ADVISORY AGENTS IN THE SEMANTIC WEB

Ralf Bruns, Jürgen Dunkel
Department of Computer Science, University of Applied Sciences and Arts Hannover,
Ricklinger Stadtweg 120, D-30459 Hannover, Germany

Sascha Ossowski
AI Group, E.S.C.E.T.,Universidad Rey Juan Carlos Madrid,
Campus de Mostoles, Calle Tulipan s/n, E-28933 Madrid, Spain

Keywords: Semantic Web Application, Ontology, Agents, E-learning.

Abstract: In this paper, we describe the advances of the Semantic E-learning Agent project, whose objective is to develop virtual student advisers that render support to university students in order to successfully organize und perform their studies. The advisory agents are developed with novel concepts of the Semantic Web and agent technology. The key concept is the semantic modeling of the domain knowledge by means of XML-based ontology languages such as OWL. Software agents apply ontological and domain knowledge in order to assist human users in their decision making processes. Agent technology enables the incorporation of personal confidential data with public accessible knowledge sources of the Semantic Web in the same inference process.

1 INTRODUCTION

E-learning has started to play a major role in the learning and teaching activities at institutions of higher education worldwide. The students perform significant parts of their study activities decentralized via the Internet. The main focus of current E-learning systems is to provide an appropriate technical infrastructure for content engineering and information exchange.

The emerged individual ways of study are location- and time-independent, consequently requiring a permanently available and direct support to answer questions and give advice. A recent comparison of modern E-learning environments (CCTT, 2004) revealed that intelligent advisory agents are not applied so far in E-learning systems.

The objective of the Semantic E-learning Agent (SEA) project (Dunkel, 2004) is to develop virtual student advisers that render support to university students, assisting them to successfully organize und perform their studies. The experiences of human course advisers show, that most students have similar problems and questions. The advisory agents should help to resolve these problems. Typical questions concern the regulations of study (e.g. does a student possess all requirements to participate in an examination or a course?) or organizing student mobility.

To achieve these goals, we propose a software architecture where virtual student advisers are developed with novel concepts from Semantic Web (Berners-Lee, 2001; Horrocks, 2002) and Intelligent Agent (Wooldrige, 1995) technology. The basic idea is to model the structure of our E-learning domain by means of ontologies, and to represent it by means of XML-based applied ontology languages. Software agents apply the knowledge represented in the ontologies during their intelligent decision making process. We claim that this is a promising approach because E-learning systems that successfully support students in organizing their studies are still to come. This paper reports on the experiences gained from the development of an advisory system that effectively integrates both, Semantic Web and Intelligent Agent technology.

The first use case that has been implemented reflects the counseling situation where a student intends to study a semester abroad within the European Erasmus/ Socrates exchange program. Together with the international coordinator, the student has to choose the foreign university and the

Bruns R., Dunkel J. and Ossowski S. (2005).
ADVISORY AGENTS IN THE SEMANTIC WEB.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 90-96
DOI: 10.5220/0002546500900096

foreign study program that matches best her/his personal interests and her/his individual situation of study. Subsequently, a study plan for the semester at the host university must be determined that corresponds to the home university syllabus. This study plan constitutes the so-called Socrates Learning Agreement.

The paper is structured as follows: In the next section the employed knowledge representation techniques and the developed knowledge models are presented. The third section shows how automated inference can be carried out on the knowledge models. Subsequently, the software architecture of the agent system is outlined. Finally, the last section summarizes the most significant features of the project and provides a brief outlook to future lines of research.

2 KNOWLEDGE MODELING

The key concept of a semantic advisory system is the semantic modeling of the domain knowledge (e.g. university organization, degree requirements, course descriptions, examination regulations) as well as an individual user model, which reflects the current situation of study (e.g. passed exams, current courses). The fundamental structures of the available domain knowledge as well as the basic facts (e.g. offered courses) are defined in appropriate models.

In our system, the structural part of the knowledge base is modeled by means of ontologies, which formally define domain entities and the relations among them. For this purpose, we apply Semantic Web technology based on XML. We have chosen the W3C standard ontology language OWL (Web Ontology Language) (W3C-OWL, 2004) to model the knowledge required in the advisory system. Software agents use this information as the basis for their reasoning and negotiation. Due to the standardization of these technologies, knowledge models can easily be shared and reused via the Internet. Thus, the developed ontologies can serve as standardized and open interfaces for the interoperability of different E-learning systems.

2.1 Ontologies

In order to implement the counseling situation of the Erasmus/ Socrates exchange program, information is necessary about the possible exchange universities and their offered degree programs. In addition, further information about the living conditions of a particular university city and its urban infrastructure may influence the decision.

Several interrelated ontologies have been developed for our advisory agents: Two central ontologies describe the organizational structure of a university and the offered courses in a semester. To facilitate the comparison of different study places and course contents, two subordinated ontologies are used. The individual study situation of a specific student is represented by a separate ontology.

Dividing the knowledge base of the advisory system in several different ontologies is crucial to yield a coherent scope of each ontology and to facilitate reusing existing ontologies (Noy, 2001). In the following, we describe the responsibilities of the employed ontologies in some more details.

- **University Ontology**
  The university ontology is the core knowledge base of the SEA project. It models the essential parts of the organizational structure of a particular university and the departments with the different programs of study. Its main domain concepts are: university, department, degree program, offered degrees.

  The following example shows an excerpt of an instance of the university ontology.

```
<uni:DegreeProgram
              rdf:ID="FHH_Master_CS">
 ...

 <uni:numberOfStudents rdf:datatype=
            "http://.../XMLSchema#int">
     547
 </uni:numberOfStudents>

 <uni:hasContent rdf:resource=
   "http://../subject.owl#softwareEng"/>

  <uni:hasContent rdf:resource=
     "http://../subject.owl#compGraph"/>

 ...
</uni:DegreeProgram>
```

  At first, a degree program instance with id `FHH_Master_CS` is created. The property `numberOfStudents` specifies how many students are enrolled and has the XML schema data type `int`. The property `hasContent` describes the contents of the degree program and refers to a computer science instance of the subject area ontology specified by the URI.

- **Course Ontology**
  The course ontology models the courses per semester for a degree program. This information changes from semester to semester and can only be provided by the responsible department. Several properties describe an individual course, e.g. course name, teaching language, number of credit points, keywords describing the course content, and the semester when the course takes place. This knowledge will be used in the

second step of our sample use case when open courses of the home syllabus are matched with courses at the exchange university.

Each university participating in the Socrates program should build its own instance of these ontologies. Additionally, for our counseling scenario we need further information that is provided by two additional ontologies.

- **Regional Ontology**
  The regional ontology models the relevant properties of a study place, e.g. in which country, state, and region it is located, number of inhabitants, which infrastructure is available (e.g. airport, station, theatre). Each study place is represented by an instance of this ontology, thus allowing a comparison due to the students living preferences. It is expected that for many cities this information will be available on the Semantic Web in the near future.
- **Subject Area Ontology**
  To find an appropriate study plan at the exchange university, home and foreign courses must be compared based on their contents. A simplified taxonomy is modeled in the subject area ontology, e.g. one instance for computer science, one instance for mechanical engineering, and so forth.

These two ontologies define some transitive properties that are used for inference and reduce the number of facts significantly. An example for transitivity is the property `isLocatedIn` of the regional ontology.

```
<owl:TransitiveProperty
                      rdf:ID="isLocatedIn">
    <rdfs:domain rdf:resource="#region"/>
    <rdfs:range rdf:resource="#region"/>
</owl:TransitiveProperty>
```

For example, from the two facts, that Hannover is located in Lower Saxony, and that Lower Saxony is located in Germany, it can be concluded that Hannover is located in Germany. In a similar way a hierarchy of subtopics is modelled in the subject area ontology.

In contrast to these ontologies, which model public accessible information, the user ontology serves as the knowledge model of a particular user, e.g. student or faculty member and, consequently, contains confidential information.

- **User Ontology**
  The major classes of this ontology are `Student` and `Faculty`. Relevant information of a student are, e.g. login name, student ID, current semester, passed/failed courses etc. Every student owns her/his own instance file of this ontology, reflecting her/his individual progress of study. This information allows the adviser to give a personalized advice considering the individual situation of a student.

Note that the different ontologies are not isolated, but related to each other. So, e.g. a student instance of the user ontology is related to a course instance of the university ontology via the property `isEnrolledIn`. Figure 1 shows the entire structure of the ontologies with the interrelating properties and some of their classes.

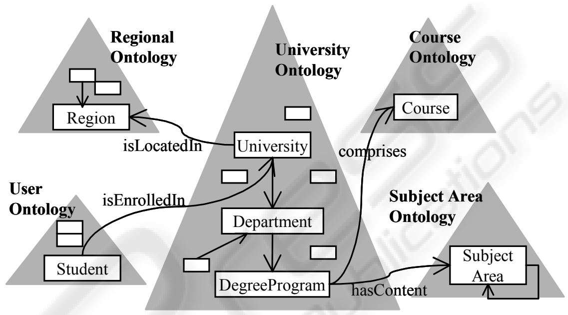

Figure 1: Sketch of ontology structure

In a Semantic Web infrastructure the knowledge is spread over the Internet in form of different OWL-files. We can distinguish two types: OWL schemas and OWL instances. In our advisory system there are five different OWL schema files, each containing just one of the described ontologies. To prevent inconsistencies, OWL schema files are located only once on a central web server.

However, the OWL instance files are created and maintained locally. It is crucial, that the OWL instances conform to the language specification defined in the OWL schemas and refer also to other instances.

2.2 Ontology Development

The previous section described the knowledge base, i.e. the ontologies and their corresponding facts, from a logical point of view. To make the knowledge usable for the advisory agents, internally or via the internet, they must be defined in a formal ontology language suitable for reasoning. For this purpose, we applied the W3C standard ontology language OWL (Web Ontology Language) (W3C-OWL, 2004) based on XML and RDF/ RDF Schema (W3C-RDF, 2004). The expressiveness of OWL-DL was sufficient to model our domain knowledge. Only a few shortcomings of OWL came up, which we resolved within the inference engine, as describe in the next section.

To develop complex ontologies an adequate tool support is indispensable. OWL is intended for the usage of software programs and cumbersome for humans, as the short OWL example in the previous section illustrates. In our project we used the well-known Protégé Version 2.1 with the OWL Plugin (Protégé, 2004) for ontology development. Except some smaller technical problems we made good experiences with this tool. It allowed to specify ontologies with a graphical user interface and to generate the corresponding OWL files, avoiding a potentially error prone "manual" OWL coding. Furthermore, facts in form of OWL instances were created on base of these ontologies.

3 INFERENCE

The semantic advisory agents should act similar to human advisers according to their knowledge modeled in the ontologies. This is achieved by using the rule-based inference engine JESS (Java Expert System Shell) (Friedman-Hill, 2004) to carry out the automated inferences entailed by the semantics of OWL. JESS provides a convenient way to integrate reasoning capabilities into Java programs. With the JESS language complex rules, facts and queries can be specified.

3.1 OWL Transformation

To make use of the knowledge modeled in an ontology, the OWL semantics must be mapped into facts and rules of an inference engine. Because JESS does not provide an interface to import an OWL ontology, we employed the tool OWL Engine to load OWL ontologies and OWL instances into a JESS knowledge base (OWL Engine, 2004), which provides an XSLT-based transformation process.

The OWL inference engine consists of three different parts. One file contains JESS rules describing the OWL meta model, i.e. the OWL built-in rules. Two XSLT stylesheets transform files with OWL schemata or with OWL instances into JESS assertions.

A major advantage of the XSLT stylesheets approach is that the stylesheets can be easily adjusted to individual requirements. In our project we expanded the transformation rules for the `owl:transitiveProperty` and the `owl:UnionOf` OWL constructs.

3.2 Ontology Reasoning

Mainly, the advisory agents reason on the basis of the OWL knowledge model loaded into the JESS knowledge base. For our sample use case, the semantic expressiveness of OWL is nearly sufficient. But to express more complex expert knowledge, e.g. complex examination regulations, domain-specific rules must be developed. Inference engines such as JESS provide their own languages to specify complex rules for developing rule-based systems. A simple example for a domain-specific rule, out of the scope of OWL, is a JESS rule that categories cities according to their size.

The data modeled in OWL is usually domain specific, but independent of a certain application. The OWL properties define rules, which represent the general structure of the knowledge. They are mainly data-oriented, usage-independent and applicable to different applications. Additional rules specified in an inference engine are process-oriented; they specify the reasoning capabilities of an advisory system and are tailored to a specific use case.

4 AGENT ARCHITECTURE

The software architecture of an advisory system should reflect the situation of a real counseling interview. In our use case, a student intends to study abroad for one semester and consults the international coordinator of the department to get advice. Together they first look for an appropriate exchange university and then for a study plan, which fits best with the course program at the home institute.

All students are characterized by their personal situation and intents; the international coordinators give their advice on base of a profound knowledge of the study regulations and the different exchange programs.

Multi-agent systems provide a software paradigm that fits well to the described situation (Woolridge, 2002). The advisory system can be viewed in terms of autonomous agents of two different types: student agents and international coordinator agents. These agents interact to find an exchange university and a suitable study plan. Multi-agent technology provides the right level of abstraction to model a negotiation process between independent partners (Jennings, 2000; Kraus, 1997) and, consequently, is well-suited for our purposes.

4.1 Agent Structure and Semantic Web

Figure 2 outlines the internal structure of the advisory system with two different types of agents: the student agent and the international coordinator agent. The two agent types are conceptually identical: both reason on a knowledge base using JESS as inference engine.

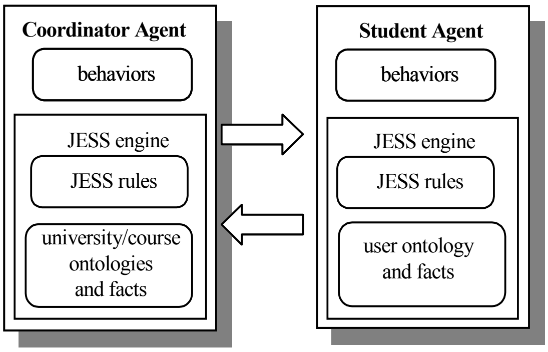

Figure 2: Agent structure

Each agent loads its individual knowledge base dynamically according to the actual counseling situation. As described above, the knowledge is specified in OWL-files and spread over the Internet. To build up its knowledge base each agent has to process the following steps:

1. According to the status of the interview, the agent determines the required information for the actual counseling context.
2. If the information is publicly available, the agent locates the corresponding OWL files in the Internet.
3. It downloads the OWL instances, transforms them and imports them into JESS using OWL Engine.

The international coordinator agent requires information about all exchange universities and their course contents. To gain this knowledge, it can dynamically expand its knowledge base by accessing the locally stored OWL-files of the universities registered in the advisory system. Beyond the information the coordinator agent collects in the Internet, it can hold some private knowledge. For example, it may know about all exchange agreements of its university or the utilization of the courses in its department.

The student agent is characterized by its individual study situation, which can be described by the study year, the attended lectures, and the passed exams. Of course, this information is confidential and, therefore, it is represented in a personal OWL instance file, which is protected against unauthorized access.

Figure 3 depicts the distribution of the knowledge sources in the Semantic Web. The coordinator agent and the student agent reside on different servers, where their private knowledge is stored in corresponding OWL instance files.

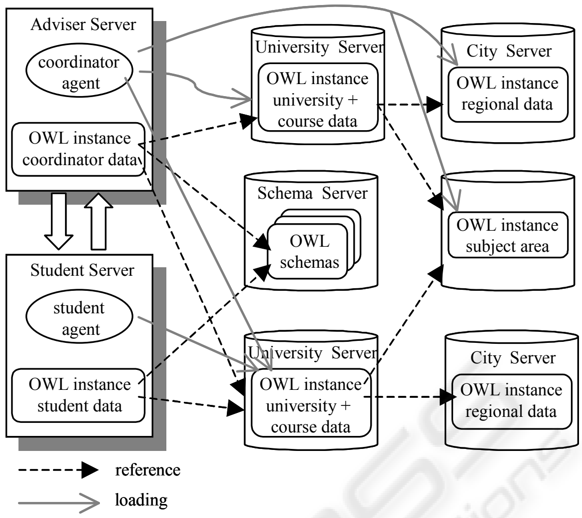

Figure 3: Distributed knowledge sources in the Semantic Web

Furthermore, each agent can have more sophisticated reasoning capabilities expressed by some further JESS rules, as explained in subsection 3.2.

4.2 Agent Interaction and Negotiation

In a real counseling situation a problem is resolved by a communication and negotiation process, which is characterized by an information exchange among the different dialog partners. In a multi-agent system the communication between the agents reflects this negotiation process between clients and advisers. Depending on how the consultancy is developing, different information is exchanged between the agents. The agent behaviors implement the negotiation protocol determining the rules that govern the interaction (Jennings, 2000; Ossowski, 2002). The following use case scenario outlines how the agents interact.

1. A student starts his/her personal student agent (SA) to search for a suitable exchange semester, and logs in.
2. The SA loads the OWL user ontology and the OWL instance data representing the students specific study situation into its JESS knowledge base.
3. The student can enter some preferences regarding the exchange university (e.g. the subject of study, the teaching language, desired location). The SA extracts the specified parameters and queries further personal data (e.g. the aimed degree) from the knowledge base. Then the SA sends a request to the international coordinator agent (ICA).

4. The ICA collects instance data about all universities registered in the system as well as about the study places and loads them in its knowledge base. During its initialization, the ICA has already loaded all ontology schema files.
5. Then the ICA reasons on the knowledge base, aggregates the results and sends a ranked list of appropriate foreign degree programs to the SA.
6. The SA receives the result and presents it to the student, who chooses his/her favorite exchange university and degree program. The student's decision and further user instance data are send to the ICA (e.g. the study program based on the passed exams).
7. The ICA accesses the OWL course instance data of the selected foreign degree program via the Internet, and loads it into its knowledge base. Usually the courses information is maintained in each exchange university separately. On the basis of the expanded knowledge base the ICA suggests the foreign courses that are fitting best to the study program of the home university, see figure 4.

Figure 4: Study plan proposal of the coordinator agent

8. The SA receives the results from the ICA and the student chooses manually the desired course plan out of the different suggested options. Finally, the SA generates a formal document, called Socrates Learning Agreement, determining the personalized exchange study plan.

The student agent protects the confidential information of its human owner. Similar to a real consultancy situation it only reveals sensitive private data, if it is indispensable for finding a solution. The knowledge of the student agent is rather restricted; it mainly knows the personal situation of its owner. The international coordinator agent has a much broader knowledge, which it dynamically expands in the Semantic Web.

In the current implementation the international coordinator agent has no own intentions, i.e. it leaves all decisions about the exchange program to the user agent, who delegates them to its human user. Of course, the behavior of both agents could implement personal desires and intentions. For example, the coordinator agent could present only a selection of possible exchange universities, depending on the exchange agreements or the number of applicants.

4.3 Implementation Issues

Powerful agent development frameworks facilitate the development of multi-agent systems. The semantic advisory agents are developed with JADE (Java Agent Development Framework) (Bellifemine, 2002), which complies with the FIPA (Foundation of Intelligent Physical Agents) standards (FIPA, 2003). JADE includes two main components: a FIPA-compliant agent platform and a framework to develop Java agents. The core part of the FIPA architecture is a standard for agent communication, i.e. its ACL (Agent Communication Language). The interaction between the student and the international coordinator agent is based on the exchange of ACL messages.

To avoid that a user has to install the student agent on her/his computer, we chose a web architecture: the user agent resides on a central server and has a web interface implemented with JavaServer Pages.

5 CONCLUSION

In this paper, we described how Semantic Web and Agent Technology can be integrated to build an intelligent advisory system for an E-learning environment. Our goal is to create and deploy semantic advisory agents capable of supporting university students in successfully organizing and performing their studies.

Due to the use of Semantic Web languages the developed knowledge models can easily be used in distributed systems and shared among software agents via the Internet. In dependence of the state of the consulting interview, agents acquire dynamically useful knowledge from distributed sources in the Semantic Web and integrate it in their personal knowledge base. Agent technology enables the incorporation of personal confidential data with publicly accessible knowledge sources of the Semantic Web.

The major difficulty encountered was the integration of the different concepts – on the one hand the knowledge bases written in RDF and OWL, on the other hand the inference engine JESS and the agent environment JADE. We implemented a prototype system, where the agents were able to reason upon the knowledge base in the desired manner. Our experiences show that the employed technologies are mature and well-suited for the implementation of advisory systems.

In our future work, we will implement more use cases for the Semantic E-learning Agent project. For example, advisers should be able to announce new opportunities for students who are looking for suitable thesis subjects and to answer questions regarding the regulations of study.

REFERENCES

Bellifemine, F, Giovanni, C., Trucco, T. and Rimassa, G., 2002. *JADE Programmers's Guide*. Retrieved August 25, 2004, from http://jade.tilab.com/doc/programmersguide.pdf

Berners-Lee, T., Hendler, J. and Lassila, O., 2001. The Semantic Web. *Scientific American*, Vol. 5.

CCTT - Center for Curriculum, Transfer and Technology, 2004, Retrieved August 25, 2004, from http://www.edutools.info/course/compare/all.jsp

Dunkel, J., Bruns, R. and Ossowski, S., 2004. Semantic E-Learning Agents. In *Proc. of the Sixth International Conference on Enterprise Information Systems* ICEIS 2004 (Porto, Portugal), 271-278.

FIPA - Foundation of Intelligent Physical Agents, 2003. Retrieved August 25, 2004, from http://www.fipa.org

Friedman-Hill, E., 2000. *JESS, The rule engine for the Java platform*. Retrieved August 25, 2004, from http://herzberg.ca.sandia.gov/jess/

Horrocks, I. and Hendler, J. (eds.), 2002. *The Semantic Web, First International Semantic Web Conference*. Springer.

Jennings, N.R., Parsons, S., Sierra, C. and Faratin, P., 2000. Automated Negotiation. In *Proc. 5th Int. Conf. On Practical Application and Intelligent Agents and Multi-Agent Systems* (Manchester, UK), 23-30.

Kraus, S., 1997. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2), 79-98.

Noy, N.F. and McGuiness, D.L., 2001. *Ontology Development 101, A Guide to Creating your first Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05.

Ossowski, S.; Omicini, A., 2002. Coordination Knowledge Engineering. *The Knowledge Engineering Review 17 (4)*, 309–316

OWL Inference Engine, Retrieved from http://mycampus.sadehlab. cs.cmu.edu/public_pages/OWLEngine.html

The Protégé Project. Retrieved August 25, 2004, from http://protege.stanford.edu/

Wooldridge, M. and Jennings, N., 1995. Intelligent Agents - Theory and Practice. *Knowledge Engineering Review* 10 (2), 115–152.

Woolridge, M., 2002. *An Introduction to Multiagent Systems*. Wiley.

W3C – The World Wide Web Consortium. *RDF Primer – W3C Recommendation 10 February 2004*. Retrieved August 25, 2004, from http://www.w3.org/TR/rdf-primer/

W3C – The World Wide Web Consortium. *OWL Web Ontology Language Reference – W3C Recommendation 10 February 2004.* Retrieved August 25, 2004, from http://www.w3.org/TR/owl-ref/