SRS: Semantic Reuse System

Bruno Antunes  
KIS - CMS - CISUC  
University of Coimbra  
Portugal  
bema@dei.uc.pt

Nuno Seco  
KIS - CMS - CISUC  
University of Coimbra  
Portugal  
nseco@dei.uc.pt

Paulo Gomes  
KIS - CMS - CISUC  
University of Coimbra  
Portugal  
pgomes@dei.uc.pt

André Costa  
KIS - CMS - CISUC  
University of Coimbra  
Portugal  
accosta@student.dei.uc.pt

Pedro Oliveira  
KIS - CMS - CISUC  
University of Coimbra  
Portugal  
pcoliv@student.dei.uc.pt

ABSTRACT
One of the problems that software development companies face today is the increasing dimension of software development projects, which have grown in size and complexity, as well as in the number of technologies that are involved. The knowledge generated during the software development process can be a valuable asset for a software company, in order to take advantage of this knowledge the company must acquire, store and manage it for reuse. This paper describes the Semantic Reuse System (SRS), a system for management and reuse of software development knowledge based on the Semantic Web technologies.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Management; D.2.13 [Software Engineering]: Reusable Software; I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods; I.2.7 [Artificial Intelligence]: Natural Language Processing

General Terms
Management, Languages and Standardization.

Keywords
Knowledge Management, Semantic Web, Ontologies, Software Development.

1. INTRODUCTION
The new vision of the web, the Semantic Web [1], aims to turn the web more suitable for machines, thus making it more useful for humans. It brings mechanisms that can be used to classify information and characterize it's context. These mechanisms enable the development of solutions that facilitate the access and exchange of relevant information, which are central issues in knowledge management [2]. These technologies provide knowledge representation structures that can be used to improve the storage and retrieval functionalities of common knowledge management systems, facilitating inter-operability between knowledge containers at the same time. One of the problems that software development companies face today is the increasing dimension of software systems. The resources produced during the design and implementation phases of such projects, are an important source of knowledge inside software development companies. In order to make this knowledge useful and easily shareable, knowledge management systems that efficiently capture, store, manage and retrieve all kinds of knowledge produced in the development of software systems are needed.

In this paper we describe the Semantic Reuse System (SRS). The goal of SRS is the acquisition, management and reuse of software development knowledge in the form of what we call Software Development Knowledge Elements (SDKE's). As software development knowledge, we consider the various elements that result from the software development process, such as specification documents, design diagrams, source code, etc. The system makes use of the mechanisms provided by the Semantic Web, such as RDF, RDFS and OWL [3], to represent the knowledge used. The remaining of the paper starts with an overall description of the system. Then we present the ongoing and future research work on SRS, and conclude with some final remarks.

2. SEMANTIC REUSE SYSTEM
The SRS platform provides a way to capture, manage and reuse software development knowledge in the form of SDKE's. Besides being stored in the file system and accessible through a framework, the SDKE's are described using a Representation Ontology and classified through a Domain Ontology. These knowledge structures are used to empower the retrieval mechanisms, thus making easier the reuse of the stored knowledge. The Representation Ontology models and describes the different types of SDKE's managed by the system. The Domain Ontology is used to represent the knowledge of the domain that the system supports, having concepts that are referenced by words and relations between concepts. The SDKE's stored in the platform are indexed to
An evolution of SRS taking into account these requirements and permissions was not included in the first version of SRS. The notion of projects that have associated resources, users and documents stored in the VCS repository is then stored and indexed in SRS for analysis.

3. ONGOING RESEARCH

The first phase in the development of SRS is now completed. Some preliminary experimentation was conducted in the system and it is already at use in our research laboratory. Departing from the accomplished structure, several research works are now starting and already in course. These works include improvements on the reasoning mechanisms of SRS and the development of new features and tools, which we will describe in the next paragraphs.

The notion of projects that have associated resources, users and permissions was not included in the first version of SRS. An evolution of SRS taking into account these requirements is being developed. One of the features that has been added within this evolution is the ability to associate a Version Control System (VCS) repository, such as CVS\(^1\) or Subversion\(^2\), with a specific project. This way, the source code and documents stored in the VCS repository is then stored and indexed in SRS, becoming available for reuse. The VCS repository logs, which are another source of valuable information about the usage of these repositories, are also retrieved from the repository and stored in SRS for analysis. The SRS Project Manager was developed as a new management tool for SRS, providing an interface for project related functionalities. Besides the usual operations of project management, this prototype already implements some log analysis features, such as analysing users and the files they have modified, analysing the files that have been modified together and log filtering.

As described before, the Indexing Module provides an interface to submit and index knowledge in SRS, in the form of SDKE’s. This module is considered to be crucial to the performance of the whole system, because the way knowledge is indexed in the platform influences all the processes that make use of this knowledge. The preliminary experimentation conducted in SRS revealed some weaknesses in the way knowledge is indexed, mainly because it is largely subjective and hard to evaluate, but also because it is not exploring all the potential hidden for instance in the structure of documents and source code. In order to improve the indexing of knowledge in the platform, we are currently working on the redesign of the Indexing Module. The redesign process will make use of state of the art tools and techniques of Text Mining, Information Retrieval and Natural Language Processing. Also, the indexing of knowledge will also take into consideration the structure of source code and documents, allowing indexing the resources and their most relevant parts independently. Besides the redesign of the indexing process within SRS, a prototype named SRS Legacy Code will be developed to automatically retrieve, store and index in SRS, legacy source code and documents from different kinds of repositories.

The Domain Ontology in SRS is dynamically updated as new concepts are needed to index new knowledge. The concepts that are represented in the Domain Ontology are retrieved from WordNet\(^4\), which is a large lexical database of English. This approach is very reductive, for instance taking into account that WordNet is very generalist and is limited to the English language. To overcome this situation, research effort is being put on a solution capable of retrieving the needed concepts from other source ontologies. The SRS OntoDig is a first prototype that implements a multi-agent framework, based on the architecture presented by Wooldridge [5], for retrieving concepts from different ontologies. This approach uses expert agents that deal with specific ontologies, or repositories of ontologies, and retrieve concepts from those ontologies upon a query from SRS. A broker agent mediates the responses from these expert agents and chooses the best result, which is then supplied to SRS. This prototype will now be integrated with SRS and a research effort will be put on evolving it into a robust solution.

4. CONCLUSIONS

The SRS platform intends to explore new ways to capture, store and reuse software development knowledge. The use of ontologies and the Semantic Web technologies enable the association of semantics to software artifacts, which can be used by inference engines to provide new functionalities such as semantic retrieval, or suggestion of relevant knowledge. We have described SRS and ongoing research that is being developed on it.

5. REFERENCES


\(^1\)http://www.nongnu.org/cvs/
\(^2\)http://subversion.tigris.org/