Agent-Based Support for Requirements Elicitation

Chad Coulin, Didar Zowghi
Department of Software Engineering, University of Technology Sydney
PO Box 123 Broadway NSW 2007 Australia
{chadc, didar}@it.uts.edu.au

ABSTRACT

The elicitation of requirements is a difficult and expensive process but critical to the overall success of any system development. So far relatively little work has been devoted to providing intelligent tool support for this complex and labor-intensive activity. The quality of requirements from the elicitation process currently depends greatly on the experience and expertise of the participating requirements engineers, and the commitment and cooperation of the system stakeholders.

In this paper we describe an agent-based approach to intelligent tool support for requirements elicitation. Given the multiple roles a requirements engineer must perform during elicitation, we suggest a multi-agent system may be developed as an intelligent assistant for this process. It is proposed that some of the tasks performed by requirements engineers during the elicitation process may be supported and in some cases automated by individual agents or several agents working cooperatively.

It is expected that the use of intelligent agents would produce better requirements in terms of their completeness, correctness, consistency and clarity. This would be achieved partly by improving the elicitation process through greater efficiency with respect to time and cost, and increased effectiveness by way of rigorous and structured execution.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications – *Elicitation methods, Methodologies, Tools.*

General Terms

Design, Theory.

Keywords

Requirements elicitation, intelligent tool, multi-agent system.

1. INTRODUCTION

Requirements elicitation is a very complex process involving many activities with multiple techniques available to perform those activities [7]. The multi-disciplinary nature of tool support for requirements elicitation only adds to this complexity with strong relationships to the fields of knowledge engineering, artificial intelligence, information systems, cognitive psychology and the social sciences. This is in addition to the large body of work in the obvious and more general areas of systems, software and requirements engineering.

.

It is generally understood that requirements are elicited rather than captured or collected. This implies both a discovery and development element to the process [5]. Requirements may be elicited from a variety of sources including the many different types of possible stakeholders in the future system, and documentation and processes from the existing systems.

Few attempts have been made to develop intelligent tools to support requirements elicitation especially where there is direct interaction with human stakeholders without the need for a requirements engineer driving the process or the use of a semi-formal modeling and analysis technique. In order to improve the quality of requirements and the elicitation process itself we introduce the use of agents as intelligent support for the requirements engineer during this phase of system development.

The paper is structured as follows: Section 2 explains the meaning of agents as intelligent assistants in the context of the requirements elicitation process. Section 3 describes a multi-agent approach to the requirements elicitation process. The use of agents for domain knowledge is examined in Section 4, and in Section 5 we investigate the particularly challenging area of discourse agents with respect to requirements elicitation. Finally in Section 6 we present a discussion with some conclusions and possibilities for future work.

2. INTELLIGENT AGENTS

We use the definition of an agent as a computer system situated in some environment that is capable of flexible autonomous action in order to meets its designed objectives [4]. In this definition there is additional importance placed on the term ‘flexible’ in that it refers to the responsive, pro-active and social nature of agents. For our purposes an agent can further be described as an active software component or entity utilizing intelligent technology in terms of its communication with is environment and its dynamic behavior [9]. Another important point to make is that agents may be organized into a hierarchy or social structure to interact and perform tasks with other agents and entities within that system [2].

Therefore three key concepts concerning agents can be defined as autonomy, adaptation and cooperation [1]. By this it is meant that an agent should have the ability to make independent decisions without external intervention, be aware of its environment and able to make changes to its behavior accordingly, and interact with other agents and entities and in some cases work cooperatively with them [6].

Agents are different to objects in the traditional software engineering sense in that an agent must ultimately exhibit control over its own behavior [4]. However agents may depend on other entities such as knowledge bases and the Internet to be able to meet their goals and complete their tasks. We take the view that an agent may act as a proxy for a requirements engineer in order to perform one or more tasks on their behalf.

The advantages of using agents include the ability to manage and reference large amounts of real time and historical data, and have greater control and consistency over the output and results of the process. This includes assuring that equal attention is paid to both the problem and solution domains, and that the final documentation and presentation conforms to accepted standards.

Agents could also enable the automation of some of the more mundane tasks a requirements engineer is required to perform during the elicitation process. This is particularly important given the time consuming nature of requirements elicitation especially when dealing with complex systems and large organizations.

3. A MULTI-AGENT APPROACH

3.1 Roles of the Requirements Engineer

Multi-agent systems are typically very complex and difficult to develop. Despite this it is more likely that a multi-agent system would be developed to support some of the many activities performed by a requirements engineer as opposed to a single 'fat' intelligent agent providing all the functionality. This is mainly due to the nature and behavior of agents, and the state of technology currently available.

We can begin to investigate the implementation of a multi-agent agent system for requirements elicitation by examining the various roles performed by the requirements engineer during this process.

A fundamental part of the entire requirements engineering process is related to project management. This activity involves more than the obvious decision-making and prioritization tasks. Project managers are also commonly required to initiate meetings with stakeholders, produce project status reports to inform stakeholders of progress, remind stakeholders of their responsibilities, and answer questions from stakeholders regarding the project, the process, and the system being developed.

When eliciting requirements by conducting interviews the requirements engineer does not only ask questions and record the responses but also must guide and assist the participants in answering these questions in order to elicit the most correct, complete and relevant information. The interviewer is also responsible for ensuring that participants feel comfortable and confident with the process in order to achieve the best possible results from this activity.

Conflicts between stakeholder requirements are inevitable. When this occurs the requirements engineer is often required to act as a mediator in working towards a suitable resolution. All elicited requirements must be validated and verified against each other and the previously established goals of the system. This may involve various semi-formal and formal modeling and analysis activities.

Requirements engineers are often required to assume the roles of the developer community during requirements elicitation such as system architects, designers, programmers and testers. Decisions made during the requirements stage will inevitably effect the later phases of system development.

The responsibility for the output of the elicitation process also lies with the requirements engineer. Typically this exists in the form of a requirements document or detailed system model. This role is particularly important as it represents the results of the elicitation process and forms the foundation for the subsequent project phases. Evaluation of the elicitation process and the work performed by the requirements engineer is based on these resultant artifacts which in some situations will form the basis of a contractual agreement as in the case where a system is to be developed for an organization by an external supplier.

3.2 Elicitation Activity Agents

For each of the roles detailed in the previous subsection we can identify one or more possible types of intelligent agent application to support the required activities.

3.2.1 Personal Assistant Agents

Personal assistant agents could be responsible for proactively organizing and driving the project tasks of human stakeholders and assisting the execution of these tasks by providing the necessary guidance proactively and when requested.

3.2.2 Information Acquisition Agents

Information acquisition agents could be used to search through existing documentation and knowledge bases to validate existing requirements and to discover new ones.

3.2.3 Elicitation Technique Agents

Elicitation technique agents could be used to interview, model, analyze and document requirements based on stakeholder input. Additional sub-agents could be developed to check for conflicts and consistency of requirements during elicitation, inform the stakeholders of the specifics, and advise on possible solutions.

Surveys are one of the most common elicitation techniques used by requirements engineers and include questionnaires and structured and unstructured interviews [3]. Agents may be used to design and propose questions based on meta-models and schemas, case stories, and abstractions of previously developed systems. It is also possible to conduct surveys based on goal refinement and other knowledge acquisition techniques.

Scenario and task analysis involves 'walking' stakeholders through existing or proposed system operations and defining each possible step and exception condition. An agent may be used to model, simulate and incorporate feedback for the described operations from either a user or system perspective. The use of graphical representation is especially useful during this process and could be incorporated into the behavior of the agent.

Some elicitation techniques are inherently more suited to the possibility of implementation through agents than others. These elicitation agents may be arranged into a structure with a parent or decision agent responsible for selecting the appropriate elicitation technique and agent to employ depending on information about the system stakeholders and other environmental constraints such as time and the availability of resources.

Agents may also be used to support rapid prototyping and other agile methods used for requirements elicitation. This could involve the use of interactive software construction agents during the early stages of system development. Further investigation is also required into how agents may be used to support observational and ethnographical techniques during requirements elicitation.

In practice requirements elicitation is an iterative process and typically a combination of techniques is used to discover and develop system requirements. For example a requirements engineer may conduct a follow-up discussion with the system stakeholders after validating the information gathered in a prior interview by observing the existing system in use.

3.2.4 Administration Management Agents

Administration management agents could perform much of the day-to-day administration tasks often assigned to the requirements engineer such as reserving meeting rooms and other resources, producing and distributing regular status reports, and reminding stakeholders of their obligations.

3.3 Agent Coordination

In the multi-agent system required to perform the various tasks a requirements engineer must conduct during the requirements elicitation process the agents would need to run concurrently and not interfere or conflict with each other. In some cases it may be necessary for agents to work cooperatively in order to satisfy their individual goals. For this reason it is critical to ensure that agents in this type of environment do no have conflicting goals and adhere to defined coordination strategies and protocols.

We can take the simple example of a decision agent required to determine which elicitation techniques are best utilized for a particular project and which of the stakeholders to involve. In this case the requirements engineer informs the web-based agent of basic project and organizational details. The agent has access to a knowledge base of various elicitation techniques, the conditions under which they can be performed, and their respective strengths and weaknesses. This is combined with information from stakeholders captured via an online questionnaire on their individual details, the goals and constraints of the project, the problem domain, and the availability of resources. The system includes a feedback mechanism also via an online questionnaire to evaluate the relative success or failure of the selected elicitation techniques. This is to enable the agent to additionally reference past experiences for future questionnaires and decisions and therefore continuously improving the performance of the system.

Furthermore a series of sub or contractor agents may be employed to perform the necessary subsequent tasks. An administration agent could be used to arrange a meeting where the appropriate elicitation agent would develop and conduct a structured interview based on the predetermined high-level goals and constraints defined for that particular system development project. The parent decision agent would need to communicate the number of participants and the type of meeting to the administration agent which may then in turn schedule the meeting based on the electronic calendars of individual participants and reserve an appropriately sized and equipped conference room.

A personal assistant agent would be responsible for reminding the stakeholder participants of the interview and determining any prerequisite tasks each stakeholder must perform prior to the meeting. The elicitation agent may decide to use a template, model or analogy to support the interview or employ an information acquisition agent to retrieve documentation of the existing system to use as the basis for further inquiry. Here the elicitation agent would need to communicate to the information acquisition agent the scope of the search and the type of information required.

It is beyond the scope of this paper to delve further into the specific details of the intelligent and behavioral aspects of the proposed agents although it is tempting to do so. However we can see from this example how the communication and coordination between agents in a multi-agent system is paramount to its success.

4. AGENTS FOR THE DOMAIN

Domain knowledge represents an important part of requirements engineering however the collection of domain knowledge is a very time consuming process. During requirements elicitation both the problem and solution domains need to be examined. This type of information can be exploited for requirement engineering in a variety of ways. Requirements engineers will use previous experience in the domain as a kind of mental template for group discussions and interviews. Domain analogies and abstractions of existing situations are used as baselines to acquire information in order to identify and model possible solution systems. This also provides the opportunity to reuse specifications from like and unlike domains, and validate new ones against existing domain knowledge as detailed in the work of Sutcliffe and Maiden [8].

Agents could be implemented as experts to provide assistance in not only the collection of domain knowledge but also its presentation for further information acquisition activities. We can define a domain expert as having an extensive knowledge of the domain area, the ability to identify similarities and differences between domain instances, and access to a catalogue of existing examples in the given domain for reference. Therefore it is possible to conceptualize an intelligent agent working in cooperation with a domain knowledge base to support the requirements engineer during the elicitation and modeling of requirements.

5. DISCOURSE AGENTS

Discourse agents that can actively participate in conversations directly with one or more human subjects at a time present us with some unique and complex challenges. Requirements engineers often use group discussions involving multiple stakeholders in elicitation. Examples of this include focus groups and joint application development (JAD) groups. In these cases the requirements engineer may or may not be required to have significant domain expertise in order to facilitate discussions.

We can investigate the difficulties of developing discourse agents by examining the role of group discussion facilitator that is often performed by the requirements engineer during elicitation. In this environment the requirements engineer is not only responsible for managing and guiding the other participants on the topics established for discussion but also to inquire when more information or clarification is required on a subject and maintain the relevance of all discussions to the problem at hand. It is important that the requirements engineer ensures that all parties represented in the discussion group are given sufficient opportunity to voice their positions and provide appropriate input to the conversations. The requirements engineer is sometimes also required to negotiate on behalf of absent stakeholders, validate previously established requirements with the current group of participating stakeholders, and mediate disagreements between stakeholders.

To perform these tasks effectively a requirements engineer must not only be able to just see and hear the participants but also analyze their body language and speech patterns. The requirements engineer uses this type of information throughout the elicitation process to gauge the level of importance and understanding of requirements within the group and detect possible conflicts and concerns the stakeholders might have.

It is difficult to envisage an intelligent agent capable of conducting this role given the technologies currently available. In this case an extremely complex system of both software and hardware would be required with substantial multimedia and sensory capabilities. Despite this is it interesting to look at how intelligent agent technologies might contribute to this area in future research.

6. DISCUSSION

From our preliminary investigation it has been determined that there are many conceivable applications of agents in the activities performed during requirements elicitation however whether or not one or more of these can be successfully implemented into an intelligent multi-agent system to support the requirements engineer in this process is yet to be completely examined both theoretically and practically.

Multi-agent systems provide us with some significant new opportunities and advantages over the more traditional expert systems. Typically multi-agent systems integrate several general tasks that can be personalized to the individual user as opposed to expert systems that perform a limited number of specific tasks in a fixed manner for all users. Agent-based applications are inherently more active, adaptive and mobile than expert systems in their behavior and interaction with users and other information sources.

It is worth mentioning that such multi-agent systems as proposed in this paper are currently very expensive and complicated to design and build. Additional obstacles such as the cultural change and acceptance of multi-agent systems within organizations would also need to be addressed for this type of technology to be adopted. Given this it is more realistic that single agents will be developed for generic applications such as searches and scheduling and then modified and applied to requirements engineering as opposed to multi-agent systems being designed and built specifically for requirements elicitation activities.

More research on agent development, architecture and application is still needed, especially in social environments and multi-agent systems where the issues of conflicts between agents, load balancing, belief revision, and the re-organization of agent commitments are still being examined.

It is not suggested that agents present a substitute for the role of the requirements engineer but instead that certain elements of the requirements elicitation process may be automated, supported and improved through the implementation of an intelligent agent-based system. We believe that the development of information systems presents a particularly good opportunity to apply this type of technology in both the early and later stages of requirements elicitation due to the large body of work in this area and the volume of available and relevant literature, domain expertise and case stories.

7. REFERENCES

[1] Dardenne, A., van Lamsweerde, A., and Fickas, S. 1993. Goal-directed requirements acquisition. Science of Computer Programming 20, 3-50.

[2] Giorgini, P., Kolp, M., and Mylopoulos, J. Multi-Agent and Software Architectures: A Comparative Case Study in Proceedings of AAMAS ’02, Bologna, Italy, May 2002.

[3] Goguen, J. A., and Linde, C. Techniques for Requirements Elicitation in Proceedings of RE ’93, San Diego, CA, January 1993, 152-164.

[4] Jennings, N. R., Sycara, K., and Wooldridge, M. 1998. A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1, 7-38.

[5] Jirotka, M., and Goguen, J. (Ed.) Requirements Engineering: Social and Technical Issues. Academic Press, London, UK, 1994.

[6] Kim, M., Lee, S., Park, I., Kim, J., and Park, S. Agent-Oriented Software Modeling in Proceedings of APSEC ’99, Takamatsu, Japan, December 1999, 318-325.

[7] Nuseibah, B., and Easterbrook, S. Requirements Engineering: A Roadmap in Proceedings of The Future of Software Engineering, Limerick, Ireland, May 2000, 35-46.

[8] Sutcliffe, A., and Maiden, N. 1998. The Domain Theory for Requirements Engineering. IEEE Transactions on Software Engineering 24 (3), 174-196.

[9] Zhou, Y., and Pan, Y. 2000. Agent-Oriented Analysis and Modeling. ACM SIGSOFT Software Engineering Notes 25 (3), 36-40.