

QSL Editor: a structure editor for questionnaire specification language

Amantuer Rewuhan[†] and Yuichi GOTO[†]

Saitama University[†]

1.Introduction

Questionnaire Specification Language (QSL) serves as a formalized specification for specifying various "e-questionnaire, e-testing, and e-voting systems (QTV systems for short)."

QSL is based on Extensible Markup Language (XML).The grammar of QSL is defined by XML Schema. Users want a tool that support to describe and edit QSL specification easily. Without the knowledge about QSL syntax. XML editor is not enough to use it as a QSL Structure Editor for describing QSL specification and editing QSL specification.To propose a tool that could describe and edit QSL specification easily without the knowledge of QSL syntax that we designed the first QSL structure editor. QSL structure editor should provide the stakeholders with an intuitive interface to easily choose the desirable requirements from an exhaustive requirement list and generate the requirement specification.

The purpose of this study is making user could highly improve effectiveness without the knowledge of QSL syntax in making and editing specifications of QTV systems by using QSL structure editor which has an understandable UI and many functions that XML editor doesn't.

2.QSL: questionnaire specification language

2.1.Overview

QSL is an XML-based language that proposed to provide exhaustive requirements list of QTV systems to the stakeholders so that they can design, develop, and evaluate their demanded systems easily.There are 5 kinds of the stakeholders of QTV systems, which are sponsor,

evaluator, executor, respondent, and supporter. It is necessary to promote the effective communications among those stakeholders by a communication tool to easily link each stakeholder's requirements for systems and services of QTV.

2.2.Construction of QSL

QSL provides 93 tags as terminology because QSL is an XML-based specification language.There are two kinds of tags for QSL. Firstly, 54 tags are defined according to entities. Using these tags, we can describe requirements of systems and services of e-questionnaire, e-testing, and e-voting formally. Secondly, 38 tags are defined according to the construction. In addition, QSL provides 52 complex types of the hierarchy of the tags and 34 simple type for the constraint of the tags. Using the tags, we defined QSL schemas to constrain the requirements in a formal way. QSL schema is a collection of requirements formalized by XML schema, and clear definition of the relationship among the requirements.

2.3.Usage of QSL

For using QSL, we must understand the 94 components of QSL. Further more, before using QSL we need to have a deep understanding of 67 QSL schemas. Here is an example could explain how complex QSL is. About question types we could build with these schemas, there are four basic question types and there are several the possible combinations of basic question types. As we can see, it is barely difficult to understand definition of each schema and remember the meaning of each component for normal users. Not to mention how hard to use these schemas to build different kinds of questions

only by reading and editing file.

3.Requirement Analysis

To propose a tool that could describe and edit QSL specification easily without the knowledge of QSL syntax that we designed QSL structure editor. QSL structure editor should provide the stakeholders with an intuitive interface to easily choose the desirable requirements from an exhaustive requirement list and generate the requirement specification. And here are specific requirements of QSL structure editors.

R1:QSL structure editor should provide services to guide the users to choose the corresponding requirement templates according with their suitable identities.

R2:QSL structure editor should provide services to perform the requirement list with a intuitive user interface that hide the code in the background and present the content to the user in more user friendly forms and show guidance according to the suitable terminology.

R3:The editor must provide services to hide or show the requirements, which are logically associated with a chosen requirement.

R4:QSL structure editor should provide services to guide to fill the appropriate values, and to validate and verify whether the filled values are valid or not in conformance with QSL schemas, and point out the invalid values and defined formats.

R5: QSL structure editor should provide services to output QSL format requirement specification replaced by the familiar terminology.

4.Design

In designing QSL structure editor, we pay attention to stakeholders of sponsor, executor and supporter.

Users are organized as a unit of team. A team could have several sponsors, executors and supporters. Sponsor could only see specification files that were made. Executor could create specifications and see specification files that were made. Supporter could create specifications and edit specification files that were made.

According to the type of specification that user chose to create or edit, we provide the questions that user have to answer and provide the suggestion chooses that user could make optionally. Besides creating a xml-tag like file we also provide to create a nature language file that could be understand only by reading an article. Both of these functions were under a easy

understandable and friendly UI.

5.Implementation

5.1 Development environment

We choose to use Vue3 and nodejs and mysql.

5.2 Prototype of QSL editor

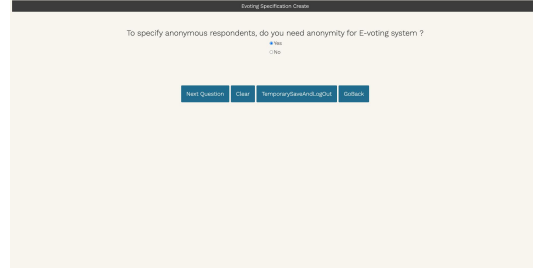


Fig.1.Creating Specification

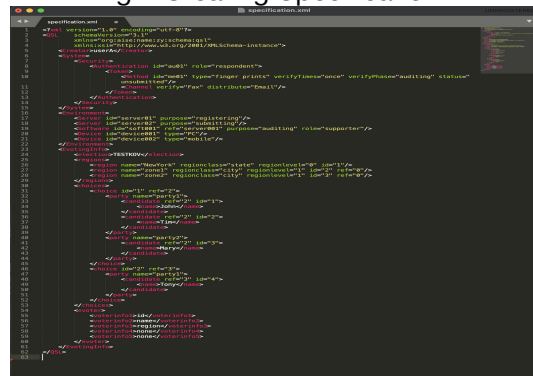


Fig.2. QSL File

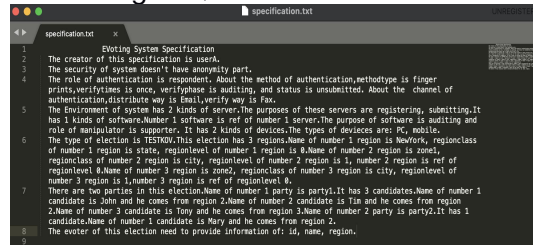


Fig.3. Nature Language File

6.Concluding

Building the first QSL structure editor that we analysis the structure of QSL schema, and get the different requirements of users of QTV systems, finally finished e-voting part of QSL structure editor.

References

- [1]. Yuan. Zhou, Yuichi Goto, and Jingde Cheng: QSL: A Specification Language for E-Questionnaire, E-Testing, and E-Voting Systems, IEICE Transactions on Information and Systems, Vol. E102-D, No. 11, IEICE, Nov. 2019.
- [2]. Yuan Zhou: QSL Manual (ver. 3.1), <http://www.aise.ics.saitama-u.ac.jp/QSL> (accessed Jan. 7. 2021).
- [3]. Vue3: <https://v3.vuejs.org> (accessed Jan. 7. 2021).