The SRAMO Technique for Analysis and Reuse of Requirements in Multi-agent Application Engineering

Alisson Neres Lindoso, Rosario Girardi
Research Group on Software Engineering and Knowledge Engineering (GESEC)
Department of Informatics (DEINF) – Federal University of Maranhão (UFMA)
Av. Portugueses, s/n, Campus Universitário do Bacanga, São Luís-MA, Brasil, 65080-040
alissonlindoso@uol.com.br, rgirardi@deinf.ufma.br

Abstract

The increasing interest on the multi-agent software development paradigm turns necessary the elaboration of appropriate techniques and methodologies. A Multi-agent Application Engineering process is characterized by the construction of applications based on agent-oriented reusable software artifacts. This paper presents SRAMO, an ontology-driven technique for requirement analysis of multi-agent applications by reusing domain models representing common and variable requirements of a family of multi-agent applications in a domain. The requirements of a particular application are represented in models obtained through the instantiation of the ONTORMAS ontology that expresses the knowledge of the MAAEM methodology for Multi-agent Application Engineering, MAAEM integrates SRAMO and techniques for design and implementation of a specific application in a domain family. Part of a case study employing SRAMO for modeling RecomTour, a multi-agent application for recommendation of touristic packages through Web usage mining and collaborative filtering is also described.

1. Introduction

Multi-agent systems are considered an excellent metaphor to characterize complex systems and the agent software abstraction is very useful for understanding, engineering and using this kind of systems.

Many techniques, methodologies, tools and environments for multi-agent application development have been proposed [8][9][11][17][20]. However, software reuse has been little explored in those approaches. Some experiences already exist in this direction; however, they are limited to promoting the reuse of software patterns [17].

Multi-agent Application Engineering (MaAE) is a process for the development of specific applications through the reuse of software artifacts produced in Multi-agent Domain Engineering (MaDE), a complementary and interdependent process. The MaAE process is divided in three phases: analysis, design and implementation [21].

During the analysis phase, the requirement specification of an application is performed through the reuse of domain models that represent common and variable requirements of application families in their respective domains. In the design phase, an architectural solution and a detailed design are developed for the specified requirements, through the reuse of multi-agent frameworks. In the implementation phase, design concepts are mapped in terms of agents, behaviors and communication acts of the JADE platform [5], which is the adopted implementation technology in both the MaDE and MaAE processes.

Ontologies [1][4][11] are knowledge representation structures particularly useful for the specification of high level software abstractions [24]. Because their characteristics such as lack of ambiguity, formality, reusability and adaptability, they provide a terminology that can be shared by everyone involved in a development process. Moreover, they are easily extensible and adaptable, thus enabling reuse.

This work introduces SRAMO ("Specific Requirement Analysis Method based on Ontologies"), an ontology-based technique for requirement analysis of multi-agent applications through the reuse domain models. SRAMO is part of MAAEM ("Multi-Agent Application Engineering Methodology") [2], a methodology that guides the phases of a Multi-agent Application Engineering process and that has its knowledge represented in ONTORMAS ("Ontology for Reusing Multi-agent Software") [2], which also express the knowledge of MADEM ("Multi-agent Domain Engineering Methodology") [23], a methodology supporting the Multi-agent Domain Engineering process.

The requirements of a specific multi-agent application are modeled as instances of ONTORMAS, in whose knowledge base the software artifacts [24][25] reused by SRAMO are available. From an epistemological point of view, it must be noticed that MADEM and MAAEM share the semantics of modeling concepts (e.g.: goals, responsibilities, roles etc), being such concepts semantically related, which makes searches and inferences effective facilitating the selection, understanding and specialization of the MADEM products, reused by MAAEM.

The paper is organized as follows. Section 2 introduces the ONTORMAS ontology. Section 3 describes the SRAMO technique in the context of a Multi-agent Application Engineering process, detailing each one of its steps and the reused and produced software artifacts with examples extracted from a case study on Web usage mining and collaborative filtering for providing touristic packages recommendations. Section 4 discusses related work on techniques and methodologies for the development of agent-oriented applications. Finally, Section 5 concludes the paper and discusses further work being conducted.

2. The ONTORMAS ontology

The ONTORMAS ontology works as a modeling tool and a storage repository for products constructed on the Multi-agent Domain Engineering and Multi-agent Application Engineering processes. Agent-oriented software products of both MADEM and MAAEM are represented in ONTORMAS as instances of the corresponding subclasses of the *Modeling Tasks*, *Modeling Products*, *Modeling Concepts* and *Modeling Relationships* main classes, following the semantics established in the ontology, constructed with the *Protégé* ontology editor [12].

Modeling tasks and products are fairly related in the context of ONTORMAS, since the accomplishment of a task always originates a respective product. Some of these relationships are illustrated in Figure 1, a semantic network elaborated with the *Ontoviz Tab* [19] showing the product *Goal Model*, the task *Goal Modeling*, the concepts *General* and *Specific Goal*, *Responsibility* and *External Entity* and the relationship *Get Information*, among other intermediate elements.

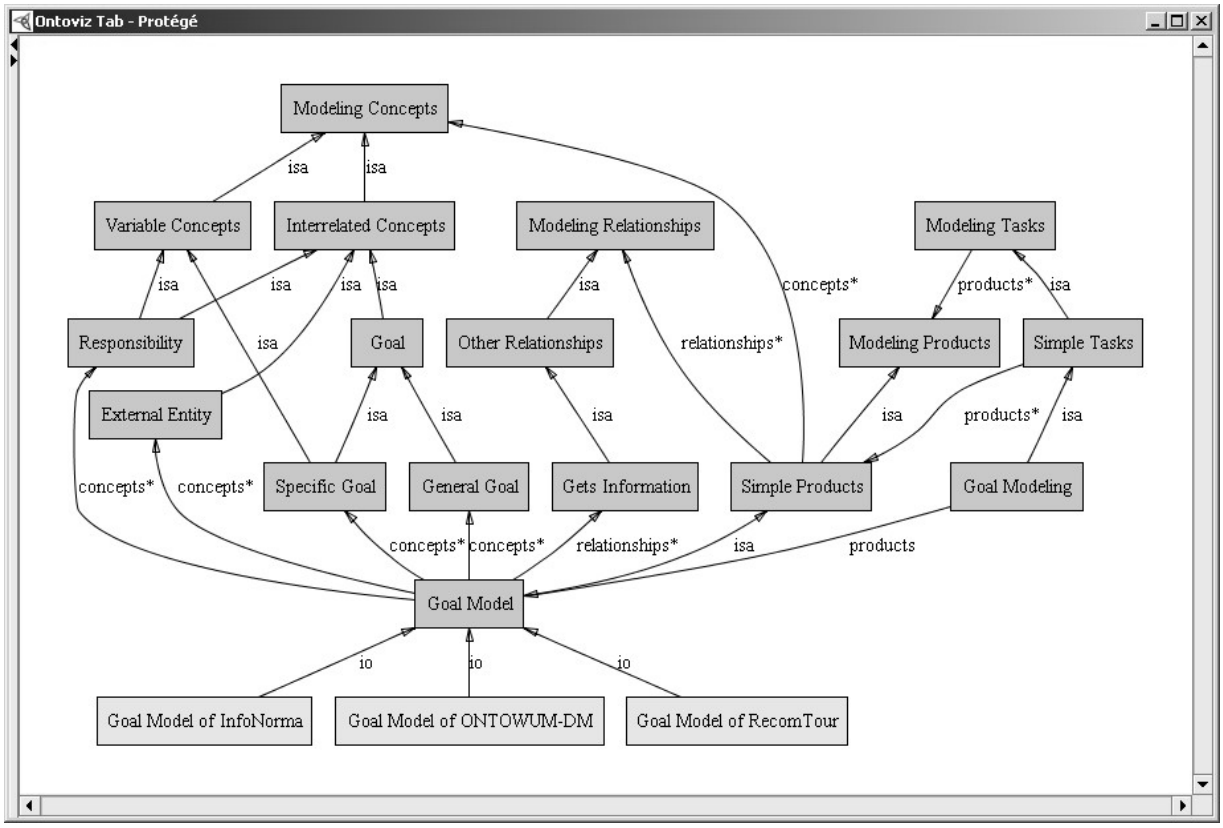

Figure 1. Semantic network relating the key modeling elements of ONTORMAS

Variability specification is essential on the representation of reusable artifacts because it discriminates how the common and variable modeling concepts differ on the applications of a family [10]. Such notion influences directly the selection and adaptation of the concepts for later product composition. It is expressed in ONTORMAS through the attribute *variability type* in the class *Variable Concepts* (Figure 2).

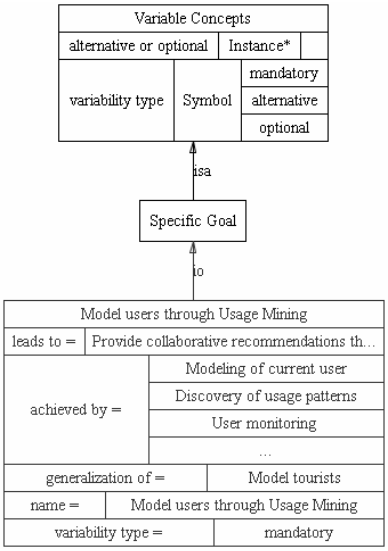

Figure 2. Instance of the *Variable Concepts* subclass with a "mandatory" *variability type*

The way as reuse is performed on ONTORMAS is also important. Figure 3 exemplifies a query done with the *Algernon Tab* [3] for the selection of appropriate architectural patterns to guide the modeling of the *RecomTour* architecture [26]. Through this query, instances of the *Architectural Pattern* class whose *context* and *problem* attributes containing, respectively, the expressions "multi-agent" and "abstraction level" are searched, resulting on the architectural pattern named *Multi-agent Layer* [25].

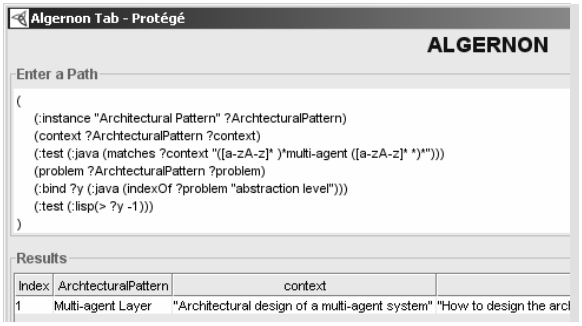

Figure 3. Query showing the selection of architectural patterns in ONTORMAS

This example also illustrates the effectiveness of the query results and, therefore, of the selection process of a software artifact by exploring their semantic representation. The concepts and their semantic relationships describing patterns and systems of patterns were previously developed in the ONTOPATTERN ontology [22], which was later integrated in ONTORMAS.

Frequently, the reuse of an agent-oriented software artifact involves its previous adaptation to adjust it to the specific requirements of the application being developed. In ONTORMAS, this is done by instantiating the concrete subclasses of the *Interrelated Concepts* abstract class, illustrated in Figure 4.

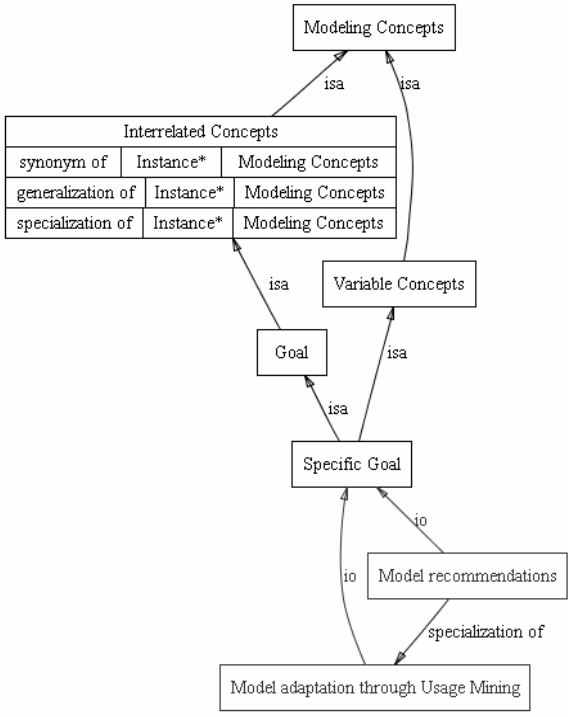

Figure 4. Semantic relationships of the *Interrelated Concepts* class of ONTORMAS

The adaptation of a modeling concept is performed by creating a new concept of the same type, indicating if it is a *synonymous*, *specialization* or *generalization* of the old one, according to what has been adapted, respectively: just its denomination, without modifying the meaning; or also its description conferring it a stricter or wider meaning.

Figure 5 shows an example in which the "Model adaptation through usage mining" instance of the *Specific Goal* class, found in the goal model of ONTOWUM-DM domain model [16], was specialized to "Model recommendations" for being adjusted to the

problem of recommending touristic packages, focused by the *RecomTour* application [26]. The original instance was previously designed for the personalization of Web interfaces and then such goal has become specific to offer destination suggestions for tourists.

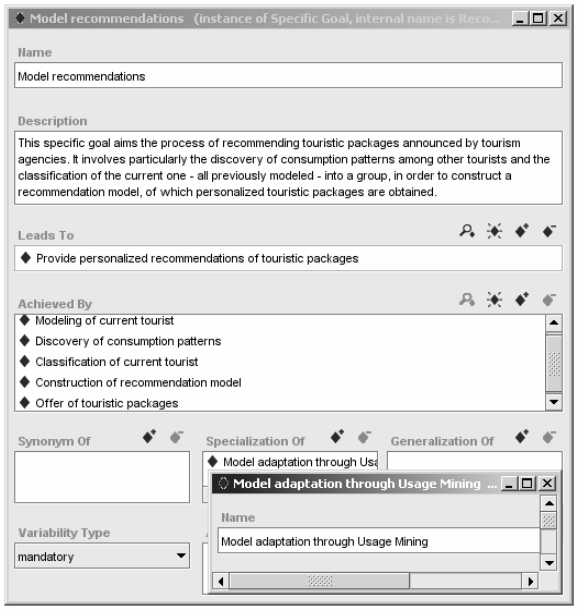

Figure 5. Example of adaptation by specialization of a *Specific Goal* of ONTOWUM-DM

Finally, a composition results from the integration of semantically related modeling concepts in simple modeling products or in composed ones, according to the corresponding modeling task and following the guidelines of the SRAMO technique.

Figure 6 shows an example of a modeling product composition, where the resulting *Application Specification* model is composed by other models produced as described in the next section.

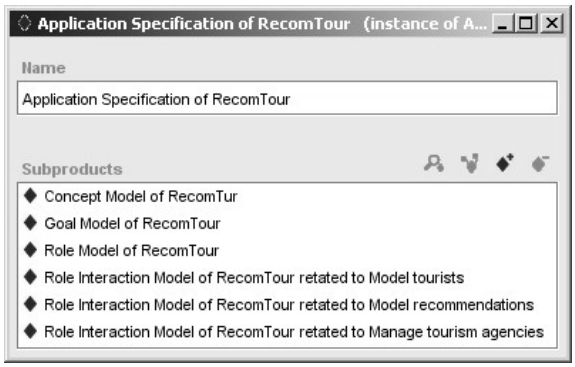

Figure 6. Example of composition resulting in the *Application Specification* of *RecomTour*

3. The SRAMO technique

The SRAMO technique (Figure 7) supports the analysis phase of MaAE process, guiding the requirement specification of a certain application. In this phase, the major task of a software developer is to reuse a set of common requirements shared by all applications of a family in a domain, along the variable ones that refer only to this specific application, complementing them with particular requirements not available for reuse.

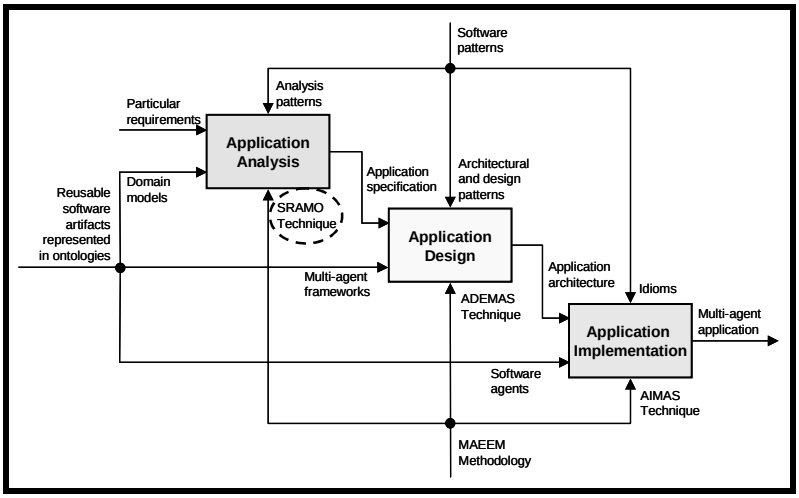

Figure 7. A Multi-agent Application Engineering process, including the Application Analysis phase supported by the SRAMO technique

Products reused by SRAMO are mainly domain models, high level software abstractions produced on the corresponding phase of the MaDE process (Table 1). Therefore, the requirements of a particular multi-agent application can be selected among the common and/or variable requirements specified in a domain model. While the common requirements are always reused, the reuse of variable ones depends on the needs of the specific application. In any case, both types of requirements can suffer adaptations before the final composition of the application.

Table 1. Steps, resources and products of the SRAMO technique

Steps	Resources (*MaDE*)	Products (*MaAE*)	
Concept Modeling	Domain Models	Concept Model	Application Specification
Goal Modeling		Goal Model	
Role Modeling		Role Model	
Role Interaction Modeling		Role Interaction Model	

The SRAMO steps involve the following notions. *Concepts* identify the entities of the domain and the existing relationships between them, reflecting the expert knowledge and allowing distinguishing a problem in some area. *Goals* are the target that an organization aims to reach, defining *responsibilities* to the individuals that integrate it, which play *roles* and are able to execute *activities*. For this, they use and produce *knowledge*, satisfy pre and post-conditions and have particular *skills*. Although autonomous, roles, which are *internal entities*, establish interactions between themselves and also with *external entities* in order to execute more complex activities, beyond their capacities.

In the following sub-sections, the description of the step by step of SRAMO is exemplified with the requirement specification of *RecomTour* [26], a multi-agent application for recommending touristic packages through Web usage mining and collaborative filtering, based on the reuse of ONTOWUM-DM [16], a domain model that specifies the requirements of an application family for providing Web recommendations based on usage mining and collaborative filtering.

3.1. Concept modeling

In the development of multi-agent applications, the definition of domain concepts is essential because it guides the later construction of the ontology representing the knowledge shared by the application agents. The product of this step is a *Concept Model*, which is specified in a semantic network where nodes represent domain concepts and arcs, the relationships between them.

Concept modeling serves as a brainstorming and goes from an informal analysis of the application requirements, when an initial set of concepts is gathered, to their extension through gradual refinements. Once identified the key concepts of the application, it is initiated the querying and selection of concepts available in the ONTORMAS repository, in a similar way as is done in the example of Figure 3. Such key concepts are considered as parameters for querying ONTORMAS. The variability of selected concepts allows enriching the semantic network.

The composition of the *Concept Model* is a consequence of the semantic relationships between concepts in the model, considering both the reused and the new ones. The result is a uniform and coherent representation of the basic knowledge of the problem focused by the application.

As an example, Figure 8 shows the *Concept Model* of RecomTour, which aggregates the main concepts related to the personalized recommendation of touristic packages, some of them reused from the respective model of ONTOWUM-DM domain model.

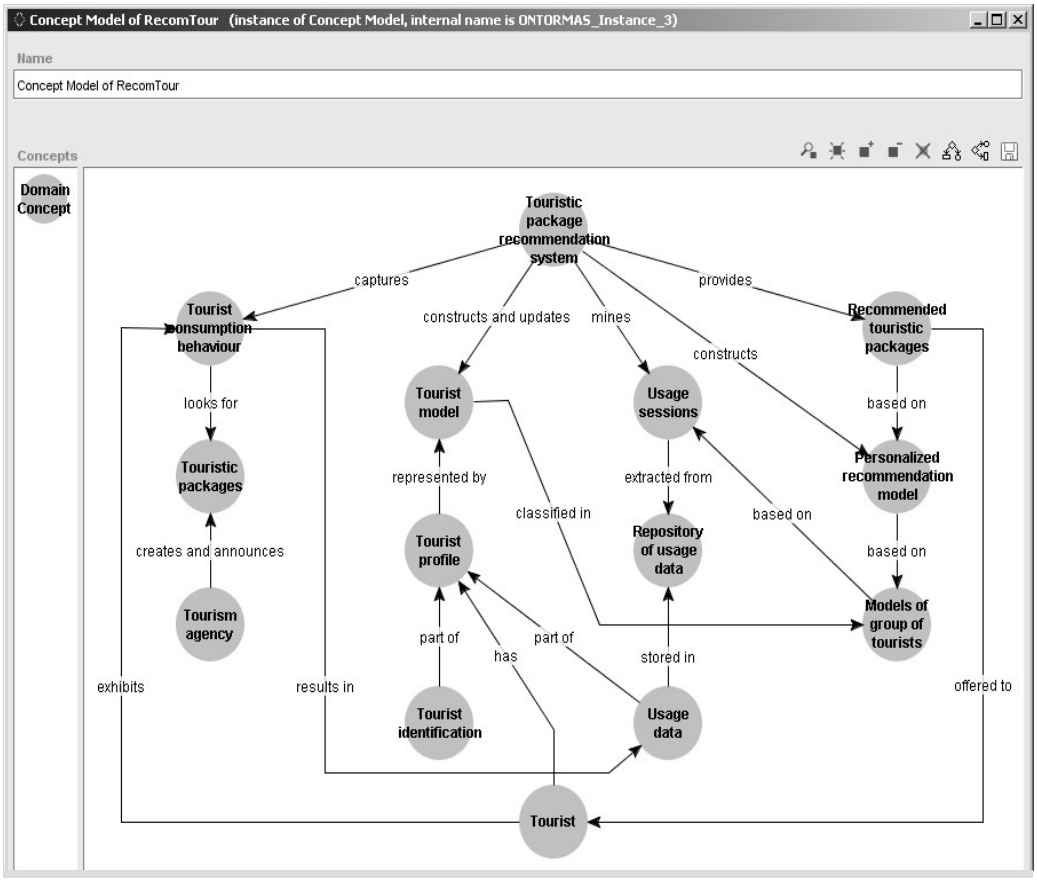

Figure 8. *Concept Model* of RecomTour

3.2. Goal modeling

The definition of a hierarchy of goals, a set of responsibilities and the external entities that exchange information with the application corresponds to another part of the requirement specification guided by SRAMO. The product of this step is a *Goal Model*, which strongly supports the remaining steps, having the form of an organizational diagram with several levels.

The querying and selection in ONTORMAS is initiated by establishing the main goal of the application. After that, and employing domain models related with the approached problem, it is performed an operation similar to the one shown in Figure 3, through which goals that match partially or totally the application goal are then selected.

Goals are organized in a hierarchy as general and specifics ones. When the application goal corresponds to the general goal of a certain goal model in the ONTORMAS repository, then this goal and all specific goals whose variability is mandatory are selected. On the other hand, when the correspondence occurs with a specific goal, then the selection includes only this goal and the mandatory related specific goals. In both cases, if the remaining specific goals are optional, only the ones that contribute for reaching the application goal are selected. Among the alternative ones, those that suit to the application purpose are also selected.

After this, responsibilities are selected among the ones allowing reaching the reused goals and considering their variability. The same occurs with external entities. Once all goals, external entities and responsibilities have been selected, they are eventually adapted to compose the *Goal Model* containing: at the superior level, the application goal as general goal; in the intermediate levels, the specific goals, according to the hierarchy existent in the reused models; and in the inferior level, the identified responsibilities.

As an example, Figure 9 illustrates the *Goal Model* of RecomTour, which was produced by reusing part of the corresponding model of the ONTOWUM-DM domain model relative to modeling users and adapting systems through Web usage mining.

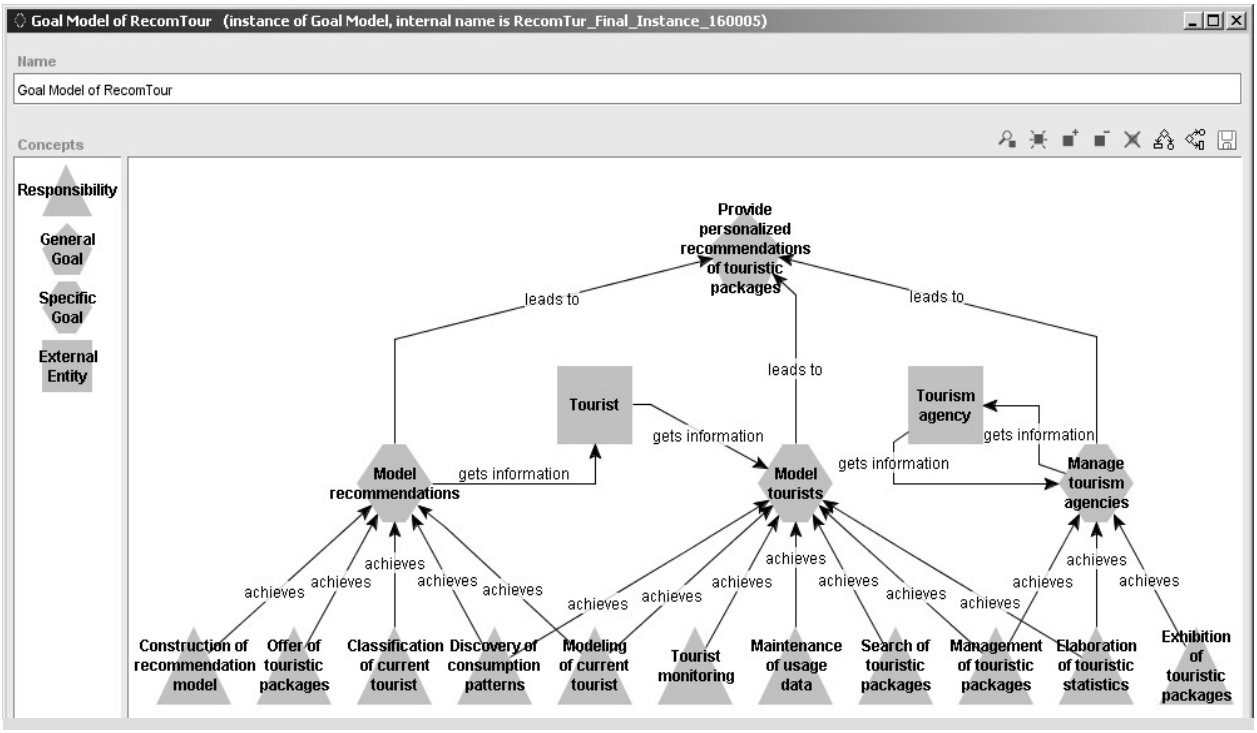

Figure 9. *Goal Model* of RecomTour

3.3. Role modeling

Once identified the responsibilities, then roles are associated to them, as well as the used and produced knowledge, the pre and post-conditions and the required skills. Moreover, the external entities that contribute for the execution of such responsibilities are introduced, which can be exercised through activities that present the same nature but are more specialized. The product of this step is a *Role Model*, graphically represented in an organizational diagram with three levels.

The selection of roles and external entities is based on the domain models previously reused. Through a procedure like the one shown in Figure 3, roles related to each reused responsibility are searched, as well as the external entities that have appeared during goal modeling. Also, the activities that decompose a responsibility must be selected and adapted, if necessary. Everything should be done according to the variability that each activity presents.

After that, the selection and adaptation of input/output knowledge of each responsibility or activity is done, in a way that the ones used have been previously produced.

Skills, considering their variabilities, are dealt in a general manner during analysis phase, as a set of possible techniques that roles can apply to perform their responsibilities or activities. Skills are specialized in the design phase.

Concerning the conditions, there are two types, considering if they are verified before or after the execution of a responsibility or activity. Thus, the verification of one or a set of pre-conditions is enough to enable the execution of a responsibility or activity. On the other hand, the execution of a responsibility or activity turns verifiable one or more post-conditions.

In this way, the *Role Model* results of the composition of all modeling concepts selected and adapted based on the reused role models, being in the first level, the roles; in the second level, the responsibilities, activities, knowledge and conditions; and in the third level, the skills.

As an example, Figure 10 shows part of the *Role Model* of RecomTour, obtained through the reuse of the corresponding model in ONTOWUM-DM.

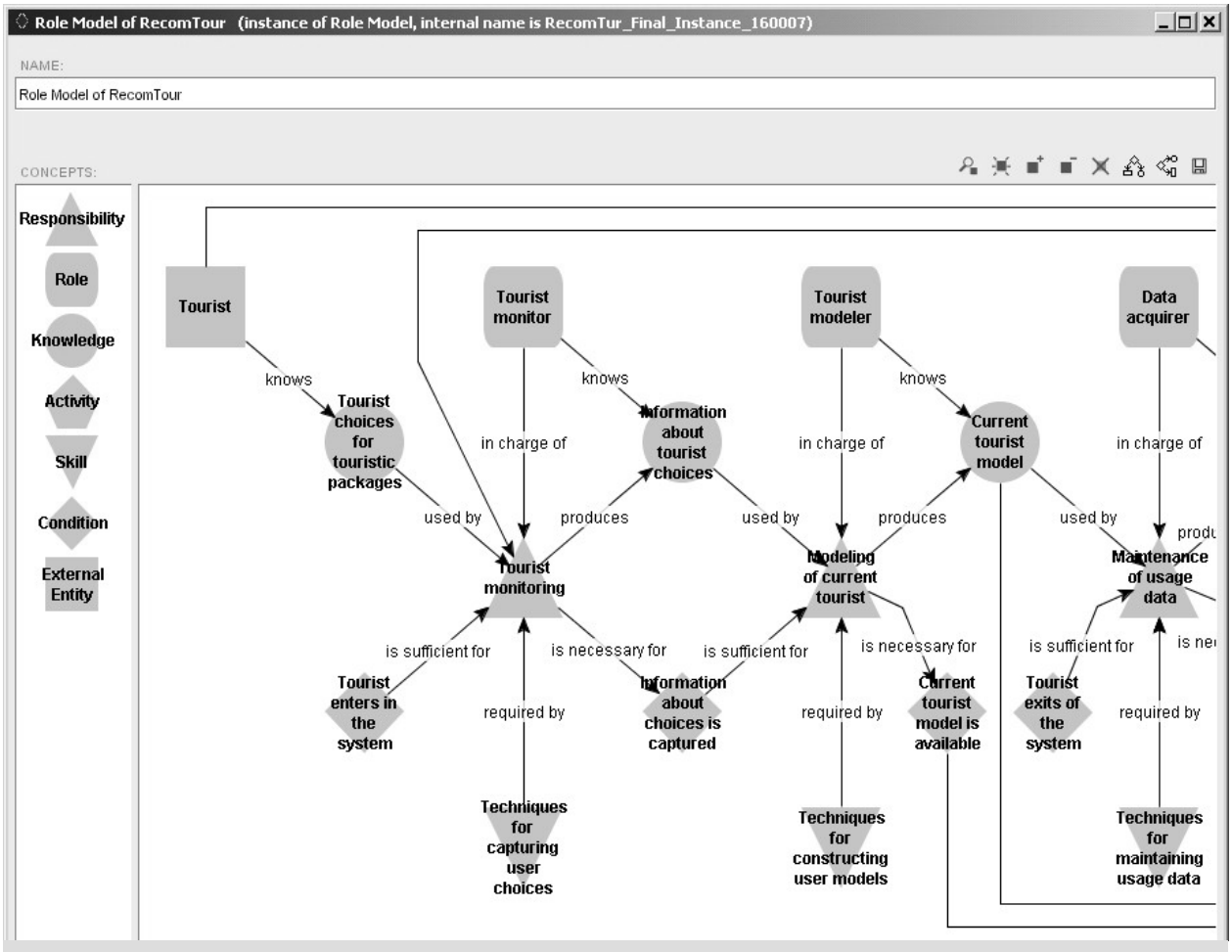

Figure 10. Part of the *Role Model* of RecomTour

3.4. Role interaction modeling

Concluding the requirement analysis phase of an application, after all roles have been selected, adapted and composed, it is necessary to determine how they interact between themselves and with external entities. The product of this step is a set of *Role Interaction Models*, one for each specific goal on the lowest level of the hierarchy. Its graphical representation is similar to the one of a collaboration diagram of UML [6].

Each interaction between roles has the form of a method invocation from the originating role, indicating a responsibility or activity of the destination role having as parameters the knowledge exchanged. The order indicating when the interactions occur is also indicated through a cardinal numeration.

Interactions between roles and external entities represent events, such as the display of a window interface.

A *Role Interaction Model* corresponding to a specific goal that has been reused without any adaptation will be selected and reused from the corresponding domain model. Those corresponding to adapted specific goals will be reused and adapted with the new or modified roles, external entities and interactions. A *Role Interaction Model* corresponding to a specific goal specifically created for the application being modeled, must be created from scratch.

As an example, Figure 11 illustrates one of the *Role Interaction Models* of RecomTour, obtained from the reuse of part of the ONTOWUM-DM domain model.

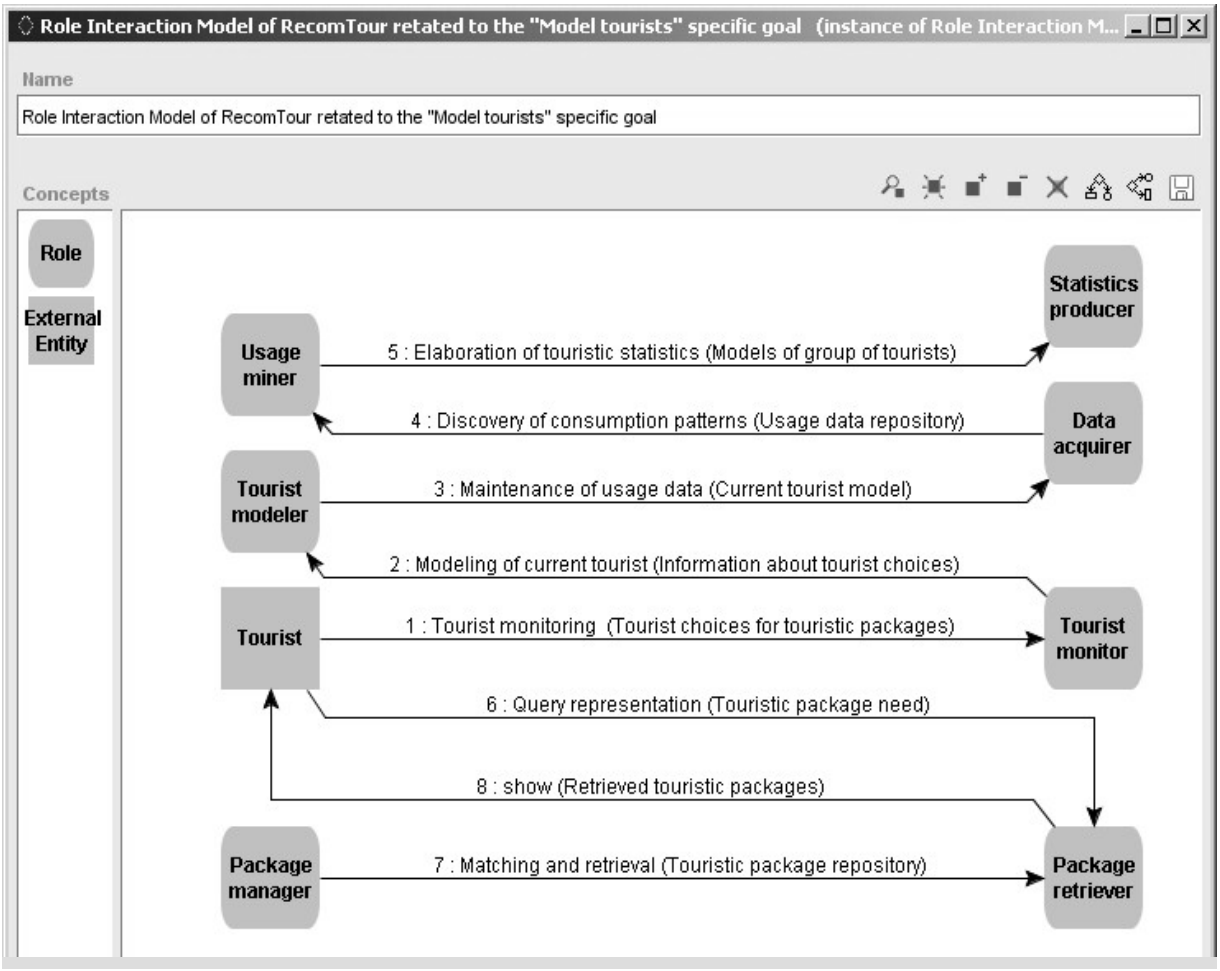

Figure 11. *Role Interaction Model* of RecomTour related to the *model tourists* specific goal

4. Related work

Several techniques and methodologies for the development of agent-oriented applications have been analyzed [8][9][11][17][20]. Works on software reuse, and Domain Engineering and Application Engineering processes [7][13][14][15][18] have also influenced the conceptualization of the SRAMO technique.

Most techniques for requirement analysis of multi-agent applications are inspired or extend the concepts of the object-oriented paradigm. Others are based on concepts from the Knowledge Engineering area. Usually they focus goals, roles, activities and interactions modeling tasks related to individuals that constitute an organization.

Two main features distinguish SRAMO from other existing approaches. First, it provides support for the reuse of agent-oriented software artifacts, and second, it is a knowledge-based technique where the application specification is represented as instances of the ONTORMAS ontology.

5. Concluding remarks

This paper has described SRAMO, an ontology-based technique for analysis and reuse of requirements in multi-agent development. The requirements of a multi-agent application are specified by reusing domain models, which represents common and variable requirements of a multi-agent application family in a domain.

Domain models and requirement specifications are represented as instances of the ONTORMAS ontology, which serves as repository for reusable software artifacts and also as tool supporting the application development.

Using SRAMO, a case study has been developed, where the requirements of *RecomTour* [26], a multi-agent application in the touristic domain, were specified. For that, ONTOWUM-DM [16], a domain model for Web usage mining, was reused thus considerably minimizing the modeling effort.

6. Acknowledgments

We would like to thank Lucas Rêgo Drumond for reviewing this paper.

7. References

[1] A. Gómez-Pérez, and V. Benjamins, "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods", *Proceedings of the International Joint Conference on Artificial Intelligence (KRR5/IJCAI'99), v. 18*, Stockolm, Sweden, 1999, pp. 1.1-1.15.

[2] A.N. Lindoso, "A Ontology-based Methodology for Multi-agent Application Engineering", *Dissertation (Master Degree in Electrical Engineering, Area of Computer Science)*, Federal University of Maranhão (CPGEE/UFMA). São Luís-MA, 2006. [in Portuguese]

[3] Algernon Tab. Rule-Based Programming. Available in: <http://algernon-j.sourceforge.net/doc/algernon-protege.html>. Access in: Feb. 7th, 2006.

[4] B. Chandrasekaran, J.R. Josephson, and V.R. Benjamins, "What are Ontologies, and why do we need them?", *IEEE Intelligent Systems, v. 14 n.1*, January 1999, pp. 20-26.

[5] Bellifemine, F. et al. "JADE Administrator's Guide". Available in: <http://jade.tilab.com/doc/administratorsguide.pdf>. Access in: Nov. 24th, 2005.

[6] Booch, G., J. Rumbaugh, and I. Jacobson, Unified Modeling Language User Guide, Reading, Addison-Wesley Publishing Co., New York, 1999.

[7] Czarnecki, K., and U.W. Eisenecker, Generative Programming: Methods, Tools, and Applications, ACM Press/Addison-Wesley Publishing Co., New York, 2000.

[8] F. Zambonelli, *et al*., "Developing Multiagent systems: The Gaia methodology", *ACM Transactions on Software Engineering and Methodology, 12(3)*, 2003, pp. 417-470.

[9] G. Caire, et al., "Agent-Oriented Analysis using MESSAGE/UML", Proceedings of the Second International Workshop on Agent-Oriented Software Engineering (AOSE 2001), 2001, pp. 101-108.

[10] I.M.S. Gimenes, G.H. Travassos, "The Product Line Approach for Software Development", Proceedings of the 21th Jorney of Update in Informatics (JAI 2002) in 22th Congress of Brazilian Society of Computing (CSBC 2002). Florianópolis, Santa Catarina, Brasil, July 15-19, 2002. [in Portuguese]

[11] J. Dileo, T. Jacobs, and S. Deloach, "Integrating Ontologies into Multi-Agent Systems Engineering", *Proceedings of 4th International Bi-Conference Workshop on Agent Oriented Information Systems (AOIS 2002)*, Bologna, Italy, July 2002, pp. 15-16.

[12] J. Gennari, M.A. Musen, R.W. Fergerson, *et al*., "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development", *Technical Report SMI-2002-0943*, 2002.

[13] J.V. Withey, "Implementing Model-Based Software Engineering in Your Organization: An Approach to Domain Engineering", *Software Engineering Institute, Carnegie Mellon University*, Pittsburgh, April 1994.

[14] K. Kang, et al., "Feature-Oriented Domain Analysis (FODA) Feasibility Study", *Technical Report, Software Engineering Institute, Carnegie Mellon University*, Pittsburgh, Pennsylvania. November 1990.

[15] L. Geyer, and M. Becker, "On the Influence of Variabilities on the Application-Engineering Process of a Product Family", *Proceedings of the Second International Conference on Software Product Lines*, Lecture Notes In Computer Science, Springer-Verlag, v. 2379, London, 2002, pp. 1-14.

[16] L.B. Marinho, "A Multi-agent Framework for Web Personalization Based on User Modeling and Usage Mining". *Dissertation (Master Degree in Electrical Engineering, Area of Computer Science)*, Federal University of Maranhão (CPGEE/UFMA). São Luís-MA, 2005. [in Portuguese]

[17] M. Cossentino, L. Sabatucci, S. Sorace, and A. Chella, "Patterns reuse in the PASSI methodology", *Proceedings of the Fourth International Workshop Engineering Societies in the Agents World (ESAW'03)*, Imperial College London, UK, October 2003, pp. 29-31.

[18] M. Harsu, "A Survey of Domain Engineering", *Report 31, Institute of Software Systems*, Tampere University of Technology. December 2002.

[19] Ontoviz Tab. Visualizing Protégé Ontologies. Available in: <http://protege.stanford.edu/plugins/ontoviz/ontoviz.html>. Access in: Feb. 7th, 2006.

[20] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "TROPOS: An Agent-Oriented Software Development Methodology", *Journal of Autonomous Agents and Multi-Agent Systems, v. 8, n. 3*, Kluwer Academic Publishers, May 2004, pp. 203-236.

[21] R. Girardi, "Agent-based Software Engineering", *Proceedings of 4th Brazilian Conference on Computer Science (CBCOMP 2004)*, Ed. UNIVALI, Itajaí, Santa Catarina, Brasil, October 08-12, 2004, pp. 913-937. [in Portuguese]

[22] R. Girardi, and A.N. Lindoso, "An Ontology-based Knowledge Base for the Representation and Reuse of Software Patterns", *ACM Sigsoft Software Engineering Notes, v. 31, n. 1*, New York, United States, January 2006, pp. 1-6.

[23] R. Girardi, and A.N. Lindoso, "An Ontology-based Methodology and Tool for Multi-agent Domain Engineering", 2006. [submitted paper]

[24] R. Girardi, and C. Faria, "An Ontology-based Technique for the Specification of Domain and User Models in Multi-Agent Domain Engineering", *CLEI Electronic Journal, v. 7, n. 1, paper 7*, June 2004.

[25] R. Girardi, L.B. Marinho, and I. Oliveira, "A System of Agent-based Patterns for User Modeling based on Usage Mining", *Interacting with Computers Journal, v. 17, n. 7*, Elsevier, September 2005, p. 567-591.

[26] R. Vieira, "Modeling of a Multi-agent System for Recommendation of Touristic Packages through Web Usage Mining and Collaborative Filtering", *Conclusion Work (Undergraduate Degree in Computer Science)*, Federal University of Maranhão (CGCC/UFMA). São Luís-MA, 2006. [in Portuguese]