A Hybrid Approach for Corporate Memory Management Systems in Software R&D Organizations

Christiane Gresse von Wangenheim[1], Daniel Lichtnow[2,3], Aldo von Wangenheim[2]

[1]Universidade do Vale do Itajaí, São José, Brazil
gresse@sj.univali.br

[2]Universidade Federal de Santa Catarina, Florianópolis, Brazil
awangenh@inf.ufsc.br

[3]Universidade Católica de Pelotas, Pelotas, Brazil
lichtnow@atlas.ucpel.tche.br

Abstract

The ability to systematically manage knowledge contributes to the success of Software Research and Development organizations. Based on our experiences, we propose a hybrid approach for a technical infrastructure on knowledge management for software R&D organizations. The approach integrates various types of information and knowledge and provides intelligent mechanisms for knowledge access, as well as the continuous evolution and improvement of the Corporate Memory Management System throughout its life cycle. The principal strength of the approach lies in the integration of techniques from various areas, such as Case-Based Reasoning, Information Retrieval and Natural Language Processing into one infrastructure creating a comprehensive intelligent assistant for software research and development. The approach is currently being implemented and evaluated in the context of an international research project.

Keywords: knowledge management, organizational learning, corporate memory.

1 Introduction

Software Research and Development (R&D) organizations aim at developing complex software products and services outstanding in terms of innovation and creativity. Typically, they are composed of multiple interacting communities, each characterized through highly specialized knowledge. Therefore, one important success factor of software R&D organizations is their ability to manage knowledge regarding their experience, understandings, know-how and skills. Researchers need to have sound theoretical knowledge, practical experiences and skills in various areas, including the application domain (e.g., computed tomography of the human brain), computer science technologies (e.g., image analysis algorithms), software engineering techniques (e.g., to design a system using UML), as well as on research methodologies (e.g., on how to plan the research work). The importance of systematically managing relevant knowledge is further emphasized in environments characterized through locally distributed groups, constant staff turn-over, varying levels of knowledge and capabilities, and limited resources.

In this context, knowledge management (KM) focuses on how organizations can understand what they know, what they need to know and how to make maximum use of the knowledge [10,18,25]. It enables the consolidation of know-how and skills into competencies, shortening the learning curve for new technologies and empowering the organization to adapt faster to changes and challenges [20]. However, today KM is generally done in an ad-hoc, informal manner, where the decision to reuse is made by individuals and the type of experience reused is usually limited to personal experiences.

As a logical and physical structure for the continuous build-up of know-how in a software organization, the *Experience Factory* approach [6] has been proven to be a successful solution. It proposes an infrastructure for analyzing and synthesizing all kinds of experiences, acting as a repository for those, and supplying these experiences to projects on demand. To operationalize this approach in practice, a technical infrastructure, denoted *Corporate Memory Management System* (CMMS), is required. A CMMS includes *Corporate Memories* (CM) for the storage of information and knowledge assets, as well as tools to manage these knowledge-bases. The tools should support the rapid and effective access to the right information or knowledge, the continuous acquisition of new experiences and their integration and storage in the existing CM, as well as the continuous adaptation and maintenance of the CMMS to the specific environment.

In this paper, we present a hybrid approach for a CMMS tailored to software R&D organizations in order to provide an operational platform for an intelligent KM assistant[1]. The approach integrates technologies from different areas, includ-

ing database, hypertext and e-mail systems, Information Retrieval and Case-Based Reasoning. The approach is currently being implemented and evaluated in the context of the *Cyclops Project* [9], a German-Brazilian research project which aims at the development and transfer of new methods, techniques and tools in the field of Telemedicine and Medical Image Analysis.

2 Requirements for a Technical Infrastructure

The development of an effective and efficient CMMS in software R&D organizations is not trivial. Based on our experiences in the Cyclops Project and literature [2,7,11], the following requirements have been derived:

Multimodal support. The CMMS has to provide manifold support enabling the access to various types of information or knowledge (e.g., documents, WWW sites) for various purposes (e.g., facilitate a research on the state-of-art or guide the solution of a programming problem), from different viewpoints (e.g., computer scientist, medical researcher).

Efficient and effective access to useful assets. Intelligent techniques are required enabling the similarity-based retrieval of useful assets without overloading the user with irrelevant information. The mechanisms have to allow the formulation of queries in natural language (e.g., in Portuguese and English in case of the Cyclops Project) and enable multilingual retrieval (e.g., searching also for English assets for queries formulated in Portuguese).

Pro-active distribution of knowledge. Besides enabling the retrieval of assets, the CMMS should also support the proactive recommendation of knowledge wrt. to an user´s specific interests (e.g., informing about new published papers wrt. her/his research area).

Continuous evolution of corporate memory. In the context of a software R&D organization, relevant information and knowledge is not completely available when creating the CM. Therefore, KM in software R&D organizations requires support for the continuous evolution of the knowledge base and has to be able to deal with incompleteness and inconsistency. This includes the continuous acquisition of new assets as integrated part of the research and development activities (i.e., reporting a FAQ on an recently occurred programming problem) and the indexing and integration of the new acquired assets into the existing CM.

Maintenance of the CM. As the software domain is characterized through continuous changes and technology advances, the maintenance of the CMMS becomes especially important. This includes support for the maintenance of the knowledge assets, the general domain knowledge, as well as the improvement of the knowledge access mechanisms.

Intelligent assistant. Due to its complexity, the integration of human experts in the KM process is indispensable. This involves the insertion of new assets by research group members, the review of new acquired assets by experts and the maintenance of the CM and CMMS tools by a knowledge engineer. Therefore, the CMMS has to be designed to be an intelligent assistant which facilitates the KM tasks.

These requirements show that sophisticated and comprehensive support is required for the development of a CMMS in a software R&D organization.

3 State of the Art and Practice

Today, a wide range of Information Technologies is used to implement CMMSs in general [3], as well as specifically in the software domain [21]. This includes, for example:

- **Database Management Systems** (DBMS), which handle large amounts of data (e.g., [5]), but without providing specific KM capabilities.
- **Knowledge maps**, which guide the localization of knowledge instead of containing the knowledge itself (e.g., [16]).
- **Group support systems**, which provide communication links among members and manage documents and historical records of decisions (e.g., [17]).
- **Hypertext systems**, which enable the navigation through documents via links (e.g., [13]).
- **Information Retrieval** (IR) [23], which provides a generic indexing and retrieval engine.
- **Case-Based Reasoning** (CBR), which provides support for the development of learning knowledge-based systems (e.g., [12]) by focusing on experiential knowledge and enabling similarity-based retrieval.
- **Information filtering** used to sort information and to make customized recommendations (e.g., [15]).

However, in order to develop a CMMS that truly contributes to the effectiveness for KM in software R&D organizations, various approaches need to be integrated (e.g., [14,22]). Yet, none of the existing approaches completely fulfills the requirements described in Section 2 for a CMMS in a software R&D organization (e.g., regarding the incorporation of various types of knowledge or multilingual retrieval).

4 A Hybrid Approach to a CMMS

In this section, we present a hybrid approach for a CMMS in the context of a software R&D environment. The objective is to create a technical infrastructure that enables storage and access to various types of information and knowledge in the CM and the continuous evolution and maintenance of the CMMS. The approach integrates technologies from different areas: CBR techniques are used for knowledge representation, similarity-based retrieval and incremental learning. IR and NLP techniques are the basis for information extraction from natural language queries and documents, as well as the evolution of general domain knowledge. Hypertexts are used to enable the interactive exploration of assets suggested by the CMMS. Information filtering techniques are used to recommend assets of potential interest to the user.

In the following sections, we describe each of the principal components of the CMMS in detail.

1. Managerial KM aspects are beyond the scope of this paper.

4.1 Knowledge Representation

A main goal of the CMMS is to capitalize existing information and knowledge. This includes sources which are explicitly available in the organization, i.e., documents, as well as the externalization of tacit knowledge from personal experiences (e.g., by writing notes on solution strategies). Due to its concrete and experiential character, the information and knowledge in the CM is represented in form of cases, denoted as *CM assets*. This includes various types:

- Documents records constituting a personalized library, recording referential information and comments, as well as allowing the up-/download of electronic documents.
- Frequently Asked Questions (FAQ) stating frequently asked questions and their answers provided by an expert.
- How-to-do recipes describing step-by-step how frequently occurring tasks have to be done.
- WWW maps listing and commenting relevant Web sites.
- Yellow pages indicating human-resource capabilities.
- Starter´s kits summarizing information or knowledge important to a beginner wrt. a specific research area.
- News messages communicating any news of relevance to the research organization.
- Software parcels packaging and commenting software code or executables.

The information and knowledge represented in the CM cover all relevant research areas wrt. the specific R&D organization, such as application domain (e.g., radiology, anatomy), computer science and software engineering (such as image interpretation algorithms, software process model), as well as research methodologies (e.g., literature study). The various types of CM assets are modeled in a CMMS domain model using a flexible, object-oriented frame-like representation formalism. The CM assets are indexed by indicators on the content of the asset as a basis for efficient retrieval.

Besides the CM assets, general domain knowledge is represented defining terminology and basic concepts, including:

- **vocabularies**: representing indicative expressions for a predictive indexation of CM assets in the specific domain. For example, the vocabulary on the programming language Smalltalk includes, the terms “class”, “collection”, etc.
- **thesauri**: indicating similar terms wrt. associative or hierarchical relations in the given domain. For example, in the context of the programming language Smalltalk the terms “class” and “object” are considered as synonyms.
- **bilingual dictionaries**: indicating the translation of domain-specific terms. In our specific application, we focus on Portuguese-English dictionaries, including, for example, “class -> classe”.

For each of the research areas of interest, a domain vocabulary, thesaurus and dictionary is developed and stored in the CM. In addition, general vocabularies and normalization rules on the Portuguese and English language are used for the interpretation of natural language queries.

4.2 Knowledge Access

The primary objective of the CMMS is to (re-)use the information and knowledge stored in the CM. Therefore, an effective and efficient access to useful CM assets is essential for the success of the CMMS. We distinguish here between retrieval and distribution, depending on, if the information or knowledge is delivered actively or passively. In case of retrieval, a researcher consults the CMMS trying to find CM assets of interest. In case of distribution, CM assets of potential interest are automatically distributed to researchers.

4.2.1 Retrieval of CM Assets

Based on a query formulated by the user, the CM is searched and the most relevant CM asset(s) are returned. The result is presented as a hypertext listing the most relevant CM asset(s) and enabling the user to explore them in detail. If the CMMS does not return a satisfactory result, the user can automatically direct her/his query to a domain expert via e-mail. Once the answer from the expert is available, it is automatically forwarded to the user and, in addition, a new CM asset is created and stored in the CM. In order to allow the access of several types of information and knowledge for different objectives, the CMMS offers various search and retrieval techniques. When searching by navigation, the user specifies a research area of interest (e.g., “Smalltalk”) and as result all CM assets related to the specified area are returned as a hypertext classified per type of asset (e.g., document or FAQ). Then, the user can explore the individual CM assets through browsing. When searching by attributes, the user specifies value(s) for certain attributes of the CM asset to be retrieved (e.g., “Author: José Silva”). The search returns a hypertext listing all CM assets which perfectly match the specified attribute values. When searching by content [8], the user enters a question in natural language or arbitrary search terms (e.g., “What is a controller?”). The query is analyzed, including also spelling correction and normalization and relevant indexes are automatically extracted. Then, the query indexes are partially matched with the assets in the CM applying global and local similarity measures [1]. According to their degree of similarity to the given query, a partial order is induced among the assets in the CM and the most similar asset(s) are presented to the user as result of the retrieval process. In addition, by specifying certain attributes (e.g., “research area: Smalltalk”) the search can be limited to CM assets which perfectly match these attribute values. When searching by examples, the user indicates a CM asset as an example for what s/he is looking for. By obtaining the respective attribute values of the indicated asset, a query is automatically constructed. As in case of content search, all CM assets, which are sufficiently similar to the query, are returned.

As the assets stored in the CM, as well as the queries may be expressed in more than one language (e.g., in Portuguese or English), a multilingual retrieval technique [24] based on bilingual domain dictionaries (also denoted as multilingual thesauri [24]) is integrated into the CMMS.

4.2.2 Distribution of CM Assets

A knowledge distribution approach is integrated into the CMMS, which does not depend on a user actively searching

for information, but which pro-actively provides assets of potential interest to the user's long-term information goals. For making recommendations, we use three different approaches [19], including demographic filtering which is used to identify types of users that are interested in a certain topic, content-based methods, which make recommendations based on CM assets that have been of interest to the user in the past and collaborative approaches, which recommend CM assets that have been reused by other users with a similar profile.

4.3 Continuous Collection and Integration

As the CM has to evolve continuously, support has to be provided for the collection and quality validation of new assets, their appropriate indexing and integration into the CM.

4.3.1 Collecting CM Assets

The objective of collecting CM assets is to capture new or improved information or knowledge of relevance. This includes explicit knowledge and tacit knowledge [18]. Explicit knowledge can be found in the documents of the organization: articles, manuals, etc. Supplementary information is added (e.g., indicating the relevance of a certain asset) in order to provide further guidance for the (re-)use of the asset. Tacit knowledge is externalized by writing how-to-do recipes (e.g., on how to implement a database connection). The collection of new assets is as much as possible intertwined into the existing processes, this includes, for example, the spontaneous collection (e.g., when discovering a new WWW site), event-driven collection (e.g., when a new scientist joins the group), planned collection (e.g., when initially creating a FAQ system) or retrieval-based collection (e.g., by capturing a manually answered question as a new FAQ). The collection is done via on-line forms (e.g., for the registration of a new document) which also enable the upload of document files.

4.3.2 Indexing CM Assets

The inclusion of new CM assets into the existing CM, requires their appropriate indexing for retrieval and distribution. The indexing process is based on the CMMS domain model aiming at the instantiation of the respective attributes defined for the specific type of asset (see Section 4.1). The indexes are set based on information provided by the collector (e.g., author, title etc. of a document) and by semi-automatically extracting information from the respective asset (e.g., indicative expressions of a FAQ question) using domain specific vocabularies. The result is revised and, if necessary, improved by a domain expert.

4.3.3 Evaluation of CM Assets

Once created, the new CM asset is revised by a domain expert and the knowledge engineer wrt. to its quality and its reuse potential. The focus of the expert is on correctness, completeness, understandability and relevance of the asset, whereas, the knowledge engineer focuses the newness of the asset, completeness, and consistency. Depending on the result of the evaluation, the CM asset may be accepted with or without modifications, or rejected. Once accepted the CM asset is published and made available for reuse in the CMMS.

4.4 Maintenance of the CMMS

The maintenance of the CMMS encompasses the revision and adaptation of the assets and knowledge stored, as well as the access and collection mechanisms. As in general, most of the maintenance activities have to be done manually by the knowledge engineer (supported by domain experts), the objective of the technical infrastructure is to provide support through the pre-processing of available input data.

Maintenance of CMMS domain model. The CMMS domain model is initially developed based on a domain analysis. During its application in practice, the CMMS domain model is periodically revised wrt. to new upcoming research areas or asset types. If necessary, the model is appropriately adapted or enhanced by the knowledge engineer.

Maintenance of CM assets. The initial CM is created based on available assets in the organization, e.g., documents or FAQs. In order to keep the CM up-to-date, new CM assets are continuously collected and integrated (see Section 4.3). In addition, assets may be manually generalized (e.g., if two FAQs provide an answer to the same question) and outdated assets may be deleted by the knowledge engineer (e.g., on technologies not longer of interest to the organization).

Maintenance of general domain knowledge. The general domain knowledge is initially developed through a domain analysis and updated each time a new asset is included into the CM. In order to keep the required effort for the identification of new vocabulary terms as low as possible, candidate terms are identified based on the inverse document frequency measure [23], which induces a partial order among the terms depending on their number of appearances in a specific document and among all documents of the CM. Once, new relevant terms have been confirmed by the domain expert, the domain thesauri and dictionaries have to be updated appropriately. The inclusion of new terms into the domain thesauri is supported by automatically generating a partially order list of potentially related terms based on the statistical co-occurrence measure [23]. Based on these suggestions, the expert can modify or add new term relations.

Maintenance of user profiles. For each user, a profile is defined as a basis for the distribution of CM assets of potential interest, including, e.g., the assets reused by the user. This profile is set up initially by categorizing the user's areas of interests. The content-based indexes of the user profile are updated continually based on the analysis of indexes of queries and CM assets that have been reused by the user. In order to enable recommendations based on collaborative filtering methods, a historical record is stored on which assets have been visited and/or reused.

Evaluation and improvement of knowledge access and collection mechanisms. The knowledge access and collection mechanisms have to revised and adapted during the whole life cycle of the CMMS. This is done manually by the knowledge engineer, based on feedback from the application of the CMMS in practice concerning its performance and user acceptance. Feedback is collected through usage proto-

cols, which track usage patterns, as well as user profiles and interviews with the users and experts. Based on the identified improvement opportunities, the CMMS has to be appropriately adapted. For example, when observing that distributed assets are frequently not reused, the user profile mechanisms have to be revised and appropriately modified.

5 Tool Architecture

The CMMS is implemented based on a 3-layer client-server architecture consisting of interface, application, and storage layer. In the system, CM assets are stored in a Database Management System. General domain knowledge and document files are stored in a file system. Core parts of the application layer of the CMMS are the retrieval/distribution, collection and maintenance tools. The communication with the user is realized via Intranet, through web browsers and e-mail systems implementing the connection to the application layer via HTTP servers. So far, the retrieval (search by navigation and attribute search) and the collection of documents and WWW sites of various research areas in the Cyclops project have been implemented. Mechanisms for the management of FAQs on the programming language Smalltalk including content-based search of Portuguese natural language queries have been implemented, as well as techniques for the maintenance of the general domain knowledge (including Portuguese domain vocabulary, dictionary and thesaurus on Smalltalk issues).

6 Conclusion

In this paper, we present an approach for the development and maintenance of a technical infrastructure for KM in software R&D organizations. In comparison to other available KM tools, the approach is tailored to completely fulfill the requirements of software R&D organizations by considering a broader scope of information and knowledge represented and integrating advanced retrieval, information filtering and natural language processing techniques. The approach shows how techniques applied in KM can become more powerful by their integration into one infrastructure creating a comprehensive intelligent assistant for software research and development activities. The continuous and systematic collection of information and knowledge enabled through the CMMS is further expected to promote learning on an organizational level, building up software research and development competencies. Currently, we are implementing and applying the approach in the research project Cyclops. Based results of an evaluation of its performance and perceived usefulness, we intend to continue the implementation of the approach and to broaden the scope of research areas covered.

Acknowledgments

The authors would like to thank A. Bortolon, D. D. Abdalla, E. M. Barros, P. Dellani and F. Secco for their support.

References

[1] A. Aamodt, E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, 17(1), 1994.

[2] K.-D. Althoff et al. Systematic Population, Utilization, and Maintenance of a Repository for Comprehensive Reuse. In G. Ruhe, F. Bomarius (eds.) Learning Software Organizations. Springer Verlag, 2000.

[3] A. Abecker, S. Decker. Organizational Memory. Proc. of the Workshop on Expertsystems, Germany, 1999.

[4] M.S. Ackerman, T. W. Malone. Answer Garden: A Tool for Growing Organizational Memory. Proc. of the Conference on Office Information Systems, Cambridge, MA, 1990.

[5] V. R. Basili et al. The Software Engineering Laboratory - An Operational Software Experience Factory. Proceedings of the 14th International Conference on Software Engineering, 1992.

[6] V. R. Basili, G. Caldiera, H. D. Rombach. Experience Factory. In John J. Marciniak, ed., Encyclopedia of Software Engineering, vol.1, pp. 528-532. John Wiley & Sons, 1994.

[7] A. Birk, F. Kröschel. A Knowledge Management Lifecycle for Experience Packages of Software Engineering Technologies. G. Ruhe, F. Bomarius (eds.) Learning Software Organizations. Springer Verlag, 2000.

[8] A. Bortolon. Desenvolvimento e Implementação de uma Abordagem Hibrida para a Gerência de Documentos FAQ em Portugues. Master Thesis, Federal University of Santa Catarina, 2001.

[9] The Cyclops Project (www.inf.ufsc.br/cyclops)

[10] T. H. Davenport, L. Prusak. Working Knowledge. Harvard Business School Press, 1998.

[11] C. Gresse von Wangenheim et al. Evaluation of Technologies for Packaging and Reusing Software Engineering Experiences. IESE-Report No. 055.98/E, Fraunhofer Institute for Experimental Software Engineering, Germany, 1998.

[12] S. Henninger. Capturing and Formalizing Best Practices in a Software Development Organization. Proc. of the 9th Int. Conf. on Software Engineering & Knowledge Engineering, Spain, 1997.

[13] F. Houdek, H. Kempter. Quality Patterns - An Approach to Packing Software Engineering Experience. Software Engineering Notes, 22(3), 1997.

[14] IKnow (www.knowlix.com/products/iknow.htm).

[15] B. Krulwich, C. Burkey. The Information Finder Agent. IEEE Expert, Sept.-Oct. 1997.

[16] KnowledgeX (www-3.ibm.com/software/swprod).

[17] Lotus Notes (www.lotus.com).

[18] I. Nonaka, T. Takeuchi. The Knowledge-Creating Company. Oxford University Press, Cambridge, UK, 1995.

[19] M. J. Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review, 1999.

[20] C.K. Prahalad, G. Hamel. The Core Competence of the Corporation. Harvard Business Review, 68(3), May 1990.

[21] G. Ruhe, F. Bomarius (eds.) Learning Software Organizations, Springer Verlag, 2000.

[22] RetrievalWare.(www.excalib.com/products/rw/index.shtml).

[23] G. Salton, M. J. McGill. Introduction to Modern Information Retrieval. New York: McGraw Hill, 1983.

[24] D. W. Oard, B. J. Dorr. A Survey of Multilingual Text Retrieval. UMIACS-TR-96-19, University of Maryland, 1996.

[25] K. M. Wiig. Knowledge Management : Where did it come from and where will it go?. Expert Systems with Applications, 13(1), 1997.