Diogene: a Semantic Web-Based Automatic Brokering System

Nicola Capuano[1,3], Matteo Gaeta[1,3], Alessandro Micarelli[4] and Enver Sangineto[2]

1 CRMPA Centro di Ricerca in Matematica Pura ed Applicata, c/o Università degli Studi di Salerno – DIIMA, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy.

2 CRMPA Centro di Ricerca in Matematica Pura ed Applicata, Sezione "Roma Tre", c/o Università degli Studi Roma Tre – DIA, Via della Vasca Navale, 79, 00146 Roma, Italy.

3 DIIMA Dipartimento di Ingegneria dell'Informazione e Matematica Applicata, Università degli Studi di Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy.

4 DIA Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre, Via della Vasca Navale, 79, 00146 Roma, Italy.

The aim of the European funded project Diogene (which is going to be closed the 31st of October 2004) [Diogene] has been the construction of an automatic "brokering environment" for e-learning in a Semantic Web scenario. The Diogene platform is able to act as an intermediary between the learners and different "content providers", specialized training organisations which provide learning material suitably indexed by means of common Semantic Web standards and following a knowledge representation paradigm developed during the project [Capuano 2002, Capuano 2003]. This proposed knowledge representation framework on one hand allows teachers (or "learning designers") to easily represent pedagogic information and on the other hand the system to efficiently reason about such information in order to assembly and personalize the learning material while answering the student's training requests.

The knowledge representation framework the system is based on is composed of three different sub-structures, respectively used for the representation of the students (by means of *Student Models*), the Learning Objects (by using a metadata standard suitably extended) and the pedagogic relationships among the topics of the didactic domain the system is specialized on. When the learner submits a *training query* (a set of topics she/he is interested in) to the system, Diogene generates a course by assembly the available Learning Objects and taking into account the didactic information among the topics of interest as well as the student's profile.

First of all, the system describes pedagogical information about the didactic domain it is specialized on (e.g., Computer Science, Math, History, and so on). To this aim, during the project we have developed a knowledge representation methodology for didactic knowledge description. It is based on an Ontology composed of a restricted set of relations among the topics of interest of the domain (each topic being represented by an Ontology's element, that we call *Domain Concept*). These relations allow the system to achieve a very good trade-off between knowledge representation expressiveness and reasoning computational efficiency. The main properties of the proposed knowledge representation are the following.

1. We can represent hierarchical decompositions and some important pedagogic relationships among the elements of the didactic domain.

2. The automatic deductions the system needs for the course generation have a computational cost which is linear in time and space with the size of the Ontology (number of Domain Concepts).

3. The knowledge representation framework is completely domain-independent and can be used to describe any specific didactic application.

The Ontology relations together with their semantics are described below.

- *HP* (*Has Part*): *HP* $(x, y_1, y_2, ..., y_n)$ means that the concept *x* is composed of the concepts $y_1, y_2, ..., y_n$, that is to say: to learn *x* it is necessary to learn y_1 *and* y_2*, and, ..., and* y_n.
- *R* (*Requires*): *R (x, y)* means that to learn *x* it is necessary to have already learnt *y*. This is a pre-requisite relation which also fixes a *partial order* on the Domain Concepts (DCs) learning sequence (let us call it *Learning Path*).
- *SO* (*Suggested Order*): *SO (x, y)* means that it is preferable to learn *x* and *y* in this order. Note that also this relation fixes a constraint on the DCs' order but in this case it is not necessary to learn *y* if we are interested only in *x*.

It is worth noting that the proposed knowledge representation paradigm is not intended to provide a general-purpose knowledge representation framework. For instance, there is no possibility to express subset relations, because each DC is an individual element (possibly decomposable in other elements) and there is no representation of set of elements or classes like in common semantic networks. The reason is that such simple knowledge representation rules are tailored to the Diogene's planning needs, while a universal treatment of Ontology representation (valid also for non-didactic purposes) was out of the scope of the project.

Even if the proposed didactic knowledge representation is completely domain-independent (none of the abovementioned relations refers to a specific application domain), during the project we have applied it to a specific subject, building an Ontology for the whole Computer Science. We started from a relatively restricted set of concepts taken from the ACM classification ("Computing Classification System" (CCS) [ACM]) and we built on this base lower detail levels (i.e., new sub-concepts linked by *HP* relations) and added didactic information (*R* and *SO* relations) [Stefanov 2003]. Finally, the resulting Ontology has been represented by *RDF Schemas*.

The Diogene's Ontology is used by the system to describe abstract didactic relations among a domain of interest. The concrete learning material explaining the topics of such a domain is provided by the system's Content Providers. Learning Objects (LOs) related to the DCs are Web deliverable objects (textual or hyper-textual documents, interactive exercises, simulations and others) each one explaining one or more DCs. LOs are indexed by means of Metadata, which allow the machine to "understand" and manipulate them. Each LO's Metadata is filled-in by a teacher (or "learner designer") belonging to one of the Content Providers' enterprises. We use the *IMS Metadata 1.2.2* standard which has been extended by including in the Educational category four more attributes. These attributes are used in order to describe the *teaching styles* of each LO, following a pedagogical categorization suggested by Felder in [Felder 88]. Finally, the "taxonpath" attribute of the IMS Metadata Classification category is used to link a LO with one or more DCs.

Starting from pre-existing digital courses and learning material, the Content Provider partners of the project have used the standard chosen (with the mentioned extensions) to index the LOs. Furthermore, they have also made different interactive exercises used by the system to assess the students on-line. The interactive tests' creation and the LOs' Metadata editing have been done by means of a suitable tool (called *KMS*) developed during the project, while for the graphical creation of an Ontology we used Protege [Protege]. Tests are indexed using the KMS tool and the *IMS QTI 1.2* standard.

Last but not least, the system traces the learner's profile by building a *Student Model*. This model represents the student's skills concerning every Ontology's Domain Concept together with the student's *Learning Preferences* such as the *learning styles* proposed by Felder. The representation of the student's skills (called *Cognitive State*) is updated by the system every time the student makes an interactive test. Also the Learning Preferences can change by using some statistical analysis on the successes/failures of the student. The Cognitive State is represented by means of a set of couples <*name*, *value*>, where *name* is an Ontology's DC and *value* is a fuzzy value describing the knowledge degree of the student concerning the topic *name*. We use the *IMS LIP 1.0* standard (suitably extended) to represent the Student Model.

The knowledge representation structures abovementioned (the Student Model, the Ontology and the LOs' Metadata) are used by the system to automatically generate and personalize courses. The "brokering" service offered by Diogene works as follows. The student inputs the system with a training query which specifies a set of Concepts (called the *Target Concepts*) of the Ontology which the student is interested in. Using the Ontology and the student's Cognitive State the system can automatically generate a course by composing a Learning Path containing all those Concepts needed to learn the Target Concepts. The Learning Path generation is easily performed by the system by looking at the Ontology and including in the Learning Path all those sub-concepts and pre-requisite concepts linked (by means of the *HP* and *R* relations) to the Target Concepts and not already sufficiently known by the learner (we can check this fact using the Student Model). This operation is linear in time and space occupation with respect to the number of Ontology's DCs. The Learning Path is also *totally ordered* by following the Ontology ordering constraints (the *R* and *SO* relations). Also in this case, the operation of making a total order from a partially ordered set has a well-known linear computational cost.

Finally, for each DC of the Learning Path Diogene chooses a LO taking into account the student's Learning Preferences which are matched with the LOs' Metadata (e.g., we match the teaching styles of a given LO with the learning styles of the student). The collection of LOs so obtained is called *Presentation* which is finally delivered to the student.

The scientific work presented in this paper is the result of the European Community founded project Diogene (5th Framework Programme, Information Society Technologies, contact number IST-2001-33358, see [Diogene] for more details), which will finish in October 2004.

References

[ACM 2004] ACM Web Site. http://www.acm.org/class/.

[Capuano 2002] Capuano N., Gaeta M., Micarelli A. and Sangineto E. An integrated Architecture for Automatic Course Generation. *Proceedings of the IEEE International Conference on Advanced Learning Technologies* (*ICALT 02*), Kazan, Russia, 2002.

[Capuano 2003] Capuano N., Gaeta M., Micarelli A. and Sangineto E., An Intelligent Web Teacher System for Learning Personalization and Semantic Web Compatibility, *Proceedings of the Eleventh International PEG Conference*, St. Petersburg, Russia, 28 June-1 July 2003.

[Diogene] Diogene: a Training Web Broker for ICT Professionals (IST-2001-33358): http://www.diogene.org.

[Felder 88] Felder R. M., Learning and Teaching Styles in Engineering Education, *Engr. Education* 78 (7), 674-681, 1988.

[Protege] Protege Web Site. http://protege.stanford.edu

[Stefanov 2003] Stefanov K., Todorova K., "Computing Ontology Creation", International Congress MAASEE'2003, Borovets, Bulgaria, September 15-21, 2003.