CG-PerLS: Conceptual Graphs for Personalized Learning Systems

F. Kokkoras[1*], D.G. Sampson[2] and I. Vlahavas[1]

[1] Department of Informatics, Aristotle University, 54006, Thessaloniki, GREECE
email: {kokkoras,vlahavas}@csd.auth.gr

[2] Informatics and Telematics Institute, Center for Research and Technology - Hellas
1 Kyvernidou str, Thessaloniki, 54639, GREECE
email: sampson@ath.forthnet.gr

Abstract: Two of the most important standardization efforts for e-learning technologies are related to the definition of metadata describing educational resources and metadata describing the learner's profile. The internal details of systems that utilize these metadata is still an open issue since these efforts are primarily dealing with "what" and not "how". Under the light of these emerging efforts, we present CG-PerLS, a knowledge based approach for organizing and accessing educational resources. CG-PerLS is a model of a WWW portal for learning objects that encodes the learning technologies metadata in the Conceptual Graph knowledge representation formalism, and uses related inference techniques to provide advanced, personalized functionality. CG-PerLS allows learning resource creators to manifest their material, client-side learners to access these resources in a way tailored to their individual profile and educational needs, and dynamic course generation based on fine or coarse grained educational resources.

1 Introduction

As the World Wide Web matures, an initial vision of using it as a universal medium for educational resources is, day by day, becoming reality. There is already a large amount of instructional material on-line, most of it in the form of multimedia HTML documents, some of which are enriched with Java technologies. Unfortunately, most of these approaches have been built on general purpose standards and fail to utilize Web's potential for distributed educational resources that are easily located and interoperate with each other.

Information technology assisted education has reached sophisticated levels during 90's, taking into account issues like pedagogy, individual learner and interface, apart from the basic educational material organization. Following the recent "e-" trend, these approaches are just beginning to appear on the internet. The reason for this late adoption is mainly the substantial effort that is required to bring them on the Web since all of them have been designed without the Web in mind.

On the other hand, it is common place that our society has already moved away from the "once for life" educational model. The complexity and continuous evolution of modern enterprises' activities requires continuous training of their personnel. The networked community enables the management and enhancement of knowledge in a centralized - yet personal way, while keeping track and merging new intellectual resources into that process. The above requirements and advances lead us to the *"Lifelong Learning"* concept. The idea is to integrate the WWW technology with a novel, dynamic and adaptive educational model for continuous learning. The result will be a learning environment that will enable the individual learner to acquire knowledge

* The author was supported by a scholarship from the "Informatics and Telematics Institute".

just-in-time, anytime and anywhere, tailored to his/her personal learning needs. From the educational resources developer point of view, the standardization of all the education related aspects and the transition to more intelligent computer assisted education, will lead to autonomous, on-line educational resources that will be used by multiple tutorials and will operate independently of any single tutorial.

In this paper we present the CG-PerLS, a knowledge based approach on organizing and accessing educational resources. CG-PerLS is a model of a WWW portal for learning objects. CG-PerLS allows learning resource creators to manifest their educational material, client-side learners to access these educational resources in a way tailored to their individual profile and educational needs, and dynamic course generation based on either fine or coarse grained educational resources.

The remain of the paper is organized as follows: Section 2 describes the current trends in e-Learning. Section 3 outlines the basic features of Conceptual Graphs (CGs). Section 4 describes the CG-PerLS model and the functionality it provides. Section 5 outlines related work and, finally, section 6 concludes the paper.

2 Trends in e-Learning

The term e-Learning is used to describe a wide range of efforts to provide educational material on the web. These efforts include a diversity of approaches, ranging from static HTML pages with multimedia material to sophisticated interactive educational applications accessible on-line. The state-of-the-art in this latter category are the "Adaptive and Intelligent Web-based Educational Systems" (AIWES). As the term indicates, such approaches have their roots in the fields of Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia Systems (AHS). Actually, most of the current implementations are web-enabled adaptations of earlier stand alone systems. The main features of AIWES are [2]:

Adaptive Curriculum Sequencing: The material that will be presented to the learner is selected according to his learning request which is initially stated to the system. The sequencing refers to various levels of granularity; from which concept (topic, lesson) should be presented next to which is the next task (exercise, problem) the user should deal with. Either type of sequencing if performed based on the learners model, that is, the perception that the system has about the learner's current knowledge status and goals.

Problem Solving Support: We can identify three levels of support. In "*Intelligent analysis of learner's solution*" the system waits for the final solution and responds with the errors the student has made. In "*Interactive Problem Solving Support*" the system is continuously monitoring the learner and is capable of giving hints, signaling errors or even auto-executing the next step of an exercise. Finally, in "*Example based Problem Solving Support*" the system is able to suggest relevant problems that the user has successfully dealt with in the past.

Adaptive Presentation: This feature refers to the system's ability to adapt the content of the supplied curriculum to the learner's preferences.

Student Model Matching: This is a unique feature in AIWES. It allows the categorization of the learners to classes with similar educational characteristics and the use of this information for collaborative problem solving support and intelligent class monitoring.

None of the currently available e-learning systems delivers the advanced functionality described earlier. Most of the current e-learning approaches are limited to simple hyperlinks between content pages and "portal pages" which organize a set of related links. This happens because most of the existing educational material can be easily published on-line thanks to MIME data format standards. This is very important since educational material is expensive to create in terms of cost and time. This reusability however does not exist at the content, tutorial, or pedagogical levels.

The lack of widely adopted methods for searching the Web by content makes difficult for an instructor or learner to find educational material on the Web that addresses particular learning and pedagogical goals. In addition, the lack of standards prevent the interoperability of educational resources. Towards this direction and under the aegis of the IEEE Learning Technology Standards Committee (LTSC), several groups are developing technical standards, recommended practices and guides for software components, tools, technologies and design methods that facilitate the development, deployment, maintenance and interoperation of computer implementations of educational components and systems. Two of the most important LTSC groups are the Learning Object Metadata (LOM) group and the Learner Model (LM) group. The former is trying [7] to define the metadata required to adequately describe a learning object (LO) while the latter [10] deals with the specification of the syntax and semantics of attributes that will characterize a learner and his/her knowledge abilities.

One of the most ambitious efforts on e-Learning that make use of educational metadata is the Advanced Distributed Learning initiative [13]. Recently, ADL released the SCORM (Sharable Courseware Object Reference Model) that attempts to map existing learning models and practices so that common interfaces and data may be defined and standardized across courseware management systems and development tools.

All the above standardization efforts will require enough time to mature. More time will be required to build systems to conform to these specifications. The internal details of such systems is an open issue since these standardization efforts are primarily dealing with "what" and not "how". Meanwhile, everyday more and more educational material is becoming available. Therefore, there is an urgent need for methods to efficiently organize what is available today and what will become available in the near future, before the educational resource providers conform to the results of the standardization efforts.

In this paper, under the light of the emerging standards, we present the CG-PerLS, a knowledge based approach on organizing and accessing educational resources. CG-PerLS is a model of a WWW portal for learning objects. It transparently encodes LOM and LM metadata into CGs and uses related inference techniques to provide its functionality. CG-PerLS allows learning resource creators to manifest their educational material even if this material is not LOM aware, client-side learners to access these educational resources in a way tailored to their individual profile and educational needs, and dynamic course generation based on either fine or coarse grained educational resources.

3 Conceptual Graphs: Primitives and Definitions

The elements of CG theory [14] 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. 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. Each relation is constrained to which concepts it can connect. A CG is a finite, connected, bipartite graph consisting of concept and relation nodes. 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 CG model of knowledge representation is a practical way to express a large amount of pragmatic information through assertions. All of the algorithms defined on CGs are domain-independent and every semantic domain can be described through a purely declarative set of CGs. CGs have the same model-theoretic semantics with KIF (Knowledge Interchange Format) and are currently under a standardization process [4].

4 Conceptual Graphs for Personalized Learning Paths

This section describes CG-PerLS, a knowledge based approach (model) for manifesting and accessing educational resources over the Internet.

4.1 A CG binding of LOM and LM

The LOM and LM standards provide a semantic model for describing the properties of learning objects and learners, respectively. Any LOM/LM compliant system is free to internally handle these metadata in any way, but should be able to meaningfully interchange such metadata with other systems.

<RECORD>	
...	...
<CLASSIFICATION>	
<KEYWORD	
<LANGSTRING lang=en>Kepler's Law</LANGSTRING>	[LO:#123] → (format) → [KEYWORD: %en {" Kepler's Law ", "simulation"}]
</KEYWORD>	
< KEYWORD >	
<LANGSTRING lang=en>simulation</LANGSTRING>	
</ KEYWORD >	
</CLASSIFICATION >	
...	...
<EDUCATIONAL>	
<INTERACTIVITYTYPE>	[LO:#123] → (interactivity) → [INTERACTIVITY_TYPE: %en "Active"]
<LANGSTRING>Active</LANGSTRING>	
</INTERACTIVITYTYPE>	
...	
<INTERACTIVITYLEVEL>3</INTERACTIVITYLEVEL>	[LO:#123] → (interactivity) → [INTERACTIVITY_LEVEL: 3]
<SEMANTICDENSITY>2</SEMANTICDENSITY>	[LO:#123] → (educ_char) → [SEMANTIC_DENSITY: 3]
...	...
<TYPICALAGERANGE>	
<LANGSTRING>12-99</LANGSTRING>	[LO:#123]←(theme)←[USE]→(agent)→[Person]→(typ_age)→[AGE_RANGE:%en 16-99]
</TYPICALAGERANGE>	
<DIFFICULTY>2</DIFFICULTY>	[LO:#123] → (difficulty) → [DIFFICULTY_LEVEL: 2]
<TYPICALLEARNINGTIME>	
<DATETIME>0000-00-00T03:00</DATETIME>	[LO:#123] → (duration) → [DATETIME: 0000-00-00T03:00]
</TYPICALLEARNINGTIME>	
...	...
<LANGUAGE>en_US</LANGUAGE>	[LO:#123] → (human_lang) → [LANG: "en_US"]
</EDUCATIONAL>	
...	...
<RECORD>	

Table 1: LOM record (partial) in XML (left) and CG binding (right)

We selected to map these metadata to CGs. CGs define knowledge both at the type and instance levels. They are particularly well suited to model/organize/implement learning repositories at the knowledge level, since they support [5]:

- *Classification and partial knowledge*: Before gathering and storing information about things, we neither need to define all possible concepts that exist in the application domain, nor need to know all properties of things. Furthermore, we can define different concepts to classify things into (multiple classification), each one corresponding to a different perspective.

- *Category and/or instance in relationship:* It is possible to represent associations between categories and things.
- *Category or instance in metamodel*: Concepts give information about instances, but they may be also seen as instances in a metamodel that gives information about the concepts themselves. This is very important, since an element should allow to be viewed as a concept or an instance. This allows to define categories of categories, i.e. it is possible to integrate higher-order information in the same knowledge base. For example, "Java" can be defined both as a concept and an instance of the concept [programming_language].

We have defined CG templates to capture the semantics of the LOM elements. Table 1 displays a syntactical comparison between the XML and our proposed, CG-based, LOM binding. Apart from the more compact representation of the CG binding, the resulting CGs have, in some cases, better semantics. This is because some elements of the LOM record do not refer to the learning object itself but to some other entity like, for example, the intended learning object user. Similarly, the learner's model is encoded in CGs derived from the XML LM metadata (Table 2). Arbitrary level of detail can be gradually asserted in to a learner's model as his/her model evolves. The reverse conversion (i.e. from the CG notation back to the XML), has been also defined.

```
<EMAIL_CONTACT>
    <CONTEXT_LABEL>work</CONTEXT_LABEL>
    <EMAIL_ADDRESS>kokkoras@csd.auth.gr</EMAIL_ADDRESS>
</EMAIL_CONTACT>
```

```
[LEARNER:#12] → (owner) → [EMAILADDRESS: "kokkoras@csd.auth.gr"]
                                   |
                                   └→ (context) → [work]
```

Table 2: A simple Learner's Model element in XML and CG encoding

4.2 The CG-PerLS model

The CG-PerLS model is defined as a 5-tuple of the form:

$$\texttt{CG-PerLS} = (\texttt{LOs}, \texttt{Map}_1, \texttt{LOM}, \texttt{Map}_2, \texttt{KB})$$

where $\texttt{LOs}$ is a set of learning objects/resources located on the web ($\texttt{lo}^i \in \texttt{LOs}, i=1,...,n$) and $\texttt{LOM}$ is a set of LOM records ($\texttt{lom}^i \in \texttt{LOM}$) that describe the learning objects (LOs). The mapping relation $\texttt{Map}_1$ defines the correspondence between LOM records and learning objects. For instance, the mapping $\texttt{Map}_1(\texttt{lo}^3, \texttt{lom}^3)$ defines an one-to-one relationship between learning object 3 and the corresponding LOM record. $\texttt{KB}$ is a knowledge base encoded in CGs and $\texttt{Map}_2$ is an one-to-many mapping relation that maps a LOM record to concepts in the *content* and *learner* parts of the KB, since only these parts of the KB are directly derived from the educational metadata.

Thus, the KB logically consists of the following four parts (Figure 1):

- *system knowledge*: this mostly includes rules about how to handle CGs (formation rules, inference rules etc.),
- *content knowledge*: this is the knowledge related to the content of the learning resources; it is derived from their metadata records.
- *domain knowledge*: this includes knowledge related to but not explicitly defined in LO, such as the type hierarchy, concept/relation definitions and complete course descriptions. The latter are tree structures that describe, at the content level, the dependencies between LO. For example, Linear Algebra requires Matrix Calculus.
- *learner's knowledge*: this is the knowledge related to the learner who is accessing the learning resources. It represents the system's understanding of the student by defining parameters like who is the student (UserID, Password, e-mail address, Surname etc), what are his/her capabilities, preferences, equipment etc.

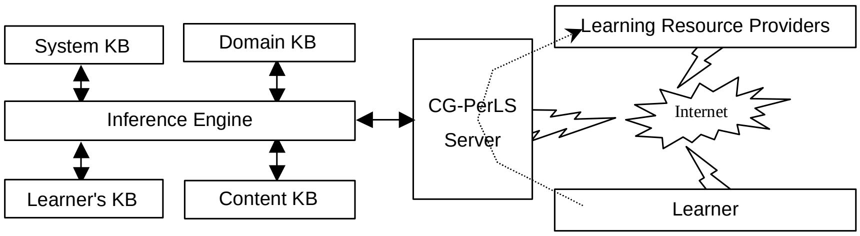

Figure 1: Abstract diagram of a CG_PerLS based educational portal.

The knowledge derivation process is fully automated for the case of LOM compliant educational resources. Otherwise, the resource provider should manually give that information filling in a form/questionnaire. All the inference is performed by a Prolog inference engine which acts as an application server. Therefore, we have implemented CGs in Prolog.

The basic building blocks in the knowledge base are CGs, concepts and conceptual relations. Following is the definition of these terms in the CG-PerLS data model using a Prolog-like notation. Conceptual Graphs are defined as predicates of the form:

`cg(ID, RelationList)`

where `ID` is a unique identifier associated with this CG and `RelationList` is a list that stores the conceptual relations of the specific CG.

A conceptual relation is defined as:

`cgr(RelationName, ConceptIDs)`

where `ConceptIDs` is a list of concept identifiers that this specific conceptual relation joins and `RelationName` is the name of the conceptual relation.

Concepts are represented as predicates of the form:

`cgc(ID, MetRecID, Context, ConceptName, ReferentField)`

where `ID` is a unique identifier associated with this concept, `MetRecID` is a unique identifier of the metadata record (LOM or LM) that contains this concept, `Context` is either *`normal`* for the case of normal concepts or *`special`* for the case of contextual concepts. `ConceptName` is the type-name of a normal concept or the context name of a contextual concept (situation, proposition etc.). `ReferentField` is a list that holds the referent field of the specific concept.

The concept and the relation type hierarchy are defined using `is_a` relationships. For example, `is_a(video_format,mpeg)` denotes that `mpeg` is a `video_format`. Note that other kinds of hierarchical relations such as equivalent, scope and association are not required to be explicitly defined since they can be expressed using proper relations in the domain KB.

The above Prolog oriented notation can handle complex CGs without information repetition since concepts are indexed separately. Furthermore, a CG can be traversed starting from any of its concepts [11]. Note also that, all the identifiers are only system-wide consistent since, at the moment, there is no specified method for the creation of a globally unique LOM record identifier (LTSC suggests not to use, at the moment, the metadata tag `General.Identifier`).

The $\mathtt{Map}_2$ mapping scheme is included into the above representations. The `MetRecID` argument in the `cgc/5` predicate holds this information as a term of the form:

- `lom_id(i)` which means that the concept is mapped to the LOM record `i`
- `lm_id(i)` which means that the concept is mapped to the LM record `i`
- `system` or `domain` for concepts belonging to the system and domain knowledge respectively.

The CG-PerLS model supports multiple-strategy, knowledge-based educational resource access which includes operator-based queries and dynamic course generation. We describe these access methods in the following sub-sections.

4.3 Operator-based CG-Queries

An operator-based user query is a set of one or more query CGs connected with logical operators (AND, OR etc). A query CG is a CG that the user wants to match against the KB. Usually, it contains concepts with unbound referent fields. A CG-PerLS query is defined as:

`Q(QCGs, SemanticFlag, MaxResults)`

where `QCGs` is a list of query CGs, `SemanticFlag` is a flag (*`true/false`*) denoting whether to use semantic match or not, and `MaxResults` is the desired maximum number of returned LOs.

The utilization of the KB is expected to increase the effectiveness of the operator-based learning resource access. This is due to the fact that exact term matching suffers in the following cases:

- *poor recall*: in this case, useful learning resources are not retrieved because their metadata contain a synonym or a semantically similar term rather than the exact one presented in the CG-PerLS query, and
- *poor precision*: too many learning resources contain the given term(s), but not all the retrieved ones are actually semantically relevant to the query.

The use of the KB (particularly the concept and relation type hierarchy) alleviates the poor recall problem. For example, an attempt to match the term "Logic Programming" with educational resources containing the term "Prolog" in their metadata record will succeed as soon as the KB includes knowledge about Prolog being a programming language for Logic Programming.

The KB can be used to improve precision as well. If a query has produced too many results, it is possible to use this knowledge to construct system queries, that is, queries constructed by the system and presented to the user, to improve the precision of the returned LOs. For example, if searching for video clips demonstrating Kepler's law has returned too many resources, then, the system can ask the learner if he/she is interested in any particular video encoding, given that such information is not included in the initial query. This system-side behavior is controlled by domain knowledge which defines such response patterns and is currently limited.

The existence of the KB provides two modes of operator-based query evaluation: *raw matching* and *semantic matching*. The mode is determined by the value of the `SemanticFlag` argument (*`false`* or *`true`*, respectively) in a query expression. The first case is straightforward: a learning resource "answers" a query if its metadata (in the CG form) match with the query CGs posed by the user and all the constraints introduced by the operators are satisfied. In the second case, a *similarity measure* (often called *semantic distance*) is required to be able to determine the extent to which two CGs may be considered "similar". Calculation of the similarity of two CGs depends upon the prior identification of appropriate "sources" of similarity. Such sources are the extend of use of the concept-type hierarchy and the ratio of arcs in the maximal join CG to the total number of arcs in the larger of the two CGs that participate in the maximal join operation (a form of unification between CGs). The contribution from any of the above sources of evidence of similarity can be equal or weighted. In general, the total similarity is defined as:

$$\mathtt{TotalSimilarity} = w_1 \bullet \mathtt{Evidence}_1 + w_2 \bullet \mathtt{Evidence}_2 + \ldots + w_N \bullet \mathtt{Evidence}_N$$

where w_i are the weights and $\sum w_i = 1$. This combined similarity allows for superior retrieval to that obtained by any individual form of evidence [1]. It is used as the measure to rank the results of an operator-based query.

Let us give an example of how the use of the concept-type hierarchy can be used as a similarity measure. For a semantic match, if two concepts are syntactically different from each other but they belong to the same branch of a concept-type hierarchy, the more specific one can be repeatedly generalized to shorten the semantic distance between them. Between two semantic matches, the one that uses fewer successive generalizations is more important since the semantic distance between this one and the matching concept is shorter. Thus it has higher rank. We restrict to generalization since specialization does not always preserve truth [14]. For example, specializing the concept [`mathematics`] to [`algebra`] is not correct in all contexts. Polysemy cases (ex. *bank* – financial institute / river) are dissolved based on the different conceptual definitions of the polysemy terms, together with the rest of the metadata elements of the metadata record in which the polysemy term occurs. These elements help to select the right CG definition automatically. If this is not possible, the ambiguity is manually dissolved by the user.

The operator-based query evaluation is performed by recursively decomposing it into sub-expressions and matching them against the domain knowledge. If a query term does not match a metadata element entry, a try to generalize this query term using the concept-type hierarchy is performed. On successful generalization, the matching try is repeated, this time for the term which is the result of the generalization. This is depicted in Figure 2. The user decides whether to use raw or semantic term matching in a query. If not enabled, the semantic term matching method is automatically invoked when the raw term matching fails to give results or the number of results is below a user defined threshold.

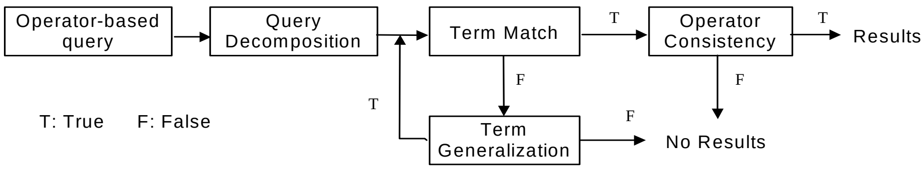

Figure 2: Semantic term matching through "Term Generalization".

Notice that the effect of an operator-based query to a CG-PerLS metadata KB is the derivation of a ranked subset of learning resources. This allows users to recursively refine their queries, that is, the query evaluation process is recursive.

4.4 Dynamic Course Generation

The term Dynamic Course Generation refers to the identification of the learning objects that the learner should successfully attend to improve his/her knowledge, according to his learning request and current knowledge state. At the current state of our model, we are not considering assessment results. This would required a CG binding for the related metadata which is, for the moment, not included in the CG-PerLS knowledge base. Therefore, we assume that the learner gets the knowledge of a LO as soon as he/she accesses it.

Given an individual's current knowledge state KS_1 and a target knowledge state KS_2 defined by his/her learning request, we want to find a way (*learning path*), in terms of proper curricular elements, that will enable the learner to evolve from KS_1 to KS_2, taking at the same time into account his/her preferences. As soon as the system locates a learning object that satisfies the user's learning request, it uses appropriate (`record.relation`) metadata elements of this LO in an attempt to "break" it down into fine grained LOs. For example, in order for a learner to successfully "interact" with a learning resource about "Linear Algebra" he should be familiar with "Matrix Calculus".

This process is illustrated in Figure 3, where **ΔΚΣ** is the knowledge of the LO_1 that covers directly the user's learning request, augmented with the knowledge of additional LOs the user should "attend" to become able to "attend" LO_1. The set of all the required LOs constitute the Learning Path that will transfer the learner from KS_1 to KS_2.

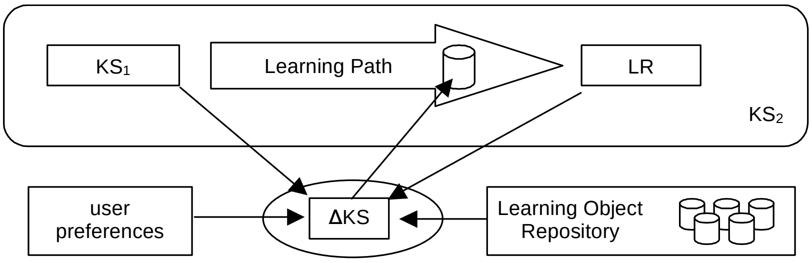

Figure 3: Abstract definition of Personalized Knowledge Path construction

5 Related Work

A very early adoption of CGs for semantic information retrieval can be found in COMFRESH [6] where the content of nodes in a hypertext network is selectively encoded in CGs and a CG based inference engine provides semantic query and retrieval of hypertext data, as well as dynamic, knowledge based hypertext links.

AI-Trader [12] is a type specification notation based on Conceptual Trees (a special case of CGs) to support service trading in open distributed environments where services are freely offered and requested. Our work resembles AI-Trader in the spirit of trading educational services, and goes one step further introducing the metadata usage.

WebKB [8], is a tool that interprets semantic statements (CGs) stored in Web-accessible documents. WebKB aims towards the semantic Web concept and, from one perspective, can be seen as an elaborated COMFRESH model.

Corby et al [3] 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, Corese has some relation to WebKB and mainly to COMFRESH, but goes beyond both of them since it uses the modern RDF/RDFS statements as an information source and presents its results with XSLT style sheets.

Murray describes [9] a framework called Model for Distributed Curriculum (MDC) that uses a topic server architecture to allow a Web-based tutorial to include a specification for another tutorial where the best fit to this specification will automatically be found at run time. A specific reasoning mechanism towards this functionality is not presented though.

DGC [15] is a tool that generates individual courses according to the learner's goals and previous knowledge and dynamically adapts the course according to the learner's success in acquiring knowledge. Unlike CG-PerLS which is based on the metadata info, DGC uses "concept structures" as a road-map to generate the plan of the course.

6 Conclusions - Future Work

We have presented CG-PerLS, a knowledge based approach on organizing and accessing educational resources. CG-PerLS is a model of a web based educational brokerage system which combines the descriptive power of metadata with the inference power of Conceptual Graphs to allow learning resource providers to manifest proprietary or LOM aware educational resources, client-side learners to access these educational resources in a way tailored to their individual profile and educational needs, and dynamic course generation.

A system based on the CG-PerLS model is in development. The core of our approach has been already successfully tested in the prototype of the COMFRESH system [6]. What remains is to prove the efficiency of the approach. This will be presented in a future, evaluation paper.

Furthermore we plan to add a cooperating evaluator module, in the form of a client-side agent that will further rank the knowledge transfer of a learning resource according to the assessment results the learner got. This information can be sent back to the CG-PerLS server and used to improve its overall knowledge transfer to the learner by preferring to serve him/her with a specific learning object from a set of similar ones, based on the assessment results obtained by learners with similar profiles. This will require substantial research on aspects related to the learner's model.

Acknowledgements

This work is partially funded by the European Commission Information Society Technologies (IST) Program through the IST No 12503 KOD "Knowledge on Demand" Project.

References

[1] Brkich F. and Wilkinson R.: "A Conceptual Graph Matching Algorithm for Information Retrieval", *Proceedings 1st Australian Conceptual Structures Workshop*, Univ. of New England, 1994.

[2] Brusilovsky P.: "Adaptive and Intelligent Technologies for Web-based Education", in *Kunstliche Intelligenz*, (special issue on Intelligent Systems and Technology) by Rollinger C. and Peylo C. (eds.), 1999.

[3] Corby O., Dieng R. and Hebert C.: "A Conceptual Graph model for W3C Resource Description Framework", *Proceedings 8th International Conference on Conceptual Structures (ICCS'2000)*, Springer, 2000.

[4] "Conceptual Graphs Standard Working Draft", Available at http://www.jfsowa.com/cg/cgstand.htm

[5] Gerbe O.: "Conceptual Graphs for Corporate Knowledge Repositories", *Proceedings 5th International Conference on Conceptual Structures (ICCS'97)*, Springer, 1997.

[6] Kokkoras F. and Vlahavas I.: "COMFRESH: A Common Framework for Expert Systems and Hypertext", *Information Processing and Management*, Vol.31 (4), 1995.

[7] IEEE P1484.12/D6.0: "Draft Standard for Learning Object Metadata", Feb-2001.

[8] Martin Ph. & Eklund P.: "Embedding Knowledge in Web Documents", *Proceedings 8th International WWW Conference* (special issue of The International Journal of Computer and Telecommunications Networking), Toronto, Canada, 1999.

[9] Murray T.: "A Model for Distributed Curriculum on the WWW", *Journal of Interactive Media in Education*, (5), 1998, [www-jime.open.ac.uk/98/5].

[10] IEEE P1484.2/D7: "Draft Standard for Learning Technology - Public and Private Information (PAPI) for Learners (PAPI Learner)", Nov 2000.

[11] Petermann H., Euler L. and Bontcheva K.: "CG-Pro: A Prolog Implementation of Conceptual Graphs", Technical Report, Hamburg University, FBI-HH-M-251/95, 1995.

[12] Puder A., Markwitz S. and Gudermann F.: "Service Trading Using Conceptual Graphs", *Proceedings 3rd International Conference on Conceptual Structures (ICCS'95)*, Springer, 1995.

[13] ADL: "Sharable Courseware Object Reference Model (SCORM)", Version 1.1, Jan 2001, http://www.adlnet.org

[14] Sowa J.: "Conceptual Structures: Information Proccesing in Mind and Machine", Addison-Wesley, 1984.

[15] Vassileva J.: "Dynamic Course Generation on the WWW", *Proceedings 8th World Conference on AI in Education (AI-ED97)*, Knowledge and Media in Learning Systems, Kobe, Japan, 1997.