Towards Enterprise-Oriented Software Development Environments

Karina Villela
Guilherme Horta Travassos
Ana Regina Cavalcanti da Rocha

Federal University of Rio de Janeiro
COPPE - Programa de Engenharia de Sistemas e Computação
Caixa Postal 68511 CEP 21945-970 Rio de Janeiro, RJ – Brazil
Tel: (21)590-2552 R:334 Fax: (21)590-2552
E-mail: {kvillela, ght, darocha}@cos.ufrj.br

Abstract

Software developers need to make intensive use of knowledge throughout the software development process. There are different kinds of relevant knowledge in this context, such as domain knowledge, organizational guidelines, best practices, software techniques and methods, prior experiences using these techniques, methods and the software process. This knowledge is accumulated by the company's staff and can be useful in supporting software projects and organizational learning on software development. However, knowledge identification, organization, storage and usage are not trivial. To support the usage of such knowledge during software development we propose a new family of software development environments: the Enterprise-Oriented Software Development Environments (EOSDE). The EOSDE approach is based on both Knowledge Management (KM) and Experience Factory (EF) concepts. EOSDE are intended to support two types of Companies: Software Companies, in which the business activity is to develop software solutions for several clients, and Non-Software Companies, in which the software development activity aims at supporting the business activity.

Key-Words

Software Development Environment, Domain-Oriented Software Development Environment, Knowledge Management, Experience Factory, Enterprise-Oriented Software Development Environment

1. Introduction

Knowledge is produced and used during software development. Several reasons cause important and strategic knowledge in this context to be scattered in the minds of many people and/or documents stored in different media and places. Knowledge in people's minds is difficult to access and can be easily lost. Knowledge in documents is also difficult to access, share and update because it is often stored in private and unknown places. Both problems make it difficult or even impossible to identify previous mistakes when performing identical or similar tasks and to reuse pre-qualified solutions for the same problem. The consequence is failure to solve client's problems on time, loss of both productivity and quality and a considerable increase in software costs.

We understood the importance of this problem when developing software in partnership with several Brazilian companies. Different software teams deal with identical or similar problems, whose solution would benefit from previous experiences, but the companies lack systematic efforts to derive knowledge from prior experiences and/or new experiments and to apply this knowledge to company's projects.

In order to improve software development, software developers must have all relevant knowledge to perform their tasks easily available. Such knowledge includes domain knowledge, organizational guidelines, best practices, previous experiences with techniques, methods and the software process. This thesis proposes an infrastructure to support software developers during software development process by providing knowledge that has been acquired and improved by the company throughout time. Our goal is to provide this knowledge as part of a Software Engineering Environment originating what we call Enterprise-Oriented Software Development Environments.

2. From Domain-Oriented Software Development Environments to Enterprise-Oriented Software Development Environments

A Software Development Environment (SDE) is a computational system that provides support for the construction, management and maintenance of a software product. An SDE consists in a repository that stores all the information related to a software project throughout its life-cycle, and a set of tools that support technical and management activities.

Recent researches have emphasized the importance of building solutions for a specific domain [1,2,3]. Domain-Oriented Software Development Environments (DOSDE) [4] is an extension of the traditional SDE that includes knowledge of a specific domain as the main factor for assistance during the software process. Thus, besides a repository and a set of tools, DOSDE also require two essential features: (a) the existence of domain knowledge, and (b) the usage of this knowledge during software development.

However, we have realized that, besides the domain knowledge, other kinds of knowledge are also necessary and useful during a software development project. This was the motivation for defining Enterprise-Oriented Software Development Environments, which has the following goals: (a) to provide software developers with all relevant knowledge for software development accumulated by the company, and (b) to support organizational learning about software development. To accomplish these goals our research is based on two related research areas: Knowledge Management (KM) and the Experience Factory (EF).

The goal of KM is to get knowledge into the organizational level, to improve the flow of knowledge within the company and to promote new knowledge [5,6], recognizing that knowledge is one of the most important assets of a company [7]. Some activities have been proposed to deal with KM [5,6,7]. ABECKER *et al.* [7] have identified the elements of knowledge infrastructure: *Organizational Memory* and *Knowledge Management Services*. In

this context, knowledge bases and ontologies are underlined as important technologies. Knowledge bases represent the content of the *Knowledge Management System* and ontologies define shared vocabulary to allow knowledge communication, search, storage and representation [8].

The EF approach [9] defines an environment for controlled experimentation, knowledge reuse, development processes analysis and experience packaging. The environment consists of both the Project Organization, which focuses on delivering the software product according to defined requirements, and the Experience Factory Organization, which focuses on learning from experience as well as on the improvement of software development practices in the organization [10]. The retrieval of similar software engineering experience is one of the main research subjects of this approach [11,12].

EOSDE aim at applying the concepts of KM and the EF approach to SDE. To construct this kind of SDE we will use the infrastructure of TABA workstation [13], a long-term project developed at COPPE/UFRJ intended to assist in the definition and instantiation (construction) of SDE. TABA workstation has already been extended to instantiate Domain-Oriented Software Development Environments (DOSDE) [4]. By using TABA workstation in this new project we will take advantage of the knowledge infrastructure already used in DOSDE.

3. Types of Companies

Companies involved in the development and maintenance of software hold different kinds of knowledge and have different goals for this activity. In the context of this project we have classified the companies into *Software Companies* and *Non-Software Companies.*

Software Companies (such as software houses or software factories) have as their business activity the development of software for several different clients. Their goal is a combination of client's satisfaction, operational excellence and/or product leadership. They must use and master many different software engineering tools, techniques, methods and process models to meet the requirements of their current and potential clients. These companies can accumulate software development experiences in several different domains. However, when a novel project dealing with an application domain is to be developed, the necessary domain knowledge is not usually available in the company.

In *Non-Software Companies* (such as hospitals, banks and telecommunications companies), the software development activity aims at supporting the business activity. These companies develop software for their own usage and only need to be knowledgeable on those software engineering tools, techniques, methods and process models that are commonly used in the company. As time goes by, they can accumulate experience in both their specific domain and the software engineering practices used. When a new software engineering project begins, the necessary domain knowledge is normally available in the company.

4. Enterprise Oriented Software Development Environments: an initial proposal for its requirements and infrastructure

The identification of these two kinds of companies that deal with software development led us to the definition of two different kinds of EOSDE with specific requirements and infrastructure. However, there are also some requirements and aspects of the infrastructure that are common to all EOSDE. In this section, the common requirements and infrastructure for EOSDE are presented first. Then, the requirements and aspects of the infrastructure that are specific for each kind of EOSDE are discussed.

4.1 EOSDE: Common Requirements and Infrastructure

In order to accomplish its goals, EOSDE must meet the following requirements: (i) provide the representation of the organizational structure and make it easy to find the experts whose knowledge and experience can be useful for a certain purpose; (ii) store specialized knowledge about software development and maintenance, and supply this knowledge for the project team when necessary, and (iii) support the continuous evolution of the knowledge in the SDE.

If the project team has access to the organizational structure, it will be easier to find the experts who can contribute to their project. This is the reason for including requirement (i). To address requirement (ii), EOSDE should include knowledge about software engineering activities that are always carried out regardless of specific clients or projects, which constitute the software process of the company. Moreover, EOSDE should contain knowledge about the experience on software development acquired by the company, which includes: standards and guidelines, updated news about the technologies used, best practices related to the software process, methods and languages, analysis of software engineering tools, deliverables with potential for reuse, lessons learned from previous projects and organizational performance measures. Knowledge and experience evolve with time and thus it is important to support the evolution of knowledge in EOSDE (requirement (iii)).

In order to meet these requirements, we propose an architecture for the infrastructure of knowledge in an SDE (Figure 1). Each component of this infrastructure has its own goals and contains important knowledge. The *Task Description* component contains the description of generic tasks. A task description consists of a high-level description, a task ontology and bibliographic references. The *Task Ontology* contains concepts and attributes associated with the generic tasks. The *SE Theory* contains the *SE Ontology*, which defines a common vocabulary to guide the registration/distribution of software engineering knowledge by the EOSDE and the design of the company's map of knowledge. The *Company Description* contains a description of the company, identifying the generic tasks that are performed and the software engineering knowledge necessary in the context of the organizational structure and company processes. The main component of the *Company Description* is the *Company Ontology*, which provides concepts and attributes related to the organizational structure and company processes. The *Information Ontology* involves all the information and aspects of knowledge that are not specific to content. *Knowledge and Data Bases* store the knowledge and data acquired through many software engineering projects and knowledge/data about the organizational structure. Finally, the *Knowledge Management Services* allow the storage of knowledge in the organizational memory and the dissemination and evolution of the stored knowledge. Some of these components are mentioned by authors in related works [4,7,14].

4.2 EOSDE for Software Companies: Requirements and Infrastructure

Software Companies usually have clients in several application domains. Holding knowledge (even if only partially) about these domains can mean a strategic advantage in the competition for new projects. In addition to the common requirements discussed above, we have identified specific requirements of EOSDE for Software Companies: (i) to provide the representation of the organizational structure of their clients to allow easier identification of useful experts for a certain purpose in the clients' companies, and (ii) to store clients' domain knowledge and to supply the development team with this knowledge when necessary through a software project.

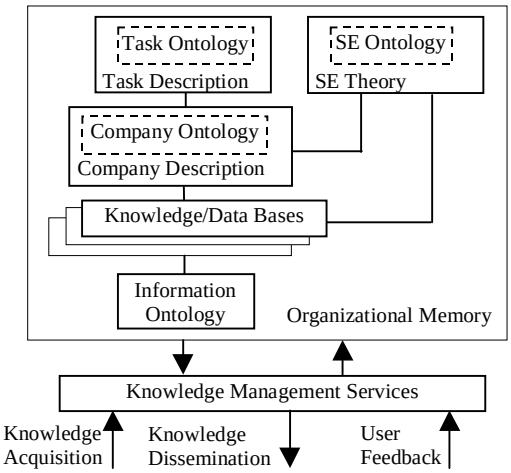

Figure 1 - Common Knowledge Infrastructure

Figure 2 presents the architecture we have devised through the addition of the *Client's Company Model* to the *Common Knowledge Infrastructure*. The *Client Description* describes the client's company, identifying the generic tasks that are performed and the domain knowledge used in the context of the organizational structure and the company processes. The *Client Ontology* provides concepts and attributes related to the organizational structure and the processes of the client's company. The *Domain Theory* organizes the domain knowledge by using the *Domain Ontology* and identifying generic tasks related to the domain. *Client Knowledge and Data Bases* store knowledge and data about the organizational structure of the client's company.

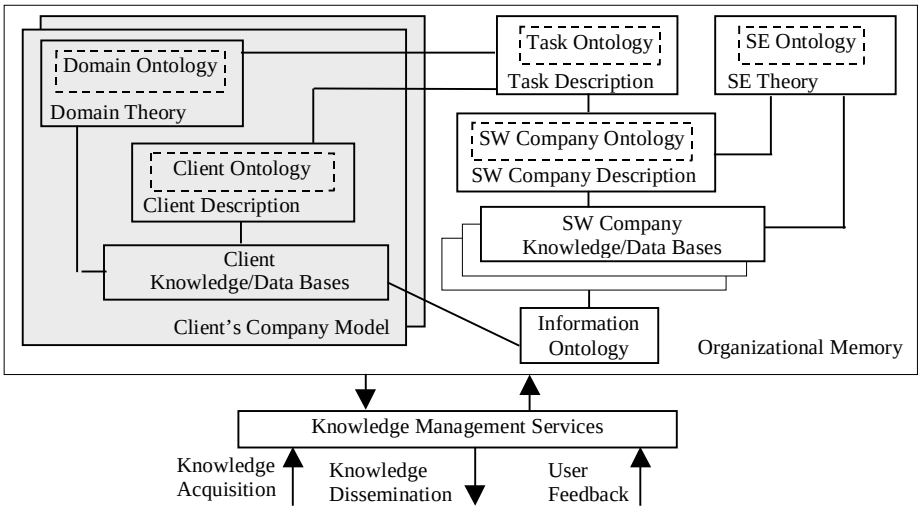

Figure 2 - Knowledge Infrastructure of EOSDE for Software Companies

4.3 EOSDE for Non-Software Companies: Requirements and Infrastructure

EOSDE for a *Non-Software Company* can be seen as an extension of DOSDE. DOSDE as proposed by OLIVEIRA [4] aim at supporting software development in a specific domain. Thus, the requirements discussed in section 4.1 were added to the requirements previously defined for DOSDE. The infrastructure is the same as the *Common Knowledge Infrastructure* with the addition of the *Domain Theory* as a new component, which has interfaces with the *Task Description, Company Description* and *Knowledge/Data Bases* components.

5. Research Methodology

The work on this thesis proposal is just beginning as it was approved on March 2000. After the definition of the infrastructures of EOSDE and their requirements, we started our research, which has been carried out in 6 steps. The first step intends to define what kind of knowledge should be stored in EOSDE. This step includes bibliographic and field research. Field research will be conducted in some companies and it also intends to identify when a certain kind of knowledge is needed during the software development process and how to evaluate the quality of acquired knowledge. In the second step, we will define the knowledge management process and its relation with the software development process. Our third step aims at assessing the requirements and the infrastructure proposed in this paper and then defining the architecture of EOSDE. In the fourth step, our goal is to define the methodologies to be used for knowledge acquisition. The fifth and sixth steps will be respectively to build EOSDE, by using the infrastructure of TABA workstation [4,13], and to perform case studies.

We know that a lot of work has to be done in order to make EOSDE a reality, but we do believe that, by using the infrastructure that we have defined, we will offer an important tool to assist software developers in performing their tasks, improving software quality and productivity.

References

[1] TAYLOR, R. *et al.*, 1995, "Software Development Using Domain-Specific Software Architectures", *Software Engineering Notes*, v. 20, n. 5 (Dec), pp. 27-38.

[2] FISCHER, G, 1994, "Domain-Oriented Design Environments", Automated Software Engineering - The International Journal of Automated Reasoning and Artificial Intelligence in Software Engineering, v. 1, n. 2 (Jun), pp. 177-203.

[3] GOMAA, H. et al., 1996, "A Knowledge-Based Software Engineering Environment for Reusable Software Requirements and Architectures", *Automated Software Engineering*, v. 3, pp. 285-307.

[4] OLIVEIRA,K, MENEZES, C., TRAVASSOS, G.H, ROCHA, A, R, 2000 Ambientes de Desenvolvimento de Software Orientados a Domínio, XIV Simpósio Brasileiro de Engenharia de Software, João Pessoa, PB, Out.

[5] HENDRIKS, P., VRIENS, D., 1999, "Knowledge-based systems and knowledge management: Friends or foes?", *Information & Management*, v. 35, n. 2 (Feb), pp. 113-125.

[6] SKYRME, D., 1998, "Knowledge Management Solutions - The IT Contribution", *SIGGROUP Bulletin*, v. 19, n. 1 (Apr), pp. 34-39.

[7] ABECKER, A. *et al.*, 1998, "Toward a Technology for Organizational Memories", *IEEE Intelligent Systems*, v. 13, n. 3 (May/June), pp. 40-48.

[8] O'LEARY, D. E., 1998, "Using AI in Knowledge Management: Knowledge Bases and Ontologies", *IEEE Intelligent Systems*, v. 13, n. 3 (May/Jun), pp. 34-39.

[9] BASILI, V., 1989, "The Experience Factory: Packaging Software Experience". In: *14th Annual Software Engineering Workshop*, SEL-Software Engineering Laboratory, NASA, USA.

[10] SEAMAN, C. *et al.*, 1999, "An Experience Management System for a Software Consulting Organization". In: *24th Annual Software Engineering Workshop*, SEL-Software Engineering Laboratory, NASA, USA.

[11] BROOMÉ, M., RUNESON, P., 1999, "Technical Requirements for the Implementation of an Experience Base". In: *11th International Conference on SE & KE*, pp. 1-9, Kaiserslautern, Germany, Jun.

[12] WANGENHEIM, C. et al., 1999, "Intelligent Retrieval of Software Engineering Experienceware". In: *11th International Conference on SE & KE*, pp. 128-135, Kaiserslautern, Germany, Jun.

[13] TRAVASSOS,G.H.,1994 "O Modelo de Integração de Ferramentas da Estação TABA"; PhD Thesis, COPPE/UFRJ

[14] ALTHOFF, K. *et al.*, 1999, "Managing Software Engineering Experience for Comprehensive Reuse". In: *11th International Conference on SE & KE*, pp. 10-19, Kaiserslautern, Germany, Jun.