A Rapid Authoring Tool for Converting Existing Online Resources into Widgets

Lujia Zhang, Carsten Ullrich, Ruimin Shen
Department of Computer Science & Engineering
Shanghai Jiao Tong University
Shanghai, 200030, China
dyzhanglujia@126.com, ullrich_c@sjtu.edu.cn, rmshen@sjtu.edu.cn

Keywords: authoring, reuse, widgets

I. INTRODUCTION

Today, e-learning 2.0 [1] calls for learner-driven learning. Beyond LMSs, which often play as an advanced photocopier, Personal Learning Environments (PLEs) [2] enable teachers and learners to assemble their individual environment from tools available in today's Web 2.0. However, this raises the problem that a large number and different kinds of tools and learning objects LOs need to be available in a format suited for assembly [3]. For example, in the School of Continuing Education of Shanghai Jiao Tong University (SOCE), even though the technology for enabling PLEs within the employed LMS was available, few teachers used it due to a lack of resources useful to their specific topics. The problem of too few resources is aggravated by the high costs of authoring of new resources.

One main idea for reducing costs is to enable reuse of LOs and tools (in the Web, tools are often made available as widgets, small interactive software applications typically created with HTML and JavaScript). Reuse with such a broad scope is tackled by different approaches, such as OpenSocial (OS), IMS LTI, IMS QTI, and SCORM.

Recent work investigates simplifying the creation of interactive resources by different means. "Authoring by example"[4] eases tutor development by allowing the author to demonstrate instead of programming. Commercial widget authoring tools like Widgetbox offer on-line authoring of widgets. However, the types of widgets which the online tools are able to create are too restricted for widespread and frequent use in education.

To overcome this problem of PLEs, we put forward a general framework which aims at increasing the usage of interactive learning objects and keeping authoring costs low.

II. REQUIREMENTS

Our framework must satisfy the following requirements: a) reuse all kinds of online resources; b) allow for the widest possible range of interactivity instead of prescribing a restricted set of interactive features; c) enable a tight-coupled integration between the resource and the learning environment; d) extend the functionality of resources, for example, by enabling to collect, analyze and visualize students' interaction information; and e) cross-platform instead of platform-specific.

We compared six different approaches (see TABLE I.). Three of these are general Web technologies, namely links, first-generation widgets (W1, using HTML elements other than links to embed online resources), and second-generation widgets (W2, such as OpenSocial and W3C widgets). The other three are education related, namely IMS LTI, IMS QTI and SCORM. In the table, "+/-/0" resp. stand for high/little/medium support of a particular feature.

TABLE I. RESULT OF COMPARING SIX DIFFERENT APPROACHES

Approach	Interactivity	Reuse	Tight integration	Functionality	Cross-platform
Links	+	+	-	-	+
W1	**0**	**0**	-	**0**	**0**
W2	+	+	+	+	+
LTI	+	-	+	+	-
QTI	-	-	**0**	-	+
SCORM	+	**0**	**0**	**0**	-

Links allows reusing all types of interactivity. However, additional functionalities which can be added to pure links are limited. The interactivity of W1 depends on the target system. Tight integration is difficult, since no pre-defined APIs define such integration. The implementation of additional functionalities also depends on the LMSs and PLEs. W2 supports social networking by offering powerful APIs, which also enable the implementation of additional functionalities. In addition, W2 is able to retain the wide range of interactivity from resources and has become a practical cross-platform approach recently. IMS LTI aims to establish a tight integration between learning applications which cover a variety of interactivity and a LMS. However, due to the high complexity of IMS LTI only few systems support it. IMS QTI significantly limits the possible interactivity, and implementing additional functionalities is not possible. SCORM allows reusing resources cover a wide range of interactivity. The APIs in SCORM are rather restricted and do not allow the implementation of completely new additional functionalities. Also, few systems support SCORM in practice. Based on these considerations, the best pragmatic way was the W2 approach.

III. WIDGETSPACES AND AUTHORING TOOL

Our implementation is based on a Moodle extension which offers OS functionality using so-called widgetspaces [6]. Teachers can easily create a widgetspace in their courses just like any other Moodle resource.

Our authoring tool allows teachers to easily transform existing resources into OS widgets usable in their widgetspaces. This tool consists of two parts: the client part consists of a

978-0-7695-5009-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ICALT.2013.17

webpage which teachers can access through browsers to provide the minimal metadata of a widget required by OS. The server part validates the online resources and creates widgets.

The generation of a widget and its automatic placement in Moodle happens as follows. In each widgetspace, the teacher can add the authoring tool (which is an OS widget, too) by clicking a button. Then, he inputs the URL of the existing resource, and adds metadata. Finally, the authoring tool generates the source code file of the new OS widget, and automatically inserts it into the widgetspace.

Our tool enables teachers to add predefined additional functionalities to widgets during the authoring. This way, the functionality of Moodle itself can be extended. For example, teachers can collect and visualize about how students are using the widgets.

IV. EVALUATION

We performed two rounds of evaluations to assess whether the authoring tool implemented according to our approach would indeed be easy enough to use and provide the intended usability. For the evaluations, we employed a survey based on TAM3 [5], a standard measure to gain information about perceived usefulness (PU), perceived ease of use (PEOU), perceived enjoyment (ENJ) and behavioral intention (BI).

Before implementing the tool, we performed a first evaluation with five higher education teachers from SOCE, who had different expertise in usage of widgets. The evaluation was performed in 30 minutes long one-on-one sessions in which the interviewer presented the overall concept of widgets, followed by a demonstration of a mock-up of the authoring tool. Then, the subject was asked to "author" a resource on his own. Afterwards, the subjects rated 14 TAM3 statements such as "Using the system improves my performance in my job" with a scale from "does not improve my performance at all" (1) to "improves … a lot" (5).

Three teachers fully completed the survey, and two only partly (they argued that the evaluation format did not give them sufficient information for a justified assessment). All in all, the scores (shown in the second column of Table II) were high which indicates that the teachers see a value in such a tool. The outlier is ENJ, with a score of 2.44. We drew the conclusion that the tool would be helpful to teachers but the user experience had to be improved. From the teachers' oral feedback, we realized that the functionality for automatic adding of the newly created widget into the widgetspace is especially important.

TABLE II. RESULT OF TWO ROUNDS EVALUATION

Constructs	First evaluation	Second evaluation
PU	4	4.95
PEOU	4.17	4.45
ENJ	2.44	4.87
BI	4.55	4.93

The implementation took the feedback collected from the first evaluation into account, and thus, the final tool added the functionality for automatic placement. In addition, we added help information to the tool.

For the evaluation of the implemented tool, we chose the same survey and method as before. The only difference was that we skipped introducing the now familiar concept of widgets. The five new subjects were different from the first ones. By comparing the scores of two evaluations (third column of Table II), we can see that the improvement of the tool was positive. The data also shows that PEOU was rated lowest. Teachers explained that the explanation of the different user interface elements was not detailed enough. In addition, the teachers state that they were required to input too much information, even though the data was already minimal. The other constructs PU, ENJ, and BI received a slightly higher score as in the first evaluation. The results indicate that teachers recognized that our tool provides an approach for them to integrate external resources in their courses, which can make their courses more interactive.

The drawback of our survey was that our survey was performed with only few teachers. The sample was rather small, which might distort the result. However, the inter-subject agreement was quite high.

V. CONCLUSIONS AND FUTURE WORK

We presented an approach to quickly transform online resources into widgets. This approach has several advantages over previous solutions. Two small-scale evaluations indicate that our solution empowers the average teachers to become a widget author, without any programming knowledge. The teachers showed their enthusiasm to use this tool in the near future and believe that this authoring tool will help them both during the authoring and teaching process.

There are still many features required to be added to improve our tool. For example, the automatic generation of metadata might improve usability and ease of use even more, especially when adding category information to the metadata to ease later reuse for existing widgets.

ACKNOWLEDGMENT

This project was supported by FP7/2007-2013) under grant agreement no 231396 (ROLE project).

[1] S. Downes, "E-Learning 2.0", elearn 2.0, Oct. 2005.

[2] G. Attwell, "Personal Learning Environments-the future of eLearning?" eLearning Papers, vol. 2(1), pp. 1-7, 2007.

[3] S. Downes. "Learning objects: Resources for distance education worldwide." International Review of Research in Open and Distance Learning, 2001.

[4] K. R. Koedinger, V. Aleven, N. Heffernan, B. McLaren, and M. Hockenberry, "Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration." In Intelligent Tutoring Systems, pp. 7-10, Springer Berlin/Heidelberg, 2004.

[5] V. Venkatesh and H. Bala, "Technology Acceptance Model 3 and a research agenda on interventions." Decision Sciences, vol. 39(2), pp. 273-315, 2008.

[6] E. Bogdanov, C. Ullrich, E. Isaksson, M. Palmer, and D. Gillet, "From LMS to PLE: a step forward through opensocial apps in moodle." In Advances in Web-Based Learning-ICWL, pp. 69-78, Springer Berlin Heidelberg, 2002.