

TOWARDS REUSABILITY AND TAILORABILITY IN COLLABORATIVE LEARNING SYSTEMS USING IMS-LD AND GRID SERVICES

M.L. Bote-Lorenzo^{*}, D. Hernández-Leo^{*}, Y.A. Dimitriadis^{*}, J.I. Asensio-Pérez^{*},
E. Gómez-Sánchez^{*}, G. Vega-Gorgojo^{*}, L.M. Vaquero-González^{*}

Abstract

CSCL applications are complex distributed systems that pose special requirements towards achieving success in educational settings. Flexible and efficient design of collaborative activities by educators is a key precondition in order to provide CSCL tailorable systems, capable of adapting to the needs of each particular learning environment. Furthermore, some parts of those CSCL systems should be reused as often as possible in order to reduce development costs. In addition, it may be necessary to employ special hardware devices, computational resources that reside in other organizations, or even exceed the possibilities of one specific organization. Therefore, the proposal of this paper is twofold: collecting collaborative learning designs (scripting) provided by educators, based on well-known best practices (collaborative learning flow patterns) in a standard way (IMS-LD) in order to guide the tailoring of CSCL systems by selecting and integrating reusable CSCL software units; and, implementing those units in the form of grid services offered by third party providers. More specifically, this paper outlines a grid-based CSCL system having these features and illustrates its potential scope and applicability by means of a sample collaborative learning scenario.

Key Words

CSCL systems, reusability, tailorability, IMS-LD, Grid Services

1. Introduction

Computer-Supported Collaborative Learning (CSCL) is a research paradigm within the field of educational software that underlines the key role that social interactions play in the process of learning [1]. Collaborative learning theories, common practices used by educators, and economic factors pose two fundamental requirements on CSCL systems: they should be able to be configured and tailored by educators without requiring

special technical skills; and, they should be reused in as many different learning scenarios as possible.

Tailorability and reusability are related concepts within this context. A CSCL system valid for one particular learning scenario can hardly be reused completely in a different one: while part of its functionality could still be valid for the new requirements, the remaining aspects of the solution should be reworked. Therefore, one potential approach to the goal of reusability could be based on the identification of common (and therefore reusable) CSCL functional blocks: new CSCL solutions would thus consist of the integration of such blocks. Taking into account this approach, tailorability could be achieved by translating educators' requirements into the selection and integration of the appropriate set of CSCL functional blocks [2].

Therefore, the reusability requirement can be translated into the problem of identifying common CSCL functional blocks whereas tailorability depends on the educators-driven selection and integration of those blocks into CSCL solutions. These are the two key problems this paper focuses on.

Regarding the mapping of educators' requirements into a selection and integration of CSCL functional blocks, one recent *de facto* standard of educational technology is a potential solution: IMS *Learning Design* (IMS-LD) specification [3]. IMS-LD provides an Educational Modelling Language (EML) that enables the formal description of teaching-learning processes for a wide range of pedagogies in online learning, including collaborative learning. In this sense, IMS-LD specifications provide a formal way of collecting educators' decisions on learning scenarios and even the requirements of technological tools that could potentially support them. Therefore this paper proposes and justifies the use of the IMS-LD specifications reflecting educators' requirements as the starting point for the selection and integration of CSCL functional blocks. This integration would lead to the achievement of tailored CSCL solutions for particular collaborative learning scenarios.

On the other hand, regarding the identification of common CSCL functional blocks, several approaches

^{*} School of Telecommunications Engineering, University of Valladolid, Camino Viejo del Cementerio s/n, 47011 Valladolid, Spain; email: {migbot@, dherleo@ulises., yannis@, juaase@, edugom@, guiveg@, lvaqgon@ulises.}tel.uva.es

have already been proposed and employed by the authors [4]. This paper proposes a new approach: the identification and formalization of common practices in collaborative learning by means of the so-called CLFPs (Collaborative Learning Flow Patterns). CLFPs describe well-accepted ways of arranging participants in collaborative learning sessions, sequencing types of collaborative learning activities, assigning contents to those activities, etc. CLFPs can be used to identify common types of collaborative learning activities that might potentially be mapped onto CSCL reusable functional blocks. Furthermore, this paper also proposes the use of IMS-LD as the candidate formalism for representing CLFPs.

In order to apply all the above ideas in real practice, a computing system capable of interpreting IMS-LD specifications (potentially following the common ideas of CLFPs) and selecting and integrating existing CSCL common functional blocks (identified by means of the CLFP approach) should be developed.

This paper outlines a distributed system based on *grid services* technology [5] that would fulfill the above requirements. Such a system would be in charge of selecting and integrating CSCL functional blocks provided in the form of grid services according to the contents of IMS-LD specifications provided by educators. The use of the grid services technology would provide some interesting advantages to the CSCL field: scalability, support for supercomputing needs, cross-organizational service usage, support for the use of heterogeneous hardware devices, etc.

The rest of this paper is structured as follows. Section 2 introduces IMS-LD and proposes its use so as to collect educators' requirements that could potentially guide the tailorability of CSCL systems. Section 3 describes CLFPs, which can help identifying types of reusable CSCL building blocks. Section 4 defends the use of grid services technology for implementing reusable CSCL functional blocks that can be easily integrated to develop new CSCL applications. Section 5 outlines a system that could select and integrate CSCL grid services according to the prescriptions of IMS-LD documents, thus generating tailored CSCL solutions. Section 6 introduces an example of all the above ideas. Finally, section 7 concludes the paper and describes future research topics.

2. Scripting as the Basis for Tailorability: an Approach based on IMS-LD

Free collaboration does not automatically produce learning, and one way to enhance collaborative learning is by means of structuring interactions between students [6]. Structured interactions are defined in *collaboration scripts*. A collaboration script is a set of instructions

prescribing how students should form groups, how they should interact and collaborate and how they should solve a problem [6]. A collaboration script can thus describe a collaborative learning scenario in terms of the activities to be performed by participants, the interactions between students while performing such activities as well as the characteristics of tools that should be employed to support each activity.

The realization of a learning scenario described in a collaboration script requires properly tailoring a collaborative learning system. A computer system is said tailorable if it provides users with some means to modify its functionalities in order to better suit their needs [2]. In this sense, tailoring a collaborative learning system in order to support a given scenario described in a collaboration script would involve two issues: first, integrating suitable CSCL functional blocks to support the scenario within a single collaborative learning environment. This adheres to the concept of "tailoring by integration" described in [2]; and second, integrating suitable user interfaces of the learning environment according to the interactions to be supported. This can be regarded as "tailoring by customization" according to [2].

In order to enable the automation of these two operations, a formal description of collaboration scripts is required. In this sense, the IMS Learning (IMS-LD) design specification has recently emerged as a candidate solution for formal representation of collaboration scripts. The IMS-LD specification defines a structured XML-based language that can be employed to formally express *learning designs*. A learning design is a description of a method enabling learners to attain desired learning objectives by performing predefined learning activities in a certain order within the context of a given learning environment [3].

More specifically, a learning design describes a *learning scenario* in terms of a *learning flow*, and a set of *environments*. 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. Environments are 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.

IMS-LD specification states that it supports group and collaborative learning of different kinds. It enables the design of processes that include several roles, each of which can be played by several people (a group). A collaborative learning experience can then be described by associating multiple people and/or multiple roles to the same learning activity. Furthermore, IMS-LD enables their activities to be specified in coordinated learning flows that are similar to groupware workflows [7]. Therefore, IMS-LD is a reasonable candidate as a language with which to formalize collaboration scripts.

Nevertheless, while a main feature of CSCL applications is the set of mechanisms that support the collaborative interaction, IMS-LD provides no means to specify how the members of a group interact within each learning activity. It only states that if multiple individuals are to collaborate or work together at the same time, this has to be done through a service in their assigned environment which supports this collaborative capability [8]. Therefore, the concept of service is central in IMS-LD for CSCL. An IMS-LD service specification describes the characteristics of a *resource* that supports a learning activity. When applying an IMS-LD learning design to an actual learning scenario (e.g. a class) the learning designer (e.g. the teacher) must specify the resources that, at least, provide the implementation of the defined services, thus obtaining a so-called *unit of learning*. These resources range from a simple blackboard or a paper sheet to a complex e-learning or CSCL application.

However, IMS-LD only proposes and defines four basic services, two of which are (to a certain extent) collaborative: discussion forum and e-mail. Other needed services (collaborative or not) should be specified by the designers of learning scenarios. The problem is that IMS-LD does not allow the designers to describe collaboration-related capabilities when defining a new service: type of awareness information needed and provided by the service, floor control policy that guides learners actions, communication skills required to these learners, etc. Within this context we have proposed an extension of the IMS-LD service definition consisting in the definition of a special type of service, called *groupservice*, whose main characteristics are more deeply described in [9].

This generic characterization of collaborative services, together with the definition of *learning flows* provided by IMS-LD, enables a formal description of collaborative learning scenarios. The realization of such scenarios depends on the development of a tailorable collaborative learning system capable of interpreting IMS-LD scripts and supporting the scenarios by automatically performing the two tailoring operations described above. The design of such a system is the focus of section 5.

3. Collaborative Learning Patterns as the Basis for Reuse: the CLFP Proposal

The scenario depicted in previous sections is based on the IMS-LD script-guided integration of existing CSCL functional blocks. As stated in the introduction, the success of those CSCL blocks depends on their level of reuse in a large set of collaborative learning scenarios (prescribed by their associated collaboration scripts).

In order to identify reusable CSCL functional blocks, we followed several approaches [4]: the development of conceptual frameworks that try to map common principles and ideas of collaborative learning theories into reusable software artifacts (top-down approaches); and, the mining of existing CSCL solutions so as to identify shared functionality (bottom-up approach). None of these two approaches generated satisfactory results: blocks that were identified by the top-down approach were very generic and difficult to use in specific scenarios; and, those proposed by the bottom-up approach were very biased towards specific situations and therefore difficult to reuse.

Taking into account these results, we focused neither on high-level abstractions nor on low-level application-related ideas, but tried to understand how collaborative learning practitioners organize and structure collaborative learning scenarios. As a result, a set of *common techniques* used by collaborative learning practitioners should be detected. This procedure would ease the identification of broadly used functions in collaborative scenarios that could eventually be supported by CSCL functional blocks. Consequently, that identification process would result in more reusable blocks.

The identification of common techniques is an approach that is very similar to that of determining *design patterns* in software engineering. According to [10]: “each pattern describes a problem which occurs over and over again in our environment, and then describes the core solution to that problem, in such a way that you can use this solution a million times over”. The pattern approach is used in several domains; in fact, one of them is the learning field [11].

In the collaborative learning domain, different types of common techniques that are suitable for being formulated as design patterns can be identified: awareness techniques, group formation schemes, evaluation or scaffolding methods, collaborative learning flows, etc. Moreover, all these patterns and their interrelations might be arranged in a *pattern language* for CSCL.

Our current work is focused on one particular type of those common techniques used for formulating collaboration scripts: *collaborative learning flow structures*. They dictate common ways of structuring interactions among participants in different collaborative learning activities, as well as the types of information they exchange [12]. They are particularly significant as they dictate the structure of collaboration scripts, which could eventually be reused in different collaborative learning scenarios.

The formulation of these best practices as patterns leads to the proposal of the so-called *Collaborative Learning Flow Patterns* (CLFPs). CLFPs represent common techniques, used by practitioners, in structuring the flow of broadly-accepted types of activities

Table 1
CLFP structure and its application to two collaborative learning common techniques

Facet	Explanation	Example #1	Example #2
Name	Name of the CLFP	Pyramid	Jigsaw
Problem	Learning problem to be solved by the CLFP	Complex problem, usually without a specific solution, which resolution requires the achievement of gradual agreement among all participants	Complex problem which resolution requires handling and/or collecting information that can be easily divided into disjoint sets and that can be used for the resolution of independent subproblems
Example	A real-world learning activity capable of being structured according to the CLFP	Collaborative proposal of the design of a computing system where each participant contributes with a design that is subsequently compared with other contributions and refined	Collaborative design of a computing system where the study of each subsystem is assigned to a particular participant
Context	Environment type in which the CLFP could be applied	Several participants facing the collaborative resolution of the same problem	Several small groups facing the study of a lot of information for the resolution of the same problem
Solution	Description of the proposal by the CLFP for solving the problem	Each individual participant studies the problem and proposes a solution. Groups of participants compare and discuss their proposals and, finally, propose a new, shared solution. Those groups join in larger groups in order to generate new agreed proposals. At the end, all the participants must propose a single agreed solution	Each participant in a group (jigsaw group) studies a particular subproblem. The participants of different groups that study the same problem meet in an "Expert Group" for exchanging ideas. At last, jigsaw group participants meet to solve the whole problem. Each participant contributes with his "expertise"
Actors	Actors involved in the collaborative activity described by the CLFP	- Teacher - Learner - Evaluator	- Teacher - Learner - Evaluator
Types of Tasks	Types of tasks, together with their sequence, performed by the actors involved in the activity. (NOTE: due to space restrictions only types of tasks performed by learner and teacher are shown)	Learner: 1. Access to the information 2. Individual study of the problem 3. Individual solution proposal [REPEAT 4. Group formations 5. Group discussion 6. Common solution proposal] (Until only one group remains) 7. Process self-evaluation Teacher: 1. Global problem definition 2. Provision of useful information 3. Group dimensioning 4. Decisions about control of time 5. Activity progress monitoring 6. Result evaluation	Learner: 1. Access to the information related with the subproblem 2. Individual study of the subproblem 3. Subproblem discussion in the experts group 4. Problem resolution in the jigsaw group 5. Result proposition 6. Process self-evaluation Teacher: 1. Global problem definition 2. Division of the problem in subproblems 3. Creation of jigsaw groups 4. Assignment of subproblems 5. Provision of useful information 6. Floor control system establishment 7. Decisions about control of time 8. Activity progress monitoring 9. Result evaluation
Types and structure of Information	Description of the types of information identified in the collaborative activity and how they are related	- Input information needed for global problem resolution - Intermediate resolution proposals - Global problem resolution proposal - Correct global problem resolution (optional)	- Input information needed for global problem resolution - Partial information assigned to subproblems - Subproblem resolution proposal - Global problem resolution proposal - Correct global problem resolution (optional)
Types and structure of Groups	Description of the types of groups of learners identified and how they are related	- Growing pyramid groups	- Jigsaw groups - Experts groups in charge of subproblems

commonly used in collaborative learning scenarios. CLFPs are supposed to be specified by collaborative learning practitioners and represented according to a formalism shown in Table 1.

As it can be appreciated, CLFPs do not contain technical information of any kind. Nevertheless, software developers can use CLFPs for identifying which types of different functional blocks might eventually be needed in order to support the types of tasks performed in the collaborative learning scenarios that are compliant with a particular CLFP. An important subset of these blocks could potentially be reused in the support of several of those scenarios. For instance (see Table 1), for those collaborative learning scenarios based on the *jigsaw* CLFP, CSCL blocks providing partial access to the base information, discussion support, process self-evaluation, etc. are always needed.

On the other hand, common techniques collected in the form of CLFPs constitute another starting point for collaborative learning practitioners to generate new collaboration scripts. This fact would enable new scenarios in which adequate computer-based authoring tools would guide educators in the process of generating collaboration scripts by fulfilling the prescriptions of existing CLFPs.

These uses of CLFPs cannot be achieved without a suitable formalization of the natural language-based representation sketched in Table 1. In this context, the authors propose the use of IMS-LD, introduced in section 2, so as to formalize CLFPs. Using IMS-LD, a CLFP can be understood as an "incomplete" learning design that has to be customized in order to generate a complete one. An IMS-LD definition of a CLFP includes the description of group-based services in which some of their characteristics are not specified. Section 6 will show an example of an IMS-LD based specification of a CLFP.

4. Grid Services as the Technological Basis for Tailorable Collaborative Learning Systems

The term *grid computing* [13] is commonly used to refer to a large-scale infrastructure that allows the sharing of both software and hardware distributed heterogeneous resources [14]. In analogy with the electric power grid that provides pervasive access to electric power, the computational grid provides transparent and ubiquitous access to software and hardware resources.

Recently, the emergence of OGSA as the *de facto* standard [15] for grid middleware has generated a

noticeable shift towards a service-oriented architecture [5]. Following OGSA, all resources in a grid must be offered in the form of a *grid service*, which can be regarded as a web service with some additional features including instance creation, lifetime management, notifications and security [5].

More specifically, a grid service is a software or hardware resource offered by a third-party provider that is exposed through a standard interface adhering to OGSA specifications. The creation of an instance of a given grid service can be requested to its corresponding *grid service factory*, which is in turn another grid service. The grid service instance can then be invoked using standard protocols. Providers typically publish their services in well-known directories thus enabling service discovery.

A grid could thus be employed in order to provide tailorable collaborative learning systems with a large pool of tools (the functional blocks mentioned in the previous sections) offered by third-party providers in the form of OGSA-compliant grid services [16]. This way, collaborative systems would be able to select and integrate grid service tools within a customized collaborative learning environment following the specifications of an educator (e.g. in the form of a IMS-LD document).

The use of grid services for the integration of customized learning environment helps to tackle three important problems of CSCL systems that have been identified in literature. First, CSCL tools are difficult to integrate within learning environments due to their lack of clear standardized interfaces and incompatibility between conceptual models and design foundations [17]. In this sense, OGSA defines a set of conventions and uniform semantics that support dynamic integration of grid services within larger applications.

Besides, scarce reuse of educational software owes to the use of application building blocks representing low level abstractions that do not fit with the mental model of educators, which often use these blocks to build customized applications [18]. The use of grid services would thus promote software reuse since they represent high-level abstractions that can be employed to assemble customized collaborative learning environments.

Finally, the technification problem refers to the need of technical skills that makes it difficult for teachers and students to use learning systems [19]. The fact that OGSA service-oriented philosophy pushes tool deployment and set-up responsibilities to the grid service provider contributes to avoid this problem.

It is also noteworthy that grid middleware enables providers to supply to the grid community any tool requiring supercomputing capabilities or specific hardware resources in the form of grid services. This way, it would be possible to integrate customized environments

in order to support collaborative learning in many scenarios in which such tools are needed.

For instance, consider a surgery school where students learn collaboratively to operate. In such a scenario, supercomputing capabilities are required in order to compute high-quality visualizations of a complex human body model, which is collaboratively manipulated by students in real time, and to display the computation result on remote screens. CoVis [20] is an example of a collaborative system requiring supercomputing capabilities.

Tools requiring specific hardware resources are employed in scenarios such as a Computer Architecture course in which students collaborate to decide the best computing solution for the requirements posed by a given customer. In order to support this scenario, a learning system should provide not only collaborative tools such as debate and voting tools, but also benchmarking tools that use specific hardware resources such as the different real machines to be tested by students. In [21] an example of a collaborative learning system that integrates specific hardware resources, micro-robots, is also described.

However, it must be noticed that the use of grid services is not restricted to applications requiring supercomputing or specific hardware capabilities. Applications without such requirements can be also integrated following a service-oriented approach using grid services.

Additional advantages for collaborative learning systems derive from grid's large scale and geographically distributed nature. The large scale of the grid may allow the collaboration between a large number of single/group participants. Though in purely synchronous collaboration the number of participants is usually low, in scenarios involving only asynchronous interactions, or a mixture, a high number of participants may be involved. For example, applications allowing children from different schools to collaborate asynchronously in order to publish and consult an electronic magazine may involve a large number of users at different locations in a wide area.

The wide geographical distribution of grid resources should enhance the participation of users from very distant places. Again, many CSCL applications have their users co-located, but others [22] such as distance universities have users spread along one country or even a larger area.

5. Towards a Grid-Based Tailorable Collaborative Learning System

As it was shown in section 2, an IMS-LD document may be employed in a collaborative learning system as a *collaboration script* that structures interactions between students in order to enhance collaborative learning.

Besides, that same IMS-LD document describes the CSCL blocks that should be integrated within a single collaborative learning environment in order to properly support the learning scenario defined in it. CLFPs, described in section 3, constitute some means to identify such blocks that are potentially reusable in several learning scenarios.

Finally, in section 4, we discussed the suitability of using grid services as the enabling technology for implementing those building blocks to be potentially integrated in a tailorable CSCL system.

In this section, we outline such a grid-based tailorable CSCL system that would bring together the benefits of IMS-LD scripting for the detailed description of a learning scenario as well as the computational tools needed to support it, along with OGSA grid services for easy assembly of customized collaborative learning environments.

This system is intended to enable educators to provide the description of a collaborative learning scenario in an IMS-LD document. Such scenario would thus be expressed in terms of a sequence of activities to be performed by students, as well as a generic description of the tools that make up the collaborative environment that should be available for the realization of each activity. IMS-LD documents could be stored in the system for later retrieval and use.

The educator would then choose one of the existing learning designs, so that the system could look for suitable tools supplied by third-party providers in an OGSA-based grid according to the selected IMS-LD document. Particularly, grid service registries would be queried for such purpose. In case more than one tool service is found matching a tool specification of the IMS-LD document, the educator would be able to opt for any of them.

Next, the educator would provide the list of participants allowed to join the execution of the learning design (i.e. the realization of a collaborative learning scenario) as well as the roles they should play in such execution. Once this operation is completed, the educator would be able to launch the execution of the learning design and students could therefore join this execution and start to collaborate. Fig. 1 shows this sequence of interactions.

During execution, the system would determine the sequence of activities to be performed by each participant leading to the desired learning objectives according to the IMS-LD document being interpreted. This includes the identification of the different roles involved in the learning unit execution and the necessary tools to perform the activities. In order to achieve this, an IMS-LD engine similar to Cow [7] or CopperCore [23] could be employed.

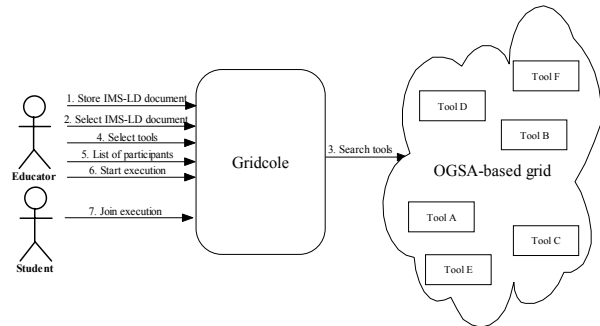


Figure 1. Typical interactions of educator and student users with the proposed system before execution of a learning design

Besides, the system would request from the corresponding factories the creation of instances of the grid services needed to support a given activity. For each participant, the system would provide an application desktop in which the collaborative environment defined for the current activity is reified. The desktop should integrate the graphical interface panes of grid service tools supporting the realization of current activity. Participants could interact with grid services through these interfaces in order to fruitfully collaborate with other users by means of collaborative tools, or to work with non-collaborative tools.

The outlined system can be said to enable tailoring in the sense described in section 2. More specifically, the system would allow easy integration of suitable tools within a single environment in order to support a given collaborative learning scenario. In addition, it would also integrate suitable user interfaces according to the roles played by each participant.

Furthermore, this system might allow the integration of tools requiring supercomputing capabilities or specific resources thus enabling the support of new collaborative learning scenarios. However, it must be pointed out that the use of this type of tools would be optional, since tools without such requirements can also be offered as OGSA-based grid services and therefore integrated by the system. Moreover, the customized collaborative learning environments integrated by this system could also profit from the advantages of grid's large scale and geographical distribution identified above.

6. An Illustrative Example

This section further illustrates how IMS-LD specification can be employed by educators to describe a real collaborative learning scenario requiring tools with special hardware needs. Besides, we discuss the suitability of the collaborative learning environment that can be generated, by the grid-based tailorable CSCL system proposed above, in order to support such scenario.

6.1 Educational Context

The collaborative learning scenario considered in this section is intended to be applied in a course on *Computer Architecture* for undergraduate students at our University. This course is organized around a computer architecture design and evaluation project which in turn is divided in three subprojects (see [24] for details). In this project, students organize themselves in groups of four pairs and collaboratively play the role of *consultants* that have to advise a given *customer*, which is played by the teacher, on a computing solution (machine, operating system, software, network, etc.).

The scenario considered here concerns the first subproject, in which students get to know the customer, model the customer's presumed computational load by mixing standard benchmarks, test real machines using the benchmarks, and finally make a recommendation to their client. This subproject pursues clear learning objectives: on the content side, it is expected that students learn how to use benchmarks, and get a quantitative impression on a few real machines (with different CPUs, memories, etc.); on the skills side, several abilities, such as interpreting and selecting information, arguing, and taking compromise solutions are promoted. The subproject lasts for six two-hour sessions on approximately four weeks.

6.2 Scenario Description

The collaborative learning scenario described here has been designed by teachers of the course in which it is to be applied. The design of the scenario is partially based on the *jigsaw* CLFP (see Table 1). The activities, tools and contents defined for this scenario are briefly outlined next.

For the first activity, students should *study customer needs* while the educator should in turn play the role of the client in order to *clarify customer needs*. This activity is supported by documentation collecting client requirements, a collaborative concept map tool and a debate tool.

In the following activity, students *model the computational load* of the customer with a voting tool that assists the decision-making process and a collaborative questionnaire tool that allows filling in some required forms. In the next activity, students will *distribute* four groups of different real *machines* among them, so that each student benchmarks a group of machines. Here, a collaborative task assignment tool and a chat tool will be employed.

Next, the well-known jigsaw collaboration pattern [6] (also formalized as a CLFP in Table 1, section 3) is applied. First of all (see learners' *type of task* #1 and #2

for CLFP *jigsaw* in Table 1: "Access to the information related with the subproblem" and "Individual study of the subproblem"), students will *benchmark* those machines that have been assigned to them, collecting the results and studying the documentation of the benchmarks and the machines. Benchmark documentation and several benchmarking tools that run in computers with different architecture are required to support this activity.

Students who have benchmarked the *same machine* debate the suitability of such machine for their customer, according to benchmark results (learners' *type of task* #3 for CLFP *jigsaw* in Table 1: "Subproblem discussion in the experts group"). This activity will be supported by a debate tool and a chat tool. Finally, students have to *debate* the results with other members of their group that have benchmarked *different machines* (learners' *type of task* #4 for CLFP *jigsaw* in Table 1: "Problem resolution in the *jigsaw* group"). As a result, each group should generate a technical report presenting and arguing the best solution for their customer (learners' *type of task* #5 for CLFP *jigsaw* in Table 1: "Result proposition"). This activity will be supported by a debate tool and a collaborative text editor tool. All these tools, are potentially reusable in other collaborative learning scenarios based on the *jigsaw* CLFP. Of course, those tools that are closely related to the particularities of the scenario are more difficult to reuse (e.g. the benchmarking tool). But those tools that are more generic (e.g. chat tool, debate tool, collaborative text editor tool,...) could be easily reused in other *jigsaw*-based scenarios.

6.3 Discussion

Educators could easily tailor the grid-based collaborative learning system proposed above by means of an IMS-LD document that describes the unit of learning corresponding to the scenario considered in this section. Table 2 illustrates the three-stage process for obtaining this example unit of learning (see section 2) using the *jigsaw* CLFP as a starting point. This unit of learning consists of a customized CLFP-based learning design and a set of particular resources.

The first column of Table 2 illustrates part of the IMS-LD formalization of the *jigsaw* CLFP. The learning flow dictated by this CLFP is included in a play of the IMS-LD method: one act of the play is devoted to the individual study of a subproblem; the discussion of the subproblem by expert groups takes place in the following act; and a third act is devoted to the *jigsaw* group *debate*, in which participants are supposed to solve the whole problem. These acts correspond to learners' *types of tasks* 2/3/4 for *jigsaw* CLFP in Table 1.

The second column of Table 2 represents the teacher customization of the *jigsaw* CLFP IMS-LD description

Table 2

Excerpt showing the process for obtaining the jigsaw CLFP-based unit of learning that describes the learning scenario considered in section 6. A sample definition of an activity and an environment can be found under `<learning-activity>` and `<environment>` tags respectively. The description of learning flow is shown under `<method>` tag

1. IMS-LD description of the jigsaw CLFP (The description of a common technique)	2. A jigsaw CLFP-based learning design (The teacher customizes the previous description of the CLFP for the example scenario)	3. A jigsaw CLFP-based unit of learning (The binding of the previous learning design for the example scenario with concrete CSCL tools)
<pre> <learning-design identifier="CLFP-jigsaw" uri="" level="B"> <components> ... <learning-activity identifier="LA-discuss-experts"> <environment-ref ref="E-discuss-experts"/> <activity-description> <item identifier="" /> </activity-description> </learning-activity> ... <environment identifier="E-discuss-experts"> <service identifier="S-discuss-experts"> <conference conference-type=""> <participant role-ref="R-expertLearner"/> <item identifier="" /> </conference> </service> </environment> ... </components> <method> <play identifier="PLAY-CLFP-jigsaw"> <act> <role-part identifier="RP-individual-work"> <role-ref ref="R-expertLearner"/> <learning-activity-ref ref="LA-subproblem-study"/> </role-part> <role-part identifier="RP-expert-groups"> <role-ref ref="R-expertLearner"/> <learning-activity-ref ref="LA-discuss-experts"/> </role-part> <role-part identifier="RP-jigsaw-groups"> <role-ref ref="R-jigsawLearner"/> <learning-activity-ref ref="LA-problem-resolution"/> </role-part> </act> </play> </method> </learning-design> </pre> <p>The CLFP dictates an individual study of a subproblem, then this subproblem is discussed in a group of experts, and finally a jigsaw group meets to solve the problem</p>	<pre> <learning-design identifier="CLFP-jigsaw" uri="" level="B"> <components> ... <learning-activity identifier="LA-discuss-experts"> <environment-ref ref="E-discuss-experts"/> <activity-description> <item identifier="" /> </activity-description> </learning-activity> ... <environment identifier="E-discuss-experts"> <service identifier="S-discuss-experts"> <conference conference-type="synchronous"> <participant role-ref="R-expertLearner"/> <item identifier="" /> </conference> </service> </environment> ... </components> <method> <play identifier="PLAY-CLFP-jigsaw"> <act> <role-part identifier="RP-individual-work"> <role-ref ref="R-expertLearner"/> <learning-activity-ref ref="LA-benchmarking"/> </role-part> <role-part identifier="RP-expert-groups"> <role-ref ref="R-expertLearner"/> <learning-activity-ref ref="LA-discuss-experts"/> </role-part> <role-part identifier="RP-jigsaw-groups"> <role-ref ref="R-jigsawLearner"/> <learning-activity-ref ref="LA-choose-machine"/> </role-part> </act> </play> </method> </learning-design> </pre> <p>In the example scenario, which is synchronous, the subproblem is to test a group of the available real machines using benchmarks. In this case the solution of the problem is to decide which is the best machine for a specific customer</p>	<pre> <imscp:manifest ...> <imscp:organizations> <imsl:learning-design identifier="CLFP-jigsaw" uri="" level="B"> <imsl:components> ... <imsl:learning-activity identifier="LA-discuss-experts"> <imsl:environment-ref ref="E-discuss-experts"/> <imsl:activity-description> <imsl:item identifier="RES-discuss-guideless"/> </imsl:activity-description> </imsl:learning-activity> ... <imsl:environment identifier="E-discuss-experts"> <imsl:service identifier="S-discuss-experts"> <imsl:conference conference-type="synchronous"> <imsl:participant role-ref="R-expertLearner"/> <imsl:item identifier="RES-GSIC-chat"/> </imsl:conference> </imsl:service> </imsl:environment> ... <imsl:components> <imsl:method> <imsl:play identifier="PLAY-CLFP-jigsaw"> <imsl:act> <imsl:role-part identifier="RP-individual-work"> <imsl:role-ref ref="R-expertLearner"/> <imsl:learning-activity-ref ref="LA-benchmarking"/> </imsl:role-part> <imsl:role-part identifier="RP-expert-groups"> <imsl:role-ref ref="R-expertLearner"/> <imsl:learning-activity-ref ref="LA-discuss-experts"/> </imsl:role-part> <imsl:role-part identifier="RP-jigsaw-groups"> <imsl:role-ref ref="R-jigsawLearner"/> <imsl:learning-activity-ref ref="LA-choose-machine"/> </imsl:role-part> </imsl:act> </imsl:play> </imsl:method> <imsl:learning-design> <imscp:organizations> <imscp:resources> <imscp:resource identifier="RES-discuss-guidelines"/> <!-- It exposes what pupils must discuss about their benchmarking results --> <imscp:resource identifier="RES-GSIC-chat"/> <!-- It is a concrete tool that support the discussion within this learning activity --> </imscp:resources> </imscp:organizations> </imsl:learning-design> </imsl:components> </imsl:learning-design> </imsl:learning-design> </imscp:organizations> </imscp:manifest> </pre> <p>Particular resources used in a particular performance of the example scenario</p>

for the depicted example: a particular jigsaw CLFP-based learning design. This learning design specifies that, in this case, the problem involves deciding which is the best machine for a specific customer and the subproblem entails testing a group of the available real machines. Besides, it also states that expert groups debate synchronously.

Once the teacher has determined the binding of this learning design to actual tools or documents, the unit of learning is achieved. The third column of Table 2 shows how two specific resources are referenced within the CLP-based learning design. One of them is a document that includes the guidelines learners have to follow during the discussion. The other one is a specific chat tool that enables this debate.

As a drawback, it may be considered that the edition of IMS-LD documents is currently difficult for most educators, since it is encoded in XML format. However, authoring tools announced by IMS-LD community will tackle this problem.

The support of this scenario would require a number of tools that are not likely to be found in a single non-tailorable collaborative learning system, while the system

we outlined in section 5 could be tailored to integrate all tools as long as they are offered in the form of OGSA-grid services by any provider.

Following the specifications provided in this document, our candidate system would locate suitable tools to support the scenario in an OGSA-based grid. Particularly, a tool with special hardware requirements would be needed: the benchmarking tool (not shown in the IMS-LD excerpt of Table 2) should be executed in machines with different architecture that may be considered of interest by the educators. The use of grid services as building blocks for the assembly of customized collaborative learning environments assures that such a tool would be easily integrated by the outlined system.

It is also noteworthy that in the previous course configuration, benchmarking was only performed in machines owned by our University. As a consequence, the range of architectures that could be chosen for evaluation was not as broad as teachers would like. In this sense, the use of a system as the one outlined in section 5 would increase opportunities for students to benchmark remote machines owned by grid service providers.

7. Conclusions and Future Work

This paper has proposed the use of collaboration scripts, formalized by means of IMS-LD, as the way of collecting educators requirements needed to guide the selection and integration of CSCL software blocks into tailored CSCL solutions. Also, the paper has introduced the use of CLFPs, also formalized by means of IMS-LD, as a suitable way of collecting common techniques in collaborative learning that can help software developers to identify potentially reusable CSCL software blocks. Furthermore, the paper has justified and proposed the use of grid services technology in order to implement those CSCL blocks. Grid Services add to the CSCL field supercomputational capabilities, heterogeneous hardware support, and scalability. In addition to their suitable abstraction level, grid services also contribute to the reduction of the technification problem that characterize CSCL solutions. Finally, the paper has outlined a grid-based, tailorable CSCL system by applying all the above ideas. The use of this system has been illustrated with an example of a collaborative learning environment that would benefit from several of these properties.

There are several short and medium term research topics that can be derived from the contributions presented in this paper. Two of them are worth mentioning: first of all, we are currently developing a prototype, called *GridCole*, of the outlined grid-based CSCL system that we plan to use so as to validate our different proposals; secondly, we are also working on an authoring tool that would help educators in the process of generating learning designs in the form of IMS-LD documents. That tool will use CLFPs as a guide for requesting educators' design decisions to be contained in the resulting IMS-LD specification.

Acknowledgements

This work has been partially funded by European Commission Project EAC/61/03/GR009 and Spanish Ministry of Science and Technology Project TIC2002-04258-C03-02. The authors would also like to thank the rest of "Intelligent & Cooperative Systems Research Group" at the University of Valladolid for their support and ideas.

References

- [1] T. Koschmann, *CSCL: theory and practice of an emerging paradigm*, (Mahwah, NJ, USA: Lawrence Erlbaum, 1996).
- [2] A. Morch, Three levels of end-user tailoring: customization, integration and extension, *Proc. of the 3rd Decennial Aarhus Conference*, Aarhus, Denmark, 1995, 41-45.
- [3] IMS learning design information model v1.0, final specification, <http://www.imsproject.org/learningdesign/index.cfm>.
- [4] Y. A. Dimitriadis, J. I. Asensio, A. Martínez, & C. A. Osuna, Component-based software engineering and CSCL in the field of e-learning, *Upgrade (digital journal of European Professional Informatics Societies), special issue on e-learning - borderless education*, 4(5), 2003, 21-28 .
- [5] I. Foster, C. Kesselman, J. M. Nick, & S. Tuecke, The physiology of the grid. In F. Berman, G. Fox, & A. Hey (Eds.), *Grid computing: making the global infrastructure a reality* (Chichester, UK: John Wiley & Sons, 2003, 217-249).
- [6] P. Dillenbourg, Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (Heerlen, Open Universiteit Nederland, 2002, 61-91).
- [7] T. Vantrouys & Y. Peter, COW, a flexible platform for the enactment of learning scenarios, *Proc. of the 9th International Workshop on Groupware, CRIWG 2003*, Grenoble, France, 2003, 168-182. Lecture Notes in Computer Science, 2806.
- [8] D. Hernández, *From IMS-LD to MDA: approaches to component-based CSCL applications modeling (in Spanish)*, MSc. Thesis, University of Valladolid, Valladolid, Spain, 2003.
- [9] D. Hernández, J. I. Asensio, & Y. Dimitriadis, IMS learning design support for the formalization of collaborative learning patterns, *The 4th International Conference on Advanced Learning Technologies (to be published)*, Joensuu, Finland, 2004.
- [10] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, & S. Angel, *A pattern language: towns, buildings, construction*, (New York, USA: Oxford University Press, 1977).
- [11] P. Avgeriou, A. Papasalouros, S. Retalis, & M. Skordalakis, Towards a pattern language for learning management systems, *Educational Technology & Society*, 6(2), 2003, 11-24.
- [12] J. I. Asensio, Y. A. Dimitriadis, M. Heredia, A. Martínez, F. J. Álvarez, M. T. Blasco, & C. A. Osuna, From collaborative learning patterns to component-based CSCL applications, *European Conference on Computer Supported Collaborative Work (ECSCW03), Workshop: From Good Practices to Patterns*, Helsinki, Finland, 2003.
- [13] F. Berman, G. Fox, & A. Hey, *Grid computing: making the global infrastructure a reality*, (Chichester, UK: John Wiley & Sons, 2003).
- [14] M. L. Bote-Lorenzo, Y. A. Dimitriadis, & E. Gómez-Sánchez, Grid characteristics and uses: a grid definition, *PostProc. of the 1st European Across Grids Conference*, Santiago de Compostela, Spain, 2004, 291-298. Lecture Notes in Computer Science, 2970.
- [15] I. Foster, C. Kesselman, J. M. Nick, & S. Tuecke, Grid services for distributed system integration, *Computer*, 35(6), 2002, 37-46.
- [16] M. L. Bote-Lorenzo, J. I. Asensio-Pérez, G. Vega-Gorgojo, L. M. Vaquero González, E. Gómez-Sánchez, & Y. A. Dimitriadis, Grid computing and component-based software engineering in computer supported collaborative learning, *Proc. of the International Conference on Computational Science, ICCS 2004*, Kraków, Poland, 2004, 503-506. Lecture Notes in Computer Science, 3036.
- [17] G. Bourguin & A. Derycke, Integrating the CSCL activities into virtual campuses: foundations of a new infrastructure for distributed collective activities, *Proc. of the European Conference on Computer Supported Collaborative Learning, Euro-CSCL 2001*, Maastricht, The Netherlands, 2001, 123-130.
- [18] J. Roschelle, C. DiGiano, M. Koutlis, A. Repenning, J. Phillips, N. Jackiw, & D. Suthers, Developing educational software components, *Computer*, 32(9), 1999, 50-58 .
- [19] M. J. Spector, An overview of progress and problems in educational technology, *Interactive Educational Multimedia*, (3), 2001, 27-37.
- [20] M. K. Ramamurthy, R. B. Wilhelmson, R. D. Pea, Louis M., & D. C. Edelson, CoVis: A national science education collaborative, *Proc. of the American Meteorological Society 4th Conference on Education*, Dallas, TX, USA, 1995.
- [21] C. Despres & S. George, Computer-supported distance learning:

an example in educational robotics, *Proc. of the 9th International PEG Conference*, Exeter, UK, 1999, 344-353.

- [22] J. M. Marquès, L. Navarro, & A. Daradoumis, WWG: a distributed infrastructure for learning in groups, *Proc. of the IEEE Conference: Frontiers in Education 2001*, Reno, NV, USA, 2001.
- [23] CopperCore project website, <http://coppercore.org/>, last visited: 2004.
- [24] Y. A. Dimitriadis, A. Martínez, & B. Rubia, Cooperative learning in computer architecture: an educational project and its network support, *Proc. of the Frontiers in Education Conference, FIE 2001*, Reno, NV, USA, 2001.

Biographies



Miguel Luis Bote-Lorenzo received his M.S. degree in telecommunications engineering from University of Valladolid, Spain, in 2001. 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 CSCL systems and grid computing.



Davinia Hernández-Leo received her M.S. degree in telecommunications engineering from University of Valladolid, Spain, in 2003. She is currently studying towards a PhD degree in telecommunications engineering at the University of Valladolid. Her research interests include CSCL and modeling techniques

applied to the educational field.



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.



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.



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.



Guillermo Vega-Gorgojo received his M.S. degree in telecommunications engineering from the University of Valladolid, Spain, in 2000. Currently he is a teaching assistant at the same University. His current research interests are the use of semantic web and grid technologies to support educational settings.



Luis Miguel Vaquero-González received his engineering degree in electronical systems in 2001, and his MSc degree in telecommunications engineering in 2003, both from University of Valladolid, Spain. He is currently studying toward the PhD degree in telecommunications engineering at University of Valladolid. His research interests

include CSCL, bioinformatics and grid computing.