A Knowledge Based Approach on Educational Metadata Use

Fotios Kokkoras[1], Dimitrios Sampson[2], and Ioannis Vlahavas[1]

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

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

Abstract. One of the most rapidly evolving e-services is e-Learning, that is, the creation of advanced educational resources that are accessible on-line and, potentially, offer numerous advantages over the traditional ones like intelligent access, interoperability between two or more educational resources and adaptation to the user. The driving force behind these approaches is the definition of the various standards about educational metadata, that is, data describing learning resources, the learner, assessment results, 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". 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 web portal for learning objects that encodes the educational metadata in the Conceptual Graph knowledge representation formalism, and uses related inference techniques to provide advanced functionality. The model allows learning resource creators to manifest their material and client-side learners to access these resources in a way tailored to their individual profile and educational needs.

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.

Y. Manolopoulos et al. (Eds.): PCI 2001, LNCS 2563, pp. 201-216, 2003.

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 commonplace 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 just-in-time, anytime and anywhere, tailored to his/her personal learning needs.

Until recently, the majority of the, so-called, "e-learning" approaches were limited to simple hyperlinks between content pages and "portal pages" organizing a set of related links. 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 prevented 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 group and the Learner Model 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.

The above standardization efforts, together with generic and robust knowledge based approaches for problem solving, developed during the last two decades, offer a fertile ground for the development of advanced on-line educational services. In addition, 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. Furthermore, the standardization will dramatically improve the reusability of the educational material. This is very important since educational material is expensive to create in terms of cost and time.

The wishful state of the art in e-Learning systems described as "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). 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, and the learner's model, that is, the perception that the system has about the learner's current knowledge status and goals.
- Problem Solving Support: the system offers Intelligent analysis of learner's solution, Interactive Problem Solving Support and Example based Problem Solving Support.
- *Adaptive Presentation*: that is, the system's ability to adapt the content of the supplied curriculum to the learner's preferences.
- *Student Model Matching*: based on a categorization of the learners to classes with similar educational characteristics, the system will be able to provide collaborative problem solving support and intelligent class monitoring.

Such e-Learning systems will be easier to implement when the standardization efforts come to an end. The internal details of these systems are an open issue since the standardization efforts are primarily dealing with "what" and not "how". However, the IMS Global Learning Consortium has developed a representation of learning object and learner's metadata in XML, namely the IMS-LOM [21] and IMS-LIP [22] specifications (or simply LOM and LIP, respectively).

One of the most ambitious efforts on e-Learning that make use of educational metadata is the Advanced Distributed Learning (ADL) initiative [13]. Recently, ADL released the Sharable Courseware Object Reference Model (SCORM) 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.

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 resources that is based on a Conceptual Graph binding of educational metadata and related inference techniques. 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 the user's learning request.

The rest of the paper is organized as follows: Section 2 presents the basic features of Conceptual Graphs (CGs), a formalism for knowledge representation and reasoning. Section 3 describes an integration of the educational metadata and the conceptual graphs technologies, in the form of the CG-PerLS model. Section 4 outlines related work and, finally, section 5 concludes the paper.

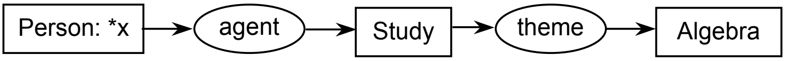

Fig. 1. A Conceptual Graph stating that "there is some person *x* studying Algebra"

2 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 (Fig.1). 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 (Fig.1). A relation type (also called the *signature* of the relation) determines the number of arcs allowed on the relation. Each relation has zero or one incoming arcs and one or more outgoing arcs. For example, the signature of the relation *agent* is (*Act*, *Animate*), which indicates that the type of the concept linked to its first arc must be *Act* or some subtype, such as *Go*, and the type of the concept linked to its second arc must be *Animate* or some subtype, such as *Person*.

A CG (Fig.1) 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 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 *cat* would also have all the properties of *mammal*.

In the CG formalism, every context (situation, proposition, etc.) is a concept. Thus, contexts are represented as concepts whose referent field contains a nested CG (contextual concepts). 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. The canonical formation rules are:

- *Copy* creates a copy of a CG.
- *Restriction* takes a graph and replaces any of its concept nodes either by changing the concept-type to a subtype or adding a referent where there was none before.
- *Joining* joins two graphs with a common concept over it, to form a single graph. The resulting graph is the common specialization of the two graphs.
- *Simplifying* removes any duplicate relations between two concepts.

Other operations on CGs include:

- *Contraction* tries to replace a sub-graph of a given CG with a single, equivalent concept (or relation), using the CG definition of this concept (or relation).
- *Expansion* is the opposite of the contraction operation.
- *Maximal Join* is a join of two CGs followed by a sequence of restrictions, internal joins and simplifications so that the maximum amount of matching and merging of the original graphs is achieved.

- *Unification* is the complex process of finding the most general subtypes for pairs of types of concepts. The resulting graph of two graphs being unified has exactly the same information as the two individual graphs together.
- *Projection* is an complex operation that projects a CG over another CG. The result is a CG that has the same shape/structure as the projected CG (this is also a requirement to be able to perform projection), 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.

Inference rules based on CGs have also been defined. A rule $R:G_1 \Rightarrow G_2$ is composed of two CGs, G_1 and G_2, which are called *hypothesis* and *conclusion*, respectively (Fig.2). There may be coreference links between concepts of 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 Fig.2 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.

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 the Knowledge Interchange Format (KIF) and are currently under a standardization process [4].

A number of CG related software tools have been implemented by various researchers and/or research teams. One such tool is the CoGITaNT (Conceptual Graphs Integrated Tools allowing Nested Typed graphs) [17], a library of C++ classes that allows to design software based on Conceptual Graphs. CoGITaNT is available under a GNU General Public License [18].

3 Conceptual Graphs for Educational Services

Since the use of artificial intelligence methods towards Adaptive and Intelligent Web-based Educational Systems (AIWES) seems inevitable, the seamless integration of educational metadata into a knowledge based framework will offer better possibilities for AIWES. We propose next such an integration approach that binds certain educational metadata onto CGs. We use that binding to establish CG-PerLS, a knowledge based approach (model) for manifesting and accessing educational resources over the Internet.

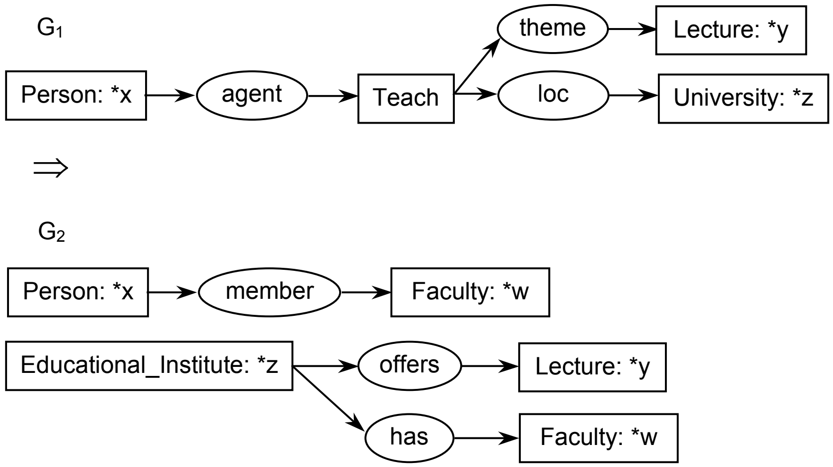

Fig. 2. An example of a CG Rule

3.1 A Conceptual Graph Binding for Educational Metadata

The LOM and LIP standards precisely define data models and their semantics for describing the properties of learning objects and learners, respectively. Any LOM/LIP compliant system is free to internally handle these metadata in any way, but should be able to meaningfully interchange such metadata with other systems without loss of any information.

We selected to map these metadata to CGs in order to seamlessly incorporate them into a knowledge based educational portal. 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 defining 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].

In addition, 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 [20]. CGs and the operations defined over them can be used not only as powerful tools to create ontologies, but also as reasoning tools over ontologies [16]. In that way, the previously static knowledge representation of an ontology is becoming a dynamic, functional reasoning system. This is a very important feature because the educational ontologies that will be required towards AIWES will be better integrated into a knowledge based educational system if both share common representation formalism. Furthermore, the graphical representation of CGs allows easier interpretation of the knowledge that they encode. This is not the fact for XML encoded data, since the XML notation aims at better, general purpose, machine data handling.

Therefore, we suggest that, a metadata binding into a knowledge representation formalism, such as CGs, allows for better integration of metadata into knowledge based e-Learning systems and particularly to systems that built upon the CG formalism.

```
<record>
        <general>
                <identifier>123</identifier>
        ...
        </general>
        ...
        <classification>
                <keyword>
                        <langstring xml:lang="en">Kepler's Law</langstring>
                </keyword>
                <keyword>
                        <langstring xml:lang="en">simulation</langstring>
                </keyword>
        </ classification>
        ...
        <educational>
                <interactivitytype>
                        <source>
                                <langstring xml:lang="x-none">LOMv1.0</langstring>
                        </source>
                        <value>
                                <langstring xml:lang="x-none">Active</langstring>
                        </value>
                </interactivitytype>
                ...
                <typicalagerange>
                        <langstring xml:lang="x-none">16-99</langstring>
                </typicalagerange>
                ...
                <typicallearningtime>
                        <datetime>0000-00-00T00:45</datetime>
                </typicallearningtime>
                ...
        </educational>
        ...
<record>
```

Fig. 3. LOM record (partial) in XML binding

We have defined CG templates to capture the semantics of the LOM elements. Fig.3 and 4 display 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 LOM is primarily a data model with it's semantics lying into the specification document of the model. For example, the <*typicalagerange*> element of the LOM record do not refer to the learning object itself but to the intended learning object user. This is better depicted into the CG binding of LOM (Fig.4). Similarly, the learner's information is encoded in CGs derived from the XML LIP metadata. Arbitrary level of detail can be gradually asserted into a learner's model as his/her model evolves.

```
...
[LO:#123] → (classification) → [KEYWORD:%en "Kepler's Law"]
[LO:#123] → (classification) → [KEYWORD:%en "simulation"]
...
[LO:#123] → (interactivity)→[INTERACTIVITY_TYPE:%x-none "Active"]
...
[LO:#123] ← (theme) ← [USE] → (agent) → [Person] → (typ_age) → [AGE_RANGE: :%x-none "16-99"]
...
[LO:#123] → (has_duration) → [DATETIME: 0000-00-00T00:45]
...
```

Fig. 4. The LOM record of Fig.3, into the CG binding

We use this CG binding of metadata to establish the CG-PerLS*,* a knowledge based approach (model) for manifesting and accessing educational resources over the Internet.

3.2 The CG-PerLS Model

The CG-PerLS is a web-enabled model of a knowledge-based system that uses the Conceptual Graph knowledge representation formalism and related inference techniques to reason over educational metadata expressed as CGs. The outline of the model is illustrated in Fig.5 where the CG-PerLS plays the role of an application server in a WWW educational portal. CG-PerLS consists of:

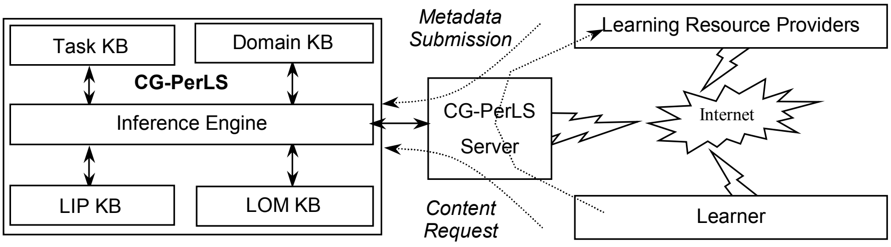

Fig. 5. Abstract diagram of a CG_PerLS based educational portal

- an *inference engine* capable of performing forward and backward reasoning over CGs and CG rules, and having TCP/IP networking capabilities,
- the *lom KB*, which is the knowledge related the learning resources; it is automatically derived from their metadata records as long as these resources are declared to the system,
- the *lip KB*, which 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 (user identification, password, e-mail address, surname etc), what are his/her capabilities, preferences, equipment etc.,
- the *domain KB*, which includes knowledge related to but not explicitly defined in learning objects, such as the concept type hierarchy, concept/relation definitions and the definition of courses. A *course* is described as a set of CGs outlining the structure of the course. For example, a course includes some semester curricula, which consist of modules, each of them having a number of lectures about some theme. There is no direct link to any particular educational material at this point. Since, in the general case, there would be more than one available learning objects aiming at some lecture theme, the selection of the proper learning objects to cover a course (or a part of a course) for a particular learner, would be an inference problem. That is, the application of the axiomatic knowledge presented in the *domain* and *task* KBs to the fact CGs (CGs without generic referents) in the LOM and LIP KBs, in order to arrive to some conclusion. Furthermore, the dependencies between themes are also described. For example, "Linear Algebra" requires "Matrix Calculus".
- the *task KB*, which materializes the services the system is able to offer to the user as well as its internal operations. Such services include, without limited, the binding of learning objects over a course description, the packaging and delivery of all the information describing in detail what the learner should study, bidirectional transformations between LOM/LIP and CG-LOM/CG-LIP bindings, generation of dynamic hyperlinks for the user to traverse and the communication of the CG-PerLS server with other similar servers.

The knowledge derivation process is fully automated for the case of LOM compliant educational resources. This is done by a module that reads the XML LOM metadata that come with the learning resource and converts then to the system's internal CG representation, according to the CG-LOM binding. Otherwise, the resource provider should manually give that information by filling in a form/questionnaire. All the inference is performed by a CG inference engine (Fig.5) implemented using the CoGITaNT library [18].

Forward and backward execution methods of CG rules have been implemented. *Forward Chaining* (or data driven) inference, is typically used in order to explicitly enrich facts with knowledge which is implicitly presented in some knowledge base. If G is a fact and R is a rule, then if G fulfills the hypothesis part of R, then the conclusion of R can be added to G. Forward chaining is primarily used to enrich the learner's model according to his/her progress.

Backward chaining is primarily used when we want to "prove" something. Since this operation requires some splitting and unification operations over CGs it is used mainly in the course generation task, where a course template, which refers to generic (unbound) learning themes, is enriched with specific learning resources.

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.

3.3 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. It contains concepts with either individual markers (bound referent fields) or generic markers (unbound referent fields). A CG-PerLS query is defined as:

```
Q(QCGs, SemanticFlags, MaxResults)
```

```
[LO: x] → (classification) → [KEYWORD:%en "Kepler's Low"]
        ↘ (interactivity) → [INTERACTIVITY_TYPE:%x-none "Active"]
```

Fig. 6. A query CG with a transparent AND operator

where *QCGs* is a set of query CGs connected with logical operators, *SemanticFlags* is a set of flags (*true*/*false*) denoting whether to use semantic match or not on specific concepts of the query CG, and *MaxResults* is the desired maximum number of returned learning objects.

Currently, our inference engine can transparently handle only the AND logical operator. We demonstrate this with the following example: Consider a learner that wants some learning objects related to "Kepler's Low" which are of interactivity type "Active". In terms of elementary CGs (see Fig.4) this request is expressed like:

[LO: x] → (classification) → [KEYWORD:%en "Kepler's Low"]
AND
[LO: x] → (interactivity) → [INTERACTIVITY_TYPE:%x-none "Active"]

where *x* is a coreference point between the two CGs. *x* should be bound to an identification number of some individual learning object. That would require, from the inference engine point of view, to handle the two CGs separately and interpret the logical operator at a later stage. Instead, the inference engine can directly handle the more compact and semantically equivalent representation presented in (Fig.6).

Note that this representation is the exact expression that is given to the inference engine. Other types of logical operators are currently handled outside the CG formalism. That is, the individual CG queries are answered separately and then the operator consistency is checked.

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 *domain 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 *domain 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 learning objects. 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 task 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:

$$\texttt{TotalSimilarity} = w_1 \bullet \texttt{Eviden}_1 + w_2 \bullet \texttt{Eviden}_2 + \ldots + w_N \bullet \texttt{Eviden}_N \;. \tag{1}$$

where w_i are the weights and $\Sigma w_i=1$. This combined similarity allows for superior retrieval to that obtained by any individual form of evidence [1]. Currently, we only utilize the concept type hierarchy as a similarity measure.

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.

If a part of a query CG does not match some CG-LOM metadata element entry, a try to generalize certain concepts 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 Fig.7. The system can automatically switch to semantic term match if direct term match produces no result. However, the user decides whether to use semantic term matching or not on some concepts using the semantic flag of each concept.

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.

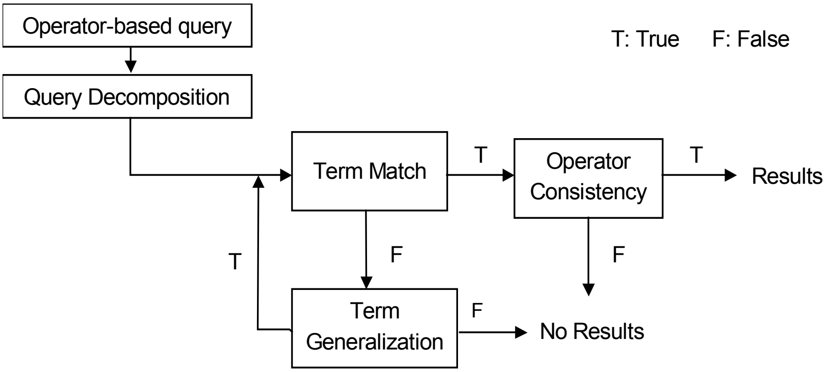

Fig. 7. Semantic term matching through "Term Generalization"

3.4 Dynamic Tutorial Generation

The term Dynamic Tutorial Generation refers to the identification of the learning objects that the learner should successfully attend in some specific order, to improve his/her knowledge, according to his learning request and current knowledge state. At the current state our model does not take into account any 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 learning object as soon as he/she accesses it.

Given an individual's current knowledge state KS_1 and a target knowledge state KS_2 where $KS_2=KS_1+DKS$, we want to find a way (*learning path*), in terms of proper curricular elements, that will enable the learner to "get" the extra knowledge DKS and evolve from KS_1 to KS_2. At the same time we want to ensure that the suggested curricular elements will not violate any user preferences such as difficulty level, media type, hardware requirements, time constraints etc. 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 learning object in an attempt to "break" it down into fine-grained learning resources. 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 outlined in Fig.8, where DKS is the knowledge of some learning object, say LO_1, that covers directly the user's learning request, augmented with the knowledge of additional learning objects the user should "attend" to become able to "attend" LO_1. The set of all the required learning objects constitute the Learning Path that will transfer the learner from KS_1 to KS_2.

Two example rules that are used in the dynamic tutorial generation are presented in Fig.9. Rule R_1 augments a learning object *x* that has as prerequisite the learning object *y* physically located at URL *z*, with a dynamic hyperlink to that URL. That is, the conclusion of the rule enriches the [LO:*x] learning object with additional information. Rule R_2 is similar to R_1 in the first part but it creates a system event that forces the CG-PerLS server to communicate with peer servers, instead. The particular rule will force the system to ask some known peer server for the physical location (URL) of the learning object with identifier *id2*.

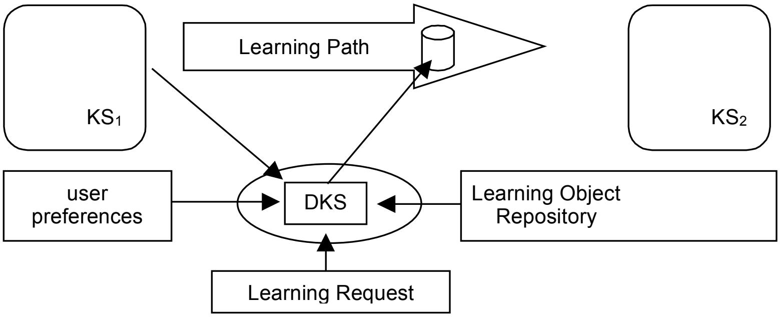

Fig. 8. Abstract definition of Personalized Knowledge Path construction

R1
[LO: *x] → (requires) → [LO: *y] → (phys_loc) → [URL:*z]
⇒
[LO: *x] → (d_link) → [URL:*z]

R2
[LO: *x] → (requires) → [LO: *y] → (phys_loc) → [URL:*z]
⇒
[SYS_EVENT: p2pquery] → (q_params) → [QCG: *ref*]

where *ref* corresponds to:
" [LO: id2] → (phys_loc) → [URL:?] "

Fig. 9. CG rules of the agent's task knowledge base

Communication between peer CG-PerLS servers requires that they share a common identification scheme. Of course this is also true for the domain knowledge.

4 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 [9] describes 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.

5 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 prototype system based on the CG-PerLS model is in development on the wintel platform. Some core ideas of our approach have been already successfully tested in the past, in the prototype of the COMFRESH system [6] using Prolog.

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 work on aspects related to the learner's model. Some early thoughts on this aspect have been already outlined in [19].

Acknowledgements

The work presented in this paper 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*, University of New England, New South Wales, 1994.

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

[3] Corby O., Dieng R. and Hebert C.: "A Conceptual Graph model for W3C Resource Description Framework", *Proceedings 8th 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 Conference on Conceptual Structures (ICCS'97),* 1997.

[6] Kokkoras F. and Vlahavas I.: "COMFRESH: a Common Framework for Expert Systems and Hypertext", *Information Processing and Management*, Vol.31, No.4, 1995.

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

[8] Martin P. and Eklund P.: "Embedding Knowledge in Web Documents", *Proceedings 8th 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*, Vol.5, 1998. Available at: 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)", November 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 Conference on Conceptual Structures (ICCS'95),* Springer, 1995.

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

[14] Sowa J.: "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley Publishing Company, 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.

[16] 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 Vol. 2256, pp.130-141, 2001.

[17] Genest D. and Salvat E.: "A Platform Allowing Typed Nested Graphs: How CoGITo Became CoGITaNT", *Proceedings 6th Conference on Conceptual Structures*, Montpellier, France, pp.154-161, 1998.

[18] The CoGITaNT library. Available at http://cogitant.sourceforge.net/

[19] Kokkoras F. and Vlahavas I.: "eLPA: an e-Learner's Personal Assistant", *Proceedings Workshop on Application of Conceptual Structures*, Bulgaria, 2002. Available electronically only at http://lml.bas.bg/iccs2002/acs/kokkoras.pdf

[20] Gerbe O. and Mineau G.: "The CG Formalism as an Ontolingua for Web-Oriented Representation Languages", *Proceedings 10th Conference on Conceptual Structures (ICCS'02)*, pp. 205-219, 2002.

[21] IMS Global Learning Consortium, Learning Resource Metadata (LOM) Specification, Version 1.2.1, Final Specification, September 2001. Available at http://www.imsglobal.org/ metadata/index.html

[22] IMS Global Learning Consortium, Learner Information Package (LIP) Specification, Version 1.0, Final Specification, March 2001. Available at: http://www.imsglobal.org/profiles/index.html