Multi-Level Role Modeling in Multi-Agent Systems

Viviane Torres da Silva Carlos José P. de Lucena

{viviane,lucena}@inf.puc-rio.br

PUC-RioInf.MCC 05/02 Abril, 2002

Abstract. This paper proposes a method to help discover the agents that will be part of a multi-agent system based on the definition of the system. This method is based upon the decomposition and specialization of the system goals together with the decomposition and specialization of the system into entities. The goals are associated with the entities, which play roles by trying to achieve the goals and enter into relationships with other entities. To help and validate the refinement process, the method describes rules that ensure how the goals, entities, roles and relationships defined in a certain stage of refinement will appear in the subsequent stages of the process.

Keywords. Agents, multi-agents systems, modeling, decomposition, specialization, refinement

Resumo. Este trabalho propõe um método para ajudar o design a identificar os agentes que irão fazer parte do sistema multi-agentes baseando-se na definição do sistema. O método é baseado na decomposição e especialização das metas do sistema juntamente com a decomposição e especialização do sistema em entidades. As metas são associadas a entidades, que desempenham papéis e participam de relacionamentos com outras entidades ao tentar atingir suas metas. A fim de ajudar na validação e no processo de refinamento, o método descreve regras que asseguram como as metas, entidades, papéis, e relacionamentos definidos em um certo nível de refinamento irão aparecer nos subseqüentes níveis do processo.

Palavras-chaves. Agentes, sistemas multi-agentes, modelagem, decomposição, especialização, refinamento

1. Introduction

The agent paradigm brought with it a new concept of software engineering [11]. The agents are seen as autonomous and adaptive entities that play roles and relate to other entities to achieve their goals. This new paradigm introduced the need for developing new methodologies [2][10][12][08][3][13] for software modeling and for creating new tools [5][9] that would help the modeling and implementation process of multi-agent systems (MAS). Nevertheless, the existing methodologies do not effectively encompass the system agent discovery phase that should be based on the description of the problem and on the analysis of the application domain. The agent discovery phase is necessary because the description of the problem normally does not mention what are the system agents and what roles they should play. Just as occurs in object orientation where the description of the systems does not describe what will be the classes nor all the objects that should be modeled and implemented, the description of the systems that will be implemented as a MAS does not always describe which will be the system agents. MASs are not always simulations of the real world, where the players and their roles are already defined and can be modeled as agents fulfilling a role.

DeLoach in [2] described the MaSE methodology whose first level, design domain level, proposes the identification of the types of the agents based on role modeling [7][10], use case and collaboration diagrams [1]. The use case and collaboration diagrams are not convenient for modeling MAS since they are diagrams that have been developed specifically for object orientation. Thus, adaptations would be necessary to effectively use these diagrams for modeling agents. For its part, the use of role modeling [7][10][12] in an isolated fashion also is not appropriate for discovering the system agents. In these approaches it is not clear how the roles are discovered. Thus, since it is difficult to identify the agents it also is difficult to identify the roles. With an increase in the complexity of the problems and the distancing from the real world, it is becoming more difficult to identify the roles and the system agents. Kendall in [7], besides not making it clear how to discover the system roles bead on the description, do not make clear how the roles identified are related to the agents, that is, given the set of roles, neither is it clear which agents will play which roles.

In [10][12], Wooldridge et al. presented the GAIA methodology that proposes divide the MAS modeling process into two stages: analysis process and design process. The objective of the analysis process is to develop an understanding of the system and its structure. In order to do so, the system is seen as a collection of roles that are played by agents within a society. Wooldrige et al. propose that the roles that are played in the system typically will correspond to individuals, departments and the organization itself. Nevertheless, this association only is valid when the MASs are simulations of the real world. Another problem with this methodology is associated with the roles of the agents. The methodology limits itself to say that normally an agent plays a role but could play more than one role if they were associated. And moreover, the methodology mentions that the association with one or more roles will depend upon the trade-off between coherence and efficiency, but it also is not clear why this is so.

The methodology described in [8] proposes the use of decomposition of the systems, based upon key roles of the application domain. According to the methodology, the identification of the roles and their relationships guide the specification of the hierarchy of the classes of the agents. The methodology is divided into two stages: external development and internal development. The first step of external development, destined for discovering the agents and how these communicate with each other, proposes to identify the roles of the application domain and, then, relate these to the class hierarchy of the agents. Thus as in previous ones, this methodology is based upon the use of roles for the identification of the agents. However, also like in previous methodologies, it does not show how the roles are identified, only mentioning that the roles may be in various dimensions, such the organizational and functional.

In [3], the proposed methodology describes two phases for the MAS modeling: the phase for the discovery of the agents, which is related to our proposal, and the definition phase. The discovery phase guides the identification of the agents and their high level behavior through the trace of the application scenarios. The methodology suggests that the agents be discovered based upon analysis of the substantives that appear in the description of the problem. The agent candidates thus would be active entities. Since the MASs are indicated for solving complex systems [6], the analysis of the existing substantives in the description of the problem is not appropriate for discovering the system agents, as it is not sufficient for discovering the system objects or classes in the case of object-oriented systems.

The proposal that is closest to ours is found in [13]. This proposal divides the MAS modeling process into three phases: analysis of the roles, design of the agents and implementation. The objective of the first phase — the phase that is related to our work — is to determine and specify the roles played in the system as of the construction of a workflow that is based upon the identification and decomposition of the goals, which differentiates the previously mentioned methodologies. Only after the identification of the roles, which was done from the decomposition of the goals, are the agents identified. The roles are given to the agent according to the agent type. Our approach differs in relation to this proposal in that it considers the system to be a unique entity where not only are its goals analyzed but also the relationships with entities that are external to the system and which roles it plays upon relating to these entities. At each level of decomposition there is identification of the sub-goals, of the entities that will achieve the sub-goals, which are sub-entities of the previous ones, of those roles and of the relationships with the other entities. Through the decomposition and specialization of the goals and of the system entity, and the identification of the roles of the entities, we can identify the agents, the roles and the relationship between them. Thus, it is believed that the identification and description of the agents is facilitated, given that the agents will be the entities that will appear at the end of the decomposition process.

In order to help the process of decomposition and specialization of the goals and the entities, we defined a series of refinement rules that must be respected at each level of decomposition. The objective of the rules is to ensure that what was defined at a given level of decomposition will be maintained in all of the subsequent levels.

In the section that follows we present a detailed description of the proposed method. In Section 3, we present the rules for the refinement process. In Section 4 we present a modeled example using the method and the refinement rules. In the final section we describe some future work and the conclusions.

2. Proposed Method

The objective of the method proposed in this paper is to help the designer of the MAS to discover which are the agents that should be modeled. The method is based upon decomposition and specialization of the system goals and the association of these goals to entities that are encountered as the decomposition and specialization of the system entity.

During the refinement process, each goal that is identified is associated with an entity that must achieve the goal. Since the entities play roles and relate with other entities when trying to achieve their goals, the method also identify the relationships and roles of the entities in all of the refinement process levels. This characteristic of the method is expressed in the title of the paper: multi-level role modeling in MAS.

The application of the method generates a sequence of models and tables that are built based upon the refinement of the goals and of the entities of the system, where the entities modeled on the last model are the system agents. The last model will contain the agents' goals, the agents' roles and the relationships that exist between them. The number of models generated in the refinement process will depend upon the complexity of the system and how much more complex the refinement should be in order to find the system agents. The method is divided into four steps, as described below.

1st Step:

In this step, the system is considered to be a unique entity (unit) that possesses a single goal or set of goals and that plays roles and relates to external entities in order to meet its goals. Thus, during this step the goal or set of goals of the entity should be identified along with its roles and relationships with external entities.

As a result of this step, one table and a model should be produced. The table must contain the external entities with which the entity relates, the description of each relationship and the roles played by the entity upon relating with the external entities. The model generated in this step should contain the entity, its roles and the relationships with the external entities.

2nd Step:

The objective of the second step is to sub-divide the entities identified in the previous step into sub-entities, more specialized, capable of working together to achieve the goals defined for the entities. Thus, the sub-entities of a given entity will be related to the sub-goals that were derived from the entity's goal. This is done in the following way: (i) in the event there is just one goal defined for an entity from the previous step, the goal should be sub-divided into sub-goals (goal decomposition/specialization) and associate sub-entities (decomposition/specialization of the entities) to the sub-goals. (ii) in the event there is more than one goal described in the previous step, associate a sub-entity to at least one goal. A sub-entity can be related to more than one goal. Each sub-entity is responsible for reaching the sub-goal related to it.

The result of his step is two tables in the case of (i) and one table in the case of (ii), for each entity identified in the previous step. In the case of (i), the first table relates the set of goals of the entity with the sub-goals, while the second table relates the sub-goals to the set of new system entities (sub-entities). In the case of (ii), the table relates the set of goals of the entity to the set of new systems entities (sub-entities).

3rd Step:

This step must be applied to each entity defined in the previous step. For each entity identified, a sub-goal or a set of sub-goals must be defined for the current goal. Next, it is necessary to identify the relationships that an entity has with the other system entities and the external entities that let it achieve its goals. Moreover, the roles played by the entity should be associated.

Similar to Step 1, one table for each entity identified in the previous step, and one model should result from this step. The table must contain the external and internal entities of the system with those the entity relates to, a description of each relationship and the roles played by the entity upon relating to the entities. For its part, the model must take into consideration all of the entities, their relationships with the other entities and their roles.

4th Step:

Step 4 defines the condition of the specialization/decomposition of the system entities. While the specialization/decomposition of the entities and their goals have not yet reached the end, Steps 2 and 3 should be executed consecutively, permitting the generation of new sub-goals and sub-entities. In this fashion, the method foresees a cycle of steps 2 and 3 for the refinement of the entities and goals. Thus, the model generated in Step 1 is refined in the sequence of Step 3, which can occur while the refinements of the goals and of the entities has not yet ceased. The system agents are in the last model generated in the refinement.

The decomposition/specialization of the goals and the entities should cease when the following conditions have been identified:

1. When an agent can achieve the set of goals defined for each system entity; that is, the set of goals is sufficiently detailed so that the basic characteristics of each system agent may be identified.
2. When the set of goals of the model of the entities has approached the level of detail described in the system; that is, when it is possible to relate the goals identified for the entities with the set of characteristics described in the system.

Note that the details about how the goals will be achieve through the agent's plans and tasks with the help of its beliefs will not be described in this method. This method is designed to identify the system agents based upon the identification of the goals and of a set of activities. The method that is proposed must be followed by a methodology that approaches the design of the intra-agent part by detailing its plans, tasks and beliefs, relating them to the already identified goals of the agent.

3. Refinement Rules

The purpose of the refinement rules is to help the decomposition and specialization process of the models generated as of the proposed method. The rules were defined in order to ensure that the properties[1] described in the highest level model of abstraction were present during the entire refinement process until the lowest level model of abstraction, last model, where the agents are described. Furthermore, the rules intend to demonstrate how the properties can appear in the other levels of the model.

Goals and entities

As the method itself suggests, the goals and the entities must be detailed and specialized during the refinement process. Given a goal related to an entity, it can be decomposed into sub-goals that must be associated with the sub-entities of the entity to which the goal relates. Furthermore, more than one sub-goal maybe associated with the same entity. The sub-goals that refer to a goal of a given entity may not be associated with sub-entities of another entity that is not that to which the goal is related. It is possible to associate more than one sub-goals to the same entity.

Suppose $A(M)$, that is entity A has a goal of M, $M=\{M_1,M_2,M_3\}$, that is, goal M can be sub-divided into sub-goals M_1, M_2 and M_3 and $A=\{A_1,A_2\}$, that is, entity A can be sub-divided into sub-entities A_1 and A_2:

$$\begin{aligned} & A(M) \land M=\{M_1,M_2,M_3\}\land\}\land A=\{A_1,A_2\} \rightarrow \\ & (A_1(M_1)\land A_2(M_2)\land A_2(M_3))\lor \\ & (A_1(M_2)\land A_2(M_1)\land A_2(M_3))\lor \\ & (A_1(M_3)\land A_2(M_1)\land A_2(M_2))\lor \\ & (A_2(M_1)\land A_1(M_2)\land A_1(M_3))\lor \\ & (A_2(M_2)\land A_1(M_1)\land A_1(M_3))\lor \\ & (A_2(M_3)\land A_1(M_1)\land A_1(M_2)). \end{aligned} \tag{1}$$

Roles

With regard to the roles of the entities, it should be ensured that the roles identified in the entities will be played by their sub-entities; that is, given a role played by an entity, at least one sub-entity will play it. In refinement, new roles also may be created.

Suppose $A(P_1,P_2)$, that is A plays roles P_1 and P_2, and $A=\{A_1,A_2\}$, that is, entity A may be sub-divided into A_1 and A_2.

$$\begin{aligned} & A(P_1,P_2)\land A=\{A_1,A_2\} \rightarrow \\ & (A_1(P_1)\land A_2(P_2))\lor \\ & (A_2(P_1)\land A_1(P_2))\lor \\ & (A_1(P_1)\land A_1(P_2))\lor \\ & (A_2(P_1)\land A_2(P_2)). \end{aligned} \tag{2}$$

[1] Property means goals, entities, roles and the relationships that appears in the model.

Relationships

The relationships defined by the entities of the highest level must be complied with by the sub-entities of the lowest level. If an entity A relates with an entity B then the sub-entities of A and of B must relate with each other; that is, there must exist at least one sub-entity of A that relates with at least one sub-entity of B.

Besides ensuring that the relationship is maintained, we also must ensure the semantic of the relationship, that is when entity A relates to entity B this relationship means $X \wedge Y$ then the relationships between the sub-entities of A and the sub-entities of B must also mean $X \wedge Y$. The semantic of the relationship will be specialized, maintaining the definition mentioned in the highest level of abstraction.

Suppose $R(A,B)= X \wedge Y$, that is, A relates to B and the relationship of A with B has the semantic $X \wedge Y$; $A=\{A_1,A_2\}$, that is, A_1 and A_2 are sub-entities of A; and $B=\{B_1,B_2\}$, where B_1 and B_2 are sub-entities of B.

$$\begin{gathered}
(R(A,B)=X\wedge Y)\wedge(A=\{A_1,A_2\})\wedge(B=\{B_1,B_2\}) \rightarrow \\
(R(A_1,B_1)=X\wedge Y) \vee \\
(R(A_1,B_2)=X\wedge Y) \vee \\
(R(A_2,B_1)=X\wedge Y) \vee \\
(R(A_2,B_2)=X\wedge Y) \vee \\
((R(A_1,B_1)=X\wedge R(A_2,B_1)=Y)\vee(R(A_1,B_1)=Y\wedge R(A_2,B_1)=X))\vee \\
((R(A_1,B_1)=X\wedge R(A_2,B_2)=Y)\vee(R(A_1,B_1)=Y\wedge R(A_2,B_2)=X))\vee \\
((R(A_1,B_2)=X\wedge R(A_2,B_2)=Y)\vee(R(A_1,B_2)=Y\wedge R(A_2,B_2)=X))\vee \\
((R(A_1,B_2)=X\wedge R(A_2,B_1)=Y)\vee(R(A_1,B_2)=Y\wedge R(A_2,B_1)=X)).
\end{gathered} \tag{3}$$

4. Case Study

The method that was proposed was evaluated through three case studies. In this paper we will present the application of the MAS Portalware [4], mentioning each step of the method. Portalware is a Web environment that supports the construction of news portals. The portal editor is responsible for selecting the topics of the articles that will be produced, while the content suppliers are responsible for producing the content, that is the articles, based upon the topics that were indicated. The reviewers are responsible for examining the articles produced by the content suppliers, and changing them whenever they believe it is necessary. The editor of the newspaper distributes the topics of the articles to the content suppliers and distributes the articles to the reviewers according to a set of topics of interest to them. The content suppliers and the editors also possess a maximum number of tasks that they can carry out and that the editor does not know about. In the event a content supplier is not found to produce the content of an article, then the editor is responsible for producing the content. In the event a reviewer is not found to review an article, then the editor is responsible for review it.

1st Step: On this step, one should define the system as an entity and identify its goals, its roles and the relationships with external entities, as shown in the next table and in Fig. **1**.

Entity: Portalware		
Goal(s): Build news portals		
Role	**Relationship**	**Entity**
Topic receiver	entity receives information about articles to be produced from…	(e)Editor
Content receiver	entity sends information about articles to be produced to… and receives the article from…	(e) Content supplier
Review receiver	entity sends articles produced to… and receives reviewed articles from…	(e)Reviewer

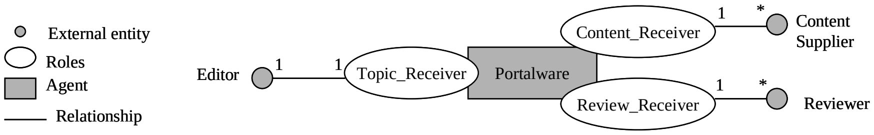

Fig. 1 – 1st model generated from the Step 1 of the method

2nd Step: On next two table is the identification of the sub-goals and of the sub-entities of the system entity.

Entity: Portalware	
Goal	**Sub-Goal**
Build news portals	Manage article editing
	Provide for article production
	Provide for article reviewing

Sub-entities of the entity: Portalware	
Goal	**Entity**
Manage article editing	Editing manager
Provide for article production	Articles manager
Provide for article proofing	Reviewing manager

3rd Step: For each sub-entity defined in the 2nd step, it is one table that identify its goals, its roles and the relationship with other entities. At the end of his step, is a more detailed model than the preview one, Fig. 2.

Entity: Editing Manager		
Goal(s): Manage article editing		
Role	**Relationship**	**Entity**
Topic receiver	entity receives the topics from…	(e)Editor
Article client	entity sends along topics to… entity receives the articles from…	Articles manager
Reviewing client	entity sends along articles to… entity receives revised reviews from…	Reviewing manager

Entity: Articles Manager		
Goal(s): Provide for article production		
Role	**Relationship**	**Entity**
Article supplier	entity receives the topics from… entity sends along the articles to…	Editing manager
Article receiver	entity sends along the topics to… entity receives the articles from…	(e)Content supplier

Entity: Reviewing manager		
Goal(s): Provide article reviewing		
Role	**Relationship**	**Entity**
Reviewing supplier	entity receives the articles from… entity sends along the reviewing article to…	Editing manager
Reviewing receiver	entity sends along the articles to… entity receives the reviews from…	(e)Reviewer

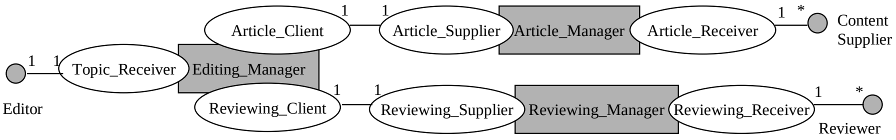

Fig. 2. – 2nd model generated the of Step 3 of the method

4th Step: Cycle in the 2nd Step: For each entity defined in the preview step, it is two tables. One table identifies the sub-goals of the entity and the other table relates each sub-goals to an sub-entities of the entity.

Entity: Editing Manager	
Goal	**Sub-goal**
Manage the editing	Pass along the topics and collect the articles
	Pass along the articles and collect the reviews
	Produce the article (if there is no content supplier for the topic or available)
	Produce the review (if there is no reviewing for the topic or available)

Sub-entities of the entity: Editing Manager	
Goal	**Entity**
Passes along topics and collects the articles	Distributor
Passes along the articles and collects the reviews	Distributor
Produce article	Representative_Editor_ContentSupplier
Produce review	Representative_Editor_Reviewer

Entity: Article Manager	
Goal	**Sub-Goal**
Provide for article production	Distribute the topics according to the preferences of the Content Suppliers
	Collects the articles if Content Suppliers available

Sub-entities of the entity: Article Manager	
Goal	**Entity**
Distribute the topics according to the preferences of the Content Suppliers	Distributor_Topics
Represent a Content Supplier collecting the articles if Content Supplier available	Representative_ContentSupplier

Entity: Reviewing Manager	
Goal	**Sub-Goal**
Provide the reviews of the articles	Distribute the articles according to preferences of the reviewers
	Collect the reviews if reviewers available

Sub-entities of the entity: Reviewing Manager	
Goal	**Entity**
Distribute the articles according to the preferences of the reviewers	Distributor_Articles
Represent the reviewers collecting the reviews of the articles if reviewers available	Representative_Reviewing

4º Step: Cycle in 3rd Step: For each sub-entity, this steps identify its goals, its roles and the relationships with other entities.

Table 1. The next three tables presents the sub-entities of the Editing Manager entity.

Entity: Distributor		
Goal(s): Passes along topics, collects the articles, passes along the articles, collects the reviews, produces the article and produces the reviews		
Role	**Relationship**	**Entity**
Topics receiver	entity receives topics from editor from…	(e)Editor
Article client	entity passes along the topics to… entity collects the articles from…	Article Manager
Reviewer client	entity passes along the articles to… entity collects the reviewers from…	Reviews Manager
Article client	entity passes along the topics to… entity collects the articles from…	Representative_Editor_ContentSupplier
Reviewer client	entity passes along the articles to… entity collects the reviewers from…	Representative_Editor_Reviewer

Entity: Representative_Editor_ContentSupplier		
Goal(s): Produce article		
Role	**Relationship**	**Entity**
Article supplier	entity receives the topics from… entity passes along the articles to…	Distributor
Article receiver	entity passes along the topics to… entity collects the articles from…	(e) Editor
Entity: Representative_Editor_Reviewer		
Goal(s): Produce review		
Role	**Relationship**	**Entity**
Review supplier	entity receives the articles from… entity passes along the reviews to…	Distributor
Review receiver	entity passes along the articles to… entity collects the reviews from….	(e) Editor

Table 2. The next two tables presents the sub-entities of the Article Manager entity.

Entity: Distributor_Topics		
Goal(s): Distribute the topics according to the preferences of the content supplier		
Role	**Relationship**	**Entity**
Article supplier	entity receives the topics from… entity supplies the articles to…	Editing Manager
Article client	entity only supplies the topics to… that are in accordance with the preferences of the content supplier entity receives the articles from…	Representative_ContentSupplier

Entity: Representative_ContentSupplier		
Goal(s): Represent a content supplier collecting the articles if content supplier available		
Role	**Relationship**	**Entity**
Article supplier	entity receives the topics from… entity supplies the articles to…	Distributor_Topics
Article receiver	entity supplies the topics to… if content supplier available entity receives the articles from…	(e)Content Supplier

Table 3. The next two tables presents the sub-entities of the Reviewing Manager entity.

Entity: Distributor_Articles		
Goal(s): Distribute the articles according to the preferences of the reviewers		
Role	**Relationship**	**Entity**
Review supplier	entity receives the articles from… entity supplies the reviews of the article to…	Editing manager
Review client	entity only supplies the articles that are in accordance with the preferences of the reviewers to… entity receives the reviews from…	Representative_ Reviewer

Entity: Representative_Reviewer		
Goal(s): Represent a reviewer collecting the reviews if reviewers available		
Role	**Relationship**	**Entity**
Reviews supplier	entity receives the articles from… entity supplies the reviews to…	Distributor_ Articles
Reviews receiver	entity supplies to… the articles if the reviewer available entity receives the reviews from…	(e) Reviewers

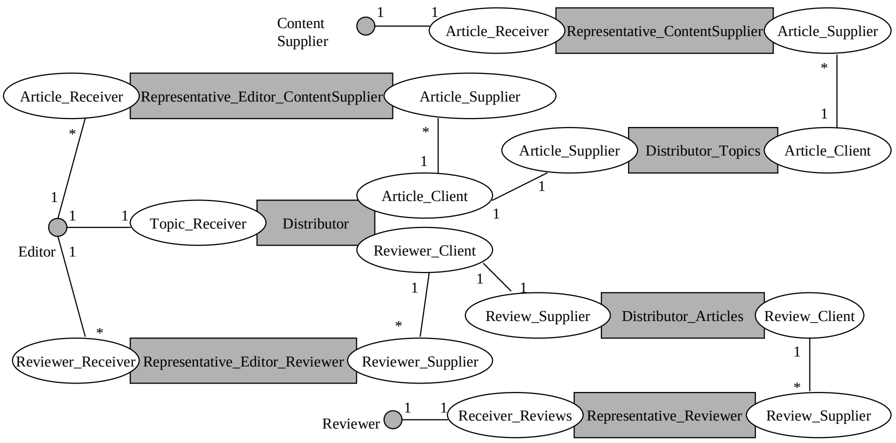

Fig. 3 – 3rd and last model generated form the Step 3 in the cycle of Step 4 of the method

Refinement Rules

Below we will associate each one of the refinement rules with a demonstration of how they run in the Portalware example.

1. Rules (1) The sub-entities of the entity "Editing Manager" are responsible for achieving the sub-goals of the "Manage the edition" goal defined as the "Editing Manager" goal.
2. Rules (1) More than one goal is associated with the same "Distribuitor " entity
3. Rules (2) The roles defined for the "Articles Manager," "article supplier" and "article client", were passed along to the "Distributor_Topics" and "Representative_ContentSupplier" entities.
4. Rule (3) The relationship that exists between the "Editing Manager" entity and the "Editor" external entity was decomposed into three relationships between the "Editor" entity and the "Distributor", "Representative_Editor_Reviewer" and "Representative_Editor_ContentSupplier" entities.

5. Conclusion and Future Work

The work herein proposed intends to help in the process of identifying the agents that will be implemented in an application based upon the analysis and decomposition of the system goals and the system itself in terms of sub-entities. Refinement rules were suggested to aid in the process of decomposition and refinement of the entities, roles, goals and relationships identified at each level of the refinement.

In terms of future work, we intend through the application of the method to more complex examples to be capable of defining new goals that help validate the refinements and the proposed rules. Next, we intend to develop a tool that, based upon the definition of the method and in the refinement rules, will be capable of helping and validating the refinement of the models that are generated and then, of helping to identify the systems agents.

Reference

1. Booch, G., Rumbauch, J., Jacobson, I.: The Unified Modeling Language User Guide, Addison Wesley (1999).
2. DeLoach, S. A.: Multiagent Systems Engineering: a Methodology and Language for Designing Agent Systems," Proceedings of Agent Oriented Information Systems (AOIS99), Seattle Washington, May (1999).
3. Elammari, M. and Lalonde, W.: An agent-oriented methodology: High-level and intermediate models. In G. Wagner and E. Yu, editors. Proc. of the 1st Int. Workshop. on Agent-Oriented Information Systems (1999).
4. Garcia, A. F., Cortés, M. I, and Lucena, C.J.P.: A Web Environment for the Development and Maintenance of E-Commerce Portals based on a Groupware Approach
   2001 Information Resources Management Association International Conference
   (IRMA) - Web Engineering for E-Commerce Applications, (2001) 722-724.
5. I. Reticular Systems.: *AgentBuilder: An Integrated Toolkit for Constructing Intelligent Software Agents*, revision 1.3 edition, February (1999).
6. Jennings, N. R.: On agent-based software engineering. Artificial Intelligence, 117(2), (2000) 277-296.
7. Kendall, E. A.: Agent Roles and Role Models. Intelligent Agents for Information and Process Management (1998).
8. Kinny, D. and Georgeff, M.: Modelling and design of multi-agent systems. In: Müller, J. P., Wooldridge, M., and Jennings, N. R. (Eds.), Intelligent Agents III (LNAI Volume 1193), Springer-Verlag, (1997) 1-20.
9. Lange, D. and Oshima, M.: Programming and Deploying Java Mobile Agents with Aglets, Addison Wesley (1998).
10. Wooldridge M., Jennings N. R., and Kinny D.: A methodology for agent-oriented analysis and design. Proceedings of the Third International Conference on Autonomous Agents (Agents'99), ACM Press, (1999) 69-76.
11. Wooldridge, M. and Jennings, N. R.: Pitfalls of agent-oriented development. Proceedings of the Second International Conference on Autonomous Agents (Agents'98), ACM Press, (1998) 385-391.
12. Wooldridge, M., Jennings, N. R., and Kinny, D.: The Gaia methodology for agent-oriented analysis and design. Journal of Autonomous Agents and Multi-Agent Systems, 3, (2000) 285-312.
13. Yu, L.; Schmid B.: A conceptual framework for agent-oriented and role-based work ow modeling. In G. Wagner and E. Yu, editors, Proc. of the 1st Int. Workshop. on Agent-Oriented Information Systems (1999).