Modeling Multi-Agent Systems for Knowledge Management in Software Development Process

Lina Salles, Kênia Sousa, Lívia Nojoza, Jô Oliveira, Vasco Furtado
Universidade de Fortaleza – UNIFOR, Mestrado em Informática Aplicada – MIA
linasalles@hotmail.com, kenia@unifor.br, livia.mia@unifor.br,
jo.oliveira@terra.com.br, vasco@unifor.br

Abstract

One of the requirements for Knowledge Management is to define and implement mechanisms to build the corporate memory. Traditional systems for maintaining the corporate memory are based on user's intervention to update knowledge. Adequate approaches for this should however consider the continuous and transparent process for capturing and recovering knowledge. It must be a process inserted in the user-working context. This article proposes the use of intelligent agents technology to manage knowledge in the context of software development process, adopting a methodology for helping in the creation and maintenance of a corporate memory and a framework for helping in the construction of agents. The use of intelligent agents technology will allow the collaborative and transparent construction of a corporate memory.

Keywords: Knowledge Management, Organizational Memory, Intelligent Agents.

1. INTRODUCTION

Knowledge Management (KM) can be understood as a systematic set of processes to capture, store, and disseminate knowledge in an organization in order to contribute with knowledge sharing and reuse in the organization. Organizations that noticed the knowledge relevance to the creation of new businesses have become more competitive in the global market [1].

When we mention organizational knowledge, there are more than just databases and information systems involved; there are also processes and, mainly, people. Knowledge is intrinsic to people [2]. Knowledge management happens when there are mechanisms that structure the entire knowledge and build the corporate memory. Traditional systems for corporate memory maintenance require the user intervention to update knowledge. Such additional step in the user-working process has proven to be inefficient. It is expected that knowledge management solutions to be inserted in the user-working environment. Knowledge acquisition and recovery need to happen in a continuous and transparent manner since the users are focused on their daily working activities. Therefore, the corporate memory must be built as a consequence of users' work [3].

This article aims to present how intelligent agents can help in creating and maintaining the corporate memory, specifically in software development process, by applying a methodology to model the organizational knowledge. The use of agent technology allows the construction of the corporate memory collaboratively. The main point in the proposed methodology is the identification of intelligent agents. For each model generated by the methodology, a class of agents is provided. The agents are, called K-Agts (Knowledge Agents), which are specialized, as the models are detailed.

This article is organized as follows: section 2 presents the state-of-the-art of the problem domain. Section 3 explains the K-Org methodology used for mapping the agents. Section 4 describes the K-Org methodology application. Section 5 presents the results achieved after the methodology application. Section 6 describes the framework to implement the agents. Finally, section 7 concludes this work.
2. THE STATE OF ART

The state of art in this research work is related to knowledge management methodologies and intelligent agent technologies. This section presents related works with them.

2.1. Knowledge Modeling Methodologies

In order to achieve benefits derived from building a corporate memory, it is necessary to model the organizational knowledge. Modeling the organizational knowledge is not an easy task. For instance, in large-scale companies, the knowledge distribution among its various departments make it more difficult to perform knowledge mapping, that is, organize it to facilitate sharing. Representing experts’ tacit knowledge – an organizational valuable resource – is another obstacle in the modeling process. Therefore, it is mandatory to adopt a methodology in order to aid in the knowledge modeling process in any organization.

The use of Enterprise Models in works concerning KM is widely diversified: some of them focus in only one model aspect, others focus on various relevant aspects. A modeling approach that considers multiple perspectives, in general, uses various modeling languages [4]. The set of tools described by Stader at [5] includes the Agents Tool Kit to develop agents, the Procedures Builder to model processes, the Task Manager to integrate, visualize and control tasks; and the Enterprise Ontology to be used for communication purposes.

In Zachman [6], architecture for enterprise modeling used various modeling languages to capture and describe different aspects of a domain. Ulrich [7] also used a multiple notation approach and proposed the architecture for a KBS, inspired in an enterprise modeling method called MEMO (Multi-perspective Enterprise Modeling). Decker [8] proposed the integration among Knowledge Engineering (KE) techniques and the business process modeling context. Its main contribution is related to the link among business processes modeling from the ARIS (Architecture of Information Systems) architecture and the knowledge models from the MIKE (Model based and Incremental Knowledge Engineering) approach.

Rosemann [9] proposed a Framework, with three dimensions, which structures the knowledge necessary for managing and deploying Enterprise Resource Planning (ERP): the first one specifies the knowledge life cycle; the second one explains the main steps in the cycle of an ERP; and the third one identifies the types of knowledge required to implement an ERP. This work demonstrated how an organizational specific knowledge, such as business, technical, and products knowledge, could be captured through a reference to processes model.

Schreiber [10] was concerned with KADS evolution, which resulted in changing its name to CommonKADS. The focus on previous works concerning the development of Knowledge Based Systems (KBS) remained in proposed methods and models, which evolved to a structured approach in the organizational context. CommonKADS models are based in four principles: creating models containing different aspects of the human knowledge; conceptual knowledge structure, not including programming details; the knowledge contains a stable internal structure, which is analyzed according to its types and functions; and the knowledge project management must be done according to a spiral approach, which allows a structured learning process.

2.2. Intelligent Agents

To be more effective, the corporate memory needs to fill some practical requirements related to the collection, organization and exploration of the information to be stored. The agent technology plays a key role in fulfilling these requirements. Some alternatives of the use of agents for KM through the corporate memory are [11]:

---

[11]
• Systematic collection and organization from distributed information sources. The necessary knowledge in the work process is dispersed for some sources such as documents, databases, e-mails, and personal notations. Distributed and independent agents, for its decentralized characteristic, can in a parallel way explore more diverse sources of knowledge with a coordinate mediation;

• Reduction of the knowledge engineering task because organizations may be reluctant to invest time and money in a new technology for which results are distant and uncertain. In addition, users do not want to spend time with the knowledge acquisition process. Agents for its autonomous characteristic can automatically explore most of the available information (electronic database or documents), providing benefits quickly;

• Exploring user feedback for the maintenance and evolution so that deficiencies and suggestions are, respectively, decided and incorporated. Intelligent agents know the user profile, being able to intermediate the interaction process of people with the corporate memory in an adaptive and intelligent way.

Concerning to the use of multi-agent systems, Cao [12] proposed a society of agents in charge of wrapping external HTML documents that are relevant to the activities of the organizational. The work followed an organizational top-down analysis where the multi-agent system architecture was tackled, as in human society, in terms of groups, roles and relationships, starting from the highest level of abstraction of the system (i.e. the society) and going down by successive refinements (i.e. nested sub-societies) to the point where the needed agent roles and interactions could be identified.

Jacobson [13] introduced the notion of a developer assistant do provide some of the expert knowledge that a developer needs. The developer assistant is implemented as a multi-agent system composed by role agents, activity agents and artifact agents. Each agent can support the developer via a fairly number of rules. The agent presents the rules, and it is then often up to the developer to select a single rule and apply it in the context.

3. K-ORG METHODOLOGY

The K-Org methodology [5] aims to analyze an organization by observing the interaction among people and the activities and information shared among them in order to map a system that allows the construction of the corporate memory. The K-Org macro-workflow, depicted in Figure 1, presents the phases performed by organizational actors in order to generate models and identify intelligent agents.

![Figure 1. K-Org Macro Workflow](image)

The contextual phase corresponds to modeling the organizational knowledge concepts and the communication process among professionals. The conceptual phase corresponds to
understanding the organization, its structures, business processes, and requirements. The project phase corresponds to mapping the knowledge and abilities of the user, and knowledge engineer point of view. The implementation phase corresponds to publishing the contents in the knowledge base and creating interfaces to maintain the corporate memory according to specific needs.

In the contextual and conceptual phases, the generated models are: organizational, task, actor, knowledge and communication models. In the design phase, the generated model is the design model. In the implementation phase, the generated model is the knowledge portal. These models were adapted from the CommonKADS methodology.

For instance, in the contextual and conceptual phase, general agents are generated for each model; in the design phase, general agents are specialized to attend specific needs; and in the implementation phase technology-dependent agents are implemented. As the models are created, the identified agents become more specialized. The Figure 1 depicts the models used by the K-Org methodology and the agents for each one of the models.

Following, we describe the K-Org models with its respective generated agents.

- The organizational model focuses on the organization, its problems, opportunities, possible changes, and their impacts. The organizational K-Agt aims at verifying actions taken by professionals in order to identify strategic perspectives in the information, technology, and knowledge management areas;
- The task model focuses on analyzing a task as part of the business process. The analysis of this model leads to the identification of task K-Agts that aim at registering task scenarios with practical cases in order to facilitate understanding the execution of tasks;
- The actor model focuses on describing characteristics of professionals, their abilities, authority to make decisions, and restrictions. The analysis of this model leads to the identification of actor K-Agts. The actor K-Agt aims at understanding professionals’ roles and abilities within the organization in order to perform their tasks;
- The knowledge model focuses on understanding the knowledge necessary to perform a certain task. The analysis of this model leads to the identification of knowledge K-Agts. This model is divided in three parts: domain, inference, and task knowledge. The knowledge K-Agt aims at acquiring and representing knowledge related to specific tasks;
- The communication model focuses on representing the interactions among professionals involved in performing a task. The analysis of this model leads to the identification of communication K-Agts that aim at capturing the information exchanged among professionals while performing their responsibilities.

4. K-ORG METHODOLOGY APPLICATION

The K-Org methodology was applied in the software development process. In order to achieve this, we interviewed two software companies, one from Brazil and one from the United States, in order to analyze their peculiarities, such as cultural, market, and social differences. As a result, we defined a software company general structure.

Software organizations’ structure basically involves two main processes: software development, which involves the activities performed by the development team aiming at generating a software product with quality and: project management, which is responsible for planning and monitoring the activities performed during the first process.

The solution for most problems faced by software development organizations is related to knowledge acquisition and sharing, and individual experience necessary to generate organizational knowledge. The outcome must not depend on individual knowledge. The loss
of an employee can lead to great losses in the organization if the knowledge was not previously shared or registered.

The Organizational Model generated by K-Org allows the perception of all these aspects and the mapping of business processes, in which it is necessary to invest in knowledge management. As a result of modeling the organization, we identified Organizational K-Agts. They maintain the corporate memory updated in terms of the enterprise processes. The K-Agt Software Development aids software developers in performing their tasks by providing relevant knowledge when necessary. It is composed of a set of specialized K-Agts useful to support each phase in the process – the K-Agt Requirements, the K-Agt Analysis, the K-Agt Design, the K-Agt Implementation and the K-Agt Deployment; The K-Agt Project Management aids project managers in performing their tasks by providing relevant knowledge when necessary. It is composed of a set of specialized K-Agts useful to support each phase in the process – the K-Agt Project Planning and the K-Agt Project Monitoring; The K-Agt Competence maintains professionals’ competences updated.

Continuing the K-Org application, we elaborated the Task Model. The chosen task to be analyzed in the software development process is requirement elicitation. The task model is useful to identify intelligent agents that can help professionals in performing their tasks. As a result of modeling the organizational tasks, we identified the Task K-Agts. The K-Agt Team allocates professional’s according to their profile; The K-Agt Responsibility maintains professional’s responsibilities updated; The K-Agt Elicitation: identifies users requirements; The K-Agt Architecture: suggests the software architecture based on previous similar projects.

The actor model specifies the main characteristics of professionals within the organization. The professionals focused in this model are the project manager, the business analyst and the software programmer, that are involved at the main processes of software development. As a result of modeling the organizational actors, we identified some Actor K-Agts. The K-Agt Training detects problems in the execution of tasks and suggests training courses; The K-Agt Incentive detects correctness in the execution of tasks and suggests incentives; The K-Agt Profile detects professional’s responsibilities.

The knowledge model selected specifies the main characteristics of the knowledge related to the requirement elicitation. Figure 2 focuses on the requirement elicitation knowledge detained by the project manager and the system analyst to identify users’ needs and expectations for the software. The knowledge mapped by this model will guide the specialization of the task agents and the agent behavior definition.

<table>
<thead>
<tr>
<th>Knowledge Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Item</td>
</tr>
<tr>
<td>Detained by</td>
</tr>
<tr>
<td>Used in</td>
</tr>
<tr>
<td>Domain Knowledge Category</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Inference Knowledge Category</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Function</td>
</tr>
<tr>
<td>Transfer function</td>
</tr>
<tr>
<td>Tasks Knowledge Category</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Figure 2. The Knowledge Model for the Requirement Elicitation Process
This model presents information about a knowledge item necessary for a specific process within the organization, and details it by specifying the knowledge in three categories: domain, inference, and tasks. As a result of modeling the organizational knowledge, we identified some K-Agts. The K-Agt Cases presents similar cases from previous projects with possible solutions; The K-Agt Documents presents documents from related projects; The K-Agt Mapping organizes documents grouped by knowledge; The K-Agt Workflow controls the steps followed by professionals while performing their tasks.

The communication model specifies the communication aspects necessary during the performance of the requirement elicitation: the information manipulated, the actors involved, the communication plan, and the associated restrictions. As a result of modeling the organizational communication, we identified these Communication K-Agts. The K-Agt Search looks for professionals who can help other professionals in performing their tasks; The K-Agt E-mail detects e-mail exchange with information related to the current project; The K-Agt Client detects tasks performed in the client site.

5. AGENT COMMUNITY FOR SOFTWARE CORPORATE MEMORY

The application of the K-Org methodology resulted in the definition of the knowledge management system composed by communities of agents. The communities are People Management, Software Development, and Project Management. These three communities interact with each other in order to facilitate knowledge sharing and acquisition among experts involved in the software development process, as well as to update knowledge in the corporate memory. Following, we will describe these communities by focusing on the software development community.

5.1. People Management Community

The People Management Community is responsible for analyzing the professionals profile and responsibilities in order to update the organizational chart. The People Management Community can be viewed in two sub communities: People Management Planning and People Management Monitoring. The People Management Planning sub community aims at providing support to the selection and allocation process. The People Management Monitoring sub community intends to help in the decision-making process concerning the professionals.

5.2. Project Management Community

The Project Management Community is responsible for planning and monitoring projects. Project planning involves risk analysis, schedule and cost estimates, project communication plan definition. The estimates are calculated based on previous projects and on simulations. The project monitoring involves verifying whether or not all professionals are following the sequence of steps necessary to maintain the project within the predefined schedule and cost in order to minimize risks.

5.3. Software Development Community

The Software Development Community aims at supporting the software development process by helping in all its phases. This community is composed of K-Agts identified in the organizational, task, actor, knowledge, and communication models. This community is composed of sub-communities that represent each phase in the software development process.
The community is made of various K-Agts. K-Agt Documents is responsible for presenting documents from previous related projects in order to help professionals in executing their responsibilities. These documents can be acquired by K-Agt Mapping, which organizes documents and groups them by knowledge. Another document acquisition manner is through K-Agt E-mail that detects e-mail exchange, in which the subject messages contain information related to a certain project. Since most of the work professionals perform happens in the client site, K-Agt Client can be installed in a computer in the client site, as a plug-in, in order to detect the activities performed by the professionals. The K-Agt Cases is responsible for presenting similar cases from previous related projects, optionally followed by possible solutions. The K-Agt Search is responsible for looking for professionals that are able to help others in the execution of their responsibilities. This is possible by using K-Agt Profile that detects professionals’ responsibilities through the use of the K-Agent Responsibility that maintains professionals’ responsibilities up-to-date. The K-Agt Workflow is responsible for guaranteeing that all professionals are following the sequence of steps necessary to generate relevant documents, and to maintain the accordance to the pre-defined schedule.


A new automated environment called K-Org Framework was developed to register the methodology models and to construct the mapped K-Agts.

Figure 3 shows the K-Org Framework architecture, which is composed of:

- **Graphical User Interface**: allows the user to interact with the framework. Through this interface the user registers all K-Org models and K-Agts’ definitions;
- **Database System**: a repository to store models’ and K-Agts’ definitions;
- **K-Agt Constructor**: generates Java specifications for each K-Agt according to K-Agts definitions stored at the database system. Three classes of K-Agts are generated: Domain K-Agts: agents related to the modeled domain; Control K-Agts: agents responsible for management activities: access control, configuration management, process control, dialogue control, etc.; Technology K-Agts: specialized agents that can apply the use of a technology. These agents interact with Domain K-Agts and Control K-Agts. Examples: inductive learning K-Agt, natural-language processing K-Agt.

![Figure 3. K-Org Framework Architecture](image1)

![Figure 4. K-Agt Constructor Interface](image2)

We can see in Figure 4 the K-Agt Constructor interface. The user describes the K-Agts by registering class, name, status, inheritance, attributes, behaviors and knowledge. The class property distinguishes a control K-Agt, a domain K-Agt or a technology K-Agt. The behavior property represents the message exchanging between K-Agts. The knowledge property
describes the K-Agt knowledge by if-then-else rules implemented in JEOPS format (Java Embedded Object Production System) as shown in Figure 6.

When the K-Agt definition process finishes, the K-Agt Constructor generates a Java implementation code. Figure 5 presents a Java codification to the K-Agt Architecture. This Domain K-Agt was identified by the methodology application at the requirement elicitation process, described in section 4. Other K-Agts of this phase are K-Agt Elicitation, K-Agt Project Cases and K-Agt Documents.

As new K-Agts definitions happen, the K-Agts reuse become easier, specially the reuse of Control K-Agts and Technology K-Agts. The task of get similar cases, for instance, is done by means of message exchange with the K-Agts Project Cases as it can be seen in Figure 5. The exchanged parameters defines the case base as input and the list of similar cases as output.

Domain K-Agts, Control K-Agts and Technology K-Agts were grouped into communities, to compose a Multi-Agent System. This multi-agent system interacts with the user environment, sensoring user necessities and providing knowledge if necessary.

To understand how this interaction works, let’s follow the activities performed in a requirement elicitation scenario. Suppose that the user is working with Rational RequisitePro [14] to specify the system requirements. Figure 7 presents the requirement type definition in Rational RequisitePro. The user defined a requirement as a use case model. A requirement type determines how the project requirements will be managed.

If the user desires to select an appropriate architecture to the project, the multi-agent system activates three K-Agts to help the user: the Control K-Agt Requirements, the Domain K-Agt Architecture and the Technology K-Agt Project Cases. K-Agt Requirements makes the sensoring of the user activities in his/her environment. After that, K-Agt Requirements sends a message to K-Agt Architecture to ask for candidate architectures. This K-Agt sends a message to K-Agt Project Cases, which searches the corporate memory to find similar cases. The K-Agt Project Cases was implemented as an extension of a K-Agt Cases that uses case-based reasoning technology. The similar cases founded by K-Agt Project Cases are sent to K-Agt Architecture, that sends them to K-Agt Requirements. The knowledge obtained is presented to the user, through an interface. This process is described in Figure 8.

Figure 9 presents the interface provided to the user. All similar projects that were found in the corporate memory are listed. The user can select a link and discover the knowledge stored
in that link. The analysis of this knowledge would help the user to make a decision on the requirement elicitation process.

![Figure 7. Requirement Type Definition](image1)

![Figure 8. Multi-Agent System Interaction](image2)

![Figure 9. Multi-Agent System Interface](image3)

The implementation of this scenario, concerning to requirement elicitation process, helped to validate the use of both the K-Org Methodology and the K-Org Framework. The construction of K-Agts by the K-Org Framework simplifies the development of knowledge management systems for corporate memory update.

K-Org Framework was implemented in Java, with a Windows user interface, accessing a relational database system (MySQL). The code complement was inserted in JADE environment (*Java Agent Development Framework*). The rule database that represents K-Agt knowledge is stored in JEOPS.

7. CONCLUSION

This work described the application of the K-Org Methodology in software development process. This attention on the knowledge control is very rewarding to the organization, since the knowledge necessary for professionals to perform their responsibilities is easily made accessible to the right person at the right moment.

The main contribution of this work is the modeling of a general knowledge system based in Intelligent Agents for software development process. The Agents’ Communities have the goal of creating, and maintaining a corporate memory.

Generally, related works propose approaches to use multi-agent systems. These approaches, however, don’t propose a formal methodology to model the agents or a framework to construct the agents. The approach proposed in this article focuses on the idea that the use of a methodology and a framework to model and construct the agents will support the implementation of a multi-agent system. The insertion of a multi-agent system in the user-work context will bring an automatic and transparent creation of a corporate memory. For instance, we intend to avoid professionals in dedicating a great amount of time in feeding the
knowledge base by filling out long forms, since the agents are responsible for detecting and organizing the knowledge created as a result of their daily tasks.

These proposals intend to help software development process in creating, maintaining, and sharing knowledge among its professionals in an easy and efficient manner in order to allow them to be more competitive in the market place. Further works intend to enhance the agents’ implementation, including other software development phases.

ACKNOWLEDGMENTS

We thank Galt Associates’ representatives and Instituto Atlântico’s representatives for supporting this research by providing information related to their organizations.

REFERENCES