

A Tailorable Infrastructure to enhance Mobile Seamless Learning

Delfina Malandrino, Ilaria Manno, Giuseppina Palmieri, Vittorio Scarano, Luca Tateo, Daniele Casola, Ivan Ferrante, Francesco Foresta

Abstract—The widespread use of mobile devices is leading towards their adoption in the learning process, even if some pedagogical challenges are still not fully addressed when integrating mobile-assisted activities into regular curricula activities. In this paper, we first define some guidelines to design a general, tailorable, and platform-independent mobile learning system. Secondly, we present the aCME system, a mobile infrastructure, developed by adhering to the defined guidelines, to provide a general-purpose system, easy to start up with and friendly to user, and finally, tailorable, i.e., that can be easily adapted to the needs of different learning environments. The aCME system has been implemented as a Web-based architecture, to provide content and functionalities for different contexts, accessible from any location and by using any communication device. Finally, we describe a tool, deployed into aCME, that allows responding to quiz-based questionnaires during learning activities. A preliminary evaluation was performed to analyze usability and user satisfaction when interacting with the system we developed. The paper is concluded with some comments and future research directions.

Index Terms—Mobile Infrastructure, Mobile Learning, Web-based Systems, Tailorability, Usability

1 INTRODUCTION

The use of mobile technologies in learning processes, both in formal and informal educational contexts, has exponentially increased in the last twenty years [1]. It has given rise to a mix of enthusiasm, creativity, but also resistance to this new way of using personal devices in 1:1 settings (i.e., one mobile device per learner). The use of personal devices is perceived as a threat for the control of the learning process on a collective dimension. The formal educational contexts still have problems in integrating the philosophy of mobile learning (mobility, use of time resource, personalization, etc.) into everyday school routines [2]–[5]. Nevertheless, there are several reasons that encourage the integration of the mobile learning into the formal education. Mobile is a pervasive and familiar technology that has already colonized our daily life, and therefore, no specific training is required for both students and teachers. This is not a secondary advantage, as the most important existing artifacts of learning -the book and the computer- are appropriated by the individuals since the very young age. Mobile devices are quick following the same fate. Thus, the question is not whether mobile devices are suitable as learning tools or not, but which is the specific educational context of activity they can be used for. In this sense, we are going to present some

pedagogical challenges that should be addressed to make effective the integration of mobile devices into learning activities (Section 1.1). Then, we compare these pedagogical challenges with the features provided by mobile-assisted seamless learning systems available in the literature (Section 1.2). Additionally, to address the defined pedagogical challenges, we present a set of design guidelines that are able to ensure a full-featured mobile seamless learning system (Section 1.3).

1.1 Pedagogical motivations

We believe that mobile devices are not able to improve learning activities just because they are *mobile*: the real pedagogical issue is about how to embed mobile-assisted activities in regular curricula activities. In this sense, the first Pedagogical Challenge (*PC1*) we define is about the *integration of learning theories and mobile learning activities*, a crucial point still under development [6]. The number of provided applications, in fact, has been often technology-based, since the technology was the driving force to foster the experimentation of new didactic activities without a parallel development of educational models.

From a pedagogical point of view, learning is not a simple transfer of knowledge from a teacher to a learner, but is rather an active process of involvement and construction of knowledge. It takes place in a real context, in which students must be encouraged to take advantage of learning resources that exist both inside and outside the classroom. Students are asked to use their life experiences to make meaning of material introduced in classes [7]. This perspective on human

- D. Malandrino, I. Manno, G. Palmieri, V. Scarano, D. Casola, I. Ferrante and F. Foresta are with Dipartimento di Informatica, Università di Salerno (Italy).
E-mail: (delmal, manno, palmieri, vitsca)@dia.unisa.it
- Luca Tateo is with Centre for Cultural Psychology, Aalborg University, Denmark. E-mail: luca@hum.aau.dk

learning cannot be developed in a segregated unity of time and space, such as the classroom. It requires to be unfolded in a wider context by enabling the learner to use information across time and space and acting upon the world, in collaboration with others in different life contexts. Therefore, the second Pedagogical Challenge (*PC2*) we define is how to *integrate mobile learning activities in a real context*, not only inside or outside the classroom but also by covering a variety of topics and activities.

While the “seamless learning” implies that students can learn whenever they are curious in a variety of scenarios and that they can switch from one scenario to another easily and quickly using the personal device as a mediator [8], we believe that just the usage of mobile devices cannot address these pedagogical challenges and that mobile-assisted learning systems should be properly designed to enable effective interactions.

1.2 Mobile Seamless Learning

Relevant features in the design of mobile-assisted seamless learning (MSL) systems were evaluated in a wide and thorough review by Wong and Looi [9]. They defined 10 different features that MSL systems should provide to evolve from “learning by using a personal device as mediator” towards “mobile-assisted seamless learning”. In particular, they found that the MSL research community was focused on *encompassing formal and informal learning, encompassing personalized and social learning, mediating learning across time and locations, providing ubiquitous knowledge access, and encompassing physical and digital worlds*.

On the other hand, they found that more work is needed to *encompass multiple pedagogical or learning activity models*: how multiple pedagogical or learning activity models could be incorporated into an MSL activity design, and be seamlessly stitched together in the learning flow (MSL10 in [9]). This confirms that the *PC1* about the *integration of learning theories and mobile learning activities* is still an open problem.

Moreover, they recommend further studies to support the *seamless switching between multiple learning tasks* (MSL8 in [9]). To this aim, they encourage the integration of a variety of personal and group inquiry tasks into the learning flows. Indeed, MSL systems should be able to satisfy a wide set of settings and learning aims [10] (general debating or domain-specific topics, enhancing critical thinking, problem solving, students’ engagement, performance evaluation), and should be flexible enough to cover a large number of topics [11] (i.e., environmental education, science education, mathematics, geometry, language education and so on). This strengthens the definition of the *PC2* about the need to integrate mobile learning activities in a real context.

In addition, Wong and Looi suggest greater efforts to address the *seamless use of multiple device types*

(MSL7 in [9]). Indeed, the usage of multiple device types is intended by the authors as mixed usage of mobile and stable technologies (Desktop PC). Here we want to extend this consideration, since in a learning context no assumption can be made in advance about devices owned by the students. As a consequence, the system should work with any hardware and operating system, which is a critical issue when using mobile devices.

The MSL features identified by Wong and Looi represent the technological aspects of the Pedagogical Challenges that we identified in Section 1.1. We believe that these issues can be faced through a proper design, and therefore, in the following we provide a set of non-functional guidelines which allow to address the identified pedagogical challenges and provide a full-featured MSL system.

1.3 Our MSL Design Guidelines

The first crucial aim of the architecture design is to ensure that the system could support a wide set of usage scenarios, in terms of context of use and functionalities. These goals are well-known in the CSCW and CSCL (CSCW-L) research areas, where some guidelines have been defined to design groupware. We believe that these guidelines should be taken into account even in the development of systems to support mobile learning. Therefore, the system should be designed to be as “General” as possible, to provide basic functionalities non limited to a specific domain only. At the same time, the system should support an easy integration of new functionalities. This concept is well-known as “Tailorability”: the system should be able to be adapted to different settings as well as to evolve to fit new scenarios [12]–[19]. The groupware Tailorability can be considered from different points of view:

- Tailorability by Customization: it allows to configure the basic properties of the system to modify the behavior of the provided functionalities
- Tailorability by Integration: it allows the user to choose the desired functionalities from a predefined set of provided ones
- Tailorability by Expansion (or Extension): it allows to integrate new functionalities, so that the system can evolve to envision new users’ needs and usage scenarios.

Generality and Tailorability are able to address the *seamless switching between multiple learning tasks* (MSL8 in [9]), since they allow to design a system not devoted to a specific domain and to extend the system functionalities to support new tasks. Moreover, these guidelines allow to design a system able to *encompass multiple pedagogical or learning activity models* (MSL10 in [9]). Indeed, the user can customize the system and can choose the most appropriated functionalities

to follow the desired pedagogical model, addressing therefore the *PC1*.

Other fundamental guidelines are about the availability of the system: each learner, with any device and with any technical skill, should be able to easily access to the system. To this aim, a fundamental guideline in the design of an MSL system is the **“Platform-Independence”**: it should work with any hardware and operating system, since it is not possible to make assumptions about the devices owned by students. The systems that support mobile devices must take into account their limitations and issues, in particular the screen size limitation and the variety of existing HW/SW platforms. Moreover the system should provide a **“Simple User Interface”**; of course, this is true for every software, but it is crucial for mobile systems used in real time by groups of users since a complicated usage could discourage their adoption. In particular, in the classroom, the system should be ready to use in just a few minutes to prevent that students get quickly bored. In addition, the system should offer a **“Low cost deployment”**, allowing users to adopt the system without facing neither installations problems nor difficult configuration of complex systems. These guidelines aim not only to address the *seamless use of multiple device types* (MSL7 in [9]), but also to face an indirect pedagogical issue about the inclusion and involvement of learners that use different devices and with diversified skills.

Organization of the work.

The rest of the paper is organized as follows. In Section 2 we present an overview of our software infrastructure, designed by adhering to the defined MSL Design Guidelines, to support mobile-assisted seamless learning. The architecture of the system is tailorable and platform-independent, allowing the system to support a wide set of usage scenarios, in terms of supported devices, learning aims and learning domains. We also defined a set of functional requirements (Section 2.1) which define common functionalities that an MSL system should provide. In that Section we provide an overview of the system, a description of the provided functionalities and the envisioned usage scenarios. In Section 3 we present the architecture of aCME system and also how the system can be extended by integrating additional functionalities, therefore addressing the tailorability guideline. In Section 4 we describe the *Text Quiz Tool*, a mobile implementation of a Student Response System deployed into the aCME system. We also report a first evaluation of the usability and the user satisfaction when interacting with the tool. In Section 5 we present a comparison with existing works in the same field, by highlighting the main differences with our work. Finally, in Section 6 we conclude with some final remarks and future works.

2 SYSTEM OVERVIEW

The aCME system¹ has been developed to allow people to get involved in mobile learning activities and improve their overall engagement and performance. Several users can be supported at a time, whereas each of them can access to the learning content through any device, from any location, by using any access network and without installing any software.

The Web infrastructure is populated by a myriad of devices (smartphones and tablets) having different capabilities and input/output modalities. Built-in and pre-installed software, like Web browsers, having heavy processing tasks executed on robust server infrastructures, could represent a solution to meet different clients’ capabilities and a wide range of users’ preferences.

2.1 Functional Requirements

In addition to the design guidelines defined in Section 1.3, the CSCW-L areas have also defined several functional requirements for the development of groupware systems. A classification of such requirements and the corresponding services are illustrated in Fig. 1, with basic services listed at the bottom and the advanced ones in the top of the pyramid. In the following we describe the services, while details about the relationships among them are presented in [19].

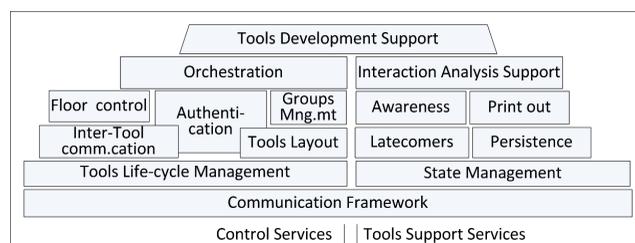


Fig. 1: Pyramid of services supporting the development of learning-based and collaborative tools.

Since the groupware involves several users who collaborate to reach a common goal, the application should leverage a *Communication Framework* to support the communication among users (and therefore among several instances of the application). The organization of the further requirements is strictly related to the Tailorability guideline and to the possibility of integrating new functionalities as additional modules or *Tools*. In general terms, a tool offers a set of self-contained functionalities for specific tasks (i.e., chat, diagram drawing and so on). A system which allows the integration of additional tools requires a set of services to manage the tools. Moreover, it should provide a set of services shared by all the tools to improve the overall system integration. Therefore, services can be organized in *Control Services* and *Tools Support Services*.

1. <https://sourceforge.net/projects/acme-groupware/>

The *Control Services* provide general functionalities to manage the tools:

- The *Tools Life-cycle Management* is strictly related to the Tailorability guideline: if the system allows to integrate new functionalities as additional modules, then it should also provide functionalities to manage their life-cycle (i.e., discovering, start and stop and so on).
- Since the system permits the integration of several tools with different functionalities, the *Inter-Tool communication* service allows to exchange information among different tools.
- The *Floor control* service lets the system to regulate users' interactions, enabling and disabling users' actions on the system.
- The *Authentication* service allows users to be identified, by using only nicknames or full credentials.
- The *Groups Management* service supports the organization of users into groups.
- The *Tools Layout* service concerns the visualization of the tools. In mobile-based systems, effectiveness of visualization is an important issue since their hardware limitations (i.e., screen size).
- The *Orchestration* service allows to organize in advance, as a structured sequence of activities, the interactions that have to be executed with specific tools; a more flexible orchestration can allow users to select the desired tools on demand.

The *Tools Support Services* provide orthogonal functionalities, common to each provided tool:

- The *State Management* service manages the state of the whole system (set of activated tools and their sequence of activation) and provides support to manage the content of each tool.
- The *Latecomers Management* service provides support for late users who join a lesson when it has already started. The system should provide a mechanism so that latecomer users can achieve the set of active tools and their content. It exploits the facilities provided by the *State Management* service.
- The *Awareness* service allows to gather information about users' activities and provides corresponding feedback (for example, which tool a user is using and how many contributions he has made). The kind of feedback provided depends on the specific used tool.
- The *Persistence* service allows to save and reload the state of the system (the set of activated tools with their activation sequence and the content of each tool).
- The *Print out* service is a functionality offered at the core level. Each tool has to implement its own output functionalities while the core is able to create a well-blended output. Similar to the *Latecomers Management* service, it exploits the facilities provided by the *State Management* service.

In addition, a fundamental step to improve the expandability of a groupware is the *Tools Development Support*, which should provide facilities for the implementations of new tools.

The design guidelines defined in Section 1.3 and the requirements just described have been used in the development of CoFFEE [18], [19], a suite of collaborative stand-alone desktop applications supporting learning activities in the classroom. CoFFEE has been developed within the European-funded Research Project called "LEAD" (VI Framework), and it is available open-source on Sourceforge². Additionally, it offers 15 different tools providing different functionalities [20]–[22]. The long-lasting adoption of CoFFEE, due to its tailorable architecture, lead us to extend this work on a mobile environment.

2.2 Overview

From the technical point of view, the aCME system is a distributed three-tier architecture developed in Java. The overview of the aCME system is shown in Fig. 2. Specifically, the *Server Component* is composed of modules deployed in the JBOSS application server³. The OSGi Alliance technology⁴ (or Open Service Gateway initiative) allows to integrate new modules (i.e., bundles), by leveraging its plug-in based architecture.

The *Client Components* are Web browsers executing a GUI enhanced with Vaadin widgets and add-ons⁵. Three types of applications have been developed, the *Administrator*, the *Controller* and, the *Discusser* Applications. Finally, persistent data are stored in a MySQL database, while communication data (JSON) are exchanged through the HTTP protocol.

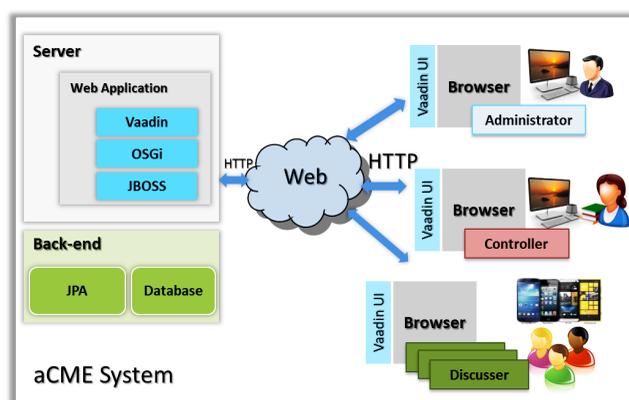


Fig. 2: The aCME System Architecture Overview.

The main characteristics of the aCME system are platform independence, tailorability, and low cost deployment. In particular, the Tailorability by Extension

2. It has been downloaded more than 13.895 times from Sourceforge, since October 2008, by more than 128 countries

3. <http://www.jboss.org/products/eap.html>

4. <http://www.osgi.org/>

5. <https://vaadin.com/home>

guideline of the aCME system will be described in detail in Section 3.4. Additionally, collaborative functionalities have been envisioned to allow the development of collaborative tools.

The technological choices, made during the development of the aCME system, were influenced by the need to address all these requirements.

The Vaadin Framework. The Vaadin Framework is a Java Web application development framework that is designed to develop Rich Internet Applications (RIA). Its main goal is to allow the easy creation and maintenance of high quality Web-based user interfaces.

The OSGi service platform and JBoss. The OSGi platform provides a dynamic service platform where modules (components and services) can be deployed and un-deployed at runtime without rebooting the system. The implementation of the aCME system components heavily relies on the OSGi bundles and web-bundles that can be deployed directly within the JBoss Application Server, an extensible, reflective, and dynamically reconfigurable Java application server.

Java Persistence API. Java Persistence API (or JPA 2.0) represents a Java specification for accessing, persisting and managing data between Java objects and a relational database. The process of mapping Java objects to database tables and vice versa is known as Object-relational mapping (ORM). The implementation we used for the aCME system is Hibernate⁶. Persistent data are stored in a MySQL database.

2.3 Functionalities

In the aCME system we can distinguish three types of roles with specific tasks: (1) the *Administrator* role, created at the first startup of the system; (2) the *Controller* role identifies the instructor (i.e., the teacher, the presenter of a talk, and so on); (3) the *Discusser* role identifies students (enrolled in a course), participants to a talk, and so on. People in the first two roles (Administrator and Controller) can access to the system through a Desktop PC or a tablet, while people entering the system with a role of Discusser can efficiently use smartphones or tablets.

The *Administrator* Application manages all tasks concerning administrative support. Examples of tasks include users management (users information and credentials), scenarios management (Organization, Courses, Subject, and Lessons information) and so on.

The *Controller* Application allows the instructors to organize their courses or allows the presenters to manage their talks. Specifically, it displays on its User Interface: (1) *Courses*: the teacher can choose among a list of all courses taught by him/her; (2) *Conference*: the presenter can create a new Conference with a common

one-time password for the authentication, given to the attendees to take part in the presentation.

After starting the learning activity (Course or Conference), the instructor can use all the tools provided by the system. By default, a *User Presence* list is provided to the instructor, in order to have information about the users that are currently online. All tools that we developed follow the Tailorability by Customization guideline, i.e., all tools have configurable options, so that they can fit different users' needs. Some of the configurable options are common to all the tools, as an example, the anonymity. We envisioned it since, as discussed in [23], the anonymity encourages people to participate and reduce groupthink by evaluating ideas more objectively. Before enabling a tool, the instructor, through the *Controller* Application, can select one of the following options:

- *Full Anonymity*: Participants see all others contributions but not the identity of the owners of those contributions. Only the identity of the teacher is known. On the screen of participants' mobile phones, a red line will highlight this feature.
- *Partial Anonymity*: Only the teacher knows the identities of the participants and can recognize their contributions. On the screen of participants, a blue line will highlight this feature.
- *No Anonymity*: It represents the default option. On the screen of participants, a green line will highlight this feature.

In addition, the instructor can regulate users' interactions by blocking/unblocking their individual activities (Floor Control). The students continue to see the interactions among the other participants, although they are blocked and cannot actively participate. Besides, the *Controller* Application allows the instructor to create a printable report of the content of the tools, to keep track of all interactions and discussions during the learning activity.

Finally, the *Discusser* Application is used by each student to participate in a discussion. It provides two different types of access: (1) A simple *Login*, by providing credentials, to see the list of all courses taught by the teacher, with all the corresponding lessons. After having selected a course, and next a lesson, students can start with the activity scheduled by the teacher; (2) *Login Fly* without credentials: the *Discusser* Application allows the student to attend a Conference. The student has only to provide the one-time password, appositely chosen by the instructor/presenter for that activity.

2.4 Usage Scenarios

The aCME system is able to support different usage scenarios. In the following we describe some learning activities specifically envisioned for a *School Scenario* and for a *Conference Scenario*.

6. <http://hibernate.org/>

- **Asking for questions.** If during a lesson a concept is not fully understood by a student, he/she can send a question to the teacher without blocking him/her. The teacher, at the end of the explanation, checks the received questions and provides an answer to each of the students.
- **Classroom debate.** A teacher can ask his/her students to study a specific topic, to debate about it, and to collaborate trying to address their doubts and perplexities. Next, the teacher can use the aCME system tools to start the discussion. Students will be able to help each other, by talking and discussing through these tools. At the time of writing, the tools provided by the aCME system are the *Chat Tool* and the *Text Quiz Tool*. New tools can be quickly developed since the flexibility and the modularity of the aCME system.
- **Assessment questionnaire.** A speaker at the end of a talk can decide to submit an evaluation questionnaire to all participants to verify the quality of the given talk. Participants can access to the aCME system and, in an anonymous way, provide responses to all the posed questions. Afterwards, the speaker can collect the results to analyze them.

We have also defined an abstract model in order to map the different entities that can be managed by the aCME system in the designed scenarios. At the top of the hierarchy we have the involved *Organization* (i.e., “University”, “Secondary school” or “Primary school”). The second layer of our abstract model describes the specific *Courses*, while the third and the fourth ones describe the generic arguments (i.e., *Subject* Entities) covered by the *Lessons*, respectively. As an example, for the *School* Scenario, the *Course* Entity corresponds to a specific class (i.e., Class 1A, 2A, 3B, etc.) while for the *University* scenario, it corresponds to a specific field of study (i.e., Computer Science, Chemistry, Physics and so on). The *Subject* Entity corresponds to a specific subject (i.e., Mathematics) in the *School* Scenario, and to a course taught in the *University* Scenario (i.e., Network Programming, Security, and so on). In the *Conference* scenario, the *Organization* corresponds to the entity responsible for an event and *Lessons* corresponds to talks, while *Courses* and *Subjects* are not needed.

3 ACME ARCHITECTURE

In this Section we describe the layered architecture of the aCME system, shown in Fig. 3. Specifically, we will describe the functionalities provided by each of the 3 implemented layers, that is, the “*Data Management Layer*”, the “*Services Layer*”, and the “*Presentation Layer*”, respectively.

3.1 Data Management Layer

The main goal of the *Data Management Layer* is to manage users and materials needed for the organization of a learning activity. Specifically, as shown in Fig. 3, it manages the following entities (described in Section 2.4):

- *Organization*: it manages information about the involved Organization
- *Users*: it manages the different users that can access to the system, e.g. administrators, instructors, and finally, learners/students/attendees
- *Courses*: it allows to manage the learning materials for all the envisioned scenarios
- *Tools Content*: it allows to manage the content of the tools designed to support the learning activities
- *Lessons*: it allows to manage the lessons and the tools required for each of them
- *Subjects*: it allows to manage combination of teachers-subjects.

As discussed in the previous Section, we used JPA to implement the Java classes for the management of these entities and their mapping into a MySQL database. Moreover, the *Data Management Layer* provides the persistence service to the upper layers.

3.2 Services Layer

The *Service Layer* represents the *core layer* of the aCME system and provides a wide set of functionalities designed and developed to address the requirements⁷ described in Section 1.3. We will describe these functionalities in the following, paying attention to the mapping with the diagram shown in the Fig. 1.

The *Tool Services* manages the interactions of the *Controller* Application with the *Presentation Layer* with regard to the activation of the tools. It creates a *Tool Management* service for each tool requested by the *Controller* Application. Additionally, it provides functionalities that are common to each tool and ensures the independence of the tool from other parts of our system. Specifically, the *Tool Management* is in charge of discovering and starting the tools (*Tools Life-cycle Management* Requirement in the pyramid in Fig. 1). It is also able to manage multiple instances of multiple tools. The aCME system provides this functionality leveraging the OSGi architecture, which offers a well established mechanism for the integration of new tools. Moreover, the *Tool Management* allows to bind the *Controller* Application with several instances of the *Discusser* Application, by enabling the communication between the participants (*Communication Framework* in Fig. 1).

The *Tool Management* service, being responsible for all messages exchanged among the tool components,

7. Except the Persistence requirement in Fig. 1, which is provided by the Data Management Layer

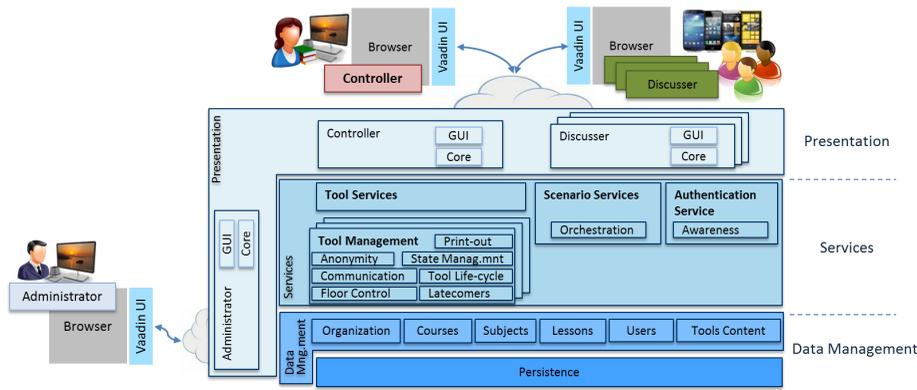


Fig. 3: Layered architecture of the aCME system. Presentation, Services and Data Management Layers.

stores them in a list in order to: (a) save the whole system state, in terms of the tools activated and their sequence of activation, (b) create the report of the content of the tools (the core is able to create a well-blended output by using the specific output functionalities provided by each tool) and (c) support latecomer users by providing them with a “history” of the messages exchanged (*State Management*, *Latecomers* and *Print out* in Fig. 1).

Finally, the *Tool Management* service provides the *Anonymity* functionality by applying the filtering process on the exchanged messages, and the *Floor Control* functionality by disabling/enabling the GUI of the tools. In this way, students will continue to see the interactions flowing within the aCME system even if they have been blocked, and not able to participate anymore to the ongoing activity (*Anonymity* and *Floor control* in Fig. 1).

The *Scenario Services* manages the interactions of the *Controller* and *Discussers* Applications with the entities involved in the usage scenarios: the *Controller* (i.e., the teacher) and the *Discussers* (i.e., the students) can select the course and lesson to attend, then the *Controller* can configure the lesson by choosing among provided tools. It may decide to organize the lesson with more than one tool and each lesson can be configured by selecting the anonymity functionality. Finally, the *Controller* starts with the lesson. The functionalities provided by the *Scenario services* allow to structure the activities in advance, providing orchestration functionalities (*Orchestration* in Fig. 1).

Finally, the *Authentication Service* manages the authentication phase. It provides the *Controller* Application with notifications of the users who are connected and with optional information about specific users (*Authentication* and *Awareness* in Fig. 1).

Some further requirements shown in Fig. 1 have not been considered in our context. In particular, the *Tools Layout*, intended as visualization of multiple tools, is not needed because of the devices’ space constraints. Other functionalities as *Groups Management*, *Inter-Tool*

communication, *Interactions Analysis Support* are tasks that we are planning to complete in the near future.

Finally, the *Tools Development Support* represents a fundamental step to improve the expandability of a groupware. Our approach to provide extensibility in the aCME system is described in detail in Section 3.4.

3.3 Presentation Layer

The *Presentation Layer* or Front-End, at the top of our layered architecture, represents the Web view for the final users. A session is created the first time users (teachers, students or administrators) access to the system. Specifically, a session starts when a user opens an aCME system application (i.e., *Controller*, *Discussers* or the *Administrator* Application), and ends when the session expires in the server or when it is explicitly closed.

For the implementation of the Front-End, we employed the Vaadin technology which supports two different programming models: server-side and client-side. The server-side development model employs the Vaadin Client-Side Engine to render the user interfaces in the browser (see Fig. 4). The client-side model allows to develop widgets and applications in Java, which are compiled in JavaScript and executed in the browser. The two models can share widgets, themes, and back-end code and services. Specifically, in our architecture, each Vaadin application (*Controller*, *Discussers* or *Administrator*) is composed of a *User Interface* (UI) that runs on the Web browser and allows the user to interact with the business logic and the application data. The business logic, implemented in a *Core* module, runs as a Java Servlet (Desktop Servlet for the *Administration* and the *Controller* Applications, and Mobile Servlet for the *Discussers* Application, as shown in Fig. 4) in a Java application server (i.e., JBoss).

For the *Administrator* Application, the *Core* module only manages the communication with the *Persistence Layer* for users and scenarios management (e.g., *Organization*, *Courses*, *Subjects* and so on). For the *Controller* and *Discussers* Applications, it also manages the communication with the *Services Layer* for the

management of the services (i.e., Scenario Services, Tool Services and Authentication Service).

Finally, the Vaadin TouchKit⁸ widget and the JFreeChart widget⁹ have been used to implement the UI of the *Discusser* Application and to display charts in the *Controller* Application, respectively. Details about used widgets, add-ons, and themes are shown in Fig 4.

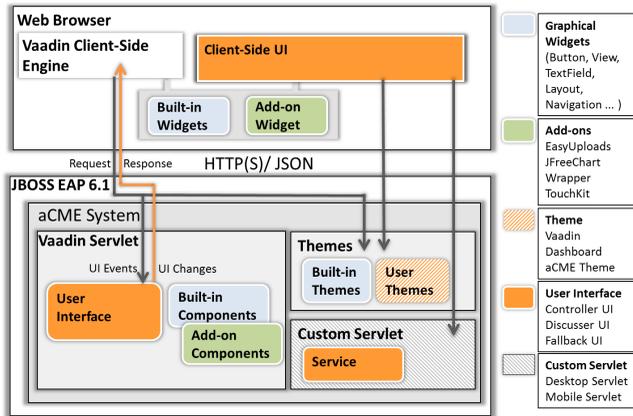


Fig. 4: aCME runtime architecture and technologies.

3.4 System Extensibility

The aCME system has been designed to ensure extensibility, in order to integrate new functionalities developed as *aCME Tools*. The components of a generic tool are illustrated in Fig. 5. aCME Tools are OSGi bundles (the box delimited by the thin dotted line in Fig. 5) with a component integrated in the *Controller* Application (i.e., the *Generic Tool Controller-side component*), a component integrated in the *Discusser* Application (i.e., the *Generic Tool Discusser-side component*) and, finally, a component used by the *Services Layer* to activate the tool (i.e., the *Tool Manager component*).

The Controller-side and the Discusser-side components of the tool provide the GUI and the implementation of the functionalities to the *Controller* and to the *Discusser* Applications, respectively. The *Tool Manager* component is responsible for the activation of the tools.

Tools activation. The user, through the *Controller* Application, can activate each of the tools available in the system. The sequence of steps for a tool activation is shown in Fig. 5 (continuous arrows with labels 1, 2, 3 and 4) and can be summarized as follows:

- the *Controller* Application is responsible for creating and managing an instance of the *Tool Management* component for each activated tool
- the *Tool Management* component manages the tool life-cycle and provides the tool with the orthogonal functionalities of Anonymity, Latecomers,

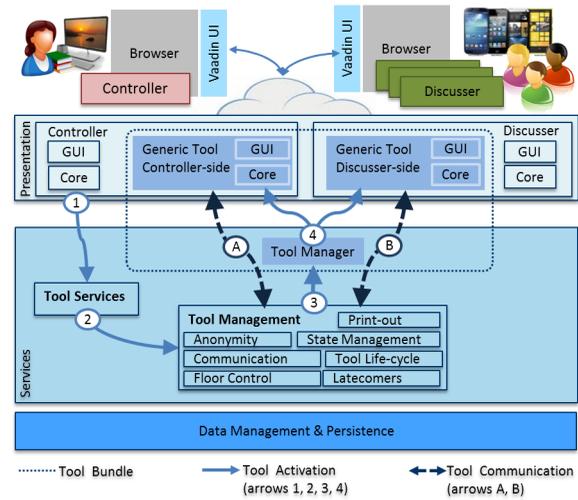


Fig. 5: Tool activation and communication.

- Floor Control and Communication; it starts the tool by creating the *Tool Manager* component
- the *Tool Manager* component is responsible for creating the Controller-side and the Discusser-side components of the tool.

Tools communication. The Controller-side and the Discusser-side components of a tool communicate through the *Tool Management* component (fat dotted arrows in Fig. 5 with labels A and B). This component manages the message exchanged, as well as, details about how to guarantee anonymity (by filtering the user names in the tools for which the anonymous mode has been enabled), and latecomers management (by providing the content of the tool in the form of history of messages).

Tools development. Each new tool of the aCME system extends the Generic Tool component to create its own specific implementation. Our goal was to simplify the development of new tools by allowing the developers to focus on the application logic of the new tool, and exploiting the basic functionalities provided by the system itself. Furthermore, the development of new tools has been also supported through the creation of a wizard, that is able to automatically create the structure of an aCME Tool, therefore, simplifying the setup phase in the development process.

4 TEXT QUIZ TOOL

In this Section we first describe the tool we have developed and deployed into the aCME system to assess its extensibility, and next, the preliminary evaluation studies that we conducted to analyze its usability and user satisfaction.

The *Text Quiz Tool* is a mobile implementation of a Student Response System (SRS). In learning environments, SRSs allow teachers to pose questions to their students in the classroom, receiving answers

8. <https://vaadin.com/add-ons/touchkit>

9. <http://www.jfree.org/jfreechart/>

from them in real time. Teachers can use SRSs to enhance the reflexive and creative capability of the learner about the topic of the activity. Several studies showed that clickers, the first implementation of SRSs, are able to produce improvements in students' engagement, participation, feedback to the teacher, and learning performance. The main drawback of using clickers is related to their cost, a critical issue both for individual students and public institutions. In literature, several papers describe benefits and challenges of clickers [24]–[26]. Recently SRSs have been realized as Web-based software (i.e., GoSoapBox¹⁰, Poll everywhere¹¹, EverySlide¹²). Similar to clickers, they are able to improve participation, engagement, and feedback about students' understanding [27], [28].

Recent results about the usefulness of Quiz-based Tools on improving students' engagement and participation in the classroom [21], lead us to develop a mobile version of an SRS, by deploying it into the aCME system. We present its functionalities and a preliminary evaluation of its usability in the following.

4.1 Design and Functionalities

In this Section we present the *Text Quiz Tool* which allows the teacher to give a quiz-based questionnaire to the students, and collects their answers in real time. Results are shown to the teacher as a chart. They can be projected on a screen visible to everybody and used to start a discussion. The teacher can get quick feedback on the performance of the students, so that he can immediately evaluate their involvement and comprehension about the covered topics.

The *Text Quiz Tool* allows teachers to encourage and motivate the students' participation during learning activities. An important aspect of this tool is that students can answer anonymously. In this way, students can feel more comfortable when they are involved in discussions, when they have to express opinions, and finally, when they answer to questions.

As discussed in Section 2.4, a typical usage of the *Text Quiz Tool* is the following: the teacher explains the arguments of the lesson in the classroom; then, he tries to understand if that topic was understood by the students by giving them a questionnaire with questions inherent the discussed argument. This tool enables teachers to have an immediate feedback for each question, only by glancing at the chart, displayed in the interface of the teacher, as shown in Fig. 6(a).

Students will receive a feedback about the correctness of their answers. Specifically, they will receive the correct answer and possible hints by the teacher (See Fig. 6(b)).

The *Text Quiz Tool* may contain several types of questions, that is, multiple choice questions, questions

requiring numeric answers and so on. These questions are prepared by the teacher in advance by using Hot Potatoes¹³. The results can be exported as Excel files. The loading of the quiz inside the aCME system has been realized by using moodle libraries¹⁴, serialized in Java through implemented PHP libraries.

4.2 Preliminary Evaluation

In this Section we first describe the methodology that we employed for our evaluation studies, afterward we discuss the results as well as the improvements achieved when comments, feedback and suggestions by participants at the studies were addressed.

4.2.1 Method

In an earlier study, we showed that software tools can help students to collaborate and improve performance [29]. Here we want to analyze the participants' reaction to the software and their perceptions about the usability of the tested tool.

To analyze the usability of the aCME system and, specifically, both the usability of the *Text Quiz Tool* interface and the user satisfaction when interacting with it, we carried out two different usability studies. The first was carried out on 18th October, 2013. We analyzed the results of this first evaluation study, then we addressed all students' criticisms and suggestions by translating them into software changes and new system features. Finally, on 9th December, 2013 we repeated the study by recruiting a second sample of students (different from the sample of the first study). The idea was to evaluate if both usability and general experience were improved thanks to our intervention on the aCME system.

For our usability studies we recruited 19 participants among students of the Computer Science Department at the University of Salerno. Both studies were conducted at the ISISLab research laboratory, at the same University. Students were recruited through email announcements to mailing lists, and word of mouth advertising. The participation was voluntary, anonymous, and free since the students were not compensated for taking part in the studies. Finally, participants were informed that all the information they provided would remain confidential.

Our methodology, that we employed in an earlier work [30], envisioned two different phases. In the first phase we asked students to provide demographic information. Next, we asked them to participate in a testing session using the *Text Quiz Tool*.

The questionnaire provided to the students was a quiz-based questionnaire (loaded into the *Text Quiz Tool*), composed of 10 questions, about General

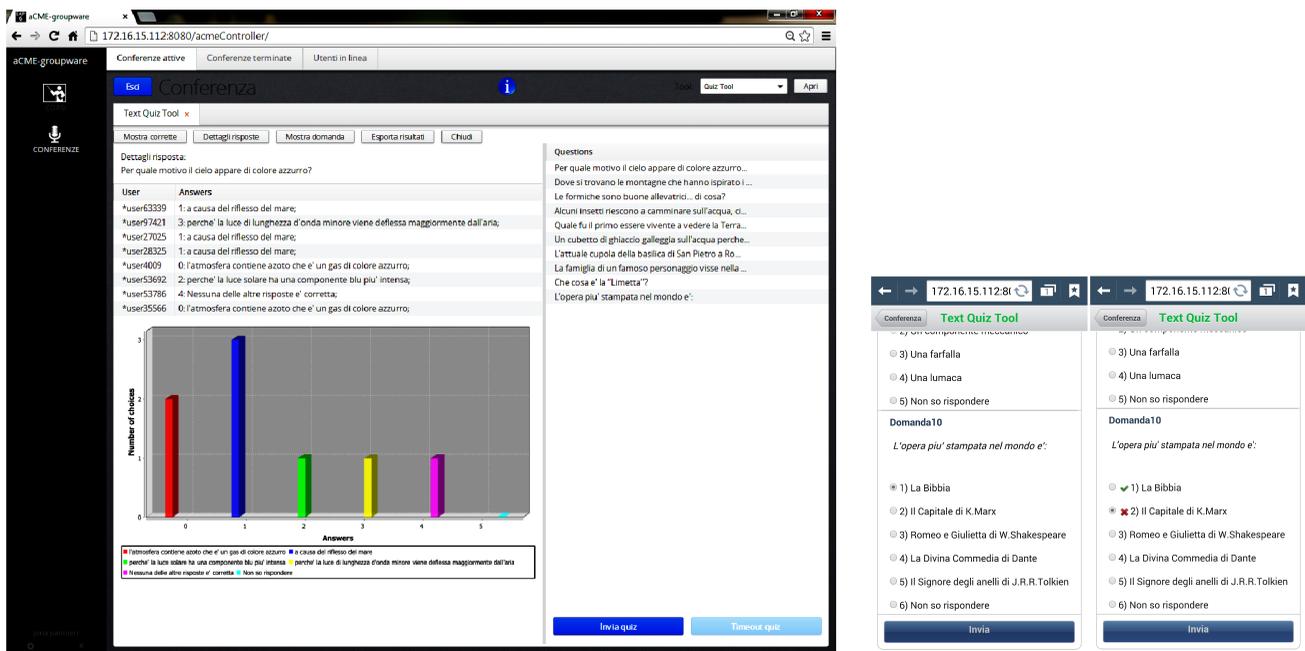
10. <http://app.gosoapbox.com/>

11. <http://www.polleverywhere.com/>

12. <http://everyslide.com>

13. <http://hotpot.uvic.ca/>

14. <https://moodle.org>



(a) Controller UI.

(b) Discusser UI

Fig. 6: (a) Feedback to the teacher about students' answers for a General Knowledge questionnaire. (b) UI of students before and after the feedback about their answers.

Knowledge Questions¹⁵. In the first study, the questions were given to the students one by one, while in the second study the questions are received all together. In the latter case, the students were given the opportunity to select the answer for each question and then send the set of answers all at once.

In the second phase, immediately after the end of the *Text Quiz Tool* testing phase, we asked students to spend other 10 minutes to fill out the standard QUIS¹⁶ and CSUQ¹⁷ questionnaires. The aim was to provide information about system usability and user satisfaction when using the aCME system. Specifically, the original QUIS questionnaire, developed at the University of Maryland, was composed of 27 questions. We dropped those that did not seem to be appropriate to our tool. Each question was a rating on a 10-point scale with appropriate anchors at each end (e.g., "Overall Reaction to the software: Terrible/Wonderful"), where small values corresponded to unsatisfactory or negative responses and large values corresponded to satisfactory results. The original CSUQ questionnaire, instead, was composed of 19 questions with answers expressing agreements or disagreements through a 7-point Likert scale with *strongly agree* and *strongly disagree* as verbal anchors. Both studies lasted between 30 and 40 minutes. Finally, both the QUIS and CSUQ questionnaires are

available online¹⁸.

4.2.2 Results

Participants were students in their third year of a Bachelor degree, and in their first and second years of Master degree at the University of Salerno. The majority of participants were male (84%) with a mean age of 25 years.

To analyze the results of the CSUQ questionnaire, we organized its questions into 6 different metrics, that is, "Satisfaction", "Easy of Use", "Easy of learning", "Usefulness", "Efficacy" and, finally, "Clearness". Before discussing the results, we have to emphasize that, as a result of the first study, we performed a series of changes to the aCME system in order to address all criticisms raised by the students. Additionally, since we allowed the students to provide an explanation for each of the questions included in the CSUQ questionnaire (we did the same for the QUIS questionnaire), a series of useful comments and suggestions were acquired. From the analysis of these comments we drew out some perplexities about the clearness and the intuitiveness, as well as, as a consequence, about the easiness of use and learning of the tested system.

Overall, as we can see in Fig. 7, all the posed questions were rated positively. While the Satisfaction and Usefulness metrics showed the same behavior across the two studies (we did not make changes that can justify improvements here), the other four

15. <http://www.di.unisa.it/~delmal/research/usability/aCME.pdf>

16. <http://oldwww.acm.org/perlman/question.cgi?form=QUIS>

17. <http://oldwww.acm.org/perlman/question.cgi?form=CSUQ>

18. <http://www.di.unisa.it/~delmal/research/usability/TLT.pdf>

metrics showed important improvements. At the end of the first study, the students expressed their dissatisfaction about the complexity of the User Interface of the system and the lack of feedback to users when some events happen. Specifically, 4 students out of 19 expressed their dissatisfaction about the choice to allow the horizontal scroll instead of a more intuitive vertical one. Additionally, 6 out of 19, stated that performing tasks was not straightforward, given the consistent number of needed actions to respond to a question (i.e., Select the answer–Accept the answer–Send the answer, with 3 different windows for each of the three different actions). Moreover, no feedback was envisioned to inform the users about the correct functioning of the tool (i.e., Are answers correctly sent to the instructor?), or about the delay due to connection problems (3 out of 19 students experienced this type of situation, by confusing the delay due to an overload of the network with a system crash).

In summary the changes that we made to the system were able to improve users' experience and to provide more positive comments to the questions posed in the second study. Specifically, we made several changes to the UI and we envisioned a series of both help and error messages, to allow the students to immediately recover from critical situations. Other changes were performed to solve minor problems (i.e., text not correctly shown).

The QUIS questionnaire shows the same improvement across the two studies. As we can see from Fig. 8, the most interesting results are about the "Screen" and "Learning" categories. Specifically, our changes were able to improve users' perceptions about the easiness and the satisfaction when using the tested system. Additionally, the UI changes were fruitful to improve perceptions about the "Organization of Information", "Sequence of screens", and "Performing task is straightforward" metrics.

Finally, the reliability of the various scales of the questions in the QUIS and CSUQ questionnaires was good (Cronbach's $\alpha = 0.95$ and 0.93 , respectively).

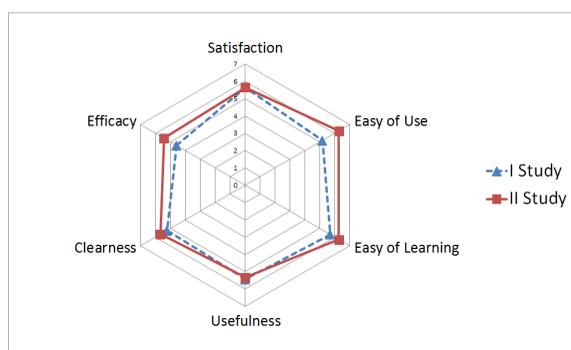


Fig. 7: CSUQ results according to: Satisfaction, Easy of Use and Learning, Efficacy and Clearness.

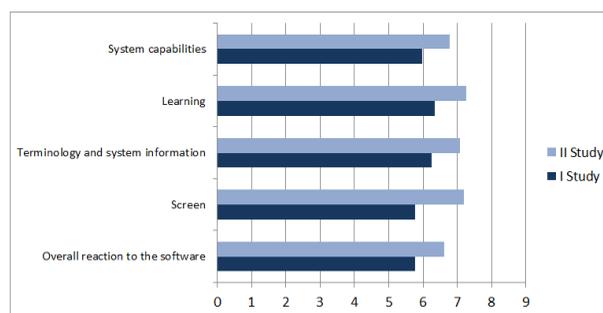


Fig. 8: QUIS results with agreement or disagreements through a 10-point Likert scale.

5 RELATED WORKS

In this Section we compare our system with related works in literature, paying particular attention to the defined design guidelines and MSL features. As discussed in Section 1, Wong and Looi [9] found that more efforts are needed to support the *seamless use of multiple device types*, the *seamless switching between multiple learning tasks* and, to *encompass multiple pedagogical or learning activity models*. The aCME system supports seamless use of multiple device types since it is device-independent. Moreover, it has been designed to be general and tailorable to improve seamless switching between multiple learning tasks and to foster the adoption of learning theories by offering support for a wide set of usage scenario.

However, the issue of platform-independence has been faced in several studies. Some of them are focused on content presentation on different client devices. Christ and Feißt [31] proposed a Web-based approach where the content is separated from its visual representation, so that the content visualization can be adapted after devices' identification and classification. Meawad and Stubbs [32] developed a Web-based framework to offer platform-independent content visualization on mobile devices for a heterogeneous set of Virtual Learning Environments (VLE, such as Moodle, Sakai, ATutor). The framework includes a mini browser developed by using Java-ME. Despite the Java claim "write once, run anywhere", the authors clashed with several problems reaching partially the platform independence. Another approach to provide platform-independent content visualization is described by Giemza et al. [33]. The authors leverage the Model-View-Controller design pattern to define a shared-model approach, which provides the possibility to implement devices-specific view-controller components for the same underlying data model. Another Web-based platform-independent approach is presented in the COLLAGE project [34]. The authors developed a platform to create *games* related to specific sites (museums, archeological sites, and so on); for each game the teacher can define multiple-choice questions (*learning path*). The students can answer the questions and can provide comments.

All these works provide platform-independent visualization, but they do not exhibit tailorable capabilities. However, it must be emphasized that adaptation and presentation, needed to make content accessible *anytime, anywhere, and from any access device*, is a well-known issue in the field of mobile and ubiquitous environments, regardless of the specific context in which mobile devices are used [35]–[37].

Few works present studies addressing tailorability and extensibility functionalities. Boticki et al. [38], [39] present an extensible framework providing generic collaboration features as foundation for domain specific modules. Among the others, the authors present two applicative modules: the FAO application, supporting fractions teaching, and the Chinese-PP application, supporting the Chinese language teaching. The framework is based on a client-server architecture, where the clients can be applications for desktop computers and applications for mobile devices. The applicative modules for mobile devices were implemented as applications to be installed, therefore, requiring a specific implementation for each mobile device platform. Therefore, while it is tailorable, this work is not platform-independent and moreover, it requires software installation on the devices.

Our system architecture goes beyond the state of the art since it is designed to be general and tailorable as well as platform-independent, therefore enabling system usage in a wide set of possibilities, in terms of supported devices and learning aims and domains. Moreover, the aCME system has a low cost deployment, since it does not require to install any software or to configure the mobile device.

6 CONCLUSION

The high proliferation of smartphones and hand-held devices is leading to an increasing interest in the adoption of such technologies in learning processes. However, mobile devices are not able to improve learning activities just because they are *mobile*: mobile learning systems should be designed to support the *integration of learning theories and mobile learning activities* (Pedagogical Challenge 1) and the *integration of mobile learning activities in a real context* (Pedagogical Challenge 2), by covering a variety of topics and activities across time and space. In their review, Wong and Looi [9] found that further studies are required to *encompass multiple pedagogical or learning activity models*, to support *seamless switching between multiple learning tasks* as well as to enhance *seamless use of multiple device types*. In order to face the pedagogical challenges and to provide a full-featured MSL system, we defined our MSL Design Guidelines. In particular, an MSL system should be able to satisfy a wide set of domains (*Generality* and *Tailorability*) and should be easy to use (*Simple User Interface*). Moreover, to address mobile issues, it should be easy to start up (*Low*

cost deployment) and it should work on any hardware and operating system (*Platform-independence*).

Beyond these design guidelines, we also identified a set of functional requirements ranging from the communication among the involved actors (students and teachers) and the possibility to integrate new functionalities. Following these guidelines and requirements, we developed the aCME system, a Web-based infrastructure to support mobile learning activities. Our Web-based approach guarantees device-independence, with our software components being used on a variety of devices, regardless of the underlying hardware. No software has to be installed and any plugin has to be loaded into the browser. The application logic has been implemented server-side, to reduce the burden on the mobile devices, given their limited processing capabilities. Moreover, the Bundle-based architecture of the aCME system is able to guarantee modularity, flexibility and finally extensibility. The Layered architecture allowed us to develop and deploy new modules, exploiting the core functionalities of the system. The linkage between the requirements (Design Guidelines and MSL features) and the adopted solutions (architectural design and technologies) is detailed in Fig. 9.

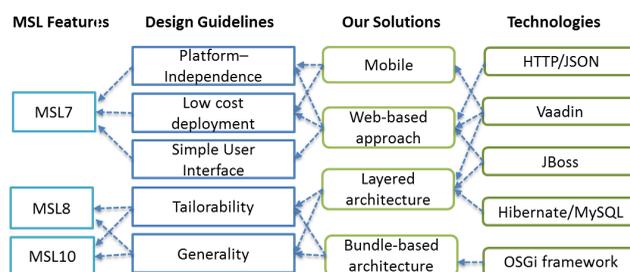


Fig. 9: Mapping from MSL features and Design Guidelines to solutions and technologies.

To prove the Tailorability guideline we presented the *Text Quiz Tool*, an instrument that implements Student Response System functionalities. We also analyzed the usability and the user satisfaction when using it in a Conference scenario setting. Specifically, we performed a preliminary study with students and teachers from the academia involved in two different evaluations. Criticisms, feedback and suggestions by the participants during the first evaluation study lead us to develop a new tool from scratch. The new tool was positively rated in the second evaluation study, with the most positive results achieved about the easiness of use and learning of the provided tool, and also its efficacy when used in a learning activity.

An interesting side effect of our usability survey was that we were able to test the flexibility of the underlying infrastructure on a real case. In fact, we were able to quickly address users' suggestions and, in about two weeks, we were able to design, implement, test and deploy a new tool that was successfully

evaluated by the second evaluation study. Moreover, we can exhibit some evidence of the flexibility of our framework, since the tools have small size and impact very lightly on the developer because they are fully supported by aCME system. To wit, an empty "Hello World" tool (as created by our wizard) is roughly 100 Lines of Code (LOC)¹⁹, while a simple chat is around 400 LOC, and the *Text Quiz Tool* (that requires a complex communication mechanism for the orchestration of the interactions and the different users' roles) is about 2000 LOC, where the whole infrastructure is one order of magnitude larger²⁰, not to mention the off-the-shelf JBoss application server where our framework is deployed.

We would like to emphasize that the evaluation of our system is somewhat limited. In fact, our preliminary evaluation studies only aimed at measuring user's satisfaction when interacting with the developed system and its interface. We are planning, as main future work, a more complete experimental study, in order to analyze whether the aCME system can improve overall engagement and performance when employed in educational environments. Perceived usefulness and ease of use metrics could be analyzed to derive insights about the acceptance of the software, by considering factors, such as, work effectiveness and productivity, and finally, system accessibility [40]. In particular, accessibility could significantly affect the system's acceptance from users with disabilities. Additionally, larger and more diverse sample sizes could result in more generalizable and significant results.

Moreover, another future direction will include benchmarking tests in order to analyze the overall performance of the system, in terms of reliability, responsiveness, and also, in terms of battery consumptions of mobile devices [41].

Finally, we are planning to develop new tools to enhance collaborative functionalities to allow cooperation during learning activities.

REFERENCES

- [1] D. Frohberg, C. Gth, and G. Schwabe, "Mobile Learning projects a critical analysis of the state of the art," *Journal of Computer Assisted Learning*, vol. 25, no. 4, pp. 307–331, 2009.
- [2] M. Ligorio and L. Tateo, "Mobile phones use in classroom activities: A psycho-social educational approach," *International Journal of Psychology*, vol. 43, pp. 502–502, 2008.
- [3] W. M. Baker, E. J. Lusk, and K. L. Neuhauser, "On the use of cell phones and other electronic devices in the classroom: Evidence from a survey of faculty and students," *Journal of Education for Business*, vol. 87, no. 5, pp. 275–289, Jan. 2012.
- [4] M. J. Bugeja, "Distractions in the wireless classroom," *Chronicle of Higher Education*, vol. 53(21), 2007.
- [5] D. R. Tindell and R. W. Bohlander, "The use and abuse of cell phones and text messaging in the classroom: A survey of college students," *College Teaching*, vol. 60, no. 1, p. 19, 2012.

- [6] Y. Park, "A pedagogical framework for mobile learning: Categorizing educational applications of mobile technologies into four types," *The International Review of Research in Open and Distance Learning*, vol. 12, no. 2, 2011.
- [7] G. D. Kuh, "Guiding principles for creating seamless learning environments for undergraduates," *Journal of College Student Development*, vol. 37, no. 2, pp. 135–148, 1996.
- [8] T.-W. Chan, J. Roschelle, S. Hsi, K. Kinshuk, M. Sharples, T. Brown, C. Patton, J. Cherniavsky, R. D. Pea, C. Norris, and others, "One-to-one technology-enhanced learning: An opportunity for global research collaboration," *Research and Practice in Technology Enhanced Learning*, vol. 1, p. 329, 2006.
- [9] L.-H. Wong and C.-K. Looi, "What seems do we remove in mobile-assisted seamless learning? a critical review of the literature," *Computers & Education*, vol. 57, no. 4, pp. 2364–2381, Dec. 2011.
- [10] J. Herrington, A. Herrington, J. Mantei, I. Olney, and B. Ferry, *New technologies, new pedagogies: Mobile learning in higher education*, University of Wollongong ed., 2009, ch. Using mobile technologies to develop new ways of teaching and learning, pp. 1–14.
- [11] A. Kukulska-Hulme, M. Sharples, M. Milrad, I. A. Sanchez, and G. N. Vavoula, "Innovation in mobile learning: A European perspective," *IJMBL*, vol. 1, no. 1, pp. 13–35, 2009.
- [12] M. Koch and G. Teege, "Support for tailoring CSCW systems: adaptation by composition," in *PDP '99*, 1999, pp. 146–152.
- [13] J. Hummes and B. Merialdo, "Design of Extensible Component-Based Groupware," *Comput. Supported Coop. Work*, vol. 9, no. 1, pp. 53–74, 2000.
- [14] R. Slagter, M. Biemans, and H. ter Hofte, "Evolution in Use of Groupware: Facilitating Tailoring to the Extreme," in *CRIWGW '01*, 2001, pp. 68–73.
- [15] A. Dimitracopoulou, "Designing collaborative learning systems: current trends & future research agenda," in *CSCL '05*, 2005, pp. 115–124.
- [16] J. Lonchamp, "Multi-Dimensional Model-based Genericity in Omega+," in *Advanced Learning Technologies, 2006. Sixth International Conference on*, 05-07 July 2006, pp. 730–734.
- [17] Lonchamp, "Supporting synchronous collaborative learning: a generic, multi-dimensional model," *CSCL Journal*, vol. 1, no. 2, pp. 247–276, June 2006.
- [18] R. De Chiara, A. Di Matteo, Ilaria Manno, and V. Scarano, "CoFFEE: Cooperative Face2Face Educational Environment," in *CollaborateCom*, 2007, pp. 243–252.
- [19] R. De Chiara, I. Manno, and V. Scarano, *In Educational Technologies for Teaching Argumentation Skills*. Bentham eBooks, 2012, ch. CoFFEE: an Expandable and Rich Platform for Computer-Mediated, Face-to-Face Argumentation in Classroom.
- [20] F. Belgiorio, D. Malandrino, I. Manno, G. Palmieri, and V. Scarano, "Computer-Supported WebQuests," in *EC-TEL*, vol. 5794, 2009, pp. 712–718.
- [21] D. Malandrino, I. Manno, G. Palmieri, V. Scarano, and G. Filatrella, "How quiz-based tools can improve students' engagement and participation in the classroom," in *Int. Conf. on Collaboration Technologies and Systems*, 2014, pp. 379–386.
- [22] I. Manno, F. Belgiorio, D. Malandrino, G. Palmieri, D. Pirozzi, and V. Scarano, "Introducing collaboration in single-user applications through the Centralized Control architecture," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2010, pp. 1–10.
- [23] S. Ainsworth, G. Gelmini-Hornsby, K. Threapleton, C. Crook, C. O'Malley, and M. Buda, "Anonymity in classroom voting and debating," *Learning and Instruction*, vol. 21, 2011.
- [24] R. H. Kay and A. LeSage, "Examining the Benefits and Challenges of Using Audience Response Systems: A Review of the Literature," *Comput. Educ.*, vol. 53, no. 3, pp. 819–827, 2009.
- [25] H. L. Dangel and C. X. Wang, "Student response systems in higher education: Moving beyond linear teaching and surface learning," *Journal of Educational Technology Development and Exchange*, vol. 1, no. 1, pp. 93–104, 2008.
- [26] J. L. Terrion and V. Aceti, "Perceptions of the effects of clicker technology on student learning and engagement: a study of freshmen Chemistry students," *Research in Learning Technology*, vol. 20, no. 0, Mar. 2012.
- [27] E. Koppen, G. Langie, and B. Bergervoet, "Replacement of a clicker system by a mobile device audience response system,"

19. <https://developers.google.com/java-dev-tools/codepro/doc>
20. User interfaces are excluded by the LOC metrics, because they can be of arbitrary complexity.

- in *Proceeding of the 40th SEFI annual conference 2013: Engineering Education Fast Forward*, 2013, p. 0200.
- [28] M. Schultz, "Use of an On-line Student Response System: an Analysis of Adoption and Continuance," *Journal of Innovation in Science and Mathematics Education*, vol. 21, no. 5, 2013.
- [29] D. Malandrino, I. Manno, G. Palmieri, and V. Scarano, "Face-to-face vs. Computer-mediated: Analysis of collaborative programming activities and outcomes," in *12th International Conference on Advanced Learning Technologies*, 2012, pp. 329–331.
- [30] D. Malandrino, V. Scarano, and R. Spinelli, "How Increased Awareness can Impact Attitudes and Behaviors toward Online Privacy Protection," in *2013 International Conference on Social Computing (SocialCom)*, 2013, pp. 57–62.
- [31] A. Christ and M. Feißt, "SW-Architecture for Device Independent Mobile Learning," *Architectures for Distributed and Complex M-Learning Systems: Applying Intelligent Technologies*, pp. 72–93, 2009.
- [32] F. Meawad and G. Stubbs, *MobiGlam: a framework of interoperability and adaptivity for mobile learning*, 2009, pp. 52–71.
- [33] A. Giemza, L. Bollen, M. Jansen, and H. U. Hoppe, "An architecture for supporting heterogeneous multi-device learning environments," in *WMUTE*, 2012, pp. 67–71.
- [34] E. Chryssafidou, S. Sotiriou, P. Koulouris, M. Stratakis, A. Miliarakis, M. Barajas, M. Milrad, and D. Spikol, "Developing Tools that Support Effective Mobile and Game Based Learning: The COLLAGE Platform," *Architectures for Distributed and Complex M-Learning Systems: Applying Intelligent Technologies*, pp. 1–34, 2009.
- [35] R. Grieco, D. Malandrino, and V. Scarano, "SEcS: scalable edge-computing services," in *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 2005, pp. 1709–1713.
- [36] R. Grieco, D. Malandrino, F. Mazzoni, and D. Riboni, "Context-aware provision of advanced Internet services," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2006, p. 4.
- [37] C. Canali, M. Colajanni, D. Malandrino, V. Scarano, and R. Spinelli, "A Novel Intermediary Framework for Dynamic Edge Service Composition," *Journal of Computer Science and Technology*, vol. 27, no. 2, pp. 281–297, 2012.
- [38] I. Boticki, L. H. Wong, and C.-K. Looi, "Designing Technology for Content-Independent Collaborative Mobile Learning," *IEEE Trans. on Learn. Technologies*, vol. 6, no. 1, pp. 14–24, 2013.
- [39] I. Boticki, V. Mornar, and N. Hoic-Bozic, "Context-Awareness and Distributed Events in Mobile Learning," *Architectures for Distributed and Complex M-Learning Systems: Applying Intelligent Technologies*, pp. 211–235, 2009.
- [40] U. Erra, G. Iaccarino, D. Malandrino, and V. Scarano, "Personalizable edge services for Web accessibility," *Universal Access in the Information Society*, vol. 6, no. 3, pp. 285–306, 2007.
- [41] S. D'Ambrosio, S. De Pasquale, G. Iannone, D. Malandrino, A. Negro, G. Patimo, A. Petta, V. Scarano, L. Serra, and R. Spinelli, "Mobile phone batteries draining: is Green Web Browsing the solution?" in *5th International Green Computing Conference*, ser. IGCC'14. Dallas, Texas, USA, Nov. 3-5 2014.

Delfina Malandrino received the PhD in Computer Science from the University of Salerno (Italy) in 2004. From September to December 2006 she has visited the AT&T Research Labs., New Jersey USA, working with Balachander Krishnamurthy in the field of privacy protection during Web navigation. From 2007, she is Assistant Professor in Computer Science at the "Dipartimento di Informatica" of the University of Salerno. Her research activities include: distributed systems on the World Wide Web, collaborative and learning systems, privacy and social networks, Web components benchmarking. Recently, her research interests are also devoted to social and criminal network analysis, power-aware software, green and sustainable computing, and finally, swarm intelligence for service discovery systems.

Iaria Manno received her PhD in Computer Science at the University of Salerno (Italy) in 2009. Since then, she is Post-Doctoral fellow at the Dipartimento di Informatica of the University of Salerno, cooperating in the research activity of the ISISLab laboratory. Her research activities include: Computer Supported Collaborative Work and Learning, Component Based Systems and Architecture. Recently, her research interests are also devoted to the Bioinformatics field, and she achieved a Master Degree in "Research and Innovation in Life Sciences" in February 2014.

Giuseppina Palmieri is research fellow at the ISISLab of the Dipartimento di Informatica of the University of Salerno (Italy). Her main research activities include Distributed Systems on the World Wide Web, Web-based Information Systems (WIS), Computer Supported Collaborative Learning (CSCL) and Computer Supported Cooperative Work (CSCW).

Vittorio Scarano received the PhD in Applied Mathematics and Computer Science at the University of Naples (Italy) in 1995. In 1992, he visited the University Eotvos Lorand in Budapest (Hungary). From 1992 to 1994, he visited the Department of Computer Science at the University of Massachusetts at Amherst (USA) working in the team with Prof. Arnold Rosenberg in the field of parallel algorithms and architectures. He also was instructors at Mount Holyoke College (USA) in 1994. Since 2001 he is an Associate Professor at the University of Salerno. His research focussed on Multimedia and Distributed Systems on the World Wide Web, covering several aspects from a theoretical perspective (P2P systems and architectures) to applications (intermediaries, cooperative systems and multimedia). Recently, his research interests are also devoted to information visualization, interactive virtual environments and applications of the videogame engines to non-ludic contexts.

Luca Tateo is research fellow at the Department of Communication & Psychology, University of Aalborg (Denmark). He is Marie Curie Intra-European Fellow with the project EPICS Epistemology in psychological science, the heritage of Giambattista Vico and the cultural psychology. He has been research fellow at the University of Salerno and the University of Sassari (Italy). His research interests are social psychology of education and epistemology of psychological science, revisiting the work of scholars such as Vico, Cattaneo, Wundt, Lewin, Moscovici to reflect upon the future trends of psychological research.

Daniele Casola received his bachelor Degree in Computer Science at the University of Salerno (Italy) in 2012.

Ivan Ferrante received his bachelor Degree in Computer Science at the University of Salerno (Italy) in 2012.

Francesco Foresta received his bachelor Degree in Computer Science at the University of Salerno (Italy) in 2012.