Semantic Description of Collaboration Scripts for Service Oriented CSCL Systems

Guillermo Vega-Gorgojo 1, Miguel L. Bote-Lorenzo, Eduardo Gómez-Sánchez, Yannis A. Dimitriadis and Juan I. Asensio-Pérez

School of Telecommunications Engineering, University of Valladolid, Spain

Abstract. Many CSCL systems have embraced scripting and service oriented computing to achieve effective collaboration and system flexibility, respectively. While learning standards, such as IMS-LD, can be used for scripting, we have encountered some problems to describe activity types, their collaboration properties and learning tools with this standard. The usability of collaboration scripts is limited, since some important features cannot be described. Furthermore, poor description of tools hinders the search of tools, offered as services, in service oriented CSCL systems. To overcome these difficulties, we propose an ontology that can be used to enrich the description of activities and tools in a script. Besides, the authoring process of a learning design is eased due to enforced restrictions in the ontology as well as the use of off-the-self ontology editors. Furthermore, formal and explicit semantics in a script can be exploited to automate the search of tools. This way, service providers can describe their tools in terms of the ontology, while educators can search for them using domain concepts.

Keywords. CSCL, Scripting, IMS-LD, Ontologies, Service Discovery

1. Introduction

Computer Supported Collaborative Learning (CSCL) [16] is a well established discipline that promotes the use of technology to support collaborative (besides individual) learning activities. Achieving effective collaborative learning is more than providing some tools that allow students’ interaction. Indeed, free collaboration does not systematically produce learning. “Scripting is a means to enhance the effectiveness of collaboration by prescribing how students should form groups, how they should interact and collaborate and how they should solve the problem” [7]. Scripting is not linked to computational infrastructures, since it can be used to describe learning scenarios without computer support. However, scripts can be implemented in learning systems that provide computational resources when necessary and can integrate script interpreters in order to manage the sequence of learning activities. An example of a script interpreter is CopperCore [5].

Besides, developing CSCL applications is a demanding task that implies a considerable effort. As [21] 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 could be tackled with the service oriented computing model [19], since services represent high-level abstractions that are closer to educators’ mental model and thus enable and promote educational software reuse in the integration of learning environments. Many learning systems like Gridcole [2] or Edutella [18] have adopted the service oriented approach.

Combining scripting with service oriented computing may seem a “silver bullet” solution for CSCL systems. Indeed, educators and learning designers could build their own collaboration scripts or adapt existing ones for their educational scenarios. Next, a script interpreter within a CSCL system would validate the script and arrange the sequence of activities. Then, external resources and tools offered as services needed to support the scenario described in the script would be discovered and integrated in the system. Finally, the set of users (possibly organized in groups) would join the resulting set up.

Developing such a system involves many challenging issues. First of all, an Educational Modelling Language (EML) is needed to describe the collaboration scripts. This description should be formal in order to be unambiguously interpreted. This way, a script player could manage the flow of activities to be performed in an educational system, as well as the arrangement of needed learning resources. The IMS Learning Design (IMS-
A collaboration script comprises a flow of activities that can be performed individually or collaboratively. Each activity is supported by a set of learning resources that can integrate tools and contents. Although the IMS-LD model uses these abstractions, we have encountered some difficulties when using IMS-LD to formalize collaboration scripts. First, activity types are not defined. Each activity type, e.g. an edition or a debate, has some distinguishing properties, such as specific outcomes and roles, that should be identified in a collaboration script. Second, collaborative activities cannot be properly described [10]. This is critical to state how learners should interact to perform a collaborative activity. A third issue is the description of learning tools in a script. IMS-LD can integrate descriptions of learning objects in a learning design using standards such as IEEE LOM [12] or the service elements included in the IMS-LD specification, e.g. a conference. However, only a limited set of tools can be specified, as standards of learning objects do not even define a vocabulary of learning tools. On the one hand, these problems reduce the expressiveness of collaboration scripts precluding usability in educational scenarios and script sharing, since significant information cannot be expressed. On the other hand, poor description of learning tools severely limits automated discovery of appropriate tools, offered as services, to be integrated in a service oriented CSCL system. If learning tools were properly described in a learning design, a computer agent could support the discovery of suitable services. Instead, an educator should manually search for learning tools, reducing the usability of such CSCL system.

To overcome these difficulties, a collaboration script should better describe learning activities with meaningful information about activity types and collaboration features. Furthermore, describing the required learning tools to support a collaboration script would ease the binding of specific tools during the enactment of the scenario. In his sense, an ontology could be employed to capture the semantics involved in the description of learning activities and tools. Ontologies [4] are used to explicitly formalize knowledge in a shared manner, enabling rich descriptions and robust information retrieval systems. Thus, in this paper we propose an ontology that can be used to enrich the description of the activities and tools involved in collaboration scripts, while easing the authoring process. Besides, in a service oriented learning system tools offered as services can be searched using the learning abstractions described in the ontology. In previous work we analysed current service discovery mechanisms and proposed the use of educational ontologies to ease educators to search for learning services using their own terms [23]. In the e-learning field, ontologies have been also proposed to semantically annotate learning contents enabling their retrieval [3]. Interestingly, the Learning Palette [14] is an ontology-aware authoring system for learning designs, although it does not address the problems detected here.

The rest of the document is organized as follows: in Section 2, learning standards for scripting and the description of tools are discussed as well as the aforementioned problems. Section 3 deals with the proposal of an ontology that can enrich the description of activities and learning tools involved in collaboration scripts. Section 4 illustrates the application of such ontology in a collaborative learning scenario: activities and tools are described using the ontology abstractions, while this semantic description is exploited to perform the search of tools in the context of a service oriented collaborative system. Finally, the main conclusions of the study are shown as well as current research work.

2. Problems of Learning Standards to Describe Activities and Learning Tools

A service oriented CSCL system can integrate a script interpreter in order to manage the sequence of activities. To support these activities, learning tools offered as services can be integrated in the system. This section first analyses current learning standards that can be used for scripting as well as for the description of learning tools to enable the discovery of appropriate services that implement described tools. Next, the three main problems of these standards to describe activities and learning tools in a collaborative learning context are discussed.

2.1. Standards for Scripting and Description of Learning Tools

To allow for interoperability among learning systems and learning providers, educational standards aim to provide a common understanding. A shared standard for scripting allows that a script could be played in different learning systems if they can interpret the script’s language. To study the use scripting for collaborative learning, this review is focused on modelling languages that can describe collaborative learning activities. The IMS Learning Design (IMS-LD) specification [13] is the most remarkable initiative in this sense. Other educational languages for scripting are OUNL-EML [15] (the precursor of IMS-LD) and PALO [20]. Both languages cannot describe collaborative scripts, so they are not covered in this paper.

IMS-LD defines an XML schema that can formally express learning designs. The structure of the design is modelled as a workflow of learning activities. The learning flow specifies the sequence of activities that learners should perform in order to reach predefined learning objectives according to the different roles that they may...
play in a learning design. Activities take place in an environment which is described in terms of resources, i.e. tools and contents, that should assist learners during the realization of each activity according to the role played.

Learning resources in an IMS-LD environment can be described as a structured collection of **learning objects and services**. “A learning object is any reproducible and addressable digital or non-digital resource used to perform learning activities or support activities”, while “services are resources that cannot be given a URL at design time” [13]. IMS-LD defines a limited set of service types, specifically: send-mail, conference, monitor and index search. Referring to learning tools, they can be described as learning objects or as services. In the former case the specific instance of the tool that will support the scenario is known at design time, while in the latter it will be known during the enactment. Besides, learning tools can be annotated with metadata for classification purposes or to ease the discovery of services that implement a learning tool, for example.

IEEE LOM [12] is frequently employed to annotate learning contents with metadata. Specifically, LOM specifies nine categories for over 70 metadata elements with learning resources, although all the categories and elements are optional. Since LOM was developed to describe any kind of learning resources, it could be used to describe learning tools. There are other different e-learning metadata standards, as depicted in [22]. However, they have not different capabilities since most of them are based on LOM.

The reminder of this section discusses the three main difficulties encountered when using these standards to describe activities and learning tools in collaboration scripts.

### 2.2. First Problem: Describing Activity Types

IMS-LD defines two basic types of activities: learning activities and support activities. A learning activity is aimed at achieving a learning objective, while a support activity is meant to facilitate a role performing one or more learning activities. To further describe activities, IMS-LD defines an activity description element to provide a textual description intended to be the actual cue given to the user to perform the activity.

Although the IMS-LD activity description is sufficient to guide the user to perform an activity, it does not specify activity types which would be useful for classification purposes. For instance, a learning activity could be an edition, an exposition, a debate, etc. Besides, each activity type has some distinguishing characteristics that should be included in a design, e.g. the outcome of an edition activity is a document, supported roles are editor and reviewer and the activity is supported by a document editor tool. This meaningful information can be exploited in some ways. Since authoring a learning design is an error-prone and time-consuming task, an authoring system could embed this information to support the user when authoring a design. Furthermore, describing activity types can be used to perform the automatic discovery of tools that can support such activity. This is relevant for service oriented CSCL systems that must tackle the search of services.

### 2.3. Second Problem: Describing Collaborative Activities

Another issue is how to model collaborative activities in a learning design. IMS-LD provides no means to specify how individuals collaborate within each learning activity. The specification supports the association of multiple roles (a role can be played by one or more individuals) to an activity. However, depending on the associated resources, an activity could be individual or collaborative. For instance, individuals engaged in an edition activity can perform it collaboratively, if supported by a synchronous collaborative editor, or individually, if multiple instances of an editor are created. This way, the specification states that if multiple individuals are to collaborate or work together at the same time, this has to be done through a service element in their assigned environment which supports this collaborative capability. However, only two services defined in IMS-LD are collaborative to some extent, send-mail and conference, and the IMS-LD service element does not define collaborative capabilities.

To overcome this problem, [10] proposes an extension of the IMS-LD service definition. Although the proposed groupservice element can describe some collaboration capabilities, they are defined at tool level, rather than at activity level. This is a workaround, since a collaboration activity may not need a tool to support it, e.g. a face-to-face synchronous debate. Moreover, a non-collaborative tool could support a collaboration activity, e.g. participants in a synchronous debate could use an individual document viewer to read some contents.

### 2.4. Third Problem: Describing Learning Tools

A major concern in IMS-LD is the description of learning tools. As previously depicted, a learning tool can be defined as a learning object or as a service in IMS-LD. This distinction is due to a technological issue: defining a tool as a learning object ties the design to the specific implementation of a tool, precluding reutilization of learning designs. Indeed, using a service oriented CSCL system to enact a learning scenario allows the use of learning tools offered by third-party service providers. In such case, the precise tools that will be used are not known at design time, but during the enactment or even at runtime.
Therefore, learning tools should be described as IMS-LD services in a learning design to enable the binding with specific tools after the design. In this situation it is critical to describe them properly. As discussed above, only four types of IMS-LD services are defined in the specification. Although the IMS-LD extension proposed in [10] can be applied to describe some collaboration capabilities of learning tools, it is important to define their functionality in order to support the search. In this sense, it would be useful to exploit relationships among learning tools and activity types, as exemplified in section 2.2 for an edition activity type.

Since LOM was proposed to describe any type of learning resources, it could be used to describe learning tools. However, a first issue is that the collaboration properties of learning tools do not fit well in the LOM specification. Moreover, LOM does not even define a vocabulary for learning tools, although it could be specified in the future. This way, some of the LOM elements can be applied to describe learning tools, but interoperability is precluded since it is the provider’s responsibility to employ meaningful words to describe a learning tool. Search of learning tools is hindered, since there is not a previous agreement in the vocabulary among providers and consumers. A common vocabulary is a first step to enhance searches, but adding semantics to further describe tools, their properties and their semantic interconnections allows for more expressive queries than only keyword-based. Ontologies [4] can be employed to formalize semantics enabling knowledge sharing and powerful information retrieval systems. Ontologies have been applied to semantically annotate and retrieve learnings contents using a subset of the LOM specification [3].

3. Describing Collaboration Scripts with an Ontology of Activities and Learning Tools

The analysis so far illustrates that IMS-LD has some important limitations to describe activities, in particular collaborative activities. Besides, it is difficult to specify the tools required to support an activity, since neither IMS-LD nor LOM properly cover this issue. These facts severely limit the expressiveness of learning designs, reducing reusability and sharing of scripts among educators. Another concern is the search of learning tools in a service oriented learning system. If activities and required tools are not properly described, the learning infrastructure can provide little support for educators to perform the discovery of appropriate services. To overcome these issues, an IMS-LD-compliant collaboration script could be further described with the semantic annotation of the activities and learning tools included in the script. An ontology of activities and learning tools for CSCL could be employed to formalize this knowledge with explicit semantics which can be easily shared and it is machine-interpretable.

While delivering proprietary extensions to describe the desired capabilities not covered in these standards could be a plausible solution (such as the IMS-LD extension proposed in [10]), this approach has some shortcomings. First, semantics are not explicitly stated, allowing for ambiguous interpretations [6]. Second, proprietary extensions reduce interoperability of learning designs, since it is needed that a script interpreter can understand added extensions. Moreover, non standardized vocabularies further limit interoperability, since there is a high risk that different people express the same concept in different ways. In contrast, ontologies comprise a set of knowledge terms, including the vocabulary, the semantic interconnections and some simple rules of inference and logic for some particular topic [9], and they are easily shared (usually published as files in the Web). Formal and explicit semantics avoid ambiguous interpretations. Besides, although it is possible that other people use different ontologies or schemas, it is easy to provide mappings among concepts in the ontology, allowing for interoperability.

Therefore, an ontology could describe the activities and learning tools of a learning design in an unambiguous way. Besides, such an ontology could embed some knowledge about pre-established relationships and restrictions among concepts, easing the authoring process of learning designs. For instance, when creating an activity of type “Speech”, the ontology-enabled authoring system could automatically include “Speaker” and “Attendant” roles and suggest “Audioconference”, “Videoconference” or “SlideProjector” tool types to support the activity. Other authors have also proposed the use of ontologies for authoring learning designs [14]. The main advantage of their authoring system is reduced complexity of the design process, while it is compliant with learning standards such as IMS-LD. However, though it can describe some collaboration features of activities, it does not further describe neither activity types nor tools.

3.1. An ontology of Activities and Learning Tools

The ontology we propose here is the result of an iterative process that we have developed collaboratively among a group of technicians and educators. It was initially conceived to support educators in the search of learning tools offered as services using educational abstractions [23]. However, it has grown and evolved to overcome the limitations of scripting languages exposed in section 2, while it still enables the semantic search of learning tools. The language employed to formalize the ontology is OWL [1], a popular and expressive ontological language. We have used the ontology editor Protégé [11] to ease the authoring.
The conceptual model of this ontology is shown pictorially in figure 1. Comparing to the conceptual model of IMS-LD [13, figure 3.1, p. 21], core concepts in both models are roles, activities and learning resources. However, the proposed model cleanly differentiates Content and LearningTool resources to avoid tying up the description of a learning design to a runtime facility. Therefore, it emphasizes pedagogical issues enabling reusability of learning designs in different contexts with distinct facilities. Another design objective is decoupling the description of activities from the description of tools and contents. With this aim, a subset of the roles that perform an activity can use a content or a learning tool, achieving enhanced flexibility in the design, whilst in a IMS-LD script all the roles that perform an activity can access the resources specified in the associated environment.

We tackle now how the proposed ontology addresses the problems detected in section 2. First, although not exhaustively shown in figure 1, activity types are defined, namely: Assessing, Debate, DocumentExchange, Edition, GroupForming, Mentoring, ProblemSolving, Scheduling, Speech, Reviewing and Studying. We have encountered that these activity types can be applied to a broad range of collaboration scenarios. The compromise here was defining a compact and highly reusable set of activity types. Therefore, we have avoided too much detail to reinforce the commonalities. For instance, a Debate could be further specialized: brainstorming, negotiation, synthesis, peer teaching, voting, etc. Besides, each activity type has a set of applicable role types as well as some suitable tool types. The ontology embeds this knowledge that can be used when authoring a design: if an educator wants to describe a DocumentExchange activity, he can query for appropriate tools to support this activity type and the ontology will suggest Repository and ElectronicMail.

Second, collaboration capabilities of activities can be expressed using the ontologies. The well-known categorization using time and space [8] is employed here. Therefore, an activity can be individually or collaboratively performed. Some activity types are always individual, (e.g. Studying), others collaborative, (e.g. Debate) and others could be individual or collaborative, (e.g. Edition). Besides, collaboration properties capabilities are defined at activity level and at tool level, since a specific learning tool, e.g. an individual text editor, can have different collaboration properties than the activities that it supports, e.g. a debate. Moreover, a learning tool can be configured with different collaborative properties, e.g. a synchronous collaborative text editor can also be used in an individual edition.

Finally, learning tools can be described using the educational abstractions modelled in the ontology. Defined tool types are the following: Agenda, AudioConference, Chat, ConceptualMapTool, ElectronicMail, Game, GroupFormationTool, SlideProjector, QuestionnaireTool, Repository, Simulator, SpecificHardwareResource, StickyNotes, TextEditor, VideoConference, Viewer and Whiteboard. Again, the compromise was to propose a compact and reusable set of tool types. We are conscious that these tool types are broad in conception and should be refined to describe specific tool types such as an Oscilloscope (subclass of SpecificHardwareResource). However, many tool types are domain-dependent, so other domain-specific ontologies could extend this one to suit particular educational scenarios. Besides, collaboration properties, inputs, outputs, roles and supported activities can be specified to describe a tool. Inputs and outputs of a tool can be linked to contents, outcomes and the remaining elements of the conceptual model, enabling a more cohesive script description. For instance, the input of a DocumentViewer could be a Content used in a particular Study activity. Another example is the association of the Output of a TextEditor with the Outcome of an Edition activity.

Next section deals with the application of the proposed ontology to describe the activities and tools involved in a collaboration learning scenario. Afterwards, such description is exploited to search appropriate tools in a service oriented CSCL system.
4. Application in a Collaboration Learning Scenario

4.1. Description of a Collaborative Learning Scenario with the Proposed Ontology

Due to space constraints, a very simple collaborative scenario is proposed in order to illustrate the capabilities of the ontology for scripting. It is based on the well-known “snowball” collaboration pattern [7] and it comprises three sequential activities. The first one consists on reading a document. After the reading, learners must individually respond to a questionnaire about the document. Finally, there is a synchronous debate activity in which participants have to agree a common response for the same questionnaire. They all have access to the responses that they submitted before, and one participant is in charge of the new submission.

This simple script can be formalized in IMS-LD. However, problems detected in section 2 should be addressed in order to enable the actual realization of the scenario. Therefore, involved activities and tools can be described in terms of the proposed ontology, using an ontology editor such as Protégé. This way, we begin instantiating an activity A1 of type Studying (see table 1). Automatically, A1 is qualified as Individual and the Learner role is assigned to perform the activity. We could alter these settings, e.g. adding new roles to the activity, but we decide to assert them. We continue creating a content D1 that corresponds to the document provided for the activity and that can be read by role Learner. Then, we ask the ontology what tool types can support a Studying activity type, and it suggests to use a Viewer tool type. So, we create an instance T1 of a Viewer that is used individually by role Learner to read D1.

In a similar way, we proceed to complete the specification of the remaining two activities. Table 1 outlines the description of the activities and resources of the whole scenario. Although activity A3 is significantly more complex, due to collaboration and the description of multiple contents and tools, the authoring is not so complicated since the graphical editor and enforced restrictions captured in the ontology simplify the task. Besides, activity types and their collaboration properties are clearly stated, while required tools are also deeply described.

Table 1. Description of a sample collaborative learning scenario. It comprises three activities: A1 and A2 are individual, while A3 depicts a collaborative debate. These activities, as well as the contents and tools that support them, are described using the abstractions modelled in the proposed ontology, shown in figure 1.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Content</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref Type</td>
<td>Collab Roles</td>
<td>Out</td>
</tr>
<tr>
<td>A1 Studying</td>
<td>Indiv Learner</td>
<td>-</td>
</tr>
<tr>
<td>A2 Assessment</td>
<td>Indiv Submitter D3</td>
<td></td>
</tr>
<tr>
<td>A3 Debate Collab</td>
<td>Distant Submitter</td>
<td>D4</td>
</tr>
<tr>
<td></td>
<td>Learner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2. Search of Tools in a Service Oriented CSCL System

Service oriented computing advocates increased flexibility and reusability to deliver software, enabling the use of third-party providers’ services. An important concern is how to discover appropriate services in order to realize such systems. Registries are commonly used for this issue, enabling providers to publish their service announcements, while consumers can find services that meet their criteria. Since common registries have poor service discovery capabilities, ontologies have been proposed to semantically enrich service descriptions, allowing for enhanced searches [17]. In the case of service oriented CSCL systems, educators use to settle the arrangement of the scenario, including the search of tools. They should be capable to perform this search in a convenient way. Therefore, educators could use the educational abstractions formalized in the proposed ontology to search for tools if providers commit to this ontology. An extensive discussion about this topic is offered in [23].

Thus, we briefly present here a sample tool search based on the scenario described in section 4.1. Currently, we are about to finish the implementation of an ontology-enabled registry based on the proposed ontology for the service oriented CSCL system Gridcole [2]. This system provides a service oriented grid infrastructure
Table 2. Learning service announcements in an ontology-enabled registry. Providers announce their services using the concepts in the proposed ontology, shown in figure 1.

<table>
<thead>
<tr>
<th>Service Ref</th>
<th>Tool Type</th>
<th>Supported Activities</th>
<th>Type of Interaction</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Editor</td>
<td>Edition</td>
<td>Individual</td>
<td>Writer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Studying</td>
<td>Individual</td>
<td>Learner</td>
</tr>
<tr>
<td>S2</td>
<td>Viewer</td>
<td>Studying</td>
<td>Individual</td>
<td>Learner</td>
</tr>
<tr>
<td>S3</td>
<td>Chat</td>
<td>Debate</td>
<td>Collaborative, Synch, Distant</td>
<td>Debater</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>Questionnaire</td>
<td>Assessment</td>
<td>Individual</td>
<td>Submitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Collaborative, Asynch</td>
<td>Submitter, Reviewer</td>
</tr>
</tbody>
</table>

that integrates an IMS-LD script interpreter for its use in educational settings. This registry compels providers to describe offered services in terms of the ontology, while it allows educators to query for services using educational abstractions. For instance, table 2 shows some service announcements describing the tools they implement as well as their collaboration properties and supported activity types. Therefore, a computer agent can interpret the semantic tool descriptions in the script (ranging \( T1 \) to \( T5 \) in table 1) and automatically query the registry. For tool description \( T3 \), the agent would search for services of tool type Chat and synchronous collaborative type of interaction, retrieving service \( S3 \). Besides, an educator can also search for services using the ontology concepts, e.g. he can query for services that support Studying activities and then he could choose between services \( S1 \) and \( S2 \).

Although this is only a very simple example, it can be extrapolated to more complex scenarios with a large number of available services. In that case, rich search capabilities and smart user support are critical. Besides, a user could refine the queries to obtain more targeted results. For instance, description of supported inputs and outputs of a service allows querying for specific types, such as “viewers that interpret PDF documents”.

5. Conclusions and Future Work

Current educational standards for scripting have some limitations to describe collaboration scripts. Activity types and learning tools, as well as their collaboration capabilities, cannot be properly specified. The ontology proposed in this paper overcomes these problems enabling the semantic description of these features, while still conforming to existing standards, such as IMS-LD. Off-the-self ontology editors and enforced constraints in the ontology significantly reduce the complexity of authoring such descriptions.

This way, educational scenarios can be deeply described allowing for enhanced usability, since the underlying learning infrastructure can take appropriate actions to enact the scenario, such as the unambiguous assignment of resources to roles. Besides, semantic description of tools and activities can be exploited to automate the search of tools, offered as services, in a service oriented CSCL system. Furthermore, educators can request desired tools using the learning abstractions expressed in the ontology. An ontology-enabled registry coerces service providers to commit their announcements to the concepts in the ontology, enabling semantic searches.

Although the emphasis of the ontology resides on collaboration scripts, it can be used to describe individual learning scenarios, since they are easier to formalize in a script. Indeed, the example provided presents a mixture of individual and collaborative activities. While the proposed ontology was designed to describe a broad range of learning scenarios, it is not exhaustive and should be extended to fit particular settings. For instance, in a science laboratory context, a domain ontology of tools could be used to add new elements, such as an oscilloscope or a microscope.

Upcoming work in the near future includes the integration of these features in the real CSCL system Grid-cole. With this aim, we are completing an ontology-enabled registry that allows for the registration of service announcements as well as the search of tools. In addition, we are adapting an IMS-LD script interpreter to understand the new capabilities modelled in the proposed ontology for the enactment of learning scenarios.

Acknowledgements

This work has been partially funded by Spanish Ministry of Science and Technology project TIC-2002-04258-C03-02, the European Commission e-learning project EAC/61/03/GR009 and the Kaleidoscope network of excellence FP6-2002-IST-507838. Special thanks to the MOSIL research project for the MOSIL deliverable 2.
References


