CLEI ELECTRONIC JOURNAL, VOLUME 18, NUMBER 1, PAPER 4, APRIL 2015

Simulation Based Studies in Software Engineering: A Matter of Validity

Breno Bernard Nicolau de França, Guilherme Horta Travassos
Universidade Federal do Rio de Janeiro, COPPE,
Rio de Janeiro, Brasil,
{bfranca, ght}@cos.ufrj.br

Abstract

CONTEXT: Despite the possible lack of validity when compared with other science areas, Simulation-Based Studies (SBS) in Software Engineering (SE) have supported the achievement of some results in the field. However, as it happens with any other sort of experimental study, it is important to identify and deal with threats to validity aiming at increasing their strength and reinforcing results confidence. OBJECTIVE: To identify potential threats to SBS validity in SE and suggest ways to mitigate them. METHOD: To apply qualitative analysis in a dataset resulted from the aggregation of data from a *quasi*-systematic literature review combined with *ad-hoc* surveyed information regarding other science areas. RESULTS: The analysis of data extracted from 15 technical papers allowed the identification and classification of 28 different threats to validity concerned with SBS in SE according Cook and Campbell's categories. Besides, 12 verification and validation procedures applicable to SBS were also analyzed and organized due to their ability to detect these threats to validity. These results were used to make available an improved set of guidelines regarding the planning and reporting of SBS in SE. CONCLUSIONS: Simulation based studies add different threats to validity when compared with traditional studies. They are not well observed and therefore, it is not easy to identify and mitigate all of them without explicit guidance, as the one depicted in this paper.

Keywords: Simulation-based studies, simulation models, threats to validity.

1 Introduction

Simulation-Based Studies (SBS) consist of a series of activities aiming at observing a phenomenon instrumented by a simulation model. Thomke [1] reported the adoption of this sort of study as an alternative strategy to support experimentation in different areas, such as automotive industry and drugs development. Criminology is another field where researches have taken place with the support of SBS [2].

In the direction of these potential benefits, Software Engineering (SE) community has also presented some initiatives in using SBS to support investigation in the field. Indeed, apart from some interesting results, the SBS presented in the context of SE [3] allowed us to observe its initial maturity stage when compared with SBS concerned with the aforementioned areas. Lack of research protocols, *ad-hoc* experimental designs and output analysis, missing relevant information in the reports are some examples of observed issues into this context.

Based on the findings of our previous review [3] and on existing Empirical Software Engineering (ESE) guidelines for other investigation strategies, such as case studies and experiments, and simulation guidelines from other research areas, we proposed a preliminary set of guidelines aiming at providing guidance to researchers when reporting SBS into the SE context [4]. Later, we performed a first assessment of this set of reporting guidelines based on the approach presented in [5]. As a result, these guidelines have evolved to comprehend planning issues such as the problem, goal, context and scope definitions; model description and validation; experimental design and output analysis issues; the supporting environment and tools; and reporting issues such as background knowledge and related works, applicability of results, conclusions and future works.

One expected contribution of the guidelines' application is the identification of potential threats to validity that may bias a SBS in software engineering. The identification of threats and their mitigation from the initial problem and goals definition to the output analysis is one of guideline's concerns, reducing the risks of misinterpreted results. However, in order to perceive such benefits we believe it can be worth organizing a body of knowledge concerned with threats to validity already reported by the SE community when performing SBS. In addition, it is also important to depict the differences between common threats to validity (as those usually observed at *in vivo* and *in vitro* studies), and highlight those ones specifically identified at *in virtuo* and *in silico* studies. Therefore, we have conducted a secondary analysis of the data collected in [3] under the perspective of potential threats to validity found in SBS, which we are now presenting in this paper. As far as we are aware, there is no other work like this into the context of Experimental Software Engineering involving SBS. Such threats to validity compose the body of knowledge, organized as the new version of the proposed guidelines. Additionally, we have related these threats to Verification and Validation (V&V) procedures for simulation models previously identified in the technical literature in order to illustrate how to deal with such threats in SBS. Finally, we deliver some recommendations for using such body of knowledge when planning and reporting SBS, which also are going to compose a bigger set of guidelines (in progress).

The remaining sections of this methodological paper are organized as follows. Section 2 presents the background for our research. Section 3 presents the adopted research methodology. Section 4 presents the threats to validity identified through a qualitative analysis performed on a set of SBS, both in the SE technical literature and papers from other areas discussing this subject. Section 5 presents a list of technical V&V procedures applicable for simulation models. Section 6 presents the analysis on how the threats and the V&V procedure relate in order to provide more reliable SBS and deliver some recommendations in this sense. Finally, section 7 presents the final remarks and the way ahead.

2 Background

The work presented in this paper comprehends a broader effort in trying to organize a body of knowledge regarding SBS in the context of SE. Apart from earlier motivations and previous work on simulation models [6][7], we undertook a systematic literature review aiming at characterizing how different simulation approaches have been used in SE studies [3] following the guidelines proposed by [8] and adopting the PICO [9] strategy to structure the search string. In this review, population, intervention and outcome dimensions are considered to support the answer of the research question. The comparison dimension is not used because, as far as we are aware, there is no baseline to allow comparison. Therefore, this secondary study is classified as *quasi*-systematic literature review [34]. The search string had been calibrated by using nine control papers, previously identified through an ad-hoc review.

We searched for simulation-based studies in SE (population), using simulation models as instruments based on different simulation approaches (intervention). From them, we expected to obtain characteristics (outcome) from both the simulation models and studies in which they were used as instruments.

This way, we applied the search string in three digital search engines due the high coverage they usually offer: Scopus, EI Compendex and Web of Science. After applying the selection criteria by reading the titles and abstracts, we selected 108 studies including two other secondary studies regarding SBS. So, our inclusion criteria encompassed only papers available in the Web; written in English; discussing simulation-based studies; belonging to a Software Engineering domain; and mentioning one or more simulation models. Papers not meeting one of these criteria were excluded.

The information extracted from these research papers included the simulation approach, model purpose and characteristics, tool support, the Software Engineering domain, verification and validation procedures used to evaluate the simulation model, the study purpose and strategy (controlled experiment, case study, among others), output analysis procedure and instruments, and main results including applicability of the approach and accuracy of results. Such information has been organized with the *JabRef* tool [10].

After full papers reading, from the selected 108 relevant research papers, only 57 SBS were found, distributed over 43 research papers. The remaining papers rely on simulation model proposals. In other words, it was not possible to identify an investigation context, with a well-defined problem and research questions for them. A quality assessment took place and the main criteria regarded the existence or not of relevant information in the reports. The overall quality assessment indicated a poor quality of reports regarding SBS due the lack of relevant information such as research goals, study procedures and strategy, and the experimental design.

We identified a number of issues regarding reporting concerns, which led us to propose a set of reporting guidelines for simulation studies in SE [4]. Besides, we also observed issues regarding the methodological aspects involving the lack of (1) definition of research protocols for SBS, since aspects of research planning are usually overlooked when performing SBS; (2) proposals and application of V&V procedures for simulation models (Ahmed et al [11] also mention this topic in a survey involving modeling and simulation practitioners regarding software processes); (3) analysis and mitigation of threats to validity in SBS, which is strongly related to the validity of SE simulation models;(4) definition of criteria for quality assessment of SBS and the type of evidence we can acquire from it; (5) replication in simulation-based studies, given the absence of relevant information on studies reports.

Given these methodological issues and challenges, we moved forward SBS planning needs for research protocols by starting with the basics, like the context, problem, and research goals and questions definition. However, as we advanced, some issues about how to deal with the model validity and potential threats to validity in simulation experiments came up. So, it made clear to us the need to identify the main and recurrent threats to validity in simulation-based studies and to understand how such threats can be mitigated. For that, we primarily based our search in the outcomes from the *quasi*-systematic literature review. However, as previously observed in [3], the terminology is not consensual and authors in this field rarely discuss threats to validity using terms such as "threats to validity" or related ones. Thus, we decided to apply a systematic approach to handle the threats descriptions under the same perspective. For that, we adopted the qualitative procedures that are going to be described in the next section.

3 Research Methodology

During the execution of the *quasi*-systematic literature review [3], our interests were in characterizing simulation models and how SE researchers and/or practitioners use to organize, execute and report SBS. So, there was no focus on threats to validity at that moment.

Systematic Literature Reviews (SLR) appeared in Software Engineering in the early 2000s, inspired on the Evidence-Based paradigm [35]. Earlier works on this topic used to name all reviews performed with some systematic process as Systematic Reviews. However, many of them did not follow specific fundamental aspects or characteristics usually expected in systematic reviews, such as comparison among the outcomes w.r.t. their quality and possibilities of synthesis or aggregation. In this context, the term *quasi*-systematic review [34] appeared as a definition for reviews following SLR guidelines, but not covering at least one aspect, which is the case for our review. So, the "*quasi*" term stands for the unfeasibility of comparing outcomes due to lack of knowledge on the field or specific domain of investigation, also limiting the definition of quality profile for the available evidence, based on a hierarchy for evidence in Software Engineering.

After analyzing the content of 57 SBS, we proposed a preliminary set of reporting guidelines for SBS in Software Engineering, with the purpose of orientating researchers in such simulation studies activities [4]. In addition, we expect these guidelines can help researchers to identify (a priori) potential threats to the study validity. For that, in the current paper, we performed a secondary analysis over the 57 studies (distributed over 43 research papers), making use of some qualitative approach's procedures, namely the Constant Comparison Method [13] to identify common threats of validity across the studies. Additionally, we performed an additional *ad-hoc* review in order to identify whether other research areas outside SE have already discussed threats to validity in SBS, since we perceived the necessity for additional sources due to the terminology in SE simulation studies rarely refer to threats to validity using such terminology. In this opportunity, we identified and included in our analysis two research papers [2][26] discussing threats to simulation studies validity.

The Constant Comparison Method (CCM) [13] is represented by many procedures intercalating both the data collection and analysis to generate a theory emerging from such collected and analyzed data. It is important to note we have no ambition at this work in generating theories, but to use the analysis procedures from CCM to support the identification of threats to simulation studies validity.

Concepts are the basic unit of analysis in CCM. To identify concepts, the researcher needs to break down the data and to assign labels to it. The labels are constantly revisited in order to assure a consistent conceptualization. Such analytic process is called *coding*, and it appears in the method in three different types: *open coding*, *axial coding* and *selective coding*.

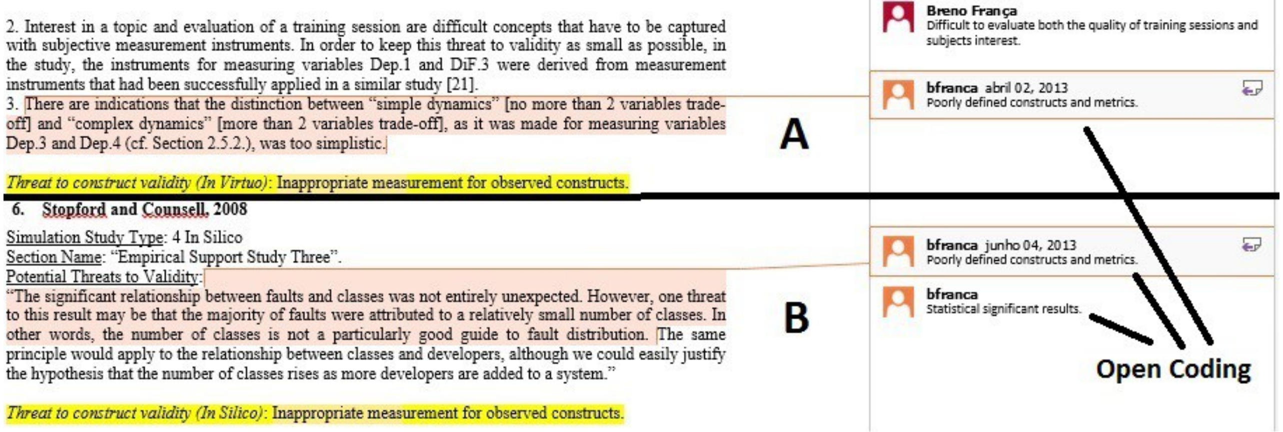
Open coding is the analytic process by which data is break down and conceptually labeled in codes. The codes may represent actions, events, properties, and so on. It makes the researcher to rethink about the collected data under different interpretations. In *open coding*, the concepts are constantly compared to each other to find similarities and then grouped together to form categories. On a higher level of abstraction, in *axial coding*, categories are associated to their subcategories and such relationships are tested against the collected data. This is also constantly done as new categories emerge. Finally, the *selective coding* consists in the unification of all categories around a central core category and other categories needing more explanation are filled with descriptive detail.

For data collection, it was necessary to define an additional information extraction form, containing the study environment, whether *in virtuo* or *in silico* [12], and the potential threats description (identified in the research papers as limitations, assumptions or threats to validity). The environment is important since *in virtuo* contexts are supposed to be risky, mainly by the involvement of human subjects.

This way, first we extracted the threats to validity descriptions and grouped them by paper. Only 13 out of 43 research papers contain relevant information regarding threats to validity. For the two additional research papers concerned with threats to simulation studies, we performed the data collection intercalated with the analysis of the ones obtained through the *quasi*-systematic literature review. Different from the SE studies, we observed a shared consistency between the terminology used in these papers and the current terminology as presented in [14], leading us to constantly review back the adopted SE terminology and search for discussions where it is possible to recognize threats to validity, limitations or assumptions.

After that, we performed an initial (open) coding, assigning concepts to chunks of the extracted text. So, for each new code, we compare to the other ones to understand whether it was or not about the same concept.

In Figure 1, we present the example of two threats descriptions (A and B).



2. Interest in a topic and evaluation of a training session are difficult concepts that have to be captured with subjective measurement instruments. In order to keep this threat to validity as small as possible, in the study, the instruments for measuring variables Dep.1 and DiF.3 were derived from measurement instruments that had been successfully applied in a similar study [21].
3. There are indications that the distinction between "simple dynamics" [no more than 2 variables trade-off] and "complex dynamics" [more than 2 variables trade-off], as it was made for measuring variables Dep.3 and Dep.4 (cf. Section 2.5.2.), was too simplistic.

Threat to construct validity (In Virtuo): Inappropriate measurement for observed constructs.

6. Stopford and Counsell, 2008

Simulation Study Type: 4 In Silico
Section Name: "Empirical Support Study Three".
Potential Threats to Validity:
"The significant relationship between faults and classes was not entirely unexpected. However, one threat to this result may be that the majority of faults were attributed to a relatively small number of classes. In other words, the number of classes is not a particularly good guide to fault distribution. The same principle would apply to the relationship between classes and developers, although we could easily justify the hypothesis that the number of classes rises as more developers are added to a system."

Threat to construct validity (In Silico): Inappropriate measurement for observed constructs.

Breno França
Difficult to evaluate both the quality of training sessions and subjects interest.

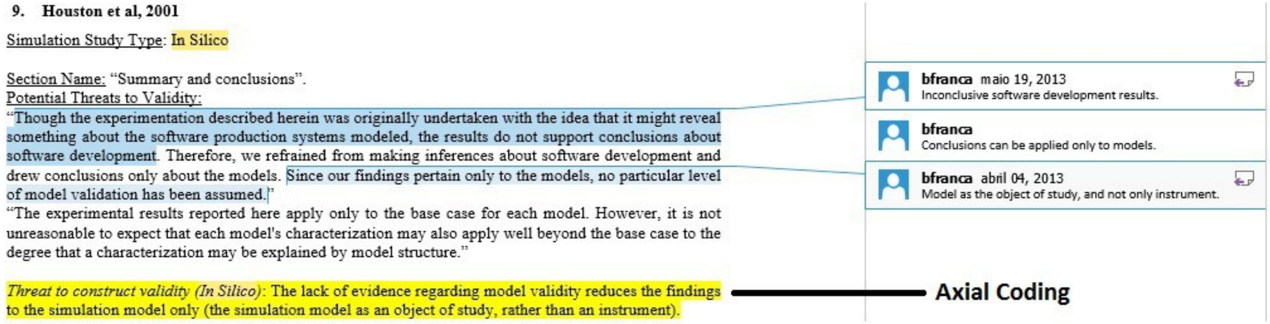
bfranca abril 02, 2013
Poorly defined constructs and metrics.

bfranca junho 04, 2013
Poorly defined constructs and metrics.

bfranca
Statistical significant results.

Figure 1. Open coding example, including repeated codes.

In the right side of Figure 1 are the codes assigned to chunks of text describing relevant aspects of the threats. For both descriptions, there is a common code assigned "Poorly defined constructs and metrics". This codes lead to a threat defined in the axial code (highlighted text bellow the text description). The main idea in this part of the analysis relates to the surrogate measures defined for the interested constructs not really representing the concepts under investigation.



9. Houston et al, 2001

Simulation Study Type: In Silico

Section Name: "Summary and conclusions".
Potential Threats to Validity:
"Though the experimentation described herein was originally undertaken with the idea that it might reveal something about the software production systems modeled, the results do not support conclusions about software development. Therefore, we refrained from making inferences about software development and drew conclusions only about the models. Since our findings pertain only to the models, no particular level of model validation has been assumed."
"The experimental results reported here apply only to the base case for each model. However, it is not unreasonable to expect that each model's characterization may also apply well beyond the base case to the degree that a characterization may be explained by model structure."

Threat to construct validity (In Silico): The lack of evidence regarding model validity reduces the findings to the simulation model only (the simulation model as an object of study, rather than an instrument).

bfranca maio 19, 2013
Inconclusive software development results.

bfranca
Conclusions can be applied only to models.

bfranca abril 04, 2013
Model as the object of study, and not only instrument.

Figure 2. Example of axial coding.

Furthermore, we reviewed the codes and then started to establish relationships among codes through reasoning about the threat description to generate the categories, which are the threats to validity. This way, each reasoning is written as a threat to validity, which the category represents the name of the threat in the next section. For instance, in Figure 2, we present an example of an emerged code from the interaction of three other codes.

For the threat presented in Figure 2, the inconclusive results for software development and the use of the model as object of study limit the results to the model by itself, not allowing extrapolating behaviors from the model to explain the real phenomena. It shows one of the implications of not having information regarding the model validity.

Finally, we grouped these open codes into four major categories (axial coding), namely conclusion, internal, construct and external validity, based on the classification for threats to experimental validity proposed by[14], but that could be extended in case we understand that it was needed. We did not perform selective coding, since the main goal was to identify and categorize the threats to validity.

This way, the main result of this secondary analysis is a list containing the potential threats to SBS validity, labeled using the grounded codes and organized according the classification proposed by Cook and Campbell, as presented in [14].

Additionally, we performed an analysis by matching threats to validity and V&V procedures for simulation models. The bases for the matching analysis are both the input and focus of each V&V procedure and threat to validity. The goal of such analysis is to identify whether the procedures can fully prevent from threats occurrences. Finally, deliver some recommendations on how to avoid them, all grounded on the findings of the systematic review and additional information collected from the literature on Simulation.

4 Threats to Simulation Studies Validity

The identified threats to validity are organized according to the classification presented in [14], in the following subsections. The title (in bold) for each threat to validity reflects the generated codes (categories) in the qualitative analysis. It is important to notice that we did not analyze threats of validity for each study, but only collected the reported ones. Indeed, it is possible to observe other potential threats to validity in each study, but we decided not to judge them based on the research paper only. For sake of avoiding repeating threats already discussed in others Experimental Software Engineering forums, we will concentrate on threats more related to *in virtuo* and *in silico* studies and not discussed on SE papers yet.

From the 28 identified threats to validity, we can distribute them into the subsets of conclusion validity (four), internal validity (ten), construct validity (ten) and external validity (four). We have not found nor were able to classify any threat to a different subset. The SE technical literature has already discussed most of the identified threats to validity regarding *in virtuo* studies, which strongly relates to the presence of human subjects "disturbing in some sense" the study. The expression "disturbing in some sense" concerns with the not controllable aspects of human behavior that we typically address in internal validity issues. On the other hand, threats to *in silico* experiments concentrate more on construct validity. This way, one may be tempted to point out this perspective as more critical. However, some threats can be more severe depending on the simulation goals.

4.1 Conclusion Validity

This validity refers to the statistical confirmation (significance) of a relationship between the treatment and the outcome, in order to draw correct conclusions about such relations. Threats to conclusion validity involve the use of inappropriate instruments and assumptions to perform the simulation output analysis, such as wrong statistical tests, number of required scenarios and runs, independence between factors, among others. For instance, stochastic simulations always deal with pseudo-random components representing uncertainty of elements or behaviors of the real world. Therefore, experimenters need to verify whether the model is able to reproduce such behavior across and within simulation scenarios due to the actual model configuration or caused by internal and natural variation. The main threats to conclusion validity identified in SBS are:

- **Considering only one observation when dealing with stochastic simulation, rather than central tendency and dispersion measures** [2]: different from the threats previously mentioned, we observed it into *in silico* context, where the whole experiment happens into the computer environment: the simulation model. It involves the use of a single run or measure to draw conclusions about a stochastic behavior. Given such nature, it has some intrinsic variation that may bias the results if not properly analyzed. We present an example of this threat from [2], where the authors say, "*If the simulation contains a stochastic process, then the outcome of each run is a single realization of a distribution of outcomes for one set of parameter values. Consequently, a single outcome could reflect the stochastic process, rather than the theoretical processes under study. To be sure that the outcome observed is due to the process, descriptive statistics are used to show the central tendency and dispersion of many runs*".

- **Not using statistics when comparing simulated to empirical distributions** [2]: also observed into the *in silico* context, this threat involves the use of inappropriate procedures for output analysis. It should be avoided comparing single values from simulated to empirical outcomes. It is recommended to use proper statistical tests or measures to compare distributions with a certain level of confidence.

We also observed other threats to conclusion validity at *in virtuo* environments, for instance, a small population sample hampering the application of statistical tests [16], which is similar to the one mentioned by Wohlin et al [14] as "Low statistical power". Besides, we identified the uneven outcome distribution (high variance) due to purely random subjects assignment [16-17], which is mentioned in [14] as "Random heterogeneity of subjects".

4.2 Internal Validity

This validity refers to the assurance that the treatment causes the outcome, rather than an uncontrolled external factor, i.e., avoid the indication of a false relationship between treatment and outcome when there is none. As the experimental setting in SBS often relies on different input parameters configurations, the uncontrolled factors may be unreliable supporting data, human subjects manipulating the model when performing *in virtuo* experiments or bias introduction by the simulation model itself. Events or situations that may impose threats in these inputs are to skip data collection procedures or to aggregate different context data, not giving an adequate training for subjects or lacking knowledge regarding the simulated phenomenon, and the lack of explanation for the phenomenon occurrence, respectively. Thus, the main internal validity identified threats in SBS are:

- **Inappropriate experimental design (missing factors) [16-19]:** apart from disturbing factors, the experimental design plays an important role on the definition of which variables (both *in virtuo* and *in silico* experiments) are relevant to answer the research questions. We observed this threat occurring only into *in virtuo* context, all of them from replications of the same research protocol, regarding to unexpected factors related to human subjects manipulating the simulation models. It is not common to miss factors on *in silico* studies, especially in SE simulations where models are mainly limited in number or input parameters. However, it is important to be caution when dropping out factors to simplify the experimental design, as in fractional factorial designs.

- **Simulation model simplifications (assumptions) forcing the desired outcomes [2,20,21,22,23,24]**: this is the most recurrent threat reported in the analyzed papers. Always identified into the *in silico* context, it is concerned with the simulation model itself. In this threat, the simulation model contains assumptions implemented in a way that they impact directly on the response variables. Or establishing the intended behavior or hypothesis as truth directly from the input to output variables, or giving no chance to alternative results to occur. For instance, in one of the six studies we observed this threat (reported as an assumption) the authors say

> "*In order to introduce the Test-First Development practice into the FLOSS simulation model, we make the following assumptions: (1) The average time needed to write a line of production code increases; (2) The number of defects injected during coding activities decreases; (3) The debugging time to fix a single defect decreases*".

In this case, it is possible to observe that the hypotheses (or beliefs) that Test-First Development productivity for coding decreases, the quality increases, and the maintenance time decreases are directly introduced in the model as assumptions. It goes in the wrong direction of SBS, where there is a theory with a defined mechanism that explains a phenomenon, i.e., how these interactions between variables occur. In such case, there is no room for simulation, since the outcomes are predictable without run the simulations. Such a black box (without mechanisms) approach is the typical situation where in vitro experiments are more applicable.

- **Different datasets (context) for model calibration and experimentation [25]:** it is difficult to realize how external or disturbing factors may influence a controlled computer environment (*in silico*). Nevertheless, the supporting dataset, often required by the simulation models, may disturb the results whether data from different contexts have been compared. This is the case when calibrating the simulation model with a specific dataset, reflecting the context of a particular project, product, or organization and using the same calibration to run experiments for another (different) context. For example, try to use cross-company data to simulate the behavior of a specific company.

We also observed other seven threats to internal validity, regarding *in virtuo* studies, similar to the ones already mentioned in [14]. It is the case of lack of SE knowledge hiding possible implications due to unknown disturbing factors [16-19], insufficient time to subjects' familiarization with the simulation tool and premature stage of the simulation tool (instrumentation effect) [16-19]. Also, non-random subjects' dropout after the treatment application (mortality) [16-19], different number of simulation scenarios (instruments) for each treatment [16-19] and available time to their performing [16-19], maturation effect by the application of same test both before and after treatments [16-19] and different level of expertise required by the instruments for both control and treatments groups (instrumentation effect) [16-19].

4.3 Construct Validity

This validity refers to the assurance that experimental setting (simulation model variables) correctly represents the theoretical concepts (constructs), mostly observed into the *in silico* context, where the simulation model plays the main role in the study. Threats to construct validity may occur due to the lack of model variables precision and relationships definition (and their respective equations), representing human properties, software products or processes, so the collected measures do not actually represent the desired characteristics. Davis et al [26] claim that the nature of simulation models tends to improve construct validity, since it requires formally defined constructs (and their measurement) and algorithmic representation logic for the theoretical mechanism, which explains the phenomenon under investigation. However, we could observe some threats to construct validity into the context of SBS, which are:

- **Naturally different treatments (unfair) comparison [16-19]**: this happens when comparing simulation models to any other kind of model not only in terms of their output variables, but also in nature, like analytic models. We observed this threat occurring only into *in virtuo* context, all of them from replications of the same research protocol.

- **Inappropriate application of simulation [16-19]**: in the *in virtuo* context, it is possible to identify situations where the model building can be more effective than the model usage, considering that SBS involves both parts. We observed this threat occurring only into *in virtuo* context, all of them from replications of the same research protocol.

- **Inappropriate cause-effect relationships definition [20]**: this threat is associated to the proper implementation of the causal relationships between simulation model constructs explaining the mechanism under study.

- **Inappropriate real-world representation by model parameters [20]**: the choice of input parameters should reflect real-world situations, assuming suitable values that can be observed in practice and are worthy for the analysis.
- **Inappropriate model calibration data and procedure [20]**: it involves, as the previous one, data used to perform the study, mainly to instantiate the simulation model, i.e., to calibrate the model using data from the corresponding real world. It may cause unrealistic distributions or equations, scaling the effects up or down.
- **Hidden underlying model assumptions [20]**: if assumptions are not explicit in model description, results may be misinterpreted or bias the conclusions, and may not be possible to judge at what extent they correspond to the actual phenomena.
- **Invalid assumptions regarding the model concepts [27]**: this threat regards to the validity of the assumptions made in the model development. Once they are invalid, the conclusions may also be corrupted. Every assumption made on a simulation model must be checked later, it is not an adequate "device" by which one can reduce model complexity or scope.
- **The simulation model does not capture the corresponding real world building blocks and elements [20]**: this threat concerns with model compliance with real world constructs and phenomenon representation. If there is no evidence of theoretical mechanism's face validity, it is possible that the simulation model has been producing right outcomes, through wrong explanations.
- **The lack of evidence regarding model validity reduces the findings only to the simulation model [28]**: This threat regards to simulation studies where a simulation model is chosen without proper information about its validity. Therefore, no conclusion can be draw about the phenomenon, but only about the model itself. Hence, the simulation model plays the role of an object of study, rather than an instrument. As an example, the authors in [28] say: "*Though the experimentation described herein was originally undertaken with the idea that it might reveal something about the software production systems modeled, the results do not support conclusions about software development* [inconclusive results]. *Therefore, we refrained from making inferences about software development and drew conclusions only about the models. Since our findings pertain only to the models, no particular level of model validation has been assumed* [lack of validity evidence]."

We can also identify inappropriate measurements for observed constructs in SBS [27]. Wohlin et al. [14] has already reported it as "inadequate preoperational explication of constructs", and it was the only threat observed in both *in virtuo* and *in silico* contexts.

4.4 External Validity

This validity involves the possibility of generalization of results outside the experimental settings scope. In simulation studies, it is particularly interesting to know if different simulation studies can reproduce similar results (called simulated external validity [2]) or it can predict real-world results (called empirical external validity [2]). For instance, a software process simulation model not being able to reproduce the results observed in one organization or not being able to obtain consistent results across different calibration datasets. Thus, the five identified (all concerned with the *in silico* context) threats to external validity are:

- **Simulation results are context-dependent, since there is a need for calibration [20]**: simulation modeling involves the definition of both conceptual and executable models. Therefore, to run simulations, the model needs to be calibrated using data representing the context in which the experimenter will draw conclusions. Results are as general as the supporting data. In other words, simulation results are only applicable to the specific organization, project, or product data.
- **Simulation may not be generalizable to other same phenomena simulations [2]**: this threat refers to the emulation of a theoretical mechanism across different simulations. Such simulations may differ in terms of calibration and input parameters, but the results are only generalizable if they appear in such different settings. In other words, the mechanism has to explain the phenomenon under different configurations to achieve such external validity.

- **Simulation results differ from the outcomes of empirical observations [2,20]**: when simulation outcomes sufficiently differ from empirical outcomes, we may say that simulated results have no external validity. One example of such threat in [20]: "First, the results are only partly consistent with empirical evidence about the effects of performing V&V activities. While code quality can always be improved by adding V&V activities, it is not always true that adding V&V activities in earlier development is better than adding them in later phases".
- **Simulation model not based on empirical evidence [26,29]**: if the model constructs and propositions are all conjectural, i.e., with no ground in field studies or empirical experiments, integrally or partially, it is very important to invest effort on validation procedures, since the model itself cannot show any external validity [26].

4.5 Lifecycle Perspective

A different perspective for the discussed threats to validity in simulation studies can be assumed according to the lifecycle of SBS. Such studies are often [30-31] organized in an iterative process (Figure 3) comprehending the phenomenon observation (or data collection), the simulation model (conceptual and executable) development and validation, model experimentation (planning and execution of simulation experiments) and output analysis. Other activities may appear in specific processes, but these are the traditional ones.

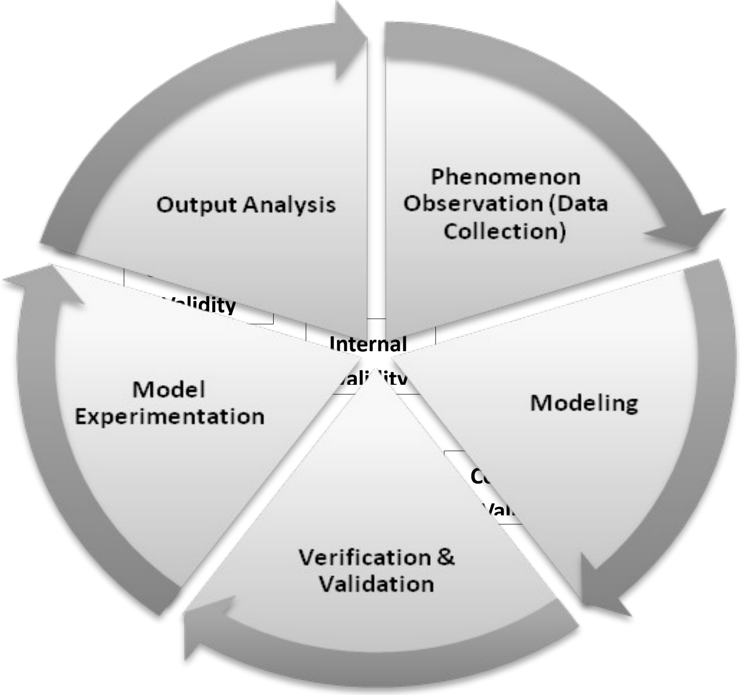


Figure 3. Simulation studies lifecycle and threats to validity.

For instance, consider a software process simulation model (SPSM) aiming at identifying process bottlenecks that can compromise the project schedule and containing pseudo-random variables to define the probability of success for a certain verification activity (review or test), The likelihood of success is based on an empirical distribution of historical effectiveness and efficiency records of the applied verification technique. In case of any verification activity succeed (i.e., identify defects on the verified artifact) there will be a correction effort to be added. An experimental design for the analysis of how verification effectiveness and efficiency impact on the project schedule will require more than one single simulation run for each scenario, in order to capture the internal variation of verification success rate, and the output analysis have to use proper statistical instruments to perform comparisons among scenarios considering the multiple runs.

In general, the effects of any threat are perceived in the output analysis stage. However, some of them can be identified in steps before.

Threats to conclusion validity tend to show up at downstream activities, more specifically on experimental design and output analysis (see Figure 3). The statistical expertise plays an important role for their occurrence, since design of the simulation experiment and the output analysis are strongly related [32]. For the previously mentioned case, a threat to conclusion validity can be the use of a single run for each of two scenarios (using high and low success rates). This choice will not allow the experimenter to determine which scenario performs better, since the results depend on the amount of variation in the empirical distribution defining the pseudo-random variables.

Planning of simulation experiments may also impose threats to internal validity. This type of threat has many possible causes, and it may be avoided or identified at all stages of the lifecycle (see Figure 3). Since data collected for calibration is one of the sources regarding threats to internal validity, the first stage of the simulation lifecycle should be performed systematically and with caution, allowing triangulation of data from different sources in order to assess data quality and validity. We observed threats to internal validity mainly into the *in virtuo* context, e.g., when human subjects pilot simulation models. Actually, all identified threats came from one single study protocol, which is replicated across different populations [16-19]. In this case, it is clear that the experimental design impacts negatively on the study results, since every threat to internal validity is identified in all replications. Besides, different parts of the design have contributed to this scenario, from the selection of a software process simulator and an analytical model (COCOMO) to represent levels of the learning instrument factor, which are not comparable from a learning perspective, to several instrumentation effects, for instance the premature stage of the simulation tool.

At *in silico* perspective, we can use our fictional SPSM to illustrate a potential threat to internal validity. Such threat regards the simulation model uses a historical dataset for its calibration, including the generation of the pseudo-random variables for verification effectiveness and efficiency, and consequently the generation of an executable model, but the simulation experiment uses data from a new project involving a new team with different background and expertise, for the input parameters. To compare such distinct contexts do not allow the determination of what is really causing the main effect, since the context may influence the outputs.

Threats to *construct* validity are all associated with the model development and validation activities (see Figure 3). In this phase, the conceptualization of constructs into model variables and propositions in terms of relationships between variables represents the translation of observations to a simulation language. Such translation should be carefully performed, using as much as possible domain experts to verify the lack of important real world variables. As an example, we expose the threat to *construct* validity regarding the inappropriate observed constructs measurement in the SPSM case. This threat concerns the alignment of the measurement program with the simulation model development. The model variables, such as the verification effectiveness and efficiency, need to be associated to metrics defined in the measurement plan for the software projects under investigation. It supports that every model variable and relationship can be tracked to the collected data, also avoiding attempts to incorrectly tie a different surrogate metric for a model variable, under the risk of biasing or hiding contextual information in the output analysis.

Apart from the lack of empirical evidence to support the simulation model development, threats to external validity are difficult to be identified before output analysis (see Figure 3). From the four threats we identified, three of them can be identified when analyzing the simulation output. On the other hand, if the model development is broken down into multiple iterations, the model developer can detect model increment that is inserting the unexpected behavior. For instance, if the fictional SPSM has been developed under multiple iterations, and in the second iteration the model does not replicate the reference behavior from the organization dataset, the second increment variables and relationships (or their equations) are assuming or implementing a wrong construct or relationship.

5 Verification and Validation of Simulation Models

Among possible approaches to avoid the occurrence of the threats to validity mentioned in previous section, we have the procedures adopted to verify and validate the simulation model and the experimental design. It is reflection of the nature of computer-based controlled environment, where the simulation model execution enables observing the phenomenon under investigation. This way, the only possible changes are in the input data or the simulation model. Consequently, the validity aspects concentrate on both the simulation model and data validities. For the scope of this paper, we are considering mainly the issues regarding the model validity affecting the study validity. In addition, it is relevant to mention that we made no analysis regarding the possible interaction among these threats to validity, in the sense that mitigating one threat may impose on the occurrence of others. However, we believe that threats related to model validity, specifically those that can be mitigated by the use of V&V procedures, do not present this sort of property, since these procedures when performed together increase the level of validity, having no impact in the results of applying any of them. Maybe other kind of threats, like the one caused by issues on the experimental design or supporting data may present side effects.

Since the SBS validity is highly affected by the simulation model validity, using a model that cannot be considered valid will bring invalid results, regardless the mitigation actions applied to deal with other possible validity threats. In other words, the simulation model itself represents the main threat to the study validity.

In [3], we identified nine verification and validation (V&V) procedures applied to simulation models in the context of SE, in 52 different research papers (included in Appendix A [3]). Besides, we merged these procedures with the ones existing in [15], which are twelve V&V procedures often performed for discrete-event simulation models in several domains. In fact, Sargent [15] presents fifteen procedures for V&V. However, we understand that three of them are useful instruments to perform verification and validation activities, rather than procedures or techniques. These three procedures regard the use of animations to graphically display the model behavior, operational graphics to present values for the model variables and outputs, and traces of the simulation runs to describe the whole variables changing in every cycle. This way, Table 1 presents the merge from the remaining thirteen procedures with the ones identified in the systematic literature review. The merge process was based on the reasoning about the procedures' descriptions, where some of them were grouped together.

The procedure "Comparison to Other Models" was identified in both the review and the list presented at [15]. Besides, we merged the software testing related procedures together in the procedure "Testing structure and model behavior", where we grouped "Degenerate Tests" and "Extreme Condition Tests", from [15].

Face validity is an expert-based evaluation approach. However, it does not have a systematic script or a set of steps. A review, an interview or even a survey may work in the same way, asking the expert about how reasonable that model and its outputs are.

Most of comparisons among simulated and actual data rely on historical or predictive validation. Sargent [15] also mentions a group called "Historical Methods", which is composed by three V&V approaches for simulation models: Rationalism; Empiricism, that "requires every assumption and outcome to be empirically validated"; and Positive Economics, that "requires that the model be able to predict the future, rather than concerned with model's assumptions or causal relationships (mechanism)".

We agree that Rationalism may contribute in V&V of simulation models. However, for the empiricism, it has a general description and seems to be just a characteristic or a type of verification, since it can be reworded as the Historical Validation or Predictive Validation procedures, for instance. It is also true for the Positive Economics, being a matter of perspective or abstraction. Finally, Sargent [15] also presents the "Multistage Validation" procedure that consists in performing the "Historical Methods", namely, Rationalism, Empiricism and Positive Economics sequentially.

As an example of application of such V&V procedures, Abdel-Hamid [21] submitted his model to several of them. The basis for developing his Software Project Integrated Model, using the System Dynamics (SD) approach, was field interviews with software project managers in five organizations, supplemented by an extensive database of empirical findings from the technical literature. Additionally, the author performed tests to verify the fit between the rate/level/feedback structure of the model and the essential characteristics of the real software projects dynamics. The project managers involved in the study confirmed this fit. However, the paper does not contain procedure descriptions for the tests and reviews. Besides, the results were not reported either. So, one may ask among other questions, "*What kinds of test were performed? How many discrepancies were identified by the project managers?*"

Another performed procedure is the comparison against reference behaviors. In this case, the author textually and graphically describes the behavior and presents the model representation using System Dynamics diagrams. The reference behavior in this case is the 90% syndrome, where developers use to miscalculate the required effort for a task and always underestimate it.

In addition, the simulation results in [21] were plotted in sequence run charts to compare against the expected behavior. Thus, the results seem to indicate the fit between the reference behavior and simulation results. Reference behaviors reproduced by the model included a diverse set of behavior patterns observed both in the organizations studied as well as reported in the literature.

The author also reports extreme condition simulations, i.e., to "test whether the model behaves reasonably under extreme conditions or extreme policies" [21].

Table 1: Verification and Validation Procedures for Simulation Models

Procedure	Description
Face Validity	Consists of getting feedback from knowledgeable individuals about the phenomenon of interest through reviews, interviews, or surveys, to evaluate whether the (conceptual) simulation model and its results (input-output relationships) are reasonable.
Comparison to Reference Behaviors	Compares the simulation output results against trends or expected results often reported in the technical literature. It is likely used when no comparable data is available.
Comparison to Other Models	Compares the results (outputs) of the simulation model being validated to results of other valid (simulation or analytic) model. Controlled experiments can be used to arrange such comparisons.
Event Validity	Compares the "events" of occurrences of the simulation model to those of the real phenomenon to determine if they are similar. This technique is applicable for event-driven models.
Historical Data Validation	If historical data exist, part of the data is used to build the model and the remaining data are used to compare the model behavior and the actual phenomenon. Such testing is conducted by driving the simulation model with either sample from distributions or traces, and it is likely used for measuring model accuracy.
Rationalism	Uses logic deductions from model assumptions to develop the correct (valid) model, by assuming that everyone knows whether the clearly stated underlying assumptions are true.
Predictive Validation	Uses the model to forecast the phenomenon's behavior, and then compares the phenomenon's behavior to the model's forecast to determine if they are the same. The phenomenon's data may come from the real phenomenon observation or be obtained by conducting experiments, e.g., field-tests for provoking its occurrence. Also, data from the technical literature may be used, when there is no complete data in hands. It is likely used to measure model accuracy.
Internal Validity	Several runs of a stochastic model are performed to determine the amount of (internal) stochastic variability. A large amount of variability (lack of consistency) may cause the model's results to be questionable, even if typical of the problem under investigation.
Sensitivity Analysis	Consists of changing the values of the input and internal parameters of a model to determine the effect upon the model output. The same relationships should occur in the model as in the real phenomenon. This technique can be used qualitatively— trends only — and quantitatively —both directions and (precise) magnitudes of outputs.
Testing structure and model behavior	Submits the simulation model to tests cases, evaluating its responses and traces. Both model structure and outputs should be reasonable for any combination of values of model inputs, including extreme and unlikely ones. Besides, the degeneracy of the model's behavior can be tested by appropriate selection of values of parameters.
Based on empirical evidence	Collects evidence from the technical literature (experimental studies reports) to develop the model's causal relationships (mechanisms).
Turing Tests	Individuals knowledgeable about the phenomenon are asked if they can distinguish between real and model outputs.

Additionally, the author conducted a case study at NASA. According to him, the DE-A project case study, which was conducted after the model was completely developed, forms an important element in validating model behavior as NASA was not part of the five organizations studied during model development. [21]

It is important to note, as also pointed out by the author, that one of these procedures alone may not provide enough validity for this model. However, taking them together can represent a solid group of positive results [21].

6 Recommendations for the improvement of Simulation Studies

Considering the V&V procedures mentioned in the previous section, now we relate them to the threats to validity identified in the context of SE simulation studies (section 4). The goal of such matching is (1) to provide explanation about how to avoid different bias imposed by the threats through performing specific V&V procedures and (2) to highlight the using of such procedures cannot avoid all the threats to simulation studies validity. From these explanations, we make some recommendations to guide researchers for SBS planning.

One general threat, not directly related to any specific recommendation given on this section, concerns the lack of evidence regarding the model validity that reduces the findings only to the simulation model. Obviously, one can avoid such threat by successfully applying a subset of the V&V procedures presented in Table 1. The main issue is that every attempt to validate a simulation model should be available to enable a proper output analysis from the experimenter perspective.

It is possible to divide the V&V procedures presented in the previous section into two perspectives: black and white box. The Face Validity procedure is the only one from Table 1 with a white box perspective. Such procedure enables the investigation of internal properties and behaviors of a simulation model, rather than dealing with it as a black box, in which just the combinations of input and output are evaluated. Usually, experts review the simulation model using their own knowledge using both conceptual (cause-effect diagrams, process descriptions, and other notations or languages) and executable models (calibrated models, simulation tools and outputs) to discuss their understandings with the model developers in terms of variables, relationships, and behaviors. Among the expected results, we can point out unrealistic model assumptions and simulation scenarios, misfit between concepts and measurements, unexpected output patterns and behaviors, and others. It can be worthwhile to perform this V&V procedure in two different moments: one still in the conceptual model development to avoid bias of desired results and when analyzing the matching between input and output values. Thus, threats to construct validity, involving the mechanisms explaining the phenomenon captured by the simulation model, are potentially identifiable by domain experts in advance. Examples of such threats are the failure on capturing the corresponding real world building blocks and elements, and inappropriate definition of cause-effect relationships.

Recommendation 1. *Make use of Face Validity procedures, involving domain experts, to assess the plausibility of both conceptual, executable models and simulation outcomes, using proper diagrams and statistical charts as instruments respectively.*

To ground the model propositions or causal relationships on empirical evidence can also help to mitigate the second threat, what sounds good to have at least one empirical evidence report regarding the embedded cause-effect relationships, showing some external validity [26].

Araújo *et al* [6] performed a set of systematic literature reviews aiming at reinforcing the validity of their SD model for observation of software evolution. In that opportunity, the reviews supported the identification of sixty reports of evidence for different relationships among the characteristics (e.g., eight reports of evidence for the relationship between characteristics Complexity and Maintainability) defined in their model.

Recommendation 2. *Try to support model (causal) relationships, as much as possible, with empirical evidence to reinforce their validity and draw conclusions that are more reliable.*

Using *Face Validity* in combination with *Sensitivity Analysis* can assist the proper selection of model's input parameters. Sensitive parameters should be made accurate prior to using the simulation model.

Recommendation 3. *Use results from Sensitivity Analysis to select valid parameters' settings when running simulation experiments, rather than model "fishing".*

In the same sense, Face Validity can be used along with the Rationalism to assess the model's assumptions regarding the underlying concepts. The concern with assumptions verification tends to make them explicit. However, when the model assumptions are hidden or not clearly stated, no Face Validity can be applied. For these cases, procedures like *Comparison to Reference Behaviors* and *Testing Structure and Model Behavior* are more suitable. The baseline or expected behaviors can give insights about the hidden model assumptions are affecting its results. Additionally, when using these two black box approaches, the design for the validation experiments need to involve the most sensitive parameters regarding the specific model assumptions. For instance, a SPSM assumes that requirements are always independent from each other (see [23] for a concrete example). In this case, validation experiments need to involve scenarios, and consequently input parameters, that enable the experts to observe whether outcomes are similar enough to expected behaviors in which they are confident about the dependency between requirements, so that they can accept such assumption.

The use of Rationalism to develop or verify the simulation model can be hampered due to lack of proved (or assumed as truth) assumptions, particularly in Software Engineering context. Thus, this procedure should be combined with empirical evidence, which is a similar approach to the "Historical Methods" mentioned by Sargent in [15].

In discrete-event models, it is usual to assume theoretical distributions to define how often events are dispatched. Such kind of assumption can be tested using the Event Validity procedure to verify if occurrences of the simulation model are similar to those of the real phenomenon, like events representing defect detection rates, requirements change requests, and others.

Recommendation 4. *Always verify model assumptions, so the results of simulated experiments can get more reliable.*

From the black box perspective, *Comparison to Reference Behaviors* can also help to identify situations where simulation results differ from the outcomes of empirical observations. However, for those cases in which there is a mismatch between the simulated and empirical outcomes, procedures like *Historical Data Validation* and *Predictive Validation* are more suitable, as long as enough data is available and both simulation output and empirical data share a common measurement context.

Black box approaches may also assist on data validity issues. Often, simulation models have a calibration procedure, and using it inappropriately may cause strange behaviors or invalid results. Turing tests may help with these situations, once these simulated results should resemble the actual ones. If a phenomenon expert cannot identify such a difference, the results have an acceptable degree of confidence. Another possibility is to use other models as a comparison baseline instead of experts, for that Comparison to Other Models. Specific for event-driven simulation, Event Validity procedure can help on improving data validity of input distributions or pseudo-random variables.

Recommendation 5. *When comparing actual and simulated results, be aware about data validity and that data under comparison came from the same or similar measurement contexts.*

For stochastic simulations, these models have their particularities and the main difference to be validated is the amount of internal variation on the outcomes. The threat of considering only one observation when dealing with stochastic simulation, rather than central tendency and dispersion measures can bias or blind the user or experimenter on the interpretation of results. The V&V procedure "*Internal Validity*" (the term adopted by Sargent [15] is overloaded with the Cook and Campbell as presented in [14] classification of threats to validity, but they have complete different meanings) helps on the understanding and measuring the amount of internal variation of stochastic models by running the model with the same input configuration and calculating both central and dispersion statistics. The results should be compared to real phenomenon observations to understand whether the both amounts of variation are proportional.

Performing one procedure or another can bring some validity to the study. The simulation models should be valid, based on evidence regarding its validity. It is important for not reducing the findings only to the simulations themselves.

Recommendation 6. *Make use of proper statistical tests and charts to analyze outcomes from several runs, compare to actual data and to quantify the amount of internal variation embedded in the (stochastic) simulation model, augmenting the precision of results.*

Once understood that V&V procedures may help to perform more confident simulation studies, it should also be pointed out that they are not silver bullets. We still can mention a series of threats that do not directly relate to such procedures, but to the adopted experimental design for the study and the output analysis procedures and instruments. For instance, threats regarding conclusion validity like considering only one observation when dealing with stochastic simulation and not using proper statistics when comparing simulated to empirical distributions (already considered in Recommendation 6). To mitigate threats like these, the experimenter needs a clear understanding of what to observe in the outcomes and the available statistical instruments to perform such analysis, since single values are neither able to capture the real trends and variance in stochastic simulations nor difference between actual data and simulations.

***Recommendation** 7. When designing the simulation experiment, consider as factors (and levels) not only the simulation model's input parameters, but also internal parameters, different sample datasets and versions of the simulation model, implementing alternative strategies to be evaluated.*

Additionally, threats to external validity like simulation results are context-dependent, since there is a need for calibration and the possibility of not generalizing the results to other simulations of the same phenomena are other examples of threats not handled by V&V procedures. In such cases, the experimental design should provide scenarios exploring different situations that one behavior is consistent across different contexts, through different datasets in which is possible to observe the phenomenon under investigation, and scenarios, using balanced combinations of factors and levels in the adopted experimental design.

Other threats still may not be mitigated using V&V procedures, but carefully planning the simulation experiments, tying the goals to research questions and to design and also verifying the feasibility of adopting simulation as alternative support for experimentation. That is what happens for threats such as: missing factors; different datasets (context) for model calibration and experimentation; naturally different treatments (unfair) comparison; and inappropriate use of simulation.

Table 2: Threats to validity associated to each recommendation

Rmd.	Threat to Validity	V&V Procedure
1	- Simulation model simplifications (assumptions) forcing the desired outcomes - The simulation model does not capture the corresponding real world building blocks and elements - Inappropriate cause-effect relationships definition	- Face Validity
2	- Inappropriate cause-effect relationships definition - Simulation model not based on empirical evidence	- Based on empirical evidence
3	- Inappropriate real-world representation by model parameters	- Face Validity - Sensitivity Analysis
4	- Hidden underlying model assumptions - Invalid assumptions regarding the model concepts	- Comparison to Reference Behaviors - Testing Structure and Model Behavior - Event Validity
5	- Inappropriate model calibration data and procedure - Simulation results differ from the outcomes of empirical observations	- Comparison to Reference Behaviors - Historical Data Validation - Predictive Validation - Turing Tests - Event Validity - Comparison to Other Models
6	- Considering only one observation when dealing with stochastic simulation, rather than central tendency and dispersion measures - Not using statistics when comparing simulated to empirical distributions	- Internal Validity
7	- Simulation results are context-dependent, since there is a need for calibration - Simulation may not be generalizable to other same phenomena simulations	N/A
Other Issues	- Inappropriate application of simulation - Inappropriate experimental design (missing factors) - Different datasets (context) for model calibration and experimentation - Naturally different treatments (unfair) comparison	N/A

At last, there is a recurrent threat to internal validity that is hard to identify: the simulation model simplifications (assumptions) forcing desired outcomes, rather than producing them based on the determination of proper scenarios and explained by a causal chain of events and actions. It configures a threat to internal validity since it reflects the model developer embedding the desired behavior into the simulation model, not allowing different results to occur by setting different scenarios. In other words, there is now way of assuring that the treatment (represented by the input parameters) is really causing the outcomes. It sounds like to know the answer for the research questions before running the simulation model and having no explanation, from the simulation results, for why that behavior was observed. From the viewpoint of simulation outputs compared to empirical observations, this one does not represent any threat. When all empirical and simulated and values are statistically similar, everything seems to be perfect. The problem lies on such limited black box view. The reason for reaching the desired output cannot be explained by a reasonable causal model or mechanism, but an explicitly generation from the input parameters to the output variables. So, one is not able of explain how to get such outcomes in real life, since there is no mechanism for a theoretical explanation. In summary, there is no way of making interventions to reproduce such behavior in real world, because the reasoning is missing and the result has probably occurred by chance. Comparison-based procedures cannot capture this type of threat. Just white box procedures like *Face Validity* involving simulation experts may help to identify such threat.

The adoption of all these recommendations has an impact in the effort and costs of the simulation study. It gets clear when realizing that the hours spent with domain experts in meetings for model reviews and also the effort demand by some data collection procedures, for example in Predictive Validation, and gathering evidence to reinforce causal relationships may be over. Thus, what is important is that both the goals for the simulation study and the expected benefits should drive the balance between the efforts for mitigating threats to validity and the risk of not doing so. Such benefits may be expressed in terms of meaningfulness (quality) of results and conclusions or amount the risks to take actions for implementing results.

Although discussed in this section, we present the association among threats to validity, recommendations on how to deal with them, and V&V procedures (Table 2). In addition, it is relevant to highlight that V&V procedures cannot mitigate four threats, since they are related to other planning issues such as simulation feasibility, data collection and experimental design definition.

7 Final Remarks

Taking simulation as a complementary research strategy for the evolution of Software Engineering knowledge, mainly in contexts where *in vivo* or *in vitro* experiments are unfeasible or risky, researchers should be aware about possible threats involved in this sort of study. The results reported on this paper advance the current state in ESE, by exposing such threats to SBS validity and matching them to V&V procedures for simulation models. Besides, seven recommendations, all of them grounded in technical literature acquired data, emerged for planning the tasks intending to reduce the possibility of occurrence of threats to validity.

We believe that the identification and compilation of such threats complemented by their discussion and analysis offers an evolved perspective that can contribute for the maturity of SBS, where most of time the main tasks have been performed *ad-hoc* due the lack of orientation, especially regarding model experimentation. Additionally, the possibility of detecting some of these threats by using V&V procedures; the understanding of how to avoid them; and presenting a set of recommendations configure an interesting contribution. As far as we are aware, there is no other work offering this sort of discussion in the experimental software engineering technical literature.

The organization of knowledge available in the technical literature regarding simulation studies in SE through secondary studies has directed our efforts. This organization involves synthesis and knowledge representation as guidelines for the planning and reporting of SBS, which is not a simple task.

As future directions, we are investigating how the *Design of Experiments* can contribute to improve the quality and confidence of simulation based studies in SE. Not only in the perspective presented by [28] and [33], but also as an enabler to explore more ambitious results than just anticipating *in vitro* and *in vivo* experiments.

Acknowledgements

Authors thank CNPq (Grants 141152/2010-9 and 304795/2010-0) for supporting this research.

References

[1] S. Thomke, *Experimentation Matters: Unlocking the Potential of New Technologies for Innovation*. Harvard Business School Press, Boston, 2003.

[2] J. E. Eck and L. Liu, "Contrasting simulated and empirical experiments in crime prevention," *J Exp Criminol*, vol. 4, pp. 195-213, 2008.

[3] B. B. N. de França and G. H. Travassos, "Are We Prepared for Simulation Based Studies in Software Engineering Yet?" *CLEI electronic journal*, vol. 16, no. 1, paper 8, Apr. 2013.

[4] B. B. N. de França and G. H. Travassos, "Reporting guidelines for simulation-based studies in software engineering," in *Proc. 16th International Conference on Evaluation & Assessment in Software Engineering*, pp. 156 – 160, Ciudad Real, Spain, 2012.

[5] B. A. Kitchenham, H. Al-Kilidar, M. Ali Babar, M. Berry, K. Cox, J. Keung, F. Kurniawati, M. Staples, H. Zhang, L. Zhu, "Evaluating guidelines for reporting empirical software engineering studies," *Empirical Software Engineering*, vol.13, no. 1, pp. 97-121. 2008.

[6] M. A. P. Araújo, V. F. Monteiro, G. H. Travassos, "Towards a model to support in silico studies of software evolution," in *Proc. of the ACM-IEEE international symposium on empirical software engineering and measurement*, pp. 281-290, New York, 2012.

[7] M. O. Barros, C. M. L. Werner, G. H. Travassos, "A system dynamics metamodel for software process modeling," *Software Process: Improvement and Practice*, vol. 7, no. 3-4, pp. 161-172, 2002.

[8] J. Biolchini, P. G.Mian, A. C. Natali, G. H. Travassos, "Systematic Review in Software Engineering: Relevance and Utility," PESC-COPPE/UFRJ, Brazil. Tech. Rep. http://www.cos.ufrj.br/uploadfiles/es67905.pdf . 2005.

[9] M. Pai, M. McCulloch, J. D. Gorman, "Systematic Reviews and meta-analyses: An illustrated, step-by-step guide," *The National Medical Journal of India*, vol. 17, n.2, 2004.

[10] JabRef reference manager, available at: http://jabref.sourceforge.net .

[11] R. Ahmed, T. Hall, P. Wernick, S. Robinson, M. Shah, "Software process simulation modelling: A survey of practice," *Journal of Simulation*, vol. 2, pp. 91 – 102, 2008.

[12] G. H. Travassos and M. O. Barros, "Contributions of *In Virtuo* and *In Silico* Experiments for the Future of Empirical Studies in Software Engineering", in *Proc. WSESE03*, Fraunhofer IRB Verlag, Rome, 2003.

[13] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications. 2007.

[14] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, A. Wesslen, *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers. 2000.

[15] R. G. Sargent, "Verification and Validation of Simulation Models," in *Proc. of the Winter Simulation Conference*, pp. 166 – 183, Baltimore. 2010.

[16] D. Pfahl, M. Klemm, G. Ruhe, "A CBT module with integrated simulation component for software project management education and training," *Journal of Systems and Software*, vol. 59, n. 3, pp. 283 – 298. 2001.

[17] D. Pfahl, O. Laitenberger, J. Dorsch, G. Ruhe, "An Externally Replicated Experiment for Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education," *Empirical Software Engineering*, vol. 8, n. 4, pp. 367 – 395. 2003.

[18] D. Rodríguez, M. Á. Sicilia, J. Cuadrado-Gallego, D. Pfahl, "e-learning in project management using simulation models: A case study based on the replication of an experiment," *IEEE Transactions on Education*, vol. 49, n. 4, pp. 451–463. 2006.

[19] D. Pfahl, O. Laitenberger, G. Ruhe, J. Dorsch, T. Krivobokova, "Evaluating the learning effectiveness of using simulations in software project management education: Results from a twice replicated experiment,". *Infor. and Software Technology*, vol. 46, n. 2, pp. 127–147. 2004.

[20] V. Garousi, K. Khosrovian, D. Pfahl, "A customizable pattern-based software process simulation model: Design, calibration and application," *Software Process Improvement and Practice*, vol. 14, n. 3, pp. 165 – 180. 2009.

[21] Abdel-Hamid, T, "Understanding the "90% syndrome" in software project management: A simulation-based case study," *Journal of Systems and Software*, vol. 8, pp. 319-330. 1988.

[22] T. Thelin, H. Petersson, P. Runeson, C. Wohlin, "Applying sampling to improve software inspections," *Journal of Systems and Software*, vol. 73, n. 2, pp. 257 – 269. 2004.

[23] M. Melis, I. Turnu, A. Cau, G. Concas, "Evaluating the impact of test-first programming and pair programming through software process simulation," *Software Process Improvement and Practice*, vol. 11, pp. 345 – 360. 2006.

[24] I. Turnu, M. Melis, A. Cau, A. Setzu, G. Concas, K. Mannaro, "Modeling and simulation of open source development using an agile practice," *Journal of Systems Architecture*, vol. 52, n. 11, pp. 610 – 618, 2006.

[25] F. Alvarez, G. A. Cristian, "Applying simulation to the design and performance evaluation of fault-tolerant systems," in *Proc. of the IEEE Symposium on Reliable Distributed Systems*, pp. 35–42, Durham, 1997.

[26] Davis, J. P.; Eisenhardt, K. M.; Bingham, C. B.: Developing Theory Through Simulation Methods. Academy of Management Review, vol. 32, n. 2, pp. 480-499. 2007.

[27] B. Stopford, S. Counsell, "A Framework for the Simulation of Structural Software Evolution," *ACM Transactions on Modeling and Computer Simulation*, vol. 18, 2008.

[28] D. X. Houston, S. Ferreira, J. S. Collofello, D. C. Montgomery, G. T. Mackulak, D. L. Shunk, "Behavioral characterization: Finding and using the influential factors in software process simulation models," *Journal of Systems and Software*, vol. 59, pp. 259-270, 2001.

[29] H. Rahmandad, D. M. Weiss, "Dynamics of concurrent software development," *System Dynamics Review*, vol. 25, n. 3, pp. 224–249, 2009.

[30] O. Balci, "Guidelines for successful simulation studies," in *Proc. Winter Simulation Conference*, pp. 25-32, 1990.

[31] C. Alexopoulos, "Statistical analysis of simulation output: State of the art," in *Proc. Winter Simulation Conference* , pp. 150 – 161, Dec. 2007. DOI = 10.1109/WSC.2007.4419597.

[32] J. P. C. Kleijnen, S. M. Sanchez, T. W. Lucas, T. M. Cioppa, "State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments," *INFORMS Journal on Computing*, vol. 17, n. 3, pp. 263-289, 2005. http://dx.doi.org/10.1287/ijoc.1050.0136.

[33] W. W. Wakeland, R. H. Martin, D. Raffo, "Using Design of Experiments, sensitivity analysis, and hybrid simulation to evaluate changes to a software development process: A case study," *Software Process Improvement and Practice*, vol. 9, pp. 107–119, 2004.

[34] G. H. Travassos, P. S. M. dos Santos, P. G. M. Neto, J. Biolchini, "An environment to support large scale experimentation in software engineering," in *Proc. of the 13th IEEE International Conference on Engineering of Complex Computer Systems*, pp. 193-202, Mar. 2008.

[35] B. A. Kitchenham, T. Dyba, M. Jørgensen, "Evidence-Based Software Engineering," in *Proc. 26th ICSE*, p.273-281, May, 2004.