

Semantic Search of Learning Services in a Grid-Based Collaborative System

Guillermo Vega-Gorgojo, Miguel L. Bote-Lorenzo,
Eduardo Gómez-Sánchez, Yannis A. Dimitriadis, Juan I. Asensio-Pérez
School of Telecommunications Engineering, University of Valladolid
Camino del Cementerio s/n, 47011 Valladolid, Spain
{guiveg, migbot, edugom, yannis, juaase}@tel.uva.es

Abstract

CSCL systems can benefit from using a grid, since it offers a common infrastructure allowing an extended pool of resources that can provide supercomputing capabilities as well as specific hardware resources. Adopting a service oriented architecture such as OGSA can further benefit CSCL systems, enabling increased flexibility to adapt and reuse learning software. However, proliferation of services enormously challenges the search of appropriate ones. Common service discovery mechanisms, such as the Index Service or UDDI, perform poor searches since they rely on keyword matching and do not properly describe service functionality. To address these difficulties, ontologies can be used to formally describe services allowing for semantic searches. This paper proposes an ontology that describes learning services using educators abstractions in order to support the search. This way, service providers can describe their services in terms of the ontology, while educators can search for them using domain concepts. The integration of this enhanced service search mechanism in Gridcole, a real grid-based CSCL system, is described to illustrate its performance.

1. Introduction

E-learning [1] is a research field that intends to promote learning through the application of information and communication technologies. Recently, the *grid* has been proposed to provide a common infrastructure to build the various e-learning implementations [13]. As a result, many research communities, such as the ELeGi project [10], have begun to develop e-learning systems on top of a grid to increase learning effectiveness.

The grid enables resource sharing in e-learning systems. Educational organizations such as schools or universities

can seamlessly federate and share their resources in a grid. This way, new learning scenarios can be performed using a grid infrastructure. For instance, a very specific simulator of parallel architectures could be available in a grid and employed in different computer architecture courses in various universities.

According to [16], *Computer Supported Collaborative Learning* (CSCL) is a paradigm that constitutes a special type of e-learning. This theory considers that knowledge and social skills are acquired through the collaboration interactions performed during learning. CSCL applications make use of technology to support collaborative learning.

CSCL can also benefit from using a grid infrastructure. For instance, large sets of students distributed among distant schools can be arranged to collaborate in the development of an electronic journal by using grid resources. Another example is CoVis [24], a CSCL application that needs supercomputing capabilities.

Although the grid can provide a significant amount of computational, software and data resources for CSCL, developing CSCL applications is a demanding task that implies a considerable effort. As [19] claims, it can only be justified if these resources can be used in multiple learning scenarios. However, this is only possible if educational software is flexible enough to cope with the enormous differences in curricula and teaching styles [25]. This issue could be tackled with the *service oriented computing* model [22] which advocates increased flexibility, adaptability, reusability and reduced complexity to deliver software. This way, instead of developing or purchasing a software product, a service offered by an adequate provider could be employed.

The *Open Grid Services Architecture* (OGSA) [12] is the emerging architecture for the construction of grids following the service oriented computing paradigm. Therefore, CSCL could embrace OGSA to exploit both resource availability in a grid and flexibility of services, such as the CSCL system Gridcole [3].

An important concern is how to discover appropriate services in a grid to support CSCL activities. There is need to provide some mechanisms to offer information about available services. Registries are widely used to perform this functionality, enabling providers to register and publish metadata about services, while consumers can find services that meet their criteria. Some well-known registries are UDDI [20] in the popular Web Services architecture [5] or the Index Service in Globus Toolkit 3 [14] (formerly MDS-2 [6]).

However, discovery capabilities of such registries are somewhat limited, since they only rely on syntactic inter-operation, as reported in the literature [17, 15]. Moreover, it is very difficult to discover an unknown service, because providers can publish any information of their announced services and there is not a previous agreement with service requesters. Instead, services could be enriched with semantic information allowing an enhanced discovery mechanism. This is the approach taken in [21], with the so-called Semantic Web Services. Ontologies [4] are used to explicitly formalize knowledge, enabling rich descriptions and robust information retrieval systems. The Semantic Grid initiative [7] is consistent with this vision to construct third generation grids.

This paper discusses the use of an ontology of learning tools in order to facilitate both service description by providers, and service discovery by educators. With its use, providers can announce their services described in terms of the ontology, and educators can search for services using the learning domain abstractions described in the ontology in educator-level terms. A registry will enforce providers and educators to commit to the concepts in the ontology, while it can infer some kind of reasoning to match educators' requests with service announcements. To illustrate the potential of this ontology, its operation in Gridcole, a real grid-based CSCL system [3], will be described with an example.

The rest of the document is organized as follows: in Section 2 current service discovery mechanisms in some popular service oriented architectures are analysed. Besides, the use of ontologies to search for services is discussed for the case of a CSCL system. Section 3 proposes an ontology to describe learning services. Section 4 shows the integration of an ontology-enabled search mechanism in the real CSCL system Gridcole, examples are provided to illustrate its performance. Finally, the main conclusions of the study are shown as well as current research work.

2. Service Discovery Mechanisms

This section first analyses the service discovery issue in some of the most relevant service oriented architectures. Next, it follows a discussion about ontology-based

approaches to perform the search of services in a CSCL system.

2.1. Service Discovery in Service-Oriented Architectures

With the proliferation of services, it is becoming increasingly difficult to find one satisfying some given requirements. The Universal, Description, Discovery and Integration (UDDI) standard [20] is a method for publishing and discovering services. Although the UDDI standard was proposed for the popular Web Services [5] architecture, UDDI can also be used in other service oriented architecture, such as OGSA [23].

Basically, UDDI is a registry of businesses and Web Services. An organization can publish in a UDDI registry information about the actual organization and the services it offers. Technical details to invoke a service (the service interface) can be attached, as well as other metadata such as taxonomical information.

When used to perform a service search, UDDI allows querying for service name, location, business or bindings. An important shortcoming is that UDDI does not represent service capabilities, since only a name and a textual description can be typically provided. Although additional features can be described by means of metadata, this mechanism is provider dependant.

Another drawback is that the matchmaking mechanism is keyword-based, resulting in a very weak search. For example, querying for a "questionnaire tool" would not find a service published as a "test tool". These problems have been reported in the literature, stating that "UDDI provides a very weak discovery mechanism which does not allow the discovery of Web Services" [17] or "UDDI is useful only to find information about known Web services, but it completely fails as a general Web Services discovery mechanism" [15].

In grid implementations other registries such as the Index Service in Globus Toolkit 3 [14] (formerly MDS-2 [6]) are commonly used. The main difference with UDDI is the need of a registry that allows for soft-state registration. This property enables the dynamic addition and deletion of information sources, which is important due to the dynamic nature of resources in a grid.

Although heterogeneous resources can be offered in a grid, OGSA mandates every resource to be represented as a Grid Service. The Index Service aggregates service data elements (SDEs) [12] of available Grid Services. However, it is not fixed what SDEs publish a Grid Service. This fact hinders service discovery, since a provider can register any type of information in the Index Service. Moreover, current implementation of the Index Service allows a very restricted set of queries: by specifying the Grid Service Han-

andler (GSH) [12] of a known Grid Service or by specifying an SDE name. Therefore, it is extremely difficult to discover an unknown service using this registry.

Another popular service oriented architecture is Jini [26]. Jini is a Java-centered service architecture; the Lookup Service is part of Jini's infrastructure and allows the service discovery. Publication is also based on soft state. Service interfaces are registered and can be queried through the Lookup Service and service attributes can be specified to support the discovery. However, the query language only allows for simple string matching on attributes.

There are more similarities than differences among these discovery mechanisms. All the three are registries and rely on keyword matches. Moreover, neither describes properly service functionality. Any provider can register a service and give it a meaningless name, hindering its search.

2.2. Ontologies to Perform Service Discovery

To improve the search, an unambiguous description of service functionality is needed. Furthermore, this description should be more than a collection of keywords. It is expected that the search infrastructure could better support the user and retrieve accurately the services that match her criteria. This implies an understanding of the semantic content of service descriptions.

Ontologies [4] constitute a technique that allows the explicit description of knowledge. Besides, an ontology is expressed in a formal language, enabling software agents to interpret it. Ontologies are the key mechanism to achieve the Semantic Web [2]: an initiative to semantically annotate Web contents so that information could be not only human but machine interpretable.

In this sense, the so-called Semantic Web Services [21] use ontologies to semantically annotate Web Services to achieve automated discovery, invocation and composition. To accomplish it, the OWL-S [18] (formerly DAML-S) ontology of services has been developed. Using this ontology, service providers can semantically annotate their services and advertise them in a OWL-S enabled registry. Such a registry understands the concepts and relations defined in OWL-S and can perform smart service searches. Although OWL-S was also conceived to allow automated invocation and composition, in this paper we are mostly concerned with the discovery. As OWL-S is an evolving ontology, only the discovery features contained in the latest release (version 1.1, November 2004) are analysed.

The OWL-S framework provides a way to describe the services offered by the providers and the services needed by the requesters. This description includes non-functional properties such as the name of the service, a textual description, contact information and quality of service parameters, although this last item is an unbounded list that can contain

any type of information.

However, OWL-S is more concerned about describing functional properties and expresses the service functionality in terms of the transformation produced by the service. Specifically, it defines the inputs required and the outputs generated; furthermore, it also defines the preconditions required to invoke a service, as well as the expected effects that result from the execution. Additionally, a service can refer to an external classification, such as the well-known UNSPSC¹ industry standard.

Therefore, when used to perform a search, the requester could specify any (or maybe a subset) of the precedent elements. For instance, we could ask for a service which receives a `StudentListing` and a `GroupSize` as input parameters and returns a `StudentGroupListing`. During the matching phase some kind of inference could be done and a service which receives a `Listing` (not restricted to students) as input could be retrieved.

However, this type of search assumes a technological view of the services that is not always suitable. Just consider the case of an educator who wants to integrate some learning tools in a CSCL system, the educator would be troubled to describe the inputs and outputs of the service. Clearly, it is needed another way to perform searches so that educators could describe learning tools in their own terms. Providers could announce their learning services using the same abstractions than educators. This way, a registry would store these announcements and could retrieve the services that match the description expressed by educators.

We advocate that an ontology of learning tools for CSCL systems could address this issue. This ontology would describe service capabilities unambiguously, and in a semantically significant way for the educator; furthermore, comparing with a keyword-based search, the discovery would be enhanced by enabling inferences such as the simple subsumption of the precedent example.

Nevertheless, we are not precluding the OWL-S ontology. Indeed, both approaches can be combined and benefit each other. OWL-S could be used to describe the functionality of a service in terms of inputs, outputs, preconditions and effects. This way, the educator would only be concerned with this domain ontology, but not with OWL-S. In very specific cases OWL-S could be useful to refine a search, although its use to automate the invocation or the composition of tools seems more relevant.

3. An Ontology of Learning Tools and Activities

In order to facilitate the description and integration of learning tools to compose a CSCL learning environment, an

¹<http://www.unspsc.org>

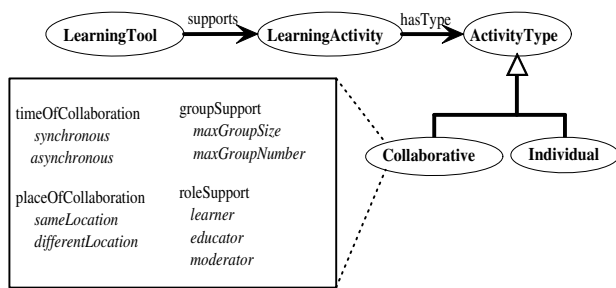


Figure 1. Information model of the ontology of learning tools.

ontology of learning tools can be used. However, the ontology should not describe specific service implementations, but rather learning tools, thus offering enough flexibility for a service provider to describe its services. In addition, the ontology should allow to describe what types of activities does a particular tool support, either individual or collaborative. This issue is expressed independently of the type of learning tool, as shown pictorially in Figure 1: e.g. a service provider offers a service that implements a text editor (tool type) that allows collaborative synchronous activities (activity type). This section illustrates how such an ontology might look like.

To perform a categorization of learning tools that can be used in a CSCL system, a first step would be the identification of such learning tools from an analysis of existing CSCL systems. In this sense, [9] provides a noteworthy study. Then, identified tool types are structured hierarchically. A snapshot of this categorization is shown in figure 2: all the tools are specializations of the `LearningTool` class, while additional subclassing serves to distinguish significant features such as the use of specific sentence openers in a structured chat.

Although deeper specialization could be achieved, it is intended not to collapse the ontology with too much detail in order to ease tool description and discovering. However, in some contexts more specialization could be advisable. For example, network simulators can be used in the laboratory of a networking course, but one could be focused on the data link layer and another on the IP layer. In this cases, it is advisable to use domain ontologies to further refine learning tools in the proposed ontologies. This can be easily done by mapping This can be the domain ontology, e.g. Network Simulator, onto the upper concept in the proposed ontology, e.g. Simulator.

It is worth mentioning that tools used in CSCL scenarios do not need to be specifically designed for CSCL systems. This way, off-the-shelf chats or text editors can be used, as well as a very specific collaborative tool of con-



Figure 2. Class hierarchy of learning tools.

ceptual maps. Moreover, a collaboration scenario usually comprises individual and collaborative activities [8]. Thus, a CSCL system might support both types of activities. It is therefore very important that the ontology allows to describe if a specific tool supports either an individual or a collaborative use, or both. This way, a text editor always supports edition activities, although a specific implementation could only allow individual edition. However, there are some prescribed relations in the ontology to reflect agreed conventions in the domain. For example, “a chat tool can only be used in synchronous collaborative activities”.

Besides, collaborative activities are complex and can be further qualified. A strongly agreed categorization is the one in [11], which distinguishes two important dimensions: time and location. With respect to time, an activity such as a debate could be synchronous, e.g. supported by a chat tool, or asynchronous, e.g. supported by a news tool. Moreover, participants in a debate could be in different locations or in the same physical place. Although in both cases a software tool can be used to support the debate, there are different requirements. For instance, providing participants awareness information is advisable in a remote setting, but not so important in a co-located scenario.

Finally, there are other significant collaboration features. Learners can be organized in groups to perform the same or different activities. Thus, a chat tool can support group debates enabling different conversations. Another important issue is that students can play different roles in a collaborative activity, e.g. a chat tool can recognize a moderator role letting her to allow or deny access to a debate.

The proposed ontology has an information model that cares for all these features, as shown in figure 1. With it,

each tool can be published to support one or more learning activities, while these activities can be refined into individual and collaborative. A collaborative activity is qualified with the characteristics extracted in the precedent discussion. Therefore, the ontology can benefit both the description of a CSCL scenario and the search of learning tools.

Concerning the description of a CSCL scenario, the ontology allows to specify the properties of the activities as well as the tools that support them, supporting novice educators by providing mappings between activities and learning tools. For example, consider an educator who intends to design an asynchronous debate activity. She selects the debate activity in the ontology and qualifies it as asynchronous. However, she does not know what learning tools can support such activity. Nevertheless, she can ask the ontology and she will find that questionnaire tools and news tools are appropriate.

Regarding the search of tools, it can be performed just selecting a specific tool in the ontology. Alternatively, this search can be restricted for a particular learning activity that could be further described. The following section describes the integration of this ontology in a real CSCL system allowing the search of learning tools.

4. Sample Search of Learning Services in Gridcole

This section introduces Gridcole, a CSCL system that combines OGSA Grid Services and scripting, and proposes changes in its architecture to support semantic searches of learning services using the ontology proposed in section 3. A sample scenario is presented to illustrate the new searches. Next, the new search capabilities are discussed as well as the proposed ontology.

4.1. Search of Services in Gridcole

Gridcole [3] is a collaborative learning system that can be easily tailored by educators in order to support their own CSCL scenarios. Learning tools can be integrated using computing services provided by third parties in the form of OGSA Grid Services. Educators employ collaboration scripts [8] in order to describe the flow of learning activities and required tools and documents to support the activities. Gridcole embodies a script player that interprets collaboration scripts and schedules the activities to be performed by each user.

Very briefly, the system outline is as follows. educators can store collaboration scripts in Gridcole. An educator can select one of them to execute the collaboration scenario described in the script. Then, Gridcole interprets the script and looks for suitable tools supplied by third-party providers in an OGSA-based grid according to the description in the

script. Next, the educator can select the most suitable ones among the retrieved services. To finish the setting-up, the educator provides the list of participants and the roles they play. Then, the educator can launch the application and students can join the execution.

To perform the search, Gridcole uses a proprietary registry named Learning Object Index Service (LOIS) [3]. Service providers can access an LOIS and register their tools offered as Grid Services. In a collaboration script, educators specify some keywords to describe the required tools. Therefore, Gridcole interprets the script and queries all LOISs known by the system using these keywords. A LOIS retrieves those services that have been registered with a superset of those keywords.

It could be argued why Gridcole does not use a standard registry, such as the Index Service. However, the Index Service is significantly complex due to soft-state registration and the use of SDEs. Besides, service search capabilities are not better, as discussed in section 2.1.

Anyway, this simplicity hinders the search of learning services. Indeed, providers and requesters have not previously agreed the vocabulary employed to describe learning services, e.g. “chat” and “conference tool”. Different languages dramatically enhance this problem. Furthermore, services are poorly described using only keywords .

Therefore, an ontology as the one described in section 3 could be used to agree semantics in tool descriptions, while enabling rich search capabilities. Providers could announce their services using the abstractions in the ontology, while educators could search the most suitable tools that meet their criteria.

An ontology-enabled registry could be integrated in the infrastructure of Gridcole to perform the search of services. This registry would understand the ontology of learning tools in order to enforce providers to accomplish their announcements with this ontology. Providers would access Gridcole and could announce their services in the registry. This way, such modified registry would replace LOIS functionality. This is the only required change in the architecture, although new interfaces should be implemented to constrain providers to the ontology concepts and to support educators to perform semantic searches. Such a registry would store service announcements describing the learning tools that they implement and the activities that they support. Besides, technical information such as contact information or Grid Service Handles (GSHs) [12] to invoke them should be provided. Service announcements could correspond to grid service factories that might create instances of grid services when executing the learning scenario. For instance, table 1 shows some service announcements describing the tools that implement as well as the supported learning activities, according to the abstractions provided in the proposed ontology. Technical information of services is

Table 2. Description of a sample collaborative learning scenario.

Activity	Educator's Description
First Activity	Tool: DocumentViewer Activity Name: DocVisualization
Second Activity	Tool: LearningTool Activity Name: Debate Activity Type: Collaborative Activity Group Support: at least 4 members
Third Activity	Tool: TextEditor Activity Name: Edition Activity Type: Collaborative Activity Time: Synchronous Activity Group Support: at least 4 members

not shown in the table, since initially it is not relevant for service discovery.

4.2. Sample Search Scenario

In order to illustrate the search capabilities of this modified Gridcole architecture, a collaborative learning scenario is proposed. Consider a design that comprises three learning activities. The first is performed individually and consists on the reading of some documents. Students are previously organized in groups of four individuals and each member of a group reads a different document. Next, group components are engaged into a synchronous debate to explain each other what they have read. Finally, each group collaboratively edits a report exposing conclusions achieved.

The information that the educator can provide to deploy such scenario is shown in table 2 and can be formalized in a script. The description in table 2 references concepts described in the proposed ontology. Then, an educator can instruct Gridcole to execute it. Gridcole's script player can interpret the script and perform the discovery of learning services that meet the specified requirements. For the first activity, the registry in table 1 will retrieve the service `browser1` since it implements a `DocumentViewer` and it supports a document visualization activity. In this case, the registry infers that a `Browser` is a subclass of a `DocumentViewer`, as described in the ontology (see figure 2). Besides, it can provide an explanation such as "you were asking for a `DocumentViewer`, and `browser1` is a special type that can visualize html documents". This service will suffice if the educator only provides html documents.

For the second activity, the registry will find services `quest` and `chat1`, since both meet the requirements spec-

ified in the script. The educator can further refine the query and restrict the debate activity to be synchronous, selecting `chat1`. In addition, the educator could consider the use of a conceptual map tool to enable students to model the abstractions discussed in the debate. Therefore, she could ask the registry for a conceptual map tool that, at least, supports four member groups. Service `maptool` fulfils these requirements and could be integrated in Gridcole, together with `chat1`.

Finally, the registry will search for text editors that allow synchronous collaborative edition and support at least four member groups. The registry will retrieve service `editorPro` and the educator could decide that this service fits the intended activity.

Once the educator has selected appropriate services, she may start the execution of the educational scenario in Gridcole after providing a list of participants which are allowed to join it.

4.3. Discussion

As shown, the search mechanism is flexible enough to allow a wide range of queries in order to meet educators' needs. Moreover, explicit and clear semantics formalized in the ontology enables a robust retrieval mechanism. Shared concepts and relationships in the ontology allows for semantic interoperation among service providers and requesters, since providers use the same abstractions to describe their services than requesters use to query. Besides, some kind of reasoning can be inferred, such as the fact that a browser is a specialization of a document viewer. In contrast, keyword matches seriously limit interoperation if there is not a previous agreement in vocabulary. Furthermore, expressiveness is rather limited just using keywords, e.g. it is not possible to query for services that support groups composed of four or more members.

With respect to the proposed ontology, it was conceived to model the abstractions that educators use to describe their learning activities. This user-centric conception enables educators to perform the search of learning services. However, the proposed ontology is not exhaustive and can be extended. For instance, the `SpecificHardwareResource` learning tool (see figure 2) is a very broad class that can be further refined to describe very specific tools such as an oscilloscope. Anyway, many tools are domain-dependent and too much subclassing can bloat the ontology. Therefore, new domain-specific ontologies that extend this one can be developed to suit particular educational scenarios.

It is worth mentioning that the proposed ontology is not only valid for the search of services within Gridcole. Since the ontology does not deal with technology abstractions, it is not constrained to a particular implementation and can be

Table 1. Learning service announcements in an ontology-enabled registry.

Service	Tool	Activity					
		Name	Type	Time	Location	Role Support	Group Support
notepad1	TextEditor	Edition	Individual				
editorPro	TextEditor	Edition Edition	Individual Collaborative	Synch.	Same, Distance	Learner	1 group, any size
browser1	Browser	DocVisualization	Individual				
quest	Questionnaire	Assessment Debate	Collaborative Collaborative	Asynch. Asynch.	Distance Distance	Learner, Educator Learner, Educator	20 groups, any size 20 groups, any size
chat1	FreeChat	Debate	Collaborative	Synch.	Same, Distance	Learner	20 groups, any size
maptool	ConceptualMap	Modelling Modelling	Individual Collaborative	Synch.	Same, Distance	Learner	1 group, 5 members

used in a CSCL system that integrates Web Services, for example. Furthermore, this ontology can be used to model CL scenarios using provided abstractions to describe collaborative or individual activities and non-software tools, e.g. a microscope, can be specified to support them, although the ontology should probably be extended, as discussed before.

An important concern is what architecture modifications are needed to enable the semantic searches described here. In a service oriented architecture, it implies changes in the registry, as outlined in section 4.1 with Gridcole. This way, the new registry should understand the ontology to enforce providers to adjust their services to the concepts in the ontology, while enabling described semantic searches. An off-the-shelf reasoner can be used to manage these service announcements and support querying. New interfaces are needed to interact with the modified ontology-enabled registry. Finally, it is worth considering that other registry models are possible, since the proposed ontology is not tied to any. The Index Service and UDDI assumes a registry of service announcements that are subsequently matched against requests. Another possibility could be a registry of service requests instead of service announcements. Moreover, in a P2P architecture there would be no registry at all and each node could ask other nodes for services.

5. Conclusions and Future Work

This paper proposed to use an ontology of learning tools to enable semantic searches of learning services in CSCL systems. The use of ontologies to formalize knowledge provides clear and unambiguous semantics. Knowledge captured in an ontology can be easily shared and understood not only by humans, but by software agents.

Therefore, it has been shown how a real CSCL system could employ this ontology to enhance the search of learning tools. Using an ontology-enabled registry, providers can announce their services described in terms of the on-

tology. The registry classifies the announcements and responds to the requests posed by educators. Comparing to other alternatives, a significant advantage is that educators can search learning tools using domain concepts. Besides, some kind of reasoning can be inferred to deduce if a service announcement complies with the parameters of a request.

In the proposed ontology, learning tools are related with the learning activities that they can support. Indeed, the same activity can change depending on the context, and therefore a particular learning tool might not support that activity in all contexts. This way, the search for services that implement a learning tool can be restricted specifying the context of an activity. In particular, collaboration activities require learning tools with special capabilities, such as support for different roles or synchronous interaction.

With respect to the grid, learning systems are beginning to use this infrastructure to support educational scenarios. In the particular case of CSCL systems, there are many resource demanding activities that can take advantage of the grid. For instance, the collaborative edition of a journal among several schools could require an editor that allows the edition of twenty articles simultaneously, each one being synchronously edited by a group of five members. Such an editor could be offered in a grid and easily discovered if described in terms of the tool ontology. Besides, the grid enables the use of specific hardware resources, such as a telescope. The tool ontology should embody these new tools both to allow the description of new collaboration scripts and to allow the search of the new available tools.

Upcoming work in near future includes the evaluation of the search mechanism in a real environment. With this aim, we plan to complete an ontology-enabled registry that providers would use to publish their service announcements, while educators would use to query learning tools. This registry will be integrated in the Gridcole system.

In addition, OWL-S could be used to provide a low-level description of learning services, allowing the automatic in-

vocation and composition of services. Such description could be related with the proposed ontology extending it with the specification of inputs and outputs of a tool.

Finally, other information could be employed to further refine a service search. For instance, quality of service parameters can be modelled in an ontology to annotate services and decide the selection of a service according to pre-defined quality of service. Furthermore, user ratings of a service could be used to give a measure of reputation.

Acknowledgements

This work has been partially funded by Spanish Ministry of Science and Technology project TIC-2002-04258-C03-02 and the European Commission e-learning project EAC/61/03/GR009.

References

- [1] H. H. Adelsberger, B. Collis, and J. M. Pawlowski, editors. *Handbook of Information Technologies for Education and Training*. Springer-Verlag, Berlín, Alemania, 2002.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
- [3] M. Bote-Lorenzo, L. Vaquero-González, G. Vega-Gorgojo, Y. Dimitriadis, J. Asensio-Pérez, E. Gómez-Sánchez, and D. Hernández-Leo. A tailorable collaborative learning system that combines OGSA grid services and IMS-LD scripting. In *Proc. of the Tenth International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2004)*, LNCS 3198, pages 305–321, San Carlos, Costa Rica, Sept. 2004. Springer-Verlag.
- [4] B. Chandrasekaran, J. Josephson, and V. Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):72–81, 1999.
- [5] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the web services web. *IEEE Internet Computing*, 6(2):86–93, Mar. 2002.
- [6] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *Proc. of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01)*, pages 181–194, San Francisco, CA, USA, 2001.
- [7] D. de Roure, N. Jennings, and N. Shadbolt. The semantic grid: a future e-science infrastructure. In F. Berman, G. Fox, and A. Hey, editors, *Grid computing: making the global infrastructure a reality*, pages 437–469. John Wiley and Sons, Chichester, United Kingdom, 2003.
- [8] P. Dillenbourg. Over-scripting CSCL: the risks of blending collaborative learning with instructional design. In P. Kirschner, editor, *Three worlds of CSCL. Can we support CSCL?*, pages 61–91. Open Universiteit Nederland, Heerlen, 2002.
- [9] A. Dimitracopoulou and A. Petrou. Advanced collaborative learning systems for young students: Design issues and current trends on new cognitive and metacognitive tools. *THEMES in Education International Journal (in press)*, 2005.
- [10] ELeGi. European learning grid infrastructure. URL: <http://www.elegi.org>, last visited December 2004.
- [11] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some issues and experiences. *Comm. of the ACM*, 34(1):38–58, 1991.
- [12] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The physiology of the grid. In F. Berman, G. Fox, and A. Hey, editors, *Grid computing: making the global infrastructure a reality*, pages 217–249. John Wiley and Sons, Chichester, United Kingdom, 2003.
- [13] G. Fox. Education and the enterprise with the grid. In F. Berman, G. Fox, and A. Hey, editors, *Grid computing: making the global infrastructure a reality*, pages 963–976. John Wiley and Sons, Chichester, United Kingdom, 2003.
- [14] Globus. The globus project. URL: <http://www.globus.org>, last visited December 2004.
- [15] T. Kawamura, J. D. Blasio, T. Hasegawa, M. Paolucci, and K. Sycara. Preliminary report of public experiment of semantic service matchmaker with UDDI business registry. In *Proc. Of First International Conference on Service-Oriented Computing (ICSOC2003)*, Trento, Italy, Dec. 2003.
- [16] T. Koschmann. Paradigm shift and instructional technology. In T. Koschmann, editor, *CSCL: Theory and Practice of an emerging paradigm*, pages 1–23. Lawrence Erlbaum, Mahwah, NJ, USA, 1996.
- [17] D. Martin et al. Bringing semantics to web services: The OWL-S approach. In *Proc. of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, CA, USA, July 2004.
- [18] D. Martin et al. OWL-S: Semantic markup for web services (version 1.1). Technical report, DARPA Agent Markup Language Program, 2004.
- [19] A. Martínez-Monés. *Method and model to computationally support evaluation in CSCL*. PhD thesis, School of Computer Engineering, University of Valladolid, Spain, 2003. In Spanish.
- [20] Organization for the Advancement of Structured Information Standards (OASIS). Introduction to UDDI: Important features and functional concepts, Oct. 2004. URL: <http://www.uddi.org>, last visited December 2004..
- [21] M. Paolucci and K. Sycara. Autonomous semantic web services. *IEEE Internet Computing*, 7(5):34–41, Oct. 2003.
- [22] M. Papazoglou and D. Georgakopoulos. Service-oriented computing. *Comm. of the ACM*, 46(10):25–28, Oct. 2003.
- [23] R. Prodan and T. Fahringer. From web services to OGSA: Experiences in implementing an OGSA-based grid application. In *Proc. of the 4th International Workshop of Grid Computing (Grid 2003)*, Phoenix, AZ, USA, Nov. 2003.
- [24] M. Ramamurthy, R. Wilhelmson, R. Pea, M. Louis, and D. Edelson. CoVis: A national science education collaborative. In *Proc. of the American Meteorological Society 4th Conference on Education*, Dallas, TX, USA, 1995.
- [25] J. Roschelle, C. DiGiano, M. Koutis, A. Repenning, J. Phillips, N. Jackiw, and D. Suthers. Developing educational software components. *IEEE Computer*, 32(9):50–58, Sept. 1999.
- [26] J. Waldo. Alive and well: Jini technology today. *IEEE Computer*, 33(6):107–109, June 2000.