An Extension of QSL for E-testing and Its Application in an Offline E-testing Environment

Zhe Wang, Yuan Zhou, Bo Wang, Yuichi Goto, and Jingde Cheng

Department of Information and Computer Sciences,
Saitama University, Saitama, Japan
{wangzhe,shuugen,wangbo,
gotoh,cheng}@aise.ics.saitama-u.ac.jp

Abstract. E-testing is to perform all processes from preparing questions to marking collected answer of the questions in a completely electronic way. Now, various e-testing are done, and there are many kinds of e-testing systems. When people want to do an e-testing or order a new e-testing system, they should specify the e-testing or e-testing system. A Specification language helps to create precise and adequate specifications of e-testing and e-testing systems is demanded. QSL is a specification language for specifying various e-questionnaire and e-questionnaire systems. QSL is a hopeful candidate of the required specification language because both testing and questionnaire have similar processes. However, the current version of QSL does not take e-testing and e-testing systems into account. This paper presents an extension of QSL to deal with e-testing and e-testing systems, and shows a real application of extended QSL in case of an offline e-testing environment.

Keywords: QSL, Specification language, E-testing, Offline e-testing environment.

1 Introduction

Test is a general and indispensable method, and widely used to assess people's achievement, ability, and characteristics in education, enterprise, medicine, and government [16]. E-testing is to perform all processes from preparing questions to marking collected answer sheets in a completely electronic way. An e-testing system is a system that provides an environment with its users to do e-testing. In recent years, there are various e-testing systems have been developed ad hoc because purposes and procedures of tests are different from each other. When examiners want to use an e-testing system to execute various e-testing, they should specify the e-testing. If an e-testing system cannot satisfy examiners' needs, they may order a new e-testing system. They also should specify the e-testing system. Therefore, people need a specification language which can help them to create precise and adequate specifications for various e-testing and e-testing systems.

QSL [23] is a specification language for specifying various e-questionnaire and e-questionnaire systems, and it is a hopeful candidate of the required specification language owing to significant similarities of contents and structures between e-questionnaire

James J. (Jong Hyuk) Park et al. (eds.), *Advanced Multimedia and Ubiquitous Engineering*,
Lecture Notes in Electrical Engineering 352, DOI: 10.1007/978-3-662-47487-7_2

and e-testing. However, the current version of QSL does not take e-testing and e-testing systems into account. This paper proposes the extension of QSL for e-testing so that users can use QSL to specify various e-testing as well as e-testing systems. The paper also shows a real application of extended QSL in case of an offline e-testing environment. The rest of the paper is organized as followed: Section 2 gives introduces in QSL. Section 3 presents an extension of QSL for e-testing. Section 4 shows an offline e-testing environment as a use case of extended QSL. Finally, some concluding remarks are given in Section 5.

2 QSL: A Specification Language for E-questionnaire Systems

QSL serves as a communication tool among specifying, developing, and using various e-questionnaires and e-questionnaire systems [23]. QSL addresses the needs by providing various desirable functions of e-questionnaire systems, pointing out necessary items to make specification clear and precise, and being used as a format for data exchange.

QSL is based on XML [22]. QSL provides primitive elements that are used to specify various e-questionnaires and e-questionnaire systems. A primitive element consists of entity and representation. Entity is uniquely identified regardless of changing representations. Representation is to express the corresponding entity. We can describe specifications of various e-questionnaire systems by combining the primitive elements.

QSL can be used in three ways [22, 23]. At first, QSL can be used to specify e-questionnaire systems. In other words, QSL can be used to specify the functions the system provides. For example, QSL provides tags for four kinds of participants: sponsors who organize an e-questionnaire, questioners who design and ask questions, respondents who take part in answering, and analysts who analyze the result and give the results. Secondly, QSL can be used to specify e-questionnaires on a system, i.e., QSL can be used to describe restrictions in each process of e-questionnaire. For example, for an e-questionnaire, QSL is used to describe the distribution method such as e-mail, web-link, or other offline methods. At last, QSL can be used for a format of data exchange to describe questionnaire data and response data. For instance, QSL is used to describe various questions and answers.

3 An Extension of QSL for E-testing

3.1 E-questionnaire and E-testing

To clarify the differences between e-questionnaire and e-testing, we investigated 20 e-testing systems [1-15, 17-21] on the Internet. The differences between e-questionnaire and e-testing are in five aspects: participants, process, data, logic, and authorization. Firstly, participants are the people who take part in an e-questionnaire or e-testing. An e-testing system contains not only those four kinds of participants mentioned in Section 2, but also two new kinds of participants who are monitor and marker. Monitor is a person who monitors the whole e-testing for illegal behavior,

and marker is a person who marks the responses of respondents and gives the results. Secondly, both e-questionnaire and e-testing have similar procedure except marking. After collecting the response data of all respondents, marker should mark the response data and give scores of each question and total scores of each respondent. Thirdly, data of e-questionnaire and e-testing are used to describe many research papers and record many response data by different respondents. Unlike e-questionnaire, sample answer of each question is demanded to mark answers in e-testing. Similarly, score of each answer and total score of answers are also demanded. Besides, questions in e-testing involves wider and more professional field, such as mathematical formula, periodic table of chemical elements, etc. Moreover, the logic in an e-questionnaire is a facility to control which question is showed in which order. In e-testing, the logic is used to prevent cheating activities. The new logic type for e-testing is randomization which is showing the questions and options to respondents in a random order so that they cannot peep other respondents for answers. At last, the authority is used to describe the authority of all the participants both e-questionnaire and e-testing, such as the writability and readability for a research paper, response, sample answer (for e-testing) and result. However, e-testing has more strict management of authority in order to stop and prevent illegal behaviors.

To specify various e-testing and e-testing systems, these differences should be covered by QSL. However, the current version of QSL does not deal with those differences.

3.2 QSL Extension for E-testing

According to the differences between e-questionnaire systems and e-testing system, some new elements have been extended to specify various e-testing and e-testing systems.

To deal with the difference of participants, we add two new elements to describe new participants. They are *monitor* and *marker*. An example is shown below. The participant element is *monitor* which has an attribute named "id", the value of the element is used to describe the name of the monitor.

```
<monitor id="m001">John Smith</monitor>
```

In order to cover the difference of data and marking procedure, we add three new elements which are *sampleanswer* , *score*, and *result*, and we add an attribute *type* for *score* which contains “question”, “section”, “total”, “respquestion”, “respsection”, and “resptotal.” The element *sampleanswer* is defined to describe the sample answer of a question. The types “question”, “section”, and “total” are used to describe the score of question, section and the total score that defined by questioner, and the types “respquestion”, “respsection”, and “resptotal” are used to describe the score of question, section and the total score getting by respondents. An example is shown below. “Blue” is the sample answer of the question “qu1.” The score of the question is “2”. The score of the question that the respondent gets is “0”, and the respondent also get “0” of total score.

```
<result>
    <marker>Bryan</marker>
    <question>
            <sampleanswer>Blue</sampleanswer>
            <score type="question">2</score>
            <score type="respquestion">0</score>
    </question>
    <score type="resptotal">0</score>
</result>
```

For the difference of question fields, we add a new element which is *formula*. We also add an attribute *lang* for the *formula* element. The element *formula* and its attribute *lang* are defined to describe formulas written by XML-based formula languages. An example is shown below. It is a question which contains a formula written by "MathML".

```
<question>
    <text>
          <formula lang="MathML">
                …
          </formula>
    </text>
</question>
```

At last, to deal with difference of authority, we add two attributes which are *writable* and *readable*. We use four numbers to describe the authority for each part of test. The first number stands for information of test, the second number stands for sample answer, the third number stands for response, the last number stands for the result. 0 stands for no, and 1 stands for yes. An example is shown below. It means that the respondent is only writable for response part, and the respondent cannot read sample answer of the test.

```
<respondent writable="0010" readable="1011" />
```

4 Use Case: An Offline E-testing Environment

4.1 Offline E-testing Environment

The offline e-testing environment is an e-testing environment to provide users with e-testing service which can execute various offline e-testing. E-testing executed offline means that all the devices in the test progress are in a closed network which cannot connect to the Internet. We developed the offline e-testing environment as a web application shown in Fig. 1.

In the offline e-testing environment, there are four kinds of devices. They are the offline e-testing server terminal, the wireless router, the client terminals, and the USB flash memory. All test data should be specified by QSL and imported into the offline e-testing server. The e-testing will be distributed through LAN and wireless router to

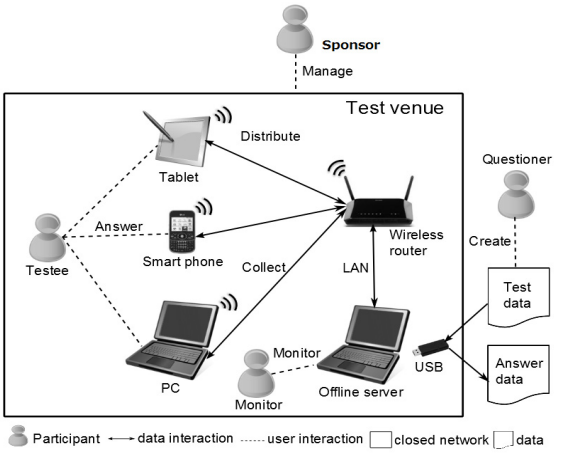

Fig. 1. Offline E-testing Environment

respondents without the Internet. The respondents will answer questions and submit their answers to offline e-testing server also as QSL format with multiple clients, such as smart phone, tablets, PC and so on. Before executing offline e-testing, an operating system and the necessary execution environment for the offline server are set into the USB flash memory as a live boot. Sticking the USB flash memory into a PC and starting the PC, the offline e-testing server can be started.

There are four kinds of participants taking part in the offline e-testing environment. They are sponsor, questioner, monitor, and respondent. The sponsor takes the responsibilities of initiating and organizing an e-testing which contains to specify the participants of the e-testing. The responsibilities of questioner are to designs questions of the e-testing and specify all the questions with QSL. Moreover, the responsibilities of monitor contain that importing test files to the offline e-testing server, managing all test that have been imported, monitoring the state of the e-testing and respondents, collecting and exporting response data of respondents. At last, the respondent takes the responsibilities of answering questions and submitting responses of the e-testing.

At present, the procedure of offline e-testing with the offline e-testing environment is as follows. A monitor imports the test file which has been described by QSL and manages all tests, the monitor monitors the connection state and test state of respondents, and respondents answer questions via web browser. The monitor orders offline e-testing server to integrate answer data automatically and the monitor exports answer files also as QSL format. Our general-purpose offline E-testing environment provides multi-language of user interface. Current environment supports English, Japanese, and Chinese. Our environment supports multi-device, smart phone, tablet, and PC all can be the client devices, and web page will adapt to the screen of device according to resolution of the device. Fig. 2 shows screenshots of the monitoring facility which uses PC as the client device (left), and the answering facility which uses smart phone as the client device (right).

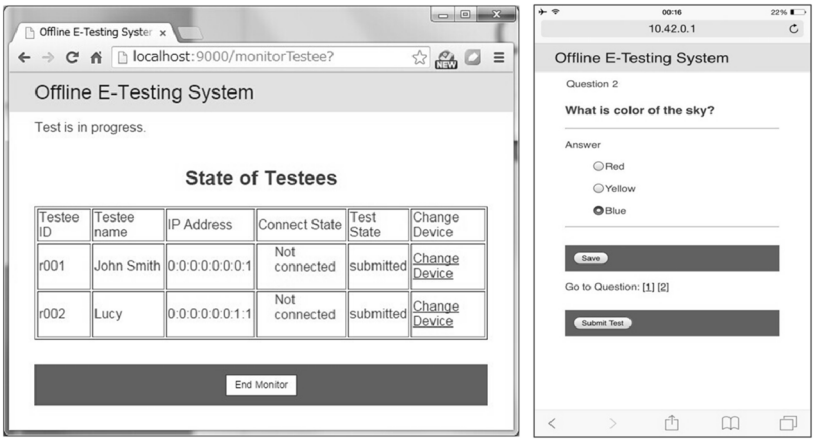

Fig. 2. Monitor Function (left) and Answer Function (right)

4.2 Specification of E-testing and E-testing System

We specified the offline e-testing environment by QSL. An e-testing which executed on the offline e-testing environment has 5 processes. They are creation, distribution, answering, monitor, and collection. All the functions are specified by QSL shown below. Offline e-testing environment distributes test data and collects answer data through wireless router. Before storing answer data into USB flash memory, all the answer data will be integrated by using zip method. For the answering process, respondents can use PC, smart phone, or tablet to answer e-testing. In the process of monitor, e-testing environment will "ping" the IP address of each respondents in order to monitor the connection state of respondents. The environment also monitors the test state of respondents. At last, the authorization of all participants is set in the security part of offline e-testing environment.

```
<qenvironment>
  <qdistribution>
    <func-distribute>
      <method type="offline">USB flash memory</method>
    </func-distribute>
  </qdistribution>
 <qcollection>
    <func-collect>
      <method type="offline">USB flash memory</method>
    </func-collect>
    <func-integrate>
      <method>zip</method>
    </func-integrate>
 </qcollection>
 <qanswer>
```

```
    <func-answer>
      <device>PC</device>
      <device>smart phone</device>
      <device>tablet</device>
    </func-answer>
  </qanswer>
  <qmonitor>
    <func-monitor>
      <method type="ping">IP address</method>
      <method type="common">Test state</method>
    </func-monitor>
  </qmonitor>
</qenvironment>
<qsecurity>
  <authorization/>
    <participants>
      <sponsor writable="1100" readable="1111"/>
      <questioner writable="1100" readable="1100"/>
      <monitor writable="0000" readable="1000"/>
      <respondent writable="0010" readable="1011"/>
    </participants>
  </qsecurity>
```

We also specified an e-testing on the offline e-testing environment. An example is shown below. Creating an e-testing named “An E-Testing Example”. It defines the language of the e-testing which is English, it also defines the start and end time of this e-testing. The e-testing contains one section. In the section, it represents a multiple-choice question. The logic is randomization which is showing options in a random order. The score of the question is 2.

```
<qdata>
  <questionnaire name="q1”>
    <text type="title">An E-Testing Example</text>
    <language>en</language>
    <time type="start">2014-12-08T09:30:00Z</time>
    <time type="end">2014-12-08T11:30:00Z</time>
    <section name=“section1">
      <question name="qu1" type="multiple-choice">
        …
        <logic type=“randomization" />
        <sampleanswer>Blue</sampleanswer>
        <score type="question">2</score>
      </question>
    </section>
  </questionnaire
</qdata>
```

5 Concluding Remarks

We have proposed an extension of QSL for various e-testing and e-testing systems. We also shown an application which is an offline e-testing environment as use case of extended QSL, and shown specifications of the environment and e-testing on the environment. Moreover, we have shown that QSL is also useful format for data exchange in offline e-testing.

In the future, we will continue working on improving QSL for various e-testing and e-testing systems, and consider extending it to contain various e-voting systems owing to significant similarities of contents and structures with e-questionnaire and e-testing.

References

1. AddPoll, http://www.addpoll.com/
2. Constant Contact, http://www.constantcontact.com/
3. Datagle, http://www.datagle.com/
4. Diao Cha Quan (in Chinese), http://www.diaochaquan.cn/
5. eSurv, http://eSurv.org/
6. ExamSoft, http://learn.examsoft.com/
7. Examcoo (in Chinese), http://www.examcoo.com/
8. Mobo Survey, http://www.mobosurvey.com/
9. MySurveyLab, http://enq-maker.com/
10. OCR, http://www.ocr.org.uk/
11. oExam (in Chinese), http://www.orivon.com/
12. OQSS, http://www.oqss.com/
13. PHPEMS Online Testing System (ver. 2.0.1), http://www.phpems.net/
14. ProProfs, http://www.proprofs.com/
15. QuestionPro, http://www.questionpro.com/
16. Suzuki, J., Goto, Y., Cheng, J.: Development of a General-Purpose E-testing Server for Ubiquitous Test. In: Zhang, W. (ed.) Software Engineering and Knowledge Engineering. AISC, vol. 162, pp. 797–803. Springer, Heidelberg (2012)
17. So Jump (in Chinese), http://www.sojump.com/
18. Sogo Survey, http://www.sogosurvey.com/
19. SurveyMonkey, http://www.surveymonkey.com/
20. Tomexam Online Testing System (ver. 2.7), http://www.tomexam.com/
21. Yong Dao Offline Testing (in Chinese), http://www.onlinedown.net/soft/49716.htm
22. Zhou, Y.: QSL Manual, ver. 1.5 (2014), http://www.aise.ics.saitama-u.ac.jp/QSL/
23. Zhou, Y., Goto, Y., Cheng, J.: QSL: A Specification Language for E-questionnaire Systems. In: Proc. Of the 5th IEEE International Conference on Software Engineering and Service Science (ICSESS 2014), Beijing, China, pp. 224–230. IEEE (2014)