Supporting Software Maintenance in Web Repositories through a Multi-agent System

Aurora Vizcaino¹, Jesús Favela², Mario Piattini¹, and Felix García¹

¹Escuela Superior de Informática. Universidad de Castilla-La Mancha
13071, Paseo de la Universidad, 4, Ciudad Real, Spain
{avizcain,mpiattini,fgarcia}@inf-cr.uclm.es
²CICESE, Ensenada, México
favela@cicese.mx

Abstract. Software Maintenance (SM) is a knowledge-intensive activity. A suitable management of this knowledge could decrease the high costs (economic and in effort) of software maintenance tasks. The challenge of managing this knowledge increases as the distributed development of software becomes more popular, and developers as well as knowledge are distributed worldwide. Increasingly web repositories are being used for the coordination of development tasks among software and maintenance engineers. Thus, an appropriate technical solution to this problem should be based on a web architecture and associated protocols. On the other hand, in order to work with all the concepts related to SM is advisable to establish different levels of abstraction, thus the complexity of the concepts, and their management, are simplified. This work presents a system that, by storing information in XMI documents, manages the data and metadata generated during SM, facilitating the work of SM engineers.

1 Introduction

Many studies [7,15,18] provide evidence that the majority of the overall expenses incurred during the life-cycle of a software product occur during the maintenance stages. Thus, in recent years researchers have focused their attention on proposing methods and techniques which help increase the efficiency of the Software Maintenance Process (SMP).

One way to improve maintenance quality and decrease software maintenance costs is to reuse previous information and knowledge [10]. However, for information to be usable it needs to be modeled, structured, generalized and stored in a reusable form, to allow for effective retrieval [1].

In order to decrease the efforts and cost in the SMP we developed a knowledge management system called KM-MANTIS. The system is in charge of storing and reusing information, knowledge and expertise generated during the SMP. KM-MANTIS is based on the experience factory concept [4,5] since it is known that an organizational memory must be maintained by an organizational unit separate from the project organizations because it is mainly concerned to keeping schedules and cost constraints, and proving knowledge would imply an extra effort for the project organizations employees.
KM-MANTIS is a multi-agent system where different types of agent manage the diverse types of information generated during SMP. Agents interchange data and take advantage of the information and experience acquired by other agents.

The main feature of KM-MANTIS is that it stores data and metadata in XMI (XML metadata interchange) [13] documents. It is critical to manage different levels of abstraction to carry out an integral management of a software process. Moreover, being XMI an open format fosters the interchange of data between distributed and heterogeneous repositories which also use XMI.

The contents of this paper are divided as follows. Section 2 discusses the importance of managing the knowledge generated during the software maintenance process. Section 3 describes the features of KM-MANTIS. Section 4 explains how and why information is stored in a XMI repository, and shows, through an example, the use of the system. Finally, conclusions are presented in Sect. 5.

2 The Need for Knowledge Management in Software Maintenance

Software maintenance is an activity where different kinds of knowledge are generated from different sources. This knowledge comes not only from the expertise of the professionals involved in the process, but it is also intrinsic to the product being maintained, and to the reasons that motivate the maintenance (new requirements, user complaints, etc). Moreover, the diverse types of knowledge are produced at different stages of the maintenance process. For instance, [6] claims that during Initial Development the maintenance staff acquires knowledge about the application domain, user requirements, roles of the application, solutions and algorithms, data formats, strengths and weaknesses of the program architecture, operating environment, etc. In the Software Evolution stage new users and environmental requirements arise and changes are produced which generate new knowledge, and this process continues in the rest of the stages.

Furthermore, this knowledge is used by different persons at different stages. Each person has partial information that is required by other members of the group. If the knowledge only exists in the software engineers and there is no system in charge of transferring the tacit knowledge (contained in the employees) to explicit knowledge (stored on paper, in files, etc) when an employee abandons the organization a significant part of the intellectual capital goes with him/her.

One of the five factors that have been identified as having a major impact on the productivity of software maintenance is the expertise of the staff members [2]. It has been found that systems maintained by relatively inexperienced programmers average significantly higher error rates [3].

Another well-known issue that complicates the SMP is the scarce documentation that exists related to a specific software system. Even if detailed documentation was produced when the original system was developed, it is seldom updated as the system evolves. For example, legacy software written by other units often has little or no documentation describing the features of the software.

In addition, organizations still lack a culture of reuse and information sharing. As [12] claims, companies only use a quarter part of its intellectual capital.

On the other hand, thanks to technology advances it is frequent for maintenance teams to be geographically distributed, which implies: