ANALYSIS AND DESIGN OF A RECOMMENDER SYSTEM WITH AGENTS

Javier Portillo-Rodríguez, Aurora Vizcaíno
Alarcos Research Group – Institute of Information Technologies & Systems, Escuela Superior de Informática
University of Castilla – La Mancha, Ciudad Real, Spain

Juan Pablo Soto, Mario Piattini
Alarcos Research Group – Institute of Information Technologies & Systems, Escuela Superior de Informática
University of Castilla – La Mancha, Ciudad Real, Spain

Keywords: Recommender system, Multi-agent system, INGENIAS.

Abstract: The aim of this paper is to provide a guideline for novel multi-agent systems developers in order to assist them in the development of this kind of systems. Papers frequently focus on describing how systems work, but seldom describe the different steps carried out to attain the final product. We shall attempt to show how the usage of a methodology facilitates the analysis, design and implementation phases, along with how the INGENIAS methodology has helped us to systematically construct a recommender system.

1 INTRODUCTION

Papers related to software agents and multi-agent systems are often focused on explaining how the agents work or the goal of the system for which either the agents or the multi-agent systems have been designed. However, explanations are rarely provided of how the methodological development of this kind of systems should be tackled, i.e. how a methodology should be used to tackle the analysis and design phases of a multi-agent system. The explanation of these methodological aspects might be of use to future developers who have to confront the problem of developing these kinds of systems by using confirmed techniques and methods.

Numerous methodologies focusing on the development of multi-agent systems currently exist. This paper will show the steps involved in developing a multi-agent recommender system. The first step was to study which was the most suitable methodology for our problem, finally INGENIAS methodology was chosen. There now follows a brief description of the system developed. This is a multi-agent system whose goal is to recommend documents in a Community. The system will therefore offer document recommendation, and will take into account both the evaluations that the documents obtain and the trust related to the evaluators. The reason for using the trust parameter to make recommendations is that some communities are often "virtual" as their members may be geographically distributed. This implies a lack of face-to-face communication which affects certain aspects of interpersonal relationships, such as the possible decrease of trust between members and we consider that it is highly important to be able to discover how trustworthy a knowledge source (in this case are the own members) is. This paper is structured as follows. The following section explains the reasons for using the INGENIAS methodology to develop the recommender system (multi-agent system). The analysis and design of this system will be explained in Section 3. Section 4 describes some implementation aspects in relation to the design described in Section 3. Section 4 also shows the tool developed and the recommendation results generated by this tool. Finally, Section 5 presents our conclusions and future work.

2 INGENIAS

In INGENIAS, the general approach used to specify

Portillo-Rodríguez J., Vizcaino A., Soto J. and Piattini M. (2009).
ANALYSIS AND DESIGN OF A RECOMMENDER SYSTEM WITH AGENTS.
In *Proceedings of the 4th International Conference on Software and Data Technologies*, pages 175-178
DOI: 10.5220/0002253801750178

Multi-Agent Systems (MAS) is to divide the problem into more concrete aspects that form different views of the system. The difference between this proposal and others which already exist is how these views are defined and built, and how they are integrated into the MAS development. Each type of view is described by using a meta-modelling language which specifies a kind of grammar with which models are built, known as views. INGENIAS recognizes five meta-models that describe system views.

After studying other methodologies, we developed an analysis with which to identify the advantages of INGENIAS in comparison with the other methodologies. First, we concluded that INGENIAS provides the best support in all the phases of the lifecycle of an agent-based application, including its management. Second, INGENIAS provides a tool support that others do not yet have. For instance, INGENIAS provides a visual language for multi-agent systems definition, which is a process to guide the lifecycle of software development based on the Rational Unified Software Development Process (RUP). In addition, the code generator itself is another tool developed for INGENIAS. It produces codes according to the instructions given by the specification of the system. This tool can be used to generate a code no matter what agent platform is being used (Pavón and Gómez-Sanz 2006).

The analysis and design of the recommender system mentioned in the introduction is described in the following sections.

3 ANALYSIS AND DESIGN WITH INGENIAS METHODOLOGY

Before describing the analysis and design it is first necessary to give a detailed description of the system that it is going to be developed, in this case a multi-agent system that must be able to recommend documents in a comunity.

Software agents must recommend documents to users that will prevent them from making costly searches which, on many occasions, do not generate the expected results. In these communities it is important to know which users are trustworthy document sources. A trust model presented in (Portillo, Soto et al. 2008) is therefore used to make recommendations and to calculate the trust between members. The model then uses these trust values to weight the evaluation values given by users to the documents once they have used them. The evaluations made by trustworthy users will therefore have more importance than the evaluations made by non-trustworthy users.

Once the kind of system to be developed is known, system requirements must be extracted. Moreover, the design requirements specify that each user is represented in the system by one software agent and that the communities will be managed by another software agent.

These requirements are the starting point of the INGENIAS analysis phase in which requirements are used to identify use cases that will be modelled by using interaction models such as the use case *Obtain Recommendation.*

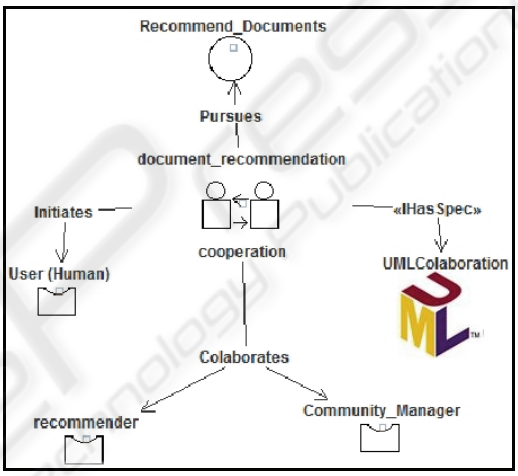

Figure 1: Obtain_Recommendation Use case.

Figure 1 shows an example of the modelling of the *Obtain_Recomendation* use case through the use of an interaction model (called *document_recommendation*). In this model the goal of the use case is to *Recommend Documents*. The roles that participate in the use case are also specified, indicating which one initiates the use case and which one collaborates. In the case of Figure 1 there are three roles: the user (played by a human being), the recommender and the community manager (both of which are played by software agents). It is important to emphasize that these kinds of diagrams are previously modelled by using UML collaboration diagrams, and this is indicated by using the *HasSpec* Relationship.

The task of specifying use cases by using interaction models must be carried out for each of the use cases previously identified. In this case, only one use case model (that which is most closely related to the main goal of the system (recommending documents) is mentioned owing to space limitations.

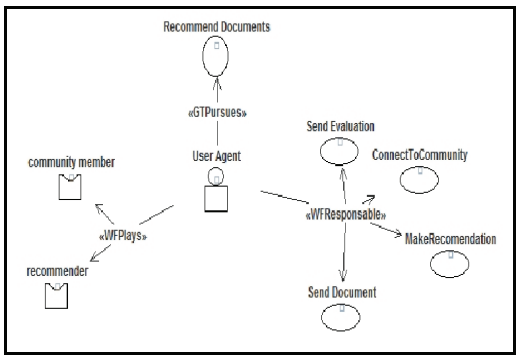

Figure 2: Agent Model for User Agent.

To specify which agents will interact in the systems and which tasks they will carry out it must be used what is termed as the 'agent model' in INGENIAS. These models must be used at the beginning of the analysis phase to identify which kind of agents must be considered in the multi-agent system but, in this case, the agents were identified with the design requirements and these models have therefore not yet been used.

As was mentioned in the design requirements, one type of software agent is needed to represent users in the communities which is able to recommend documents (User Agent) (see Figure 2), and another agent is needed to manage community information (Manager Agent)(not included due to space limitations).

Figure 2 therefore shows that the User Agent has the goal of Recommend Documents and it can play the roles of community member or recommender.

This analysis phase is completed by representing all the identified elements, including the agents, in an organization model which will represent the architecture of the multi-agent model.

One of the most important elements that must be incorporated into the architecture through the organization model are WorkFlows (WFs). WFs are sets of tasks created by grouping the tasks previously identified during the analysis phase.

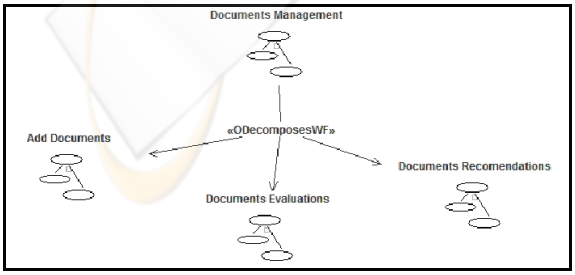

Figure 3: Agent Model for Manager Agent.

WFs are the starting point of the design phase. Two main WFs have been identified in this system:

- **Community Management WF:** This includes tasks related to community management such as registering, connections, etc.
- **Documents Management WF:** This WF is specialized in three SWFs: *Add Documents, Documents Evaluation, Documments Recommendations* (see Figure 3).

Each WF can be divided into different Sub-WorkFlows (SWFs) by analysing which tasks might be necessary to complete the SWF. These tasks will be carried out by the agents.

Once the tasks that are related to each SWF have been identified it is important to establish the sequence of execution, that is, it is vital to indicate which task will be the first to be executed, which will be the second, etc.

Moreover, when setting this sequence, it is necessary to include all the elements (in INGENIAS these are called *Facts*) that are needed for each task to be executed and those elements produced/generated for each task after being executed.

It is necessary to specify not only the sequence of execution but also the interactions generated between agents when executing these tasks in each SWF. This activity can be modelled by using the INGENIAS Interaction Models. These models specify each interaction by identifying a set of Interaction Units that represent each occasion on which an agent communicates with another agent.

By completing this specification for each WF sufficient detail can be obtained to translate these models to the implementation of the multi-agent system by using any agent platform.

The tool constructed by using the previously explained multi-agent system is also shown.

4 IMPLEMENTING THE RECOMMENDER SYSTEM

The main idea behind translating the design of the multi-agent system to implementation in JADE is to consider each SWF as a JADE Behaviour. That is, in each SWF there is a set of tasks, and some of these tasks are executed by one agent and others are executed by another agent. The intention is to codify each set of tasks that an agent executes in each SWF into a Behaviour. Thus, each agent will carry out each SFW by executing the corresponding

Behaviour when necessary. Using this idea we have developed our tool. This tool allows users to search communities in order to register or connect and ask for recommendations. Moreover, the user can obtain information regarding all the active communities to discover which might be of interest to the user.

Therefore, once the user is connected to a community there are several possibilities. The user can send documents to the community by accessing the *Contribution Menu* or solicit documents related to a topic by accessing the *Recommendations Menu*. Figure 4 shows the results of a recommendation where in the left part of the screen there is a list of documents recommended and on the right there is another list with the documents that should be evaluated.

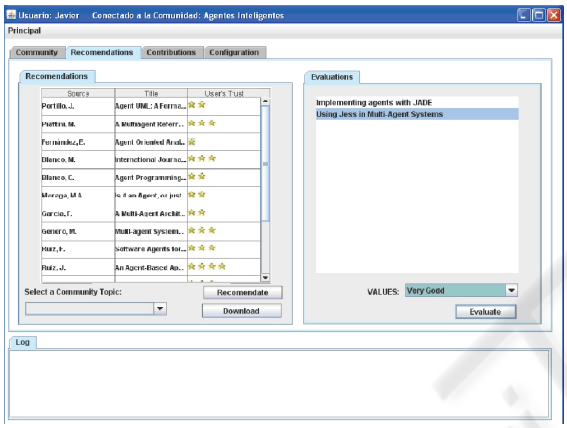

Figure 4: Recommendation results.

In order to evaluate the tool we are carrying out a performance test. The test is oriented towards the Manager Agent's performance. There is only one Manager Agent per Community and it is important to discover whether this agent is able to manage all the incoming messages from User Agents when a high number of users are connected to a Community. The preliminary results indicate that the tool starts to become less efficient when there are more than 800 users consulting at the same time. However, as we recommend this tool to small/medium-sized companies it is expected that the number of users will be less than 800 and that the tool will work efficiently.

5 CONCLUSIONS AND FUTURE WORK

This paper shows how to confront a multi-agent system development by using a systematic method which, in this case, was the INGENIAS methodology. In our case INGENIAS was chosen since it is a complete methodology which is well supported by a design tool. The only problem is that INGENIAS involves the generation of a considerable amount of documentation which is sometimes redundant.

We have explained how to carry out the activities proposed in INGENIAS in the analysis and design phases with the goal of assisting future multi-agent system developers to discover which steps to follow when developing a system. We have also described the different models used for developing a recommender system.

As future work, we intend to explore the possibility of automatic code generation that INGENIAS offers through the latest versions of its Ingenias Development Kit (IDK).

ACKNOWLEDGEMENTS

This work is partially supported by the MELISA project (PAC08-0142-3315) y ENGLOBAS project (PII2I09-0147-8235) Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, in Spain and partially supported by the ESFINGE project (TIN2006-15175-C05-05) Ministerio de Educación y Ciencia (Dirección General de Investigación)/Fondos Europeos de Desarrollo Regional (FEDER) in Spain, and CONACYT (Mexico) under the third author's scholarship grant 206147.

REFERENCES

Pavón, J. and J. Gómez-Sanz (2006). INGENIAS web site. http://grasia.fdi.ucm.es/ingenias/.

Portillo, J., J. P. Soto, et al. (2008). A Model to Rate Trust in Communities of Practice. International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona, Spain.