An Architecture Combining IMS-LD and Web Services for Flexible Data-Transfer in CSCL

Ioannis Magnisalis and Stavros Demetriadis, *Member, IEEE*

Abstract—This article presents evaluation data regarding the MAPIS3 architecture which is proposed as a solution for the data-transfer among various tools to promote flexible collaborative learning designs. We describe the problem that this architecture deals with as "tool orchestration" in collaborative learning settings. This term refers to a situation where data relevant to a collaborative learning activity need to be forwarded to and processed by various learning technological tools (e.g. a forum, a pedagogical agent, a service or a software component that provides a specific functionality, etc.), in order for the collaborative activity to be efficiently represented and teachers' pedagogical level decisions implemented. To facilitate data-transfer among the various tools and accomplish flexible interventions during runtime, the architecture employs a key component ("mediator component") which makes use of an IMS-LD based representation of the activity. By implementing the architecture tradeoff analysis method in three case studies, evaluation data regarding the proposed architecture have been recorded and are presented in this paper. Targeted stakeholders (learners, teachers, and developers) provided valuable insights on the capacity of the architecture to efficiently facilitate tool orchestration during the realization of a flexible IMS-LD based course. Additionally, limitations of IMS-LD are discussed and suggestions are made on how to tackle these constraints and increase flexibility during tool orchestration in collaborative learning deployment.

Index Terms—Software tool orchestration, IMS-LD, computer-supported collaborative learning (CSCL), software architecture evaluation, web services

1 Introduction

Collaborative learning is a major contemporary learning paradigm, engaging students in multiply (cognitively, socially, emotionally) beneficial situations where peer interaction constitutes the key learning mechanism [1]. Computer-supported collaborative learning (CSCL) consequently promotes the integration of multifaceted technological tools in the educational setting, aiming to trigger, structure and otherwise support productive peer interactions [2]. Research has consistently emphasized that collaborating students might fail to engage in productive learning interactions when left unsupported (script free case) [2]. Collaboration scripts have been proposed as a remedy and means to structure the collaborative activity with reported results of improved learning outcomes and engagement of students in fruitful learning interactions (e.g. [1], [2]).

CSCL scripting (orchestrating [3]) is a complex, nonlinear endeavor [4] and, although virtual learning environments (VLEs) have been developed to support scripting (for example LAMS [5]), in several situations it is not possible to enact a complex script using only one single technological environment or tool, also supporting design decisions at pedagogic-level. For example, there is not any single VLE that (A) performs group formation based on learners' profiles, and (B) enacts a Jigsaw script [6] with the assigned groups (pedagogic-level decision). In this case, a system performing group formation can be employed to accomplish objective A, while a different system should be used to computationally support the Jigsaw script enactment. In order to achieve an automated technological support of the whole learning activity, some type of system interlinking needs to be established, so that data from the former (for example, students' profile and group composition) may be fed transparently to the latter, without overloading the educator/user with unnecessary manual data operations.

In many similar situations, the flexible technological support of complex collaboration scripts makes necessary the interlinking of different technological tools to support different phases of the scripted activity. These tools, however, employ—most probably—different data models, and thus, some kind of technically non-trivial intervention is necessary for the accomplishment of a transparent dataflow and tool interconnection. We call this intervention "tool orchestration" and claim that it is always a key consideration in CSCL ecosystems, where a major objective is to guide and support peer interactions by seamlessly connecting and transparently transferring single user and peer interaction data among the various mediating tools used by the teacher and student groups.

From our perspective, CSCL currently lacks a mature framework (evaluated and well established) for tool orchestration. To address the problem, in this work we present and evaluate a back-end architecture called MAPIS3 that guarantees the seamless data-transfer among interconnected tools. When it comes to accomplishing flexible connectivity of different technological tools, then the MAPIS3 architecture is one of the two proposed solutions in the literature (the other one is [7]). This, we believe, justifies the innovative character of our work.

• *The authors are with the Aristotle University of Thessaloniki, Thessaloniki 541 24, Greece.*
E-mail: imagnisalis@yahoo.gr, {sdemetri, imagnisa}@csd.auth.gr.

Manuscript received 8 Feb. 2015; revised 6 Dec. 2015; accepted 12 Dec. 2015. Date of publication 25 Dec. 2015; date of current version 16 June 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TLT.2015.2512604

In the following, we first provide a background on tool orchestration in CSCL, and then explain the advantages and limitations of the IMS-LD modeling specification [8] that we employ as the key element of our solution. Next, we present our proposed MAPIS3 architecture and how we applied architecture tradeoff analysis method (ATAM) [9] to analyze architecture relevant evaluation data. Three scenario-based case studies have been implemented, engaging learners, teachers and system developers. We document the value of the proposed architecture based on data emerging from the qualitative and quantitative analysis of these cases.

2 Background

2.1 Tool Orchestration in CSCL

While collaborative learning is important for students [1], collaborators might fail to engage in productive learning interactions when unsupported [2]. To trigger and sustain fruitful peer interaction scripted collaboration techniques have been proposed, encapsulating the idea that collaboration can be structured and guided by well-designed didactic scenarios (collaboration scripts) ([2], [3]).

This perspective has certainly increased the complexity of the collaborative learning design, i.e. the process of producing computerized representations of CSCL scripts. Throughout the text we use the term "learning design" implying that it concerns the collaborative aspects and is identical to the term "Learning Design" (with capitals) or simply LD. Thus, while LD is often used to denote the authoring process, here the emphasis of the term is on deploying and making available to students a computerized representation of a CSCL didactic scenario ([4], [5]).

Several technological tools (e.g. Recourse, Reload, Collage, LDShake, ILDE etc. [10], [11]) are currently available to teachers to help them implement/author LDs of varying degrees of flexibility. However, only very recent efforts like IDLE have evolved from prototype to production level so as to support actual delivery of courses. A flexible LD may require the interoperable linking of several learning enactment tools. In the following, we use the term "flexible learning designs" to refer to the type of designs that need to be highly flexible in order to successfully adapt to various changing parameters in the learning setting (and not only adaptive to user or CSCL environment characteristics). The proposed architecture may provide an efficient solution in various CSCL advanced designs that require flexible pedagogy, representation and coordination of complex situations like the following:

a) User grouping (shown in case study 3) and regrouping (not shown in this work but could be an extension of case study 3),
b) User—or device-adjusted—interface adaptation (as presented in case studies 1 and 2)
c) Dynamic (on-the-fly) modification of learning flow or LD structure (as in case study 3), and
d) Adaptive content presentation (e.g. information filtering). An example is the adaptive role assignment (and provision of adjusted guidelines), as in case study 3.

We do not use the term adaptive as we consider flexible a more general term that encompasses adaptive behavior; that is, adaptive systems in educational systems often refer to the subset of flexibility that takes into account user characteristics [12].

The term "orchestration" appears ([3], [13]) in computer science literature with a more technically-oriented connotation (usually relevant to web service technologies). In our work, however, we use "tool orchestration" from a pedagogical perspective, referring to the capability of computationally representing complex CSCL activities by coordinating data-flow between linked tools. Thus, we call "tool orchestration" the seamless integration of various CSCL tools into a single learning design, which provides a solution at two levels: (a) at a physical level it guarantees successful data-flow among tools; (b) at a pedagogical level it allows teachers and developers to support the script prescribed peer interactions at all phases, using appropriate technological tools. Thus, "CSCL tool orchestration" emphasizes the need for a seamless integration of the various CSCL tools necessary to support the learning activity in a user-transparent and unobtrusive way.

2.2 Proposed Solutions Based on IMS-LD

Solutions for the tool orchestration problem have already been proposed in literature. Most of them focus on the IMS-LD modeling specification, as a de-facto standard for LD. These solutions are strongly related to the way that researchers employ LD using available IMS-LD compatible tools to model the collaborative activity (for example, IMS-LD tools for design and deployment like ReCourse and SLeD [14]). In this section, and in order to help the reader understand the difference between other proposed solutions and the MAPIS3 architecture, we first discuss concisely the IMS-LD specification and its limitations, and then, we review shortly these similar efforts.

IMS-LD is primarily a modeling specification using the metaphor of a theatrical play to describe a teaching-learning process ([15], [16]). Its main components are: metadata, roles, acts, environment, role-part (i.e. activities of the actor, who does what, when and how), sequence of activities within a role-part, conditions and notifications. Through an IMS-LD authoring tool one can formally express a unit of learning (UoL), that is, a complete, self-contained unit of education or training, such as a course, a module, a lesson, etc. However, IMS-LD has its limitations when flexible scripts are to be implemented. The problem, as we see it, relates strongly to the fact that the currently available standard IMS-LD specification [3] is not capable of expressing and manipulating the computationally complex data structures and processes (algorithms/computations) that have to be applied in order to implement data-flow among tools used/orchestrated within a LD. A simple exemplary case could be a regrouping in a phase/activity of the collaborative scenario based on individual data contrasted with group and whole class data. For example, while IMS-LD is capable of representing simple processing such as aggregation, more complex structures—like the 'for' loop that most programming languages support—are not available.

Although a CSCL script can be realized in various LD formats and proprietary systems, IMS-LD is the most widely cited specification for formalizing CSCL scripts [5]. However, IMS-LD has triggered an ongoing debate in the community regarding its potential for modeling flexible behavior necessary for the tool orchestration concept already discussed (for a review see [17]). On one hand, there are studies that propose totally new languages (e.g. [18], [19]) to allow for flexible learning designs. On the other hand, specific studies (e.g. [20]) claim that some aspects of adaptive system operation can be modeled and supported by the current IMS-LD capabilities.

Various proposals attempt to reach higher levels of LD flexibility while keeping IMS-LD as a core constituent. Some studies propose modifications to the IMS-LD standard, either by extending it (e.g. [21]) or by introducing a modified element subset of the standard (e.g. [22]). Other studies combine IMS-LD and grid services trying to extend IMS-LD capabilities (e.g. [7], [20]). Finally, there are studies adding components to an engine running IMS-LD based CSCL scripts (e.g. [23], [24]). Recently, IMS has developed the learning tools interoperability (LTI) specification [25], to allow remote tools and content to be integrated into a learning management system.

Nevertheless, most of the above efforts do not focus on the flexibility-adaptivity needed in CSCL scenarios, but mainly provide solutions for reusing existing tools (IMS-LD compatible or not) without, however, establishing any data-transfer connection between tools. By contrast, MAPIS3 advances the IMS-LD implementation model to deal with flexibility issues. The underlying idea is that script enactment should be flexible enough to adjust during runtime the process prescribed by the script itself. The efficiency of the MAPIS3-based implementation becomes particularly evident in cases when the flexible enactment of a script is not possible with a single VLE. In such cases, it is necessary to transfer data among tools (IMS-LD compatible or not) which are required for script enactment. An exemplary case might be one where "model-based scaffolding" is applied (e.g. advising learners based on their modeled preferences, etc.) [4], with one tool recording learner-model data (phase A), and another tool implementing model-based scaffolding (phase B). MAPIS3 supports IMS-LD modeled orchestration of data-flow between the different tools used in phases A and B.

Studies dealing with flexible CSCL scenarios are scarce: a) De la Fuente Valentin et al. [7], which provides a generic architecture (GSI) for extending IMS-LD through specific application interfaces, b) Alario-Hoyos et al. [26], proposing a middleware integration architecture (GLUE!) enabling integration of multiple existing external tools in VLEs (like Moodle), c) the Glue!PS system mentioned in [27], which employs an architecture trying to integrate external component/tools in a CSCL activity through a middle layer service adapter, and d) the study by Palomino-Ramírez et al. [13], which presents LeadFlow4LD, a method to achieve a computational representation of data with tools in CSCL processes in an interoperable and standard way. The GSI [7] and LEADFLOW4LD [13] approaches are similar to MAPIS3 as they focus on the data-flow problem among tools used in CSCL scripts; nevertheless, no LeadFlow4LD-based implementation exists and GSI introduces proprietary IMS-LD elements.

In Table 1 we summarize the findings of our review in the area of IMS-LD based attempts to provide support for flexible tool orchestration. Table 1 classifies these attempts in four major types on the basis of their implementation strategy (see strategy and description columns). Characteristic examples of such attempts are depicted in the 'Systems' column mentioning whether implementation of the proposal exists. Our proposal can be considered as a combination of the first three approaches focusing on re-using IMS-LD standard with no modifications on IMS-LD design and player tools. We facilitate flexible tool orchestration (i.e. providing the means to re-use tools in the learning process and seamlessly connect them with IMS-LD based scripts).

TABLE 1
IMS-LD Based Attempts for Tool Orchestration

Strategy	Systems (Refs.)	Application exists	Description
IMS-LD and external tools	[20], [7]	YES	Minimal IMS-LD modifications for communication between IMS-LD run engine and external tools
Enhance Design Engine	[26], [27], [13]	YES (No [13])	Output LD works in Moodle & Mediawiki (no IMS-LD player)
Enhance Run Engine	[23]	YES	Additions to IMS-LD engine to incorporate external tools
Enhance standard	[24], [19]	YES	Use of external components and calling them with IMS-LD extension tags called Widgets
New Standards	[21], [22], [18]	NO	Propose new standards to support flexible 'tool orchestration'

Concluding, the major shortcoming of IMS-LD with regard to facilitating flexible tool orchestration can be described as follows: "*Flexible CSCL designs/scripts cannot be modeled in IMS-LD terms. The reason is that CSCL scripts require data-transfer among several tools which cannot be modeled and implemented, since IMS-LD does not provide necessary conceptual constructs and modeling structures*".

2.3 Research Motivation

We have argued in previous sections that currently:

a) Tool orchestration is important if flexible CSCL learning designs are to be efficiently supported at technological level. "How do we facilitate realization of flexible CSCL scripts that orchestrate tool data-transfer seamlessly for the learner/teacher?" is a first research motivation.
b) A key requirement for tool orchestration is a modeling tool that provides the basis for tool communication and data-transfer. IMS-LD has been posited as the de-facto standard in the learning design field standard but has limitations. "How do we overcome these shortcomings?" is a major research motivation.

The aforementioned research motivations are reflected in the current study which aims to present a solid architecture upon which the CSCL community can implement flexible (for example, adaptive, intelligent, etc. [12]) systems and IMS-LD based CSCL scripts successfully deploying 'tool orchestration'.

3 The MAPIS3 Architecture

In this section we present and discuss the proposed architecture (MAPIS3). The main objective in developing this architecture is to advance (a) software components reusability, and (b) compatibility with standard specifications (such as IMS-LD and Web services [28]). The idea of the architecture was already presented in [17].

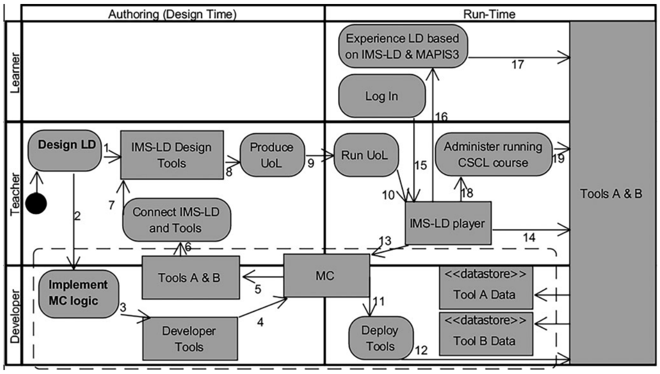

Fig. 1. MAPIS3 Architecture layered view.

3.1 MAPIS3 Requirements

The quality requirements guiding the design of the proposed architecture are as follows:

R1) Applicability: the CSCL learning designs employing MAPIS3 should be able to incorporate most of the available technologies as orchestrated tools under various contexts (e.g. real-time, different place etc.) for the learner/teacher. This requirement has been identified in study [12] and tests the ability of MAPIS3 to facilitate flexible LD deployments with a pedagogical focus. The stakeholders mainly interested in R1 are developers, teachers and learners.

R2) Flexible data-transfer: MAPIS3 should support seamless data-transfer among tools. This requirement, also mentioned in study [12], focuses on support for flexible LD execution while maintaining the learning flow transparent (black-box view of the system-script by learners, glass-box view of the system-script by developers and teachers). The stakeholders mainly interested in R2 are learners and teachers.

R3) Feasibility: MAPIS3 should make it feasible to deploy flexible scenarios at a reasonable cost. MAPIS3 should support modifications such as corrections, improvements or adaptions of the IMS-LD based script. This requirement, inspired by studies [7], [26] and [27], necessitates extensibility and generality. For example, the IMS-LD should be able to communicate with external software components based on open standards. In other words, MAPIS3 should support flexible IMS-LDs and deployment of tool orchestration for teachers. Stakeholders interested are mainly developers.

R4) Reusability: UoLs using MAPIS3 should be reusable in course instances differing in parameters like environment (face-to-face, mobile). Stakeholders interested are mainly developers and secondly teachers ([7]).

3.2 MAPIS3 Architecture Layered View

In Fig. 1 we depict the layered view of the MAPIS3 architecture using a UML 2.0 activity diagram with: a) horizontal swim-lanes among stakeholders and b) vertical swim-lanes consisting of the basic phases of IMS-LD implementation (i.e. design and run-time). Object and activity constructs are used to enforce both the behavioral and component aspects of this diagram. Moreover, for reasons of simplicity we do not use an "end" activity as this process can run iteratively. The dotted line represents the MAPIS3 contribution to an IMS-LD based tool orchestration. Notice that without it we have the normal course of actions to deliver a UoL without data-flow among orchestrated tools, a requirement that can be satisfied if we add the dotted area ("expansion region" in UML terminology). The numbering upon arrows is indicative of a normal course of actions and can change even at run-time. This means that MAPIS3 supports loose coupling and maximum flexibility by simply changing the MC at run-time.

3.2.1 Authoring (Design time)

This phase starts with the teacher who designs (step '1') a CSCL script in an IMS-LD authoring tool (e.g. ReCourse) including: a) activities calling tools in IMS-LD Level A terms (the teacher only links activities with tools and actual URI of the tool to be called at run-time), and b) properties and rules to cater for script flexibility in IMS-LD level B terms. The teacher(s) co-operate (step '2') with the developer and agree upon the tool sequence to be used and the data-flow among tools (A & B), Mediator component (MC) and IMS-LD run-engine.

Note that the developer has to implement a custom MC (step '3') for every tool and combination of tools used in the IMS-LD script. The MC is a web resource and does not impose a standard interface that must be implemented to be usable within MAPIS3. MAPIS3-based MC can be implemented in any high-level programming language (for example, JAVA, PHP, Python, C++, etc.). MAPIS3 only requires from the developer to apply web service technologies so as: a) to access data from tools to which the MC connects and b) to set/get IMS-LD properties. This generally means that a MC can be integrated in any VLE as a widget. However, it is key aspect of MAPIS3 that the MC stays VLE-independent and IMS-LD oriented for at least two reasons: a) scripts compliant IMS-LD can be authored and played by tools following the standard (e.g. Moodle can read IMS-LD scripts), b) pedagogic design decisions can be syntactically supported by IMS-LD Level-B properties and rules, a task that—although not trivial—is semantically necessary for teachers demanding flexible scripting; thus, a teacher can author pedagogic rules, while in the opposite case the teacher would lose this functionality and the developer would additionally have to implement all pedagogic script logic. The developer in step '4': (a) identifies the data-flow between tools as sets of input and output data, (b) designs the communication sequence diagram of the system components to model the interaction of the MC with the tools. No interaction of the MC with the tools occurs at design-time except from the developer who tests the MC implementation (step '5'). Then, the developer builds and tests (step '4' and '5') a MC to interface specific tools (see 'Tool A & B' object in Fig. 1 at design-time). The developer simply has to know from the teacher the name of the properties and the logic under which they are set.

After the MC is built, the developer passes URIs of the MC and tools involved to the teacher (steps '6' and '7'). The authoring system does not have to include special code in the UoL in order for the MC to carry out its functions. The developer passes just the URI of the built MC to the teacher, only when the MC is to provide semi-automatic control by the teacher. Thus, the MC is to be included as a tool in the exact point of the IMS-LD script where the teacher designs the MC functionality to be triggered. The IMS-LD editor interfaces the MC only with a URI link in the sense that the latter needs to access (get and/or set) properties of IMS-LD Level B properties. Depending on the CSCL scenario, and when the MC sets IMS-LD properties, an appropriately teacher designed IMS-LD rule is triggered and an activity behavior is realized (e.g. modified/hidden/shown). All these mean that the author (teacher usually), while authoring the script, has to: a) describe the data-flow model (i.e. input, output and transformation (such as aggregation) of these data) among tools to the developer and b) in terms of script activities include relevant properties and rules to the IMS-LD script. The activity (step '8') of connecting IMS-LD and tools (including the MC) at the current stage of our work is manually executed by the author who decides whether—and where exactly—to include the MC in the script. However, steps have been made (see discussion section) towards: a) promoting tool integration with the facility of data-flow among the tools and b) automating the design process. Finally, the author/teacher produces the UoL to run in a real CSCL setting (step '9').

A *tool* in MAPIS3 is a software component used for a learning activity (such as a Moodle forum, a chat tool, or a web 2.0 tool such as Twitter) that exposes specific functionality through a web service interface. For instance, in an example case (see [17]) a service taking as input the required number of groups performs group formation, stores the results and outputs a message indicating the successful (or not) group formation process. MAPIS3 requires that the tool allow access to the underlying data the tool handles/stores. Open source tools are usually opted for, as even when no such web service interface exists, data can be directly accessed or a new web service interface can be built by a knowledgeable developer.

The *Mediator component* is the core of the MAPIS3 architecture, incorporating the logic of getting requests from an IMS-LD run-time engine for searching and calling a functionality offered by a tool through a published web service. The MC acts as data-transfer intermediary among service tools and IMS-LD player tools. Technically, part of the MC along with the IMS-LD properties and conditions constitute the orchestration layer which caters for fetching or setting data from and to the data layer. At design-time, the learning activities flow is defined by the IMS-LD level-B conditions.

MAPIS3 supports loose coupling and high cohesion among tools used, thus there are no requirements for the IMS-LD editors (and run-time players). Moreover, the practitioners (teachers and developers) do not have to make changes to these applications in order to get a MAPIS3-based system to run. Depending on the case, this tool data-transfer is based on a learning flow controlled either totally by IMS-LD rules and/or by rules shared by the MC and IMS-LD. In technological terms this orchestration layer is implemented by IMS-LD Level-B constructs and MC-programmed control logic. In fact, the MC acts as a facilitator of the adaptive behavior of the whole CSCL script, which is controlled—otherwise orchestrated—by IMS-LD rules. Thus, the MC acts as the connector between IMS-LD and external tools/services, implements the complex parts of synchronizing services and facilitates adaptive behavior controlled by IMS-LD.

3.2.2 Run-Time

As the teacher starts the UoL (step '10'), the developer caters for the tools and MC to be deployed, running and exchanging data (steps '12' and '13'). The LD runs in an IMS-LD player (step '11') and calls (through simple URIs) the tools and the MC (steps '14' and '15'). After learners log in (step '16'), they enact with the IMS-LD player environment and are presented with the tools through which they collaborate (steps '17' and '18'). Thus, the learner experiences the running script through the IMS-LD player (e.g., SLeD [12]) as a unified system supporting tool interoperability. The teacher can use the IMS-LD player to administer the course (steps '19' and '20').

During run-time, the MC is triggered/called by the IMS-LD player which controls tool orchestration according to the UoL design phase (step '14'). The MC fetches or sets data from tools via web service interfaces into its own "data hub/bus". Then, the MC calculates, aggregates and in general transforms (according to the model agreed upon with the teacher at design-time) data from tools into output for the IMS-LD player. The MC "knows" and sets (via web service messages again) IMS-LD properties. Then, the LD behavior changes according to the properties set and the rules triggered in the run engine.

The same MC implementation can be reused when a CSCL script requires similar operations when two tools are coupled (that is, similar affordances, input and output, and web service interfaces). For instance, if a MC connects a chat and a forum tool implementing a specific data-flow model, then the same MC (with little modifications) could be used to connect Twitter to a Wiki tool, if the data-flow model is similar.

To put it briefly, the MC can handle and receive a call from an IMS-LD player tool (e.g. for a group formation service as in [17]). Then, the software component providing the required functionality of group formation is called by the MC. The result is returned to the MC through the web service. The MC finally sets the relevant property in the IMS-LD run engine. Thus, an IMS-LD engine can apply a rule accordingly and present flexible behavior (e.g. formed groups are assigned specific activities according to group member profiles).

3.3 MAPIS3 Example Implementation

In this section we present an implementation example to showcase the MAPIS3-based development process. This implementation was used for Case study 3 to be presented in Section 4.2.3. According to the IMS-LD based script, students work firstly in dyads through a chat tool (we call it "Mentorchat" [30]) to discuss on a topic. Information (i.e. posts, words, teacher rate) about their participation and contribution to chat discussion is propagated to a MC which: 1) forms groups according to a prescribed Jigsaw activity model based on data from chat, 2) visualizes previous chat participation data to learners, 3) assigns moderators (again based on data from chat) with specific role instructions in the next activity where learners form larger

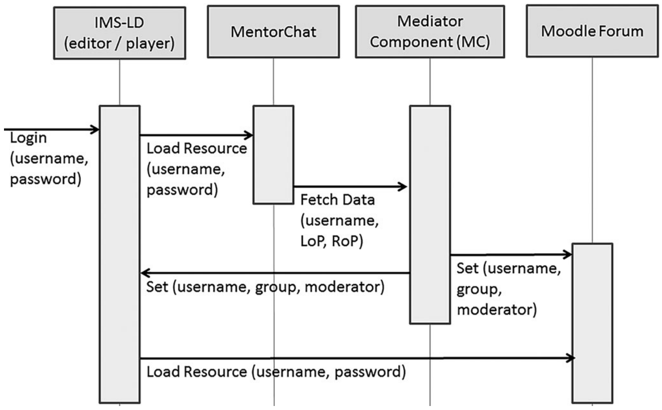

Fig. 2. MAPIS3: Sequence diagram of an example implementation.

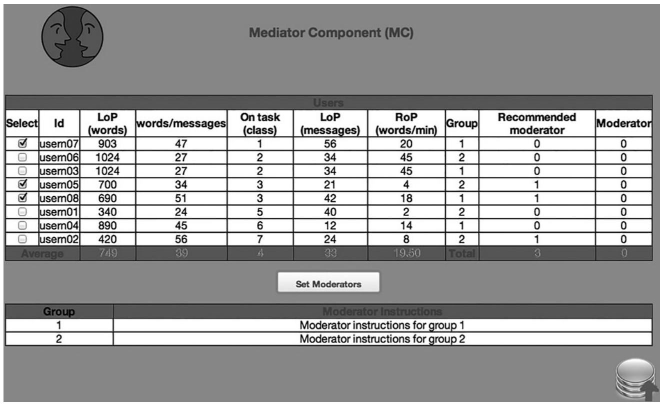

Fig. 3. User interface of the MC in example implementation.

groups (4 or 5 peers) to discuss an open issue and present their group opinion in a Moodle forum. Fig. 2 illustrates the data-flow between these two tools and through the MC which interoperate under the MAPIS3 architecture.

The MC fetches input data (i.e. posts, words, teacher rate) through web service interfaces from Mentorchat. According to a model given by the teacher and programmed by the developer, the MC a) calculates values for groups and moderators and sets them accordingly to an IMS-LD player (see Set (username, group, moderator) web service message) and b) gives information to the Moodle forum tool and control to the IMS-LD player. IMS-LD takes control and decides when to show learners the next forum activity. Finally, Moodle forum is loaded and script activity proceeds with Moodle forum groups and moderators set by the MC. The role of the MC is twofold. On one hand, it fetches data from external tools (the chat) and calculates some indicators useful for pedagogical decisions. On the other hand, the MC sets the necessary IMS-LD properties in order to facilitate flexible execution when IMS-LD conditions are triggered. For instance, a learner who is set as moderator is given different instructions (i.e. is shown a special activity) when collaborating in a Moodle forum activity. Put briefly, the MC sets properties included in IMS-LD conditions triggering activities inside the IMS-LD player environment. Thus, the MC adds flexibility to the IMS-LD based CSCL script.

We developed the MC as an intermediate orchestration layer which performs group formation techniques, and recommends a moderator for each forum of the second activity (Moodle). Thus (Fig. 2), a learner that logs in the IMS-LD player is shown an activity using chat as a discussion tool between groups of two peers. The chat tool outputs data that model in-chat peer interactivity [30]. Level of participation (LoP) is calculated by a model using words, keywords and posts/messages of peers (Fig. 2). Moreover, rate of peer participation (RoP) is calculated by words per minute a peer writes in chat tool. All these calculations are performed in the MC. Data flown from Mentorchat to the MC are transformed into MC output and finally properties set in the IMS-LD player. These settings are driven by pedagogical decisions made by the teacher such as: a) to form groups of mild heterogeneity, b) to assign a moderator to a group discussion. After these properties are set, the IMS-LD conditions are fired and rules are triggered flowing relevant data—user, group, moderator—through the MC to the next tool used in the IMS-LD script which is a Moodle forum. For instance, a learner in this chat activity can be assigned to one of two categories: a) high level of collaborative skills taking into account LoP and RoP chat tool data and b) high knowledge of the learning subject based on the "On Task" parameter (see Fig. 3). If so, then s/he is recommended by the MC to become a moderator in the next activity to be performed in the forum discussion among four peers. MC sets the IMS-LD property that learner is moderator. Then, the IMS-LD player shows forum activity; only the moderator is shown role-related instructions.

Though a typical MC is usually modeling automatic processes, there are cases where the MC can be interfaced to a teacher (or even learner) allowing semi-automatic execution of the running IMS-LD. For instance, in this prototypical implementation, MC functionality (group formation and moderator role assignment) was visualized to teachers. Thus, teachers could manually intervene in the MC's modeling proposal (Fig. 3). Technically, through the settings, the instructor can set values of the IMS-LD properties "group" and "moderator". The "group" property is included in a condition which shows the forum of the specific group for the user; the "moderator" property is included in a condition which shows "moderator instructions" set accordingly by the tutor.

Technically: a) MC rules were coded in PHP with IF-THEN-ELSE structures, b) the IMS-LD based script included aggregate properties to be used in simple "IF-THEN-ELSE-SHOW/HIDE ACTIVITY" rules, c) the data formats used in the example to transfer data among tools were tables in a persistent MYSQL database.

4 Evaluation of MAPIS3 Architecture

4.1. Architecture Evaluation Framework

Scenario-based evaluations are a powerful method for reviewing an architecture design. In a scenario-based evaluation, the focus is on the scenarios which are most important from the business perspective, and which have the greatest impact on the architecture. After careful review of the various methods and techniques proposed in literature (e.g. SAAM, ATAM, CBAM, ALMA, FAAM [31]), in this study, we considered using ATAM because: a) it is one of the most common review methodologies [9], b) it is a mature methodology encompassing also aspects of other methodologies (e.g. SAAM), c) it fits well with our purpose of MAPIS3 testing, as it focuses on scenarios engaging various stakeholders [31]. The output of an ATAM evaluation includes:

TABLE 2
Case Studies Overview

Feature	Case Study 1 (**CS1**), Duration: one week *"Forum visualization & MAPIS3-based LD"*	Case Study 2 (**CS2**), Duration: six weeks *"Forum visualization"*	Case Study 3 (**CS3**), Duration: one week *"Orchestrating synchronous and asynchronous tools"*
Tool linking	(a) Moodle forum → MC → Vis. tool (b) Questionnaire → MC → SLED	Moodle forum → MC → Visualization tool	(a) MentorChat → MC → Moodle forum (b) MentorChat → MC → Visualization tool
Data-transfer	(a) Data on: 1) posts, 2) replies received, 3) ratings earned from collaborators is propagated to visualization tool through MC (b) data on individual domain knowledge is transferred to SLED through MC.	Data on: 1) posts sent, 2) replies received, 3) ratings earned from collaborators is propagated to visualization tool through MC	Data about Mentorchat personal participation (i.e. posts words) is transferred to a Moodle forum through MC.
Flexible interventions	(a) Users/system select type of visualization (b) System provides support to individuals and groups according to atomic and group levels of domain knowledge	(a) Users select type of visualization. MC classifies in IMS-LD each group based on group statistics, and script locks to an information visualization type that could suffice group needs.	(a) Based on participation indicators in MentorChat: a) data groups are formed and b) a moderator is assigned to each group, in Moodle forum.(b) Mentorchat's participation data is visualized and motivates trans-activity [3] in Moodle forum.

1. Elaboration on quality attributes (i.e. requirements) of the software architecture under inspection.
2. Collection of scenarios/case studies that represent the stakeholders' highest usage priority. This collection comes with a utility map that assigns specific scenarios to the requirements set by the architecture being evaluated.
3. Specific analysis results, including the identification of sensitivity points, tradeoffs, and other architectural decisions that affect requirements. Decisions that have a negative impact constitute areas of risk. A sensitivity point is a property of the components that are critical for satisfying a particular requirement. A tradeoff point is a property that affects more than one attribute and is a sensitivity point for more than one attribute.

4.2 Method

We performed three case studies to provide initial evaluation data on the MAPIS3 architecture.

Each case study implemented a specific didactic scenario where the architecture facilitated: (a) tool linking, (b) data-transfer, (c) flexible user interventions. How each case study relates to the above three key issues is depicted in Table 2. Each case study was intended to evaluate the proposed architecture from two different perspectives: a) the user perspective (that is, learners and teachers as system end-users), a perspective relevant to requirements R1 and R2 above, b) the technical point of view (relevant to R3 and R4). Developers are specifically interested in implementing feasible and reusable MAPIS3-based solutions. Learners and teachers focus on a usable, applicable and seamless data-transfer supportive system.

All stakeholders (learners, teachers and developers) were informed about terms like CSCL scripts and LD in a short briefing before each case study initiation. The teachers were initially not knowledgeable of IMS-LD, thus they were introduced to IMS-LD authoring and run-time tools through a presentation of ReCourse and SLeD tools respectively. In all cases, teachers authored a script within 3 days, closely collaborating with a developer. All developers had experience with the ReCourse and SLeD tools and also with online discussion tools (such as Moodle forum). This experience enabled them to compare the burden of development tasks when a CSCL script prescribes activities involving a variety of tools. The developers had advanced programming skills in PHP and MySQL open source tools. They were introduced into the MAPIS3 architecture and asked to apply MAPIS3 guidelines to implement the specific CSCL scenario. They were not knowledgeable of IMS-LD or Web services technologies, hence they followed simple guidelines to handle data-flow among tools, for the given scenario. The following case studies are intended to cover all stakeholders' requirements (especially R1-R2 referring to learners and teachers and R3-R4 to developers). In case study 1, developers had also the role of learner, so that a person could review MAPIS3 from learner and developer viewpoints.

Selected case studies were designed in such a way that they: a) involve flexible tool orchestration (so that evaluation data on system feasibility and reusability could be recorded), and b) apply pedagogically flexible learning designs to test MAPIS3 functionality and usability.

4.2.1 Case Study 1: "Forum Visualization & MAPIS3-Based LD"

13 postgraduate students participated in case study 1 (CS1) (ages 22-30, $M = 26$, $SD = 1.2$). They all had used a forum or chat tool before and generally they were experienced technology users (average computer and information literacy level of 9.2 out of 10 based on 35 questions, similar to [32], and designed according to the B-Tile [33]). The students worked in small groups (3 or 4 peers in each group) and were guided to use a Moodle forum to: a) discuss an open issue and b) present a collaboratively constructed answer. Data regarding peer interaction (posts sent, posts answered and posts rating by co-learners) were propagated through a mediator component to a visualization tool (Google visualization API [34]). Thus, this system architecture facilitated the transfer of collaborative activity data from a Moodle forum tool to a visualization component. Additionally, users could select the preferred type of visualization to see their personal data (Fig. 4).

Moreover, the 13 students were asked to act as developers based on MAPIS3 ideas, and develop a scenario where potential learners would first answer a Web 2.0 tool, namely the Google forms tool questionnaire, to evaluate their domain knowledge. This information is propagated to an

Fig. 4. Data, in tabular and graphical formats, about group collaboration, within Moodle-forum for case studies 1 and 2.

MC that classifies each individual—depending on their questionnaire score and on a specific classification algorithm—as "advanced" or "novice". Next, this classification is transferred to an IMS-LD based tool (the SLeD player [14]) to adaptively provide learners with appropriate learning material (advanced or introductory).

Based on classification information, the SLeD player is capable of implementing flexible interventions to further support student groups. For example, in case a group is formed including several (more than 3) novice members, SLeD activates a support mechanism for this specific group, providing more guidance and support than for other groups. Thus, in this scenario, the MC connects two other components (online questionnaire and the IMS-LD based SLeD) to facilitate flexible and adaptive system behavior in supporting groups of students.

4.2.2 *Case Study 2: "Forum Visualization"*

The second implementation study took place in a secondary education school for adults. The participants were 83 students (ages 18-50, $M = 42$, $SD = 3.7$), with most of them having low familiarization level with online communication tools; only 13 had used forum/chat tools before—but none of them for educational purposes—with an average computers and information literacy level of 3.8 out of 10 (the same 35 questions of CS1 were used).

In this scenario, the students participated in online discussions through a Moodle forum. Discussion was on a project familiar to them. The event was hosted by a Moodle forum, just as in case study 1, and lasted six weeks. Peers were organized in groups of 6 or 7 with the target of presenting a collaboratively constructed answer on their common subject of discussion organized according to the web-quest inquiry-oriented lesson format.

As in case study 1, data regarding peer interaction (posts sent, posts answered and posts rating by co-learners) were processed by a MC; processing was based on a comparison model between individual and group, and a standard deviation from the average (i.e. good) participation levels among all groups. Information on collaboration within a Moodle forum tool was transferred to both: a) an IMS-LD based tool (which was the SLeD player again), and b) a visualization tool [34]. This system architecture facilitated the transfer of collaborative activity data from a Moodle forum tool to a visualization tool. Then, the IMS-LD player tool acted as the orchestrator of the learning design; that is, SLeD revealed various informative messages, according to the users' preferred type of collaborative information visualization (Fig. 4). For four weeks, the users chose visualization type. Then, for two weeks, the system randomly restricted visualization and messages to one of three types. The system (based also on the teacher's choice) changed behavior and adapted (i.e. faded out) visualization and guidance.

These types of information visualization are described as mirroring, meta-cognitive and guiding [35]. The colors (actual color-range is red-orange-green in Fig. 4) denote whether a learner in the collaboration process inside the Moodle forum is doing well or not, according to the parameter (posts/replies/rating) observed and the statistical data of this parameter calculated on all the learners of each group. So, for example, (see Fig. 4) user05 in the fifth row and third column of the table depicted is in the orange area with three replies in his/her Moodle forum group.

In CS1 and CS2 these messages contain information on both the individual learner(s) and the groups of learners. The system provides real-time information on parameters exemplifying the Moodle forum discussion process (see Fig. 4). The format (how) and the exact time (when) of the presentation of the messages to learners, consolidated a process handled by pedagogic decisions. These were expressed in IMS-LD rules and triggered after IMS-LD properties were set by the MC. For example, the system refreshed the user's preferred type of collaboration visualization only when new actions affecting equilibrium among peers, within group and among groups occurred. Thus, pedagogic decisions expressed in IMS-LD rules (conditions in Level B terms), orchestrated the whole learning design experience of enacting it. Moreover, messages did not appear inside the forum tool. Instead, they were on purpose presented within the IMS-LD run-time environment. The latter fact caters for pedagogic practices simulating good practices in classes where teacher support is/should be provided in a discrete manner. Therefore, the system allowed for supporting a student in private, or even indirectly, e.g. through another peer.

Overall, in case study 2 individual and group data are propagated to the MC. The MC organizes data, and transforms them into information according to group models of good collaboration based on the parameters observed. Then, a visualization tool is fed by the MC with information concerning peer collaborative behavior. The user's preferred visualization type is set into the running SLeD. Then, SLeD has all the information upon which to apply the predesigned rules and trigger adaptive visualization towards peers and groups. That is, visual support was flexibly triggered by the IMS-LD player tool and was illustrated in various formats to learners according to their or the teacher's or system's choice. These peer interactivity representational formats were graphical, lexical or diagrammatical (see Fig. 4 for a graphical example).

The MC code was different for each pair of tools to be mediated. Thus, in both CS1 and CS2 the same MC code is used to a) handle data-flow between Moodle forum and Google Visualization tools, and b) enable the same sequence of activities (i.e., show first the Moodle forum and then the Visualization tool). Differences between CS1 and CS2 are: a) the discussion subject and b) the learners' sample. In CS1 learners could choose the visualization presented to them through the whole case, while CS2 learners could not choose the visualization type after a period of discussion (this was locked by the system at a specific random level and time). This modification in the learning design of CS2 was implemented to stress-test R3-feasibility and R4-reusability requirements of MAPIS3.

4.2.3 Case Study 3: "Orchestrating Chat-Forum Tools"

The third experiment was implemented according to a script already presented in Section 3.3 and was realized totally on-line (teachers and learners never met each other in classroom) by undergraduate students of a university school of informatics. In case study 3 (CS3) the 39 participating students had all used forum or chat tools before but only 13 of them for educational purposes, with an average computers and information skill level of 7.9 out of 10 (the same 35 questions of CS1 were used).

In all case studies and similarly in CS3, IMS-LD rules were designed by teachers in ReCourse as IMS-LD Level B conditions. For instance, a teacher could state that "if a learner is characterized as Moderator in MentorChat, then s/he is shown an extra set of role-related instructions for the next Moodle forum activity".

Case study 3 focused on orchestration of chat and forum tools and data exchange among them (it is described as an example implementation case in Section 3.3).

4.2.4 Data Collection and Analysis

Data were collected and analyzed as follows:

a) *Questionnaire*. The participants in all case studies were asked to fill in an appropriate questionnaire (depending on their role as developer, teacher and/or learner). These questionnaires included both closed-type questions (to be answered on a Likert scale from 1-Totally Disagree to 5-Totally Agree), and open-ended ones where participants freely expressed their opinions. The questionnaires were the same for the same role across all case studies. Closed-type questions were analyzed by calculation of their average and standard deviation statistics, while open-type ones were content analyzed and classified. Table 2 presents a summary of closed-type questions along with relevant data and statistics (average and standard deviation). Each question belongs to a question set which is directly linked to a requirement (R) of Section 3.1. For example, Q2.1 belongs to Q2 set and is related to R2 requirement. Requirements analysis of ATAM directed this top-down approach of forming our questionnaire and guided the method to collect our data. Their analysis followed the principles of a mixed evaluation method ([36], [37]). That is, we reviewed quantitative data and checked whether answers to closed-type questions matched logically with answers to open-type questions. For example, if a learner answered that he had usability problems, we reviewed that his/her answer in the relevant Q2.1 closed-type questions is not Totally Agree (5). Notice that open questions followed each set of questions Q1-Q4. During interviews, the stakeholders were to argue on their articulated opinion. Valuable answers concerning MAPIS3 were analyzed by content and were classified so as to allow valuable conclusions for the systems built upon the MAPIS3 architecture. Major conclusions are mentioned in the section presenting results of our evaluation method.

b) *Interviews.* Having analyzed the questionnaires and extracted the major conclusions, we performed interviews with each stakeholder individually. Each interview lasted on average 15 minutes and focused on interesting observations of the stakeholders and the major conclusions already extracted from open-type questions. Notice that interviews had added-value as some stakeholders had both learner and developer roles in the same case study (see CS1). The classification of conclusions concerning both interviews and open-type question answers are a by-product of the analysis of the interviews.

c) *Log files of the activity*. Learners' activity within each tool was logged (i.e. post sent, posts read, time posts are read, access time of resources like forum or visualization tool, time and duration visualizations were seen, individual pre- and post-tests, grades of group deliverables etc.) and analyzed. Portions of these data were transferred to and from the IMS-LD run engine through the MC transformations. These data also helped in verifying users' answers in questionnaires concerning system seamless data-transfer (i.e. usability, see Q2.5 in Table 2). That is, log files were used mainly to verify system usage in all case studies. The analysis of the learners' system usage followed the principles of a mixed evaluation method. This approach allows us to better interpret results from the analysis of available data sources, as it triangulates data so that the evidence shown by one method/data source can be confirmed or refuted by the other [36].

Data collection is based on: a) 135 learners involved in case studies: CS1 ($n1 = 13$), CS2 ($n2 = 83$) and CS3 ($n3 = 39$), b) nine teachers involved in case studies: CS1 ($n1 = 2$), CS2 ($n2 = 3$) and CS3 ($n3 = 4$) and c) 17 developers of MAPIS3 in all case studies (i.e. 13 for CS1, 1 for CS2, 3 for CS3). The results from corresponding answers to questionnaires are presented in Table 2 across all questions and column with statistical measures. Results from interviews and log files are also analyzed in relevant sub-sections below.

4.3 Results

Results from all case studies are presented and organized according to the source from which they emerge:

a) *Questionnaires.* Results from questionnaires (across all case studies) are presented in Table 2. Average and standard deviation is presented. Moreover, a by-product of the triangulation of the data analysis process was that we detected relationships among questions even from different question sets. These relationships were identified through interviews and were verified by log data.

b) *Interviews*. Major conclusions emerging from participants' interviews include:

Learner role:

- I1: Most learners (78 percent) agree that the showcased learning designs exhibit a reasonable difficulty in implementation. The CSCL scenarios involved many tools and data-transfer among them. This fact was a unique experience for them. They understand that data concerning themselves flow between tools while the details of this data-flow are being hidden from them.
- I2: Many learners (66 percent) believe that tool linking, implemented by the MAPIS3 architecture, is challenging regarding data-transfer requirements.
- I3: Most learners (74 percent) believe that the implemented collaboration scripts demonstrate an original and interesting way of tool linking. Many stated, "*... we could not have conceived that linking tools in such a way was possible*".
- I4: MAPIS3 systems are helpful for both individual and group-work self-regulation because of real-time collaborative reflections (77 percent of learners). Learners at 74 percent in CS1 stated, "*.... I feel supported when the system recognizes advanced or novice learners and assists them accordingly*", while others in CS1 and CS2 (69 percent) said, "*... I enjoyed being able to select the type of data visualization*". Also, many learners (71 percent) stated, "*When the system lets me choose the flow of activities, then I feel supported*".

Teacher role:

- I5: From the teacher viewpoint the system is easy to deploy and control, provided that a developer supports the implementation of the "glue" (i.e. mediator component) between tools (78 percent of teachers).
- I6: 38 percent of teachers believe that "*the IMS-LD-based script is innovative but it seems difficult to be implemented in the context of K-12 Education*". When discussing the point in depth, the teachers argued that "*the script is very motivating, but I (as a teacher) may encounter difficulties while creating the material and script with the current IMS-LD tooling*" but also that "*... MAPIS3 with its flexible features may allow for such implementations and scripts like Jigsaw*".
- I7: Regarding the complexity of the authoring phase and the use of questionnaire tools, specialized tools, like a GoogleForm, are preferred over the IMS-LD vocabulary and related standards (e.g. IMS-QTI) (89 percent of teachers).
- I8: A well-designed interface makes it easy for the learners to interpret the data transferred among tools when using data visualization (64 percent of learners).
- I9: Many teachers (63 percent) agreed that interconnecting systems may enable the setting up of useful educational services, such as viewing students' progress using graphical representations.
- I10: Teachers (86 percent) agreed that scenarios support a flexible course of action, while retaining the main course flow clear. Many stated, "*...now I see the possibility of developing scripts where I can mix and match data and tools*".

Developer role:

- I11: Most developers (74 percent) agreed that the scripts implemented through the MAPIS3 architecture can be reused in other course instances with little deployment effort. While interviewed, 83 percent of the developers stated that "*... it is easy to transfer implemented scripts and apply them to other authentic situations and scenarios*". Also, 78 percent of the developers stated that "*... MAPIS3-based systems are easily expandable with other tools in other orchestration scenarios*".
- I12: Most developers (69 percent) pointed out that an expected difficulty is the seamless login procedure among various systems, often appearing when flexible orchestration of various tools is required. When discussing the point in more depth, 89 percent of developers emphasized that the most time-consuming task a developer has to deal with, is that the users should be transparently logged in all the systems interlinked (i.e. IMS-LD player, Moodle forum etc.). The most difficult part/process (identified as risk according to ATAM) in the whole architecture is when a proprietary system-tool (e.g. Twitter) is involved in the orchestration of the learning flow and seamless login from other tools needs to be implemented.
- I13: Developers pointed out (76 percent) that the creation of a MAPIS3-based LD reduced the time cost of administrative tasks (for example, in group formation procedures). For instance, a substantial administrative task is the synchronization of user accounts in all different tools used in a unified IMS-LD based script. This task is not trivial when user grouping/regrouping is concerned.
- I14: Most developers (89 percent) agreed that when proprietary tools are employed, then applying the architecture requires increased effort as opposed to using open-source tools. Moreover, 59 percent of the developers emphasized that the functional limitations of orchestrated tools may limit the flexibility of reusing the scripts in other situations.
- I15: Most (87 percent) developers agreed that MAPIS3 satisfies all requirements R1, R2, R3, R4 due to specific strategic choices. When asked to elaborate on their position, they pointed out that "*... MAPIS3 is highly decoupled*", "*... with MAPIS3 IMS-LD standard and players are left intact*" and "*... MAPIS3 is based on standards like IMS-LD and service oriented architectures-SOA*".

I16: Developers (87 percent) stated that the major identified risk in MAPIS3 implementation is the poor support and development state of IMS-LD tools (especially players).

c) *Log files (major conclusions from activity log files):*

L1: There were no login problems to the system, nor to any of the tools utilized by IMS-LD based scripts.

L2: Learners did not need to spend much time in reading help instructions (average: 5.7 minutes), nor did they access help information frequently (average: 1.3 accesses).

L3: Learners spent a considerable amount of time watching the visualized feedback information about their collaboration (26 percent of their activity time).

L4: Data-transfer among tools and visualization of collaboration led to a more balanced group collaboration (participation and contribution) in discussions (75 percent of groups utilizing visualization support showcased a balanced collaboration among peers which affected group deliverable organization, time management and quality).

L5: Data-transfer among tools and visualization of group collaboration and individual statistics (e.g. posts, replies, ratings) led to specific usage patterns of the MAPIS3-based system by learners. For instance, 89 percent of actions in discussion tools was initiated after learners had seen for some time (on average 3 minutes) visualized information about their individual and group collaborative metrics (e.g. posts sent, ratings earned, replies accepted etc.).

TABLE 3
Evaluation Questionnaire (Part)

ID	Question	Data collection & Statistics
Q1	Range of Applicability (R1)	**Role**: **Teachers** N = 9 & **Learners** N = 135
Q1.1	The showcased script was clear	M = 4.56 SD. = 0.74
Q1.2	The showcased script was realistic	M = 4.06 SD = 0.88
Q1.3	The showcased script is useful in real situations	M = 3.73 SD = 1.06
Q1.4	The showcased script is rich (i.e. with many activities and tools)	M = 4.53 SD = 0.73
Q2	Seamless Data-transfer (R2)	**Role**: **Learners** N = 135
Q2.1	It was easy to use the system efficiently and understand what to do according to scenario	M = 4.39 SD = 0.93
Q2.2	Tools were easily identifiable during script activity evolution	M = 4.53 SD = 0.60
Q2.3	The system gave me the impression of a unified system/scenario	M = 4.26 SD = 0.83
Q2.4	There was data exchange among tools during script evolution	M = 4.52 SD = 0.60
Q2.5	Login to tools involved in script was seamless	M = 4.60 SD = 0.59
Q3	Feasibility (R3)	**Role**: **Developers** N = 17
Q3.1	Benefit/difficulty ratio of data-transfer is high	M = 3.76 SD = 0.44
Q3.2	Benefit/difficulty ratio of learning design flexibility is high	M = 3.71 SD = 0.47
Q3.3	Benefit/difficulty ratio of tool orchestration is high	M = 3.88 SD = 0.33
Q4	Reusability (R4)	**Role**: **Developers** N = 17
Q4.1	Case study is feasible to be deployed in my working scenario	M = 3.88 SD = 0.33
Q4.2	Case study system is usable in real situations	M = 3.82 SD = 0.39
Q4.3	Case study system is easy to change and expand	M = 3.59 SD = 0.51

4.4 Discussion

In the following, we discuss and triangulate results organized in two subsections: a) Applicability and Seamless data-transfer (functional requirements), b) Feasibility and Reusability (relevant to the technical aspects of MAPIS3 as viewed by developers).

4.4.1 *Applicability and Data-Transfer*

The MAPIS3 architecture produces systems and IMS-LD based CSCL scripts (at least the ones provided in the cases studied) that are easy to use and understand (see Table 2, items Q1.1, Q2.2, Q2.5, interview item I1).

Learners understood the script flexible interventions (see Table 2, items Q1.1, Q1.4, Q2.2 and Q2.5). This is proven by the fact that all participants followed successfully the script activities (log file items L2, L3) and expressed subsequently a positive opinion of the experience (interview items I1, I2 and I5). Moreover, the learners used the system (both architectural and interface mechanisms—see conclusion section) with no problems (Q2.5, L1 for architectural transparency) for their collaboration, contribution strategies and learning outcomes (L4, L5).

Although answers suggest that the IMS-LDs studied were realistic and could be useful in real situations and in situations and working environments which the participants encounter (see Q1.2 and I3, I4, I9, I10), we analyzed more the negative answers of the Q1.3 question. Both learners and teachers feel that the applicability of such a type of learning material in a real situation is a matter of research. Despite the positive view of the case studies' systems, they expressed hesitation in adopting and applying these systems in their working environments (I6). Data shown in Table 3 reinforces this view: Learners considered the instance scripts to be fairly useful for a real scenario (Q1.2), but not so feasible as to be effectively deployed in real situations (Q1.3). However, teachers believe that the option of MAPIS3 to utilize a range of tools (especially Web 2.0) with various affordances is a key factor to overcome limitations of IMS-LD modeling capabilities of flexible CSCL scripts (see I7).

The previous observations are related to the results of other studies in the field of IMS-LD: the most significant obstacle for the adoption of the specification resides in the complexity of course authoring, rather than other problems like the understanding of the specification or the required training process of instructors [38].

Results in Q2.1 and Q2.3 illustrate that the participants had the experience of a seamlessly integrated system that homogenized course flow and tools involved in learning activities. That is, they did not perceive tools utilized by CSCL activities as something peripheral/external to the running IMS-LD and player. In other words, we have evidence that scripts implemented following MAPIS3 can realize usable systems, hiding complexities from user-learner (black-box learner view).

Q2.4 revealed that most learners with no experience using IMS-LD players like SLeD, had a positive opinion about the use of external to IMS-LD player tools in the course flow. They pointed out the support they received from the system in their collaboration (I5).

Moreover, most teachers agreed that MAPIS3 can potentially produce glass-box CSCL systems and learning designs with interfaces that keep learning flow transparent for the teacher (I8). This positive perception of MAPIS3 conditions management capabilities stand in contrast with other experiences on the use of IMS-LD (e.g. [39]) where the focus was on IMS-LD conditions authoring, a task which was considered error prone and difficult to understand. We acknowledge that many deductions of the current section may be linked to the way the MAPIS3 architecture was implemented and case studied. This fact is also mentioned in the conclusion section.

4.4.2 Feasibility and Reusability

The feasibility requirement of MAPIS3 is satisfied, that is, it is feasible to deploy flexible scenarios with MAPIS3 within a reasonable development time interval (Q3.1, Q3.2, Q3.3 and I13). Case studies reveal that IMS-LDs implemented according to MAPIS3 can be reused in other course instances with little extra deployment effort and fewer administrative tasks (Q4.1, Q4.2, Q4.3, and I11).

A discussion-worthy observation is that each developer (especially in CS1) could implement the learning design scenario successfully and in a relatively small development time (dt) (dt: M = 20 man-hours, SD = 3.2 man-hours). These values are small when compared to the dt required to perform a comparable development task without using MAPIS3. No use of MAPIS3 means: a) no use of IMS-LD for the CSCL script representation and especially Level B rules for the flexibility part of the script, and, b) no use of web services for input/output data handling to facilitate data-flow. Developers roughly estimated that: a) to support the flexibility level required by a CSCL script without IMS-LD rules and MC could lead to 80 man-hours of development effort, and, b) data-flow support without web service technologies could lead to proprietary solutions requiring on average 120 man-hours of development effort. However, the most time consuming task that a developer had to deal with, was that the users should be transparently logged (I12) in the systems involved (e.g. IMS-LD player, forum etc.).

The most difficult part/process in the whole architecture (identified as risk according to ATAM) is when a proprietary tool (e.g. Twitter) is used in the script learning activity. Nevertheless, as these tools become reusable themselves, they provide specific application interfaces which can alleviate common problems such as seamless login issues. Such issues have been mentioned and dealt with by many researchers and systems (e.g. [40], [26]).

4.4.3 MAPIS3 Qualities and Implications for Practitioners

By implementing ATAM we created the following catalog of Risks, No-risks, Sensitivity and Trade-off points:

(1) *Risks*. a) Functionality constraints (R1) can impose extra burden to reusability (R4), and b) reusability can be an issue in certain circumstances when feasibility is concerned (see I14).

(2) *Sensitivity points*. a) External tools interfaces affecting interoperability (R3) and changeability (R4), b) IMS-LD rules affecting reusability (R4) and adaptability (R3).

(3) *Tradeoff Points*. a) External -to IMS-LD player- tool interfaces affecting interoperability and changeability. MAPIS3 provides a possible solution to this issue, because we can interchange the usage of a tool in a learning activity with another tool with similar affordances. This solution presupposes a repository already foreseen in the MAPIS3 architecture as a future enhancement (not shown here). b) IMS-LD rules & properties affecting adaptability, reusability. MAPIS3 provides a solution to this issue when complex rules and sets of properties are needed; if one cannot scale well with IMS-LD properties and rules, then the developer can use the MC and distribute complexity between IMS-LD and the MC.

MAPIS3 satisfies all requirements R1, R2, R3, R4 due to specific facts: a) the IMS-LD standard and players are left intact, b) the MAPIS3 architecture is highly decoupled, c) MAPIS3 is based on standards (see I15 where most developers agree). The major identified risk in the MAPIS3 application is the poor support and evolution of IMS-LD players and supporting administrative tools (see I16). Although there are plenty of tools for the IMS-LD design, run-time systems and players seem to be neglected at the level of adaptive tool orchestration support as they have not evolved towards production systems to be used in actual delivery of courses. This poses a risk to the MAPIS3 architecture and could be a problem in case of a system with heavy production requirements (e.g. real-time latency, simultaneous connections etc.). Efforts like IMS-LTI [25] are ameliorating the risk identified. Our view is that an orchestration layer that is dependent upon IMS-LD tooling available at design and run-time can be substituted by tooling that supports the tasks of script authoring and running. Thus, widely used learning environments like LAMS [5], Moodle or WebCollage [41] could be enhanced. A rule-based engine inside these environments (like .LRN enhanced system [44]) could play the role of the orchestrator for the various tools and MCs.

In comparison with IMS-LTI [25], which caters only for the connection of tools with IMS-LD, MAPIS3 deals also with an IMS-LD controlled/orchestrated data-transfer among these tools. In contrast to all other similar contributions (e.g. [7], [13], [26], [27] mentioned in the background section), the MAPIS3 novelty lies in the following facts: 1) MAPIS3 is based solely on current standards i.e. IMS-LD and Web services for the interconnectivity part, (on the contrary [7] and [26] employ non-standards-based service adapters; also, Glue!PS in [27] does not produce IMS-LD player compatible scripts), 2) IMS-LD is left intact as we do not introduce any new XML element in IMS-LD to perform any adaptation (unlike [7], [26] and [27]) or even proprietary documents to model learning flow (unlike [13]), 3) MAPIS3 achieves seamless data-transfer between tools, unlike Glue! [26] and Glue!PS [27] which focus on calling tools and instantiating working groups and peers within learning activities. All the aforementioned approaches fail to meet the basic requirement that MAPIS3 deals with, which is data-flow among tools orchestrated/controlled by a CSCL scenario modeled in pedagogical rules that in our approach are IMS-LD based.

We argue that MAPIS3 architecture surpasses a customized approach to CSCL application development because it supports: a) component oriented development (Fig. 1) as opposed to designing a system from scratch, b) reusability of software components, c) reusability and portability of an existing learning design, because a MAPIS3 and IMS-LD based LD can be utilized in various applications, d) rapid development of prototype systems.

Limitations of the current study include: a) the small number of stakeholders with teacher's role, b) the relatively narrow background of the learners (see CS2), and c) the data analysis method, which provided a percentage tendency which is by no means definitive and deserves further investigation and d) the relative low variety of interconnected tools in the case studies.

Regarding point (d) above, we mention that we have already implemented many more MAPIS3-based solutions in educational and business environments, not yet studied in detail. Scripts implemented include some well-known scenarios like Jigsaw, Pyramid, Think-Pair-Share ([6]) and combinations of them. Tools like Twitter, Openmeetings [42], Moodle Questionnaires, GoogleForms are example tools/systems that have been successfully integrated in a specific IMS-LD based script in less than a day's work for a developer who is knowledgeable of: a) MAPIS3, b) IMS-LD, c) web services, and d) a programming language like PHP or Java.

It is worth noticing that the way MAPIS3 was implemented in each case study may have influenced system usability. That is, without a developer a teacher could not achieve the task of 'tool orchestration'. In our case studies, the developer(s) needed on average three days of work to design an IMS-LD with two online communication tools. In general, we expect this time span (three days) to be a typical developer's workload in any similar scenario orchestrating no more than three tools.

From the practitioners' (developers and teachers') viewpoint MAPIS3 also prescribes the use of multiple MCs depending on the script requirements. MAPIS3 can support cases for even more complex situations, where a script flow demands (for example): a) data from many tools (thus requiring many simple MCs (as in our case studies) or fewer complex MCs with an even more complex logic and b) calculated outputs based on several inputs from these tools. In such cases, a complex MC can be thought of as consisting of many simple MCs; this case resembles Web services orchestration [28].

5 Conclusions and Future Research

We have explored the effectiveness of MAPIS3 architecture in three different case studies and provided evidence that the architecture can support IMS-LD based CSCL scripts, with flexible tool orchestration aided by "mediator components". Unlike other relevant solutions, our approach: a) does not require any proprietary technologies for the design and implementation parts, but only well- established standards such as IMS-LD and web services; b) all tools involved in a CSCL scenario execution (for example, tools exchanging data and players like SLeD) are left intact. That is, we fully re-use available technological tools building on top of them flexible MC capabilities. By contrast, while GSI [7] achieves data-flow among tools, it requires IMS-LD authoring and execution tools to introduce new functionalities, thus hampering backward compatibility and re-usability. MAPIS3 can be applied to a diversity of CSCL scripts and deployed over a variety of tools; to support this, we have tested MAPIS3 under the following deployment architectural settings: a) distributed systems/tools over the same domains (see CS3), b) distributed systems with Web 2.0 tools over different domains (e.g. Moodle, GoogleForms).

In the near future, we plan to: a) organize a teacher workshop in order to further evaluate the MAPIS3 architecture from a teacher's point of view; b) systematically investigate the orchestration of diverse services and interfaces (e.g. widgets) based on the proposed MAPIS3 architecture; c) explore the efficiency of developing an integrated MCs toolbox library to support the easy and technically transparent (for non-expert users) tool orchestration in various pedagogical scenarios. We have already started working towards the last aim in an attempt to provide flexibility in IMS-LDs (see [41]). Thus, we aim to support teachers to undertake the role of a course author and gradually need the developer less.

The 'tool-orchestration-involving-data-flow' problem is technically demanding. However, we believe that we have provided a well-defined solution for a team of practitioners (teachers in collaboration with a capable developer) to implement flexible LDs. How to further automatize our solution for the end-user is a matter of future research. Thus, a possible future MAPIS3 development could be an IMS-LD editor/player using a semantic-based repository to dynamically search, discover and select tools suitable for the flexible script interventions to be realized. An example could be enhancing OntoolSearch [43] to support tool orchestration and learning data-flow semantics. Additionally, a mechanism for automatic (or at least semi-automatic) MC composition is currently being investigated to be deployed over MAPIS3 architecture.

References

[1] P. Dillenbourg, "What do you mean by collaborative learning?" in *Collaborative-Learning: Cognitive and Computational Approaches*, P. Dillenbourg, Ed., 1999, pp. 1–19.
[2] P. Dillenbourg and F. Fischer, "Basics of computer-supported collaborative learning", *Zeitschrift für Berufs- und Wirtschaftspädagogi*, vol. 21, pp. 111–130, 2007.
[3] L. Prieto, M. Abdulwahed, M. H. Dlab, W. Balid, and I. Gutiérrez, "Orchestrating technology-enhanced learning: a literature review and a conceptual framework," *Int. J. Technol. Enhanced Learning*, vol. 3, pp. 583–598, 2011.
[4] P. Tchounikine, "Operationalizing macro-scripts in CSCL technological settings," *Int. J. Comput.-Supported Collaborative Learning*, vol. 3, no. 2, pp. 193–233, 2008.
[5] (2015, Feb. 2). LAMS. [Online]. Available: http://lamsfoundation.org/
[6] D. Hernandez-Leo, J. I. Asensio, and Y. Dimitriadis, "Computational representation of collaborative learning flow patterns using IMS learning design," *J. Educ. Technol. Soc.*, vol. 8, pp. 75–89, 2005.

[7] L. de la Fuente Valentin, A. Pardo, and C. K. Delgado, "Generic service integration in adaptive learning experiences using IMS learning design," *Comput. Educ.*, vol. 57, pp. 1160–1170, 2011.

[8] IMS-LD. (2003). IMS Global Learning Consortium: Learning Design Specification. [Online]. Available: http://www.imsglobal.org/ specifications.html

[9] ATAM. (2013, Jan. 17). Architecture Tradeoff Analysis Method. [Online]. Available: http://www.sei.cmu.edu/architecture/tools/ evaluate/atam.cfm

[10] (2015, Dec. 13). IDLE (Integrated Learning Design Environment). [Online]. Available: http://ilde.upf.edu/

[11] (2015, Feb. 2). LD Tools. [Online]. Available: http:// jiscdesignstudio.pbworks.com/w/page /26408663/Learning Design tools

[12] I. Magnisalis, S. Demetriadis, and A. Karakostas, "Adaptive and intelligent systems for collaborative learning support: A review of the field," *IEEE Trans. Learning Technol.*, vol. 4, no. 1, pp. 5–20, Jan.–Mar. 2011.

[13] L. Palomino-Ramírez, et al. "LeadFlow4LD: A method for the computational representation of the learning flow and data flow in collaborative learning," *J. Universal Comput. Sci.*, vol. 19, no. 6, pp. 805–830, 2013.

[14] SLeD. (2005). Service Based Learning Design Player. [Online]. Available: http://sled.open.ac.uk/

[15] D. Burgos, C. Tattersall, and E. J. R. Koper, "Representing adaptive and adaptable units of learning. How to model personalized eLearning in IMS learning design," in *Computers and Education: E-Learning—From Theory to Practice*. New York, NY, USA: Springer, 2006.

[16] E. J. R. Koper, Modelling units of study from a pedagogical perspective: The pedagogical meta model behind EML, 2001.

[17] I. Magnisalis and S. Demetriadis, "Orchestrating adaptive and complex CSCL scenarios through a choreography among IMS-LD and external services," in *Proc. 4th Int. Conf. Intell. Netw. Collaborative Syst.*, 2012, pp. 89–96.

[18] A. Paramythis and A. Cristea, "Towards adaptation languages for adaptive collaborative learning support," presented at the 1st Int. Workshop Individual Group Adaptation Collaborative Learning Environ. held in Conjunction With 3rd Eur. Conf. Technol. Enhanced Learning, Maastricht, The Netherlands, 2008.

[19] C. Ferraris, C. Martel, and L. Vignollet, "Helping teachers in designing CSCL scenarios: A methodology based on the LDL language," *Proc. 8th Int. Conf. Comput. Supported Collaborative Learning*, 2007, pp. 193–195.

[20] M. L. Bote-Lorenzo, E. Gomez-Sanchez, G. Vega-Gorgojo, Y. A. Dimitriadis, J. I. Asensio-Perez, and I. M. Jorrin-Abellan, "Gridcole: A tailorable grid service based system that supports scripted collaborative learning," *Comput. Educ.*, vol. 51, no. 1, pp. 155–172, 2008.

[21] F. Ghali, A. I. Cristea, and M. Hendrix, "Augmenting e-learning standards with adaptation," in *Proc. 11th Conf. Comput. Adv. Technol. Educ.*, 2008, pp. 79.

[22] G. Durand, L. Belliveau, and B. Craig, "Simple learning design 2.0," in *Proc. IEEE 10th Int. Conf. Adv. Learning Technol.*, 2010, pp. 549–551.

[23] A. Harrer, N. Malzahn, and A. Wichmann, "The remote control approach—An architecture for adaptive scripting across collaborative learning environments," *J. Universal Comput. Sci.*, vol. 14, no. 1, pp. 148–173, 2008.

[24] P. Sharples, D. Griffiths and W. Scott, "Using widgets to provide portable services for IMS learning design," in *Proc. 5th Int. TENCompetence Open Workshop*, Sofia, Bulgaria, 2008, pp. 57–60.

[25] (2013, Jan. 17). Learning Tools Interoperability (LTI). [Online]. Available: http://www.imsglobal.org/lti

[26] C. Alario-Hoyos, M. L. Bote-Lorenzo, E. Gómez-Sánchez, J. I. Asensio-Pérez, G. Vega-Gorgojo, and A. Ruiz-Calleja, "GLUE!: An architecture for the integration of external tools in virtual learning environments," *Comput. Educ.*, vol. 60, no. 1, pp. 122–137, 2013.

[27] L. P. Prieto, J. I. Asensio-Pérez, J. A. Muñoz-Cristóbal, I. M. Jorrín-Abellán, Y. Dimitriadis, and E. Gómez-Sánchez, "Supporting orchestration of CSCL scenarios in web-based distributed learning environments," *Comput. Educ.*, vol. 73, pp. 9–25, 2014.

[28] (2015, Feb. 2). Web Services Architecture, W3C. [Online]. Available: http://www.w3.org/TR/ws-arch/

[29] C. Peltz, "Web services orchestration and choreography," *IEEE Comput.*, vol. 36, no. 10, pp. 46–52, Oct. 2003.

[30] S. Tegos, S. Demetriadis, and A. Karakostas, "MentorChat: Introducing a configurable conversational agent as a tool for adaptive online collaboration support," in *Proc. 15th Panhellenic Conf. Informat.*, 2011, pp. 13–17.

[31] M. T. Ionita, D. K. Hammer, and H. Obbink, "Scenario-based software architecture evaluation methods: An overview," in *Proc. Workshop Methods Tech. Softw. Archit. Rev. Assessment, Int. Conf. Softw. Eng.*, 2002.

[32] (2015, Feb. 2). Computer Literacy Survey. [Online]. Available: http:// www.fscj.edu/tutorials/media/online_skill_assess.pdf

[33] P. M. Beile, "Development and validation of the Beile test of information literacy for education (B-TILED)," Ph.D. dissertation, College Educ., Central Florida Univ., Orlando, FL, USA, 2005.

[34] (2015, Feb. 2). Google Visualization API. [Online]. Available: https:// developers.google.com /chart/ interactive/docs/reference

[35] A. Soller, A. Martínez, P. Jermann, and M. Muehlenbrock, "From mirroring to guiding: A review of state of the art technology for supporting collaborative learning," *Int. J. Artif. Intell. Educ.*, vol. 15, no. 4, pp. 261–290, 2005.

[36] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 2nd ed. Newbury Park, CA, USA: Sage, 2002.

[37] A. Martínez, Y. Dimitriadis, B. Rubia, E. Gómez, an P. De La Fuente, "Combining qualitative evaluation and social network analysis for the study of classroom social interactions," *Comput. Educ.*, vol. 41, no. 4, pp. 353–368, 2003.

[38] S. Neumann, M. Klebl, D. Griffiths, D. Hernández-Leo, L. de la Fuente Valentín, H. Hummel, F. Brouns, M. Derntl, and P. Oberhuemer, "Report of the results of an IMS learning design expert workshop," *Int. J. Emerging Technol. Learning*, vol. 5, pp. 58–72, 2010.

[39] M. Derntl, S. Neumann, D. Griffiths, and P. Oberhuemer, "Investigating teachers' understanding of IMS learning design: Yes they can!," in *Proc. 5th Eur. Conf. Technol. Enhanced Learning Conf. Sustaining TEL: Innovation Learning Practice*, 2010, pp. 62–77.

[40] L. De la Fuente Valentin, D. Leony, A. Pardo, and C. D. Kloos, "User identity issues in mashups for learning experiences using IMS learning design," *Int. J. Technol. Enhanced Learning*, vol. 3, no. 1, pp. 80–92, 2011.

[41] I. Magnisalis, S. Demetriadis, and Y. Dimitriadis, "Flexible tools for online collaborative learning: Integration of adaptation patterns functionality in the WebCollage tool," in *Proc. 17th Panhellenic Conf. Informat.*, 2013, pp. 114–121.

[42] (2015, Feb. 2). Openmmetings. [Online]. Available: http://code.google.com /p/openmeetings/ adapt

[43] G. Vega-Gorgojo, M. L. Bote-Lorenzo, J. I. Asensio-Pérez, E. Gómez-Sánchez, Y. A. Dimitriadis, and I. M. Jorrín-Abellán, "Semantic search of tools for collaborative learning with the Ontoolsearch system," *Comput. Educ.*, vol. 54, no. 4, pp. 835–848, 2010.

[44] L. de la Fuente Valentín, A. Pardo, and C. Delgado Kloos, (2007). Experiences with GRAIL: Learning Design support in. LRN, 2007. [Online]. Available: http://dspace.ou.nl/handle/1820/1019

Ioannis Magnisalis received the BSc degree in mathematics from the Aristotle University of Thessaloniki (AUTH), Greece, and the MSc degree in business systems analysis and design from the City University of London. He is currently working toward the PhD degree at AUTH. His research interests include adaptive and intelligent systems for collaborative learning.

Stavros Demetriadis received the BSc degree in physics, the MSc degree in electronic physics, and the PhD degree in multimedia educational technology from the Aristotle University of Thessaloniki, Greece. He is currently an associate professor in the Department of Informatics, Aristotle University of Thessaloniki, teaching courses and conducting research in the area of learning technologies. He is a member of the IEEE.