A Semantic Approach to Discovering Learning Services in Grid-Based Collaborative Systems

Guillermo Vega, Yannis Dimitriadis, Juan I. Asensio-Perez, Eduardo Gomez-Sanchez, Miguel Luis Bote-Lorenzo

A Semantic Approach to Discovering Learning Services in Grid-Based Collaborative Systems

Guillermo Vega-Gorgojo, Miguel L. Bote-Lorenzo,
Eduardo Gómez-Sánchez, Yannis A. Dimitriadis and
Juan I. Asensio-Pérez

School of Telecommunications Engineering, University of Valladolid. Camino del Cementerio s/n, 47011 Valladolid, Spain

Abstract

CSCL systems can benefit from using grids since they offer a common infrastructure enabling the access to 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 offered by third party providers. However, service discovery is a challenge for educators, since they cannot use their own domain abstractions to search for learning services that may support their educational settings. Common service discovery mechanisms, such as the Index Service or UDDI, provide limited discovery capabilities since they rely on keyword matching and cannot deal with the description of service properties. In order to address these drawbacks, formal semantics of ontologies can be employed to represent semantic descriptions of services that can be exploited for service discovery. This paper proposes an ontology of CSCL tools that uses meaningful learning abstractions to describe them. That ontology is the basis of a service discovery facility that is developed for allowing educators to search service-based CSCL tools using learning concepts.

Key words: Semantic Web Services, Service Discovery, Learning Services, CSCL Systems

1 Introduction

Computer Supported Collaborative Learning (CSCL) [24] is a mature research field of increasing interest in recent years. CSCL promotes the use of information and communication technologies to support collaborative (besides individual) learning activities. Recently, the *grid* has been proposed to support

educational settings [15], enabling organizations such as schools or universities to seamlessly federate and share their resources in a grid. Accordingly, some research communities have begun to use grid infrastructures to develop e-learning systems, such as the ELeGi project [13]. In the CSCL field there is also a growing interest on using grids, since they allow the delivery of large amounts of computational power and the use of applications requiring specific hardware resources. For instance, CoVis [36] is a collaborative system for geosciences that requires supercomputing capabilities and could be benefited from using grids. Another example is PEARL [30], a remote electronics laboratory for CSCL activities that uses specific hardware resources (signal generators, oscilloscopes and microcontrollers among others) that could be seamlessly integrated in a grid infrastructure.

Although the grid can provide a significant amount of computational, software, data and hardware resources for CSCL, developing CSCL applications is a demanding task that implies a considerable effort. As [37] claims, this development can only be justified if the resulting applications 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. This issue can be tackled with the *service oriented computing* model [35], since services represent high-level abstractions that are close to educators' mental model and thus enable and promote educational software reuse in the integration of learning environments [3]. In this sense, the *Open Grid Services Architecture* (OGSA) [14] is the emerging architecture for the construction of grids following the service oriented computing paradigm. CSCL can embrace OGSA to exploit both resource availability in a grid and flexibility of services, as it occurs for example in the CSCL system Gridcole [4]. Within this context, CSCL solutions are envisioned to be created by the discovery and later integration of learning services that satisfy educator requirements for every particular collaborative learning scenario.

An important issue is how to discover appropriate services in a grid to support CSCL activities. Since educators are the intended users, they should be able to express their preferences of learning services to suit their needs for a specific educational setting. Therefore, a service discovery facility for a CSCL system should allow educators to formulate those queries by using concepts of the learning domain, i.e. not technological.

Since service discovery is a known issue in service oriented architectures, some mechanisms already exist for providing this functionality. Service registries are widely used to publish metadata about services, while consumers can find services that meet their criteria. Some well-known registries are the Universal, Description, Discovery and Integration (UDDI) standard [33] in the popular Web Services architecture [8] or the Index Service in Globus Toolkit 3 [16].

However, the discovery mechanism of such registries is based on keyword matching [21] and they do not provide capability based search [28], hindering service discovery. This is important for service-based CSCL tools to allow searching for asynchronous services, for example. Alternatively, services could be enriched with semantic information allowing enhanced discovery mechanisms. This is the approach followed in [34], with the so-called Semantic Web Services. Ontologies [5] are used to explicitly formalize knowledge, enabling rich descriptions and robust information retrieval. The Semantic Grid initiative [9] is consistent with this vision to construct third generation grids using semantics to enhance the discovery of resources in a grid, among other capabilities.

Therefore, semantics can be considered to improve service discovery, while enabling users to state high-level expressive queries. To realize such a system a formalization of the domain knowledge into an ontology is needed. This paper proposes an ontology which describes CSCL services. Such an ontology can be used to semantically annotate learning services. The resulting knowledge base can be exploited by a search facility in order to perform the service discovery. A preliminary version of this facility has been integrated in Gridcole [4], a grid-based CSCL system, and its operation is illustrated in this paper with an example.

This proposal is conceived to be used in a single virtual organization integrated by multiple providers and consumers of CSCL services within the context of a grid. In such a scenario, it is feasible to assume a common ontology, such as the one described in this paper. It may be shared by all participants in order to enable the search of services within the virtual organization. This implies that in a more heterogeneous environment involving multiple virtual organizations in a grid, each one may employ different ontologies and techniques such as ontology alignment, should be considered to enable interoperability among the virtual organizations. However, the research of this interoperability issue is out of the scope of this paper.

The rest of the document is organized as follows: in section 2 requirements for the discovery of learning services are discussed. Besides, current service discovery mechanisms are analysed as well as semantic approaches for the case of a CSCL system. Section 3 proposes an ontology to describe learning services, enabling their discovery. In section 4 a new service discovery infrastructure for the real CSCL system Gridcole is proposed using the ontology depicted in section 3. Finally, the main conclusions of the study are shown as well as current research work.

2 Alternatives for the Discovery of Learning Services

This section first deals with the requirements for service discovery in the context of service-based CSCL systems. Some of the most relevant service discovery mechanisms already available in common service architectures are analysed. Next, the use of semantics for service discovery is discussed, as well as a proposal for the discovery of learning services.

2.1 Requirements for the Discovery of Learning Services

In order to use a CSCL system, educators have to properly arrange the system to support their particular learning scenarios. In service-based CSCL systems, service discovery is key to find appropriate services to support the collaborative and non-collaborative learning activities. In a typical discovery process, educators submit a query and the search facility retrieves available services that meet their criteria. Therefore, it is critical to properly support educators during the discovery process and ease their interaction with the search facility. To achieve this, they should employ their own learning abstractions to describe desired CSCL tools, hence avoiding technical details. Besides, allowed queries should be expressive enough to cope with the complexity of CSCL tools and CSCL educational settings.

As reflected in [24], CSCL applications assume a variety of forms and can be employed in very different scenarios. Some reviews of CSCL tools can be found in the literature such as [23], [6], [25] or [12]. For instance, an extensive work within a European project reported in [25] provides a comprehensive classification that is outlined next in order to show the various kinds of tools that can be discovered in the context of a service-oriented CSCL system. The first category comprises *conventional single-user programs reapplied in a collaborative context* such as spreadsheets, simulators or multimedia authoring tools. While these tools can be used in situations different from collaborative learning, there are other *single-user applications with special interfaces for facilitating collaboration*. These are specially planned to support collaborative activities, mainly through a graphical representation of the students' own activity path that can be applied for collaborative reflection. Both types of tools are conceived to interact *around* computers, but it is also possible to design applications to interact *through* computers [7]. In the latter case, tools enable students' collaboration by supporting *communication*, *joint activities* and *knowledge sharing*, as also reported in [23]. Some examples are e-mail, chat and news for communication; collaborative editors and collaborative argumentation tools for joint activities; shared repositories for knowledge sharing.

4

In addition to this classification of CSCL tools, it is important to take into account which queries should be expressed by domain users in order to properly support the search of CSCL services. With this objective, we have carried out a non-exhaustive preliminary study with some practitioners in our University. More specifically, eight CSCL practitioners in education, computer science and telecommunications engineering posed three queries each for learning tools. Representative examples of their queries are:

- *I want a chat.*
- *I want a TCP/IP simulator for a course on computer networks.*
- *I want a tool for the edition of a .doc formatted document by a group of four members.*
- *I want a tool for document exchange.*
- *I want a tool to support asynchronous debates among twenty participants.*

One insight of these queries is that they are expressed in terms of learning domain concepts, reinforcing the need of using these abstractions to describe learning services. Besides, educators provide their own perception of CSCL tools, pointing out significant properties. Thus, educators asks for specific CSCL tool types (e.g. *chat, simulator*), although sometimes they further qualify these queries by providing additional capabilities. In this sense, the functionality of a tool is described in terms of the learning tasks that can support (e.g *edition, information sharing, debate*). Other characteristics refer to group sizes (e.g. *four members*), document types (e.g. *.doc formatted document*) or supported collaborative interactions (e.g. *asynchronous*).

It is also remarkable that sometimes educators just describe the learning task that they want to perform, as in the last three queries. Another finding is that some tools are tied to a very specific domain in the curricula (e.g. *TCP/IP, computer networks*).

2.2 A Review of Service Discovery Mechanisms in Service-Oriented Architectures

After stating the requirements for the discovery of learning services, we can analyse the adequacy of common mechanisms for service discovery. UDDI [33] is a method for publishing and discovering services. Although the UDDI standard was proposed for the popular Web Services [8] architecture, UDDI can also be used in other service architectures, such as OGSA [31].

Basically, UDDI is a registry of businesses and Web Services. An organization can publish in a UDDI registry information about the actual organization as well as the services it offers. When used to perform a service search, UDDI only allows querying for `service name` (published by the service provider), `key`

`reference` (which must be unique for a service) and `category-bag` (which lists all the business categories that register a service).

As reported in [28], an important shortcoming of UDDI is that it does not represent service capabilities. As a consequence, UDDI does not support searches on the properties of a service. Another drawback is that the search is done by string matching, resulting in a very weak discovery mechanism. For example, querying for a 'questionnaire tool' would not find a service published as an 'assessment tool'.

In grid contexts other registries such as the Index Service in Globus Toolkit 3 (GT3) [16] can also be used. The main difference with UDDI is that this registry 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. However, we only deal with service discovery in this paper. Therefore, the Index Service aggregates service data elements (SDEs) [14] of available grid services and allows a limited set of queries: by specifying the Grid Service Handler (GSH) [14] of a known grid service or by specifying an SDE name. Since a service provider can give any name for GSHs or SDEs, it is extremely difficult to discover an unknown service using this registry.

Another popular service architecture is Jini [40]. Jini is a Java-centered service architecture; the Lookup Service is part of Jini's infrastructure and allows the service discovery. Publication of services also supports soft state registration. 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. This approach is based on the strong assumption of a shared agreement about service names among providers and consumers.

In summary, these discovery mechanisms do not allow querying for CSCL tool properties, since none of them can properly describe service capabilities. Besides, keyword-based matching performs poorly in the discovery, and allows just simple queries based on the presence or absence of a word, but definitely with restricted expressiveness compared to the complex queries proposed by practitioners in section 2.1.

2.3   Semantic Approaches for Service Discovery

The aforementioned problems can be lessened by the use of semantics. Indeed, semantics is driving the next generation of the Web, known as the Semantic Web [2], which enables automated approaches to exploit Web resources. In this vision, ontologies [5] constitute a key technique that allows the explicit

description of knowledge in a formal language that is computer-interpretable.

According to [38], three forms of semantics can be distinguished. Ontologies correspond to the most formal type that can be used to represent information with definite associated semantic interpretations. Representation mechanisms with formal and explicit semantics make them more machine-processable and afford more intelligent applications. However, the trade-off involves the complexity of developing an ontology as well as reaching an agreement on the knowledge modelled in the ontology.

In this sense, the so-called Semantic Web Services [34] use ontologies to semantically annotate Web Services in order to achieve automated discovery, invocation and composition. To accomplish this, the OWL-S [29] (formerly DAML-S) ontology of services has been developed. Using this ontology, services can be annotated with semantic metadata to represent their capabilities and this meaningful information can be used for the discovery of services.

The functionality of a service is described with OWL-S in terms of the transformation produced by the service. Specifically, it defines the required inputs and the generated outputs; furthermore, it also defines the preconditions required to invoke a service, as well as the expected effects that result from the execution. 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. Note that the elements `StudentListing` or `GroupSize` refer to entries in some ontology external to OWL-S.

However, it seems uncomfortable for an educator to ask for a learning service in such a way. This becomes even worse when querying for CSCL services with complex service interfaces. It is not feasible that an educator can describe the supported operations in terms of inputs and outputs. Furthermore, OWL-S is focused on the description of data-oriented services, as in e-commerce, that emphasize the information transformation process. On the other hand, many CSCL applications are interactive and presentation-oriented, so there are other important features in a CSCL tool besides inputs and outputs.

These concerns are consistent with some difficulties outlined in [27] for the discovery of Semantic Web services in bioinformatics. Specifically, [27] detects the need to provide better service descriptions than required inputs and outputs. It also claims that describing services is demanding, even with simplified service interfaces. Moreover, it suggests that semantic service discovery should be tailored towards the users' notions of services.

Table 1
Comparison of approaches for the discovery of learning services.

	Adequacy and expressiveness for educators	CSCL tool properties
UDDI, Index Service, Jini	Simple Low expressiveness Poor search capabilities	Not defined
OWL-S	Not very suitable Low level information transformation view	Limited to inputs, outputs, preconditions, effects
Ontology of learning tools	Educational abstractions High expressiveness	User types, tasks, groups, collaborative interactions

To meet these requisites, we advocate the use of a specific ontology describing capabilities of CSCL learning tools using concepts understandable by educators. This way, the discovery of learning services will be eased for educators since they could express a query using high-level learning abstractions to describe required services. Moreover, significant capabilities of CSCL tools, such as group support or collaboration properties, can be modelled since they determine the suitability of a CSCL tool for a particular learning scenario.

A comparison of the different approaches analysed in sections 2.2 and 2.3 is shown in table 1. The first row corresponds to analysed syntactical approaches that rely on keyword matching, achieving low expressiveness and weak service discovery capabilities. While the use of semantics tackles these problems, the OWL-S ontology is not well suited for a CSCL context, as discussed before. Therefore, we propose an ontology of CSCL tools that can be used for service discovery and that is described in the following section.

3 An Ontology of CSCL Learning Tools

In this section we describe an ontology developed with the objective of facilitating the discovery of learning services, as discussed in the precedent section. While other ontologies could be possible in the general domain of CSCL, we propose a feasible and suitable one for this specific objective.

3.1 Description of the Ontology

Since developing an ontology is a time-consuming and difficult task, it is worth considering reusing existing sources for the description of CSCL tools. In the e-learning field there are some efforts to annotate learning resources with metadata. In this sense, the IEEE LOM standard [19] is perhaps the most remarkable initiative. However, LOM is not specially suited to describe learning tools but learning contents, as discussed in [39]. Alternatively, the IMS Learning Design (IMS-LD) specification [20] is an educational modelling language that can be used to describe collaboration scripts. A collaboration script comprises a flow of activities that can be performed individually or collaborativelly. Although IMS-LD has some limitations to describe collaboration scripts [39], learning abstractions modelled in this specification have served as a basis to construct the proposed ontology.

Technically, the proposed ontology of CSCL learning tools has been formalized using the ontology language OWL [1]. OWL is a widespread, expressive language that allows the use of external reasoners to compute the consistency of the model, classify the ontology, query the model and retrieve individuals described in terms of the ontology. The ontology is intended to describe the educational aspects of CSCL tools to enable their discovery by educators. Thus, it does not define technological or implementation details. As a result, it can be used to annotate any type of CSCL tool, implemented as a grid service, a web service, a collection of Java classes or whatever. In the case of services, they can be searched using the educational abstractions in the ontology, while additional technical metadata could be provided, for example using the OWL-S ontology.

The ontology relies on a conceptual model shown pictorially in figure 1. The structure of a learning scenario with corresponding objectives, activities, roles and resources is reminiscent of the IMS-LD conceptual model [20, figure 3.1, p. 21]. However, the proposed model further details the properties of `Learning Tools`, since this is the core concept.

In order to facilitate the discovery of learning tools, a high level functional description is provided. Thus, a `Learning Tool` is defined as a software tool that can be used to perform one or more `Tasks` that eventually lead to learning. Indeed, educators required support for this type of query (see Section 2.1) and a similar approach was pursued in [42] to define the tasks that can be accomplished with a service in the bioinformatics domain. To identify the tasks that can be performed with a tool we have revised the CSCL literature. [24] identifies the uses that may be served by a tool: *presentation, support the creation of representational formalisms, computation, mediate communication* and *provide archival storage*. Since we intend to show the point of
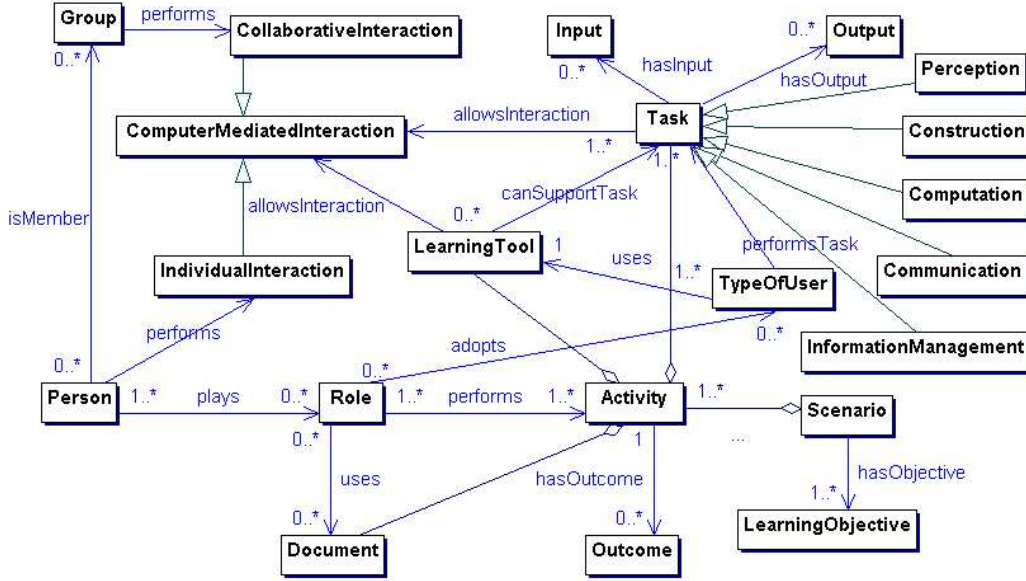
9

Fig. 1. Simplified conceptual model of the ontology of CSCL tools.

view of the actor that performs the task in order to facilitate the discovery, these terms have been reified to `Perception`, `Construction`, `Computation`, `Communication` and `Information Management`, as shown in figure 1. These tasks can be further specialized, such as `Edition` as a specific type of construction task. Other authors [23,25] identify three tasks, *joint activities*, *communication* and *knowledge sharing*, that can be interpreted as a simplified view of the former model.

Other aspects of the ontology such as *roles* and *groups* are commonly adopted in collaborative learning [10]. In this sense, different `Types of Users` may use a tool, perhaps with limited access to its functionality depending on their `Role`. Thus, each type of user is allowed to perform a restricted subset of tasks. For instance, the administrator of a `Repository` can manage the users, but cannot upload or download `Documents` of the repository. A key characteristic of CSCL tools is how users interact through the tool. While a learner can interact `Individually` with a tool, a `Group` of learners can collaborate to perform a task mediated by a `Collaborative Tool`, such as a group drawing with a `Whiteboard`. Moreover, distinction on *synchronous* and *asynchronous* collaborative interactions is usually considered in the CSCL literature as, for example, in [12].

All these aspects are defined in the ontology using OWL constructs. Concepts defined are not merely labels, but formal definitions. For instance, the concept `Document Editor` is defined as any `Learning Tool` that supports `Edition` tasks of `Documents`. Another example is the concept `Synchronous Tool` which refers to any `Learning Tool` that allows `type:synchronous Collaborative Interaction`. With this modelled knowledge, a reasoner can classify the on-
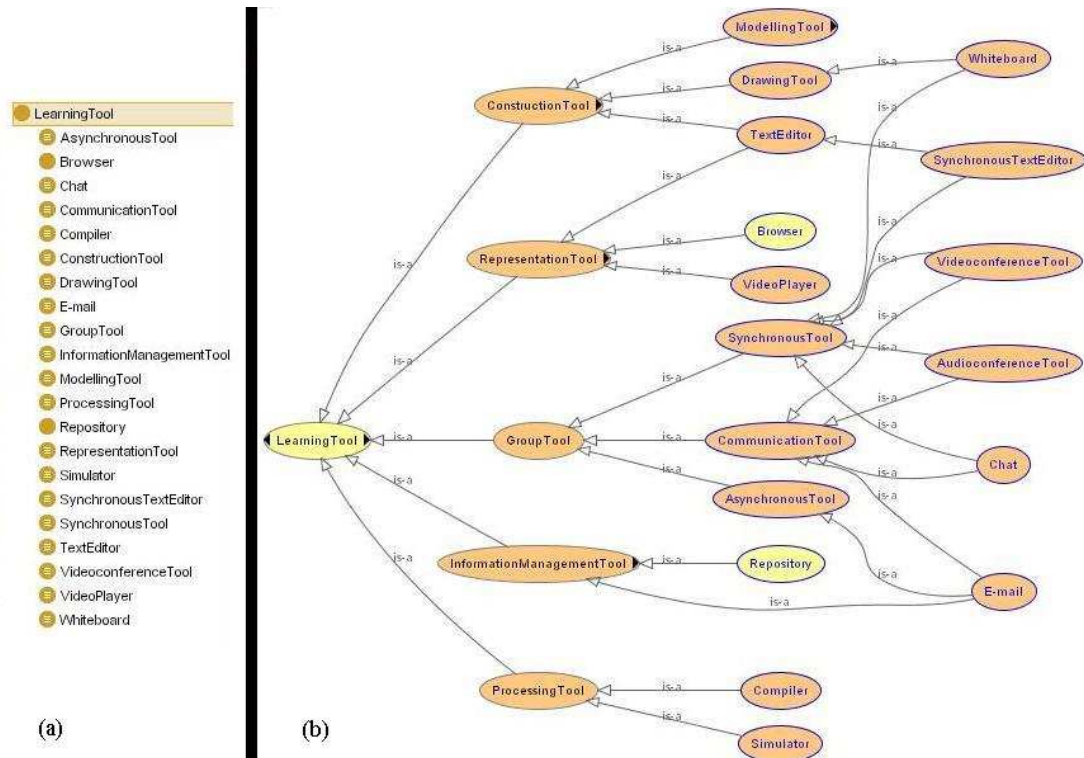
Fig. 2. Excerpt of the hierarchy of learning tool types before (a) and after reasoning has ocurred (b).

tology and infer new knowledge that was not stated explicitly. Thus, figure 2a shows an excerpt of the asserted learning tools in a completely flat classification before reasoning. Then, the Racer reasoner [18] classifies these tools as shown in figure 2b using the definitions of these concepts. In the first level, tools are categorized by the tasks they can perform, such as `Construction Tools` or `Representation Tools`. Since a `Communication Tool` is always operated by a group, it subsumes `Group Tool`. In the second level, tools are more specific, such as `Browser` or `Video Player`. Finally, concepts to the right are more complex, since they define collaboration properties. For instance, `E-mail` allows asynchronous communication among participants, but it also provides storage of messages and it is further refined as an `Information Management Tool`.

Once the ontology is formalized, it is possible to describe instances of CSCL tools in terms of the ontology abstractions. For example, figure 3 shows the resulting annotation of the `acmeditor` text editor. Two types of users are defined; the `viewer` can only visualize documents, while the `editor` is also allowed to edit documents. These learning tasks are further refined by specifying supported inputs and outputs as well as collaboration capabilities. Both tasks can be performed individually or collaboratively through this tool. In the latter case, interactions are synchronous among groups of less than six individuals. Finally, additional metadata such as technical information can be

11

TOOL NAME

```
acmeditor
```

TOOL DESCRIPTION

```
A word text editor that allows the synchronous edition of .doc
documents by groups of at most 5 individuals
```

USER TYPES

- `editor`
  - `Allowed tasks: DocumentViewing, Edition`
- `viewer`
  - `Allowed tasks: DocumentViewing`

TASKS

`DocumentViewing`	`Edition`
`input: .doc files` `computer mediated interaction` `individual` `collaborative` `synchronous` `group size: ≤ 5`	`input:  .doc files` `output: .doc files` `computer mediated interaction` `individual` `collaborative` `synchronous` `group size: ≤ 5`

TECHNICAL DETAILS

```
Implementation technology: grid service        Provider: GSIC
```

Fig. 3. Resulting annotation of the tool `acmeditor` using the proposed ontology.

provided (using OWL-S, for example), although not covered in this ontology. With such a description, the reasoner can check that this instance definition is consistent with the ontology and it can deduce that `acmeditor` belongs to concept `Synchronous Text Editor` since:

(1) `acmeditor` supports `Edition` ⇒ `acmeditor` is a `DocumentEditor`.
(2) `.doc` is a `TextDocument` ⇒ `acmeditor` is a `TextEditor`.
(3) `acmeditor` allows `type:synchronous CollaborativeInteraction` ⇒ `acmeditor` is a `SynchronousTextEditor`.

3.2   Ontology Discussion

The scope of the proposed ontology is limited to support the discovery of CSCL tools by educators. In this sense, the approach has been to develop a seed ontology that can be exploited by initial users of the system with sufficient coverage for this problem, including the questions posed by educators in section 2.1. Decoupling the description of CSCL tools from implementations allows the use of this ontology in many scenarios: discovery of learning services

(in a grid context or others), serving as a guide of learning tools for CSCL practitioners or authoring of collaboration scripts [39].

Moreover, it has been designed to facilitate its evolution to include new contributions if required by the community. To support this extensibility objective, the skeleton of the ontology has been cleanly defined in order to easily include a new definition of tool or a specialization of an existing element. Of course, users may also demand other types of queries, not supported by the proposed ontology, as e.g. questions related to group window properties or coordination policies in synchronous applications. In this case, the ontology could be extended with new modules that cover other specific features.

Lastly, it is important to validate that this ontology can accomplish the discovery of learning systems. In the literature, many works evaluate ontologies in terms of consistency of inference, lack of redundancy and lack of errors [17]. However, others have argued that the only way to evaluate an ontology is to assess its use in applications [32]. We have taken into account both approaches to validate the proposed ontology. The aim of the former approach is to prevent inconsistent or incorrect data, verifying that the ontology captures intended domain knowledge. Thus, concept satisfiability and consistency has been derived using the Racer reasoner to check logically inconsistent classes. Furthermore, we have revised and corrected any unexpected implicit subsumption relationship of the inferred classification.

The latter approach is covered in the next section, illustrating the use of a service discovery facility that employs the proposed ontology in a real grid-based CSCL system.

4 An Ontology-Enabled Service Discovery Infrastructure for CSCL Systems

This section briefly describes Gridcole [4], a CSCL system that combines OGSA Grid Services and IMS-LD scripts, and proposes a new service discovery infrastructure for Gridcole that can be integrated in it using the ontology depicted in section 3. A sample scenario is presented to illustrate the new discovery capabilities as a proof of concept that this ontology can be employed meaningfully in a real grid-based CSCL system. Then, a discussion about this proposal follows.
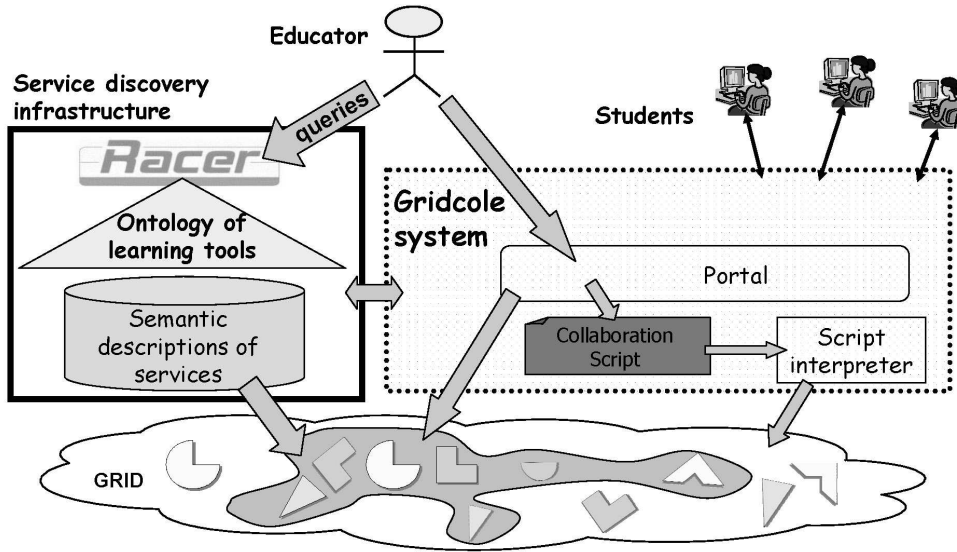
Fig. 4. Overview of Gridcole architecture and the proposed service discovery infrastructure.

4.1 A New Service Discovery Infrastructure for Gridcole

Gridcole [4] is a collaborative learning system that can be easily tailored by educators in order to support their own CSCL scenarios. Tailorability is achieved by integrating learning tools offered by third party providers in the form of OGSA grid services. That integration is driven by instructions given by educators in collaboration scripts [11] formalized in IMS-LD [20]. A collaboration script basically embodies a workflow of learning activities and the tools and documents required to support these activities. Gridcole has 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, the script interpreter validates the script and arranges the sequence of activities. The next step involves the search facility to discover suitable tools supplied by third-party providers in an OGSA-based grid according to the description in the script. Next, the educator can select suitable services among those retrieved. To finish the setup, the educator provides the list of participants and the roles they play. Then, the educator requests Gridcole to launch the application and students can join the execution.

If available CSCL tools were annotated with the proposed ontology, semantic metadata of CSCL services could be stored in repositories and queried to retrieve those services that meet some criteria. The ontology provides a uniform
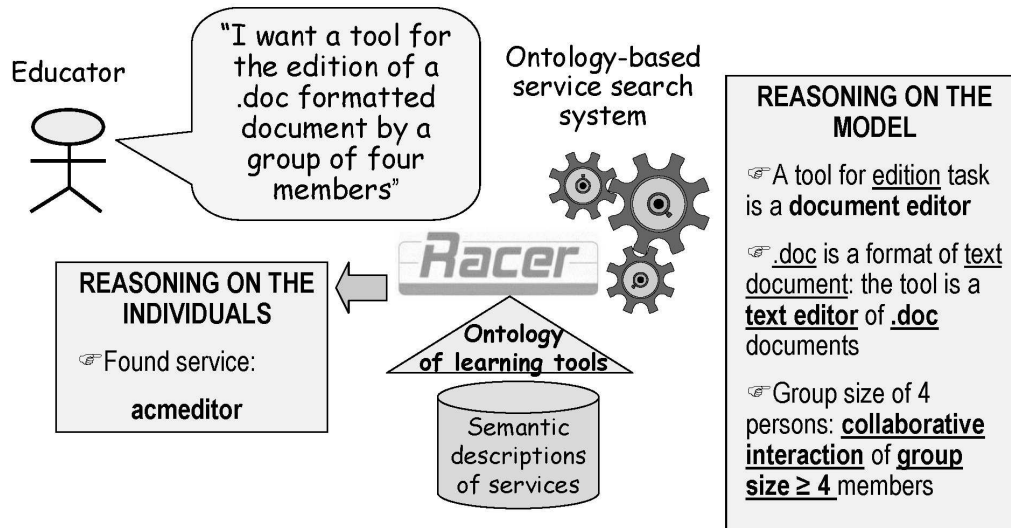
Fig. 5. Sample query with the proposed service discovery infrastructure.

and high-level view of the service descriptions stored in the repositories. An OWL-compliant reasoner is necessary to manage the resulting knowledge base, enabling the retrieval of service instances. An overview of the Gridcole system with this new service discovery infrastructure, using the Racer reasoner [18], is shown in figure 4. For the shake of simplicity we assume here a centralized repository within Gridcole, but other types of architectures are possible, even external to Gridcole.

4.2 Sample Discovery of Services

In order to illustrate the capabilities of this service discovery infrastructure when compared to other approaches for service discovery described in sections 2.2 and 2.3, a sample query for a CSCL tool in a collaborative learning activity is proposed. Consider an activity consisting on the collaborative edition of a .doc formatted document by a group of four learners. To support this activity, the educator submits the following query to the service discovery infrastructure (see figure 4): 'I want a tool for the edition of a .doc formatted document by a group of four members', the third query in section 2.1.

As shown in figure 5, when the reasoner receives this query, it builds an anonymous concept that complies with the restrictions posed in the query. This way, Racer deduces that the educator wants any collaborative text editor of .doc formatted documents which supports groups of a minimum of four participants. Next, the reasoner looks up instances of learning tools that match this criteria. The tool `acmeditor` (see figure 3) is suitable since it is a synchronous text editor that is a concept more specific than collaborative text

15

editor. Moreover, it supports .doc formatted documents for edition tasks and it can be used by four participants, fulfilling all the requisites. So, this particular tool is retrieved and offered to the educator.

Since Racer is the employed reasoner, the Racer Query language (RQL) [18] is used to query for learning services. RQL allows the expression of complex queries, such as the one in the previous example that would be expressed as follows:

```
(retrieve (?TOOL) (and (?TOOL |LearningTool|) (?TASK |Edition|)
(?TOOL ?TASK |canSupportTask|) (?TASK (some |hasOutput| |.doc|))
(?INTER |CollaborativeInteraction|)
(?TASK ?INTER |allowsInteraction|) (?INTER (min |groupSize| 4)))
```

However, it compels the user to learn the RQL syntax. Educators should not be burdened with this issue, and we are working on the development of an application with appropriate graphical interfaces to hide the RQL syntax. This way, we envisage a graphical interface that allows the user to browse the taxonomy of CSCL tools inferred by the reasoner, as well as drop-down lists that create a property based description of the desired learning service. A similar approach has been undertaken elsewhere, as e.g. in [42] and [22]. For instance, [42] illustrates the use of a GUI for service discovery that composes a service query and is submitted to a DL reasoner. Such a simple graphical interface will be convenient to users in most of the cases, although it may limit the access to the rich expressivity that can be achieved with OWL. Further work involves the research in rich graphical interfaces along with the user community.

Finally, it is difficult to use OWL-S for the discovery of services in this sample scenario. It requires the description of supported operations of a synchronous editor in terms of the inputs, outputs, preconditions and effects. The complexity of the interface of such an editor makes this process highly cumbersome and very difficult to assume by an educator.

4.3 Discussion

As already shown, the discovery 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. The only requirement is that the proposed ontology is used to describe the learning services. An open question is who annotates service instances and publishes those descriptions in a repository. Service providers could assume this issue, although domain experts could validate provided descriptions. Moreover, existing descriptions of learning tools can serve as templates, easing

16

the annotation process. In addition, the same tool can be implemented with different technologies (e.g. grid services and web services) or offered by different service providers, so it does not become necessary to describe that tool again. Furthermore, the reasoner can check the validity of a tool description, meaning that it conforms to the ontology.

The described working example in section 4.2 illustrates the performance of the service discovery facility. Other laboratory experiments have been done to test this infrastructure. One concern is that some tools are very domain-specific, such as the TCP/IP network simulator requested in section 2.1. This type of queries requires knowledge about curricula. Note that the remaining queries in section 2.1 can be expressed with the proposed ontology. The first query is very simple and asks for all the tool instances that conform to the definition of concept `Chat` in the ontology. Finally, fourth and fifth queries ask for tools that can support a specific task, `Document Exchange` and `Debate` respectively. In the latter case the `Debate` is further qualified as `asynchronous` for a `Group Size` of more than twenty individuals. While these preliminary results serve as a proof of concept of the proposal, further validation shall be achieved with an experiment involving practitioners querying the service discovery facility.

It is interesting to compare the proposed system with other similar proposals in the literature. Feta [26] is an architecture for semantic service discovery in the bioinformatics domain. It is noteworthy that in [26] OWL-S is also discarded, proposing an ontology to ease service discovery by domain users, similarly to what we propose here. The overall architecture resembles our proposal, specifying a module for service annotation and another one for service discovery to be performed by bioinformatics scientists interacting with a specific GUI. However, service discovery is enabled with a light-weight semantic engine in combination with a UDDI registry rather than using a DL reasoner like Racer. This way, service descriptions are simpler, although they argue that appears to be sufficient in the bioinformatics domain and it also reduces the computational power needed to perform a query. Besides, Feta also allows UDDI syntactic queries, such as other approaches in the literature: [28] combines OWL-S with UDDI in the e-commerce domain and [31] proposes an extension of UDDI to include arbitrary metadata for service discovery. Similar comments can be made about Biomoby [41], another semantic discovery architecture. Our approach is conceptually simpler since it employs a unique module, the Racer reasoner, to manage the knowledge base and to perform the search of services, although OWL-based service descriptions are more complex. In addition, it could be interesting to offer a UDDI interface to our proposed service discovery infrastructure.

5   Conclusions and Future Work

This paper discussed the use of ontologies to describe CSCL tools, enabling semantic searches of learning services in CSCL systems. The use of ontologies to formalize knowledge provides clear and unambiguous semantics that can be exploited in a service discovery infrastructure for service-based CSCL systems.

Furthermore, an ontology has been proposed that allows educators to formulate a broad range of queries for CSCL tools employing learning concepts that can fit many situations. For instance, they can be as general as 'communication tools' or as specific as 'synchronous collaborative editor'. Besides, educators can look up tools that may support a particular learning scenario by specifying the tasks, group sizes or collaboration properties. Since the ontology is the result of an iterative process and it is evolving, new features of CSCL tools could be added, as well as relations between the tools and the topics in the curricula when necessary.

Existing approaches for service discovery do not rely on semantics and cannot represent service capabilities, precluding their discovery. While there are other initiatives to exploit semantics for service discovery, specially OWL-S, the OWL-S ontology is not suitable for the discovery of CSCL services, as discussed in this paper.

Future work includes the development of a graphical front-end application for the discovery of CSCL services, in order to hide the complexity of the underlying system to the educators. Since the proposed service discovery facility has already been integrated in the Gridcole learning system, the complete infrastructure could then be tested in real educational settings. Then, a final evaluation could be performed on the expected advantages of the proposed solution. Besides we are studying the integration of this discovery infrastructure with an existing registry such as UDDI. This would ensure compatibility with an industry standard, while benefiting from its functionality such as an agreed interface. While the proposal assumes that an ontology is agreed among the participants of a virtual organization in a grid, further work involves research on ontology integration in the broader context of multiple virtual organizations. 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 predefined quality of service. Furthermore, user ratings of a service could be used to give a measure of reputation.

Acknowledgements

References

[1] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. Stein. OWL web ontology language reference, Feb. 2004. URL: http://www.w3.org/TR/owl-ref/, last visited February 2005.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[3] M. Bote-Lorenzo, D. Hernández-Leo, Y. Dimitriadis, J. Asensio-Pérez, E. Gómez-Sánchez, G. Vega-Gorgojo, and L. Vaquero-González. Towards reusability and tailorability in collaborative learning systems using IMS-LD and grid services. *Advanced Technology for Learning*, 1(3):129–138, Sept. 2004.

[4] 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 *Proceedings 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.

[5] B. Chandrasekaran, J. Josephson, and V. Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):72–81, 1999.

[6] R. Crawley and J. Leary. A framework for the comparison of computer-supported collaborative learning applications. In S. Lobodzinski and I. Tomek, editors, *Proceedings of WebNet 97 - World Conference on the WWW, Internet & Intranet*, Toronto, Canada, Nov. 1997. AACE.

[7] C. Crook. *Computers and the collaborative experience of learning*. Routeledge, London, United Kingdom, 1994.

[8] 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.

[9] D. de Roure, N. Jennings, and N. Shadbot. 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, first edition, 2003.

[10] P. Dillenbourg. What do you mean by collaborative learning? In P. Dillenbourg, editor, *Collaborative-learning: Cognitive and Computational Approaches*, pages 1–19. Elsevier, Oxford, United Kingdom, 1999.

[11] P. Dillenbourg. Over-scripting CSCL: the risks of blending collaborative learning with instructional design. In P. Kirschner, editor, *Three words of CSCL. Can we support CSCL?*, pages 61–91. Open Universiteit Nederland, Heerlen, 2002.

[12] A. Dimitracopoulou. Designing collaborative learning systems: Current trends and future research agenda. In T. Koschmann, D. Suthers, and T.-W. Chan, editors, *Proceedings of the Computer Support for Collaborative Learning 2005 (CSCL 2005)*, Taipei, Taiwan, May 2005. Lawrence Erlbaum Associates.

[13] ELeGi. European learning grid infrastructure. URL: http://www.elegi.org, last visited December 2004.

[14] 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.

[15] 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.

[16] Globus. The globus project. URL: http://www.globus.org, last visited May 2005.

[17] A. Gómez-Pérez. Ontology evaluation. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 251–274. Springer-Verlag, 2003.

[18] V. Haarslev and R. Möller. RACER user's guide and reference manual, Apr. 2004. URL: http://www.sts.tu-harburg.de/∼r.f.moeller/racer/racer-manual-1-7-19.pdf, last visited April 2005.

[19] IEEE Learning Technology Standards Committee. IEEE standard for learning object metadata. Technical Report 1484.12.1-2002, Nov. 2002. URL: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, last visited October 2004.

[20] IMS Global Learning Consortium. IMS learning design information model v1.0, final specification. Technical report, IMS Global Learning Consortium, 2003. URL: http://www.imsproject.org/learningdesign/index.cfm, last visited December 2004.

[21] 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 *Proceedings Of First International Conference on Service-Oriented Computing (ICSOC2003)*, Trento, Italy, Dec. 2003.

[22] T. Kawamura, T. Hasegawa, A. Ohsuga, M. Paolucci, and K. Sycara. Web services lookup: a matchmaker experiment. *IT Professional*, 7(2):36–41, 2005.

[23] A. R. Kaye. Computer supported collaborative learning in a multi-media distance learning environment. In C. O'Malley, editor, *Computer Supported Collaborative Learning*, pages 125–144. Springer-Verlag, 1995.

[24] 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.

[25] E. Lehtinen, K. Hakkarainen, L. Lipponen, M. Rahikainen, and H. Muukkonen. Computer supported collaborative learning: a review. Technical report, CL-Net Project, 1999. URL: http://etu.utu.fi/papers/clnet/clnetreport.html, last visited January 2006.

[26] P. Lord, P. Alper, C. Wroe, and C. Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In A. Gómez-Pérez and J. Euzenat, editors, *The Semantic Web: Research and Applications: Second European Semantic Web Conference, ESWC 2005*, volume 3532 of *Lecture Notes in Computer Science*, pages 17–31, Heraklion, Crete, Greece, May 2005. Springer-Verlag.

[27] P. Lord, S. Bechhofer, M. Wilkinson, G. Schiltz, D. Gessler, D. Hull, C. Goble, and L. Stein. Applying semantic web services to bioinformatics: Experiences gained, lessons learnt. In S. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *Third International Semantic Web Conference, ISWC 2004*, volume 3298 of *Lecture Notes in Computer Science*, Hiroshima, Japan, Nov. 2004. Springer.

[28] D. Martin et al. Bringing semantics to web services: The OWL-S approach. In *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, CA, USA, July 2004.

[29] D. Martin et al. OWL-S: Semantic markup for web services (version 1.1). Technical report, DARPA Agent Markup Language Program, 2004. URL: http://www.daml.org/services/owl-s/1.1/overview/, last visited May 2005.

[30] J. Martins-Ferreira, G. Alves, R. Costa, and N. Hine. Collaborative learning in a web-accessible workbench. In *Proceedings of the Eighth International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2003)*, LNCS 2440, pages 25–34, La Serena, Chile, Sept. 2003. Springer-Verlag.

[31] S. Miles, J. Papay, V. Dialani, M. Luck, K. Decker, T. Payne, and L. Moreau. Personalised grid service discovery. *IEE Proceedings software: Special issue on Performance Engineering*, 150(4):252–256, 2003.

[32] N. Noy. Evaluation by ontology consumers, in special section: Why evaluate ontology technologies? because it works! *IEEE Intelligent Systems*, 19(4):80–81, 2004.

[33] 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 May 2005.

[34] M. Paolucci and K. Sycara. Autonomous semantic web services. *IEEE Internet Computing*, 7(5):34–41, Oct. 2003.

[35] M. Papazoglou and D. Georgakopoulos. Service-oriented computing. *Communications of the ACM*, 46(10):25–28, Oct. 2003.

[36] M. Ramamurthy, R. Wilhelmson, R. Pea, M. Louis, and D. Edelson. CoVis: A national science education collaboratory. In *Proceedings of the American Meteorological Society 4th Conference on Education*, Dallas, TX, USA, 1995.

[37] J. Roschelle, C. DiGiano, M. Koutis, A. Repenning, J. Philips, N. Jackiw, and D. Suthers. Developing educational software components. *IEEE Computer*, 32(9):50–58, Sept. 1999.

[38] A. Sheth, C. Ramakrishnan, and C. Thomas. Semantics for the semantic web: The implicit, the formal and the powerful. *International Journal on Semantic Web & Information Systems*, 1(1):1–18, 2005.

[39] G. Vega-Gorgojo, M. Bote-Lorenzo, E. Gómez-Sánchez, Y. Dimitriadis, and J. Asensio-Pérez. Semantic description of collaboration scripts for service oriented CSCL systems. In C. Looi, G. McCalla, B. Bredeweg, and J. Breuker, editors, *Proceedings of the 12th International Conference on*

Artificial Intelligence in Education (AIED 2005), pages 935–937, Amsterdam, Netherlands, July 2005.

[40] J. Waldo. Alive and well: Jini technology today. *IEEE Computer*, 33(6):107–109, June 2000.

[41] M. Wilkinson and M. Links. BioMOBY: An open source biological web services proposal. *Briefings in Bioinformatics*, 3(4):331–341, 2002.

[42] C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood. A suite of DAML+OIL ontologies to describe bioinformatics web services and data. *The International Journal of Cooperative Information Systems*, 12(2):597–624, 2003.

Guillermo Vega-Gorgojo received his M.S. degree in telecommunications engineering from the University of Valladolid, Spain, in 2000. He is currently studying towards a PhD degree in telecommunications engineering at the University of Valladolid. He is a Teaching Assistant at the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. His research interests include semantic web and grid technologies to support CSCL systems.

Miguel Luis Bote-Lorenzo received the M.S. and the PhD degrees in telecommunications engineering from University of Valladolid, Spain, in 2001 and 2005, respectively. He is a Teaching Assistant at the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. His research interests include CSCL systems and grid computing.

Eduardo Gómez-Sánchez received the M.S. and the PhD degrees in telecommunications engineering from University of Valladolid, Spain, in 1996 and 2001, respectively. He is currently an Associate Professor at the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. His research interests include distributed systems and their application to the design and implementation of CSCL scenarios.

Yannis A. Dimitriadis received the engineering degree from the National Technical University of Athens, Greece (1981), the M.S. from the University of Virginia (1983) and two PhD degrees from the University of Valladolid, Spain (1992 and 1995), both in telecommunications engineering. He is currently an Associate Professor at the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. His research interests include computer supported collaborative learning and distributed systems.

22

Juan Ignacio Asensio-Pérez received the M.S. and the PhD degrees in telecommunications engineering from University of Valladolid, Spain, in 1995 and 2000, respectively. He is currently an Associate Professor at the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. His research interests include distributed systems and, particularly, distributed CSCL applications and integrated systems and network management.