The third one is the availability of tools, IDK and ACLAnalyser. Current versions of the IDK and ACLAnalyser tools are being revised with the feedback from these applications. One of the main improvements will be in usability, both for normal developers, who want to use the tools as they are, and for tool engineers, who want to adapt INGENIAS tools for specific needs. In this sense, more user-friendly interfaces are being developed and better documentation is being prepared (specially, how-to manuals). Concerning the INGENIAS process, the building of a tool or an integrated framework to assist in the specification and enactment of the MAS Development Process is being done by using Eclipse Process Framework (EPF).

References

1. Juan A. Botia, Juan M. Hernansaez, and Antonio F. Gomez-Skarmeta. On the application of clustering techniques to support debugging large-scale multi-agent systems. In R. H. Bordini et al., editor, *Programming Multi-Agent Systems Workshop at the AAMAS '2006. LNAI 4411*, pages 2219–229, 2007.
2. G. Caire, F. Garijo, J. Gómez, J. Pavon, F. Leal, P. Chainho, P. Kearney, J. Stark, R. Evans, and P. Massonet. Agent oriented analysis using MESSAGE/UML. In *Agent-Oriented Software Engineering. Second International Workshop, AOSE 2001*, volume 2222 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
3. A. Giret, V. Botti, and S. Valero. MAS Methodology for HMS. In: Holonic and Multi-Agent Systems for Manufacturing. In *Second International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005*, volume 3593 of *Lecture Notes in Artificial Intelligence*, pages 39–49. Springer-Verlag, 2005.
4. J. J. Gómez-Sanz. *Modelado de Sistemas Multi-Agente*. PhD thesis, Departamento de Sistemas Informáticos y Programación, Universidad Complutense Madrid, 2002.
5. J. J. Gómez-Sanz and R. Fuentes. Agent oriented software engineering with ingenias. In *Fourth Iberoamerican Workshop on Multi-Agent Systems (Iberagents'2002), a workshop of IBERAMIA'2002, the VIII Iberoamerican Conference on Artificial Intelligence*, pages –, 2002.
6. Ivar Jacobson, Grady Booch, and Jim Rumbaugh. *The Unified Software Development Process*. Addison-Wesley, 1999.
7. J. Pavón, J.J. Gómez-Sanz, and R. Fuentes. The INGENIAS Methodology and Tools. In B. Henderson-Sellers and P. Giorgini, editors, *Agent-Oriented Methodologies*, pages 236–276. Idea Group Publishing, 2005.
8. Juan Pavón, Jorge Gómez-Sanz, Antonio Fernández-Caballero, and Julián J. Valencia-Jiménez. Development of intelligent multi-sensor surveillance systems with agents. *Robotics and Autonomous Systems (Special Issue: CMPI 2006)*, 2007.
9. Candelaria Sansores and Juan Pavón. Agent Based Simulation for Social Systems: From Modeling to Implementation. In *11th Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2005)*, volume 4177 of *Lecture Notes in Artificial Intelligence*, pages 79–88. Springer-Verlag, 2005.
10. Guillermo Vigueras and Juan A. Botía. Diagramas de causalidad semánticos. In *Taller de Desarrollo de Agentes y Sistemas Multi-Agentes (DESMA'07), Congreso Español de Informática (CEDI)*, Zaragoza, Spain, 2007.

Reaching Consensus in a Multi-Agent System

Iván García-Magariño, Jorge J. Gómez-Sanz, and
Jose R. Pérez-Agüera **

Facultad de Informatica, Universidad Complutense de Madrid
ivan_gmg@fdi.ucm.es, jjgomez@sip.ucm.es, jose.aguera@fdi.ucm.es

Abstract. The Delphi process is useful for reaching consensus among a set of experts in a concrete topic. Its application to Multi-Agent Systems may facilitate the integration of different services. Multiple implementations of the same service will potentially provide different results on the same client request. With a Delphi survey process, these different service implementations could appear as a single one of a higher quality than isolated versions. To illustrate its applicability, the paper introduces an implementation of this Delphi survey process as Multi-Agent Systems following the INGENIAS methodology. The problem to be addressed in the paper consists in a community of expert agents providing document relevance evaluation services.

Key words: agent oriented software engineering, multi-agent systems, development

1 Introduction

A Delphi survey is a procedure for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem [1]. From the uses this procedure has, this paper focuses in the consensus agreement capabilities it brings. Reaching consensus implies there are experts providing an opinion about a concrete issue and the possibility of a disagreement among those experts. Each expert is supposed to follow different criteria and use different sources of knowledge. In this context, an external client needs to obtain a consensed opinion about an issue. This implies reaching an agreement among experts.

So far, Delphi processes have been executed by humans, sometimes with some computer assistance [2]. As a novelty, this paper addresses a totally computerised implementation of a Delphi process using Multi-Agent Systems, MAS from now on. Concretely, this paper explores the consensus capabilities of a Delphi process in the context of a document relevance evaluation scenario.

In this scenario, there is a MAS with several Delphi capable agents. Inside, there are several expert agents designed to rate documents according to the preferences of a concrete user. The rating is performed following different procedures

** In alphabetical order

and starting with a different conception of what is relevant to the user. However, by using a Delphi method, an agreed answer results.

The scenario has been constructed with the INGENIAS [3] methodology. Compared to other alternatives, INGENIAS provides a comprehensive notation as well as a set of tools supporting modelling and implementation of specifications.

The paper presents some preliminary results indicating the Delphi approach, as implemented in this paper, improves individual capabilities of expert agents. The construction of the Delphi process is generic enough to allow a high degree of reuse for other domain problems. In concrete, the solution is considered domain dependent only in the questionaire elaboration, filling in, and answer analysis stages.

The paper is structured as follows. First, the Delphi method is presented briefly in section 2. The implementation of Delphi with INGENIAS appears in section 3. Some reflections on how questionaires are elaborated and processed appear in section 4. The evaluation of the results obtained so far is discussed in section 5. To compare this work with existing ones, readers can consult section 6. Finally, section 7 introduces the conclusions.

2 Delphi Method

This method dates back to the fifties. It was created by the RAND corporation in Santa Monica, California. The method is made of structured surveys. It plans several rounds of questionaries which are send to the different involved experts. The results collected can be included partially in a new round of questionaires, but respecting the anonymity of the participants.

This method was created initially for foresight studies, i.e., long-term decisions that guide the policy of a country or a company. Besides forecasting, there are many contexts where the Delphi Method can be applied, like reaching a consensus in a community of experts [4]. The scenario considers several experts discussing about a concrete topic. By using the Delphi method, individual experts are forced to look at the reasons of other experts. This extra information can force experts to reconsider their opinions and reach agreements.

An important part of the Delphi method consists in defining different questionaires which are to be filled in by the different experts. These questionnaires intend to re-orient the initial problem. The re-orientation can be elaborated according to the different answers supplied by experts. Therefore, each questionaire will include pieces of the answers already developed. By the intervention of the questionaire elaborator, it is assumed that the process converges in a single alternative. This mediator role is usually played by a human, though it could be replaced by a computer. This leads to the *the Delphi Conference*, i.e., a computer based Delphi method [2].

The Delphi Process in general is shapeless and its structure depends on the situation. Looking for guidelines, this paper follows the steps and guidelines stated in [5].

3 Representing the Delphi Method with INGENIAS Notation

The Delphi Method devises two main roles: *expert* role, which fills in questionaires, and *monitor* roles, responsible of elaborating questionaires and analysing the answers. There is an additional role, the *client*, which is the one requesting the Delphi. There can be several monitors, at least 1, and several experts, at least 2, in a Delphi process.

Figure 1 shows the main functionality required for implementing a Delphi survey. The *evaluationUC* use case represents a client requesting a service for document evaluation by means of a Delphi survey. The service is provided by an agent playing the *monitor* role. When the *evaluationUC* use case is performed, the *ObtainDocEvG* goal is achieved. This goal represents a future state in the system where a document has been evaluated following a Delphi process. The second use case, *delphiUC*, encapsulates the access to the questionaire filling in service offered by an agent playing the *expert* role. The *monitor* asks an *expert* to fill in a form, following the spirit of a Delphi process. The results are gathered and analysed by the *monitor* who will decide to go again into another round or finishing at the current moment. Like previous use case, this one intends to achieve a concrete goal, the *AnswerQuestG* goal. This goal represents the state of the system reached when an *expert* has filled in the supplied questionaire and a *monitor* has analysed the answer.

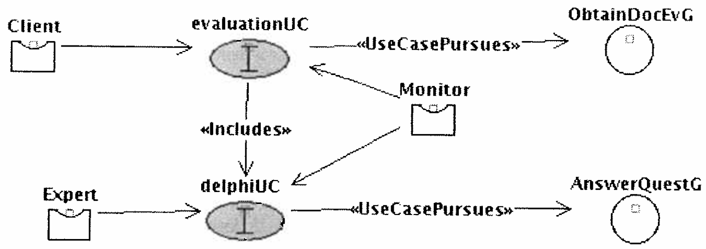

Fig. 1. Main use cases considered in the development of the Delphi process

To satisfy the first goal, *ObtainDocEvG*, an organisation is created, the *Delphi Provider* organisation. This organisation (see Figure 2) is structured into two groups, the *experts* and the *monitors*. In the *experts* group, there will be agents able to play the *expert* role. In this case, agents *ExpertAgent1* and *ExpertAgent2* are responsible of answering the different questionaires delivered by *monitors*. Though in this implementation only two *expert* agents are produced, intuitively, readers will agree that the example can be scaled up with more agents, provided they can implement the *expert* role. The commitment to participate in the deliver of filled in questionaires is translated as associating the goal *AnswerQuestG* to the *expert* agents.

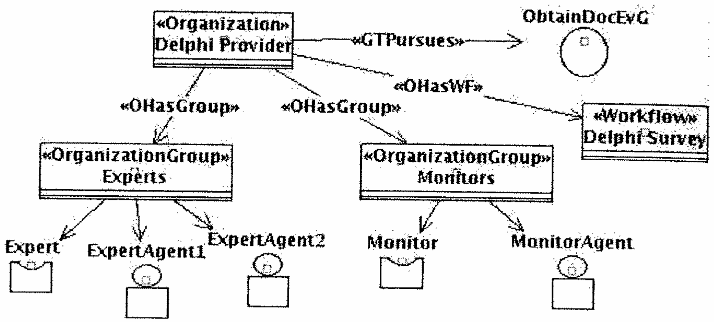

Fig. 2. MAS organization providing the document relevance evaluation

The organisation is able to provide a service by means of the *monitor* role. The service is implemented as a workflow named *Delphi Survey*. Following again Delphi instructions, the method requires at least two round of questionaires. The interaction among individuals in the workflow is controlled by two interactions, *AskingEval* and *DelphiCoop*, whose corresponding protocol appears in Figure 5. The first encapsulates the interaction between the *client* and *monitor* roles to request the evaluation service. The second contains the questionaire elaboration, deliver, and answer gathering activities.

The workflow itself gathers the tasks shown in Figure 3. The workflow presented in Figure 3 starts with a client requesting the service with the task *chooseDoctT*. This task is supposed to provide the document to be evaluated by a *Delphi provider* organisation. The document is received by the *monitor* and a customised questionaire is elaborated with task *InitQuestT*. The questionaire is answered by experts by means of a task *AnsweQuestT*. The answer is processed by the *monitor* with a task *ProcessAnswerT*. As a result of this task, another round can be derived or not. If a new round occurs, the task *CreateOtherQuestT* should be executed. This would force another elaboration of questionaires and a new answer deliver by experts. If no more rounds occur, then the *monitor* delivers the result to the *client*, which processes the evaluation with task *ResultObtainedT*.

Some of these tasks have the responsibility of launching interactions. This is the case of *ChooseDocT*, *InitQuestT*, and *CreateOtherQuestT*. The first creates an interaction of type *AskingEval*, while the second and third create one of type

DelphiCoop. As it will be seen later in Figure 5, the interaction complements the workflow definition by telling what information is passed to each agent and what tasks are expected to be triggered as a result of that information transfer.

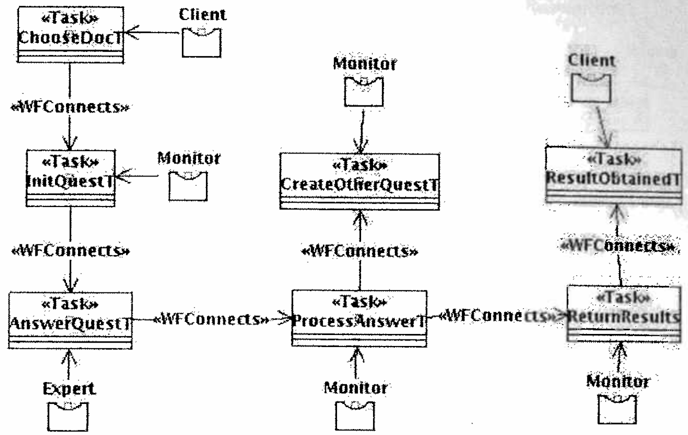

Fig. 3. Overview of the workflow used to implement the Delphi process

A questionaire is represented with a *FrameFact* type, the *QuestToBeAnsweredFF* entity (see Figure 4). This entity has a slot containing the questionaire in form of a string. Readers can assume the questionaire is codified as a string and passed as a slot inside of a *QuestToBeAnsweredFF*. This *QuestToBeAnsweredFF* is consumed in Figure 4 by two different tasks, *AnswerQuestExpr1T* and *AnswerQuestExpr2T*, belonging to two different experts of the organisation, the *ExpertAgent1* and the *ExpertAgent2*. As a result, the tasks produce a *QuestReplyFF* entity with the answer of each expert. As with *QuestToBeAnsweredFF*, *QuestReplyFF* contains the questionaire in form of a string. To perform these tasks, it is necessary the assistance of three pieces of external software, represented in the Figure 4 with *LogGUI*, *ExpertUtils1*, and *ExpertUtils2*. The first acts as a general log to show debug information. The second provides the fill in questionnaire functionality for *ExpertAgent1*. The third does the same for *ExpertAgent2*. These tasks are not included in the workflow from Figure 3 because they are domain specific, i.e., developed ad-hoc to capture concrete means of filling in a questionaire. These tasks would take as input the output of task *AnswerQuestT*, which does belong to the workflow, and would provide outputs for the next tasks in the workflow.

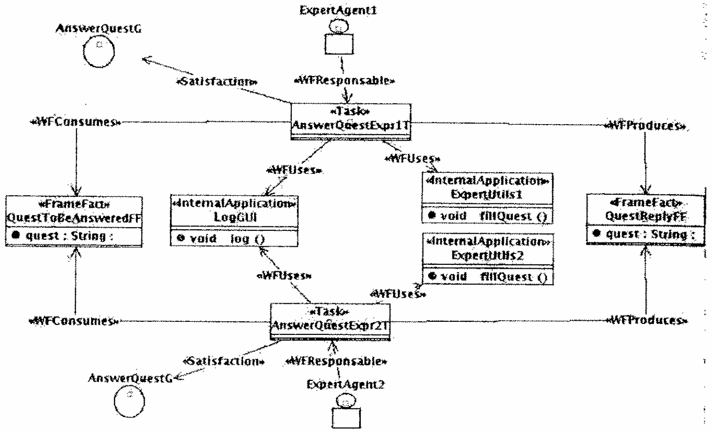

Fig. 4. Tasks representing the answering procedures of individual experts

The protocol for sending questionnaires and receiving answers is presented in Figure 5. The protocol interleaves entities of type *InteractionUnit* with task entities. Each *InteractionUnit* type entity represents a communication between a *Monitor* and an *Expert* role. It has associated an speech act and the information to be transmitted. For instance, the *DistQuest* entity transmits the *QuestFF* entity. When the entity is transferred, the *expert* role is expected to execute *AnswerQuestT*. This task will produce results, concretely an *QuestToBeAnsweredFF* entity, that need further processing to obtain the information the protocol requires to continue, a *QuestReplyFF* entity. In this paper, it is assumed this extra processing is provided by tasks from Figure 4, which implement the expert criteria. Once received the answer from the expert, the agent playing the *monitor* role either finds a consensus or decides to initiate another round of questionnaires. The first case implies engaging into a *Agree* interaction unit and sending an *AgreFF* entity containing the consensus. In the second case, the task *CreateOtherQuestT* creates another instance of the interaction following the protocol from Figure 5. Also, it informs the expert that there was not an agreement by transmitting an *NotAgreeFF*.

So far, this description is generic enough to fit into most applications of Delphi. The problem specific part is the elaboration of questionaires and the analysis of the answers, reviewed briefly in Figure 4. This will be considered in more detail into the following section.

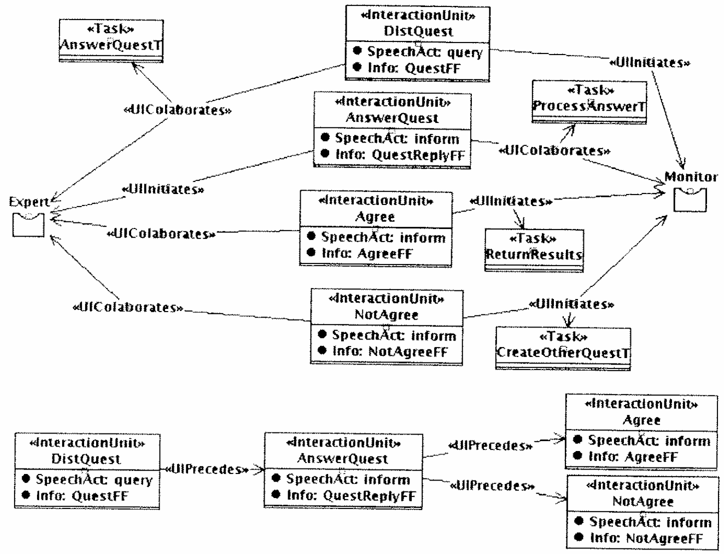

Fig. 5. Protocol for passing a questionaire and receiving the answer

4 Elaborating Questionaires and Analysing Answers

Each *expert* agent encapsulates a collection of documents defining the interests of the user. Each one of these experts is offering as service filling in a questionaire related to the relevance of information with respect this collection. The service assumes a regular structure in the questionaires. In concrete, it expects questions to be sets of words. Experts are expected to rate the relevance of these sets of words with a rational number between 0 and 1; and include additional words which the expert considers related to the ones supplied. These words are terms extracted from the top ranked documents returned by the first-pass retrieval on the expert profile. At this moment, the experts use Lucene[1] Vector Space Model [6] implementation[2] for question relevance evaluation.

The elaboration of questionaires distinguishes between first round questionaires and second round questionaires. The main difference refers to the inclusion of additional information in second round questionaires. This is due the requirements of Delphi method, as seen in section 2. The idea is similar to query ex-

[1] http://lucene.apache.org/java/docs/
[2] http://lucene.apache.org/java/docs/scoring.html

pansion process. In fact, pseudo-relevance feedback [6] is applied here to extract the terms candidates to become comments.

A first round questionaire contains questions made of words of sentences extracted from the document. To extract each sentence from the document, the implementation uses Lingpipe library[3]. After extracting the sentences from the original documents, these are analyzed with term frequency - inverse sentence frequency [7] algorithm to select the phrases containing the most informative terms from the document. Answers to each question will be a relevance value and a set of related words, the comments.

A second, and following, round questionaire is similar to the first round one. As a novelty, it incorporates comments from all experts, i.e., those additional words considered by these experts as relevant in the last round. Selecting which words will appear is a problem of weighting each returned comment with a Rocchio algorithm. If the value returned by this algorithm exceeds a threshold of 0.2, a value determined during the experimentation of this paper, it is considered as relevant and included in the next round of the questionaire.

The definition of the second round questionaire implies each round carries out a new expansion on the questions contained in the questionaire. Also, it is expected that each round questionaire returns different relevance values, as more terms are included in the questions. The rounds will follow until a consensus or a certain number of rounds is reached. The consensus is reached when there is a round when the mean of all relevances returned by all experts to all questions exceeds a concrete threshold. It is still a matter of study to determine if the method really ensures experts' opinion converges to either 0 or 1 before the limit of rounds is reached. During the experimentation, this convergence has occurred, but it requires further study the conditions under this property is satisfied.

5 Evaluation

To test the approach, the experiment uses a set of tests provided by CLEF (Cross-Language Evaluation Forum) [8]for the Spanish language. The collection and tests used come from EFE94. This document collection came from the international news agency EFE, from all the news received during 1994 and consists of 215.738 documents stored in files with SGML format.

This collection was pre-processed by extracting documents belonging to the relevance assessments for the years 2001 and 2002. These relevant documents have been used to define expert profiles. In the second step, a battery of documents has been built to be evaluated by the system. This battery has been built using a sub-set of relevant and non-relevant documents contained in the relevance assessment for the years 2001 and 2002 and avoiding overlapping between expert profiles and battery tests.

The evaluation of the performance of the MAS system using DELPHI uses the well-know measures of Precision and Recall. Precision is defined as the ratio

[3] http://www.alias-i.com/lingpipe/

of good assessments (relevant/non-relevant) selected to total number of assessments. Recall id defined as the ratio of relevant documents selected to total number of relevant documents available.

	Without DELPHI	With DELPHI	Improvement
Precision	0.86	0.92	+6.5%
Recall	0.84	0.96	+12.5%

Fig. 6. Evaluation of the Delphi Method

In these preliminary results (Figure 6), it can be observed the use of Delphi method achieved an improvement of the performance, greater than the one achieved without cooperation among agents. On the other hand, a very good general performance is obtained, because our system is capable to detect on average, 9 out of every 10 relevant documents.

6 Related Work

The problem of reaching consensus in Multi-Agent Systems is not radically new. Negotiation, for instance, can be seen as a decision-making problem where two or more parties try to find a consensus [9]. So far, approaches to this kind imply complex theories, like game theory. The solution addressed in this paper is not at the same level, since it is applied mostly to humans and requires less formal methods.

Another related work is from Hannebauer [10]. In this work, disagreement between different problem solving methods is solved by means of choosing the most frequent answer. In this paper, the approach is different in the sense that opinion from experts may be interpreted in different ways as their answers to the questionaires are collected. In fact, experts are allowed to change their mind when more information arrives.

The diversity of answers can be handled as well by using results from trust and reputation models [11]. The difference between these approaches can be find in the final goal of trust and reputation models: the interest in finding only one provider of the service which can be trusted enough. With Delphi, the problem is not finding one trusted service provider, but finding ways in which all service providers can be accounted.

7 Conclusions

The Delphi Method provides an original way of reaching agreements in communities of agents. The approach is rather intuitive, since it bases in the deliver of questionaires. The difficult part of this method consists in determining which questions to appear in the questionaire and a proper method of analysing answers. Here, a solution has been given for a document relevance evaluation problem. As it is now, the work is partially reusable. The process and protocols could be applied to other problems. Nevertheless, the questionaire elaboration and analysis are still domain dependent tasks. Authors expect to provide more guidelines for dealing with this issues in future work.

The performance of this Delphi methods against other consensus methods is still to be studied. So far, preliminary performance results using Delphi indicates an improvement over single experts performance. Nevertheless, further experiments mixing more relevance evaluation algorithms as well as more heterogenous sources of information are needed.

Acknowledgements This work has been supported by the project Methods and tools for agent-based modelling supported by Spanish Council for Science and Technology with grant TIN2005-08501-C03-01, and by the grant for Research Group 910494 by the Region of Madrid (Comunidad de Madrid) and the Universidad Complutense Madrid.

References

1. Linstone, H., Turoff, M.: The Delphi Method: Techniques and Applications. Addison-Wesley Pub. Co., Advanced Book Program (1975)
2. Turoff, M., Hiltz, S.: Computer Based Delphi Processes. Gazing into the Oracle. The Delphi Method and its Application to Social Policy and Public Health, Jessica Kingsley Publishers, London (1996) 56–85
3. Gomez-Sanz, J.J., Fuentes, R., Pavon, J.: The INGENIAS Methodology and Tools. In: Agent-oriented Methodologies. Idea Group Publishing (2005) 236–276
4. Dalkey, N., Helmer, O.: An Experimental Application of the Delphi Method to the Use of Experts. Management Science **9**(3) (1963) 458–467
5. Cuhls, K.: Delphi method. Technical report, Fraunhofer Institute for Systems and Innovation Research (2003)
6. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
7. Larocca Neto, J., Santos, A.D., Kaestner, C.A.A., Freitas, A.A.: Document clustering and text summarization. In: Proceedings of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining, London (2000) 41–55
8. Peters, C., Braschler, M.: European research letter: Cross-language system evaluation: The clef campaigns. JASIST **52**(12) (2001) 1067–1072
9. Rosenschein, J., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press (1994)
10. Hannebauer, M.: Multi-phase consensus communication in collaborative problem solving. In: Proceedings of the Third Workshop on Communication-based Systems, Kluwer (2000) 131–146
11. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1 (2002) 475–482

Agent Oriented Programming applied to story telling: UApolis as a case of study

Abel Bernabeu, Juan Reverte, Francisco Gallego and Faraón Llorens

{abelbg, jreverte, fgallego, faraon}@dccia.ua.es
University of Alicante
Alicante, Spain

Abstract. In this paper we will look at the process of creation a of an interactive story from the perspective of an industrial process. We have chosen as case of study a promotional computer game currently under development in the University of Alicante: UApolis. Our purpose is to find a story specification system able to cover the needs of a computer game adventure. As an outcome of our observations we propose the reader a very concrete software architecture inspired in Agent Oriented Programming well suited for stories specification.

1 Introduction

In December of 2006 the Vicechancelor of Technology and Educative Innovation of the University of Alicante proposed us the development of a computer game adventure that could achieve two goals: contribute to our corporative image and, of course, being pioneers offering the future students some information that they previously received by means of more traditional medias such as leaflets or by surfing through our institutional web site. The project (UApolis) was really amazing for us from the perspective of changing the way in which the communication with the students was done, but being people with a technical orientation as most of us were, we soon noticed that the project was challenging from a technical perspective too.

The main problem we faced was how to make an specification of our interactive story. There were some proposals available in the literature about interactive story telling, but we needed to choose one and discard the others as soon as possible in order to release our project in time (in this sense our constraints were more or less those that you could expect in the commercial computer-games industry).

We think that the presence of a constraint similar to the limited time-to-market frequently present in commercial computer games development adds some extra value to the observations collected in this paper.

1.1 Previous proposals

The formal study of story telling is not a new discipline and one could probably find surprising the seminal work of the Russian author Vladimir Propp (1885-1970), who studied semantic-patterns present in folktales trying to find some building blocks commonly repeated all over them [1]. Some of these story patterns included: mark (the hero