Creating an Advanced Software Engineering Laboratory by Combining Empirical Studies with Process Simulation

Jürgen Münch, Dieter Rombach, Ioana Rus

Abstract—**Laboratory experiments have been proven as an essential means for determining the effects of software engineering technologies. We propose the use of process simulation for adding a virtual capability to a software engineering laboratory in order to create a new and advanced type of experimental laboratory. One major motivation for such an advanced software engineering laboratory is cost reduction by simulating human behavior and the environment of the local process to be examined. The article describes the benefits of such an advanced laboratory and discusses possible combinations of empirical studies and process simulation. Additionally, an ongoing project for implementing a platform for such a laboratory is introduced and open questions for future research are sketched.**

***Keywords* — Experimental software engineering, empirical studies, software process modeling, process simulation, software measurement.**

I. INTRODUCTION

The development of high quality software systems requires the use of engineering methods. These methods (e.g., design methods, inspection methods) are predominantly human-based. Their effects depend on characteristics of the persons involved, such as skills, motivation, experience, and individual preferences for problem solving. A systematic selection of appropriate methods during project planning requires knowledge about their effects in a specific project context.

The software domain can be characterized by a huge set of software development methods but little knowledge regarding their effects in practical environments. A trial and error approach is cost-, human, and time-intensive. This situation implies that the transfer of innovative methods is being hindered by a lack of knowledge about their strengths and weaknesses [10].

The effects of software development methods under varying human and technological conditions can be identified and modeled through the use of experiments [2][14]. The need for experiments in software engineering has been acknowledged for many years [11]. New technologies or changes in the process should be tested before they are implemented, in order to determine their impact in the specific context.

The idea of software engineering laboratories has been around since the 1970s [1]. Laboratories are understood as controlled environments for performing experiments in a specific domain. Software engineering laboratories are not too widely spread for many reasons, the main one being the cost and time involved in running experiments in software development, where most projects are struggling to stay within budget and delivery time. Similar to experiments in social sciences, experiments always require the inclusion of human subjects and are therefore very expensive.

We propose here to use process simulation for adding a virtual capability to a real laboratory, in order to create a new and advanced type of experimental laboratory. One major motivation for such an advanced software engineering laboratory is cost reduction by simulating human behavior and the process environment of the method to be examined. Additionally, such a laboratory allows for better demonstrating the effects of a method in an understandable way [12]. In particular, the multitude of variations of an experiment that is often necessary to cover different impact factors can be performed, and costs can be reduced enormously. Consequently, learning cycles can be shortened. A virtual lab would also significantly support the automation of data collection for experiments, which is a big overload in a real world setting.

Such an advanced lab could provide support to many different areas: technology evaluation, process improvement evaluation, decision-making in project management, controlling and training (see [6] for more detailed information about possible usages of process simulation).

When firmly supported by real-life studies and data, a virtual capability (implemented by using process simulation) can not only greatly extend our understanding of the observed phenomena, but also provide valuable support for deciding where to perform further research.

In particular, empirical studies and process simulation can be combined in such a way that 1) empirical knowledge is used for the development and calibration of simulation models, 2) results from process simulation are used for supporting real experiments, 3) real experiments and process simulation are performed in parallel (i.e., online simulation). In the following Sections, these three combination types are

Jürgen Münch is with the Fraunhofer Institute for Experimental Software Engineering (IESE) in Kaiserslautern, Germany (corresponding author's phone: +49-6301-707-139; fax: +49-6301-707-200; e-mail: muench@iese.fraunhofer.de).

Dieter Rombach is with the Fraunhofer Institute for Experimental Software Engineering (IESE) and the University of Kaiserslautern, Germany (e-mail: rombach@iese.fraunhofer.de).

Ioana Rus is with the Fraunhofer Center for Experimental Software Engineering, College Park, MD, 20740, USA (e-mail: irus@fc-md.umd.edu).

This is the author's version of the work. The definite version was published in Proceedings of the 4th Intl. Workshop on Software Process Simulation and Modeling (ProSim 2003), Portland, Oregon, USA, May 3-4, 2003.

discussed in more detail.

II. Using Empirical Knowledge for Simulation

Using a simulation model for prediction and understanding requires that the model reflects real world behavior for the intended contexts as accurately as possible. Simulation models are abstractions of the real world and, therefore, represent only selected aspects while lacking details. Several ways exist for constructing simulation models that represent real world behavior as accurately as possible:

Descriptive software process modeling, for instance, can be used to elicit static process knowledge (such as process and product structure, product flow, or resources). Surveys can be used for identifying state-of-the-practice procedures. Literature studies and expert interviews can be used for identifying influences between input and output parameters. An important source for the creation of accurate simulation models is empirical knowledge from real software engineering empirical studies. This knowledge can originate from previously performed experimental studies (if the context is comparable) or from experiments that are especially designed to support the development of a simulation model. Experimental software engineering studies can be classified into two domains [2]: 1) “teams” (i.e., [possibly single-person] groups that work separately), and “projects” (i.e., separate problems on which teams work). A team, for example, is a group of inspectors, and a project can be the inspection of a code document with a certain inspection technique. The sizes of these two domains can be used to provide a general classification of real experiments (see Figure 1).

# Teams per project	#Projects	
	one	more than one
one	Single project	Multi-project variation
more than one	Replicated project	Blocked subject-project

Figure 1. Classification of real experiments [2].

It should be mentioned that the degree of control varies: On the one hand, single project studies (such as case studies) usually provide only little control and, therefore, do not deliver statistically significant results. The advantage of single project studies is that they can typically be conducted in realistic contexts. On the other hand, blocked subject-project studies allow for a high degree of control and for statistically significant results. The disadvantage of blocked subject-project studies is that they typically have a very narrow focus and it is difficult to conduct those studies in a realistic setting. In the following, we describe how results from these study types can be used for simulation modeling:

- Empirical data from blocked subject-project studies (trends and behavior of software development parameters) can be used for developing simulation models. This data can help to identify patterns and relations between parameters. Furthermore, blocked subject-project studies can be especially designed to determine the effects of technologies in defined contexts, i.e., detailed influence relations can be quantified.
- Empirical data from single project studies can be used to initialize the input parameters and to calibrate a simulation model to a specific context (e.g., project, environment).
- Replicated project studies can be used to increase the significance of empirical results. Consequently, using data from replicated project studies in simulation models improves the empirical basis of the model and can lead to better calibrations.
- Multi-project variation studies involve a single team in a set of projects. Multi-project variation data can also be used for calibrating the simulation model to a specific context. Using empirical data from multi-project variation studies requires careful analysis of learning effects.

The use of data from experimental studies can involve the application of further data analysis steps. Quantifying relationships can be done, for instance, by using stochastic analysis or data mining techniques. Data mining techniques can be used to generate new rules to describe relations that were not considered before or were not elicited by expert interviews. The available empirical data determines the data mining technique. An example for applying data mining techniques to simulation model development can be found in [7].

Furthermore, data from experimental studies can be used to validate a model against real world behavior, and further fitting of the model can be done. Deviations between model behavior and real world behavior can have several causes. Examples for such causes are a mismatch between the real context and the scope of validity of the model, incorrect assumptions in the model, or the non-consideration of an influence factor in the model.

III. Using Simulation for Real Experiments

Simulation can be seen as a means to improve the planning, design, and analysis of real experiments. Additionally, it can be applied to generalize the results of real experiments (within limits). In detail, process simulation can be used for supporting real experiments in the following ways:

- New hypotheses can be investigated empirically before performing expensive real experiments. As a consequence, uncontrolled and controlled parameters of real experiments can be determined more precisely. Sensitivity analyses, for instance, allow for investigations with respect to parameter combinations that have significant impact on the variables of interest. Investigating a hypothesis with a simulation model might also lead to observations that are questionable. Both situations can motivate real experiments. Using simulation helps to determine which real experiments are needed most, and to prioritize real experiments.

- Simulations of empirical studies and their variations can be executed in different contexts (by changing simulation parameters' values) with negligible cost and time. Independent variables in real experiments are typically fixed or varying in an uncontrolled manner. During the design of a real experiment, it should be taken into consideration that the varying uncontrolled variables do not have an impact on the dependent variables (i.e., the parameters of interest). The independent variables that are fixed in a real experiment characterize the context of the experiment. In order to apply the model to different (but similar) contexts, a variation of these variables is necessary. This can be done cost-effectively with simulation. It should be mentioned that the variation of the context affects the external validity of the results. A careful analysis of threats to validity should be performed (for further information, see [5]).
- The scale of empirical studies can be enlarged. The variability of controlled parameters in real experiments is typically small. The variation range of a controlled variable can be extended within a simulation model. Furthermore, simulation models can be developed that cover the scopes of several real experiments. In both cases, a careful validity analysis is necessary before using the simulation results.
- Global effects can be studied by modeling the bigger process that includes the empirically studied sub-process. Due to practical constraints, real experiments very often cover only small parts of a software process (e.g., an inspection process). Practical questions often require a broader process view. An example question is, how the inspection effectiveness impacts the overall cost of a project. Modeling the global process in a simulation model can help to determine the global effects of a local real experiment.
- Simulation can be used to determine data needs for real experiments. The development of a simulation model before a real experiment enforces the determination of the dependent and independent variables. This alleviates the instrumentation of the real experiment with metrics. The derivation of metrics can be done in a goal-oriented way (as described, for instance, in [4]). In addition, the development of a simulation model reveals what kind of data is available and what is missing. A next step towards the design of a real experiment is to adjust the context and determine the level of control.

IV. Online Simulation

Real experiments and process simulation can be combined in such a way that they are performed in parallel. This so-called online simulation offers the following additional capabilities to the advanced software engineering laboratory:

- Real processes and virtual processes can be performed in parallel and influence each other. The scale-up of real experiments can be performed in real time. This allows for examining effects in a real setting without neglecting the global effects.
- Data from real experiments can be validated in real time. This offers additional data validation capabilities to traditional validation techniques (as described in [3]).
- Simulators can be used as a means for training. Simulation allows for practice-oriented and scenario-oriented training of software developers, managers or other target groups. The learner can, for example, act as if he was playing his role in the real world and see the consequences.

V. Introduction to the SEV Project and its Goals

To address issues of combining real and virtual experiments, a cooperation project between several Fraunhofer institutes (IESE, ITWM, FIT), named *"SEV (Simulation-based Evaluation and Improvement of Software Development Processes)"* [13], was initiated in 2001. This cooperation project is supported by the German Federal Ministry of Education and Research (BMBF) and the Stiftung Rheinland-Pfalz für Innovation. Several project results profited from close collaboration with the Fraunhofer Center for Experimental Software Engineering, Maryland. The project aims at developing a simulation platform for software development processes and experiments. Simulation is used to support decisions on process alternatives for a project on the basis of existing knowledge. Thereby, new development knowledge can be gained faster and more cost effectively. The goal is divided into the following sub-goals:

- Support forecasting and conducting 'what-if-games' for the selection of software development processes and development approaches. The decision about process alternatives for a concrete project shall be reached on the basis of existing knowledge. Results can be used for project planning, systematic improvement of processes, and risk analyses. In addition to this, the integration of optimization algorithms promises to identify potentials for further improvements of software development processes, which could only be identified at high costs with purely empirical methods.
- Reduce the costs of new experiments by using simulation. Well-understood parts of experiments can be simulated as well as accompanying processes such as technical processes. By analyzing results of simulation runs (e.g., by performing sensitivity analyses), fields for

real experiments can be identified.

- Support practice-oriented teaching and training of project managers and planners. Planning and project scenarios can be executed and the effects of planning decisions can be visualized graphically (see [8] for an example approach).

To reach the goals of this project, a simulation platform consisting of an evaluation and improvement method, an integrated tool environment, and a simulation cockpit for the effective application of the method, is being developed. The simulation platform effectively supports project managers and planners, people performing experiments, process engineers (in the context of process improvement), and trainers (in the context of practice-oriented teaching). Using the simulation platform promises the following advantages:

- Cost reduction by simulating software development processes and human behavior.
- Cost reduction by better selecting and focusing the scope of real experiments.
- Coupling with optimization methods promises results that could not be obtained by varying the impact factors in real or simulated experiments.
- The demonstration of the benefits of new methods in the context of an industrial development environment.
- Practice-oriented learning with respect to project planning and management can be performed in a scenario-oriented way.
- The simulation platform with the corresponding method can be integrated into process improvement programs.

Important issues of this project are:

- Use of empirical studies (e.g., data from blocked subject-project studies and replicated project studies) to develop simulation models,
- Combination of real and experiments and simulation modeling in the area of defect detection techniques,
- Use of simulation for the definition of real experiments,
- Use of data mining techniques for analyzing data from empirical studies,
- Modularization of simulation models in order to allow for fast and cost effective modeling and the construction of large-scale models,
- Development of guidance that helps to select appropriate modeling techniques according to the purpose, the scope and the key result variables,
- Development of visualization techniques specialized for the simulation of software development processes,
- Identification and exploitation of potentials for optimization, and
- Possibility of estimating the importance of model parameters and their sensitivity.

VI. CONCLUSIONS AND OUTLOOK

Process simulation can be seen as a way of packaging and enriching the results of empirical studies. However, the relationship between empirical studies and process simulation is relatively unexplored [9].

The implementation of advanced software engineering laboratories leads to several new research questions such as: How to effectively combine real and virtual experiments? How to validate hypotheses for real experiments with process simulation? How to scale results from real experiments without significantly lowering the validity?

The industrial use of such laboratories leads to several further challenges (e.g., customer-oriented visualization of the results, mapping the effects on business goals, identifying market demands for simulation-based training).

ACKNOWLEDGMENT

This work was supported in part by the German Federal Ministry of Education and Research (SEV Project) and the Stiftung Rheinland-Pfalz für Innovation (ProSim Project, no.: 559). We would like to thank Sonnhild Namingha from the Fraunhofer Institute Experimental Software Engineering for reviewing the first version of the article.

REFERENCES

[1] V. R. Basili, F. McGarry, R. Pajerski, R., and M. Zelkowitz, "Lessons learned from 25 years of process improvement: The rise and fall of the NASA Software Engineering Laboratory", In Proceedings of 24th International Conference on Software Engineering (ICSE) 2002, Orlando, Florida, May 2002.

[2] V. R. Basili, R. Selby and D. Hutchens, "Experimentation in Software Engineering", IEEE Transactions on Software Engineering, July 1986.

[3] V. R. Basili, D. M. Weiss: A Methodology for Collecting Valid Software Engineering Data. TSE 10(6): 728-738, 1984.

[4] L. C. Briand; C. Differding; D. Rombach, "Practical Guidelines for Measurement-Based Process Improvement". Software Process - Improvement and Practice, 2(1996), 253-280.

[5] C. Judd, E. R. Smith, and L. Kidder: Research Methods in Social Relations, 6th ed., Harcourt Brace Jovanovich College Publishers, ISBN 0-03-031149-7.

[6] M. I. Kellner, R. J. Madachy, D. M. Raffo, "Software process simulation modeling: Why? What? How?", Journal of Systems and Software 46, 2-3, 1999, 91-105.

[7] H. Neu, T. Hanne, J. Münch, S. Nickel, and A. Wirsen, "Simulation-Based Risk Reduction for Planning Inspections," In: Proceedings of the 4th International Conference on Product Focused Software Process Improvement (Profes 2002), December 9-11, Lecture Notes in Computer Science 2559, (M. Oivo, S. Komi-Sirviö, eds.), Springer-Verlag, pp. 78-93, December 2002.

[8] D. Pfahl, "An Integrated Approach to Simulation-Based Learning in Support of Strategic and Project Management in Software Organisations", Stuttgart: Fraunhofer IRB Verlag, 2001.

[9] D. Raffo, T. Kaltio, D. Partridge, K. Phalp, J. F. Ramil. "Empirical Studies Applied to Software Process Models". In Empirical Software Engineering, Vol.4, No.4, pp. 353-369, Dec. 1999.

[10] D. Rombach, "Experimentation - Engine for Applied Research and Technology in Software Engineering", Proc. NASA's 24th Annual Software Engineering Workshop SEL '99, Software Engineering Laboratory, Greenbelt, Maryland, USA, 1999.

[11] H. D. Rombach, V. R. Basili, R. W. Selby. Experimental Software Engineering Issues: Critical Assessment and Future Directions. Lecture Notes in Computer Science. Springer Verlag, 1993.

[12] I. Rus, S. Biffl, M. Hallig, "Systematically Combining Process Simulation and Empirical Data in Support of Decision Analysis in Software Development". In Proceedings of the 1st International Workshop on Software Engineering Decision Support (SEEDECS'2002), Ischia, Italy, 2002.

[13] Simulation-based Evaluation and Improvement of Software Development Processes (SEV). www.sev-fraunhofer.de

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