AN ONTOLOGY-BASED TECHNIQUE FOR THE SPECIFICATION OF DOMAIN AND USER MODELS IN MULTI-AGENT DOMAIN ENGINEERING

Rosario Girardi, Carla Gomes de Faria

Department of Computer Science, Federal University of Maranhão, Av. Dos Portugueses s/n - Campus do Bacanga, São Luis - MA, Brazil Email: {rgirardi@deinf.ufma.br; carlaslz@yahoo.com.br}

Abstract. Because of its facilities for the generalization and specialization of concepts and the unambiguous terminology they provide, ontologies are being used for the representation of reusable software artifacts. This work describes GRAMO, an ontology-based technique for the specification of domain and user models in Multi-Agent Domain Engineering. ONTODUM, an ontology-based tool supporting GRAMO is also introduced. ONTODUM represents the knowledge of GRAMO. Some case studies developed to evaluate GRAMO are also briefly described.

1 Introduction

Domain analysis [1] [25] [29] [32] [37] [38] [40] identifies reuse opportunities and specifies the common requirements of a family of applications. The product of this phase is a domain model. Domain design looks for a documented solution to the problem specified in a domain model. The product of this phase is composed of one or more frameworks and a collection of design patterns, documenting good solutions in that domain. Reusable components integrating the framework are constructed during the phase of domain implementation.

Our research group is working on a process for Multi-Agent Domain Engineering [15] [20] [21] [43]. We are constructing a software development environment composed of a set of development tools and libraries of high-level reusable software abstrations (domain models, domain specific languages, user models, agent-based architectural and detailed design patterns, multi-agent frameworks, and reusable software agents) for both compositional and generative agent-based application development. Experiments are being conducted on the legal, tourism, and pecuary domains and for problem-solving tasks of information retrieval and filtering [6] [30] [31] [39].

Ontologies are being used to represent both the knowledge of techniques for Domain Engineering and generated products [11] [12] [13] [17] [18] [44]. Ontologies [5] [23] [24] are knowledge representation structures especially useful for the specification of high-level reusable software. They provide an unambiguous terminology that can be shared by all involved in the development process. Also, an ontology can be as generic as needed allowing its reuse and easy extension.

One of the specific problem solving tasks we are approaching is agent-based user modeling for the adaptation of systems. Most software systems have not the ability to satisfy the heterogeneous needs of its users. Each user has particular knowledge level, needs, abilities and preferences. In that context, there is a need for building systems adaptable to each user type or group of users with common characteristics. It is possible to adapt systems according to the knowledge and experience of users; the history of previous actions, the cognitive properties (learning style, personality); goals and intentions; interests and preferences.

An important component of an adaptable interactive system is the ability to model the users of the system. User modeling is essential in systems that try to adapt their behavior to users in order to interact in a more intelligent and individualized way.

In our approach, domain and user modeling produces the requirement specification of a family of similar systems in an application domain. The specification includes both the required functionality of the domain and features of end users as well. Domain and user models are the reusable software products generated by the Domain and User modeling phase of Multi-Agent Domain Engineering.

A domain model - domain dependent and specified at a high level of abstraction - represents the formulation of a problem, knowledge or activity of the real world. The formulation is generic enough to represent a family of similar systems. Ontologies are being used to represent domain models establishing the vocabulary and semantics for the elements, processes and relationships in the systems.

A user model specifies features, needs, preferences and goals of end users [14]. It is a knowledge source that contains information, explicit or implicit acquired, of all the important user features to be used in the process of adaptation of a software application. Ontologies are also being used to represent user models.

The technique GRAMO ("Generic Requirement Analysis Method based on Ontologies") defines the activities to be accomplished in the construction of domain and user models in Multi-Agent Domain Engineering. A reusable ontology, ONTODUM, guides the construction of the models, which are created by instantiating the hierarchy of meta-classes of ONTODUM. Thus, both domain and user models are represented in frame-based ontologies where concepts, activities and relationships in the domain and user profiles are represented in frame-based classes according to the representation criteria of ONTODUM.

This work introduces the technique GRAMO, and the reusable ontology ONTODUM.

The article is organized as follows. Section 2 introduces the GRAMO technique and the ONTODUM reusable ontology. Section 3 briefly describes some case studies developed to evaluate GRAMO. Section 4 describes related work on Domain analysis and Requirement Engineering of Multi-Agent Systems. Section 5 concludes the paper with some remarks on further work being conducted.

2 The GRAMO technique

GRAMO [11] is an ontology-based technique for the specification of domain and user models in the analysis phase of Multi-Agent Domain Engineering, according to the knowledge of a particular application domain and needs, preferences, knowledge and abilities of a user or group of users. Domain and user models are constructed through the instantiation of ONTODUM [18], a reusable ontology that encodes the knowledge of the GRAMO technique (Fig. 1).



Fig. 1. Inputs and products of GRAMO

Methods for Requirement Engineering of multi-agent systems [3] [4] [7] [9] [10] [22] [26] [35] [45] [46] [47] usually focus on modeling goals, roles, activities and interactions of individuals of an organization. An organization is composed of individuals with general and specific goals that establish what the organization intends to reach. The achievement of specific goals allows reaching the general goal of the organization. For instance, an information system can have the general goal of "satisfying the information needs of an organization" and the specific goals of "satisfying dynamic or long term information needs". Specific goals are reached through the exercise of responsibilities that individuals have. Individuals play roles with a certain degree of autonomy and exercise their responsibilities through the execution of activities. For that, they dispose of a set of resources. For instance, an individual can play the role of "information retriever" with the responsibility of executing activities to satisfy the dynamic information needs of an organization. Another individual can play the role of "information filter" with the responsibility of executing activities to satisfy the long-term information needs of the organization. Resources can be, for instance, the rules of the organization to access and structure its information sources. Sometimes, individuals have to communicate with other internal or external individuals to cooperate in the execution of an activity. For instance, the individuals playing the roles of information retriever and information filter may need to interact with an individual (e.g. Information source) having the responsibility of the storage and update of the information items of the organization.

According to these definitions, in GRAMO, domain and user modeling of multi-agent systems is performed according to the following modeling tasks (Table. 1).

Phases	Tasks		Products	
Domain Modeling	Concept Modeling	Goal Modeling		
		Role Modeling		
		Interaction	Goal Model, Role Model and	
		Modeling		
	Variability Modeling		Interaction Widder)	
User Modeling	Acquisition		User Model	
	Representation			
	Maintenance			

Table. 1. Phases, tasks, subtasks and products of the GRAMO technique

2.1 Phases, tasks, subtasks and products of the GRAMO technique

2.1.1 Domain Modeling

Domain modeling is performed according to the following tasks.

- *Goal modeling*. Considering the problem that the system intends to solve, the general goal of the system can be identified. Specific goals are obtained through a refinement of the general goal. A goal model is constructed as a product of this modeling task.
- *Role modeling*. For each specific goal, the responsibilities that will be exercised by internal and external roles are identified. Then, the activities allowing the exercise of each responsibility are defined. During this refinement process, it can be identified that the same activity or a set of related activities are executed by several roles. In this case, it should be appropriate the creation of an independent role having the responsibility of executing these activities on behalf of the other roles. The resources that an individual playing a role will need to execute his/her activities are also identified. A role model is constructed as a product of this modeling task.
- *Interaction modeling*. Through an analysis of their respective activities, the interactions between internal and external roles are identified. An interaction model is constructed as a product of this modeling task.
- *Concept modeling.* Consulting domain specialists and sources of information about the domain, main concepts of the domain and relationships between them are identified. Existent software applications in the domain are also analyzed in order to identify commonalities and differences between them. A conceptual model is constructed as a product of this modeling task.
- *Variability modeling*. According to a set of predefined rules [16], the goal and role models are refined to classify goals, roles, responsibilities, activities and resources as fixed or variables concepts.

2.1.2 User Modeling

User modeling in GRAMO looks for identifying which information about a user or group of users needs to be modeled, and how to acquire, represent and maintain this information.

Main information that may need to be captured about users is:

- Personal information (name, address, telephone number, geographical data, age, sex, education, civil status).
- User knowledge about concepts, and relationships between concepts in an application domain;
- User abilities on how to perform the tasks in the domain;
- User interests and preferences;
- User goals and plans on the interaction with the system;
- Usage data, i.e. the way as a user interacts with the system.

The techniques for the acquisition of user information can be direct or indirect. Using direct techniques, information is supplied directly by the users [28]. Using indirect techniques, user information is obtained by observing the interactions of the user with the system without his/her direct participation. Stereotype classification is a technique frequently used for direct acquisition [2]. A stereotype represents a collection of common attributes of a user group. The system can exhibit a form for the user to fill out and, starting from this information, classify his/her in pre-defined stereotypes. Later, they can be made predictions based on those stereotypes.

For indirect acquisition, machine learning techniques are frequently used [36] [41].

2.2 ONTODUM: An ontology-based tool for domain and user modeling

ONTODUM is a reusable ontology, which guides the elicitation and specification of the concepts and tasks to be accomplished in a domain and of particular features of end users.

The development process of ONTODUM consists of two phases: the definition and the design of the ontology. The techniques used in each phase are inspired in several proposals for ontology development [27] [34].

In the definition phase, the knowledge of the GRAMO technique for constructing domain and user models are represented in a semantic network. In the design phase, concepts and relationships in the semantic network are mapped to ONTODUM, a frame-based ontology.

2.2.1 Ontology definition

In this phase, the knowledge about GRAMO is represented in a semantic network.

Fig. 2 shows the part of the semantic network representing the concepts of modeling, their relationships and attributes. For instance, role is a modeling concept, which has the attributes responsibility and resource.



Fig. 2. Semantic network of modeling concepts

Fig. 3 shows the part of the semantic network representing the domain modeling task and generated products. Domain modeling builds a domain model composed of a goal, role, interaction and domain concept models constructed through the subtasks Goal, Role, Interaction and Concept modeling, respectively. Goal modeling builds a goal model, representing the general and specific goals of the system. Role modeling builds a role model, representing the roles of the system. Interaction modeling builds a model of interactions, representing the interactions between the roles and the external entities. Concept modeling builds a model of domain concepts, representing the concepts of the domain and the relationships between them.

Fig. 4 shows the part of the semantic network representing the user modeling task and generated products. User modeling constructs a model of a user or group of users through the subtasks, user model acquisition, representation and maintenance. Using implicit or acquisition techniques, the acquisition subtask obtains the user information used by the representation subtask to create an internal representation of the user model. The maintenance subtask updates the user model according to changes on the user or group of users profile. The user model represents the user attributes: personal information, goal, preferences, cognitive and physical skills.



Fig. 3. Semantic network of the domain modeling task and generated products

2.2.2 Ontology design

In the design phase, concepts and relationships in the semantic network are mapped to ONTODUM, a frame-based ontology composed of a hierarchy of meta-classes, according to the following modeling rules. Nodes are mapped to meta-classes. Nodes related by a link of type a kind of are mapped in a hierarchy of sub-classes and superclasses. Other links are mapped to slots of the corresponding meta-class. Appropriate facets are associated to each slot, like type and cardinality.

Fig. 5 shows the hierarchy of meta-classes of ONTODUM and the Domain Modeling meta-class in the Protégé editor [42] which specifies the subtasks of Domain Modeling (subtasks slot) and the product of this modeling task (models slot).



Fig. 4. Semantic network of the user modeling task and generated products

🛜 Ontodum Protégé-2000 (C:\Documents and Set	tings\alissonlindoso\Desktop\Pri	ofa Girardi\Bntodum tese	(corrigida)\Ontodum.pprj)		_ & ×
Project Window Help					
Cinclasses Sill Slots 📰 Forms 🆚 I	istances 📫 Queries				
Relationship Superclass 🔻 V C 🕸 🗙	C Domain Modelling (type=:ST	ANDARD-CLASS)			C ×
🛛 🗣 🤆 Modelling Concepts A	Name	Documentation		Constraints VC +	-
C Gnal A	Domain Modelling				
- C Responsibility					
- CResource	Role				
C External Entity	Concrete	–			
P C Domain Concept	Tomniato Sinte			VVCX+	
e c skil	Noma	Type Cardinali	v	Other Fonds	
C Preferences	S subtasks	Instance multiple	classes=(Interaction Mode	elling,Concept Modelling,Role Mode	illing
🖗 😇 Acquisition Techiques	S models ^O	Instance multiple	classes={Domain Model}		
- (C) Implicit	S DOCUMENTATION	String single String multiple			
- C Forms	S .ROLE	Symbol single	allowed-values=(Abstract,	Concrete) defaull=(Concrete)	
- © Steriotype	S DIRECT-TEMPLATE-SLOTS	Instance multiple	classes={:SLOT}		
C Machining Learning Technic	S DIRECT-TYPE	Class single Instance multiple	parents={:CLASS}		
C Domein Modelling	S DIRECT-SUPERCLASSES	Class multiple	parents={:THING}		
Concept Modelling	S DIRECT-SUBCLASSES	Class multiple	parents={:THING}		
- C Goal Modelling	SLOT-CONSTRAINTS	Instance multiple	classes={:CONSTRAINT}		
C Interaction Modelling					
- C User Modelling					
🖉 — 🧟 User Model Acquisition					
User Model Representation					
Products of Modelling A					
🗌 🖳 😳 Domain Model					
- Concept Model					
C Bole Model					
- 💿 Interaction Model					
C User Model					
M					

Fig. 5. Meta-class Domain Modeling

2.3 Constructing domain and user models with GRAMO

The construction of a domain model is made through the instantiation of the meta-class Domain Modeling (Fig. 5) that will create the class Domain Model, containing the specification of the concepts and tasks in the domain. This requires the instantiation of the meta-classes Concept, Goal, Role and Interaction Modeling, which will create the classes Concept, Goal, Role and Interaction Models.

The construction of a user model is made through the instantiation of the meta-class User Modeling that will create the class User Model, containing the user information. This requires the instantiation of the meta-classes User Model Acquisition, User Model Representation and User Model Maintenance.

3 Evaluation

In order to validate the GRAMO technique, several case studies have been developed. These case studies approached the construction of ONTOINFO [30], ONTOJUS [31], ONTOTOUR [6] and ONTOPEC [39] domain models by reusing ONTODUM through the application of the GRAMO technique. These ontology-based domain models represent the concepts and common requirements of families of software applications for information retrieval and filtering (ONTOINFO), legal (ONTOJUS), tourism (ONTOTOUR) and pecuary (ONTOPEC) domains. A full description of all these case studies is out of the scope of this paper.

Fig. 6 shows the example of the Goal Model of the domain model of ONTOTOUR.



Fig. 6. Goal Model of the Tourism Domain

4 Related Work

Considering their advantages, the use of ontologies for the representation of software products is increasing [8] [33]. Particularly, some methodologies for the development of multi-agent specific applications are being extended for ontology support [3] [7] [9] [10].

Some approaches for Ontology-based Domain Analysis have been proposed [8], which integrate methodologies for ontology building with techniques for Domain Analysis. Our approach differs from these proposals mainly in two aspects. First, we are proposing a meta-domain model - ONTODUM – which represents the knowledge of both techniques for requirement analysis of multi-agent systems and techniques for Domain Analysis. Second, ONTODUM is a CASE modeling tool, which guides the application of the technique GRAMO.

5 Concluding remarks and further work

This article introduced GRAMO, a technique for the construction of domain and user models to be reused in the development of multi-agent applications. ONTODUM is an ontology-based meta domain model representing the knowledge of techniques for the specification of the requirements of a family of multi-agent systems in an application domain.

The technique is being evaluated and improved through the specification of domain models in the legal, tourism and information access domains.

ONTODUM and the domain models we are constructing with it are being used in the definition of a technique based on patterns [19] and ontologies for the construction of multiagent frameworks [12] [13]. The final goal is the specification of a methodology for Agentbased Domain Engineering, and a framework for ontology-based specification of agent-based reusable artefacts, exploring both compositional, like the one introduced in this paper, and generative approaches. In our generative approach, domain models are being used as the main resources for the construction of Domain Specific Languages [17] [43] [44].

Some of the advantages of using ontologies for the representation of reusable products have been shown in this article. Although ONTODUM has been designed for its integration in a software development environment for Multi-agent Domain Engineering, the approach can be generalized to other development paradigms. For that, the ontology should be redesigned according to the particular knowledge of those development paradigms techniques.

6 References

- [1] Arango, G. "Domain Engineering for Software Reuse". Ph.D. Thesis, Department of Information and Computer Science, University of California, Irvine, 1988.
- [2] Bracha, Shapira, Peretz, Shoval and Uri, Hanari. Stereotypes in Information Filtering Systems, Information Processing and Management 33(3): 273-287, 1997.
- [3] Caire, G., et alii. Agent-Oriented Analysis using MESSAGE/UML, In: M. Woodridge, P. Ciancarini, and G. Weirs, editors. Second International Workshop on Agent-Oriented Software Engineering, AOSE 2001, pp. 101-108, 2001.
- [4] Castro, Jaelson, Kolp, Manuel, Mylopoulus, John. A Requirement-Driven Software Development Methodology, 13th International Conference on Advanced Information Systems Engineering CAISE01, Interlaken, Switzerland, pp. 4-8, June 2001.
- [5] Chandrasekaran, B. and Josehson, J. "What are Ontologies, and Why Do We Need Them?", IEEE Intelligent Systems, Vol 14, nº1, pp. 20 26, 1999.
- [6] Cerveira, N. "Development of an Ontology-based Domain Model for the Touristical Área", CEAPS-DEINF-UFMA, Final degree work, 2003. (In Portuguese).

- [7] Cossentino, M., Burrafato, P., Lombardo, S., Sabatucci, S., Introducing Pattern Reuse in the Design of Multi-Agent Systems, AITA02, Workshop at NODE 02, Erfurt, Germany, pp. 8-9, October 2002.
- [8] De Almeida Falbo, R.; Guizzardi, G. and Duarte, K. C., "An Ontological Approach to Domain Engineering", In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE'92), Ischia, Italy, July 15-19, 2002.
- [9] Dileo, J., Jacobs, T., Deloach, S. Integrating Ontologies into Multi-Agent Systems Engineering, Proceedings of 4th International Bi-Conference Workshop on Agent Oriented Information Systems (AOIS 2002), pp. 15-16, Bologna (Italy), July 2002.
- [10] DeLoach, S.: Multi-Agent Systems Engineering: A Methodology and Language for Designing Agent Systems In: Proceeding of the 1st Bi-Conf. Workshop on Agent-Oriented Information Systems (AOIS'99), Heidelberg, Germany, 1999.
- [11] Faria, C. "A Technique for the Acquisition and Construction of Ontology-based Domain and User Models in Multi-Agent Domain Engineering", Master degree dissertation, Federal University of Maranhão – CPGEE, 2004. (In Portuguese).
- [12] Ferreira, S. "A Technique and Tool for Domain Design in Multi-Agent Domain Engineering", Master degree dissertation, Federal University of Maranhão – CPGEE, 2004. (In Portuguese).
- [13] Ferreira, S. and Girardi, R. "Specification of a Generic Ontology for Domain Design", Proceedings of III Workshop of Software Engineering (WIS 2003) at Chilean Computing Week 2003. Chilán, Chile. November 3rd to 8th, 2003. (In Portuguese)
- [14] Fleming, M. and Cohen, R., User Modeling in the Design of Interactive Interface Agents, Proceedings of UM99 (User Modeling), 1999.
- [15] R. Girardi, "Agent-based Software Engineering", Proceedings of the 4th Brazilian Congress on Computer Science, Itajaí, Santa Catarina, Brasil (To appear) (In Portuguese)
- [16] Girardi, R. and Serra, "Identifying and Specifying Variability in Domain Models", 2004 (Submitted paper)
- [17] Girardi, R. and Serra, I. "Using Ontologies for the Specification of Domain-Specific Languages in Multi-Agent Domain Engineering", Proceedings of the 2004 Agent_Oriented Information Systems Workshop (AOIS 2004), 2004 Conference on Computer Aided Software Engineering, Riga, 2004 (To appear)
- [18] Girardi, R. and Faria, C. "A Generic Ontology for the Specification of Domain Models", Proceedings of 1st International Workshop on Component Engineering Methodology (WCEM'03) at Second International Conference on Generative Programming and Component Engineering, Ed. Sven Overhage and Klaus Turowski, pp. 41-50. Erfurt, Germany. September 22-25, 2003.
- [19] Girardi, R, Ribeiro, I., and Bezerra, G. "Towards a System of Patterns for the Design of Agent-based Systems". Proceedings of The Second Nordic Conference on Pattern Languages of Programs" (VikingPLoP 2003). Bergen, Norway 2003.
- [20] Girardi, R. "Multi-Agent Domain Engineering", Proceedings of the 3th. Ibero-American Symposium on Software Engineering and Knowledge Engineering (JIISIC 2003), Cooperation Session, Universidad Austral de Chile. November 26th to 28th, pp. 15-20, 2003.
- [21] Girardi, R., "Reuse in Agent-based Application Development", Proceedings of 1st International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'2002), Internationa Conference on Software Engineering (ICSE'2002), May 2002.
- [22] Girardi R. and Sodré A. A Methodology for Multi-Agent Application Development, In: Proceedings of the ITS'2002 Workshops - Architectures and Methodologies for Building Agent-Based Learning Environments, 6th International Conference on Intelligent Tutoring Systems, www.fapeal.br/its2002workshop_agents, 2002.
- [23] Gruber, T. R. "Towards Principles for the Design of Ontologies used for Knowledge Sharing". International Journal Human-Computer Studies. Nº 43, pp. 907-928, 1995.
- [24] Guarino, N. "Formal Ontology and Information Systems", Proceedings of FOIS'98, Trento, Italy. Amsterdan, IOS Press, pp. 3-15, 6-8 June, 1998.
- [25] Harsu, M. "A Survey on Domain Engineering", Report 31, Institute of Software Systems, Tampere University of Technology, December 2002, 26 pp.
- [26] Iglesias, C. et al., A Survey of Agent-Oriented Methodologies, In: Intelligent Agents V. Agents Theories, Architectures, and Languages, Lecture Notes in Computer Science, Springer-Verlag, 1998.
- [27] Jones, Dean, Bench Capon, Trevor and Visser, Pepjin. Methodologies for Ontology Development, Proc. It&Knows Conference of the 15th IFIP World Computer Congress, Budapest, Chapman-Hall, 1998.
- [28] Kobsa, Alfred. Personalised hypermedia presentation techniques for improving online customer relationships. GMD Report 66, 1999.
- [29] Krut, R. and Zalman, N. Domain Analysis, Workshop Report for the Automated Prompt & Response System Domain (CMU/SEI-96-SR-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.

- [30] Lindoso, A., Serra, I. and Girardi, R. "ONTOINFOJUS: A Domain Model based on Ontologies for Information Access in the Juridical Area", In: Proceedings of ECOINFO 2003. Palmas, Tocantins, October 29th to 30th, pp. 251-260, 2003. (In Portuguese)
- [31] Lindoso, A. "Specification of a Domain Ontology for the Legal Domain", CGCC-UFMA, Final degree work, 2003. (In Portuguese)
- [32] Moore, J. M. and Bailin, S. C. "Domain Analysis: Framework for reuse". IEEE Computer Society Press. Tutorial, pp. 179-202, 1991.
- [33] Musen, M. A. "Domain Ontologies in Software Engineering: Use of Protégé with the EON Architecture", Methods of Information in Medicine. Edição 37. Nº: 4-5, pp. 540-550, 1998.
- [34] Noy, N. F. and McGuinness, D. L. "Ontology Development 101: A Guide to Creating Your First Ontology". Knowledge Systems Laboratory, March 2001.
- [35] Odell, J., Parunak, H.V.D., and Bauer, B. Extending UML for Agents. Proc. Of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence, accepted role, AOIS Workshop at AAAI pp. 3-17, 2000.
- [36] Oliveira. J., A., "An Analysis of Machine-Learning Techniques for User Modeling", CGCC-UFMA, Final degree work, 2004. (In Portuguese)
- [37] Prieto-Diaz, R. "Domain Analysis and Software System Modeling". Los Alamitos, CA: IEEE Computer Society Press, 1991.
- [38] Prieto-Díaz, R. "Classifying Software for Reuse", IEEE Software, Vol. 4, nº01, January 1987.
- [39] Ribeiro, J. "Development of an Ontology-based Domain Model for the Pecuary Área", CEAPS-DEINF-UFMA, Final degree work, 2004. (In Portuguese).
- [40] Shlaer, S. and Mellor, S. An Object-Oriented Approach to Domain Analysis, ACM SIGSOFT Software Engineering Notes, Vol. 14, N° 5, pp. 66-77, July 1989.
- [41] Sobrinho, A. C. and Girardi, R. "An analysis of the applications of Genetic Algorithms on User-adaptable Systems", REIC Journal, Ed. SBC v. III, n. IV, 2003.
- [42] The Protégé Project, http://protege.stanford.edu. Accessed on April 26, 2004
- [43] Serra, I. and Girardi, R. "A Generative Approach to Multi-Agent Domain Engineering", In: Proceedings of ECOINFO 2003. Palmas, Tocantins, Brasil. October 29th to 30th, 2003. (In Portuguese)
- [44] Serra, I. "A Technique for the Development of Domain Specific Languages", Master degree dissertation, Federal University of Maranhão CPGEE, 2004. . (In Portuguese).
- [45] Wooldridge, M, . Jennings, N. R. and Kinny, D. The Gaia methodology for agent-oriented analysis and design, International Journal of Autonomous Agents and Multi-Agent Systems 2000.
- [46] Wood, M., and DeLoach, S.A. : An Overview of the Multi-Agent Systems Engineering Methodology, in: Proc. of the First Int. Workshop on Agent-Oriented Software Engineering (AOSE-2000), Limerick, Ireland, 2000.
- [47] Woodridge, M., Cicarini, P. Agent-Oriented Software Engineering: the State of the Art, In P. Cicarini and M. Woodridge, editors, Agent Oriented Software Engineering, Springer-Verlag, 2001.