Ontology-based Domain Modeling of Multi-Agent Systems

Rosario Girardi, Carla Gomes de Faria, Leandro Balby

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

Abstract. Ontologies are knowledge representation structures especially useful for the specification of high-level reusable software, like domain models and frameworks. This work describes GRAMO, an ontology-based technique for the specification of domain models in Multi-Agent Domain Engineering. ONTODM, an ontology-based tool supporting GRAMO is also introduced. ONTODM represents the knowledge of GRAMO. A case study on the application of GRAMO for domain modeling of usage mining-based Web personalization systems is also described.

1 Introduction

Domain Engineering [1] [24] is a process for creating reusable software abstractions for the development of a family of software applications in a domain. The process consists of the phases of analysis, design and implementation. Domain analysis [31] [36] 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 [16] [17] [18] [19] [20]. We are constructing a software development environment composed of a set of development tools and libraries of high-level reusable software abstractions (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 access and user modeling [6] [14] [16] [27] [37] [40].

Ontologies [5] [20] [23] are knowledge representation structures especially useful for the specification of high-level reusable software, like domain, user models and

frameworks. 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. Ontologies are being used to represent both the knowledge of techniques for Domain Engineering and generated products [13] [15] [17].

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

This work introduces the technique GRAMO, and the reusable ontology ONTODM. The article is organized as follows. Section 2 introduces the GRAMO technique. Section 3 describes the ONTODM reusable ontology. Section 4 introduces a case study on the application of GRAMO for domain modeling of usage mining-based Web personalization systems. Section 5 describes related work on Domain analysis and Requirement Engineering of Multi-Agent Systems. Section 6 concludes the paper with some remarks on further work being conducted.

2 The GRAMO Technique

GRAMO [13] is an ontology-based technique for the specification of domain models in the analysis phase of Multi-Agent Domain Engineering, according to the knowledge of a particular application domain. Domain models are constructed through the instantiation of ONTODM [17], a reusable ontology that encodes the knowledge of the GRAMO technique (Fig. 1).

Main techniques for Domain Analysis [26] [31] [38] and Requirement Engineering of multi-agent systems [3] [4] [8] [10] [19] [20] [25] [33] [34] [41] [42] [43] were considered for the definition of the GRAMO technique.

Methods for Requirement Engineering of multi-agent systems usually focus on modeling goals, roles, activities and interactions of individuals of an organization. An organization is composed of individuals. The organization has 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.

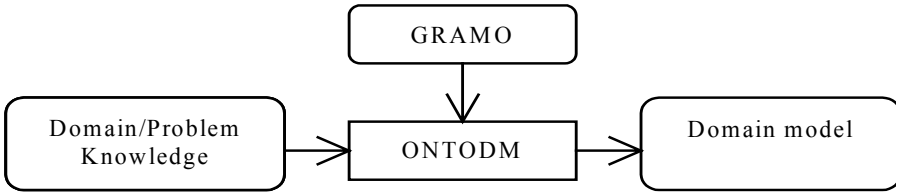

Fig. 1. Inputs and products of GRAMO through the instantiation of ONTODM

According to these definitions, in GRAMO, domain modeling of multi-agent systems is performed according to the following modeling tasks: Concept Modeling, Goal Modeling, Role Modeling, Variability Modeling and Interaction Modeling.

Domain Modeling approaches either the formulation of a problem (e.g. User Modeling) or the representation of a knowledge area (e.g. the tourism area). For the formulation of a problem, the tasks Goal Modeling, Role Modeling, Variability Modeling and Interaction Modeling are performed. A Domain Model is obtained composed of one Goal Model, a set of Role Models and a set of Interaction Models. For the representation of a knowledge area only the Concept and Variability Modeling tasks are performed. A Domain Model is obtained which consists of the developed Concept Model.

Goal modeling. Considering the problem that the system intends to solve, the general goal of the system is identified. Specific goals are obtained through a refinement of the general goal. The responsibilities that need to be exercised by internal and external roles to reach an specific goal are identified. A goal model is constructed as a product of this modeling task composed of the general and specific goals and the responsibilities of the family of systems. The Goal Model is represented graphically in a three level organizational chart. General goal, specific goals and responsibilities are represented in the first, second and third level, respectively (e.g. Fig. 3).

Role modeling. Each responsibility identified on the Goal Modeling task is associated with an internal or external role. 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 inputs, outputs and resources needed for the execution of each activity are identified. A particular role model is constructed for each defined role. The role model is represented graphically in a three level organizational chart (e.g. Fig. 4). The responsibility, activities and resources of the role are represented in the first, second and third level, respectively. Through the composition of the particular role models, a general role model is also constructed.

Variability modeling. According to the following rules, the goal, role models and concept models are refined to classify the domain concepts, goals, roles, responsibilities, activities and resources as fixed or variables features. Fixed features will be present in all subsystems of a family of systems in a domain or problem-solving area. Variable features represent specific characteristics of a particular system in the family (e.g. Table 1).

- *The general goal is fixed.* By default, the general goal of a multi-agent system is a fixed concept because it must be reached by all systems of a family.
- *Specific goals are variable*. A particular system in the family does not need to satisfy, necessary, all the specific goals. Then, some specific goals have not need to be considered in a particular system. Therefore, specific goals are classified as variable concepts.
- *Activities can be variable or fixed*. Roles exercise their responsibilities through the execution of activities. Activities that must be performed in all systems of a family are classified as fixed. Otherwise, they are classified as variable.
- *Responsibilities can be variable or fixed*. Specific goals are reached through the exercise of responsibilities. If a responsibility contributes for the achievement of all the specific goals and it is exercised through the execution of fixed activities, then it is classified as fixed. Otherwise, it is classified as variable.
- *Roles can be variable or fixed*. Responsibilities are exercised by roles. Then, roles are classified according to their responsibilities: roles are variable is they exercise variable responsibilities. Otherwise, they are fixed.
- *Resources can be variable or fixed*. For the execution of activities, roles dispose of resources. Then, a resource is classified as fixed if at least one fixed activity needs it. Otherwise, it is classified as variable.
- *Domain concepts can be variable or fixed*. Domain concepts that are present in all systems of a family are classified as fixed. Otherwise, they are variable.

Interaction modeling. Through an analysis of their respective activities along with their inputs and outputs, the interactions between all internal and external roles are identified. Considered roles are those whose responsibilities lead to the achievement of a specific goal. One interaction model for each specific goal is constructed as a product of this modeling task. The graphical representation of the interaction model is similar to the collaboration diagram of UML [2] (e.g. Fig. 5).

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 model of concepts is constructed as a product of this modeling task. The model of concepts is represented

graphically in a semantic network where nodes represent concepts and links show the relationships between concepts.

3 ONTODM: An Ontology-based Tool for Domain Modeling

The knowledge of the GRAMO technique has been represented in an ontology, ONTODM, which guides the elicitation and specification of the concepts and tasks to be accomplished in a domain. ONTODM has been developed with Protégé, an environment for the development of frame-based ontologies [39]. Fig. 2 shows the hierarchy of classes of ONTODM, and the slots of the Domain Modeling class.

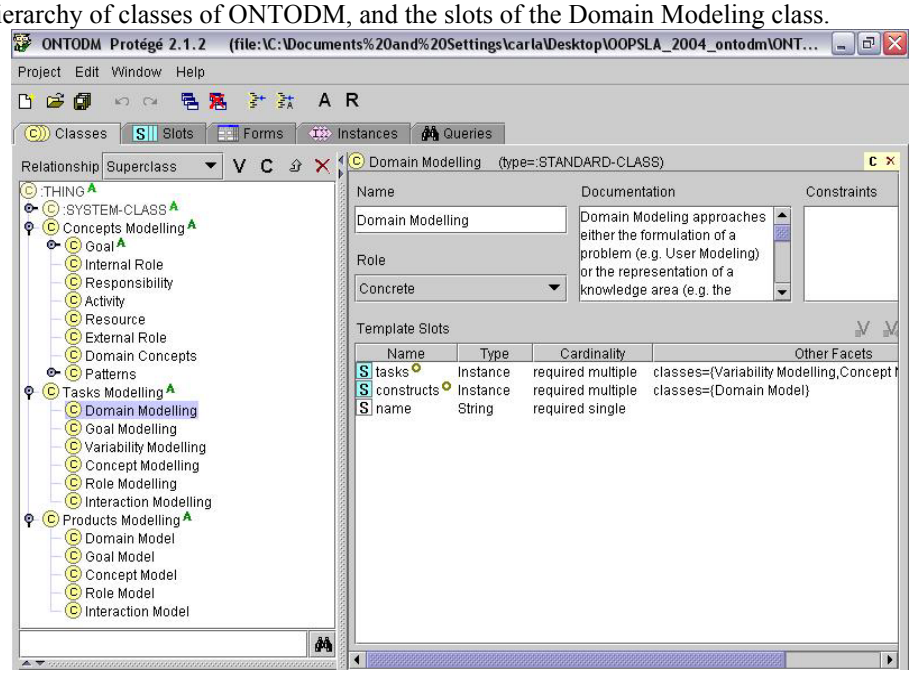

Fig. 2. Hierarchy of classes of ONTODM

The construction of a domain model is made through the instantiation of the class Domain Modeling (Fig. 2) that will create an instance of the Domain Model class, containing the specification of the concepts and tasks in the domain. This requires the instantiation of the classes Concept, Goal, Role and Interaction Modeling, which will create the instances Concept, Goal, Role and Interaction Models.

4 A Case Study

In order to evaluate the GRAMO technique, several case studies have been developed. These case studies approached the construction of ONTOMINING, ONTOINFO [40], ONTOTOUR [6] and ONTOPEC [37] domain models by reusing

ONTODM through the application of the GRAMO technique. These ontology-based domain models represent the common requirements of families of software applications for usage mining personalization systems (ONTOMINING); information retrieval and filtering (ONTOINFO), and the concepts of the legal (ONTOJUS), tourism (ONTOTOUR) and pecuary (ONTOPEC) domains.

Next section introduces a case study where the GRAMO technique is applied on the construction of ONTOMINING.

4.1 ONTOMINING: Domain Modeling of Usage Mining-based Web Personalization Systems

Web site personalization can be defined as the process of customizing the content and structure of a Web site to the specific and individual needs of each user, taking advantage of the user navigational behavior. The steps of a Web personalization process include: (a) the collection of Web data, (b) the modeling and categorization of these data (preprocessing phase), (c) the analysis of the collected data, and (d) the determination of the actions that should be performed. The ways that are employed in order to analyze the collected data include content-based filtering, collaborative filtering, rule-based filtering, and Web usage mining [7] [29] [30] [35]. A site is personalized through the highlighting of existing hyperlinks, the dynamic insertion of new hyperlinks that seem to be of interest for the current user, or even through the creation of new index pages. Here, we focus on the process of usage mining in the context of Web personalization. This process relies on the application of statistical and data mining methods to Web log data, resulting in a set of useful patterns that indicate user navigational behavior. This knowledge is then used from the system to personalize a site according to each user behavior and profile.

Goal modeling. The general goal of a family of systems for Web personalization based on usage mining is, obviously, *Web personalization*. The general goal is reached through the specific goals: *User modeling* and *System adaptation*. A user model is the main component of an adaptive system. The user model keeps information on a user, reflects their interests and preferences and, furthermore, influences the adaptation of the system. The adaptation model takes care of the adaptive features of the system and affects the way the adaptation effects are displayed to the user. The specific goal *User modeling* is reached through the following responsibilities: *Usage mining*, *User interfacing* and *Construction and maintenance of user models*. The specific goal *System adaptation* is reached through the responsibilities *User interfacing*, *Construction and maintenance of user models* and *Construction and maintenance of adaptation models*. Fig. 3 shows the goal model generated from ONTODM.

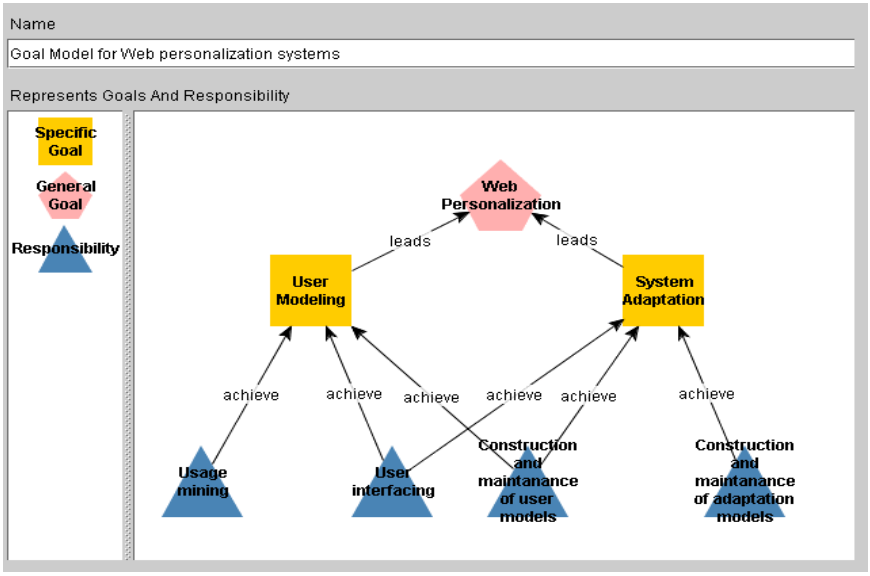

Fig. 3. Goal model for a family of Web personalization systems

Role modeling. There were identified two external roles, *User* and *Usage Data Repository,* and four internal roles, *Modeling, Interfacing, Usage miner and Adaptation*. The *Modeling* role is responsible for the *Construction and maintenance of user models*. This responsibility is exercised by the activities *User model representation* and *User model updating.* The *Interfacing* role is responsible for *Interfacing the user* with the system. This responsibility is exercised by the activities *Application of adaptation effects*, *User monitoring* and *Generation of semantic usage data*. This later activity looks for the improvement of the quality of usage data through the semantic enrichment of Web logs [32]. The *Usage miner* role is responsible for the application of data mining techniques in the usage data. This responsibility is exercised by the activities *Data collection*, *Data filtering*, *User identification*, *User session identification* and *Pattern discovery* (Fig. 4). The *Adaptation* role is responsible for the *Construction and maintenance of adaptation model* goal. This responsibility is exercised by the activities *Adaptation model representation* and *Adaptation model update.*

Interaction modeling. Two interaction models are constructed; one for the specific goal *User modeling* and another for the specific goal *System adaptation*. Fig. 5 shows the interaction model for the specific goal *User modeling.* The sequence starts with the collection and pre-processing of usage data by the usage miner role from the *Usage data repository*. The next step is user monitoring by the *Interfacing* role. This role also generates semantic usage data based on the user navigational behavior. Then, the *Interfacing* role sends information about the current user to the *Modeling* role. Finally, the *Modeling* role interacts with the Usage miner role to classify the current user.

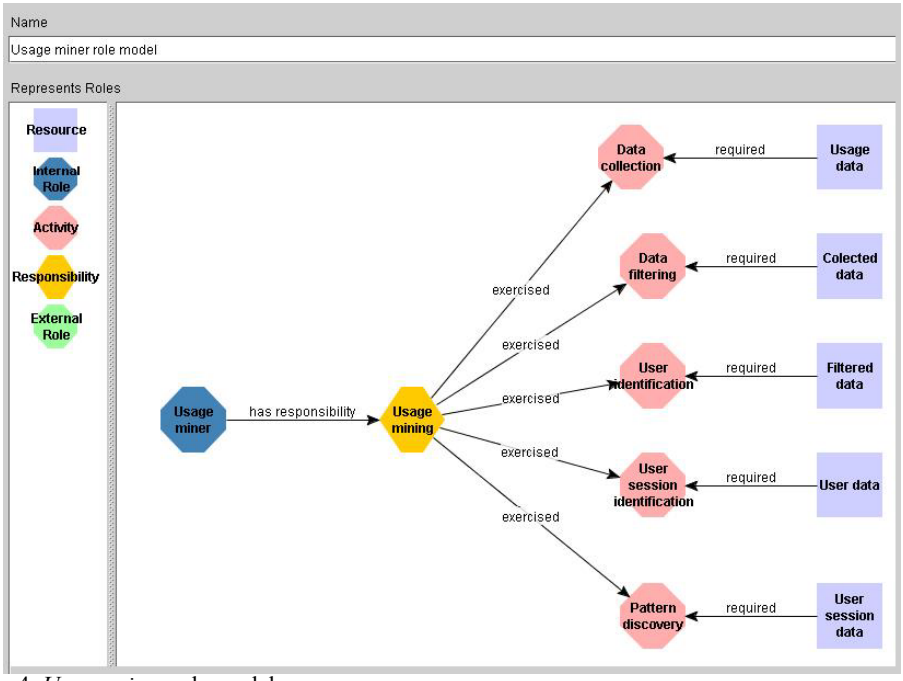

Fig. 4. *Usage miner* role model

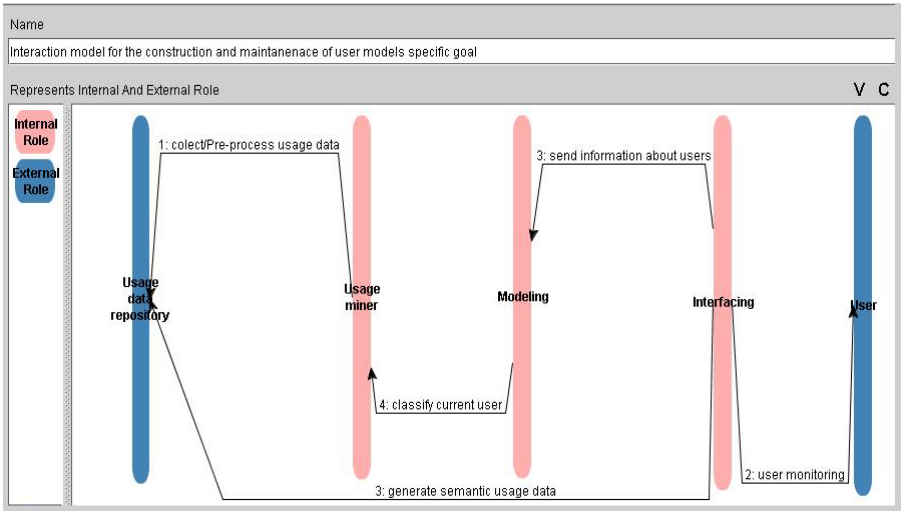

Fig. 5. Interaction model for the specific goal *User modeling*

Variability modeling. In this activity the instances of the classes *Concept*, *Responsibility*, *Role*, *Activity*, *Specific goal* and *General goal* are classified in fixed or variable (Table 1) according to the rules introduced in section 2 above.

Table 1. Variability modeling of the problem of usage mining for Web personalization

Fixed features				
General goal	**Responsibilities**	**Roles**	**Activities**	**Resources**
Web personalization	User interfacing	Interfacing	• Application of adaptation effects • User monitoring • Generation of semantic usage data	User information
	Construction/maintenance of user models	Modeling	• User model representation • User model updating	User processed information
Variable features				
Specific goal	**Responsibilities**	**Roles**	**Activities**	**Resources**
User modeling	Usage mining	Usage miner	• Data Collection • Data filtering • Data pre-processing • Pattern discovery	• Usage data • Collected data • Filtered data • User data • User session data
System adaptation	Construction/maintenance of adaptation models	Adaptation	• Adaptation model representation • Adaptation model updating	Adaptation information

5 Related Work

Considering their advantages, the use of ontologies for the representation of high-level reusable specifications, like requirement and design specifications is increasing. Particularly, some methodologies for the development of multi-agent specific applications are being extended for ontology support [11][12]. Some approaches for Ontology-based Domain Analysis have been proposed [9] [28], which integrate methodologies for ontology building with techniques for Domain Analysis.
Previous work on knowledge-based representation of software components to increase the effectiveness of software retrieval have also contributed to this proposal [10] [21].

ONTODM and the domain models we are constructing with it have been used for the definition of DDEMAS, a technique based on patterns and ontologies for the construction of multi-agent frameworks [15]. Domain models are also being used as the main resources for the construction of Domain Specific Languages [16].

6 Concluding Remarks and Further Work

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

Some of the advantages of using ontologies for the representation of reusable products have been shown in this article. Although ONTODM 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 re-designed according to the particular knowledge of the techniques of those development paradigms.

We are currently working on a process and development environment for Multi-Agent Domain Engineering through the integration of GRAMO with DDEMAS [15], a technique for the construction of multi-agent frameworks, and TOD-DSL [16] a technique for the development of Domain-Specific Languages. We are also working on techniques for Multi-Agent Application Development to construct specific software applications through the reuse of all the reusable ontology-based software abstractions we have developed.

7 Acknowledgements

This work is supported by CNPq, an institution of Brazilian Government for scientific and technologic development.

References

1. Arango, G.: Domain Engineering for Software Reuse, Ph.D. Thesis, Department of Information and Computer Science, University of California, Irvine, 1988.
2. Booch, G., Rumbaugh, J. and Jacobson, I.: Unified Modeling Language User Guide. Reading: Addison Wesley, 1999
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, J., Kolp, M. and Mylopoulus, J.: 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 Tourism Área, CEAPS-DEINF-UFMA, Final degree work, 2003. (In Portuguese).
7. Cooley, R. W.: Web usage mining: Discovery and application of Interesting Patterns from Web data, PhD thesis, Dept. of Computer Science, University of Minesota, 2000.
8. Cossentino, M., Burrafato, P., Lombardo, S. and Sabatucci, S.: Introducing Pattern Reuse in the Design of Multi-Agent Systems, AITA02, Workshop at NODE 02, Erfurt, Germany, pp. 8-9, October 2002.

9. 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, It-aly, July 15-19, 2002.
10. Devanbu, P., Brachman, R. J., Selfridge, P. G., Ballard, B. W.: LaSSIE: a Knowledge-based Software Information System. ICSE 1990: 249-261, 1990.
11. Dileo, J., Jacobs, T. and 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.
12. 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.
13. 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).
14. Faria, C., Girardi, R. and Vieira, R.: An Ontology-based Domain and User Model for the Tourism Domain, Proceedings of the 4th Brazilian Computing Congress (CBCOMP 2004). Itajaí, Santa Catarina, Brasil. October, 08-12, 2004. (To appear in Portuguese)
15. 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)
16. Girardi, R. and Serra, I.: Using Ontologies for the Specification of Domain-Specific Languages in Multi-Agent Domain Engineering, Proceedings of the Sixth International Bi-Conference Workshop on Agent-oriented Information Systems (AOIS-2004) at The 16th International Conference on Advanced Information Systems Engineering (CAISE'04), Ed. Janis Grundspenkis and Marite Kirikova (Eds.), pp. 295-308. Riga, Latvia. 07 a 11 de junho de 2004.
17. 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.
18. 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.
19. Girardi, R.: Reuse in Agent-based Application Development, Proceedings of 1st International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS´2002), International Conference on Software Engineering (ICSE´2002), May 2002.
20. 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.
21. Girardi R.; Ibrahim, B. Using English to Retrieve Software. The Journal of Systems and Software**,** v. 30, n. 3, p. 249-270, 1995.
22. 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.
23. Guarino, N.: Formal Ontology and Information Systems, Proceedings of FOIS'98, Trento, Italy. Amsterdan, IOS Press, pp. 3-15, 6-8 June, 1998.
24. Harsu, M.: A Survey on Domain Engineering, Report 31, Institute of Software Systems, Tampere University of Technology, December 2002, 26 pp.
25. Iglesias, C. et alii.: A Survey of Agent-Oriented Methodologies, In: Intelligent Agents V. Agents Theories, Architectures, and Languages, Lecture Notes in Computer Science, Springer-Verlag, 1998.
26. 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.
27. 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)
28. Marçal de Oliveira, K.; Zlot, F; Rocha, A. R. ; Horta Travassos, G.; Galotta, C. and Silva de Menezes, C. "Domain-oriented software development environment", The Journal of Systems and Software, 72(2004), 145-161.

29. Marinho, L. B. and Girardi, R.: Web Mining, REIC Journal, Ed. SBC, v. III, n. II, Jun. 2003. (In Portuguese)
30. Mobasher, B., Cooley, R. and Srivastave, J.: Automatic personalization based on Web usage mining, Technical Report TR99010, Department of Computer Science, DePaul University, 1999.
31. Moore, J. M. and Bailin, S. C.: Domain Analysis: Framework for reuse,. IEEE Computer Society Press. Tutorial, pp. 179-202, 1991.
32. Oberle, D., Berendt, B., Hotho, A. and Gonzalez, J.: Conceptual User Tracking, In the Proceedings of the First International Atlantic Web Intelligence Conference (AWIC 2003), Madrid, Spain, May 5-6, 2003.
33. 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.
34. Omicini, A.: SODA Societies and Infrastructures in the Analysis and Design of Agent-based Systems, Proceedings of First International Workshop, AOSE 2000 on Agent-Oriented Software Engineering, Limerick, Ireland, pp. 185-193, January, 2001.
35. Pierrakos, D., Paliouras, G., Papatheodorou, C. and Spyropolous, C. D.: Web Usage Mining as a Tool for Personalization: A Survey, User Modeling and User-Adapted Interaction 13: 311-372, 2003.
36. Prieto-Diaz, R.: Domain Analysis and Software System Modeling, Los Alamitos, CA: IEEE Computer Society Press, 1991.
37. Ribeiro, J.: Development of an Ontology-based Domain Model for the Pecuary Área, CEAPS-DEINF-UFMA, Final degree work, 2004. (In Portuguese).
38. Shlaer, S. and Mellor, S.: An Object-Oriented Approach to Domain Analysis, ACM SIGSOFT Software Engineering Notes, Vol. 14, Nº 5, pp. 66-77, July 1989.
39. The Protégé Project, http://protege.stanford.edu. Accessed on April 26, 2004.
40. Serra, I., Girardi, R. and Silva, J.: An Ontology-based Domain Model for Information Filtering and Retrieval, Proceedings of the 4th Brazilian Computing Congress (CBCOMP 2004). Itajaí, Santa Catarina, Brasil. October, 08-12, 2004. (To appear in Portuguese)
41. 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.
42. 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.
43. Woodridge, M. and 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.