

# A Computerized Infrastructure for Supporting Experimentation in Software Engineering

Paula G. Mian, Guilherme H. Travassos and Ana Regina C. da Rocha

COPPE / UFRJ – Computer Science Department  
Cx. Postal 68.511, CEP 21945-970, Rio de Janeiro – Brazil  
{pgmian, ght, darocha}@cos.ufrj.br

**Abstract.** *An experimental Software Engineering Environment (eSEE) is an infrastructure capable of instantiating software engineering environments to support the definition, planning, execution and packaging of experimental studies as well as knowledge produced throughout the experimentation process. This paper presents the concepts and architectural solutions for the eSEE infrastructure and the current state of this research work at COPPE/UFRJ.*

**Keywords:** *CASE, Knowledge management, Software Engineering Environments, Experimental Software Engineering.*

## 1. Introduction

One of the great interests in the experimental studies is the increasing viewpoint that Software Engineering must have strong foundations of a scientific engineering discipline, and that the techniques to improve software development and to evolve processes must be available to professionals. However, executing an experimental study especially in Software Engineering is hard, time consuming and generates great volume of information and knowledge that are complex to manage [Shull et al. 2001].

Therefore, it is necessary to define and build a computational infrastructure to support the management of knowledge involved in the experimentation process in Software Engineering.

The work of Mian et al. (2004a, 2004b) introduced the concept of *experimental Software Engineering Environment* (eSEE) – an infrastructure able to instantiate software engineering environments to support knowledge management throughout the SE experimentation process, including the definition, planning, execution and packaging of experimental studies in Software Engineering.

This paper presents the current stage of our research work regarding environments for experimentation on Software Engineering (SE). Section 2 presents the defined architecture for the eSEE's infrastructure. Section 3 presents the architecture solutions for the eSEE infrastructure. In section, we present an evolution for the experimental studies taxonomy. Finally, section 5 presents our conclusions and future work.

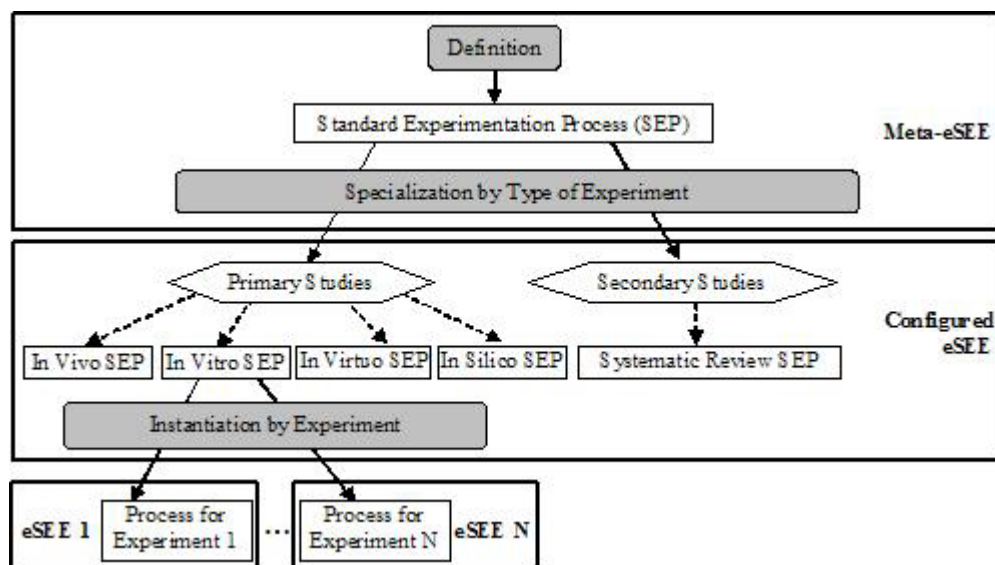
## 2. The eSEE Infrastructure

Usually, a lot of clerical activities take part in the whole experimentation process, making it tedious and repetitive mainly for novice experimenters. The eSEE infrastructure looks for ways to support some of these SE experimentation process

activities by providing automated (or semi-automated) computer based tools integrated into a software engineering environment.

An eSEE must manage the software engineering experimentation process, including knowledge acquired when defining, planning, executing and packaging experiments. eSEE makes available experimentation process models, experiment packages, data representation standards, knowledge management facilities, tools, services, quality and computerized models (simulators) for a specific experiment's types [Mian et al. 2004b].

The proposed computerized infrastructure for eSEE is based on the experimentation process and package models described by [Amaral and Travassos 2002]. To instantiate an eSEE for a specific study, the experimental study type intended to be performed must be defined. Three levels of knowledge organization about the experimentation process have been identified: knowledge for any kind of experimental study (meta level), knowledge for each experimental study (configuration level), and knowledge for a specific experimental study (instance level). These levels reflect themselves at eSEE's architecture. All the information generated throughout the experimentation process must be packaged for future use by researchers.



**Figure 1. eSEE's Infrastructure.**

The levels composing eSEE's infrastructure, as shown in Figure 1, are [Mian et al. 2004a]:

- *Meta-eSEE*: the meta-level contains common knowledge regarding SE experimental studies, including software engineering knowledge. This knowledge may be captured by software engineering ontologies [Oliveira et al 2004] integrated with an experimentation ontology. At this level, a standard experimentation process (SEP) is defined. The SEP represents the basis to instantiate standard processes for each type of experiment;
- *Configured eSEE*: the configuration level contains knowledge regarding specific experimental studies. At this level, specific eSEE's instantiation can be accomplished by choosing experimental study type and adding specific experimental study's characteristics. The instantiated environment, then, is generated with the specific knowledge to that study type;

- *eSEE*: environment supporting the definition, planning and execution of the specific experimental study type. For each new experimental study type, one *eSEE* should be instantiated to manage the experimentation process and knowledge.

Once the instantiated environments are created, some facilities must be available to support the accomplishing of experimental studies. For instance, a process execution engine should be integrated to the environment to allow process tracking in each *eSEE*. Nevertheless, experimental studies should be packed into experiment packages repositories throughout the experimentation process. Also, the *eSEE* infrastructure must make available knowledge about the experimentation process, methods, techniques and tools to assist the software engineering researcher managing the experimental study [Mian et al. 2004b].

### 3. Architectural Solutions for the *eSEE* Infrastructure

To evaluate the proposed architecture feasibility (shown in Figure 1), an initial prototype has been built [Dias Neto et al. 2004]. This prototype defined an infrastructure to build experiments knowledge base, where experimental studies can be defined, planned, executed, analyzed and packaged, and the results stored and made available to researches. It controls the experimental study's artifacts fulfilling throughout the experimentation process execution. The prototype makes available knowledge to aid the correct filling of the experimental documentation; provides experimental processes execution mechanisms and provides visualization mechanisms to show the information contained in the documents produced during the process conduction.

To build this prototype, the following architectural requirements for *eSEE* infrastructure were considered [Dias Neto et al. 2004], [Chapetta et al 2005]:

- To adopt web-based characteristics for the system's construction allowing the use on different localities and inter-institutional researches;
- To use e-services and external components to support tasks in the experimentation process;
- To provide knowledge management mechanisms so that Experimental Software Engineering is a knowledge-intensive activity; and
- To allow the flexibility and independence between the conceptual representations of the expected artifacts, related to a filling process, and their computational representation, by abstracting association maps.

These requirements were reflected into the prototype architecture, composing three macro components: Meta-Configurator, Instantiation Environment and Execution Environment.

The *Meta-Configurator* is responsible for the meta-descriptions activities and its main functionality is to define and configure the basic elements used by the others components. Theses elements are: (1) Process Models: allow the description of experimentation processes based in a Software Process Ontology [Oliveira et al 2004]; (2) Documents Models: describe documents produced throughout experimentation process based in a Document Meta-Model; and (3) Configured Services: information about e-services that could support some activity in experimentation process. Theses e-services, available or not into *eSEE*, are customized to be used in an Execution

Environment. The work of Chapetta et al (2005) describes the tools built to support the definition of each one of these elements.

The *Instantiation Environment* allows to instantiate environments to support experimental studies, based on the document model and the process model, defined in the first component. To reach this objective, some tasks have to be accomplished: (1) to select the experimental process that will be executed; (2) to define the relationship between a process model, a document model that will be filled in by the researcher, and the configured services that will be used as tools to realize tasks through an Association Map; (2) based on the selected map, to instantiate an Execution Environment [Dias Neto et al. 2004].

The *Execution Environment* is responsible for providing facilities to monitor, control and execute experimentation process. In order to perform these activities, a CASE tool that supports software engineering processes enactment and control, called enactPro [Mafra et al. 2004], was integrated to the infrastructure. Through enactPro, the SE researcher accesses the configured services defined for each process' artifact and controls the experimental process execution.

#### **4. eSEE's Supported Experimental Studies Taxonomy**

The original eSEE infrastructure proposal [Mian et al 2004a] intended to support mainly primary studies (surveys, case studies and experiments). The infrastructure instantiation would occur by choosing the experimental study type according to the experiment taxonomy described in the work of [Travassos and Barros 2003]. This taxonomy classifies primary experimental studies in four categories:

- *In vivo* experiments: such experiments involve people in their own environments. In software engineering, experiments executed in software development organizations throughout the software development process and under real circumstances can be characterized as *in vivo*;
- *In vitro* experiments: such studies are executed in a controlled environment, such as a laboratory or a controlled community. In software engineering, most *in vitro* experiments are executed in universities or among selected groups of a software development organization;
- *In virtuo* experiments: such experiments involve the interaction among participants and a computerized model of reality. In these experiments, the behavior of the environment with which subjects interact is described as a model and represented by a computer program. In software engineering, these studies are usually executed in universities and research laboratories and are characterized by small groups of subjects manipulating simulators;
- *In silico* experiments: these studies are characterized for both the subjects and real world being described as computer models. In this case, the environment is fully composed by numeric models to which no human interaction is allowed. Due to the need of a large amount of knowledge, *in silico* studies are still very rare in software engineering, being limited to areas where subject participation is not an experimental study issue or intelligent agents can replace human subjects. For instance, we can find *in silico* studies applied to software usability experimentation, such as software performance characterization.

The primary studies conduction is important to contribute to a larger body of scientific knowledge, by refining techniques, methods and tools, and generating new hypotheses. However, in a similar way to other scientific areas, SE researchers and practitioners need to efficiently integrate and validate data, generated by primary experimental studies, to provide a solid basis for rational decision-making. Systematic reviews can represent a useful tool for achieving such purpose.

The term Systematic Review (SR) is used to refer to a specific methodology of research, developed in order to gather and evaluate the available evidence pertaining to a focused topic. In contrast to the usual process of literature review, unsystematically conducted whenever one starts a particular investigation, a SR is developed, as the term denotes, in a formal and systematic way [Kitchenham 2004].

Though its importance, conducting systematic reviews is not a simple task. It evolves performing complex actives and understanding specific concepts and terms that may be unknown to researches. Besides, the review must be planned before its execution, and the whole process must be documented, including the intermediary results [Biolchini et al. 2005]. For these reasons, we believe that SE researches would benefit from a computational infrastructure, such as the eSEE one, to support the SR process and to package secondary studies results.

Therefore, beyond the experimental primary studies taxonomy [Travassos and Barros 2003], eSEE infrastructure also intends to support secondary studies, as shown in Figure 1. There is a Systematic Review Conduction Process [Biolchini et al. 2005] already described in order to guide researchers in performing systematic reviews in the Software Engineering domain. This process will compose the eSEE's standard experimentation processes repository to allow the instantiation of experimental environments to support the execution of systematic reviews.

## **5. Conclusion and Future Work**

In this paper, we presented the current state of our research regarding eSEE, an infrastructure able to instantiate software engineering environments to support knowledge management throughout the SE experimentation process, including the definition, planning, execution and packaging of experimental studies in software engineering. The main contributions for this research work were: the definition of the architectural solutions for the infrastructure, the construction of prototype to test the proposed architecture, the evolution of the experimental study taxonomy, and the definition of a systematic review conduction process.

The next steps of this work include the following activities:

- Experimentation process adaptation, regarding the four staged taxonomy [Travassos and Barros 2003] and the definition of a standard experimentation process to support each one of them;
- Experimentation ontology definition to organize knowledge about experimental software engineering in eSEE; used to identify general characteristics of each experimental study type; and.
- To define an initial set of tools to populate the eSEE infrastructure. For instance, a tool to support experimental studies execution.

Further information about all the research work that contributes to the eSEE project can be obtained at the ESE team's homepage (<http://www.cos.ufrj.br/~ese>).

## Acknowledgments

The authors acknowledge CAPES and CNPq for the financial support to this work. This research is granted by CNPq – Projeto Universal.

## References

- Amaral, E. A. G. and Travassos, G. H. (2002). “Em busca de uma abordagem para Empacotamento de Experimentos em Engenharia de Software”, *Proc. of 2<sup>nd</sup> Ibero-American Symposium on Software Engineering and Knowledge Engineering (JIISIC'02)*, Brazil.
- Biolchini, J.; Mian, P.G.; Natali, A.C. and Travassos, G. H. (2005). “Systematic Review in Software Engineering: Relevance and Utility”, *Technical Report ES67905*, PESC - COPPE/UFRJ. Available at <http://cronos.cos.ufrj.br/publicacoes/reltec/es67905.pdf>
- Chapetta, W. A., Santos, P.S.M. and Travassos, G. H. (2005). “Supporting Meta-Description Activities in Experimental Software Engineering Environments”, Submitted to the *2<sup>nd</sup> Experimental Software Engineering Latin American Workshop (ESELAW'05)*, Brazil.
- Dias Neto A.C.; Barcelos, R.F.; Chapetta, W.A.; Santos, P.S.M; Mafra, S.N. and Travassos, G.H. (2004). “Infrastructure for SE Experiments Definition and Planning”, *Proc. of 1<sup>st</sup> Experimental Software Engineering Latin American Workshop (ESELAW'04)*, Brazil.
- Kitchenham, B. (2004). “Procedures for Performing Systematic Reviews”, Joint Technical Report Software Engineering Group, Keele University, United Kingdom and Empirical Software Engineering, National ICT Australia Ltd, Australia.
- Mafra, S.N.; Barros, M.O.; Travassos G.H. (2004). “enactPro: Automatizando Processos de Software”, *Proc. of XVIII Simpósio Brasileiro de Engenharia de Software - Caderno de Ferramentas*, Brazil.
- Mian, P.; Travassos, G. and Rocha, A.R.C. (2004a). “eSEE: a Computerized Infrastructure for Experimental Software Engineering”, *Proc. of 1<sup>st</sup> Experimental Software Engineering Latin American Workshop (ESELAW'04)*, Brazil.
- Mian, P.G.; Travassos, G.H.; Rocha, A.R.C. and Natali, A.C.C. (2004b). “Towards a Computerized Infrastructure for Managing Experimental Software Engineering Knowledge”, *4<sup>th</sup> Ibero-American Symposium on Software Engineering and Knowledge Engineering (JIISIC'04)*, Spain.
- Oliveira, K.M.; Zlot, F.; Rocha, A.R.C.; Travassos, G.H.; Silva, C.G.M. and Menezes, Silva C. (2004). “Domain Oriented Software Development Environment”, *Journal of Systems and Software*, v. 72, n. 2, p. 145-161 (2004).
- Shull, F.; Carver, J. and Travassos, G. H. (2001). “An Empirical Methodology for Introducing Software Processes”, *Proc. of 8<sup>th</sup> European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9)*, Vienna.
- Travassos, G. H. and Barros, M. O. (2003). “Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering”, *Proc. of 2<sup>nd</sup> Workshop in Workshop Series on Empirical Software Engineering the Future of Empirical Studies in Software Engineering*, Italy.