FORMAL SPECIFICATION OF HYPOTHESES FOR ASSISTING COMPUTER SIMULATION STUDIES

Fabian Lorig
Colja A. Becker
Ingo J. Timm

Business Informatics I
Trier University
Behringstrasse 21, Trier, Germany
{lorigf,beckerc,itimm}@uni-trier.de

ABSTRACT

The aim of computer simulation studies is to answer research questions by means of experiments. For providing reliable evidence, the procedure of the study needs to be aligned with the question and the steps of the study need to be adjusted and combined accordingly. Supporting this process is challenging as the identification, customization, and combination of adequate tools and techniques for the systematic design of simulation experiments with respect to a hypothesis is not trivial. Hence, for providing computer-aided assistance, a language for the specification of hypotheses with respect to the credible and reproducible testing of research questions in computer simulation is needed. In this paper, we propose an approach for formally specifying hypotheses that allows for automated hypothesis testing. Based on specified hypotheses, we demonstrate the assistance of simulation studies in terms of model parametrization and analysis of results with respect to the statistically sound evaluation of hypotheses.

Keywords: Formal Specification, Hypothesis Testing, Assistance System, Parametrization.

1 INTRODUCTION

Computer simulation is a widely established means for planning, analyzing, and optimizing complex systems, e.g., in information system research (Hudert et al. 2010), sociology (Gilbert and Troitzsch 2005), and logistics (Berndt 2013). However, due to globalization, digitalization, and technological progress, the complexity of real world systems in business is increasing and so does the number of parameters influencing corresponding simulation models. In consequence, both, specialized domain knowledge for the model-building process as well as advanced simulation engineering skills for designing, executing, and evaluating experiments are required when applying computer simulation (Bley et al. 2000). To support simulation studies, a variety of frameworks and programming libraries has been developed for various scientific domains and disciplines (Kleijnen 2005, Nikolai and Madey 2009). On the one hand, creating simulation models as well as conducting simulation experiments is facilitated by these tools. But on the other hand, most tools do not cover the entire process of a study. Instead, they are limited to individual steps or a series of steps within the scope of a simulation study, e.g., estimation of necessary replications (Robinson 2005) or assessing the statistical significance of the results (Lee and Arditi 2006). For conducting computer simulation studies, this implies the necessity of a model-specific identification, customization, and combination of adequate and useful tools and techniques in a suitable way. Based on a model, it is in consequence a

SpringSim-TMS/DEVS 2017, April 23-26, Virginia Beach, VA, USA

non-trivial, extensive, and time-consuming task to provide reliable answers to specific research questions about the modeled system's behavior by means of computer simulation. This is dissatisfactory, considering that costly decisions as well as scientific theory-building processes are supported by and partially based on computer simulation (Diallo et al. 2013).

In other areas of research, e.g., medicine or aviation, standardized procedure models are used for ensuring the quality of the results of scientific studies. When designing new aircrafts, *RTCA DO-160* (a recommendation given by the RTCA (*Radio Technical Commission for Aeronautics*) which is used for the FAA's (*Federal Aviation Administration*) certification process) defines performance standards and environmental conditions that must be met when conducting tests. Equivalent to this, *clinical trials* are standardized and well-defined studies which aim at proving the efficiency of new medical treatments, e.g., drugs or vaccines, by performing experiments to test predefined research hypotheses (Simon 1989). In both cases, controlled conditions are given and the procedure itself is assisted in order to receive comparable and reproducible results.

Analogously, in computer simulation, a large number of procedure models can be found which give advice on conducting simulation studies, e.g., Law (2014) and Banks et al. (2013). Earlier research has shown that most procedure models address similar demands and describe similar steps. Yet, specific instructions on how to perform each step of the simulation study for answering the research question are rarely given and the methodological appropriateness is often disregarded (Timm and Lorig 2015). To overcome these shortcomings the integrated assistance of the entire life-cycle of a simulation study is proposed, i.e., the design, execution, and evaluation of simulation experiments (Teran-Somohano et al. 2014). For this purpose, existing approaches, scripts, services, and assistance functionalities for conducting simulation studies need to be logically linked with respect to existing procedure models.

As first step, most procedure models define the formulation of a research question or hypothesis that is to be tested during the simulation study. In terms of complex models with a large number of parameters, the identification and design of relevant experiments (i.e., the parametrization of the model) with respect to the research question (*hypothesis*) is challenging. However, the importance of a detailed characterization of the goals, hypothesis, and experiments of simulation studies and their interdependencies is emphasized (Yilmaz et al. 2016). Consequently, the first step towards an integrated assistance of simulation studies is to formally specify hypotheses for the systematic design of the study. By deriving the design of a study from its underlying hypothesis, conducting simulation studies becomes more reliable and targeted. This allows for an automated evaluation of hypotheses as well-established statistical hypothesis tests can be applied for answering research questions objectively and reproducibly based on simulation results. Furthermore, the documentation of the study is facilitated as the complete and unambiguous description of the experimenter's hypothesis is enabled.

The aim of this paper is the formal specification of hypotheses in computer simulation with respect to their automated evaluation. This allows for the systematic parametrization and evaluation of computer simulation studies. For achieving this, the characteristics and components of hypotheses need to be identified and formally described. Furthermore, an assistance needs to be developed that supports the transformation from research questions given in natural language to formally specified hypotheses that are machine-readable and testable by means of computer simulation.

The paper is structured as follows: First, foundations of the scientific use of hypothesis and statistical techniques for testing hypothesis are provided (Section 2). Following this, in Section 3, a review of the current state of the art is given. It focuses on the automation of science in terms of the generation, formalization, and testing of hypotheses and the assistance of simulation experiments. In Section 4, a concept for the formal specification of statistical hypotheses in simulation (*FITS*) is presented. Subsequently, in Section 5, the implementation of the concept as an assistance system is outlined followed by an evaluation. Finally, in Section 6, conclusions are drawn and an outlook on potential future work is provided.

2 FOUNDATIONS OF HYPOTHESES AND STATISTICAL HYPOTHESIS TESTING

In information systems research, the term hypothesis refers to an operationalized proposition that assumes relations between two or more variables (Recker 2013) which can be a speculative guess or a scientifically grounded assumption. The variables refer to the object of investigation. In this context, empirical sciences operate experimentally and formulate hypotheses based on observations (Degroot 1969). In statistics, a hypothesis is an assumption about the partially known probability distribution of one or more random variables which is examined in a test (Tietjen 2012). Statistical testing is not based on a population but on a sample according to the rules of *hypothesis tests* (Hanneman et al. 2012). It draws conclusions about the overall population from the sample. The test procedure is called *test of significance* which formally checks two mutually exclusive hypotheses: the null hypothesis and the alternative hypothesis (Freedman et al. 2007). To this end, a hypothesis is formulated accordingly such that any possible result of the examination is considered. Assuming that the null hypothesis holds, a test is then used to check whether the probability for the outcome of the examination to be a result of chance is sufficiently small to assume that the alternative hypothesis is valid instead. Commonly and depending on the discipline, a probability of ten, five or one percent (the *significance level*) is considered small enough to reject the null hypothesis in favor of the alternative hypothesis (Triola 2013). In this case, the stochastic occurrence of the result is considered so unlikely that the observation is not assumed to be a random event under the assumption that the null hypothesis holds.

There are different statistical tests and both determining an appropriate test as well as the application of the test need to be performed thoroughly. The characteristic, quantity, and level of data as well as the statistics used in the study, e.g, variance, can be used for selecting a statistical test (Bajpai 2009). There are tests with one sample, tests with several independent samples, and tests with two matched samples. Among these groups, different tests can be summarized. In other words, the application depends on various factors such as the content of the hypothesis or the assumption of a particular distribution in a population. Hypothesis testing with statistical tests can be differentiated into *parametric tests* and *nonparametric tests* (Kothari 2004). Hypotheses which can be tested with the latter include a statement about an entire distribution whereas hypotheses which can be tested with a parametric test either include a statement about a parameter of a distribution (e.g., mean) or a statement about one or two parameters of two distributions (e.g., comparison of two means). To reduce complexity, the following work will focus on hypotheses which can be tested with parametric tests: hypothesis testing of means, proportions, and variances as well as testing for comparing two means, proportions, or variances, and also testing for differences and testing of correlation coefficients.

3 STATE OF THE ART

Nowadays, data centric science (eScience) provides scientific workflows for assisting and automating recurring processes in research (Taylor et al. 2014). This includes the generation and verification of research hypotheses, e.g., in knowledge discovery in databases, where hypotheses are automatically generated from and verified with databases. In this context, Ganter and Kuznetsov (2000) proposed an approach for formalizing hypotheses based on formal concepts analysis. Also in natural sciences, methods for formalizing hypotheses are developed. They aim is the formal representation of knowledge for reuse and scientific discourse in the discipline. King et al. (2009) introduced Adam, a robot scientist, who automatically conducts biochemical experiments for testing autonomously created research hypotheses. The formal representation of the hypotheses is implemented by means of ontologies (Soldatova and King 2006). The benefits of formalizing hypotheses are discussed in social sciences (Van Zundert et al. 2012), too. However, the more complex structure of social-scientific hypotheses does not allow for applying existing techniques here.

Meta-DENDRAL is an example for an expert system for the aided formation of scientific hypotheses about chemical structures (Buchanan and Feigenbaum 1978). It is based on heuristic search algorithms and constraints which restrict the possibilities and its reusability. The framework developed by Tran et al. (2005)

aims at the formation of biochemical hypotheses, too, and makes use of reasoning on a knowledge-based system. The approach of Gonçalves and Porto (2015) enables hypothesis management and testing for large-scale simulation data. These approaches assist the process of hypothesis formation. However, hypotheses in this frameworks are too specific compared to the requirements from computer simulation and a straightforward application of statistical hypothesis tests is not possible.

Similar approaches for systematically assisting the research process can be found in computer simulation. Three decades ago, Ören et al. (1984) described the potential for assisting and automating monotonous tasks in simulation and stated that various activities which are associated with simulation can be supported by computer assistance. Nowadays, partial and specialized assistance is available for most steps of simulation studies, i.e., design, execution, and evaluation of experiments. During the design phase of simulation experiments, the choice and variation of input parameters as well as the number of replications need to be determined. For one thing, assistance can support the selection and preparation of input data (Bogon et al. 2012) and provide an assessment of the results' significance that can be expected from a concrete set of input data (Lattner et al. 2011). For another thing, the number of replications that are required for a given set of input data can be estimated (Hoad et al. 2010). Furthermore, assistance can make use of results of previous simulation runs for optimizing the simulation, e.g., by guiding the process of searching optimal values for the input parameters (Better et al. 2007). Scripts exist for assisting the execution of simulation experiments. Griffin et al. (2002) developed a set of *Perl* scripts which facilitate the process of conducting a large number of simulation experiments, focusing the systematic variation of inputs as well as the management of output data in communication network simulation. Similar collections of scripts are also available for other domains, e.g., flow simulation (Croucher 2011) or semiconductor manifacturing (Wagner et al. 2014), where domain specific ontologies are used. Additionally, in other types of simulation, e.g., stochastic simulation (Ewing et al. 1999) or Monte Carlo Simulation (Verburg et al. 2016), assistance exists. For evaluating the results of simulation runs, a number of assistance functionalities has been proposed, too. These include the automated analysis of simulation output data, e.g., for recommending the duration of the warm-up period or the run-length (Robinson 2005), and the automated statistical evaluation of the results, e.g., significance (Lee and Arditi 2006) or independence (Steiger and Wilson 2002). Also domain specific approaches, e.g., for agent-based simulation (Schruben and Singham 2011), exist.

Finally, there are frameworks for assisting simulation experiments that combine multiple assistance functionalities. Perrone et al. (2012) developed the SAFE (Simulation Automation Framework for Experiments) framework for defining, deploying, and controlling communication network simulation experiments. Another network simulation framework is STARS (Framework for Statistically Rigorous Simulation-Based Network Research) which, given an experiment, infers and conducts experiments that need to be run next (Millman et al. 2011). Also cross-domain frameworks exist, e.g., JAMES II (Himmelspach and Uhrmacher 2007), that provides a large amount of custom plug-ins for integrating individual features.

Concluding, it can be said that a wide range of assistance functionalities is available for simulation, yet, they do not integrate the full extent of simulation studies. Certainly the initial research questions which guides the entire study and which is often stated in natural language is not considered. Other disciplines that pursue hypothesis-driven research approaches developed formalisms for the integrated specification and assisted testing of hypotheses. Also in computer simulation, *experimental frames* that contain information on experimental conditions (Zeigler et al. 2000, Daum and Sargent 2001) and domain-specific languages for specifying, designing, and executing experiments were proposed (Teran-Somohano et al. 2015). However, a logical connection of assistance functionalities and formalisms with respect to a leading research hypothesis is not provided and the process of hypothesis formalization is not considered. Nonetheless, formalized hypotheses as they were proposed by King et al. (2009) are demanded in computer simulation for providing systematic assistance (Lattner et al. 2011). Consequently, a research gap can be identified such that an approach for the formal specification of research hypotheses in computer simulation is needed.

4 FORMAL SPECIFICATION OF HYPOTHESES AND USAGE IN AN ASSISTANCE SYSTEM

For the formal specification of hypotheses with regard to the assistance of the parametrization and evaluation of computer simulation studies, a theoretical concept of hypotheses in terms of computer simulation, i.e., attributes and components, needs to be developed first. After that, the interdependencies between the hypothesis and the simulation model need to be considered and defined.

A set V of (input) variables is defined by the simulation model. Some of them are constant values in a simulation run and others are values (with an initial assignment) which can be changed during a simulation run because of corresponding events in the simulation. Every single member of the set V is related to a value range which contains permissible values for an assignment of that particular variable (see (1)). Such a value range is a member of the power set of the real numbers. The mapping of a variable to its value range is given by the function ω. Since several variables may have the same value range, the function ω is injective.

$$V = \{v_1,\, \dots\,,\, v_u\} \qquad u = |V| \qquad \omega : V \to \mathscr{P}(\mathbb{R}) \tag{1}$$

An assignment of all variables in the simulation model can be described with the function ψ which relates every variable to an element r of its assigned value range (see (2)). The result of this parametrization is expressed with a tuple b. All possible assignments are summarized by the set B (see (3)).

$$\psi : V \to \mathbb{R},\ v \mapsto r \in \omega(v) \qquad b = (\ \psi(v_1)\,,\, \dots\,,\, \psi(v_u)\ ) \tag{2}$$

$$B = \ \omega(v_1)\ \times\ \dots\ \times\ \omega(v_u) \qquad b \in B \tag{3}$$

For the formal description of a hypothesis, which can be tested with a parameter test, a set Z of relation signs as well as two more sets are needed to build a statement about a respective factor (see (4) and (5)): On the one hand, a set S of parameters of a population respectively a variable which is calculated by the use of parameters of two populations. On the other hand, a set T of parameters and values which will be related with the members of the set S.

$$Z = \{<,\, >,\, \leq,\, \geq,\, =,\, \neq\} \qquad S = \{p,\, \mu,\, \sigma^2,\, \mu_1,\, p_1,\, \sigma_1^2,\, \delta,\, \rho\} \tag{4}$$

$$T = \{\mu_2,\, p_2,\, \sigma_2^2\} \cup \mathbb{R} \tag{5}$$

The assignment of the elements, which will be placed in relation to one another in the hypothesis, is expressed by the function γ (see (6)).

$$\gamma : S \to T \tag{6}$$

The associated feature or features of the hypothesis (the hypothesis contains a statement about this feature or these features) are related to the members of the sets S and T by the binary function τ. For this purpose, the set M includes all considered features (see (7)). In every simulation run, measured data will be collected for those features to generate sample data. Since some hypotheses contain statements about two parameters, each from a particular distribution of a feature, the set U is given for such a relation. Besides, the two parameters can be used to calculate a variable to make a statement about. Hence, the set U comprises all features and all possible pairs of these features.

$$M = \{m_1,\, m_2,\, \dots\,,\, m_w\} \qquad U = M \times M \cup M \qquad \tau : S \times T \to U \tag{7}$$

Now a hypothesis can be expressed as a tuple h which contains the following components (see (8)): an assignment of values to all variables of the simulation model, a statement about a parameter of a distribution of a considered feature or features and the feature itself respectively the features themselves.

$$h = (\ b,\, s,\, z,\, \gamma(s),\, \tau(s, \gamma(s))\ ) \qquad b \in B \quad s \in S \quad z \in Z \tag{8}$$

When considering a null hypothesis h_0 and an alternative hypothesis h_1, the component b must be equal in both tuples to refer a statement to the same parametrization of the simulation model. Likewise, the component s must be equal. The component z differs solely, so that the alternative hypothesis contains a contrary relation sign to the null hypothesis (e.g., "$>$" and "$\leq$"). Furthermore the specification of the significance level α and the sample size n is necessary (see (9)).

$$\alpha \in \mathbb{R}^{\geq 0} \qquad\qquad n \in \mathbb{N}^{+} \tag{9}$$

With regard to the representation of the content of a hypothesis, the question of the usage of a language arises. Here, a trade-off between the expressive power and the efficiency of the reasoning needs to be made. The natural language in writing (e.g., English) is capable of completely expressing a desired content. However, the natural language is too complex for a computer-aided processing. In order to formulate hypotheses about a system to be examined, we develop a language which can be used by researchers irrespectively of their subjects. Futher on, the language is used to conduct a parametrization of the simulation model and to decide which significance test is appropriate. For the name of the language the acronym *FITS* was chosen which stands for "**F**ormulating, **I**ntegrating and **T**esting of Hypotheses in Computer **S**imulation". Furthermore, the language can be used without a connected simulation system just to formulate hypotheses and to share them with other researches. Hereafter, a hypothesis expression in *FITS* is considered as a user input for a computer system which is connected to a simulation and integrates content of the hypothesis in this simulation by the use of parametrization. Additionally, the computer system is able to produce sample data by performing simulation runs and uses that data to execute a significance test for the entered hypothesis.

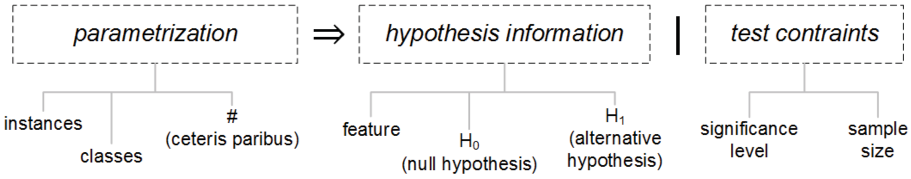


Figure 1: Structure of a hypothesis in *FITS*.

The structure of a *FITS* expression is shown in Figure 1. The expression comprises the user's hypothesis and the definition of related test properties for testing the hypothesis with the simulation model. It consists of three parts: *parametrization*, *hypothesis information*, and *test constraints*. The *parametrization* part includes the assignment (function ψ) of the variables which are described by the set V. If the simulation is connected to an ontology, where an ontology is an explicit specification of a conceptualization with at least some minimal common vocabulary (Schreiber 2008), this part will contain statements about classes or individual instances in the simulation model.

Since this part can be extensive due to a vast number of variables in the simulation model, it is possible to shorten that assignment by using the number sign ("#"). Here, the symbol has the meaning "ceteris paribus" and so an initial assignment is used for the remaining variables (alternatively a previously performed assignment could be used). After this part, a separator follows and the *hypothesis information* part which defines properties of the hypothesis. For the first separator here, an arrow is used, which is known for describing implications in logics. This is reminiscent of the formal structure of a conditional clause which a scientific hypothesis is (at least implicitly) based on.

The *hypothesis information* includes the components s, z and $\gamma(s)$ of both null and alternative hypothesis as well as the function τ with the denotation of treated features (members of the set M). Between the *hypothesis information* and the *test constraints* another separator is inserted. In the *test constraints* part, the user defines the significance level α and the sample size n for the subsequent execution of a significance test. Within the three mentioned parts the symbol "$\wedge$" is used to concatenate multiple subexpressions. The symbol "$\vee$" is used to separate the null hypothesis from the alternative hypothesis. We defined the syntax of the language more precisely with the aid of the Backus-Naur form. In the following, an example of a *FITS* expression is shown:

$$speed(a1,\, 4) \wedge position(a1,\, 61) \wedge rows(env,\, 180) \wedge breakTime(Agent,\, 20) \wedge \#$$
$$\Rightarrow \mu(operationTime) \wedge (H_0(\mu \geq 7) \vee H_1(\mu < 7)) \mid \alpha(5) \wedge n(100)$$

This expression refers to a simulation system with a connected ontology. Hence, in the first part, the members of the set V are not named by an identifier. Instead, the names of classes (starting with a capital letter) and instances (starting with a lowercase letter) of the simulation model are used in combination with a name of one of their attributes and a value for the assignment. The translation of the example in natural language reads as follows: *If agent "a1" moves with a speed level of 4 and starts at position number 61 and the environment contains 180 rows and all agents take a maximum break of 20 minutes ceteris paribus, then the measured operation time (feature) will be lower than 7 hours on average and this will be tested with a significance level of 5 percent and 100 simulation runs (sample size of 100).* The character "μ" denotes the expected value of a random variable which is the operation time in the example and also the feature of investigation. The subexpression of the null hypothesis H_0 and the alternative hypothesis H_1 contains the statement about that feature.

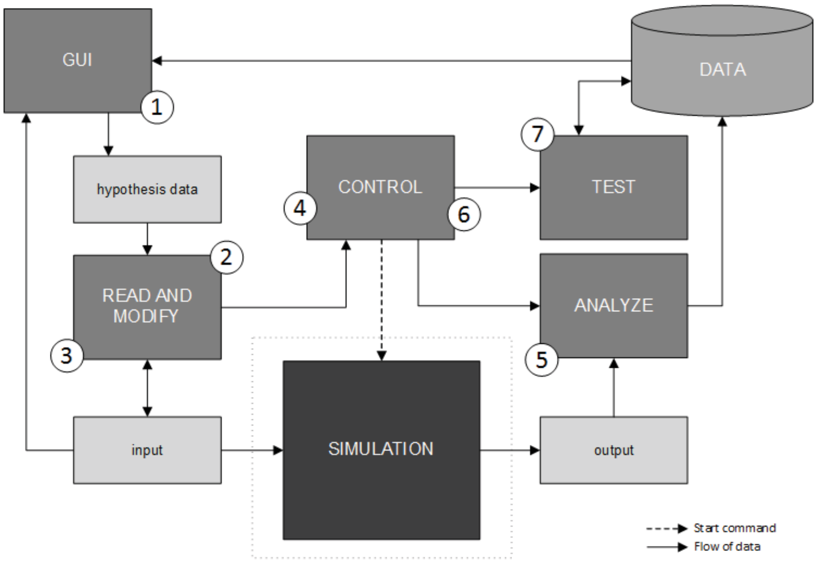
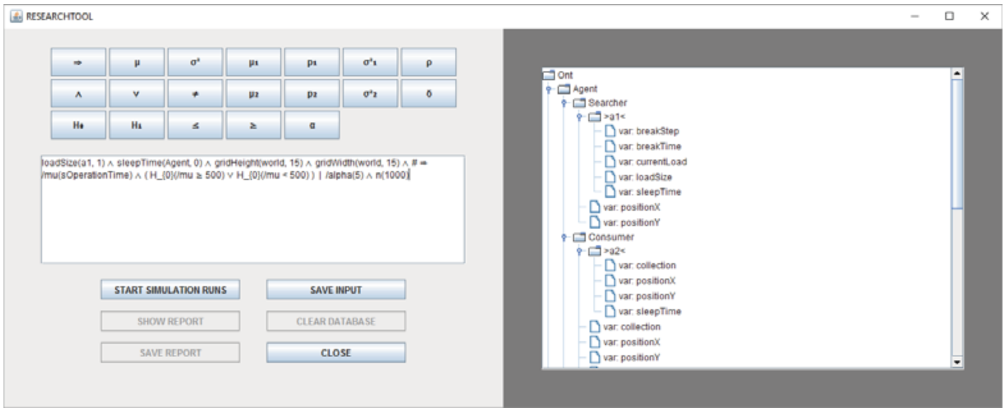


Figure 2: Concept of the research assistance system and its stepwise execution.

In Figure 2, the architecture of the developed computer system is shown. With aid of the graphical user interface (GUI), the user can build a hypothesis with predefined building blocks and an input assistance (step 1). If the simulation model is connected to an ontology, an interactive visualization of the ontology can be used to support the hypothesis building process. Otherwise, the user can select predefined variables of the simulation model from a list to insert them into the hypothesis. After the input process of the hypothesis, the computer system extracts the tuple b from the input (step 2), performs a new parametrization of the simulation model (step 3), and starts the simulation runs to accomplish the demanded sample size (step 4). The sample size n is read automatically from the last part of the hypothesis. The simulation runs are executed in an external replaceable simulation. The measurements of the declared feature or features (in the second part of the hypothesis) are realized by the assistance system and stored in a database. After every simulation run, output data is analyzed and the necessary data of the features are stored in the database (step 5) for the computations when testing the hypothesis. Following this, the appropriate significance test



```
loadSize(a1, 1) ∧ sleepTime(Agent, 0) ∧ gridHeight(world, 15) ∧ gridWidth(world, 15) ∧ # ⇒
/mu(sOperationTime) ∧ ( H_{0}(/mu ≥ 500) ∨ H_{0}(/mu < 500) ) | /alpha(5) ∧ n(1000)
```

Figure 3: Graphical user interface of the prototype.

is chosen (step 6) based on the user's input in the parts *hypothesis information* and *test constraints*. If more information is required for this decision (e.g., the distribution of a feature), the system starts a user dialog and asks simple questions. Finally, the result is calculated (step 7) and presented to the user with an additional report of all measured data and all partial results. In the following section, more details are described and a prototypical implementation is presented.

5 IMPLEMENTATION AND EVALUATION OF A PROTOTYPE

To evaluate the concept proposed in Section 4, we built a *Java* prototype of the research assistance system. In Figure 3, the user interface of the input process of the hypothesis is shown. On the left side of the pane, the user enters a hypothesis in the input field. Special characters can be inserted by using buttons. Otherwise the experienced user can use predefined character sequences to write off the whole hypothesis. For example, the character "μ" for the mean of a probability distribution of a feature can be inserted by using the corresponding button or by typing "/mu" in the input field. The picture shows further examples in the entered example hypothesis in the input field. On the right side of the pane, the user can navigate through the connected ontology of the external simulation model. By clicking on a node or a leave the user can insert templates for statements about the clicked class or instance attribute in the hypothesis input field. After entering a hypothesis the user hits the start button and the program reads the input and uses the first part of the hypothesis to parametrize the external simulation model. In order to make this possible the access and definition of all variables of the simulation model is given through an XML file and at the beginning of each simulation run the simulation has to assign all variables with the respective values in the file. Accordingly, the parametrization will be done by a modification of this external file by the research assistance system.

Thereupon, the program executes the simulation runs by starting the external simulation. Depending on the input hypothesis, the user has to answer some question panes one after another, e.g., if a normal distribution is assumed for the investigated feature and if the variance of that distribution is known. In the background, the program selects and executes the significance test using the provided information from the hypothesis input and the user dialog. The acceptance or rejection of the alternative hypothesis is communicated to the user by a message pane. As a last point, all results and data of the sample are displayed in a report which can be saved in a text file. The data for the sample is selected by the program out of all output data (after each simulation run) via the variable names which are used in the hypothesis to define the features of investigation. The interface is realized as an XML file, too. So at the end of a simulation run the simulation has to write all states of the variables to the file. The data of the sample is stored in a database and expanded by the program after each simulation run.

For evaluating the functionality of the prototype, we constructed a simulation with an established computer simulation framework independently from the research assistance system. We chose the multi-agent simulation environment *MASON* and created a simulation in the context of logistics. For the connection of the two systems we adapted the start command of the simulation in the assistance system and followed the use of the input and output interfaces.

```
=== REPORT ===
INPUT: loadSize(a1,1) ∧ sleepTime(Agent,0) ∧ # ⇒ μ(sOperationTime) ∧ ( H0(μ ≥ 500) ∨ H1(μ < 500) )) | α(5) ∧ n(10)
—— Variance of the distribution of the feature of investigation in the population is unknown
—— Feature of investigation is normally distributed in population
SIGNIFICANCE TEST: t−test
—— Model variable of the feature of investigation in simulation: sOperationTime
—— Arithmetic mean of measured feature of investigation: 485.1
—— Sampling variance: 2,352.544 (standard deviation: 48.503)
—— Decision rule: reject H0 for H1 in case v < −t_{1−alpha, n−1}
—— −t_{1−alpha, n−1} = −t_{0.95, n−1} = −1.833
—— v = (485.1 − 500.0) / (48.505 / 3.162) = −0.971
RESULT: The null hypothesis H0(μ ≥ 500) is NOT rejected in favor of the alternative hypothesis H1(μ < 500)

=== DATA ===
[...]
```

Figure 4: Extract from the report of the prototype.

We entered different hypotheses about the measured behavior of agents in the simulation model and analyzed the corresponding results. After checking the correctness of the model's parametrizations as well as the selection of the sample data, we performed the process manually without the usage of the research assistance system. Thereafter, we compared all auxiliary calculations and results. We used the generated reports of the research assistance system to get more detailed information for the comparison. In Figure 4, a snippet of a report as generated by the prototype is presented. The conclusion was satisfying: Both the manual process and the application of the prototype of the research assistance system came to the same results.

6 CONCLUSIONS

By formally specifying hypotheses, we aim at enabling the automated evaluation of hypotheses through the assistance of the parametrization of models and the evaluation of simulation results. We deem an integrated assistance for all steps of a simulation study is necessary as the complexity of simulation models as well as the amount of individual tools for supporting specific steps of simulation studies is increasing. Thus, the task of adequately identifying and combining existing tools for objectively and reproducibly testing research hypotheses by means of simulation becomes more challenging and time-consuming. The formal specification of hypotheses enables the assistance and automation of simulation studies as necessary steps are derived from the hypothesis, applied systematically, and executed experiments can be repeated easily.

FITS, the language proposed here, can be used for specifying hypotheses that contain statements regarding the behavior of a model or components of it. The assistance system uses the formalized hypothesis to execute necessary experiments, analyze the output by means of a suitable statistical hypothesis tests, and provide a statement whether or not the hypothesis holds based on the simulation results. With the developed GUI, the researcher is aided in the formal specification process as information on the variables of the model are given and an input assistance for creating hypothesis is provided.

The concept and the prototype presented in this paper constitute a first step towards the integrated assistance and replication of simulation studies. It is limited to the formal specification of hypotheses that are testable by a predefined set of parametric tests as well as the execution of simulation runs in order to meet the demanded sample size. In future work, the assistance provided by the system shall be further improved and elaborated. This includes the process of hypothesis generation, i.e., systematically deriving important factors, e.g., by means of automated sensitivity analysis, and relevant experiments from the hypothesis. Furthermore, it includes the *design*, i.e., intelligent limitation of the parameter space by the use of experimental designs, *execution*, i.e., distribution of runs as well as efficient utilization of existing hardware, and *evaluation*, i.e., optimization and documentation of the results, of simulation experiments. This will not only facilitate the statistically sound evaluation of hypotheses in simulation studies but also splitting the processes of model creation and simulation such that an expertise in model creation does not enforce detailed knowledge in simulation engineering and vice versa. Thus, an integrated assistance enables a wider public to conduct objective and reproducible simulation studies for answering model-related hypotheses.

REFERENCES

Bajpai, N. 2009. *Business Statistics*. Pearson.

Banks, J., J. Carson, B. Nelson, and D. Nicol. 2013. *Discrete-Event System Simulation*, Volume 5. Pearson.

Berndt, J. O. 2013. “Self-organizing Logistics Process Control: An Agent-based Approach”. In *Agents and Artificial Intelligence*, pp. 397–412. Springer.

Better, M., F. Glover, and M. Laguna. 2007. “Advances in analytics: Integrating dynamic data mining with simulation optimization”. *IBM Journal of Research and Development* vol. 51 (3.4), pp. 477–487.

Bley, H., C. Franke, C. Wuttke, and A. Gross. 2000. “Automation of simulation studies”. In *Proceedings of the 2nd CIRP International Seminar on Intelligent Computation in Manufacturing*, pp. 89–94. Citeseer.

Bogon, T., I. J. Timm, U. Jessen, M. Schmitz, S. Wenzel, A. D. Lattner, D. Paraskevopoulos, and S. Spieckermann. 2012. “Towards assisted input and output data analysis in manufacturing simulation: the EDASim approach”. In *Proceedings of the Winter Simulation Conference*, pp. 257. Winter Simulation Conference.

Buchanan, B. G., and E. A. Feigenbaum. 1978. “DENDRAL and Meta-DENDRAL: Their Applications Dimension.”. Technical report, DTIC Document.

Croucher, A. 2011. “PyTOUGH: A Python scripting library for automating TOUGH2 simulations”. In *Proceedings of the New Zealand Geothermal Workshop*, Volume 21, pp. 1–6.

Daum, T., and R. G. Sargent. 2001. “Experimental frames in a modern modeling and simulation system”. *Iie Transactions* vol. 33 (3), pp. 181–192.

Degroot, A. 1969. *Methodology: Foundations of Inference and Research in the Behavioral Sciences*. Psychological Studies. Walter de Gruyter GmbH & Company KG.

Diallo, S. Y., J. J. Padilla, I. Bozkurt, and A. Tolk. 2013. “Modeling and simulation as a theory building paradigm”. In *Ontology, Epistemology, and Teleology for M&S*, pp. 193–206. Springer.

Ewing, G., K. Pawlikowski, and D. McNickle. 1999. *Akaroa-2: Exploiting network computing by distributing stochastic simulation*. SCSI Press.

Freedman, D., R. Pisani, and R. Purves. 2007. *Statistics*. W.W. Norton & Company.

Ganter, B., and S. O. Kuznetsov. 2000. “Formalizing hypotheses with concepts”. In *International Conference on Conceptual Structures*, pp. 342–356. Springer.

Gilbert, N., and K. Troitzsch. 2005. *Simulation for the social scientist*. McGraw-Hill Education (UK).

Gonçalves, B., and F. Porto. 2015. “Managing scientific hypotheses as data with support for predictive analytics”. *Computing in Science & Engineering* vol. 17 (5), pp. 35–43.

Griffin, T. G., S. Petrovic, A. Poplawski, and B. Premore. 2002. *SOS: Scripts for Organizing ’Speriments*. SSF Research Network. http://www.ssfnet.org/sos/, accessed Jan. 2017.

Hanneman, R., A. Kposowa, and M. Riddle. 2012. *Basic Statistics for Social Research*. Wiley.

Himmelspach, J., and A. M. Uhrmacher. 2007. “Plug’n simulate”. In *40th Annual Simulation Symposium*, pp. 137–143. IEEE Computer Society.

Hoad, K., S. Robinson, and R. Davies. 2010. "Automated selection of the number of replications for a discrete-event simulation". *Journal of the Operational Research Society* vol. 61 (11), pp. 1632–1644.

Hudert, S., C. Niemann, and T. Eymann. 2010. "On computer simulation as a component in information systems research". In *Design Science Research in Information Systems*, pp. 167–179. Springer.

King, R. D., J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova et al. 2009. "The automation of science". *Science* vol. 324 (5923), pp. 85–89.

Kleijnen, J. P. 2005. "Supply chain simulation tools and techniques: a survey". *International Journal of Simulation and Process Modelling* vol. 1 (1-2), pp. 82–89.

Kothari, C. 2004. *Research Methodology: Methods and Techniques*. New Age International (P) Limited.

Lattner, A. D., T. Bogon, and I. J. Timm. 2011. "Convergence Classification and Replication Prediction for Simulation Studies". In *Int. Conf. Agents and Artificial Intelligence*, pp. 255–268. Springer.

Lattner, A. D., H. Pitsch, I. J. Timm, S. Spieckermann, and S. Wenzel. 2011. "AssistSim–Towards Automation of Simulation Studies in Logistics". *SNE* vol. 21 (3–4), pp. 119–128.

Law, A. M. 2014. *Simulation modeling and analysis*, Volume 5. McGraw-Hill New York.

Lee, D.-E., and D. Arditi. 2006. "Automated statistical analysis in stochastic project scheduling simulation". *Journal of Construction Engineering and Management* vol. 132 (3), pp. 268–277.

Millman, E., D. Arora, and S. W. Neville. 2011. "STARS: A framework for statistically rigorous simulation-based network research". In *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*, pp. 733–739. IEEE.

Nikolai, C., and G. Madey. 2009. "Tools of the trade: A survey of various agent based modeling platforms". *Journal of Artificial Societies and Social Simulation* vol. 12 (2), pp. 2.

Ören, T. I., B. P. Zeigler, and M. S. Elzas. 1984. *Simulation and Model-based Methodologies: An Integrative View*, Volume 10. NATO ASI Series F: Computer and System Sciences.

Perrone, L. F., C. S. Main, and B. C. Ward. 2012. "Safe: simulation automation framework for experiments". In *Proceedings of the Winter Simulation Conference*, pp. 249. Winter Simulation Conference.

Recker, J. 2013. *Information Systems Research as a Science*, pp. 11–21. Springer.

Robinson, S. 2005. "Automated analysis of simulation output data". In *Proceedings of the Winter Simulation Conference*, pp. 763–770. IEEE.

Schreiber, G. 2008. "Knowledge Engineering". In *Handbook of Knowledge Representation*, edited by F. van Harmelen, V. Lifschitz, and B. Porter, Volume 1 of *Found. of Art. Intelligence*, pp. 929–946. Elsevier.

Schruben, L., and D. Singham. 2011. "Agent based simulation output analysis". In *Proceedings of the Winter Simulation Conference*, pp. 540–548. IEEE.

Simon, R. 1989. "Optimal two-stage designs for phase II clinical trials". *Controlled clinical trials* vol. 10 (1), pp. 1–10.

Soldatova, L. N., and R. D. King. 2006. "An ontology of scientific experiments". *Journal of the Royal Society Interface* vol. 3 (11), pp. 795–803.

Steiger, N. M., and J. R. Wilson. 2002. "An improved batch means procedure for simulation output analysis". *Management Science* vol. 48 (12), pp. 1569–1586.

Taylor, I. J., E. Deelman, D. B. Gannon, and M. Shields. 2014. *Workflows for e-Science: scientific workflows for grids*. Springer Publishing Company, Incorporated.

Teran-Somohano, A., O. Dayıbaş, L. Yilmaz, and A. Smith. 2014. "Toward a model-driven engineering framework for reproducible simulation experiment lifecycle management". In *Proceedings of the Winter Simulation Conference*, pp. 2726–2737. IEEE.

Teran-Somohano, A., A. E. Smith, J. Ledet, L. Yilmaz, and H. Oğuztüzün. 2015. “A model-driven engineering approach to simulation experiment design and execution”. In *Proceedings of the Winter Simulation Conference*, pp. 2632–2643. IEEE.

Tietjen, G. 2012. *A Topical Dictionary of Statistics*. Springer US.

Timm, I. J., and F. Lorig. 2015. “A survey on methodological aspects of computer simulation as research technique”. In *Proceedings of the Winter Simulation Conference*, pp. 2704–2715. IEEE.

Tran, N., C. Baral, V. J. Nagaraj, and L. Joshi. 2005. “Knowledge-based integrative framework for hypothesis formation in biochemical networks”. In *Data Integration in the Life Sciences*, pp. 121–136. Springer.

Triola, M. 2013. *Elementary Statistics*. Pearson Education, Limited.

Van Zundert, J., S. Antonijevic, A. Beaulieu, K. van Dalen-Oskam, D. Zeldenrust, and T. L. Andrews. 2012. “Cultures of Formalisation: Towards an Encounter between Humanities and Computing”. In *Understanding digital humanities*, pp. 279–294. Springer.

Verburg, J. M., C. Grassberger, S. Dowdell, J. Schuemann, J. Seco, and H. Paganetti. 2016. “Automated Monte Carlo Simulation of Proton Therapy Treatment Plans”. *Technology in Cancer Research & Treatment* vol. 15 (6), pp. NP35–NP46.

Wagner, T., A. Gellrich, C. Schwenke, K. Kabitzsch, and G. Schneider. 2014. “Automated planning and creation of simulation experiments with a domain specific ontology for semiconductor manufacturing AMHS”. In *Proceedings of the Winter Simulation Conference*, pp. 2628–2639. IEEE Press.

Yilmaz, L., S. Chakladar, and K. Doud. 2016. “The Goal-Hypothesis-Experiment framework: A generative cognitive domain architecture for simulation experiment management”. In *Winter Simulation Conference (WSC), 2016*, pp. 1001–1012. IEEE.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press.

AUTHOR BIOGRAPHIES

FABIAN LORIG started studying Business Information Systems at Trier University in 2009. He received his Bachelor’s degree (B.Sc.) in 2012 and his Master’s degree (M.Sc.) in 2014, focussing on multi-agent systems and artificial intelligence. Since 2014 he is research assistant (PhD student) at the chair of Ingo J. Timm and works on an intelligent assistance for the design and execution of simulation experiments. His email address is lorigf@uni-trier.de.

COLJA A. BECKER studied Business Informatics at the University of Trier from 2010 to 2015. At the end of 2015, he received a Master’s degree. At the beginning of 2016 he started working at the chair of Ingo J. Timm as a research assistant and doctoral candidate. His work focus on agent technology as well as simulation technology. His email address is beckerc@uni-trier.de.

INGO J. TIMM received Diploma degree (1997), PhD (2004), and *venia legendi* (professoral thesis) (2006) in computer science from University of Bremen. From 1998 to 2006, Ingo has been PhD student, research assistant, visiting and senior researcher, and managing director at University of Bremen, Technical University Ilmenau, and Indiana University-Purdue University Indianapolis (IUPUI). In 2006, Ingo was appointed full professor for Information Systems and Simulation at Goethe-University Frankfurt. Since fall 2010, he holds a chair for Business Informatics at Trier University. In 2016 he founded and is now heading the Center for Informatics Research and Technology (CIRT) and its Research Lab on Simulation. He is author, co-author as well as editor of more than 100 scientific publications on simulation, information systems, knowledge-based systems for environmental protection, decision support in medicine, and theories and applications of multi-agent systems, especially in logistics. His email address is itimm@uni-trier.de.