A Semantics-based Evidence Portal for Learning Software Organizations

Andreas Jedlitschka*, Jörg Rech* & Dietmar Pfahl**

*Fraunhofer Institute Experimental Software Engineering
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
jedl@iese.fraunhofer.de, rech@iese.fraunhofer.de

**University of Calgary, Schulich School of Engineering, Calgary, Canada
dpfahl@ucalgary.ca

Abstract. The amount of methods, techniques, and tools in software engineering is increasingly growing. In order to keep track of the progress of these technologies and methodologies, a software engineering manager needs an easy and fast access to evidence about their effectiveness and efficiency. A semantics-based evidence portal for software organizations helps them to continuously gather and aggregate evidence about software engineering technologies and methodologies. The information provided through the portal is targeted to support managers in planning software process improvement activities based on data collected semi-automatically from published evidence.

1 Introduction

In today's software development organizations, methods and tools are employed that frequently lack sufficient evidence regarding their suitability, validity, limitations, qualities, costs, or associated risks. The need to select the best-suited method, technique, or tool in a given business context is becoming more and more important. From a business perspective the trade-off between time-to-market, quality, and cost is a crucial factor for decision-makers. While new findings from research await their transfer into industrial practice, systematic assessment, selection and infusion of these findings with regard to business objectives and context is lacking.

In order to support software managers in selecting suitable software engineering (SE) technologies it is necessary to provide a defined level of evidence about the effectiveness of a specific SE technology regarding certain goals and in a given context. Nevertheless, relevant information for the decision making in SE organizations is often hard to access or, even worse, not available [Gla04].

In this paper, we present ongoing research towards the semantics-based evidence portal (SEP). SEP will support goal-oriented decision making for SE technology selection in learning software organizations as defined by Ruhe [Ruh03] and described by Jedlitschka et al. [JPB04]. Furthermore, we summarize findings from a pilot study [JP04] and a survey. These findings were an initial input that helped us derive usage scenarios of the SEP for practitioners and researchers.

2 The Semantics-based Evidence Portal

In this section, we describe the key elements of a prototypical Software Engineering Decision Support System (SE-DSS) targeting at supporting decision-makers in technology selection. The proposed SE-DSS will take the form of a SEP. Analogous to ideas advocated in the semantic web research community, experiments are treated as structured content documents with additional metadata that describe semantic issues and might be used for further inference steps (e.g., (semi-) automated aggregation of experiments).

We first present scenarios that describe how the SEP can be used. Then we present the underlying infrastructure framework. Finally, we give an introduction into the structure of the repository storing evidence about software technology derived from various sources of information.

2.1 Scenarios for Semantics-based Evidence Portal usage

One of the major goals of SE-DS is to support software managers in selecting suitable SE technologies. Suitability implies the existence of a defined level of evidence about the effectiveness of a specific SE technology in a given context. In our pilot study interviewees stated that they have the following *information need* that should be satisfied by a SE-DSS for the purpose of technology selection [JP04]:

1. Information on *effectiveness and efficiency* of the technology with respect to defined quality aspect (e.g., performance)
2. Information on the *cost* for introduction/applying of the technology
3. *Preconditions* that have to be fulfilled prior to the application of the technology
4. *Context* information
5. *Structured access* to the content, e.g., by improved navigation and search

While the pilot study was performed with industry-oriented consultants of a research organization, a second study was performed involving industrial decision-makers. The decision-makers were asked about their responsibilities with regards to technology selection and to provide information on key performance indicators (KPIs) for product development. It turned out that technology selection needs to be supported in the light of the following ranking of *top goals*: (1) product quality, (2) no budget overrun, (3) no time overrun. This ranking was based on aggregations of more detailed KPIs.

Both studies lead to the assumption that industrial decision-makers, i.e., managers, are often not interested in the number of defects that a particular quality assurance (QA) technique might potentially find. They are interested in the impact of a particular QA technique on the overall project goals (which include many issues, e.g., functionality, quality, time-to-market, budget constraints, etc.). Furthermore, managers are interested in the impact of a particular SE technology on the overall business value.

Based on the results of our studies we define the usage scenarios for the envisioned SEP from the viewpoint of a practitioner. Users from industry, interested in finding the most suitable technologies in a given context, are the focus of the scenarios. These scenarios describe the use of the SEP to give answers on questions of the following types: “How often was it applied (successfully)?”, “Who has applied it (industry, research)?", or "What were the results (e.g., impact on cost, quality, and time, cost-benefit-ratio, return on investment)?". The answers will be augmented with all available context information in order to increase transferability of the available data to the inquirer's environment. We support two main scenarios targeting at decision-makers in software development organizations:

1. *Goal-oriented search for best practices:* The first scenario is targeted towards the search for a technology that improves a key feature of the software development process. The central questions are "How can we improve quality, reduce cost, or reduce time?" either based on curiosity or an existing problem (i.e., "We have a problem, e.g., with the product quality, what can be done?"). Characteristics to narrow the search space are: technology's impact on quality, schedule/time, and cost. In a second step the key features might be refined to narrow down the search. Quality is refined by the quality characteristics from, i.e., ISO 9126, cost subdivided into personal (e.g., work or training) and resources (e.g., tools), and time refined by the lifecycle phase (e.g., time for development, time including maintenance).
2. *Context-oriented search for best practices:* The second scenario is a context-oriented search for a better solution than currently in use. While the first scenario excluded information about the context (e.g., the domain) this scenario excludes information about the problem. Central question is "What is the best practice for my software/system engineering process?". Characteristics to narrow the search space are the application domain (e.g., automotive), SE phase (e.g., maintenance), or the SE object (e.g., architecture).

2.2 Infrastructure of a Semantics-based Evidence Portal

In this section, we sketch the underlying infrastructure framework for the SEP. The existence of such a framework is needed in order to facilitate the combination of qualitative and quantitative data from current projects with distributed experience and lessons learned from previous (internal or external) projects. The framework has three main characteristics:

- First, it offers *decision support* which is based on both internal and external experience. Typically, internal experience stems from project databases, measurement programs, and other company-specific organizational repositories. Typical sources for external experience are public web-based repositories, like the ESERNET repository [CW03].
- Second, due to the extensive involvement of research organizations in the build-up and maintenance of public web-based repositories, framework users can expect to have access to *state-of-the-art experience*.
- Third, the proposed framework will offer *integrated tools and techniques* that help to aggregate and synthesize experience across the borders of different SE techniques along the software development life cycle.

More specifically, the framework aims at supporting comprehensive decision-making in that sense that it will provide information for the evaluation of the effectiveness and efficiency of combinations of SE technologies. This information stems from isolated empirical studies each focusing on a single SE technology.

Figure 1 shows the context in which the SEP will be used. To become efficient, organization-specific improvement activities in industry need to be connected with external sources of information. In a complementary sense, the same is true for academia, i.e., research institutes and universities. In Figure 1, the arrows from the SEP ("Semantic-based Portal") to industrial organizations indicate the use of external data to enrich the company-specific software process improvement approach, especially by supporting decision-making in the context of technology selection. The arrows from and to research describe the contributions of and benefits for empirical research conducted within academic organizations (i.e., research institutes and universities). For example, results from controlled experiments that measure the effectiveness of a particular technique in a local context, might be combined with related data from case studies that provide evidence on the effectiveness of this technique in a global context, e.g., with regards to project performance measures.

The motivation and prerequisites for combining data sources (e.g., project data bases) from industry and research has been outlined in [JP03]. In order to enlarge the available experiment base in the SEP and to add more analytic power, simulation could complement empirical work [MPR05].

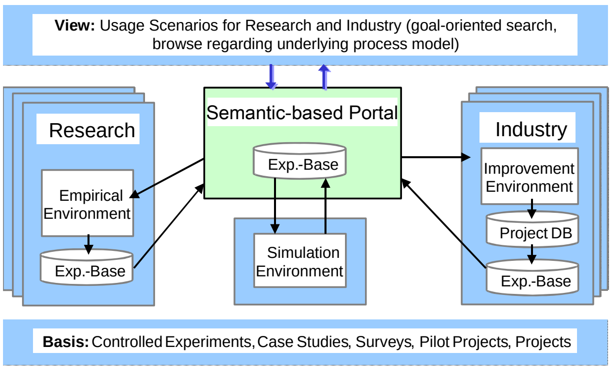

Fig. 1. The Infrastructure of the Semantic-based Evidence Portal

As described in [JPB04] we implement the SEP by following a 3-tier architecture approach. The front-end is realized by using internet enabling technologies, like Java Server Pages, the middle-tier (business logic) is realized in Java and the backend is realized by using MySQL as a database.

2.3 DB-Stucture for Sematics-Based Evidence

In order to come up with an initial structure for the repository, which also represents a minimal set of elements that should be reported by experimenters, we surveyed a number of experiments on inspection techniques [JC04]. Furthermore, we used the information elicited from the two surveys mentioned in section 2.1 to construct a database schema that has the potential to support the answering of questions by industrial and academic users. We found some semantic elements, which, if commonly reported, would help to classify the experiments and allow for aggregation of information. Triggered by a similar idea, Vegas et al. [VJB03] described and evaluated a process for developing characterization schemas to create experience bases, especially suited for testing techniques. By combining our results with those from Vegas et al., we derived a schema for reporting results from experiments in the area of defect reduction. Formalization of a document model for reporting about controlled experiments in software engineering [JP05], as well as the usage scenarios presented in this paper led us to the current schema of the repository. The document model in form of reporting guidelines for controlled experiments reflects the information needs described in the introduction section. It represents the logical model that is realized in a physical model of a relational database schema as depicted in Figure 2. The “core” of the schema is the entity *study*. It contains relevant information about the evaluation of a specific *technology* and provides links to more specific information about the evaluation itself, such as the *impact* effect or the *application domain*. The entity *technology* contains all information regarding the technology itself, e.g., a technology description, whereas *impact* describes associated evidence, its validity, and provides a link to its origin.

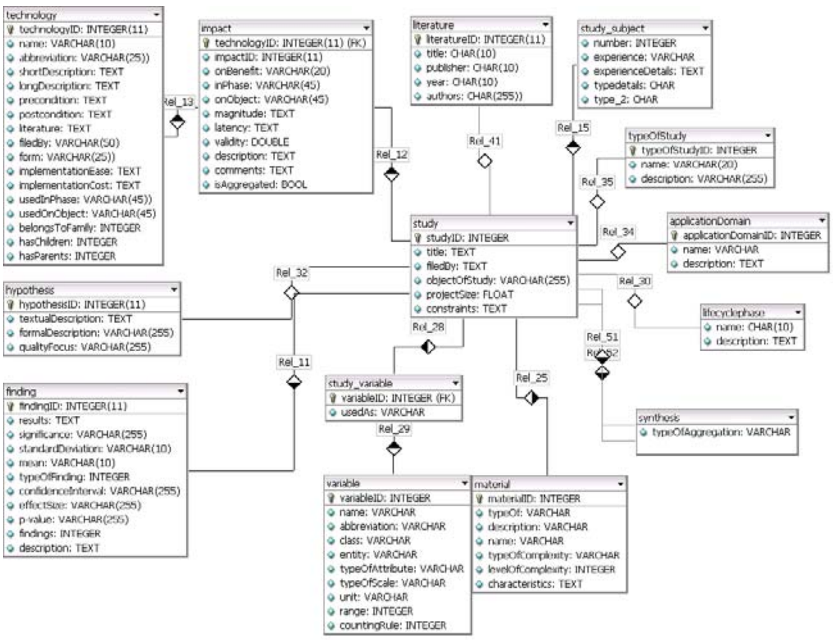

Fig. 2. Compacted design of the database schema (n:m relationship tables are not shown).

In order to facilitate the acquisition of studies and to fill the repository, we aim at a semi-automatic support. This support will consist of an internet wrapper that will collect studies from several predefined sources, such as citeseer™ or google™ scholar. Furthermore, we will use an ontology of empirical SE that will enable us to restrict the result set for our purposes. For example, if we are only interested in so-called controlled experiments, we have to define a set of keywords that are related and that could have been used by authors instead. This pre-processing will classify the results set into three categories: relevant, probably relevant, and not relevant. Without having been able to comprehensively test the system, we are conservative and assume that, unfortunately, most papers will end up in the category "probably relevant". These papers will have to be further looked at by humans. For this purpose we foresee implementation of a clearing house, following an idea described in [ST05].

After this first step the relevant papers will be further processed involving text analysis. The idea behind this analysis is to extract information from the paper and store it into the database. As of today it will be hardly possible to automatically extract the information, because the information is hidden somewhere in the publication.

Here is where reporting guidelines (logical or document model) as proposed in [JP05] can be helpful. Guidelines will help assure that information (if available) is always in the same place, headings and content of sections are comparable, and in some parts they require formal specification, e.g., of the hypothesis, variables, and findings. That will facilitate text extraction on a syntactical basis. The mapping of relevant information from arbitrary positions in the paper to the cells in the database schema (i.e., physical model) will be based on the semantic level using techniques such as latent semantic analysis (LSA). After having initially filled the data of a publication into the repository the system will automatically approach the author (using the email address in the contact information) of the publication and ask him to check whether the information extracted in the right way. In addition, an author will be asked to provide information that was not found by the automatic extraction. A similar approach has already been tried out in the ESERNET [CW03] project, where the extraction was done manually before the authors have been asked for cross-checks.

3 Discussion

This section gives a brief overview of recent research related to building software engineering decision support systems (SE-DSS). We start with decision support research in general and then focus on decision support for SE.

Building upon the research already done in the field, our emerging work towards comprehensive SE-DSS can be characterized as follows. We aim at the development of a semantics-based experiment platform focusing on project planning and control based on objectives from the industry. Currently, one of the main problems is the collection and formalization of evidence in form of experiments or case studies from published evaluations. Other groups like, Sjøberg et al., who found 103 controlled experiments in SE (which represents 1.9 percent of the SE articles they surveyed) [SH+05], partially confirm our initial findings [JC04], that in most cases information is hard to extract from a publication.

In an attempt to make it more attractive for SE experimenters to include their work into a common repository, Vegas et al. [VJB03] describe and evaluate a process for developing characterization schemas to create experience bases, especially for testing techniques. As described earlier, we have incorporated the conceptual model of Vegas et al. in an earlier version of our logical model.

In SE, many different areas for decision-making can be found along the software life cycle. Ruhe gives an overview on recent research related to SE-DS [Ruh03]. He characterizes five areas in which SE-DS research is progress, i.e. requirements, architecture and design, adaptive and corrective maintenance, project planning and control, and verification and validation. Information from the SEP might be used in a SE-DSS to support software engineers during these processes.

For the purpose of product-focused software process improvement, Birk describes how to use so-called technology experience packages (TEP) in support of technology selection [Bir00]. TEPs are a specific representation of software project experience. Based on a two-step context evaluation with respect to the given situation, appropriate TEPs are ranked and selected by the decision maker. The content of the TEPs is initially acquired from literature and iteratively adapted to the specific needs of an organization. Currently, we are investigating the usage of enhanced TEPs.

Instead of developing a new framework, or adapting an existing KM framework from specialized application domains, such as the one proposed by Biffl et al. [BH03], the comprehensive SE-DSS will be methodologically backed-up by a framework which is bases in the Quality Improvement Paradigm (QIP) [BCR01].

Beside this basic background our research is build upon, several similar approaches to support technology selection using portals are in development. For example, Shull et al. report on an implementation of a best practice clearinghouse to identify, analyze, synthesize, and package evidence about a technology's effectiveness [ST05].

SEP partially originated from ideas developed in the ESERNET [CW03] project. ESERNET aimed at supporting improvement of maturity and competitiveness of European software intensive organizations. For this purpose a web portal has been developed that provides empirical evidence regarding specific SE technologies, like inspections. This evidence was manually integrated into the underlying repository and there was no problem oriented search available. SEP enhances the concepts by automating the integration process of evidence into the repository and by providing goal-, problem-, and context-oriented search facilities. Several initiatives similar to ESERNET, have been implemented in the US (CeBASE), Scandinavia (e.g., Inter-Profit), Germany (VSEK), and Spain (VASIE).

4 Summary

The amount of methods, techniques, and tools in software engineering is increasingly rising. In order to keep track of the progress of these technologies and methodologies a software engineering manager needs an easy and fast access to evidence about them. In this paper we have presented ongoing research towards a semantic-based evidence portal for software organizations that help them to continuously gather and aggregate evidence about SE technology and methodology.

We argued that, as of today, repositories containing relevant information about evidence in SE must satisfy specific information needs of software managers in the task of technology selection. With regard to the information need of software managers we developed a database scheme that allows for answering the questions represented by the stated information need.

The next steps will include the empirical assessment of the semantic-based evidence portal in order to get more evidence about the effectiveness of the portal itself as well as feedback from the users. Furthermore, we are convinced that reporting guidelines for empirical work will support the comprehension of the available information both for managers and researchers. In addition, we will perform a survey among experts in the field to assess whether they find the information offered by the portal relevant and useful for their decision making.

References

[AJW03] Aurum, A.; Jeffery, R.; Wohlin, C.; Handzic, M. (Eds.): Managing Software Engineering Knowledge, Berlin: Springer-Verlag, 2003

[BCR01] Basili, V.R.; Caldiera, G.; Rombach, H.D.: Experience Factory. In: Marciniak JJ (ed.), Encyclopedia of Software Engineering, Vol. 1, pp. 511-519, John Wiley & Sons, 2001

[BH03] Biffl, S.; Halling, M.: A Knowledge Management Framework to Support Software Inspection Planning, in [AJW03], pp. 231-249

[Bir00] Birk, A.: A Knowledge Management Infrastructure for Systematic Improvement in Software Engineering; Ph.D. diss., Dept. of Computer Science, University of Kaiserslautern, Germany, Stuttgart: Fraunhofer IRB Verlag, 2000

[CW03] Conradi, R.; Wang, A.I.: Empirical Methods and Studies in Software Engineering – Experiences from ESERNET, Berlin: Springer-Verlag, LNCS 2765, 2003

[Gla04] Glass, R.L.: Matching Methodology to Problem Domain. In: Column Practical Programmer in Communications of the ACM /Vol. 47, No. 5, May 2004, pp. 19-21

[JC04] Jedlitschka, A.; Ciolkowski, M.: Towards Evidence in Software Engineering. In: Proc. Intern. Symposium on Software Engineering 2004 (ISESE2004), Redondo Beach, California, USA, August 2004, 2004, pp. 261-270

[JP03] Jedlitschka, A.; Pfahl, D.: Experience-Based Model-Driven Improvement Management with Combined Data Sources from Industry and Academia. In: Proc. Intern. Symp. on Empirical Software Engineering ISESE 2003, Roman Castles, Italy, October 2003, pp. 154-161

[JP04] Jedlitschka, A.; Pfahl, D.: Requirements of a Tool supporting decision making for SE Technology Selection; In Proc. of Intern. Conf. SEKE, Banff, Canada, 2004, pp. 513-516

[JP05] Jedlitschka, A.; Pfahl, D.: Reporting Guidelines for Controlled Experiments in Software Engineering. In: Intern. Symposium on Software Engineering 2005 (ISESE 2005), Noosa Heads, Australia, November 2005, pp. 95-104

[JPB04] Jedlitschka, A.; Pfahl, D. and Bomarius, F.: "A Framework for Comprehensive Experience-based Decision Support for Software Engineering Technology Selection", In Proc. of Intern. Conf. SEKE 2004, Banff, Canada, 2004, pp. 342-345

[MPR05] Münch, J.; Pfahl, D.; Rus, I.: Virtual Software Engineering Laboratories in Support of Trade-off Analyses. In: Software Quality Journal 13 (2005) 4, pp. 407-428

[Ruh03] Ruhe, G.: Software Engineering Decision Support: Methodology and Applications. In: Innovations in Decision Support Systems (Ed. by Tonfoni and Jain), International Series on Advanced Intelligence Volume 3, 2003, pp. 143-174

[ST05] Shull, F.; Turner, R.: An Empirical Approach to Best Practice Identification and Selection. In Proc. of ISESE 2005, pp. 133-140

[SH+05] Sjøberg, D.; Hannay, J.; Hansen, O.; By Kampenes, V.; Karahasanovic, A.; Liborg, N.-K.; Rekdal, A.: A Survey of Controlled Experiments in Software Engineering. In Trans. on SE, Vol. 31, 9, 2005, pp. 733-753

[VJB03] Vegas, S.; Juristo, N.; Basili, V.: A Process for Identifying Relevant Information for a Repository: A Case Study for Testing Techniques, in [AJW03], pp. 199-230