Conference on e-Science*, pages 182–189. IEEE Press, 2008.
- [27] S. Rybacki, J. Himmelspach, F. Haack, and A. M. Uhrmacher. WorMS- A Framework to Support Workflows in M&S. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the 2011 Winter Simulation Conference*, 2011.
- [28] S. Rybacki, J. Himmelspach, E. Seib, and A. M. Uhrmacher. Using workflows in M&S software. In *Proceedings of the 2010 Winter Simulation Conference*, pages 535–545. IEEE, Dec. 2010.
- [29] S. Rybacki, S. Leye, J. Himmelspach, and A. Uhrmacher. Template and frame based experiment workflows in modeling and simulation software with worms. In *Services (SERVICES), 2012 IEEE Eighth World Congress on*, pages 25–32. IEEE, 2012.
- [30] R. G. Sargent. Verification and validation: some approaches and paradigms for verifying and validating simulation models. In *Winter Simulation Conference*, pages 106–114, 2001.
- [31] R. G. Sargent. Verification and validation of simulation models. *J. Simulation*, 7(1):12–24, 2013.
- [32] R. G. Sargent, R. E. Nance, C. M. Overstreet, S. Robinson, and J. Talbot. The simulation project life-cycle: models and realities. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 863–871. IEEE, 2006.
- [33] P. Savory and G. Mackulak. The science of simulation modeling. Technical Report 29, Industrial and Management Systems Engineering Faculty Publications, University of Nebraska, 1994.
- [34] R. Shannon. *Systems Simulation the Art and Science*. Prentice-Hall, Englewood Cliffs, N.J., 1975.
- [35] R. E. Shannon. Introduction to the art and science of simulation. In *Winter Simulation Conference*, pages 7–14, 1998.
- [36] M. Sonntag, D. Karastoyanova, and E. Deelman. BPEL4Pegasus: Combining Business and Scientific Workflows. In P. P. Maglio, M. Weske, J. Yang, and M. Fantinato, editors, *Service-Oriented Computing*, volume 6470 of *Lecture Notes in Computer Science*, pages 728–729–729. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [37] M. Sonntag, D. Karastoyanova, and E. Deelman. Bridging the Gap between Business and Scientific Workflows: Humans in the Loop of Scientific Workflows. In *Proceedings of the 6th IEEE International Conference on e-Science*, pages 206–213, 2010.
- [38] W. M. P. van der Aalst and K. M. van Hee. *Workflow Management: Models, Methods, and Systems*. Number 0262720469 in MIT Press Books. The MIT Press, 2004.
- [39] A. Weidemann, S. Richter, M. Stein, S. Sahle, R. Gauges, R. R. Gabdoulline, I. Surovtsova, N. Semmelrock, B. Besson, I. Rojas, R. C. Wade, and U. Kummer. Sycamore - a systems biology computational analysis and modeling research environment. *Bioinformatics*, 24(12):1463–1464, 2008.