IMPLEMENTATION ISSUES OF THE INFONORMA MULTI-AGENT RECOMMENDER SYSTEM

Lucas Drumond, Rosario Girardi, D'jefferson Maranhão and Geraldo Abrantes
Department of Computer Science, Federal University of Maranhão, Avenida Dos Portugueses, São Luís, Brazil

Keywords: Recommender systems, Content based filtering, Collaborative filtering, Legal information systems, JADE.

Abstract: Recommender systems can help professionals of the legal area to deal with the growth and dynamism of legal information sources. Infonorma is a multi-agent recommender system that recommends legal normative instruments to users according to their particular interests using both content-based and collaborative information filtering techniques. It has been modeled under the guidelines of the MAAEM methodology. This paper discusses the main implementation issues of the Infonorma system.

1 INTRODUCTION

Recommender systems (Adomavicius and Tuzhilin, 2005) are a particular type of information filtering applications that can be used as an approach to the problem of storing, retrieving and structuring large amounts of legal information to effectively satisfy information needs of legal users.

Recommender systems perform tasks like inferring user preferences and monitoring and reasoning upon available information sources that are inherently connected to the autonomous behavior of software agents. Because of that, the multi-agent paradigm is a reasonable approach for developing recommender systems.

Infonorma is a multi-agent recommender system for the legal domain that uses content based filtering (Balabanovic and Shoham, 1997) for recommending legal normative instruments to users according to their particular interests. The knowledge about the legal domain used by Infonorma is represented in the ONTOJURIS ontology, which describes the structure of a legal normative instrument as well as the legal areas of Brazilian law. Ontologies constitute the knowledge representation standard for the Semantic Web (Antoniou and Van Harmelen, 2004)(Shadbolt and Berners-Lee, 2006) and the ONTOJURIS ontology is written in OWL, according to the W3C recommendation.

The Infonorma system was developed under the guidelines of the Multi-Agent Application Engineering Methodology (MAAEM) (Lindoso and Girardi, 2006). The MAAEM methodology is composed by three modeling phases: application analysis, design and implementation. The analysis and design of Infonorma are discussed in (Drumond and Girardi, 2008). This paper introduces the implementation phase of the Infonorma multi-agent recommender system.

This paper is organized as follows. Section 2 presents an overview of the MAAEM methodology. Section 3 introduces the Infonorma System with a brief discussion of its specification and design. Section 4 discusses the implementation phase of Infonorma according to the guidelines of the MAAEM methodology. Section 5 presents a brief discussion on related work. Finally, section 6 concludes this paper with some remarks on further work being conducted.

2 OVERVIEW OF MAAEM METHODOLOGY

MAAEM is a methodology for requirement analysis, design and implementation of multi-agent applications through the reuse of software artifacts such as domain models, multi-agent frameworks, pattern systems and software agents. MAAEM also supports the development of applications without reuse, as is the case of the development of Infonorma. The ONTORMAS ("ONTOlogy driven tool for the Reuse of Multi-Agent Systems")(Girardi and Leite, 2008)(Leite and Girardi, 2008) ontology

Drumond L., Girardi R., Maranhão D. and Abrantes G. (2009).
IMPLEMENTATION ISSUES OF THE INFONORMA MULTI-AGENT RECOMMENDER SYSTEM.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 128-133
DOI: 10.5220/0002008701280133

along with the Protégé platform is used as an ontology-driven modeling tool and a storage repository for products constructed on the Multi-agent Domain Engineering and Multi-agent Application Engineering processes. All the models presented in this article were built using ONTORMAS.

For the specification of a design solution, roles are assigned to agents structured and organized into a particular multi-agent architectural solution according to non-functional requirements. The interactions between the agents must be modeled as well as the knowledge shared by them.

Application Analysis is performed through the following modeling tasks: concept modeling, goal modeling, role modeling, role interaction modeling and user interface prototyping.

Application Design approaches the architectural and detailed design, defining a solution to the requirements specified in the analysis phase.

Application Implementation approaches the mapping of design models to agents, behaviors and communicative acts, concepts involved in the JADE framework (Bellifemine et al., 2003), which is the adopted implementation platform. The modeling tasks and respective products of the implementation phase of the MAAEM methodology are shown in Table 1.

The information source used by Infonorma is a legislative repository, a repository composed by normative instruments. Such instruments have a type (the nature of the instrument e.g. *Ordinary Law*, *Complementary Law*, etc.) and legal branches in which they are classified. The users specify their interests in terms of such types and legal branches.

Figure 1 shows the goal model of Infonorma, in which the general goal of providing legal normative recommendations is achieved by three specific goals, namely *Model legal users*, *Content based filter new legal information* and *Deliver recommendations*. The fulfillment of these specific goals requires the exercise of some responsibilities that are assigned to the agents in the system.

3 OVERVIEW OF INFONORMA SPECIFICATION AND DESIGN

Infonorma is a system that provides its users with personalized recommendations of legal normative instruments. Legal normative instruments are documents that compose the Brazilian jurisprudence. Their structure is represented in the ONTOJURIS domain ontology.

Each legal user has a profile composed by his/her own interests and identification and represented by a user model as an instance of ONTOJURIS.

The Infonorma system is composed by two kinds of agents: one *Filter* agent, in charge of the *Legislative repository monitoring, Information items classification into legal branches* and *Content based similarity analysis* responsibilities and one *User Modeler* agent for each user, in charge of the *Explicit profile acquisition*, *User model creation* and *Filtered information delivery* responsibilities.

For a detailed description of the Infonorma specification and design, please refer to (Drumond and Girardi, 2008).

Table 1: Modeling tasks and products of the Implementation phase of the MAAEM methodology.

<table>
<tr><th>Phases</th><th>Tasks</th><th colspan="2">Products</th></tr>
<tr><td rowspan="3">Application Implementation</td><td>Mapping from design to Implementation Agents and from Responsibilities to Behaviors</td><td>Model of Agents and Behaviors</td><td rowspan="2">Application Implementation Model</td></tr>
<tr><td>Mapping of Agents Interaction and Communication Acts</td><td>Model of Agent Communicative Acts</td></tr>
<tr><td>Agent Implementation</td><td colspan="2">Executable Software Agents</td></tr>
</table>

4 IMPLEMENTATION OF INFONORMA

As stated earlier, the Infonorma system was implemented using the JADE platform. The interactions of the user with the system take place through a web site. The actions of the user in the web site are recorded in a database monitored by the *User modeler* agents. This section discusses the main implementation issues of the agents of the Infonorma system.

4.1 Mapping from Design to Implementation Agents and from Responsibilities to Behaviors

In this phase, the design agents are mapped to implementation agents. As the implementation

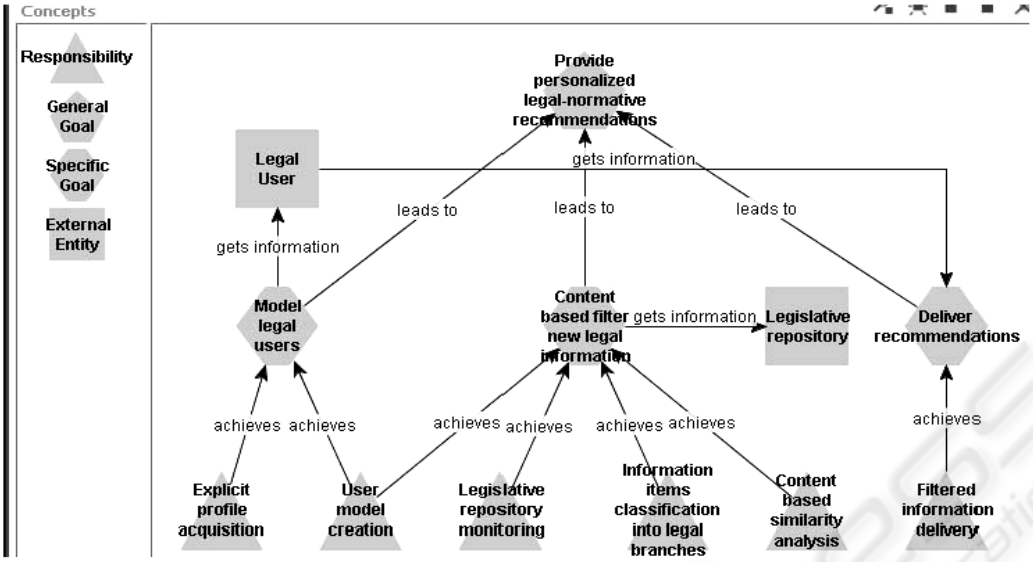

Figure 1: Goal Model of Infonorma.

platform adopted in Infonorma is the JADE framework, each implementation agent is a JADE agent. The responsibilities identified initially in the Goal model are mapped to JADE behaviors and the activities identified in Activity model are mapped to JADE behavior methods.

The *User Modeler* agent is responsible for the *Explicit profile acquisition, Filtered information delivery* and *User model creation* responsibilities. They were mapped, respectively, to the *AcquireUserProfiles*, *DeliverFilteredInformation* and *CreateUserModels* behaviors. The *AcquireUserProfiles* behavior has one main method, the *validateProfile* one, which should verify if new profiles were specified and, in this case, fire the *CreateUserModels* behavior. The *AcquireUserProfiles* behavior was specified as a *One Shot* behavior, since it is performed only once for each new user model specified.

As for each new user profile, only one user model must be created, the *CreateUserModels* behavior also is *One Shot*. For each new profile acquired, the *computeAffinities* method computes the legal affinities, i.e. the level of interest, that the legal user has with the legal branches. After that, the data acquired in the profile and the computed affinities are recorded as an instance of the *User Model* class in the ONTOJURIS ontology, in the context of the *saveLegalUserModel* method.

The *DeliverFilteredInformation* is a *Cyclic* behavior because of the *evaluateProposal* method. This method constantly checks the message queue of the agent waiting for similarity analysis proposals. For each proposal, this method checks whether the user is interested in the type of the proposed information item. When a proposal is accepted and the filtered information items are sent by the *Filter* agent, the *DeliverFilteredInformation* behavior should produce the personalized recommendations (*constructRecommendationMessage* method) and make them available to the user (*deliverRecommendation* method).

The Filter agent, responsible for the Legislative *repository monitoring*, *Information items classification* and *Content-based similarity analysis responsibilities*, has the *MonitorLegislativeRepository*, *ClassifyInformationItems* and *FilterInformationItems* behaviors respectively. Each time the *MonitorLegislativeRepository* behavior is performed, it must check if there are the new information items (*monitorInformationSource* method). This is done through the following query.

Query that checks whether there are new instruments of a certain type in the legislative repository.

```
String queryString =
"PREFIX juris: <" + ontojurisNS + "> \n" +
"SELECT ?num " +
"WHERE {" +
"  ?instrument juris:number ?num . " +
"  ?instrument juris:type ?type . " +
"  ?type juris:abbreviation \"" + abbr +
"\" . " +
"  FILTER (?num > " + number + ") " +
"}";

Vector results = query(queryString);
```

When new items are detected they are identified and the current state of the legislative repository is recorded (*extractNewItems* method). It may seem natural to specify this behavior as a cyclic one, but, as the legislative repository does not change constantly, this behavior can be executed between certain time intervals. Thus it was specified as *Ticker*. As each information item must be classified just one time, the *ClassifyInformationItems* behavior was specified as a *One Shot* behavior. The classification process determines the legal affinity level of the normative instrument with each legal branch in the ONTOJURIS ontology. It starts counting the relevant keywords that are found in the normative instrument (*preprocessing* method). Then, the *computeLegalAffinities* method computes the level of affinity with each legal branch and the *normalizeAffinities* method normalizes the computed values, so that they range from 0 to 1.

computeLegalAffinities Method from the *ClassifyInformationItems* behavior.

```
private void computeLegalAffinities() {

  for(int i = 0; i < filter.legal_branches.size();
       i++){
     Legal_Affinity la = new Legal_Affinity();
     la.setLegal_branch(
            filter.legal_branches.get(i));
     la.setWeight(0);

     Iterator<Case_Generator> it =
            la.getLegal_branch()
               .getAllCase_generators();

     while(it.hasNext()){
        Case_Generator c = it.next();
        float weight = la.getWeight() +
          frequencies.get(c.getName())
            *c.getWeight();
        la.setWeight(weight);
     }
     if(la.getWeight() != 0.0)
        normativeInstrument.addLegal_branches(la);
  }
}
```

Finally, the *proposeSimilarityAnalysis* method sends to every *User Interface* agent a proposal with the type of the current normative instrument.

When the information items are classified, they are filtered by the *FilterInformationItems* behavior, a *One Shot* behavior, since it is executed only once for each new information item. The *similarityAnalysis* method computes the similarity analysis based on the closeness between the user model and the information item. This closeness is measured by a function that measures the distance between them. For a more detailed description of these functions, please refer to (Drumond and Girardi, 2008). The methods that compute both the closeness and the distance between normative instruments are the ones that follow.

Methods for computing the *closeness* and the *distance* between information items and user models.

```
public float closeness(Legal_Branch x,
                       Legal_Branch y){
   return (float) Math.pow(V, distance(x,y));
}

public int distance(Legal_Branch x,
                    Legal_Branch y){
   if( equivalent(x, y))
      return 0;

      int dist_gen = generalization(x, y, 1);
      int dist_spec = specialization(x, y, 1);

      if(dist_gen == 0 && dist_spec == 0){
      return Integer.MAX_VALUE;
   }

   return dist_gen > dist_spec
                   ? dist_gen : dist_spec;
}
```

4.2 Mapping from Agent Interaction to Communicative Acts

Figure 2 shows the Model of Agent Communicative Acts of Infonorma. This model describes how knowledge of the multiagent society is exchanged by the agents through messages sent from one to another. The messages from the Agent Interaction Model are mapped to FIPA-ACL messages.

First of all, when new information items are available, the Filter agent sends a *PROPOSE(normativeInstrumentType)* message to each *User Modeler* agent in context of the *ProposeSimilarityAnalysis* method. This message means that *Filter* agent is proposing to perform the similarity analysis of a normative instrument, which type is specified in the message, with each user model. The *User Modeler* agent evaluates the proposal and checks whether its respective user has

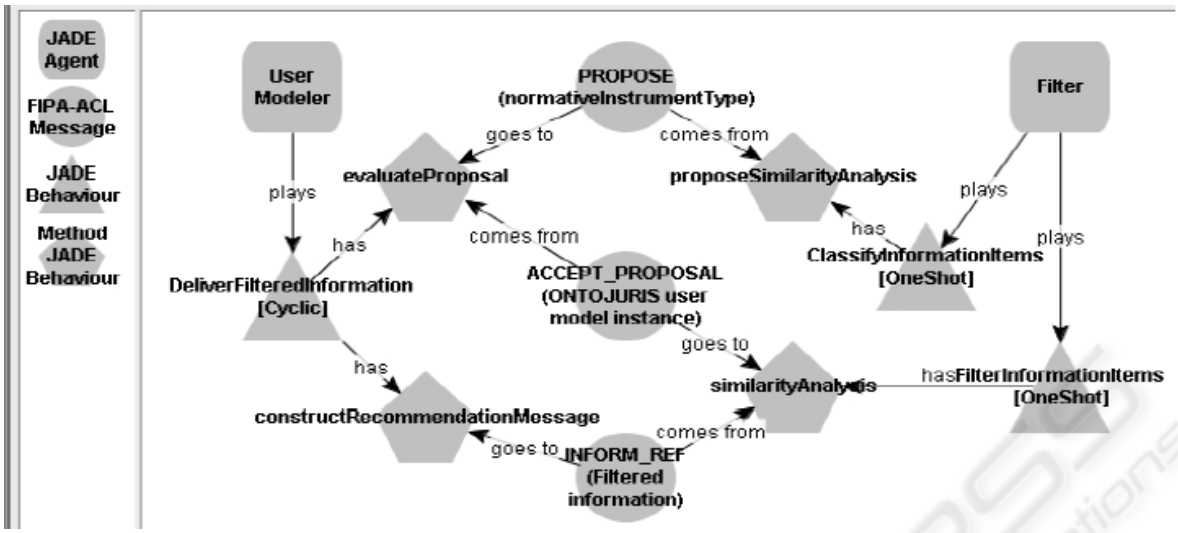

Figure 2: Model of Agent Communicative Acts of Infonorma.

interest in the type of the instrument through the *evaluateProposal* method. If so, it accepts the proposal sending an *ACCEPT_PROPOSAL(ONTOJURIS user model instance)* message which contains an instance of an ONTOJURIS user model. Otherwise the *User Modeler* agent just ignores the message.

Each *ACCEPT_PROPOSAL(ONTOJURIS user model instance)* message received by the Filter agent is answered with an *INFORM_REF(Filtered information)* message which content is a triple *<ONTOJURIS user model, normative instrument, similarity>* representing the result of similarity analysis, performed by the *similarityAnalysis* method, between the normative instrument which type was in the *PROPOSE(normativeInstrumentType)* message and user model specified in the answer to the first message. This way, the *User Modeler* agent is able to produce the personalized recommendations (*constructRecommendationMessage* method) and deliver them to the user.

The JADE platform provides a way to visualize the messages exchanged by the agents using the *Sniffer* agent.

5 RELATED WORK

Previous work on Infonorma has already been published. The specification of the system is discussed in (Drumond, Girardi and Leite, 2007) while its architectural design is introduced in (Drumond, Girardi and Leite, 2007b). Both the specification and design of the system were revised and updated and are presented with more details in (Drumond and Girardi, 2008), which also discusses the ONTOJURIS ontology and the detailed design of the system.

Work has been done on the usage of artificial intelligence techniques to improve the effectiveness of recommender systems. Some of these approaches use ontologies, like the one in (Middleton et al., 2002) and (Middleton et al., 2004) and machine learning, as in the PersoNews system (Banos et al., 2006).

The ROSA system (Girardi and Ibrahim, 1995) uses semantic cases to represent natural language queries and natural language descriptions of software components in an information retrieval system. Infonorma uses this approach adapted to information filtering. Besides that, ROSA represented semantic cases with frames while Infonorma considers the semantic cases as instances of an ontology.

There has been much work done in the domain of Artificial Intelligence and Law. Pinkwart et. alii (Pinkwart et al., 2006) developed a tutor system that uses collaborative filtering to identify weak points in students legal arguments. The development and usage of legal ontologies to represent and access legal information has been approached in (Kralingen, 1995), (Tiscornia, 2001), (Valente, 1995) and (Visser, 1995). Benjamins et. alii (Benjamins et al., 2005) introduce an overview of the application of Semantic Web technologies to the legal domain. A formal criteria for describing legal information on different levels is presented in

(Tiscornia, 2001) in the context of the Italian project Norme in rete (Law in the net).

6 CONCLUSIONS

This work discussed the implementation of the Infonorma multi-agent recommender system. This system is composed by two kinds of agents: the *User Interface* and the *Filter* agents which perform content based filtering and explicit user modeling for recommending legal information items.

Infonorma was developed under the guidelines of MAAEM, a methodology for the development of multi-agent systems, through the instantiation of ONTORMAS, an ontology-driven tool which guides the specification and design of multi-agent domain models and applications.

We are currently working on the evaluation of the effectiveness of the Infonorma system, as well as its extension with collaborative and hybrid filtering approaches and implicit profile acquisition. To this extent, a semantic portal where users can navigate through legal information and visualize the recommendations generated by Infonorma and other legal information systems is under development.

REFERENCES

Adomavicius G., Tuzhilin A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng. 17(6), 734--749 (2005)

Antoniou G., Van Harmelen F.: Semantic web primer, MIT Press (2004)

Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun ACM 40(3), pp. 66--72 (1997)

Banos, E., Katakis, I., Bassiliades, N., Tsoumakas, G., Vlahavas, I.: PersoNews: A Personalized News Reader Enhanced by Machine Learning and Semantic Filtering. In: Proc. 5th Int. Conf. on Ontologies, DataBases, and Applications of Semantics. LNCS, vol. 4275, pp. 975--982. Springer-Verlag, Montpellier, France (2006)

Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE a white paper. Exp vol. 3 n. 3, http:// www.jade.tilab.com/ (2003)

Benjamins, V., Casanovas, P., Breuker, J., Gangemi, A.: Law and the semantic web, an introduction. Lect Notes Comput Sci 3369, pp. 1--17 (2005)

Drumond, L., Girardi, R.: A Multi-agent Legal Recommender System. J. of AI and Law. 16, 175-207. Springer (2008)

Drumond, L., Girardi, R., Leite, A.: A Case Study on the Application of the MAAEM Methodology for the Specification Modeling of Recommender Systems in the Legal Domain. In Proceedings of the 9th International Conference on Enterprise Information Systems, pp. 155--160. INSTICC, Funchal, Portugal (2007)

Drumond, L., Girardi, R., Leite, A.: Architectural Design of a Multi-Agent Recommender System for the Legal Domain. In Proceedings of the 11th International Conference on Artificial Intelligence and Law, pp. 183--188. ACM Press, New York (2007b)

Girardi, R., Ibrahim, B.: Using English to Retrieve Software. J Syst Software, Special Issue on Software Reusability 30(3), pp. 249--270 (1995)

Kralingen, R.: Frame-based conceptual models of statute law, Computer/Law Series 16 (1995)

Girardi, R., Leite, A.: A knowledge-based tool for multi-agent domain engineering. Knowledge-Based Systems, vol. 21, n. 7, pp. 604—611 (2008)

Leite, A., Girardi, R.: ONTORMAS: Uma ferramenta dirigida por ontologias para a Engenharia de Domínio e de Aplicações Multiagente. The XI Iberoamerican Workshop on Requirements Engineering and Software Environment (2008)

Lindoso, A., Girardi, R.: The SRAMO technique for analysis and reuse of requirements in multi-agent application engineering. IX workshop on requirements engineering, Cadernos do IME, UERJ Press, 20, pp. 41--50. Rio de Janeiro (2006)

Middleton, S., Alani, H., Shadbolt, N., De Roure, D.: Exploiting synergy between ontologies and recommender systems. In Proceedings of the WWW International Workshop on the Semantic Web, vol. 55 of CEUR Workshop Proceedings, Maui, USA (2002)

Middleton, S., Shadbolt, N., De Roure, D.: Ontological User Profiling in Recommender Systems. ACM Transactions on Information Systems, 22, pp. 54--88 (2004)

Pinkwart, N., Aleven, V., Ashley, K., Lynch, C.: Using collaborative filtering in an intelligent tutoring system for legal argumentation. In: Weibelzahl, S., Cristea, A. (eds.) Proceedings of workshops held at the 4th international conference on adaptive hypermedia and adaptive web-based systems. Lecture notes in learning and teaching, Dublin, Ireland, pp. 542—551 (2006)

Shadbolt, N., Hall, W., Berners-Lee, T.: The semantic web revisited. Intelligent Syst 21(3), p. 96–101 (2006)

Tiscornia, D.: Ontology-driven access to legal information. DEXA 12th international workshop on database and expert systems applications, pp. 792 (2001)

Valente, A.: Legal knowledge engineering: a modelling approach. University of Amsterdam, the Netherlands, IOS Press, Amsterdam, The Netherlands (1995)

Visser, P.: Knowledge specification for multiple legal tasks; a case study of the interaction problem in the legal domain, computer/law series, n. 17, Kluwer Law International, The Hague, The Netherlands (1995)