Enabling Teachers to Deploy CSCL Designs across Distributed Learning Environments

Luis P. Prieto, Juan I. Asensio-Pérez, Juan A. Muñoz-Cristóbal, Yannis A. Dimitriadis, Senior Member, IEEE, Iván M. Jorrín-Abellán, and Eduardo Gómez-Sánchez

Abstract—Transforming teachers' pedagogical ideas into ICT infrastructures ready for the enactment of such ideas is a complex process, especially in the case of computer-supported collaborative learning (CSCL). In higher education, the current trend of using external tools (e.g., “Web 2.0”) along with the institutional virtual learning environments (what some authors call distributed learning environments—DLEs) makes such transformation even more difficult. This paper proposes GLUE!-PS, an open extensible system that allows nonexperts in ICT to deploy learning designs into multiple DLEs. GLUE!-PS does so while maintaining teachers' and institutions' preferences regarding authoring tools and learning platforms, and making DLE-based practice more likely to be sustainable. This paper presents initial evaluation data of a reference implementation of GLUE!-PS, through two workshops with university teachers and the usage of the system in a real university course. According to our evaluation, academic teaching staff from different disciplines was able to deploy their own collaborative designs using GLUE!-PS under tight time constraints. The evaluation also hints at challenges for everyday practice adoption (reliability problems inherent to such a multisystem approach, the need for runtime flexibility, or conceptual barriers on the use of CSCL approaches).

Index Terms—Collaborative learning, learning design, learning management systems, virtual learning environments, deployment, distributed learning environments

1 INTRODUCTION

A PPLYING technology-enhanced learning (TEL) in authentic formal education environments is becoming increasingly complex. Teachers have the responsibility of ensuring an effective learning experience, often using any readily available technological tools (including the plethora of resources accessible through the web). In the case of practitioners trying to apply active pedagogical approaches, such as computer-supported collaborative learning (CSCL) [1], the added social component of the expected interactions among participant students makes this application even more difficult [2].

One of the technologies most commonly used to support TEL approaches (especially in higher education) are virtual learning environments1 (VLEs) such as Moodle2 or Blackboard.3 However, despite the immense popularity of certain VLEs (especially at the institutional level), the landscape of learning platforms in authentic TEL practice is highly fragmented and mutable. Adding to this fragmentation, there is a current trend of “breaking away from the VLE,” as teachers and researchers have found limitations in the variety and nature of the learning situations possible with VLE-included tools [3], [4]. Consequently, researchers have proposed other alternatives such as personal learning environments (PLEs) [3] or the use of “Web 2.0” tools (wikis, blogs, etc.) for learning [5]. Indeed, several initiatives have approached the integration of learning platforms and external tools4 including standardization efforts (e.g., IMS Basic LTI5), open source communities (e.g., Wookie Widgets6) and other research proposals (e.g., the GLUE! architecture [6]). In the rest of this paper, we will call these systems that integrate learning platforms and external tools distributed learning environments (DLEs) [7].

Despite the growing interest in DLEs, the process of going from the pedagogical ideas that a teacher may have about a learning situation, to the point where a concrete instance of a DLE is ready for the enactment of the learning activities with students, is still far from straightforward. A common approach (as it is done in many VLEs) is for the teacher to manually create the resources needed directly through each of the concerned platforms' and services' interfaces, until the desired DLE is ready. However, this

1. Sometimes also called Learning Management Systems, or LMSs.
4. Note on terminology: Throughout the paper, we will use the term “learning environment” to denote a technological platform used to structure multiple learning activities, often with the aid of multiple built-in tools (e.g., Moodle includes a chat tool, a forum tool). We will denote with the word “tool” a technological system that is suited to support one activity or kind of activity (e.g., a Moodle forum, a Google Docs shared document). We are aware that the frontier between these two concepts is blurry—for example, a wiki can be used as a learning environment to structure several activities, and as a collaborative writing tool in a certain writing activity. Thus, in the text we refer to one or the other depending mostly on the intentions of the teacher when using the technologies (i.e., structuring multiple activities versus supporting one activity).
“bricolage” process is tedious and error-prone, and may require the intervention of technical staff. Moreover, the products of this kind of approach (e.g., a Moodle course plus external tools) are not easily reusable by the teacher herself or when shared with others, thus impacting the sustainability of this kind of practice. The fragmentation and mutability of the current learning platform panorama only make this sustainability problem of DLEs more poignant.

One possible way of dealing with this problem would be for teachers to express their pedagogical ideas using a learning design approach [9], in the sense of making the pedagogical principles and decisions behind the learning situation explicit. Having a learning design of the activities is known to be useful for reflection on teachers’ own practice [10], but also produces artifacts that can be easily shared and reused [11]. The learning design research field has produced multiple approaches and tools to aid in this labor of making the pedagogy behind learning situations explicit [12]. However, the learning designs produced by this extensive tooling (which is also highly fragmented, despite standardization efforts such as IMS-LD [13]) cannot be transformed easily into concrete DLE instances, and especially not if we are to respect teachers’ and institutions’ choices of learning platforms and learning design authoring tools [14]. This “deployment gap” [15] between the learning design tools and the different platforms used in the enactment of those designs (be them VLEs, DLEs, or others) is a major barrier to the application of a learning design approach in practice using these platforms, and might be related to the general lack of adoption of formal learning design approaches among teachers.

This paper presents GLUE!-PS, a software that enables nonexperts in ICT (e.g., most university teachers) to overcome this deployment gap, allowing them to semiautomatically deploy learning designs (expressed using the authoring tool of their choice) into different flavors of DLE (e.g., one that includes their institution’s choice of learning platform). A prototype of GLUE!-PS has been developed and evaluated iteratively, aiming especially at blended CSCL scenarios. We also present here an initial evaluation of the deployment capabilities of GLUE!-PS with university teachers from different disciplines, through two teacher workshops and a study in a real master’s-level course.

The paper is structured as follows: Section 2 delves into the problem of deploying pedagogical designs across DLEs; Section 3 describes the GLUE!-PS system, including its architecture and underlying data model; Section 4 summarizes the main features of the GLUE!-PS prototype that has been implemented, and the results of evaluating such prototype through three studies are presented in Section 5. Finally, Sections 6 and 7 gather the main issues of discussion and concluding remarks.

---

### 2 Learning Designs and DLEs: The “Deployment Gap”

The design of learning situations (understood widely as the preparation of instructional materials, activities, information resources and/or evaluation) has been present in education, in one form or another, for ages. This preparation is even more critical in the case of activities that involve technology or active pedagogies such as CSCL. Although there exist as many procedures as there are practitioners and learning platforms, the translation of pedagogical ideas into the technological environment that is meant to support the learning activities (what we may call “deploying” those ideas) falls into two broad categories. The first one (which [8] denominates a bricolage approach) involves the teacher manipulating directly the final environment (e.g., the Moodle course) until the desired result is achieved. The second one involves making the pedagogical ideas explicit through some kind of learning design, and then finding a way of automatically deploying that design into the target environment (or environments).

The bricolage approach has the advantage of requiring less effort when deploying simpler scenarios or making runtime adaptations to them. This approach can also enable a finer tuning of the deployment to take full advantage of the target environment’s affordances. However, it is also true that deploying more complex situations (such as many CSCL ones, which involve changing groupings of students who construct and reuse artifacts along a sequence of activities) can be tedious and error-prone to deploy in this fashion (plus, it may require advanced technical knowledge about the target environment). Also, the technological support resulting from the bricolage process is not so easily reusable by the teacher herself or by other teachers (especially if trying to apply the course to a different target environment).

Learning design approaches, on the other hand, provide several methods and tools to facilitate the labor of expressing pedagogically sound learning activities [16]. This kind of approach presents several advantages for teachers, such as making the pedagogical decisions explicit (thus making it useful as a reflection tool), or their ability to easily share and reuse such designs among practitioners [10]. If these designs can be expressed in a computer-interpretable format (such as, e.g., the IMS-LD specification [13]), then deploying them into the target environment could be automated (including, hypothetically, their reuse and deployment across different kinds of environments). This kind of automated deployment, along with the sharing and reusing of learning designs, would make TEL practice more easily sustainable. The obvious disadvantage of such an approach is the effort needed to explicit this learning design (although this can be mitigated by using different kinds of learning design authoring tools available [12]). However, as we will see, this automated deployment is not so trivial in the currently fragmented panorama of learning platforms.

If we look at the present-day technological landscape of formal education (especially in universities, but also in the workplace and in other contexts), one of the most common ways of supporting learning activities with technology is by...
using general-purpose web platforms (be them VLEs, PLEs, Web 2.0 tools or combinations of them such as the aforementioned DLEs). In the concrete case of teachers trying to use a DLE that combines a learning platform (e.g., Moodle) and one or more external tools (e.g., a wiki, or a shared Google document), the deployment of the (implicit or explicit) pedagogical ideas is rather complex and unsustainable in the long run. Taking the bricolage approach would require teachers to go through the different platforms and services involved, creating the different resources needed and linking them together to form the concrete DLE to be used in their particular learning situation. Again, we can see how this approach can be tedious, error-prone, and difficult to share and reuse.

Learning design approaches could, thus, be used to foster the sharing and reuse of teachers' pedagogical ideas, if we could find a way of automating the deployment of learning designs into the variety of target DLEs used by teachers. Unfortunately, current learning design tooling is also highly fragmented, and it is in general not interoperable with widely used learning platforms (such as VLEs or PLEs). Consequently, the deployment of learning designs into existing flavors of DLE is not supported either. Attempts to standardize this deployment through the IMS-LD specification [13] failed to gain acceptance from the major platform vendors, and there exist many learning design tools that do not support this standard either (e.g., LDSE,9 CompendiumLD,10 or ScenEdit11). Indeed, this “deployment gap” [15] might be one of the reasons why learning design tools and approaches have not yet gained wide adoption among the teacher community (because such gap inhibits the reusability of the designs in practice) [17]. The main exception to this lack of support are learning environments such as LAMS12 [18], which integrate a learning design tool and an enactment environment. However, using this option requires the teacher to use both the integrated design and enactment environments provided by LAMS (which may not be realistic or convenient for many teachers, due to their preferences or institutional platform choices).

To further illustrate this current situation and the potential benefits of filling out the deployment gap, let us consider the following example: Bob (a teacher) wants to perform a simple collaborative activity (following the Pyramid pattern [19], see Fig. 1) in a blended learning master’s-level course of his. The students (18 of them) are to learn about some of the main bibliographic sources of a research field (represented by an article) through a series of collaborative activities. First, in dyads, the students construct a wiki page with literature references on a field/topic, using the sample article as a seed for their search. Then, these dyads are combined to form six-people groups, which collaboratively build a graph of bibliographic references, depicting the relationships among authors and papers, which is then used to build a document stating the 10 most relevant sources, and the most typical structure followed by papers of the field. Finally, a whole-class debate is staged to compare and discuss the documents produced by the different groups, using a shared whiteboard to trace the debate and its conclusions. These activities follow a blended [20] format: they are initiated in face-to-face sessions, but they are completed in an online fashion, as homework. Since the rest of Bob’s course is centralized through the institutional VLE (a Moodle installation), Bob wants to use that VLE as the main point of entry to the learning activities, albeit he wants to use external Web 2.0 tools (in this case, a wiki and Google Docs shared applications, see Fig. 1), because they offer collaboration features not available through the institutional VLE.

Also, to understand the sustainability issues of the scenario, let us consider that Bob wants to share this idea with another teacher, Sarah, who uses a wiki (e.g., based on the MediaWiki engine) as a sort of VLE to provide access and manage her courses. Let us see what are Bob and Sarah’s options for deploying and reusing this nontrivial scenario:

- Pure bricolage. Conceptually, the most straightforward way of implementing the infrastructure for this scenario is for Bob to create the 16 resources involved (nine wiki pages for the dyads, three shared documents and three shared drawings for the six-people groups, and another shared drawing for the whole-class debate) manually. Thus, Bob would access each of the different services to create the involved tools, and then link them all together from the course at the institutional VLE. Bob will also have to create the different groupings for the dyads and the six-people groups, so that each of the resources can be associated with the corresponding group. Obviously, this kind of operation requires a considerable effort. Moreover, even if Bob shares with Sarah a backup of the Moodle course, she can only use it as inspiration, and she will have to recreate all groupings and external resources for her particular group of students, and then link the groupings and resources using her own target environment (the wiki).
• DLE-aided bricolage. Bob could also use one of the existing approaches to implementing DLEs (Basic-LTI, Wookie Widgets, GLUE!, etc.) to aid him in the implementation of his desired DLE. Depending on the approach chosen, Bob would have to perform different amounts of work. For example, if using Basic LTI (which is supported by a range of widely used VLE software\(^\text{14}\)), Bob would still have to create the 16 external resources, and the different groups in the VLE. Alternatively, if using GLUE! (which supports Bob’s VLE, Moodle), Bob would have to create the different groupings in the VLE, but the creation of multiple instances of external tools can be done directly through Moodle’s user interface (one of GLUE!’s main features). Even if this option requires less effort to deploy the designs, the reusability for Sarah would still be low because both teachers use different learning environments, and the concrete resources for Sarah’s particular situation (groups of students, wiki pages, shared documents for each group, etc.) would have to be recreated again.

• IMS-LD tooling. If Bob explicits his pedagogical ideas using an IMS-LD-compliant authoring tool, the deployment of such ideas into a VLE could be automated. However, the 16 external resources would have to be created manually and linked somehow in the target learning environment. The advantage of this approach would be that the IMS-LD design could be easily reused by Bob and shared with Sarah (although the external resources would have to be recreated again by Sarah). Nevertheless, neither Bob’s nor Sarah’s environments (Moodle and MediaWiki) support the IMS-LD standard and, thus, this option is not feasible for them (as it is not for a majority of teachers given the lack of IMS-LD support among the most popular learning environments).

• Integrated authoring and execution (e.g., LAMS). In a similar vein to the previous option, environments such as LAMS allow for the specification of learning designs and their automatic deployment. The external resources in this case would have to be created by Bob and linked somehow from LAMS. This design would be easily shared with Sarah as long as she learns how to use LAMS. Again, the fact that neither Bob nor Sarah are currently using this system makes this option unfeasible (or highly inconvenient) for them.

• LAMS-based, aided DLE (e.g., GLUE!). Yet another option is available, which would be to combine the learning design capabilities of LAMS with the ability to easily implement DLEs provided by DLE initiatives such as GLUE!. In this case, Bob could design the learning scenario, deploy it automatically in LAMS and use GLUE!’s features to aid him in creating the needed external resources. These designs could also be shared with Sarah with relative ease (as long as she also uses LAMS). However, again we find that using this approach would force both Bob and Sarah to migrate their courses to a (GLUE!-enhanced) LAMS platform. Even if they did so, they would be constrained by LAMS’s design authoring capabilities, in which formalizing a non-trivial CSCL scenario such as the one presented here is not an easy task.

We can readily see that transforming teachers’ pedagogical ideas into concrete instances of DLEs is not supported in a way that is easily sustainable and, at the same time, compliant with current choices of authoring tool and learning platform. Thus, we may wonder whether it is possible to devise an approach that can leverage learning design’s ability to automate deployment and easily share and reuse its products, at the same time respecting teacher and institution choices of technological support currently in use.

3 GLUE!-PS

In this paper, we propose the Group Learning Unified Environment—Pedagogical Scripting system (GLUE!-PS\(^\text{15}\)) as a solution to the aforementioned problems. The main goal of the system is to support teachers in the deployment of their pedagogical ideas over distributed learning environments, in a way that is sustainable within the constraints of authentic settings. This emphasis on the feasibility and potential sustainability of the approach for teachers (but also within wider TEL context of practice, which includes stakeholders like institutions or technology developers) motivates the main design principles of the GLUE!-PS system:

1. to be usable by nonexperts in ICT (e.g., teachers);
2. to be agnostic with respect to the authoring tool and DLE used (given the fragmentation in both kinds of technologies);
3. to favor interoperability between multiple authoring tools and multiple DLEs (to enable sharing and reuse across different personal and institutional choices);
4. to do all of the above without needing to modify the source code of existing authoring tools and DLEs (so that GLUE!-PS can work easily on top of existing infrastructures chosen by teachers and institutions);
5. to be open and extensible (so that it can address authoring tools and DLEs that may emerge in the future).

It would also be desirable that the system does so while minimizing the development effort (so that technology developers can easily adopt the proposal).

3.1 Architecture

Fig. 2 depicts the general architecture of the GLUE!-PS system that we propose according to those principles, and


\(^{15}\) Not to be mistaken for the aforementioned GLUE! architecture for the integration of external tools into VLEs [6]. Both architectures have similar names for historical reasons (i.e., both were proposed by the same research group, and were part of a larger architectural approach to educational systems). However, currently both proposals are independent of each other, even if they are compatible (as exemplified in [21]).
the main actors involved in a typical deployment case. In GLUE!-PS, the deployment of learning designs—expressed in different computer-interpretable formats—is achieved through two sets of adapters: first, a set of LD adapters translate the original learning design into a common set of design concepts (see next section on the GLUE!-PS data model), which are uploaded into a central infrastructure (the GLUE!-PS Manager). Here, the translated design can be modified or particularized for the teacher’s specific situation using the GLUE!-PS graphical user interface (GUI). This particularization is needed in any learning design approach because design authoring tools normally do not take into account the number of groups, specific participants or the distribution of the artifacts among those groups (e.g., should each group use a different wiki page, or should all groups use the same page?). Finally, the particularized design is deployed onto one of the multiple flavors of DLE available (e.g., based on Basic LTI, based on GLUE!, etc.), which will probably include the institutional (or teacher’s) choice of learning platform. This final deployment step is performed by a second set of adapters, the DLE adapters, which translate the particularized design from the common data format into the particular concepts of the user-selected DLE. In fact, these DLE adapters contain two different parts: one to deal with the specific DLE integration issues (including the creation of the needed external resources, and their integration into the VLE—e.g., using Basic LTI, GLUE!, or other means), and another module to deal with VLE-specific deployment tasks (converting the notions of the particularized design to VLE-specific terms, and then creating and linking such VLE structures into the selected VLE installation). Please refer to this paper’s additional materials for concrete examples of the learning design’s different representations at each stage in the deployment process.

This architecture enables a highly decoupled approach to the deployment of learning designs, allowing for the kind of multi-LD, multi-DLE interoperability needed by practitioners, given the fragmentation of current technological support in the design tool and learning platform areas. The resulting system is open and extensible (by developing new LD and DLE adapters), and supports current institutional and teacher choices of both learning design authoring tools and learning platforms, without modifying their code. Thus, using GLUE!-PS teachers would be able to use, for example, the WebCollage learning design authoring tool to define her pedagogical ideas, and deploy them to a BasicLTI-based DLE implemented with Moodle tools and Google Docs; the same system would also enable teachers to define activities using the Pedagogical Pattern Collector authoring tool, and deploying them to a GLUE!-based DLE implemented with the MediaWiki platform and Wookie widgets.

Indeed, multiple implementations of an LD or DLE adapter may be developed (e.g., by institutions themselves, or by third parties), to cater to the specific needs of different teachers in terms of how the learning design concepts are translated to and from the GLUE!-PS common data format during deployment. Moreover, this is achieved at a low average development effort, because the common adaptation functionalities are implemented in the GLUE!-PS manager, simplifying the adapters’ source code.

### 3.2 Data Model: The GLUE!-PS lingua franca

From the architecture described above, it is apparent that a crucial aspect of the GLUE!-PS proposal is the aforementioned common data format that the GLUE!-PS Manager uses as an intermediate language (or “lingua franca”) in the translations between the original learning design format and the final DLE implementation. This lingua franca needs to be expressive enough so that not too much information is lost in the two successive translations (i.e., the result still represents the teacher’s original pedagogical ideas). However, it also has to be simple enough so that the cost of developing new adapters is kept under control.

In [15], we proposed a data model for this kind of lingua franca, which is graphically represented in Fig. 3. This model was obtained by analyzing the underlying data models of various widely used learning design languages and specifications, as well as the underlying models of multiple widely used learning platforms. The analysis compared both kinds of models with common features of CSCL designs (extracted from conceptual frameworks such as the one proposed in [22]). This comparative analysis was weighed considering the relative popularity of each model (e.g., Moodle or IMS-LD models were given more importance than other, more exotic ones). The resulting data model has two main parts: one related to the decontextualized features of the learning design (including notions like activity, resource, or role, which are common in learning design languages), and another part that refers to the contextualized features of the deployment in the teacher’s particular situation (with concepts like group, participant,
tool instance, or learning environment). It should be noted that this data model is not intended to be yet another learning design language—rather, it aims at representing the most common features of learning designs that are supported in current distributed learning environments.

As it can be derived from the architecture depicted in Section 3.1, this intermediate data model is the main limiting factor that determines which designs can be expressed and deployed using GLUE!-PS (and how faithfully). This data model incorporates features (e.g., the notion of groups of participants) that make it easy to support collaborative learning activities [15]. However, given its simple way of implementing, for example, learning activity structures (through a tree-like structure), scenarios that involve complex sequencing (e.g., loops, conditions) would be difficult to express faithfully in the lingua franca. The discussion and validation that this kind of data model is expressive enough to provide acceptable translations of teachers’ original designs into the target learning environments exceeds the scope of this paper, although the issue has been explored further in [15] and [23]. These analyses show that, even if there is a slight loss of information in general, in most cases the resulting notions in the translated deployments support the essential features of the original design, as much as the target learning environments would.

4 GLUE!-PS Reference Implementation

Following the architecture and data model proposed in the previous section, we have developed a prototype of the GLUE!-PS system16 to evaluate, not only its ability to deploy learning designs from multiple authoring tools over multiple DLEs, but also to assess the usefulness that such a system could provide to teachers in their authentic practice. This prototype takes the form of a web service programmed in Java, with web technologies (such as HTML and Ajax) implementing the user interface. Two LD adapters have been implemented so far, one for IMS-LD (Level A, see [13]) as it is the most often-cited learning design language, and another one for London Knowledge Lab’s Pedagogical Pattern Collector [24], a simple learning design tool that uses pedagogical patterns to guide teachers in the process of designing.

In the DLE side, the current prototype supports DLEs implemented using the GLUE! architecture17 (e.g., Moodle v.1.9), a static deployment is produced in real time; in the case of platforms that do not allow such remote operation easily (such as Moodle v.1.9), a static deployment is produced (e.g., in the form of a Moodle course backup) and deployed by the teacher in a deferred way (i.e., the teacher uses Moodle’s course restore function to deploy the configured activities including the external tools). Fig. 4 shows the main screen of the GLUE!-PS GUI, which includes a graphical representation of the (particularized) learning design, with its activities, groups, and resources. The GUI also includes auxiliary windows on the right side, for additional operations regarding the resources to be used, as well as the participants and group composition.

5 Evaluating the Deployment of CSCL Designs with GLUE!-PS: Three Studies

First analytical evaluations of GLUE!-PS’s capabilities to deploy learning designs from multiple languages to multiple VLEs had provided encouraging results (see [15]). However, to offer answers to our main research question ("is GLUE!-PS useful for teachers to deploy their learning designs over DLEs?"), a more empirical validation was needed. In this section, we present three studies used to evaluate GLUE!-PS: two workshops aimed mainly at university teachers (to reach a considerable number and a wide variety of participants), and the usage of the system in a part of a real master's-level course. In these three studies, the GLUE!-PS prototype was used by teachers to deploy collaborative learning designs into distributed learning environments.

5.1 Context and Methodology of the Studies

Following the recommendations of multiple researchers [27], [28] and evaluation frameworks in the field of CSCL (e.g., CSCL-EREM [29]), we have used a mixed methods approach [30] to perform the evaluation studies, similar to

---

the one portrayed in [31]. Furthermore, and given our focus on the usefulness of the system in authentic settings, our evaluations were performed using naturalistic [32], observational methods (as opposed to, e.g., controlled laboratory experiments, following Zelkowitz and Wallace’s [33] classification of data collection methods) during “situated interventions.” However, due to the exploratory nature of the studies and the need to identify emergent features, we have prioritized the qualitative perspective in the data gathering and analysis of our data, as well as in its presentation below.

Regarding the analysis of the evidence gathered, our evaluations try to explore the evaluative tension or issue: “Is GLUE!-PS useful for the participant university teachers in deploying their learning designs over DLEs?” (I1). This issue in turn can be explored through multiple informative questions grouped into two topics (T1, regarding GLUE!-PS’s deploying ability, and T2, related to its use in authentic practice). This conceptual organization of the data from the evaluation is adapted from Huberman and Miles’s [34] anticipated data reduction procedure.

The main contextual features of the three studies are summarized in Table 1. The first workshop (TW1) was part of a joint project meeting of two Spanish research projects, in which several outputs of one of the projects (GLUE!-PS among them) were to be tested and discussed by members of the other project and external advisors. Ten expert teachers and teacher-researchers from different Spanish and foreign institutions agreed to participate in the session. The participants were provided with an example collaborative scenario (very similar to the one depicted in Section 2) and a detailed walk-through of the steps needed to deploy that concrete scenario using GLUE!-PS. Afterwards, participants would engage in a debate on the usefulness of the proposed tools and their applicability to real educational practice. This session was also audio and video recorded, and observed by two researchers. Furthermore, participants answered a (quantitative and qualitative) questionnaire after the session, to assess the tool.

The GLUE!-PS system was also used in an authentic deployment situation (AD1), in a real master’s-level course of the Secondary Education Teacher degree at the University of Valladolid. In this intervention, a teacher with computer science background and prior knowledge about CSCL techniques designed, deployed, and enacted a three-week section of a course on researching and discussing “learning and teaching approaches” (e.g., collaborative learning, project-based learning, etc.). Fourteen students, mostly with engineering and computer science backgrounds, attended that section of the course, which followed a blended learning format spanning three face-to-face sessions and several hours of online distance work. The teacher designed a collaborative activity using the well-known Jigsaw collaborative strategy [35], with additional peer-reviewing activities in the last phase of the strategy. In the weeks that preceded the enactment of the activities, the researcher team held five (1-2-hour) codesign and codeployment sessions in which the teacher used the WebCollage authoring tool and GLUE!-PS to deploy such design into a GLUE!-based DLE composed of MediaWiki (because that was the main learning environment that she used for teaching her other courses) and several external tools. After the enactment of the activities, a semistructured qualitative interview was conducted to gather the teacher’s impressions on using the GLUE!-PS system.

A third study (TW2) took place in the following academic year, through a 12-hour teacher professional development (PD) workshop at the University of Valladolid, on the topic of implementing collaborative activities using ICTs. Twenty-four teachers from diverse backgrounds (from engineering to education, medicine or law) participated in the workshop, which was portrayed as a continuation of a previous PD course on designing collaborative activities with ICT (which focused on the conceptual aspect of designing). In this workshop, teachers first followed a walk-through to deploy a collaborative learning design (again, similar to the one in Section 2), from the WebCollage authoring tool into a
Moodle-based DLE using GLUE!-PS. Afterwards, participants were asked to design (also using WebCollage) a collaborative scenario for their own real teaching practice, deploying it into a Moodle- or MediaWiki-based DLE. Toward the end of the workshop, participants engaged in four parallel focus groups to discuss the applicability and challenges of using the presented systems. The workshop sessions (including the focus groups) were audio recorded, and two researchers took observation notes. Participants also answered three questionnaires, a preworkshop one for profiling purposes, and two postworkshop ones to assess the systems presented and to evaluate the workshop as a PD action.

Finally, it is also interesting to note that the first two studies were conducted using an early prototype of the GLUE!-PS system (which we will call “Prototype A”), which had several user interface and reliability limitations. Indeed, a few of these problems arose during the TW1 workshop, preventing some of the participants to complete the deployments. The third study used the current prototype (i.e., “Prototype B”), which is more reliable and user friendly. Even though the reliability problems hampered the gathering of data, especially from the first study, considered altogether the three of them provide considerable evidence of the evaluation findings that we describe below.

5.2 Evaluation Results
To better understand how the evidence from the different studies supports our findings, we present below selected pieces of evidence from across the three situated studies, grouped around the two topics that explore our issue (T1 and T2, see above).

5.2.1 GLUE!-PS Deployment Ability (Topic T1)
In the first workshop (TW1), the session was riddled with technical failures not directly related to GLUE!-PS (network outages, failure of external tool services, and the Moodle server itself, or bugs in the GLUE! architecture prototype). Several reorganizations of the provided worksheet were made on-the-fly to address these problems and, eventually, many of the participants were able to finalize the deployment using GLUE!-PS (seven out of the nine participants that answered to the questionnaire [TW1-Q]). Moreover, when asked whether the deployed courses supported adequately the realization of the proposed scenario, eight out of nine respondents answered positively (and the other one did not answer negatively either) [TW1-Q]. Thus, evidence shows that GLUE!-PS could deploy CSCL designs into a DLE under restrictive time constraints (less than 2 hours were dedicated to the authoring and deployment of the design).

In the authentic deployment study (AD1), GLUE!-PS showed that an ICT-expert teacher with no prior knowledge of the system was able to deploy a collaborative learning design of medium-high complexity (based on a combination of collaborative patterns) [AD1-O]. Moreover, GLUE!-PS did so while complying with what the teacher considered the design’s essential qualities (“to the question: does the deployed wiki reflect your idea of the design?| the positive side is the structuring of activities in pages [the activity tree depicted as links to activity wiki pages], is nice, and helps [...] I had to [manually] edit the design’s goals [...] but that’s secondary” [AD1-I]). Given that MediaWiki is not able to enforce strong group-based access control on the resources, it seems that the possibility, for example, of students cheating and copying one another, was not considered a relevant risk by the teacher, while the open access to the learning community-generated artifacts was seen as important. Also, the use of GLUE!-PS in a real learning situation also prompted the teacher to propose a few extensions to the system, regarding information that might be useful while doing the particularization of the design (which resulted in minor extensions to the GLUE!-PS LF model). For example, adding a description to the particular activity that each group performs in parallel with other groups (the concept of “instanced activity” in Section 3.2) was deemed useful, so as to provide group-specific activity instructions [AD1-O]; the teacher also suggested adding timing information for the activities, for classroom management purposes (“when asked about the deployed design’s shortcomings I had to add the deadlines for the contributions [by hand]” [AD1-I]).

In the second workshop (TW2), all the participants (24 academic teachers from various disciplines) were able to deploy learning designs based on the example scenario proposed. For example, the observers noted that “group 8 sees their design deployed in Moodle [...] All deployments have worked!” [TW2-O]. Also, a great majority of the participants (18 out of 21 questionnaire respondents—85.7 percent) declared that they were able to deploy the learning designs they had created for their own courses [TW2-Q2], to a Moodle- or MediaWiki-based DLE. Moreover, those who did not finish the deployment of their designs provided reasons unrelated to GLUE!-PS (e.g., related to the authoring tool or teachers’ knowledge about CSCL: “the real case I wanted to work on does not match the pyramid structure and, since I do not know other techniques, I had trouble from the start” [TW2-Q2]). Regarding the quality of the resulting deployment in the DLE, teachers almost completely agreed that having all the DLE resources centralized under the learning platform would facilitate the enactment of activities (avg = 7.28, std = 0.84 in a 1-8 Likert scale [TW2-Q2]). Also, respondent teachers agreed strongly that the deployment of their design reflected their original ideas (avg = 6.62, std = 1.60 in a 1-8 Likert scale [TW2-Q2]).

Overall, a total of 21 informal designs (in the sense of not computer interpretable) were used for deployment across the three studies. From these designs, 38 computer-interpretable designs were specified (by teachers themselves) and deployed to a DLE (33 to a Moodle-based DLE, five to a MediaWiki-based one). The complexity of the designs was variable, albeit nontrivial (sequences of 3-10 activities, with an average of 15 different resources and 11 different groups per unique design). All designs were specified in IMS-LD using the WebCollage authoring tool for CSCL.

5.2.2 Usage in Authentic Settings (Topic T2)
In the first workshop (TW1), after participants deployed the example learning designs, a majority of the participants stated that they would use GLUE!-PS in real teaching
practice (seven out of nine respondents [TW1-Q]), especially in the case of large collaborative designs, where the efficiency gains might be more apparent: “In courses where you have lots of groups it could be an easy way to perform the orchestration,” or “In order to avoid great amounts of useless time managing groups in Moodle, and the configuration of the external tools” [TW1-Q]. However, the participants also considered that teachers would need contextualized help and examples to use such a system (“if they do not have models [examples] of use in reality, teachers are not going to use it” [TW1-R]), as well as improved usability and flexibility (“the automation of the flow beforehand is nice, but it is not so flexible [...] I want to change [the design], to move things [i.e., drop-and-drag]” [TW1-R]).

In the authentic deployment study (AD1) we had a first proof of the power and limitations of GLUE!-PS in an authentic course using DLEs. In general, the enactment was successful from the point of view of the teacher, even if she registered several complaints by students on the choice of learning tools to complete the activities (because they were not used to wikis, or to some of the proposed tools, e.g., Dabbleboard19): “[when asked about problems that students had during the enactment] some complained about the aesthetics of the wiki [...] there were problems about knowing where to put their submissions [...] they did not like Dabbleboard at all” [AD1-I]. Aside from these complaints not directly related to GLUE!-PS, probably the most striking finding of this study was the iterative nature of the design and deployment process that the teacher followed when using the LD authoring tool and GLUE!-PS, which took three iterations over five face-to-face sessions [AD1-O]. The fact that the deployment is automated makes it very important to specify the LD formalization in such a way that the final presentation will fulfill the teacher’s expectations. Thus, the teacher had to go back and forth a number of times between the different stages of deployment until satisfied by the resulting DLE (“[when asked about the design and deployment process] it was iterative, a bit of a mix [...] the design, afterwards, has been influenced by the deployment” [AD1-I]; “[when asked about the iterativeness of the design/deployment process] Activity descriptions also required a couple of cycles because, after seeing how the wiki looked finally, I modified them having that in mind” [AD1-I]). This real practice usage also highlighted other teacher needs when particularizing and deploying learning designs across DLEs, for example, to fix shortcomings in the formalization that the LD tool does of the design ideas (e.g., in WebCollage, the way it models the jigsaw pattern in IMS-LD format). Also, the need for alternative LE adapter implementations (which decide the way the particularized design is presented in the learning platform) appeared prominently, and in fact different variations of the MediaWiki LE adapter were implemented during the study, to cater for the teacher’s needs [AD1-O]. This evidence also highlights the power of the GLUE!-PS adapter architecture, which has the potential to address the needs of different teachers or different kinds of designs (even when working with the same target learning platform).

In the second workshop (TW2), performed with the improved GLUE!-PS prototype, university teachers from multiple disciplines valued highly the usability of the GLUE!-PS system (avg = 7, std = 1 in a 1-8 Likert scale [TW2-Q2]), even compared to, for example, Moodle scores (avg = 6.48, std = 1.36). Teachers asserted they would likely use the proposed tools in their everyday practice in the immediate future (avg = 6.19, std = 1.63 in a 1-8 Likert scale [TW2-Q2]), and that they were likely to use those concrete deployed designs in their practice in the immediate future (avg = 6, std = 1.64 in a 1-8 Likert scale [TW2-Q2]). Regarding obstacles for its use in authentic practice, teachers mentioned several aspects related to general beliefs about collaborative learning or the use of ICT for education (“we have also considered whether these collaborative activities may subtract from the [content] learning,” or “another problem can be the attitude of students [...] there are some students that want to work alone,” or “some of us would not use it because they think [using ICTs] is complex and not necessary, versus other simpler ways of doing collaborative work” [TW2-FG]), or their lack of confidence (“It is the first time I approach the application, I need more time to practice” [TW2-Q2]). Others highlighted the need for its integration into larger courses at the institutional platform (“there is a need of integrating this kind of scenarios in a larger [Moodle] course” [TW2-FG]), or the limits of the design authoring tools (“it is difficult to translate his design ideas to the tool, [...] he probably would have to change his activity ideas” [TW2-O]). There were also remarks about the reliability of the prototype, and the need for online help or other forms of support (e.g., “there has to be more technical support [...] manuals, [online] help, help from somebody [...] to have a guide [...] the provided worksheet is OK for the task at hand, but in other cases you might be a bit lost” [TW2-FG]).

5.2.3 Evaluation Conclusions

Overall, the evidence gathered from the three studies consistently point to the notion that the GLUE!-PS system is able to deploy CSCL designs to multiple flavors of DLEs. Moreover, university teachers seemed to be generally satisfied with the result of deploying their designs to the DLEs. Along with the evidence provided in [23], our data suggest that, even if the loss of design information during the GLUE!-PS deployment process cannot be avoided, such a loss might be acceptable for the purposes of enacting the design with a DLE.

Furthermore, many teachers asserted that they were likely to use the system in their real practice. They seemed to value especially the efficiency factor of deploying learning designs that are complex in terms of group structure and resources. Those participants who were not eager to use the system in practice often provided reasons not related to the GLUE!-PS system itself, but rather about their skepticism on the use of collaborative learning or ICTs in their classes. In this regard, it is worth noting that, even if GLUE!-PS can be used for noncollaborative pedagogies, the workshops were advertised as being about the implementation of CSCL techniques in teaching practice, thus giving raise to this kind of remarks from participants. It is also noteworthy how the usage of GLUE!-PS in a real course prompted an iterative deployment process to fine-tune the (formalized)

19. The tool has now been closed, but its blog can still be accessed at http://dabbleboard.wordpress.com (accessed 29 May 2013).
learning design and its final DLE deployment, something that was not so readily apparent in the teacher workshops.

Aside from the technical problems experienced as the prototype evolved, the evaluations hinted at some of the limitations of the proposed approach in terms of reliability, given its reliance on a number of external elements (authoring tools, external learning tools, target learning platforms, DLE implementation architectures). Some of these limitations are inherent to the concept of DLE, while others stem from the multistep nature of the deployment process with GLUE!-PS.

The iterative development process that emerged in the authentic deployment study, and certain remarks from participants in the workshops, point toward another important unsolved challenge to the authentic practice use of the system: the need for further flexibility (e.g., in terms of runtime changes to the DLE). The GLUE!-PS architecture and data model allow making changes in the particularized design (e.g., using the GLUE!-PS GUI), and redeploying them through the DLE adapter in runtime, to obtain a “tweaked” DLE (e.g., changing resources to be used, group composition, etc.). However, in terms of implementation, this kind of functionality has only been available in the second prototype of GLUE!-PS (“Prototype B,” see Table 1), and has not been evaluated extensively. Further empirical evaluations are needed to ascertain whether such flexibility support is adequate for the purposes of authentic teaching practice.

In any case, these conclusions should not be taken as statistically significant evidence, but rather as “situated evidence” about the use of the system by 35 teachers from multiple disciplines within authentic professional development actions, project meetings, and real courses, with the limitations and advantages of such interpretive methodologies (e.g., lack of generalizability, stronger external validity, etc.). The data from the evaluation happenings have varying degrees of credibility and authenticity, and present a number of clear biases and limitations. For example, the designs that were deployed throughout the workshops were sometimes conceptualized and designed by teachers (in the second half of TW2), while in other cases they were ready-made designs based on fictitious but realistic situations (in TW1 and the first half of TW2—which were in fact based on teacher designs from prior PD workshops). Also, the designs vary in their authenticity, because sometimes they were made in response to a fictitious learning situation (in TW1), while others were made for each participant’s own practice (in the second half of TW2). Another limitation is the fact that all the designs deployed during the studies were based on a limited set of collaborative patterns (the collaborative learning flow patterns proposed in [19]), and were specified using only one authoring tool, WebCollage.20 Second, the variety of target DLEs used during the workshop was limited to GLUE!-enhanced Moodle- and MediaWiki-based DLEs. This was motivated mainly by the fact that Moodle is the institutional VLE at the university, where the workshops took place. These biases should be addressed in future evaluation studies.

6 Discussion

According to our evaluation, GLUE!-PS has shown the potential to deploy a variety of learning designs into distributed learning environments, while complying with authentic educational setting constraints (such as being used by nonexperts from different disciplines, or reusing existing learning platform infrastructures). Concerning the discussion from Section 2 on the options available for teachers to implement their pedagogical ideas in DLEs, and to share and reuse them, we find that GLUE!-PS requires a certain effort of up-front formalization of the learning design prior to the deployment itself. This effort might make the deployment of the pedagogical ideas more costly than the bricolage-like alternatives in the case of simpler designs (in terms of groupings and resources). For more complex scenarios, as some of the participant teachers noticed, GLUE!-PS would be more efficient because it helps in automating the creation of the different resources and groups involved. The major advantage of the GLUE!-PS system, however, is that it enables teachers to share and reuse their pedagogical ideas (as it happens with the other learning design approaches). In this case, sharing and reuse can happen at two different levels: at the authoring tool level (if different teachers use the same or compatible authoring tools), or at the GLUE!-PS level (across incompatible authoring tools for which a GLUE!-PS adapter exists). Once shared, such reused designs can be deployed in all supported DLE options. Furthermore, when compared with other learning design approaches to deployment, GLUE!-PS supports a wider variety of learning platforms for enactment, especially DLEs that include not only a traditional learning platform such as Moodle, but also external (e.g., Web 2.0) tools. However, GLUE!-PS has the disadvantage of losing information during the translation process (e.g., contextual descriptions, timing and session information, etc.)—even if our evaluations so far (by showing teachers the final result of deploying their designs) regard such losses as acceptable for practitioners [23].

We could also compare GLUE!-PS with other recent proposals oriented toward the implementation of learning design approaches using widely available learning platforms. CADMOS21 [37] is an IMS-LD compliant authoring tool that can export its designs also in the form of Moodle courses. This makes learning designs created with this tool easily shareable and reusable, as long as the teacher uses CADMOS or another IMS-LD tool for designing, and a plain Moodle VLE for the enactment of the activities. This ad hoc mapping between the IMS-LD design and the Moodle activities could potentially be more accurate than the one done with GLUE!-PS. However, GLUE!-PS supports deploying and sharing among teachers using a wider variety of authoring tools and learning platforms (and, in fact, CADMOS designs could be deployed to other VLEs and

20. In [15], we demonstrate the feasibility of using the LDL language [36] with GLUE!-PS; designs expressed with LKI’s Pedagogical Pattern Collector have also been deployed using the current GLUE!-PS prototype. However, these tools and languages were not used in the workshops, given the particular needs and constraints of the professional development action (language, time constraints, focus on collaborative techniques, etc.).

learning experiences to their students. Unfortunately, current approaches to transforming teachers’ pedagogical ideas into their DLE-supported reification require considerable effort, and are difficult to share and reuse (and, thus, are hardly sustainable).

We have presented here the GLUE!-PS system as a solution to these deployment and sustainability problems of DLEs. Using a common data format and two sets of adapters, GLUE!-PS enables the deployment and reuse of learning designs expressed using different design languages into multiple flavors of DLEs, without forcing teachers or institutions to change their choices of technological infrastructure. The results of evaluating the GLUE!-PS system, in two workshops and an authentic deployment study with university teachers, highlight the system’s ability to deploy CSCL designs by teachers under severe time constraints. These results also highlight some of the classical tensions between learning design and more “bricolage-like” approaches to the preparation of the ICT tools for enactment (e.g., up-front design effort versus finer control over the resulting infrastructure, higher reusability versus more flexibility). The pragmatic approach taken in GLUE!-PS represents an important advance toward the learning design idea of “design once, deploy anywhere,” and it may pave the way of learning design adoption by teachers. That, however, will depend on whether a rich-enough ecosystem of LD and DLE adapters is implemented in the future. It will also depend on whether we manage to overcome existing conceptual barriers about CSCL, learning design, and how to make both applicable to teachers’ particular constraints in their everyday practice.

Our current and future work around GLUE!-PS includes the exploration of runtime flexibility of the GLUE!-PS system, and its evaluation through more authentic-setting studies in real DLE-based courses (several of which are already under way). We are also working on the expansion of this learning design approach beyond the space of web-based learning, to include also other kinds of environments such as physical spaces enhanced through augmented reality (AR) techniques [41]. Also, joint work is currently under way with several of the main research groups in the area of learning design (e.g., within the learning design grid22), to expand the GLUE!-PS ecosystem of LD adapters to support some of the main authoring tools available today.

ACKNOWLEDGMENTS

This research was partially funded by the Spanish Ministry of Economy and Competitiveness Projects TIN2008-03023, TIN2011-28308-C03-02, and IPT-430000-2010-054; the Autonomous Government of Castilla y León Project VA293/11-2; and the METIS European Project (531262-LP-2012-ES-KA3-KA3MP). The authors thank the participants in the teacher workshops, the GSIC/EMIC research team, as well as the Learning Design Theme Team funded by the European Union through the STELLAR Network of Excellence for Technology Enhanced Learning (FP7-IST-231913) for their contributions.

Luis P. Prieto received the PhD degree in ICT from the University of Valladolid in 2012. Before receiving his PhD degree, he worked as a research and development engineer for nine years. He is currently a researcher at the GSIC-EMIC Group at the University of Valladolid. His primary research interests include the orchestration of TEL and CSCL activities, professional development to support technological change, and the application of augmented reality, paper, and tangible interfaces to education.

Juan I. Asensio-Pérez received the MSc and PhD degrees in telecommunications engineering from the University of Valladolid, Spain, in 1995 and 2000, respectively, where he is currently an associate professor of telematics engineering. His research interests include CSCL scripting, the integration of external tools in VLEs, the technological support for orchestration of CSCL situations, the semantic annotation of learning tools in the web of data, and the application of augmented reality to support CSCL situations across multiple virtual and physical spaces.

Juan A. Muñoz-Cristóbal received the BS degree in telecommunications engineering and the MSc degree in ICT research from the University of Valladolid, Spain, in 1998 and 2010, respectively, the MSc degree in free software from the Open University of Catalonia, Spain, in 2010, and the MBA degree from the EAE Business School and the University of Barcelona, Spain, in 2012. After working as an engineer for telecommunication service providers for 10 years, he joined the GSIC-EMIC Research Group at the University of Valladolid, where he completed his PhD thesis on technology-enhanced learning.

Yannis A. Dimitriadis received the BS degree from the National Technical University of Athens, Greece, in 1981, the MS degree from the University of Virginia, Charlottesville, in 1983, and the PhD degree from the University of Valladolid, Spain, in 1992, all in telecommunications engineering. He is currently a full professor of telematics engineering at the University of Valladolid. His research interests include technological support to the orchestration of computer-supported collaborative learning processes. He is a senior member of the IEEE.

Iván M. Jorrín-Abellán received the PhD degree. He is an associate professor at the University of Valladolid, Spain, and director of the Center for Transdisciplinary Research in Education. His current research is devoted to the study of the educational implications of computer-supported collaborative learning scenarios, with special attention to new ways of evaluating these particular settings.

Eduardo Gómez-Sánchez received the MSc and PhD degrees in telecommunications engineering from the University of Valladolid, Spain, in 1996 and 2001, respectively, where he is currently an associate professor. His research interests include distributed systems and interoperability issues in the computational support for collaborative learning.