

RESPONDING TO QUESTIONNAIRES ON THE WEB USING XWQUEST

José Paulo Leal

*DCC-FC & CACS, University of Porto
R. Campo Alegre, 823 -- 4150-180 Porto, Portugal
zp@dcc.fc.up.pt*

ABSTRACT

This paper reports on the design, implementation and evaluation of XwQuest, a web tool for responding to questionnaires on the Web. The distinctive feature of this tool is an XML definition of the questionnaire that focus on questions and admissible answers while avoiding presentation details. This questionnaire definition is processed on the browser side and converted to an Ajax application. Collected responses are periodically sent back to the server and can be retrieved by researchers and processed on a standard spreadsheet program. The paper details the questionnaire language of XwQuest and a generator that converts them into Ajax applications. Two case studies where XwQuest was actually used with good results are also presented.

KEYWORDS

Web programming, XM languages, Ajax applications

1. INTRODUCTION

Surveys are routinely used in many research areas, from social sciences to program development or marketing. To administer them the researcher may choose from traditional methods - telephone, mail, personal interviews - and on-line methods, either email or web based. The web is usually preferred over email since it provides interactive forms, more adequate for large questionnaires. Web surveys have a number advantages that in many cases outweigh its disadvantages. In fact, web surveys are easy to create and manage, can reach almost immediately a vast number of persons, independently from their geographical location and with a very small cost. Researchers have to take in consideration that web surveys tend to have a smaller percentage of complete responses and may skew results since not all demographic groups use equally the web. Nevertheless, everything considered, there are many cases where web questionnaires are the best way to collect data for a statistical survey.

Several web survey systems are available, some of them can even be used for free. They are usually large and complex systems that control the complete life-cycle of a survey but lack interoperability features to export and reuse questionnaire definitions. Tasks that would be simple to execute on a conventional text editor, such as adapting the questionnaire to different types of respondents, or translating questionnaires to different languages, become unnecessarily difficult to perform.

The need for a web survey tool based on a questionnaire definition lead to the development of XwQuest. The distinctive feature of this tool is the XML document defining the questionnaire that is converted on client side to an Ajax web application, that collect answers and periodically stores them on the server side. Collected data can be downloaded by researchers at any time, in a format supported by any spreadsheet application.

The remainder of this paper is organized as follows. Section 2 introduces the main concepts and tools related to questionnaire construction and presents systems to process web questionnaires. Section 3 provides an overview of XwQuest with emphasis on the role of the questionnaire definition document, conforming to the language definition detailed in Section 4. Section 5 details the implementation of the Ajax application generator that processes questionnaire definitions. This tool has been effectively used on two on-line surveys

that are reported on section 6. The final section highlights the contributions of this research and points directions for future work.

2. QUESTIONNAIRE CONCEPTS AND TOOLS

A questionnaire is a sequence of questions asked to series of individuals to obtain answers that can be processed statistically (Martin 2006). Most of the questions included in a questionnaire can be divided in two general types: close ended questions and open ended questions.

Close ended questions are very popular since they are more precise, easier to answer by the respondents, contribute to a higher response rate and produce data that is simple to code, analyze and compare across surveys. However, open ended questions allow for answers unanticipated by questionnaire designers and do not introduce a bias by framing the answer with set of predefined values.

There are different types of closed ended questions, including yes-no (agree-disagree), multiple (forced) choices and scaled questions (ordered response categories). These types of questions have in common the fact that answers are reduced to a set of predefined options (yes or no; A, B, C, or D; 1 to 5). Other types of questions include contingency questions and matrix questions. Contingency questions are questions that are meaningful only to respondents that gave a particular answer to a previous question. They avoid asking questions to people that do not apply, for instance, asking men if they have ever been pregnant.

Questionnaires frequently use the same categories for a sequence of questions, forming a matrix with the categories as columns and the questions as lines. This provides a better use of layout space and respondents time since it emphasizes the similarity among questions.

There are a number of systems for running web questionnaires, such as Zoomerang, SurveyGizmo, SurveyMonkey, QuestionPro, SurveyGold and eSurveysPro, (Toledano 2008) to name only a few. An entry level version of all mentioned systems is available for free with limited features. For instance, all these systems validate answers and implement contingency questions but some only provide these feature in the paid version. All these systems have a significant number of question types, ranging from 14 to 17. Questionnaire customization (include own logo/images, "Thank you" page, remove branding) are usually available only on paid versions.

All the mentioned systems aim to control the complete life-cycle of a questionnaire and thus have limited interoperability features. These systems control every step from creation, management, deployment and reporting of a questionnaire. They all export data to in CVS, at least in their paid version. However, they do not export questionnaire definitions in a standard format.

There are few references in the literature to XML languages for describing questionnaires. The IQML project (Lamb and Fairgrieve 2000) had as goal the development of a software suite and a markup language for intelligent questionnaires and includes a specification for that purpose. The language developed for the Simple Survey System (SSS) (Bethke 2007) has a simple structure, supports few basic answer types and does not separate answer and questions definitions. There has been also effort to extend IMS Question & Test Interoperability (QTI) to survey questionnaires (García et al. 2002).

3. SYSTEM OVERVIEW

Simplicity is the main feature of XwQuest. This simplicity is the key for its efficiency and maintainability. The complete system consists of a few files, mostly processed on the client side. They concern only those activities related to questionnaire response. Other concerns, such as authentication and security, can be configured on the host HTTP server. Moreover, this system fits well in an existing web site and its graphical layout may be easily changed.

The diagram in Figure 1 identifies the main components of XwQuest and how they interplay to process a questionnaire and process responses. Several files in different programming languages, represented graphically by a rectangle with a folded corner, are published on a web server directory. Using a web browser the respondent accesses the questionnaire and initiates the process. There are three processing moments represented graphically by rotor with a label that identifies the type of engine:

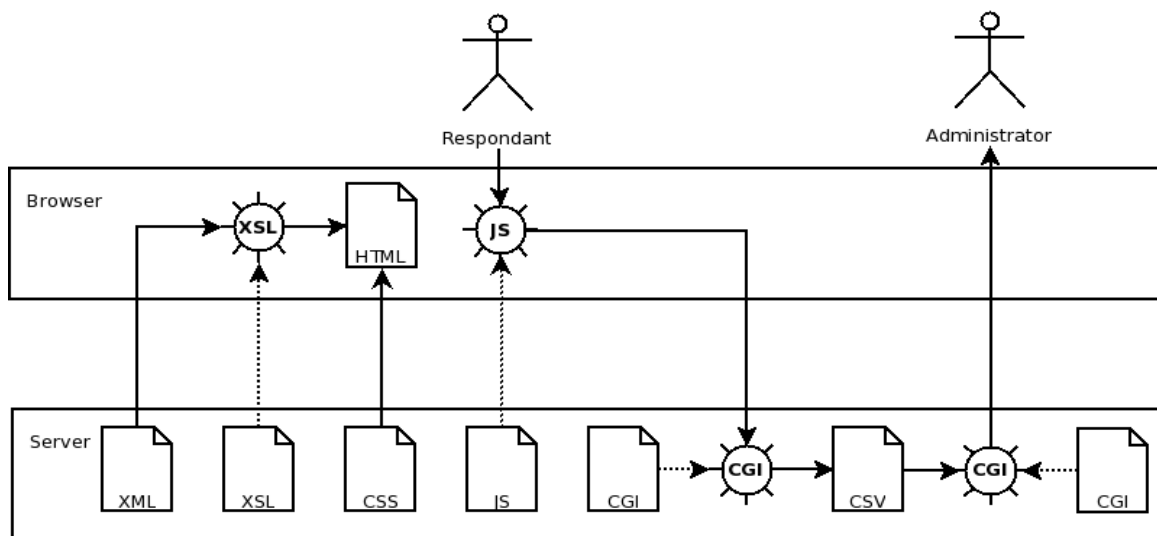


Figure 1. Overview of XwQuest

XSLT The XML file defining the questionnaire contains a processing instruction referring a XSL transformation (Clark 1999). This transformation is also loaded to the browser and processed by its XSLT engine to convert the questionnaire definition to HTML. When formatted, the questionnaire loads from the server a Cascade Style Sheet (CSS) file for controlling presentation, and a JavaScript (JS) for handling user interaction.

JS Events resulting from user interaction with the questionnaire are processed by the JavaScript engine of the browser, using Ajax techniques (Crane et al 2005). Navigation through the questionnaire panels and enforcement of mandatory questions is done using the Document Object Model (DOM) Application Interface (API). Responses collected during the interaction were sent back to the server using XMLHttpRequest API.

CGI Data sent from the browser is processed by a CGI on the server side. This program receives a stream of data in the Comma Separated Values (CSV) format and records it on the file system. Using another CGI program the survey administrator can download the recorded data from the web. These CGI programs are shared by a set of questionnaires published in the same XwQuest installation.

The Ajax application produced on the web browser presents the questionnaire as a sequence of panels. Each panel groups a collection of related questions that are presented simultaneously to the respondent. Panels have an instrumental role in questionnaire validation and navigation. Respondents may only proceed to the next unanswered panel after responding to all mandatory questions, but they can navigate through the answered panels. When a panel is answered all data collected to that point is sent back to the server, overwriting answers to that questionnaire from the same respondent. Usually, only complete questionnaires are accepted for statistical processing but the researcher has more liberty to define this criteria afterwards. In the case of a long questionnaire with few complete responses, if all intermediate data has been recorded then the researcher may select a set of answers answered by most respondents, for instance.

The XwQuest package includes also a feature to export in CSV format responses to a given questionnaire. This feature is implemented by a CGI that concatenates all files recording individual responses. Depending on configuration, this data is usually presented by the browser using a helper application, typically a spreadsheet program.

4. QUESTIONNAIRE SPECIFICATION

In XwQuest a questionnaire is specified using the XwQuest language. This language is formally defined as an XML Schema Definition (XSD) (Fallside and Walmsley 2004) that may be used to assist in editing a questionnaire specification.

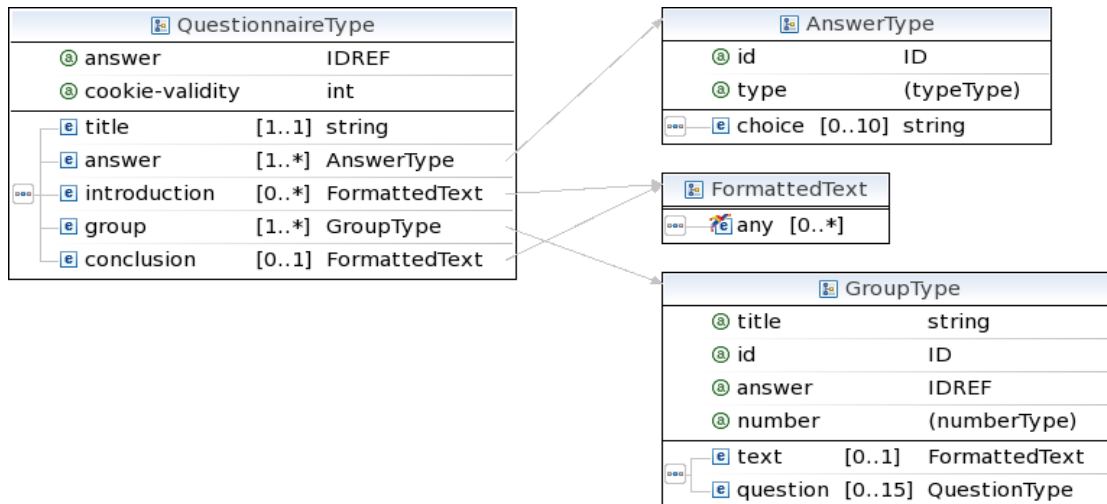


Figure 2. Top-level elements of the XwQuest language

The root element of the XwQuest language has the `QuestionnaireType` represented by the diagram in Figure 2. The most important top-level elements are `answer` and `group`. The former defines the types of answers associated to questions, while the latter are containers of related questions. Answers are identified by an XML ID and are referred through IDREF attributes. There is usually a prevalent type of answer throughout a questionnaire, hence a default answer type can be set by the `answer` attribute of the root element.

The top-level includes also elements with textual information. The mandatory element `title` contains a string to format the title of the questionnaire. The `introduction` and `conclusion` optional elements contain formatted texts that are presented to the respondents respectively in the beginning and in the end of the questionnaire.

A questionnaire may define several answer types. As mentioned before, each answer definition has a unique XML identifier and a base type. The base types currently supported by XwQuest are the following:

- `radio` -- a collection of predefined answers selected with radio buttons;
- `select` -- a collection of predefined answers selected with a menu;
- `text-line` -- single line free text answer;
- `combo` -- a combination of a `select` and `text-line`;
- `text-area` -- multi-line free text answer.

The close ended -- `radio`, `select` and `combo` -- base types require a collection of predefined answers that is defined by the `choice` sub-elements, each containing a text string.

As represented in Figure 3, `group` elements are containers of sequences of up to 15 questions. Each group has its own `id` and `title` attributes. The former is an XML ID for internal identification of the group's questions. The latter is a string used for formatting the group on the web interface. A group may start with an introductory `text`, formatted as the `introduction` and `conclusion` top-level elements.

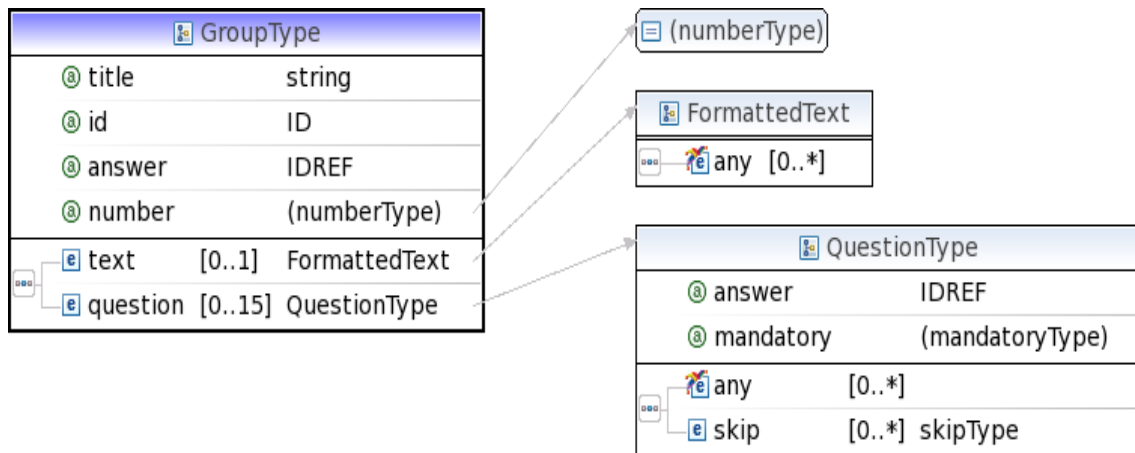


Figure 3. - Groups of questions elements

A `question` is a formatted text with an associated type of answer. By default, the answer type is inherited from the group or, if omitted there, from the questionnaire root element. If present, the `answer` attribute overrides previous definitions.

Contingency questions are implemented using the `skip` element. This element may appear in the `question` elements and associates a possible answer to group of questions to omit. The `mandatory` attribute is used to control if a question must be answer or not. By default open ended answers, those with free text base types, are optional and all other are mandatory.

5. AJAX APPLICATION GENERATOR

The questionnaire specification language presented in the previous section is automatically converted by XwQuest into an Ajax (Crane et Al 2005) application. Ajax stands for *Asynchronous JavaScript and XML* and is a web application development technique that mixes communication of XML data outside the normal request-response cycle of HTTP, with the manipulation of the page structure and content using the DOM (Hors et al 2000) API.

As explained in Section 3., the first processing step of questionnaire in a XwQuest is a XSLT 1.0 (Clark 1999) transformation. This transformation converts a questionnaire specification into an XML compliant HTML 4.1 transactional document. The target HTML document contains the complete questionnaire layout split in several overlapping and invisible panels. These panels are made visible by user interaction, one panel at the time, producing the effect of navigation through the questionnaire.

To start the XSLT transformation, a template matches the root element `questionnaire` and produces a skeleton of the HTML layout, with references to the included CSS, and JavaScript files and initialization of JavaScript variables. This template iterates over the `introduction`, `group` and `conclusion` elements in the definition and produces an invisible panel for each one. The heart of the transformation is the conversion of groups of question definitions into a layout. The template matching `group` elements iterates over `question` element and applies a template according to its type.

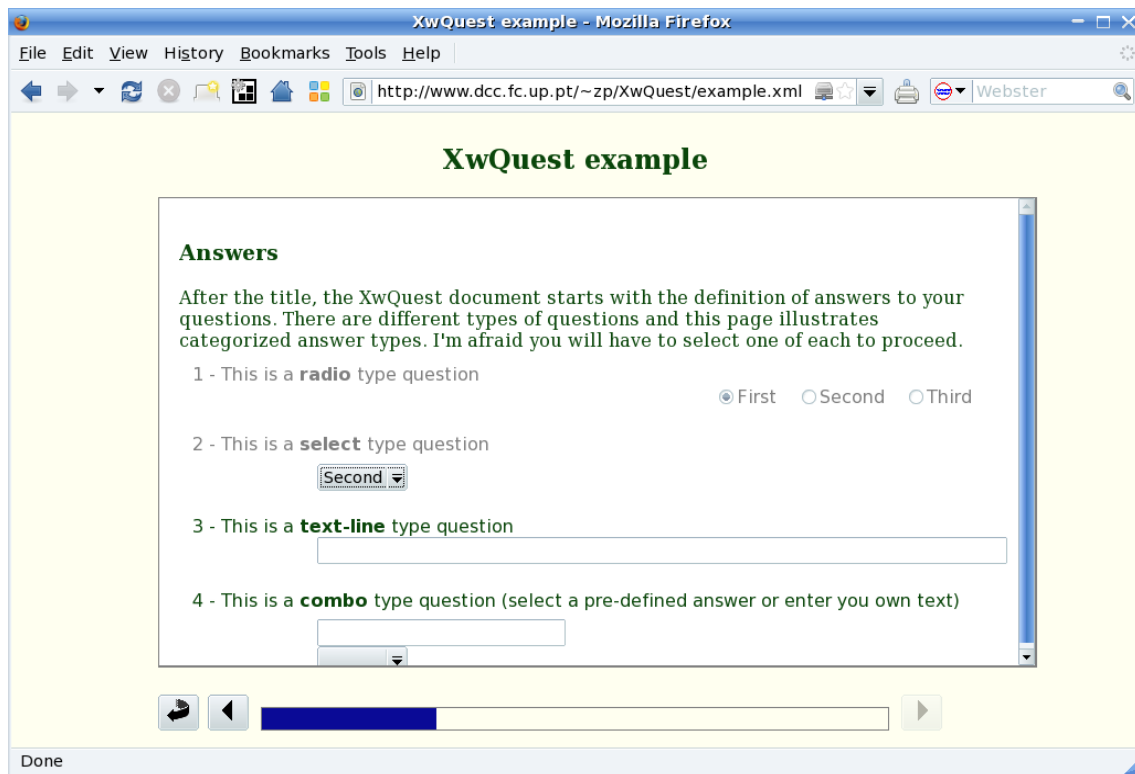


Figure 4. - A screenshot of XwQuest

A screenshot of the web interface produced for an example questionnaire distributed with XwQuest is presented in Figure 4. This image shows a typical screen of XwQuest with the questionnaire title in the header, a central panel displaying the current group of questions and a navigation area in the footer.

The central panel has a fixed dimension and presents a group of questions in a scrollable area. The header of the panel presents the group's title, followed by the introductory text of the group and the respective questions. Questions are converted into HTML controls according to their base types and options. Mandatory questions already answered are displayed in gray to highlight those questions that still need to be answered.

The center of the navigation area is a progress bar showing graphically the amount of questions already answered. This information is provided to the respondents to prevent them from abandoning the questionnaire before completing it. On each side of the progress bar, there is a button to move to the previous and following panels. The following panel button is only activated when all mandatory questions in the current panel have been answered.

Event processing in XwQuest is implemented by a JavaScript library. Event handlers in HTML form objects are responsible for launching the execution of functions in this library to implement the validation, navigation, and persistence. These three aspects are intimately connected and are triggered by the same user events.

When the user selects an option from a close-ended question, that question is immediately formatted in gray. When most questions in the current panel are already answered, those still unanswered are highlighted for the fact of being only formatted in black. This is a visual clue to help the respondent identify those questions that need to be answered in order to proceed to the next panel, not itself a validation.

Validation and persistence are connected to a specific navigation action - moving to the following panel. At the moment XwQuest checks if all mandatory questions in the current panel are answered and blocks the action otherwise. Before moving to the next panel, the current answers are persisted in the server.

Persistence is implemented using the XMLHttpRequest JavaScript object. This object enables HTTP communication outside the web request-response cycle. It was originally meant for communicating XML data but can be used for sending and receiving data in plain text. This tool uses it solely for sending text in

the Coma Separated values (CSV) format. These values are send with an associated questionnaire - the questionnaire XML definition file name – and respondent.

Respondents in XwQuest do not have to be authenticated. Instead, the JavaScript library assigns an authentication token to each respondent, to ensure that his or her responses are not mixed with those of others. This token is generated from the current time and a random digit string. The generated token is stored in the browser as a cookie so, if the respondent accesses again to the questionnaire, new answers will override the old ones. This features is controlled by the `cookie-validity` attribute of the root element of questionnaire definition that specifies a time to live for this cookie.

6. CASE STUDIES

The application generator of XwQuest was tested with sample questionnaires using all the major web browsers, namely Mozilla Firefox 3, Internet Explorer 7, Google Chrome 1, Opera 9 and Safari 3, in both Linux and Windows, depending on the used browsers. After these initial tests XwQuest was effectively used on a couple of surveys. That experience was useful to validate the proposed approach and also to detect points needing improvement. The two surveys in questions are succinctly described in this section.

User satisfaction with Mooshak

Mooshak is a web system designed in 2001 for programming contest management. It has been used in several contests, both in Portugal and abroad, including international contests. Although not initially designed for that purpose, the system is also being used as a pedagogical tool. The system was object of a user satisfaction evaluation (Leal and Santos 2008), covering both the competitive and pedagogical contexts. User satisfaction was considered as a combination of usability and usefulness of the system and the selected methodology required users to respond to questionnaires.

These questionnaires included many questions and had to be adapted to the main user profiles supported by Mooshak: teams, judges and administrators. Moreover, its was decided that these questionnaires should be available at least in two languages: Portuguese and English. Finally, 6 different questionnaires were produced, one variant for each of the three main profile and their translation to English. Due to the geographical dispersion of the users of Mooshak, scattered by several continents, a web application was the most efficient way to collect questionnaire responses and XwQuest was used for that purpose.

These user satisfaction questionnaires were available to potential respondents for a two week period during the Spring of 2008. From the 560 responses only 181 where considered valid. All incomplete questionnaires were discarded from the final study. Respondents could identify their origin and several counties were mentioned, such as Portugal, Spain, Brazil, Germany, India, China, Singapore and Mozambique. The performance of XwQuest was very good. The only problem worth mentioning was a spurious duplication of questionnaire answers, probably due to a bug in cookie handling by a non standard browser.

Internet Users

The project "*Tu e a Internet*" (Falcão and Correia 2008), meaning "You and the Internet" in Portuguese, is concerned with risks taken by teenagers when disclosing their personal data on the Internet. An on-line survey was the chosen tool to estimate the awareness and perception of vulnerability among the youth population.

The survey should be integrated on the web site of the project and adapted to its graphical design. Since XwQuest is a simple system, easy to integrate in any HTTP server with CGI support, and has a clear separation between the generated presentation and graphical formatting, these goals were easy to achieve.

Questionnaires for this project required contingency questions (i.e. questions presented to respondents depending on previous answers). Also, respondents were asked (but not forced) to provide sensitive personal information, and that requires control over the optionality of answers. These features were added to XwQuest as a result from those requirements.

An initial pilot experience was conducted with students from a Portuguese high-school. Once validated the infrastructure, the survey was publicized on social networks. Almost 5000 individuals with ages between 13 and 23 answered the questionnaire but only 3752 complete answers were considered.

7. CONCLUSION AND FUTURE WORK

This paper presents a new tool for responding to questionnaires on the web based on XML technologies. The major contributions are a questionnaire definition language and a web application generator. The XwQuest definition language focuses on the content of the questionnaire rather than on presentation issues. The questionnaire definition is a valid XML document, complying to a XML Schema definition. It provides a convenient format to create, modify, archive, share and reuse questionnaire definitions. The XwQuest generator converts the questionnaire definition into an Ajax application using XSLT transformations. A JavaScript library processes user events and reports collected data back to the server, where it can be downloaded in CSV formats for statistical processing.

The XwQuest tool has already been used in two online surveys, each with a significant number of responses. These results validated the proposed approach, contributed to improve the questionnaire definition language and to correct some glitches on the Ajax application generator. This tool is open source and is available for test and download from <http://www.dcc.fc.up.pt/~zp/XwQuest>

Both the questionnaire definition language and the web application generator have scope for improvement. The questionnaire definition language can be improved with new basic answer types, to deal with matrix answers and Likert scales, for instance. The XSLT transformation was designed with extensibility in mind, hence the impact of the supporting these features in the application generator is expected to be small. So far XwQuest has been used solely with questionnaires in static XML documents but it would be interesting to couple it with a repository of questions to generate questionnaires on the fly. Such a system could be useful to generate self evaluation tests within a eLearning platform, for instance.

REFERENCES

- Elizabeth Martin, *Survey Questionnaire Construction*, 2006, <http://www.census.gov/srd/papers/pdf/rsm2006-13.pdf> [Last visited in 2009-05-28]
- Yann Toledano, *Web Survey Tools Comparison Matrix*, 2008, <http://www.techsoup.org/binaries/files/SurveyToolsComparisonChart.pdf> [Last visited in 2009-06-08]
- J. Lamb and J. Fairgrieve, 2000, *IQML: A Software Suite and Extended Mark-Up Language (XML) for Intelligent Questionnaires*, Proceedings of the Association of Survey Computing (ASC) conference 'Automatically better? The impact of automation on the survey process, Imperial College, London
- Bethke, A.D., 2007, *Using XML as a questionnaire specification language*, Proceedings. IEEE SoutheastCon, 2007,
- Elena García and Miguel-ángel Sicilia and José-ramón Hilera and José-antonio Gutiérrez, 2002, *Extending Question & Test Learning Technology Specifications with Enhanced Questionnaire Models*, Proceedings of the International on Technology Based Higher Education and Training,
- James Clark, *XSL Transformations (XSLT)*, 1999, {<http://www.w3.org/TR/xslt> [Last visited in 2009-05-22]
- Dave Crane and Eric Pascarello and Darren James, 2005, *Ajax in Action*, Manning Publications Co, Greenwich, CT, USA
- David C. Fallside and Priscilla Walmsley, *XML Schema Part 0: Primer Second Edition*, 2004, <http://www.w3.org/TR/xmlschema-0/> [Last visited in 2009-05-11]
- Arnaud Le Hors, Philippe Le Hégarret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion and Steve Byrne, *Document Object Model (DOM) Level 2 Core Specification*, 2000, <http://www.w3.org/TR/DOM-Level-2-Core/> [Last visited in 2009-04-13]
- José Paulo Leal and Ana Santos, 2008, *Avaliação de Satisfação dos Utilizadores do Mooshak*, Proceedings of CIAWI 2008 - Conferência Ibero-Americana WWW/Internet, Lisboa, Portugal
- Filipa Reis and Manuel Eduardo Correia, 2008, *Public Awareness Concerning online Privacy Rights: Attitudes and perceptions of vulnerability among the youth*, "EU Kids Online" Conference, London