Metadata Aware Peer-to-Peer Agents for the e-Learner

Fotios Kokkoras and Ioannis Vlahavas

Abstract-- Metadata, being the first building block of the emerging semantic web, will enable computers to understand what the accessed information is all about, allowing in that way the building of advanced web services. The e-Learning domain is one of the first that is benefited by the definition, among others, of the Learning Object Metadata (LOM). From another perspective, the distributed nature of the web suggests that agent technologies will play a key role towards the use of these metadata. In this paper, we detail a Conceptual Graph (CG) binding of LOM (CG/LOM) and present the eLPA, a knowledge based, client side, metadata aware, peer-to-peer agent, that relies solely on the CG knowledge representation formalism. eLPA serves primarily as a personal memory agent for the e-learner.

Index Terms-- e-Learning, metadata, conceptual graphs, agents

I. Introduction

In the last years, we have witnessed a huge, ongoing effort towards an information society in which the various software tools will be able to understand the information behind accessed data, rather than only moving them around and displaying them. One of the key elements in most of these efforts is the definition and use of various metadata i.e. data describing the accessed data. One of the most ambitious efforts on metadata definition is in the e-Learning domain. Educational metadata describe educational resources [13], the learner's profile [11], etc. The internal details of systems that utilize these metadata is an open issue since these efforts are primarily dealing with "what" and not "how". There are currently a lot of on-line portals offering educational material. Unfortunately, most of this material is not metadata rich, although the pace of metadata adoption is high in the last two years, thanks to the availability of a stable LOM specification [13].

Besides metadata, we have also witnessed an explosive spreading of peer-to-peer (p2p) networks on which resources/data are scattered across numerous peers. These peers can be queried for these resources using (in the best case) metadata that describe the resources. Although p2p networks work well with relatively small sets of metadata (a few fields) and simple queries, like in the case of p2p networks for accessing music files, querying educational resources described by complex metadata like LOM, is not a trivial case.

On the other hand, Conceptual Graphs (CGs) ([20], [3]), are a proven solution for modeling knowledge repositories [9] and e-services [19]. The CG formalism is not only capable of describing the various metadata semantics, but is expressive enough to allow the representation of the various levels of abstraction required to model different application domains [8]. Furthermore, the CG formalism provides also the operations required to create a functional reasoning system. This allows the creation of dynamic ontologies, where static and axiomatic/rule knowledge co-exist [4].

Combining CG and agent technologies have been also tackled recently, not only in the educational metadata context [17], but also in a more abstract approach where CG based agents function as knowledge servers [18] that interoperate. Outside the scope of CGs but under a knowledge based perspective, educational metadata have been also used in an Object Oriented Deductive Database environment [1].

In this paper, which elaborates ideas presented in [15], we detail a CG binding of LOM and present the eLPA, a knowledge based, client side, metadata aware, peer-to-peer software agent, that is built upon the CG knowledge representation and reasoning formalism. The eLPA agent acts on the e-learner's behalf, monitoring him/her as he/she accesses LOM compliant educational resources and providing him/her with on-demand advanced functionality, based on LOM metadata as well as on predefined axiomatic/rule knowledge. The agent, stores the LOM metadata of the accessed learning objects in a proprietary, CG based representation (CG/LOM) and serves as a memory assistant not only over the individually accessed educational material, but also on the educational resources accessed by other users that they also use such a peer agent.

The rest of the paper is organized as follows: Section II gives some background information regarding LOM and CGs. Section III describes issues related to the CG-LOM binding used in the eLPA agent, while in Section IV, the architecture of the eLPA agent and its p2p networking abilities are presented and sample cases of the agent's functionality are given. In Section V, a comparison of our work to related one is presented, and finally, Section VI concludes the paper and gives some insight on future work.

The authors are with the Department of Informatics (LPIS Group), Aristotle University of Thessaloniki, Greece. E-mail: {kokkoras, vlahavas} @csd.auth.gr

II. An Overview Of LOM And CGs

In this section we give a short overview of the two main technologies used in the eLPA agent. First we briefly present the LOM specification and then we outline the main features of the CG formalism.

A. The LOM Specification

The IEEE/IMS LOM specification ([10], [13]) is dealing with the attributes required to adequately describe a learning object, that is, any digital or non-digital means, which can be used during technology-supported learning. Such attributes include, without limited, type of object, author, owner, terms of distribution, format, requirements to operate, teaching or interaction style, prerequisites, etc.

```
<lom>
    <general>
        <identifier>1234</identifier>
    </general>
    <technical>
        <location type=URL>http://www.mysimulation.gr</location>
    </technical>
    <educational>
        <learningresourcetype>
            <source>
                <langstring xml:lang="x-none">LOMv1.0</langstring>
            </source>
            <value>
                <langstring xml:lang="x-none">Simulation</langstring>
            </value>
        </learningresourcetype>
        <context>
            <source>
                <langstring xml:lang="x-none">LOMv1.0</langstring>
            </source>
            <value>
                <langstring xml:lang="x-none">Secondary Educa-
tion</langstring>
            </value>
        </context>
        <typicallearningtime>
            <datetime>00:45:00</datetime>
        </typicallearningtime>
        <language>en</language>
    </educational>
    <relation>
        <kind>
            <source>
                <langstring xml:lang="x-none">LOMv1.0</langstring>
            </source>
            <value>
                <langstring xml:lang="x-none">Requires</langstring>
            </value>
        </kind>
        <resource>  <identifier>456</identifier>  </resource>
    </relation>
    <classification>
        <keyword>
            <langstring xml:lang="en">Kepler's Law</langstring>
        </keyword>
        <keyword>
            <langstring xml:lang="en">simulation</langstring>
        </keyword>
    </ classification>
</lom>
```

Figure 1. A LOM record (partial) in XML.

There is a total of nine categories of information inside a LOM record. These are: General, Life Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation and Classification. Since no information is included in these standards on how to represent metadata in a machine-readable format, the IMS Global Learning Consortium has developed a representation of LOM in XML [14]. A partial LOM record in XML binding is presented in Figure 1 (some details have been removed to simplify the example).

The XML/LOM, is actually a data binding aiming primarily at efficient data exchange between LOM compliant systems. This means that the LOM specification itself has no semantics, although there are do exist intended semantics in the information model specification [13]. As a result, IMS has defined an XML schema specification and a semantic rich RDF binding of LOM [14], which makes it possible to have more automation over LOM records.

B. Conceptual Graphs Overview

The elements of CG theory ([20], [3]) are *concept-types*, *concepts*, *relation-types* and *relations*. Concept-types represent classes of entity, attribute, state and event. Concept-types can be merged in a lattice whose partial ordering relation < can be interpreted as a categorical generalization relation. A concept is an instantiation of a concept-type and is usually denoted by a concept-type label inside a box (Figure 2). To refer to specific individuals, a referent field is added to the concept ([book:*] - a book, [book:{*}@3] - three books, etc). Relations are instantiations of relation-types and show the relation between concepts. They are usually denoted as a relation label inside a circle (Figure 2). A relation type determines the number of arcs allowed on the relation as well as the type of the concepts (or their subtypes) linked on these arcs.

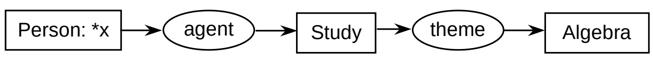

Figure 2. A Conceptual Graph stating that "there is some person *x* studying Algebra".

A Conceptual Graph is a finite, connected, bipartite graph consisting of concept and relation nodes (Figure 2). Each relation is linked only to its requisite number of concepts and each concept to zero or more relations. CGs represent information about typical objects or classes of objects in the world and can be used to define new concepts in terms of old ones.

The type hierarchy established for both concepts and relations is based on the intuition that some types subsume other types, for example, every instance of the concept *Lecture* would also have all the properties of *Teaching_Activity*. In addition, with a number of defined operations on CGs (canonical formation rules) one can derive allowable CGs from other CGs. These rules enforce constraints on meaningfulness; they do not allow nonsensical graphs to be created from meaningful ones. Among other operations defined over CGs, the most powerful is Projection. *Projection* is a complex operation that projects a CG v over another CG u which is a specialization of v ($u \leq v$), that is, there is a sub graph u' embedded in u that represents the original v. The result is one or more CGs πv which are similar to v but some of its concepts is possible to have been specialized by either specializing the concept type or assigning a value to some generic referent, or both.

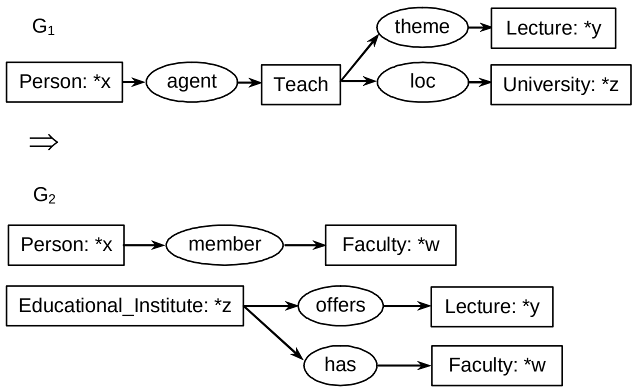

Figure 3. An example of a CG Rule.

Inference rules based on CGs have also been defined. A rule $R{:}G_1 \Rightarrow G_2$ is composed of two parts, G_1 and G_2, which are called *hypothesis* and *conclusion*, respectively (Figure 3). There may be coreference links between concepts belonging to G_1 and G_2. These are called connection points and must be of the same type. In more complex and useful situations it is possible to have more CGs in either part of the rule, joined with the logical operators. Furthermore, coreference links might exist between concepts belonging to either part of a rule. For example, the CG-rule in Figure 3 states that "if person *x* teaches lecture *y* in university *z*, then this person *x* is a member of some faculty *w* and there is an educational institute *z* that offers lecture *y* and its faculty is *w*. Notice how it is possible (using a concept type hierarchy) to relate the concept *university* in the hypothesis part with the concept *educational institute* in the conclusion part.

III. A CG BASED BINDING FOR LOM.

There are currently two bindings for LOM [14]. The first one, which is the XML binding (XML/LOM), creates a hierarchy of elements that best matches the structure of the LOM data model, that is, it is the result of choosing the most convenient syntax. Although this bunch of syntactic placeholders is adequate for the interchange of LOM records between applications, it lacks semantics that can be processed by a machine. Of course, a human reader is able to more or less understand the meaning of most of the elements, particularly if he/she is aware of the LOM information model [13]. In other words, the XML/LOM has intended semantics.

By contrast, the RDF binding of LOM (RDF/LOM) [14], which is currently elaborated, has semantic consequences because RDF is highly object oriented with objects having properties that relate them to other objects. Furthermore, the work done so far on the RDF/LOM binding, revealed that there are several possibilities for represented a LOM element, but each of them has fundamentally different semantics. Thus, while the XML/LOM describes the structure of a complete document instance, the RDF/LOM describes the structure of single metadata statements.

```
concept type LO
concept type URL < InternetAddress
relation type location(LO, InternetAddress)
[LO:123] → (location) → [URL:http://www.mysimulation.gr]

concept type URL < InternetAddress
relation type location(LO, InternetAddress)
[LO:123] → (location) → [URL: http://www.mysimulation.gr]

concept type Simulation < LearningResourceType
relation type learningResourceType(LO, LearningResourceType)
[LO:123] → (learningResourceType) → [Simulation]

concept type SecondaryEducation < LearningContext
relation type learningContext(LO, LearningContext)
[LO:123] → (learningContext) → [SecondaryEducation]

concept type DateTime
relation type typicalLearningTime(LO, DateTime)
[LO:123] → typicalLearningTime → [DateTime: 0000-00-00T00:45:00]

concept type Language
relation type educationalLang(LO, Language)
[LO:123] → (educationalLang) → [Language: en]

relation type requires(LO, LO)
[LO:123] → (requires) → [LO:456]

concept type Literal
relation type keyword(LO, Literal)
[LO:123] → (keyword) → [Literal: %en simulation]
[LO:123] → (keyword) → [Literal: %en "Kepler's Low"]
```

Figure 4. CG/LOM binding of the LOM record of Figure 1

Our main reason for deciding to create a CG/LOM binding, was the adequacy of the CG formalism to express the intended LOM semantics and, most importantly, to provide a "ready to use" framework for reasoning over these metadata. The later is not the case for RDF/LOM at the moment, since additional technologies are required for this, most of which are currently under development. It is widely acceptable that there is too much overlap between CGs and RDF/RDFS in terms of semantic expressiveness [2]. However, the expressivity of RDFS appears too much limited compared to CGs to represent axiomatic knowledge (concept definitions, domain axioms etc.) And the need for inference rules is crucial for discovering implicit knowledge on the web [6]. The CG formalism offers a unified and simple representation formalism that covers a wide range of other data and knowledge modeling formalisms and allows matching, transformation, unification and inference operators to process the knowledge that it describes [8].

A CG/LOM binding for the XML/LOM record of Figure 1, is presented in Figure 4 together with the proper concept and relation type definitions. Notice how similar structures in the XML version, like <*relation*> and <*keyword*>, are treated differently in CG/LOM. By making *simulation* a concept type and not an instance of type *LearningResourceType*, it is possible to perform additional operations, like replacing *[Simulation]* with its concept type definition (concept type expansion). Additionally, knowledge defined for *LearningResourceType* is also applicable to *Simulation* through generalization of concept *Simulation* to *LearningResourceType.*

The main advantage of encoding educational metadata in CGs, is the relevance of the CG projection operation to querying the XML data. Furthermore, the graph matching operation can be parameterised. In this way, it is possible to control the degree to which two concepts (or relations) are considered similar.

IV. AGENT AND COMMUNITY ARCHITECTURE AND FUNCTIONALITY

Since the CG formalism provides both a definitional and an execution model, it allows us to create a functional reasoning system. Having a CG/LOM binding at the same time, makes it possible to transparently combine LOM in a CG-based knowledge based system.

Figure 5 gives an outline of the architecture of our proposed, LOM aware software agent, namely the eLPA agent (e-Lerner's Personal Assistant). Its main component is the CoGITaNT engine [21], a library of C++ classes allowing the development of applications based on the CGs. CoGITaNT allows the handling of CGs using an object oriented approach and offers a great number of functionalities on them such as creation, modification, projection, definition of rules, inputs/outputs, etc. Furthermore, CoGITaNT can be extended since it provides the programming interface to define new operations, like for example, customized concept and relation matching operations and rule execution methods.

Additionally, the eLPA agent has networking abilities, since it can receive SOAP messages from other eLPA peer agents, through its SOAP Dispatcher module. As soon as such a message arrives, the SOAP Invoker component interprets the message, identifies the invocable interface this message calls, executes the call (this is actually done in the kernel), and assembles the response message.

Our agent includes a web browser ActiveX component which is constantly monitored by the agent through the Web Browser Connector module. This is the browser the user interacts with, in order to have the functionality offered by the agent. This embedded browser was clearly a design choice to simplify the prototyping, since, technically, it is possible to do the same with an external, stand alone browser.

The knowledge base (KB) of the eLPA agent consists of four parts. The first one is a *CG/LOM Repository* created from the various LOM records that accompanies any LOM compliant learning object accessed by the user. This is an automated task that transforms the XML expressed LOM records into the CG/LOM binding. The *Domain KB* includes all the relation and concept type definitions together with their partial hierarchies. In addition, it includes certain rules that control the interpretation of the CG/LOM metadata. For example, while a typical LOM record describes the difficulty of a learning object with a number in the scale of 1 to 5 ("very easy" to "very difficult", respectively), the CG-LOM allows the derivation of an internal, user adapted, difficulty factor that can take into account other metadata, like the age of the learner or his/her familiarization to other prerequisite learning objects. This enables the agent to have a personalized view of the accessed educational resources. All the information that enables personalization is kept into the *User Info KB*. Note however that this information is not compliant to the Learner Information Package (LIP) [11]. It is mainly static information regarding the user's identity, his/her kind of hardware, etc.

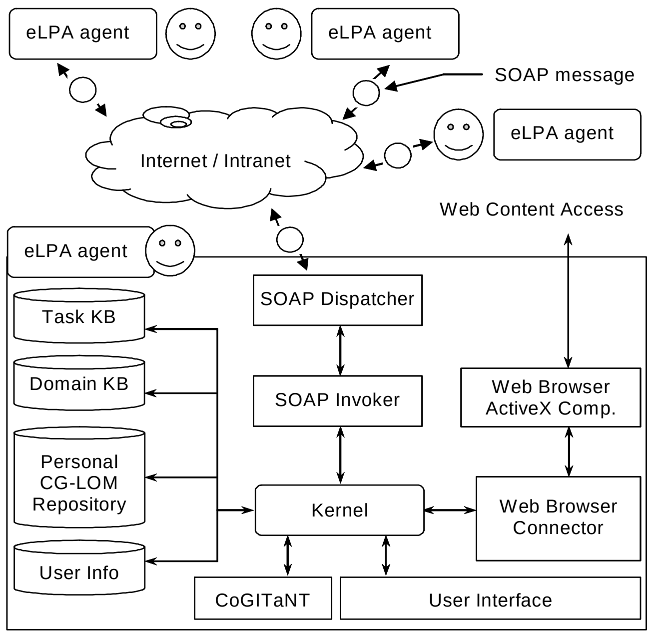

Figure 5: Architecture of the eLPA agent.

Other knowledge possibly included, is the definition of courses. A *course* is described as a graph of related, simple and autonomous tutorial items that someone who wants to study this course, should study.

Finally, there exists the *Task KB* (Figure 5), which materializes the services the agent is able to offer to the user and its internal operations. Such services include, without limitation, the generation of dynamic hyperlinks for a user to traverse and signaling of communication start between an agent with other agents. Two such rules are presented in Figure 6. Rule R_1 augments a learning object *x* that has as prerequisite the learning object *y* physically located at internet address *z*, with a dynamic hyperlink to that address. Rule R_2 is similar to R_1 in the first part but it creates a system event that forces the agent to communicate with known peer agents, instead. Notice how the rule becomes generic by using the concept type *InternetAddress* instead of *URL* (or *URI*) which are the concept types actually used in the CG/LOM repository. Upon evaluation of the rule the *InternetAddress* concept will be specialized to either *URL* or *URI*. Such operations are fixed and available to the user inside the web browser component, simply by right-clicking the mouse. More over, it is possible to ask for related learning objects for any selected word/term inside the web browser. Upon request, the system tries to semantically relate the selected word(s) with "known" learning objects from the personal CG/LOM repository. On successful much, dynamic links are created (as entries in a pop-up menu) to the physical location where these learning resources lie.

```
R1
[If: (requires [LO: *x] [LO: *y]) (location ?y [InternetAddress: *z])

[Then: (d_link ?x ?z) ]
```

```
R2
[If: (requires [LO: *x] [LO: *y]) (location ?y [InternetAddress: *z])

[Then: (q_params [SYS_EVENT: p2pquery]
                 [QCG: (location [LO: ?y] [InternetAddress: ?z]) ]) ]
```

Figure 6: CG rules of the agent's system knowledge base

Regarding networking, the agent is currently able to communicate with known peer agents that belong to different users. That is, there is no central directory service where an agent can seek peer agents. The user simply asserts into its own agent the address of "known" peer agents. The communication is established through the SOAP protocol, as already mentioned.

In such p2p situations, there are certainly various interoperability issues. The agents should somehow have a similar view of the world in order to avoid acquiring knowledge that could not be true in the context of their own KB. In eLPA, this is established with the agents having the same cannon, that is, concept and relation type definitions, type hierarchies, as well as other semantic constraints, are all the same. Based on this same KB core, the only thing that differentiates two individual agents is their personal CG/LOM repository. As a result, the only knowledge exchanged between peers is only CG/LOM assertions and queries that apply solely to CG/LOM data.

V. RELATED WORK

Although there is substantial work on using CGs towards the semantic web, to our current knowledge, the combination of educational metadata with CGs in a p2p agent framework, is a novel approach. Technically wise, the work presented in this paper is very close to our past work presented in [16] where a model (CG-PerLS) of a web portal for learning objects is described.

Corby et al [5] describe Corese, a conceptual resource search engine. It enables the processing of RDF Schemas and RDF statements within the CG formalism. Based on the functionality, it is very related to and goes further of our work, although is not dealing with educational metadata and the related domain knowledge. Since Corese uses the modern RDF/RDFS statements as an information source and there is also an evolving RDF binding for LOM, it will be very interesting to examine the CG-RDF/LOM that Corese will be able to produce.

Mineau in [18], sets some theoretical foundations on CG based agents that interoperate over the web and function as Knowledge Servers. Extending the p2p abilities of our agent and moving into more complex cooperation, will benefited from this work.

Finally, one of the most promising efforts on accessing learning resources over p2p networks, is the international project Edutella [22]. Edutella aims to produce an open-source, standards-based p2p architecture for the exchange of RDF-based meta-data. The demonstration domain is educational resources. This architecture will include services such as advanced distributed queries, semantic mappings between schemas, replication of meta-data, distributed annotation etc.

VI. CONCLUSIONS

In this paper, we have presented a CG binding for LOM (CG/LOM) and the eLPA, a CG-based assistant agent that serves its user as a personal memory for e-Learning content. The agent is able to understand and handle educational metadata describing learning objects. These metadata are incorporated into the agent's KB automatically as the user accesses learning objects and are kept internally in the proprietary CG/LOM format. In addition, the agent uses some learner related information to personalize its services to the user. Finally, we detailed the ability of several eLPA agents to co-exist in a peer-to-peer fashion and communicate between each other to fulfill their user's requests.

We are continuously improving the eLPA agent, both the internal reasoning part and most importantly, the p2p cooperation part. Much more work is required to establish cooperation with not only CG/LOM data exchange, but CG rule as well. In addition we incorporate more user related information by partially utilizing LIP metadata.

VII. REFERENCES

[1] Bassiliades, N., Kokkoras, F., Vlahavas, I., Sampson, D.: An Intelligent Educational Metadata Repository. Intelligent Systems, Techniques and Applications, C.T. Leondes (ed.), CRC Press, 2003.

[2] Berners-Lee, T. Conceptual Graphs and the Semantic Web. http://www.w3.org/DesignIssues/CG.html
[3] Conceptual Graphs Standard Working Draft, http://www.jfsowa.com/cg/cgstand.htm
[4] Corbett, D., A Method for Reasoning with Ontologies Represented as Conceptual Graphs. In: M. Brooks, D. Corbett and M. Stumptner (Eds.): AI 2001, Springer Verlag, LNAI 2256, pp, 130-141, 2001.
[5] Corby, O., Dieng, R. and Hebert, C.: A Conceptual Graph Model for W3C Resource Description Framework. Proceedings of the ICCS 2000, Springer Verlag , LNAI 1867, 2000.
[6] Delteil, A., Faron-Zucker, C. and Dieng R. Extension of RDFS based on the CGs Formalisms. Proceedings of the ICCS 2001, LNAI 2120, Springer Verlag, pp. 275-289, 2001.
[7] Genest, D., and Salvat, E., A Platform Allowing Typed Nested Graphs: How CoGITo Became CoGITaNT. In: Proceedings of the 6th International Conference on Conceptual Structures, Montpellier, France, Springer Verlag, LNAI 1453, pp. 154-161, 1998.
[8] Gerbe, O. and Mineau, G.W.: The CG Formalism as an Ontolingua for Web-Oriented Representation Languages. Proceedings of the ICCS 2002, Springer Verlag, LNAI 2392, pp. 205-219, 2002.
[9] Gerbe, O.: Conceptual Graphs for Corporate Knowledge Repositories. In: Proceedings of the ICCS '97, Springer Verlag, 1997.
[10] IEEE P1484.12/D6.4, Learning Object Metadata Standard, http://ltcs.ieee.org/doc/wg12
[11] IMS Global Learning Consortium Inc., IMS Learner Information Package Information Model, Version 1.0, 9 March 2001, http://www.imsglobal.org
[12] IMS Global Learning Consortium Inc., IMS Learner Information Package XML Binding, Version 1.0, http://www.imsglobal.org, 9 March 2001.
[13] IMS Global Learning Consortium Inc., IMS Learning Resource Meta-Data Information Model, Version 1.2.1, 28 September 2001, http://www.imsglobal.org
[14] IMS Global Learning Consortium Inc., IMS Learning Resource Meta-Data XML Binding, Version 1.2.1, http://www.imsglobal.org, 28 September 2001.
[15] Kokkoras, F., and Vlahavas, I., eLPA: An e-Learner's Personal Assistant, ICCS'02 Workshop on Application of Conceptual Structures, Bulgaria, http://lml.bas.bg/iccs2002/acs/kokkoras.pdf, 2002.
[16] Kokkoras, F., Sampson, D. and Vlahavas, I.: A Knowledge Based Approach on Educational Metadata Use. Advances in Informatics - Post-proceedings of the 8th Panhellenic Conference in Informatics, Y. Manolopoulos, S.Evripidou, A.Kakas (eds.), Springer Verlag, LNCS 2563, 2003.
[17] Kokkoras, F., Sampson, D., Vlahavas, I.: CG-PerLS: Conceptual Graphs for Personalized Learning Systems. In: Proceedings of the 8th Panhellenic Conference in Informatics, Vol.2, Livanis Press, pp. 531-540, 2001.
[18] Mineau, G.W.: A First Step toward the Knowledge Web: Interoperability Issues among Conceptual Graph based Software Agents - Part I. Proceedings of the ICCS 2002, Springer Verlag, LNAI 2392, pp. 205-219, 2002.
[19] Puder, A., Markwitz, S., Gudermann, F.: Service Trading Using Conceptual Graphs. In: Proceedings of the ICCS '95, LNAI 954, Springer Verlag, 1995.
[20] Sowa, J., Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley Publishing Company, 1984.
[21] The CoGITaNT library: http://cogitant.sourceforge.net
[22] The Edutella Project: http://edutella.jxta.org