Knowledge Management System Framework and Agent-Based Tool to Support Collaborative Software Maintenance Environment

Mohd Zali Mohd Nor  
Faculty of Computer Science and Information Technology  
University Putra Malaysia, Serdang, 43400 Selangor, Malaysia  
zali@itstar.com.my

Mohd Hassan Selamat  
Faculty of Computer Science and Information Technology  
University Putra Malaysia, Serdang, 43400 Selangor, Malaysia  
hasan@fsktm.upm.edu.my

Rusli Abdullah  
Faculty of Computer Science and Information Technology  
University Putra Malaysia, Serdang, 43400 Selangor, Malaysia  
rusli@fsktm.upm.edu.my

Masrah Azrifah Azmi Murad  
Faculty of Computer Science and Information Technology  
University Putra Malaysia, Serdang, 43400 Selangor, Malaysia  
masrah@fsktm.upm.edu.my

Abstract

Software maintenance (SM) environment is highly complex, knowledge-driven and collaborative. Issue such as inadequate knowledge is still regarded as a major challenge in SM. Knowledge management system (KMS) is therefore critical to provide an environment for sharing and sustaining knowledge among members of both business users community of practice (CoP) and SM CoP. One of the important but least available knowledge required by maintainers is the Domain Business knowledge and their relations to the applications and SM process knowledge. In this study, a KMS Framework shall be established and a Multi-Agent System (MAS) tool to support knowledge storage and sharing in collaborative SM environment is proposed. Critical to building the MAS is extending the SM business rules ontology to business domain ontology. The tool shall assist CoP members to get the right knowledge at the right time, thus improve productivity for both business users and maintainers. To evaluate the effectiveness of this agent-based KMS tool, the proposed prototype shall be implemented in a SM organization and evaluated using non-randomized control group pretest-posttest design of experiment methodology and quantitative survey.  

Keywords – Agent Technology, KMS, Software Maintenance

1. Introduction

SM is very important to organization mainly because organizations depend on how quickly and reliable the software can be changed to meet the ever changing business environment [6] and to survive the stiff competitions. SM is also recently becoming more challenging as the organization becomes more global or distributed [27].

SM is defined as “The totality of activities required to provide cost-effective support to software system. Activities are performed during the pre-delivery stage as well as the post-delivery stage.” [11,20] In term of costs, SM is apparently costlier compared to software development. Previous works by [10,16,19,25] estimate maintenance costs of between 60% to 90% of total software life cycle costs. These could be attributed to the fact that within software engineering discipline, software maintenance offers significant differences over software development, as listed below [4]:

i. Maintenance is technically more restrictive, since the system architecture, database and language are inherited from development.

ii. Whilst development is driven by requirements, maintenance is driven by external events such as changes to business model.

Meanwhile, knowledge is defined as “a fluid mix of framed experience, values, contextual information and expert insight that provides a framework for evaluating and incorporating new experiences and information. It originates and is applied in the mind of knowers. In organizations, it often becomes embedded not only in documents an repositories but also in organizational routines, processes, practices and norms.” [8].

Meanwhile, KM, in technical perspective, is defined as the strategies and processes of identifying,
understanding, capturing, sharing, and leveraging knowledge [1,3,8,26]. One of the major challenges is to facilitate flow of knowledge not only within individual, but also within teams and organizations [3].

SM activities are knowledge-driven [22], and depend on largely expertise of the maintainers. Some of these expertise are documented as implicit knowledge, but more are hidden as tacit knowledge due scarcity of documentation [27]. This is why, Nonaka and Takeuchi [17] suggested that KM requires a mechanism or ‘place’ to convert internalized tacit knowledge into explicit knowledge (information), and also allow individuals to internalize meaningful information. In order to do these, maintainers have to collaborate with colleagues and other parties to obtain various information to enable them to carry out their software maintenance tasks.

1.1. Problem statement

In SM, business domain knowledge and SM process knowledge are intertwined and affects each others. While domain knowledge are important to maintainers, they are lacking, not stored properly or not readily available [24]. As such, maintainers, especially newcomers, spend a lot of time searching, collaborating and understanding these knowledge [7]. Changes to domain knowledge due to enhancements or new business practices often affect usage of software application. Similarly, changes to application and SM processes due to corrective, preventive and perfective maintenance may affect the way users use the application. Therefore, it is imperative that the domain knowledge be structured, captured, and properly linked to the SM process knowledge. These combined knowledge should be properly managed to allow both maintainers and users easier and automated access to the updated information.

1.2. Motivation

The motivation of this study is therefore to provide tools for maintainers to perform their daily activities, by managing domain knowledge, application knowledge and SM process knowledge into a seamless, integrated system. It is envisaged that the following mechanisms shall support the above goal:

i. Domain knowledge are made available and traceable to software design and usage.

ii. CoPs shall rely on system rather than people (experts) and raw source code.

iii. The new tool shall promote automation and integration of systems or tools to support maintenance processes.

This paper is structured as follows: section II discusses the related work on the similar tools and Section III reviews the methodology for the study. Then, section IV and V shall discuss the current progress and contributions of the research.

2. Related Work

Agent technology in SM has been used in a few research prototypes such as the KM-Mantis [29]. KM-Mantis uses experience factory concept to capture and reuse knowledge and expertise generated during SM process. KM-Mantis has five software agents (product, activity, client, user and staff) to support the 3 different communities (product, activities and people).

![Figure 1 KM-Mantis Architecture](image)

Another MAS prototype is similar to KM-Mantis, but uses different agents to capture knowledge from different source documents and activities. Rodriguez et al. [23] developed a MAS architecture with 5 main agents. Staff agent provides supports to maintainers by monitoring maintainers activities, search for knowledge sources via KMA. These are Product Agent, Client Agent, Staff Agent, Project Agent and Directory Agent (see Fig. 2.)

Intelligent agent structures the knowledge in form of ontology. The uses of ontology to support the agent-based tool is critical due to the fact that Agents use ontology to share common terms, to communicate to other agents [30] and to understand the environment in which they operate. When agent retrieves or store the knowledge, it needs to know how the knowledge is structured. These semantics is the ontology of the knowledge [32]

In SM area, several researchers have developed ontologies on SM processes [4,9,13,23] which consists of the concepts, classified into the people, process, product and organization. The above works on SM ontology emphasize mostly on the SM activities. However, linkages to domain knowledge are not emphasized. This is where this study shall close the gap.
3. Methodology

This research shall be carried out in five phases as illustrated in the following research framework (Fig. 3):

Phase 1 – LR and Analysis of KM in SM
This phase involves evaluation of existing software maintenance processes and activities and literature reviews on academic journals, software maintenance books, and information search on groupware tools, Help Desk applications, Software Configuration Management (SCM) and other tools supporting software maintenance activities.

In phase 2 - Conduct Pre-Survey on KM Tools in SM, questionnaire survey shall be used to evaluate the different maintenance models used by selected SM organizations in Klang Valley, Malaysia. The survey shall cover collaborative activities and knowledge required within the software maintenance processes. Also, usage of collaborative tools and knowledge sharing tools shall be identified. This stage has been completed.

Phase 3 - Formulate a KMS Framework entails the formulation of a new model of applying KMS for collaborative software maintenance, as follows:

i. Based on earlier literature reviews (LRs) and pre-survey results, areas of concerns and different knowledge requirement areas are noted

ii. Build a KMS Framework to model based on the constructs from the above LRs. This framework

Figure 2 Research Framework

Figure 3 The KMS for Collaborative SM - Component Model

Figure 4 Proposed MAS System Overview

Figure 5 Proposed MAS System Architecture
shall be face-validated by experts. Meanwhile, phase 4 - Develop Prototype shall include the following steps:

a. Design and Specifications for Agent-Based Tool
b. Develop - Based on the above Agent Types design, the next step is to develop the prototype of Agent-Based KMS
c. Test and enhance
d. Release to Production

Phase 5 involves the evaluation to compare the effectiveness and performance of the new prototype KMS. A pretest-posttest experiment shall be employed, using Non-randomized control group Pretest-Posttest Design [15]. This experimental design is proposed due to the objective to compare the relative effects of different treatments for the same group of users and maintainers. At the same time, a control group is added to ensure higher internal validity to the experiment.

To ensure that we are measuring what we are supposed to measure, the Basili’s Goal-Question Metric (GQM) [5,14] is proposed. This approach uses a template to define metrics based on understandable and structured goals as illustrated in Fig. 4.

The proposed metrics shall also be based on Wu and Wang’s measurements on KMS success [31], which is based on DeLone and McLean Information System success model.

4. Current Progress

This research is in the stage of finalizing Phase 3 and in the midst of Phase 4. The results are as follows:

i. A Framework of KMS Components in Collaborative SM

The framework is based on theoretical constructs from the review of components from the existing KM frameworks, KMS frameworks within the area of Software Engineering. The resulted model was refined vis-à-vis a survey conducted in three in-house SM organizations in Klang Valley (see Fig. 3).

ii. Design of Multi-Agent System (MAS)

The proposed MAS shall serve as an intermediary tool that integrates SCM, Helpdesk and KMS applications. In a collaborative environment, software agents in MAS can be viewed as personal assistants and service agents [1]. Using Prometheus methodology [18] the proposed MAS functionality and architecture are depicted in Fig. 4 and Fig. 5, respectively.

iii. Design of Ontology

Based on SM ontology by Ruiz and hierarchical domain ontology proposed by Kabilan [12] and Ulrich [28], the combined SM and business domain ontology have been designed using Protégé methodology.

5. Contributions

i. Theoretical Contribution

The study shall contribute, to the body of knowledge, the conceptual KMS Framework for collaborative SM environment. The KMS Framework shall be based on the relations between the different KM components used by users and maintainers to collaborate and share knowledge within their respective community of practice. A pre-KMS survey is planned to verify the components, and a post-KMS survey shall be used to validate the KMS framework.

ii. Methodological Contribution

Three main contributions are expected from this study, as follows:

a. Methodology to design a combined Domain business knowledge and SM process knowledge
b. Methodology to design Agent types and architecture for KMS in Collaborative SM
c. Methodology to evaluate the KMS tools

iii. Practical Contribution

The study shall develop and implement a MAS-based prototype which shall be used to support KMS in collaborative SM environment. The prototype shall serve to validate the usage of software agents to assist users and maintainers to capture, store, share and use the knowledge and hence improve productivity.

In addition, a working model of combined Domain knowledge and SM ontology shall be developed, to allow software agents to manage the knowledge and collaborate.

6. Conclusion

The goal of this research is to provide tools to support maintainers to perform their daily activities, by managing domain business knowledge, application knowledge and SM process knowledge into a seamless, integrated
system. It is envisaged that the following mechanisms shall support the above goal:

i. Domain business knowledge are made available and traceable to software design and usage.
ii. CoPs shall rely on system rather than people (experts) and raw source code.
iii. The new tool shall promote integration of systems or tools to support maintenance processes.

To achieve the above objectives, a 6-phase plan is proposed to study and identify the SM processes and environments, and to formulate the framework for KMS in collaborative SM. The subsequent phases shall be the implementation on a new tool to support KMS in SM, and evaluation of the new tool. The tool is expected to streamline management of knowledge, especially the domain knowledge, to allow maintainers to perform their tasks effectively and hence increase productivity.

References


