Interactive Scalable Lectures with ASQ

Vasileios Triglianos and Cesare Pautasso

Faculty of Informatics, University of Lugano (USI), Switzerland
{name.surname}@usi.ch
http://asq.inf.usi.ch/

Abstract. Taking full advantage of the Web technology platform during in-class lectures requires a shift from the established scheme of online education delivery that utilizes the video channel to embed all types of content and gathers student feedback via multiple choice questions or textual answers. In this paper we present the design of ASQ to deliver interactive content for use in heterogeneous educational settings with a large number of students, taking advantage of the co-location of students and instructors and building upon the latest capabilities of the Web platform. ASQ is centered around interactive HTML5 presentations coupled with a versatile microformat to create and deliver various types quizzes and scalable, synchronous/asynchronous feedback mechanisms.

1 Introduction

The Web is increasingly used to deliver educational content, being the medium of choice for the popular [1] massive open online courses (MOOCs) and "flipped" (or blended or hybrid) classrooms [2]. Courses are delivered through video lectures and students are assessed either automatically, through multiple choice or text input quizzes, or via peer assessment. Student Response Systems that take advantage of the increased number of Web-enabled devices are finding their way to brick and mortar classrooms replacing traditional hardware clickers. Video as the prominent delivery format of an online lecture is not optimized for the Web medium; it is cumbersome to author and interleave with quizzes and various types of assessment; and it lacks the interactivity and the features that modern Web technologies offer. Today's Web technology can support lectures with highly interactive content such as selectable text, forms, 3D graphics, that can be reactive to a student's input and personalized [3] for different learning styles.

Also of importance are the current assessment models which have remained stale for years: formative or summative assessment is predominantly perfomed through multiple choice or free text quizzes which do not encourage experimentation and creation of original content. In terms of communication models, either between students or between students and teachers, findings suggest that synchronous communication, as a complement to asynchronous communication, can potentially enhance participation in online education [4].

Our goal is to demonstrate ASQ, a platform to create and deliver interactive lectures and gather student feedback in synchronous or asynchronous settings. ASQ makes full use of the HTML5 capabilities of modern Web browsers

S. Casteleyn, G. Rossi, and M. Winckler (Eds.): ICWE 2014, LNCS 8541, pp. 515–518, 2014.

and provides teachers with the ability to author, deliver and reflect upon the performance of their interactive educational content, while it gives students an additional communication channel to demonstrate their learning progress and actively participate during a traditional lecture.

2 ASQ in the Classroom

ASQ aims to promote the shift from the traditional frontal lecture paradigm (monologue) to interactive bi-directional presentations and discussions (dialogue), through the following features that will be demonstrated: **Delivering educational content:** Interactive lectures are shared with students through a simple URL pointing to the lecture slides. Retrieving the link will connect the student browsers to follow the online presentation. As the instructor navigates through the slides, the navigation events propagate to the connected students which can synchronously follow the material on their Web browsers. This can enhance the accessibility of the lecture, since the material is now displayed in front of the students as it is being explained to them.

Authoring educational content: ASQ manages different kinds of educational content: lecture slides and questions. These can be authored using all the rich multi-hyper-media (images, videos, text, interactive widgets, audio) capabilities of HTML5 compliant Web browsers. Exemplary interactions include selecting or highlighting text, keeping notes for a slide in provided placeholder, playing back audio and video, filling forms, dragging and dropping textual or iconic elements, and any sort of interaction that can be implemented in a Web page using Javascript, CSS3 and HTML5.

Interactive questions and microformat: Questions can be used for both formative and summative assessment, embedded in lectures, collected in quizzes and homework assignements or exams. Each question type is associated with a set of related statistics to be computed over the answers collected by the students. The configuration for rendering questions, assessing answers and visualizing the results of the assessment is controlled with a simple microformat, as shown in the following example:

```
<!-- a multiple choice quiz with id q-1 -->
<article class="asq-question multi-choice choose-0-n" id="q-1">
  <h3 class="stem"><img src="img/shape.png"></h3>
  <ol>
    <li class="option">This figure is a square.</li>
    <li class="option">This figure is a circle.</li>
    <li class="option">This figure is symmetric.</li>
    <li class="option">This figure has four corners.</li>
  </ol>
</article>
<!-- statistics for the quiz q-1 -->
<article class="stats" data-target-assessment="q-1"
    id="stats-q-1"></article>
```

The microformat parser searches for elements that contain the `asq-question` keyword in their `class` attribute. Once a question is identified the parser searches for its type, in this case `multi-choice`; and for configuration options, in this case students can choose as many of the available multiple choice options they want (`choose-0-n`, but also `choose-1` or `choose-1-n` are possible). Each question type features keywords that provide information about its structure. In the multiple choice example, the class `option` of the `li` elements instruct the parser to store these elements as the options of the multiple choice question. Similarly, statistics are processed with the parser searching for the `asq-stats` keyword. The `data-target-assessment` points to the associated question. Once all questions and statistics are parsed, they are stored in the database and then a markup generator is invoked, that injects necessary markup like form fields and buttons.

Innovative quiz types: Besides standard multiple choice (MC) and single-line text input (STI) quizzes, ASQ currently features two question types specifically targeting the Computer Science domain: code highlight and code input.

Classroom flow: ASQ enhances the educational material by weaving support for complex interactions. Instructors can highlight important points on the presentation and have the marking happen instantly on the students screens. Students can answer quizzes and questions embedded in the slides –individually or in teams– giving instant feedback to the instructor about their level of comprehension. ASQ supplies instructors with a continuous stream of events and statistics related to quizzes, like student progress, correct versus wrong answers, enumeration of actual solutions. Upon receiving the results an instructor may choose to discuss them with individual students, share them with the class through the projector or the students screens to present insightful statistics.

3 Architecture

ASQ follows a client server architecture (Fig. 1). In the backend the logic is implemented in express.js on top of node.js, a choice mandated by the need for efficient I/O operations. The Web server design follows a REST approach treating educational material (lectures, slides, questions, answers) as Web resources.

Bi-directional communication between instructors and students is implemented using WebSockets. This allows for real-time exchange of a fairly big volume of events that are crucial for monitoring progress, supporting complicated assessment modes (for example peer assessment) or fine-grained logging of student actions. User management in ASQ, involves user roles like professors, teaching assistants and students and user permissions defined for each course. Interactions with the website that involve regular HTTP requests use cookie-based authentication, while realtime communication uses token-based authentication and group-based WebSocket namespaces.

The database consists of MongoDB collections for question instances, question types, lectures, sessions, users, and answers. Session events and socket events are stored in a Redis key-value store for scalability and increased performance.

ASQ supports both client- and server-side rendering with Dust.js templates. Question instances are parsed and rendered on the backend, which allows for

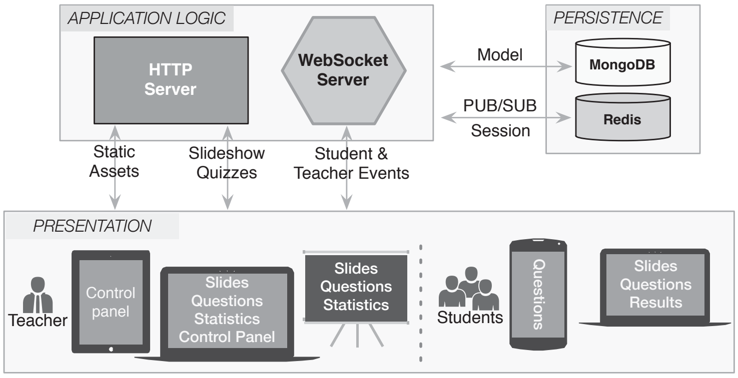

Fig. 1. ASQ Architecture

caching and saves CPU time on the clients. Dynamic elements like graphs and dialog windows are rendered client-side. Moreover, ASQ takes into account display size and user roles to optimize the rendered content. For example, in a student's smartphone with limited HTML5 features and/or a small screen, ASQ only renders the questions so that students can answer them, but does not show the statistics visualizations.

4 Conclusion

In this paper we demonstrate the main features of ASQ, a tool for delivering highly interactive educational content on the Web. Traditionally, instructors may only gather a small sample of answers from the students (who may have to speak up in front of a large audience). ASQ opens a new channel through which teachers can collect the answers of every single student attending a lecture. In the future, we will augment the question delivery system to enable the formation of groups of students to collaboratively solve problems. Finally we plan on evaluating the educational outcomes of ASQ through a summative evaluation that aims to compare a traditional classroom with an ASQ-enabled classroom.

References

1. Shah, D.: MOOCs in 2013: Breaking Down the Numbers. Teasing out trends among the unabated growth of online courses. edSurge (December 2013)
2. Bergmann, J., Sams, A.: Flip your classroom: reach every student in every class every day. International Society for Technology in Education Eugene, OR (2012)
3. Dolog, P., Henze, N., Nejdl, W., Sintek, M.: Personalization in distributed e-learning environments. In: Proceedings of WWW Alt. 2004, pp. 170–179. ACM (2004)
4. Hrastinski, S., Carlsson, S.A.: Participating in synchronous online education. Lund Studies in Informatics (6) (2007)