AISE

Questionnaire Specification Language (QSL)

Specification Version 3.0

6 July. 2017

Specification URIs

This version:

http://www.aise.ics.saitama-u.ac.jp/qsl/v3.0

Previous version:

http://www.aise.ics.saitama-u.ac.jp/qsl/v2.1

Affiliation:

Advanced Information Systems and Engineering Lab, Saitama University, Japan

QSL Members:

Y. Zhou (shuugen@aise.ics.saitama-u.ac.jp), D3
Z. Wang, M.S. graduated in 2015
D. Matsuura, B.S. graduated in 2016

Author:

Y. Zhou (shuugen@aise.ics.saitama-u.ac.jp), D3

Editors:

Y. Zhou (shuugen@aise.ics.saitama-u.ac.jp), D3
H. Gao (gaohongbiao@aise.ics.saitama-u.ac.jp), Assistant Professor

Abstract:

This document describes the background and purpose of Questionnaire Specification Language (QSL), presents the grammar of QSL in XML Schema including the whole elements, attributes, and the structure of it, and shows many specifications as templates. We devote to continuously improve QSL better and better since 2013 till now.

Status:

This document was last revised on 6 July, 2017.

Contents

Figures

1 Introduction

1.1 Terminology

QSL Specification: the QSL manual to guide user how to use QSL to specify a complete QSL-format specification.
QSL-format Specification: the specification written by user who use QSL apps or other XML editors based on QSL schema.
QSL Schema: QSL grammar for supporting QSL, as the core component of a QSL specification.

1.2 Background

AISE members, to design a formalized method for various e-questionnaire, e-testing, and e-voting systems. The mission statement is, in part, to:
"propose a standard and formalized tool to solve the problem among the stakeholders of various e- questionnaire, e-testing, and e-voting systems …"
The original objective in 2013 was to introduce a uniform and unique method to specify systems involved in the processes of e-questionnaire, e-testing, and e-voting, including data exchange. The overall focus today provides a standard that is:

- **Multinational:** Our focus is to have standards that can be adopted globally.
- **Flexible:** Effective across the various questionnaire, testing, and polling regimes.
- **Multilingual:** Flexible enough to accommodate the various languages and dialects and vocabularies.
- **Adaptable:** Resilient enough to support questionnaires, examinations, and elections in both the private and public sectors.
- **Secure:** Able to secure the relevant data from ant attempt, as appropriate to the different requirements of various e-questionnaire, e-testing, and e-voting rules.

QSL is a questionnaire specification language for various e-questionnaire, e-testing, and e-voting systems described as XML schemas in a QSL structure. QSL is flexible enough to be used for questionnaire, testing, and voting that is primarily paper-based or that are fully e-enabled.
At present, QSL includes specifications for:

- Phases: Setting up, distributing, answering, submitting, analyzing, marking, monitoring, etc.
- E-paper information
- Respondent Authentication including respondent registration
- Counts, statistics, and results

- Candidate nomination, response to nomination
- Environment information including software and hardware

This document and its schemas and other templates do not claim to satisfy the final requirements of any and all registration systems. The specification represents our best current efforts, knowledge and experience with e-questionnaire, e-testing, and e-voting systems since 2007. It is incumbent on users of this document to identify and requirement gaps, mistakes, inconsistencies or missing data and to propose corrections or enhancement to AISE.

1.3 Overview of the Document

To help establish context for the specifications contained in the XML Schema that make up QSL, the first questionnaire specification language for e-questionnaire, e-testing, and e-voting systems. AISE also designed the structure of it. This structure identifies how to combine each elements to obtain various e-questionnaire, e-testing, and e-voting systems. In this document, we present a lot of specifications as templates for references and reusability, and describe how QSL can be used to standardize the data exchanged.

1.4 Changes in this Version

The changes from QSL v 2.1 that this new version introduces are as follows:

- Extract completexType and simpleType;
- Specify the attributes values in simpleType for avoiding repeating

1.5 Advantage of using QSL

The question usually asked about why any particular standard should be used is "what advantages will it bring me?" In addition to supporting trusted e-questionnaire, e-testing, and e-voting systems, the benefits of adopting QSL are as follows:

For Sponsors:

- More choice of products and suppliers;
- Clearly mind and unambiguous;

- Supports scalability, transparency, and data reusability;
- Provides basis for make clearly work division, and responsibility;

For Executor:

- More choice of survey sheets;
- Clearly work division, and responsibility;
- Supports scalability, transparency, and data reusability;

For Respondents:

- Supports trustworthiness of systems;
- Support security of e-questionnaire, e-testing, and e-voting;

For Suppliers:

- Easily communicate with sponsors;
- Reduced development costs;
- Accommodates future changes more easily;
- Common core but allows local extension;

1.6 How to use QSL

As a specification language, QSL has had to meet various range of e-questionnaire, e-testing, and e-voting requirements. Therefore it may need to be tailored for specific scenarios and meet specific rules and practices.
First using the QSL grammar (an XSD file) and an xml-editor to translate it and to build an xml file according QSL grammar. You can also use a specification template and change it by your own mind but do not need to break up the QSL grammar definition.
You can also use our templates we provided, to change it according to the QSL grammar. You can refer to the different scenarios we described.

There are two kinds of files, you can download, are listed as follow:

- QSL Grammar (QSL Schema in XSD);
- Template (Specification specified by QSL in XML);

A recommendation of tool for supporting to edit XML-based QSL Schema, we used XML Spy[1] to design it, and edit it based on QSL Schema to create specifications. In future, we are planning to implement a QSL structure editor to edit specification in a QSL-format, and a QSL generator system to automatically generating e-questionnaire, e-testing, and e-voting systems.

[1] XML Spy Editor is the industry's best-selling XML editor for modeling, editing, transforming, and debugging XML-related technologies. It offers the world's leading graphical schema designer, a code generator, file converters, debuggers, profilers, full database integration.

1.7 Specifications Using QSL

Two examples of the specifications using QSL are shown below.

In the first snippet, we define an English QSL specification for an e-questionnaire based on QSL grammar version 3.0. In this specification, we specify a questionnaire, which has two sections. Each section has a question. The questionnaire has title, thank you page, and inserts a theme picture. Two questions are necessary to response. The first question in the first section is a multiple selection question, which is limited as at least to choose one answer. The second question is a matrix question with a single choice. There is a complex logic relationship, which is to extract the values of the checked answers in the first question, and fill the values into the second question. If the answer is not checked, the piping value is a void value.

```
<?xml version=″1.0″ encoding=″UTF-8″?>
<QSL
schemaVersion=″3.0″
xml:lang=″en″
xmlns=″http://www.aise.ics.saitama-u.ac.jp/qsl″
xmlns:xsi=″http://www.w3.org/2001/XMLSchema-instance″>
    <PaperSheet>
        <Paper id=″paper001″ type=″questionnaire″>
            <Description id=″des001″ type=″header″ value=″Survey on sweets″/>
            <Description id=″des002″ type=″paragraph″ value=″research your favorite
sweets″/>
            <Media id=″med001″ type=″img/jpg″ src=″pic/sweets.jpg″>
            <Section id=″sec001″>
                <Description id=″des004″ type=″paragraph″ value=″favorite sweet
part:″/>
                <Question id=″ques001″ type=″selection″ isMandatory=″yes″>
                    <Limitation minOccur=″1″ maxOccur=″3″/>
                    <Description id=″des005″ type=″paragraph″ value=″what is (are)
your favorite sweet(s)″/>
                    <Answer id=″ans001″>
                        <Description id=″des006″ type=″paragraph″ value=″ice
cream″/>
                        <Media id=″med002″ type=″img/jpg″ src=″pic/ice.jpg″>
                    </Answer>
                    <Answer id=″ans002″>
                        <Description id=″des007″ type=″paragraph″ value=″cake″/>
                    </Answer>
                    <Answer id=″ans003″>
                        <Description id=″des008″ type=″paragraph″ value=″pie″/>
                    </Answer>
                </Question>
                <Description id="des005" type="break"/>
            </Section>
            <Section id="sec002">
                <Description id="des009" type="paragraph" value="favorite taste
part:"/>
                <Question id="ques002" type="Matrix" isMandatory="yes">
                    <Description id="des010" type="paragraph" value="what is (are)
your favorite taste(s) for ${'pip001'} ${'pip002'} ${'pip003'}"/>
                    <Limitation minOccur="1" maxOccur="1"/>
                    <Answer id="ans004" Grid.Row="0">
                        <Description id="des011" type="paragraph" value="ics
cream"/>
                    </Answer>
                    <Answer id="ans005" Grid.Row="1">
                        <Description id="des011" type="paragraph"
value="${'pip001'}"/>
                    </Answer>
                    <Answer id="ans006" Grid.Row="2">
                        <Description id="des012" type="paragraph"
value="{'pip002'}"/>
                    </Answer>
                    <Answer id="ans004" Grid.Column="0">
                        <Description id="des013" type="paragraph"
value="{'pip003'}"/>
                    </Answer>
                    <Answer id="ans005" Grid.Column="2">
                        <Description id="des014" type="paragraph" value="melon"/>
                    </Answer>
                    <Answer id="ans006" Grid.Column="3">
                        <Description id="des015" type="paragraph" value="grape"/>
                    </Answer>
                </Question>
            </Section>
            <Description id="des003" type="paragraph" value="Thank you!"/>
            <Logic ref="paper001">
                <Route id="rou001" type="piping">
                    <Condition answerId="ans001" isChecked="yes"/>
                    <Action ref="des006" isFilled="yes" to="${'pip001'}"/>
                <Route>
                <Route id="rou002" type="piping">
                    <Condition answerId="ans002" isChecked="yes"/>
                    <Action ref="des007" isFilled="yes" to="${'pip002'}"/>
                <Route>
                <Route id="rou003" type="piping">
                    <Condition answerId="ans003" isChecked="yes"/>
                    <Action ref="des008" isFilled="yes" to="${'pip003'}"/>
                <Route>
            </Logic>
        </Paper>
        <Setting ref="paper001">...</Setting>
    </PaperSheet>
</QSL>
```

The second snippet specifies a general-purpose offline e-testing system we developed. In this QSL-format specification, it shows the specification of the system construction. The specification lists the functions in each phase. Our system can import qsl-based exam paper file. This file is distributed use a USB flash memory. The respondent login and is verified to ensure whether he is eligible or not. And the respondent answers the questions, and submits his response. During the submitting phase, the monitor can monitor the whole phase by ping the respondent' IP address and monitor their test state. After submitting, the monitor can use the USB to store the collected responses by a zip method. The marker can mark the responses. In addition, the corresponding components are listed such as servers, software, devices, and database.

```
<System>
    <Phase>
        <SettingUp>
            <Function id="func001" type="importing" format="qsl" file="paper"/>
            ...
        </SettingUp>
        <Registering>
            <Function id="func002" type="authenticating"/>
        </Registering>
        <Distributing>
            <Function id="func003" type="distributing">
                <Channel type="usb"/>
            </Function>
        </Distributing>
        <Submitting>
            <Function id="func004" type="answering"/>
            <Function id="func005" type="monitoring">
                <Method type="ping" scoping="IP address"/>
                <Method type="common" scoping="status"/>
            </Function>
            <Function id="func006" type="submitting"/>
            <Function id="func007" type="interrupting"/>
        </Submitting>
        <Collecting>
            <Function id="func008" type="collecting">
                <Channel type="usb"/>
            </Function>
            <Function id="func009" type="integrating" method="zip"/>
        </Collecting>
        <Marking>
            <Function id="func010" type="marking"/>
        </Marking>
        ...
    </Phase>
    <Component>
        <Server id="ser001" purpose="registering"/>
        <Server id="ser002" purpose="submitting"/>
        ...
        <Software id="soft001" serverId="ser002" purpose="submitting">
            <Solution type="browser" name="chrome"/>
            <Participant role="respondent"/>
        </Software>
        ...
        <Device id="dev001" softwareId="soft001" type="PC">
            <Participant role="sponsor"/>
            <Participant role="questioner"/>
            <Participant role="marker"/>
        </Device>
        <Device id="dev002" type="access point"/>
        <Device id="dev003" type="usb">
            <Participant role="questioner"/>
            <Participant role="marker"/>
        </Device>
        <Device id="dev004" type="hub">
            <Participant role="questioner"/>
        </Device>
        <Device id="dev005" type="tablet">
            <Participant role="respondent" id="res001"/>
        </Device>
        ...
        <DataBase id="db001" serverId="ser002" type="response"/>
        ...
    </Component>
</System>
```

2 Requirement

2.1 Stakeholders and Their Inconvenience

There are five kinds of stakeholders around e-questionnaire, e-testing, and e-voting systems, which are:

- **Sponsor** who organizes the event;
- **Executor** who perform task to help carry out the event that perhaps contains questioner, monitor, analyst, etc.;
- **Respondent** who is an investigated target;
- **Supplier** who provides the tech-support services that perhaps contains software developer, hardware vendor, software tester, etc.;
- **Government** concerns laws and standards;

The problems among those stakeholders and various e-questionnaire, e-testing, and e-voting systems are illustrated below. Firstly, when a sponsor, an executor uses an existing system to do a small event of e-questionnaire/e-testing/e-voting, if the system cannot satisfy him to provide desirable requirements, he needs to use other systems. It is a big burden to learn how to use a new system if they do not know different usages. In addition, it is inconvenient to reuse data because there is no unified method and interface for data interchange among the systems. It contains reusability of the respondent's information. Secondly, it divided up the roles of stakeholders in a large event of e-questionnaire/e-testing/e-voting. A sponsor orders a new ad-hoc system. 1) It is difficult to communicate between sponsor and supplier. The sponsor is difficult to clearly describe specifications for the new system, since he does not clear what are necessary in detail. Supplier is difficult to understand the specifications for the system and he is not a specialist in sponsor's field. 2) It is difficult to supply a secure tech support if sponsor cannot standardize the contractual relations, clear work division, and defines responsibility with executor. 3) Different regions have different laws and rules, and the government still updates them. The different laws and rules effect on the event and even the system design.

It is a lack of a unified method to solve the above-mentioned problems, that is a specification language for specifying e-questionnaire, e-testing, and e-voting systems, we proposed it.

2.2 Challenge and Scope

Our goal has been to design an Questionnaire Specification Language (QSL) for specifying various e-questionnaire, e-testing, and e-voting systems to all the stakeholders within the event of it. It contains a set of data definitions, control, and abstract structure specified as a set of XML Schemas during all the phases (state) of the events of e-questionnaire, e-testing, and e-voting.

QSL is meant to assist and enable phases and does not need any changes to traditional methods of executing e-questionnaire, e-testing, and e-voting. The extensibility of QSL makes it possible to adjust to various e-questionnaire, e-testing, and e-voting without affecting the phases. Conceptually QSL simply enable the data exchange among stakeholders and the systems in a standardized way.

The solution outlined in this document is non-proprietary and will work as a template for any scenario of e-questionnaire, e-testing, and e-voting using their systems for all or part of the process. The objective is to introduce a uniform method to allow the systems to interact with each other.

2.3 Design Philosophy

"How to specify various e-questionnaire, e-testing, and e-voting systems? " That is the first and the most important question in our mind when we decided to design QSL. **Mendeleyev's periodic table**[1] enlightens us, are there some primitive elements arranged by some certain rules for the most part, and this arrangement revealed the periodicity of something? The design of QSL is still with our first heart.

According to a lot of investigations of existing e-questionnaire, e-testing, and e-voting systems, the similarities and differences are found.

Firstly, we list six aspects of similarities as follow:

- **Phase:** setting up, distributing, submitting, collecting, and counting;
- **Security:** consists of authentication, anonymity, and authority;
- **Paper:** question sheet consists of section, question, answers;
- **Setting:** setting for each phase, which set up by questioner;
- **Component:** consists of server, software, device, database, interface;
- **Participant:** sponsor, questioner, analyst, monitor, and respondent;
- **Data:** consists of participant data, result data, report data, etc.

[1] The periodic table is a tabular arrangement of the chemical elements, ordered by their atomic number (number of protons), electron configurations, and recurring chemical properties.

Secondly, we list the differences as follow:

- **E-Questionnaire:**
  - **Question Type:** multiple choice, open-ended text, matrix, and ranking;
  - **Logic:** skipping, piping, extraction, and randomization;

- **E-Testing:**
  - **Participant:** marker;
  - **Phase:** marking;
  - **Paper:** sample answer, score, much more interface and question types for much more professional field, such as mathematical formula, periodic table of chemical elements, etc.;

- **E-Voting:**
  - **Security:** authentication, and anonymity;
  - **Phase:** auditing;
  - **Component:** certification server, auditing software;
  - **Participant:** candidates;

After a sketchy analysis, the differences are also classified into the similarities, and totally categorized into groups, which are listed in the appendix. Obviously, the list of Primitive elements is much more elaborated and deliberated, because Primitive elements are the train of thought of designing QSL (but not the most important of this manual).

2.4 Documentation Set

To reach our goal, we defined:

- **Specification:** This document. This introduces the QSL structure and QSL grammar from a complete manual for specifying e-questionnaire, e-testing, and -e-voting systems by defining the systematic functions, process, paper sheet, setting, data information, security of content, and describe how uses them encoded as QSL schemas.
- **QSL Schemas:** This consists of a library of XML schemas used in QSL. The XML schemas defined the formal structure of the system, security, paper sheet, data that needs to be specified and constructed through an complete event (each phase) of e-questionnaire, e-testing, and e-voting systems.
- **Templates:** for schemas of each e-questionnaire system, e-testing system, e-voting system in four parts (security, paper sheet, system, and data) providing a template that facilitated generation of the main schema structure. This aims to reduce implementer's costs of development and integration, as well as for sponsor and questioner to create the events.

3 QSL Structure

QSL structure is relatively robust, extend and upgrade easily without changing the whole configuration. QSL structure has three layers. In the innermost layer, QSL defines core elements. Specifying any e-questionnaire, e-testing, and e-voting system must specify all the core elements. The core element consists of the combinations of the elements in the middle layer. The elements in middle layer are called sub-core elements. In the outermost layer, there are three isolated ranges, which are for e-questionnaire, e-testing, and e-voting, respectively. The elements in this layer are called extended elements. For instance, if a user wants to specify an e-questionnaire system, he should specify all the elements in the innermost and the middle layer, and specify all or part of the elements for e-questionnaire in the outermost layer.

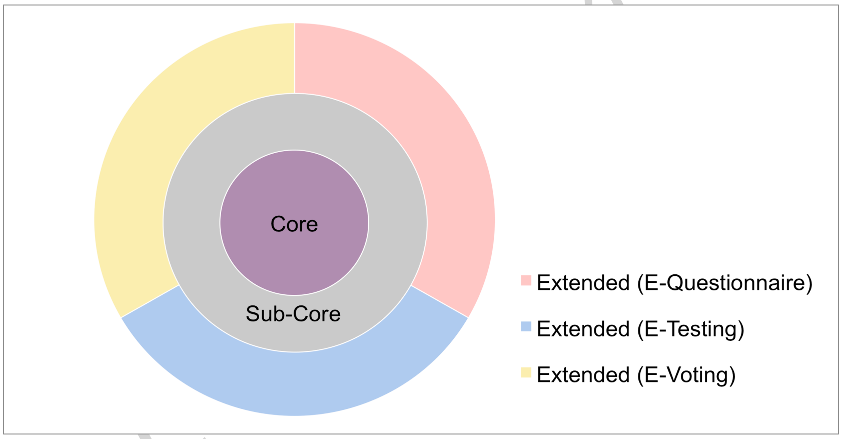


Fig. 3-1 QSL Structure

On account of combinations of sub-core elements, and extended elements, we use three-digit to mark the elements. In the hundred's place, 1, 2, 3, 4, and 5 stands for core element, sub-core element, and extended element for e-questionnaire, e-testing, and e-voting, respectively. The list of the elements with the three-digit numbers is shown below. In the ten's place, 0 stands for a special mark. The elements are associated with the namespace defined using QSL (we mark its code as 100). As the configuration of core elements, it gives a combination relationship of sub-core elements. In addition, some major elements for constructing sub-core elements and extended elements are shown below.

- Core
  - QSL (100)
  - Security (110)
  - System (120)
  - PaperSheet (130)

  - Data (140)
- Sub-core
  - Phase (210)
  - Paper (220)
  - Setting (230)
  - Component (240)
  - Participant (250)
  - Response (260)
  - Report (270)
- Extended for E-Q
  - Logic (310)
- Extended for E-T
  - Marker (410)
  - Score (420)
  - Sample (430)
  - Formula (440)
- Extended for E-V
  - Auth (510)
  - Anonymity (520)
  - Auditing (530)
  - Candidate (540)

4 QSL Grammar Outline

4.1 Structure

The Questionnaire Specification Language specification defines a vocabulary and divided messages. Thus similar items of e-questionnaire, e-testing, and e-voting are defines as elements in the core. The core also contains data definition so that be used in data exchange. Each message is specified and defined with a separate schema document, and can be combined.

As we introduced in pre-chapter, sub-core elements and extended elements are defined as messages with its unique Ids. According to the combination of messages, users can get the data what they want. Obviously, It is order by a rule we gave to. The rule is the core elements.

4.2 Viewing Schemas

QSL grammar is supplied as xml documents. For viewing the structure of it, we recommend to use an XML editor, such as XML spy. Alternatively, if you read a text document with a graphical display, it is easy to refer to.

4.3 Schema Diagrams

The diagram below represents a simple schema in a XML Spy 2011 style. The root element of an instance described by this schema is the element *A*. The content model of this element is a sequence of the elements *B*, *D* and *E*. The element *B* is of complex data type *Bstructure*. This contains a choice of either element *C* or element *F*. Element C is a restriction of another complex data type *Cstructure*. In this case, the restriction is to forbid the use of the element *G* (which is defined in *Cstructure* as optional). The other elements allowed are *H*, which can appear any number of times (but must appear at least once), and *I*, which can appear up to three times (or not at all). Element *D* is optional, and of data type *Dstructure*. This has a content model requiring all of elements *J* and *K*, which are both of type *xs:string*. Finally, element *E* is of simple data type *Etype*, which is restricted from the *xs:NMTOKEN* data type by only allowing the values 'yes' and 'no'.

It is important to remember that these diagrams do not include any attributes. In this document, these are shown in tables below the diagrams.

The full schema is shown below the diagram.

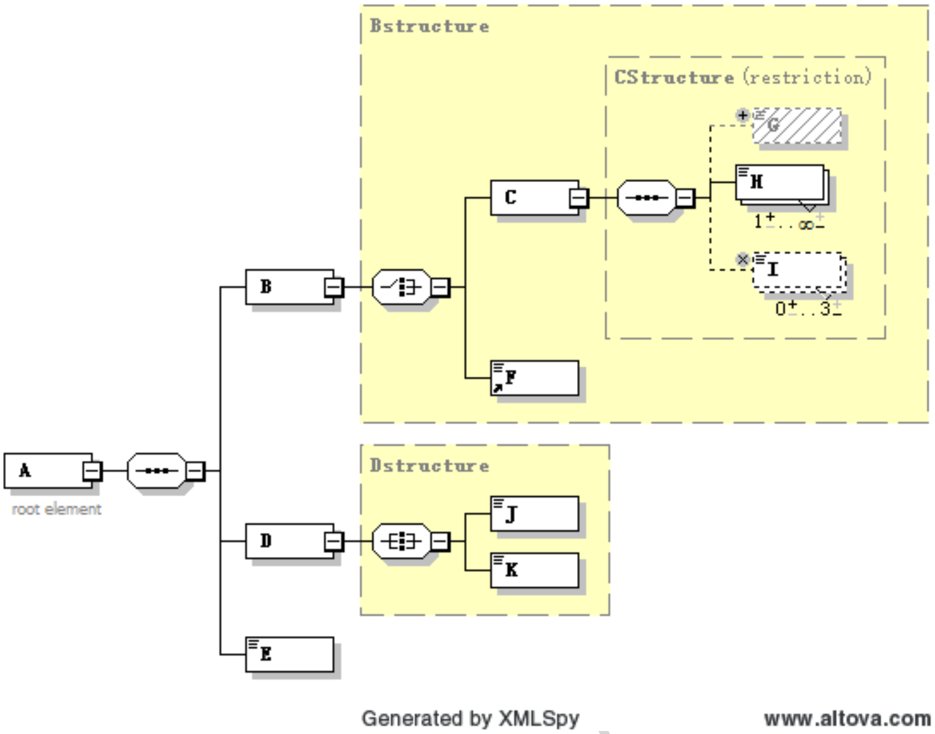


Fig. 4-1 Sample Schema Diagram

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2011 (http://www.altova.com) by Trinity (AISE Saitama) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="A">
        <xs:annotation>
            <xs:documentation>root element</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="B" type="Bstructure"/>
                <xs:element name="D" type="Dstructure"/>
                <xs:element name="E">
                    <xs:simpleType>
                        <xs:restriction base="xs:NMTOKEN">
                            <xs:enumeration value="yes"/>
                            <xs:enumeration value="no"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="B" type="Bstructure"/>
    <xs:element name="C"/>
    <xs:element name="D"/>
    <xs:element name="E"/>
    <xs:element name="F" type="xs:string"/>
    <xs:element name="G"/>
    <xs:element name="H"/>
    <xs:element name="I"/>
    <xs:element name="J"/>
    <xs:element name="K"/>
    <xs:complexType name="Bstructure">
        <xs:choice>
            <xs:element name="C">
                <xs:complexType>
                    <xs:complexContent>
                        <xs:restriction base="CStructure">
                            <xs:sequence>
                                <xs:element name="H" type="xs:string" maxOccurs="unbounded"/>
                                <xs:element name="I" type="xs:date" minOccurs="0" maxOccurs="3"/>
                            </xs:sequence>
                        </xs:restriction>
                    </xs:complexContent>
                </xs:complexType>
            </xs:element>
            <xs:element ref="F"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="Dstructure">
        <xs:all>
            <xs:element name="J" type="xs:string"/>
            <xs:element name="K" type="xs:string"/>
        </xs:all>
    </xs:complexType>
    <xs:complexType name="CStructure">
        <xs:sequence>
            <xs:element name="G" type="xs:positiveInteger" minOccurs="0"/>
            <xs:element name="H" type="xs:string" maxOccurs="unbounded"/>
            <xs:element name="I" type="xs:date" minOccurs="0" maxOccurs="3"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

4.4 Namespaces

The QSL schemas are associated with the namespace: http://www.aise.ics.saitama-u.ac.jp/qsl. This is defined using the prefix qsl. The XML schema namspace http://www.w3.org/2001/XMLSchema is identified by the prefix xs and the XML schema Instance namespace http://www.w3.org/2001/XMLSchema-instance by the prefix xsi.
schema version is 3.0. This specification is an English version.
Such messages start with a QSL element, such as:

```
<?xml version="1.0" encoding="UTF-8"?>
<QSL
schemaVersion="3.0"
xml:lang="en"
xmlns="http://www.aise.ics.saitama-u.ac.jp/qsl"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Security>...</Security>
</QSL>
```

4.5 Conventions

Within this specification, the following conventions are used throughout:

- Diagrams are shown as generated by XML Spy 2011 which was also used to generate the schemas and samples. These diagrams show element content, but not attributes;
- Elements and attributes in schemas are identified by partial Xpath expressions. Enough of a path is used to identify the item without putting in a full path.

5 QSL Core Component

5.1 Overview

The following schema component are defined in the QSL schema:

Elements		Complex Types	Simple Types
Accuracy	Monitor	AlignmentStructure	AddressType
Action	Nominating	AnonymityStructure	AuthType
Alignment	Number	AnswerStructure	BrowserType
Analysing	Numbering	AuthenticationStructure	ChannelType
Analyst	Paper	AuthorityStructure	ClientSideSolutionType
Anonymity	PaperSheet	BiometricStructure	DataType
Auditing	Participant	ComponentStructure	DescriptionType
Auditor	Phase	DataStructure	DeviceType
Authentication	Proposal	DatabaseStructure	EmailType
AutoSaving	QSL	DescriptionStructure	FieldType
Biometric	Questioner	DeviceStructure	FormatType
Candidate	Quota	FunctionStructure	FunctionType
Channel	Ratio	InterfaceStructure	GenderType
Checking	Registering	LimitationStructure	IdType
Collecting	Reminder	LogicStructure	LanguageType
Component	Reply	MediaStructure	LimitationType
Condition	Report	PaperStructure	LogicType
Counting	Respondent	PaperSheetStructure	MediaType
Data	Response	ParticipantStructure	NameType
Database	Route	ParticipantInfoStructure	OrderType
Description	Sample	PhaseStructure	OSType
Device	Score	QSLStructure	PaperType
Distributing	Secrecy	QuestionStructure	ParticipantType
Distribution	Security	ReportStructure	PhaseType
Function	Server	ResponseStructure	ProgrammingLangType
Interface	Setting	SecrecyStructure	QuestionType
Interval	SettingUp	SectionStructure	ReminderType
Language	Software	SecurityStructure	ReportType
Limitation	Sponsor	ServerStructure	SystemType
Logic	Statistic	SettingStructure	TelNumType
Marker	Submitting	SoftwareStructure	TimeType
Marking	System	SystemStructure	YesNoType
Media	Time	TokenStructure	
Method	Token		

5.2 Simple Types

The simple types are included here with their base types and any restriction applied.

5.2.1 AddressType

xs:token with restriction.
Restriction: xs:maxLength: 255
This type is a simple definition of an address.

5.2.2 AuthType

xs:token with restriction.
Restriction: xs: enumeration:
This type is a simple definition of authentication methods, which can be classified into 3 types.

type	**values**
Secrecy	password, randomized pwd
Token	ID card, job card, library card, roll card
Biometric	finger prints, iris, face recognition, signature, DNA
--	others

5.2.3 BrowserType

xs:token with restriction.
Restriction: xs: enumeration: IE, firefox, sarfi, chrome, others
This type is a simple definition of browser. Values are IE, firefox, sarfi, chrome, others.

5.2.4 ChannelType

xs:token with restriction.
Restriction: xs: enumeration: SMS, WAP, digital TV, Internet, Intranet, Kiosk, postal, telephone, digital storage device, paper, fax, email, website, abroad postal, abroad electronic, abroad other, N/A, other
This type exists to hold the possible enumerations for the channel through which

a paper and a register-sheet are submitted, a response is collected. SMS is the short message service (text message). WAP is the wireless access protocol. If other is used, it is assumed that those managing the event will have a common understanding of the channel in use.

5.2.5 ClientSideSolutionType

xs:token with restriction.
Restriction: xs: enumeration: web browser, fat-client, thin-client
This type is a simple definition of browser. Web browser approach is only used to establish the link to run on the submitting server. Fat-client approach is rich of security functionality and cryptographic algorithm that the client side software needs to be installed and executed on the submitting devices. Thin-client approach is a mix of web browser and fat-client that implements a java applet running in he web browser.

5.2.6 DataType

xs:token with restriction.
Restriction: xs: enumeration: participant, response, report, setting, paper, others
This simple type is used to define the data type for combinations. Its values are as same as the child elements of DataStructure in complex type.

5.2.7 DescriptionType

xs:token with restriction.
Restriction: xs: enumeration: header, paragraph, break
This type is a simple definition for description in paper design. This allows that paper, section, question, and answer have herder as a title, paragraph as a explanation, and a break for dividing pages. Considering the lightweight arrangement, which questioner wants, it is designed like this.

5.2.8 DeviceType

xs:token with restriction.
Restriction: xs: enumeration: PC, PDA, smart phone, tablet, fable, others
This type is a simple definition for cilent-side device.

5.2.9 EmailType

xs:token with restriction.
Restrictions: xs: maxLength: 129
xs:pattern: [^@]+@[^@]+
This type is a simple definition of an email address, pending a more complete description that is widely accepted in government. It allows any characters except the @ symbol, following by an @ symbol and another set of characters.

5.2.10 FieldType

xs:token with restriction.
Restrictions: xs: enumeration: char, varchar, number, date, int, float, double, decimal, longtext
This type is a simple definition of the field type of database. It is an optional simple type.

5.2.11 FormatType

xs:token with restriction.
Restrictions: xs: enumeration: csv, excel, qsl, xml, word, spss, others
This type is a simple definition of the format type for export and import.

5.2.12 FunctionType

xs:token with restriction.
Restrictions: xs: enumeration: gap analysis, trend analysis, comparison analysis, basis analysis, report, answer, submit, monitor, ping, integrate, back, internalStop, authenticate, login, monitor, create, edit, copy, import, export, delete, search, generate tokens, distribute, collect, mark, blind mark
This type is a simple definition of functional actions. Each value is defined as a verb easily taken from literality. The values need to be specially notified are internalStop (stop some action in the midway), blind mark (usually in e-testing for a teacher to mark the responses of anonymous respondents), generate tokens (automatically generate participants (especially respondents) admission ticket in an exam), ping (ping the IP address and the status of respondents), integrate (integrate some divided information like each respondent's response, etc.), monitor the states.

5.2.13 GenderType

xs:token with restriction.
Restriction: xs: enumeration: male, female, unknown
The gender of participant. Options are male, female, or unknown (is not allowed in all contexts).

5.2.14 IdType

xs:NMTOKEN with restrictions.
Restrictions: xs: maxLength: 10
xs: minLength: 4
xs: pattern: prefix +[0-9]*
This type is a simple definition of QSL elements' Id. To distinguish different elements, it allows the pattern value to add prefixes. The representative prefixes concern the PaperType and ParticipantType.

Element	Pattern	Element	Pattern	Element	Pattern
Paper	p[0-9]*	Section	s[0-9]*	Question	q[0-9]*
Answer	a[0-9]*	Description	de[0-9]*	media	me[0-9]*
Sponsor	sp[0-9]*	Respondent	re[0-9]*	Questioner	qu[0-9]*
Analyst	an[0-9]*	Marker	ma[0-9]*	Monitor	mo[0-9]*
Auditor	au[0-9]*	Operation	op[0-9]*	Route	ro[0-9]*
Condition	co[0-9]*	Action	ac[0-9]*	Software	so[0-9]*
Server	sr[0-9]*	Database	db[0-9]*	Method	me[0-9]*
Interface	in[0-9]*	Statistic	st[0-9]*		

5.2.15 LanguageType

xs:language.
Declaration of the type of language used in e-questionnaire, e-voting, and e-testing systems.

5.2.16 LimitationType

xs:token with restriction.
Restrictions: xs: enumeration: alphabet, figure
This type is a simple definition of limitation type.

5.2.17 LogicType

xs:token with restriction.
Restriction: xs: enumeration: skipping, piping, extraction, randomization
This type is a simple definition of logic type. Its values are basic logic type. As to the combination of them are introduce in Chapter 8 in details. This manual takes a complete chapter to explain it. It is a difficult part in this manual.

5.2.18 MediaType

xs:token with restriction.
Restriction: xs: enumeration: video/ogg, video/mp4, audio/ogg, audio/mpeg, img/jpg, img/png, img/gif
This type is a simple definition of media types. It is designed by referring to HTML's tags <video> and <img>.

5.2.19 NameType

xs:token with restrictions.
Restrictions: xs: maxLength: 129
xs: maxLength: 2
The name of participant and affiliation.

5.2.20 OrderType

xs:token with restriction.
Restrictions: xs: enumeration: ending, descending
This type is a simple definition of order type of settings.

5.2.21 OSType

xs:token with restriction.
Restrictions: xs: enumeration: Windows XP, Win7, Windows 2003, Windows Vista, Win8, Win10, ios 10.0, ubuntu 14.04, Andirod, others
This type is a simple definition of operating systems.

5.2.22 PaperType

xs:token with restriction.
Restrictions: xs: enumeration: paper, section, question, answer
This type is a simple definition of contents of the main paper structure.

5.2.23 ParticipantType

xs:token with restriction.
Restrictions: xs: enumeration: sponsor, questioner, respondent, marker, auditor, analyst
This type is a simple definition of participant roles.

5.2.24 PhaseType

xs:token with restriction.
Restrictions: xs: enumeration: setting up, submitting, registering, collecting, analysing, marking, auditing
This type is a simple definition of phase.

5.2.25 ProgrammingLangType

xs:token with restriction.
Restrictions: xs: enumeration: java, C#, vb, php, python, c, others
This type is a simple definition of programming languages.

5.2.26 QuestionType

xs:token with restriction.
Restrictions: xs: enumeration:
This type is a simple definition of question types. It is much more complex because until now there are over 36 question types in the world. Much more detailed information is defined in Chapter 7.

5.2.27 ReminderType

xs:token with restriction.
Restriction: xs: enumeration: password, token type, answer notification, answer finished, close submitting time, declaration time, report
This type is a simple definition to notify what contents.

5.2.28 ReportType

xs:token with restriction.
Restriction: xs: enumeration: basic, gap, trend
This type is a simple definition of analysis types. Basic type is a percentage report. Gap type is the comparison report. Trend type is always used in e-questionnaire for trend report.

5.2.29 SystemType

xs:token with restriction.
Restriction: xs: enumeration: system
If it need to be divided into 3e systems for extension.

5.2.30 TelNumType

xs:token with restriction.
Restrictions: xs: maxLength: 35
xs: minLength: 1
xs: pattern: \+?[0-9\(\)\-\s]{1,35}
Since this must allow for various styles of international telephone number. The pattern has been kept simple. This allows an option plus sign, then between 1 and 35 characters with a combination of digits, brackets, the dash symbol and white space.

5.2.31 TimeType

xs:dateTime.
This type abides by the format of XML Schema's xs:dateTime. The definition of the date and time uses the following format: YYYY-MM-DDThh:mm:ss Z(YYYY

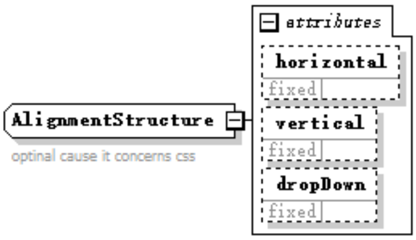
stands for year, MM stands for month, DD stands for day, T stands for the start of necessary time, hh stands for hour, mm stands for minute, ss stands for second, Z stands for the time zone identified by UTC).

5.2.32 YesNoType

`xs：token` with restrictions.
Restrictions: `xs：enumeration`: yes, no
This is a simple enumeration of yes and no and is used for elements and attributes than can only take these values.

5.3 Complex Types

5.3.1 AlignmentStructure



Generated by XMLSpy www.altova.com

Fig. 5-1 AlignmentStructure Diagram

Element	Attribute	Type	Use	Comment
AlignmentStructure	horizontal	restriction xs:token	optional	value: left, right, center, stretch
	vertical	restriction xs:token	optional	value: center, stretch, top, bottom
	dropdown	restriction of xs:token	optional	value: listView, listBox

This specified the arrangement of answer and questions. It is an optional child element of **Question**. Considering questioner could set arrangement of questions and answers (especially for selection question type). These attributes refer to xAML.

5.3.2 AnonymityStructure

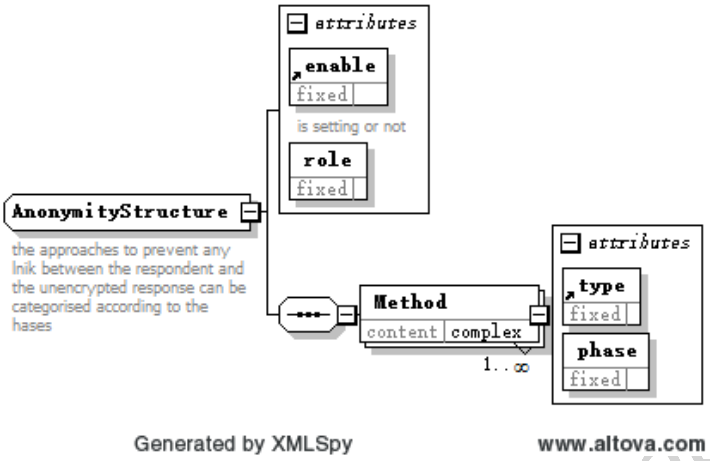


Fig. 5-2 AnonymityStructure Diagram

Element	Attribute	Type	Use	Comment
AnonymityStructure	enable	YesNoType	required	
	role	ParticipantType	required	
Method	type	restriction of xs:token	required	
	phase	PhaseType	required	

This complex type is used to specify the structure of anonymity. It is always to limit respondent for anonymous answer. These methods can be categorized according to the phase. The anonymity method types are listed below:

Method phase	**Method type**
Setting up	randomized auth token
Submitting	blind e-papers
	blinded auth token
	separation of duty
	homomorphic secret
Counting	mix net
	homomorphic encrytptio
	HSM

5.3.3 AnswerStructure

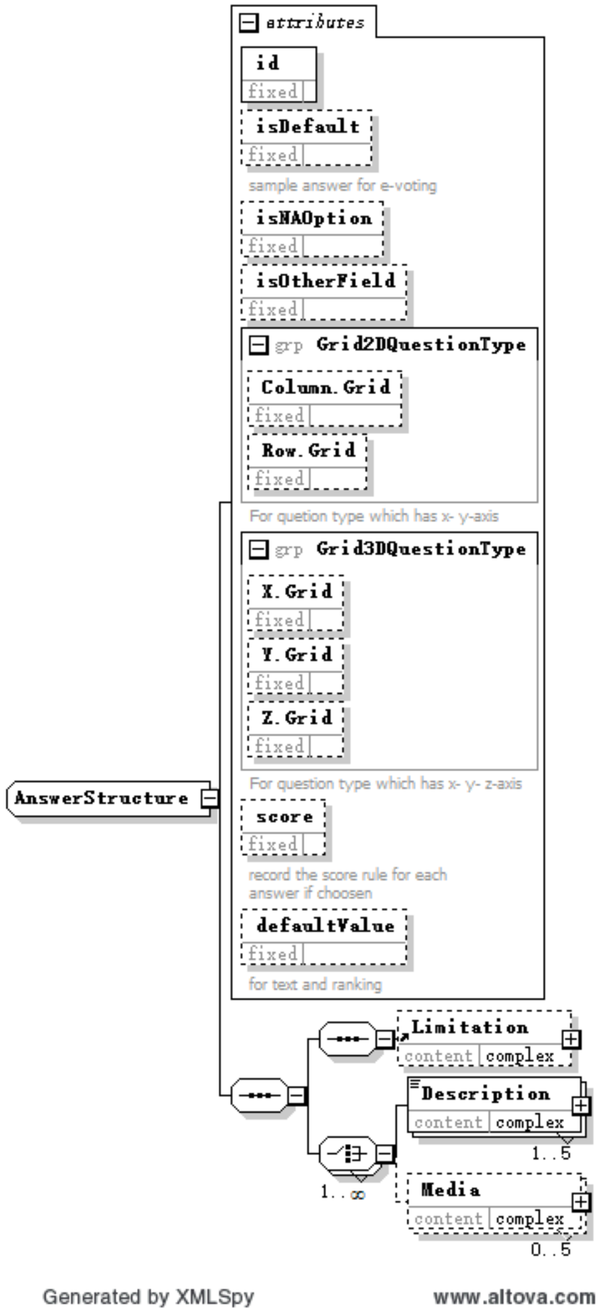


Fig. 5-3 AnswerStructure Diagram

Element	Attribute	Type	Use	Comment
Answer	id	IdType	required	answer id
	isDefault	YesNoType	optional	sample answer for e-testing; *isNAOption* & *isOtherField* exclusive

	isNAOption	YesNoType	optional	*isDefault* & *isOtherField* exclusive
	isOtherField	YesNoType	optional	*isNAOption* & *isDefault* exclusive
	Column.Grid	union: xs:nonNegativeInteger &xs:token (enumeration value: *) token for piping and extraction logic types	optional	for 2D matrix suffix from 0
	Row.Grid		optional	
	X.Grid		optional	for 3D matrix (matrix+drop-down list) suffix from 0
	Y.Grid		optional	
	Z.Grid		optional	
	score	xs:double	optional	answer score rules
	defaultValue	xs:token	optional	text and ranking sample

5.3.4 AuthenticationStructure

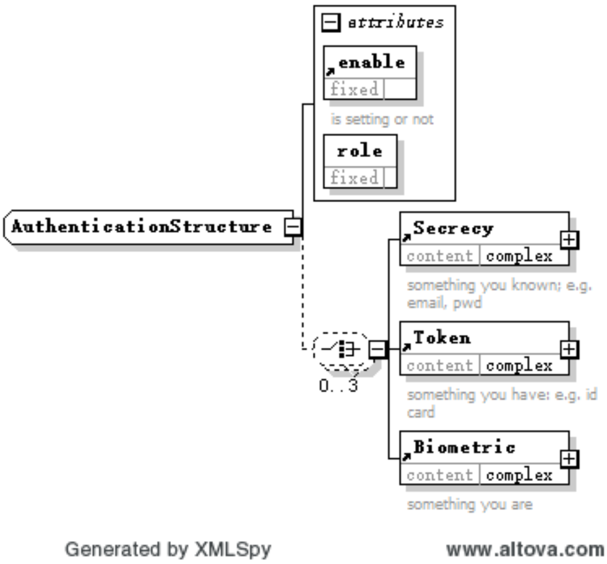


Fig. 5-4 AuthenticationStructure Diagram

Element	Attribute	Type	Use	Comment
AuthenticationStructure	enable	YesNoType	required	
	role	ParticipantType	required	

This complex type is used to specify the structure of authentication to ensure the different methods of it. There are 3 required elements but in a choice order. There are following possibilities to verify the users' identification.

Method	Child Element	Comment
sth. you known	Secrecy	
stn. you have	Token	
sth. you are	Biometric	
duplicated verification	Secrecy & Token	usually used in e-testing to check the token of respondents and let them login in with their secrecy
	Token & Biometric	usually used in e-voting to double check respondent's token and login by their biometric

5.3.5 AuthorityStructure

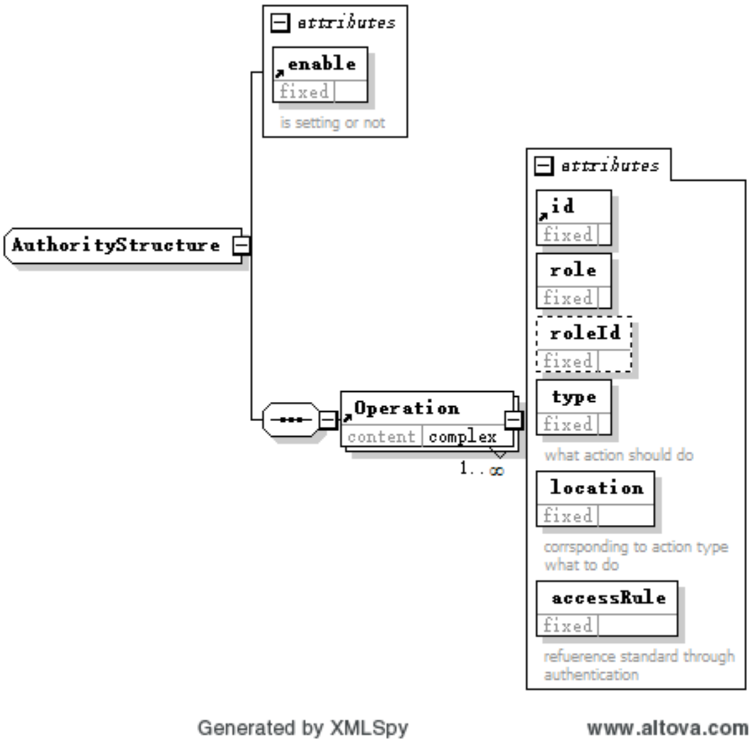


Fig. 5-5 AuthorityStructure Diagram

Element	Attribute	Type	Use	Comment
AuthorityStructure	enable	YesNoType	required	
Operation	id	IdType	required	
	role	ParticipantType	required	who
	roleId	IdType	optional	who

	type	FunctionType	required	how
	location	Union of DataType & PaperTyp	required	what
	accessRule	AuthType	required	how

This complex type is used to specify authority structure. The element **Operation** is its child element to define who (which roles) has which kinds of access control (action: type; scope: location; method: accessRule).

5.3.6 BiometricStructure

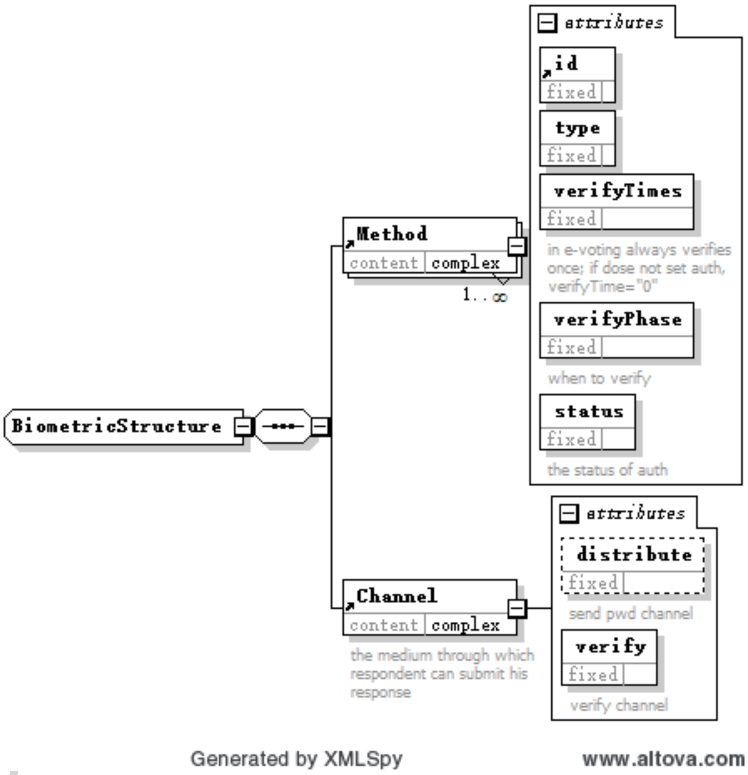


Fig. 5-6 BiometricStructure Diagram

Element	Attribute	Type	Use	Comment
Method	id	IdType	required	
	type	AuthType	required	
	verifyTimes	xs:nonNegativeInteger	required	
	verifyPhase	PhaseType	required	
	status	restriction of xs:token	required	
Channel	distribute	ChannelType	×	
	verify	ChannelType	required	

This complex type is used to specify the structure of Biometrics, which consists of biometrics method and communication channel. It support finger prints, iris, face recognition, signature, and even DNA verify by the channel. It allows the respondent to reuse the biometrics to interrupt to answer and login during different phase. As well recording the status of respondent who has this private things. (not only for respondent, also can for other participant role)

5.3.7 ComponentStructure

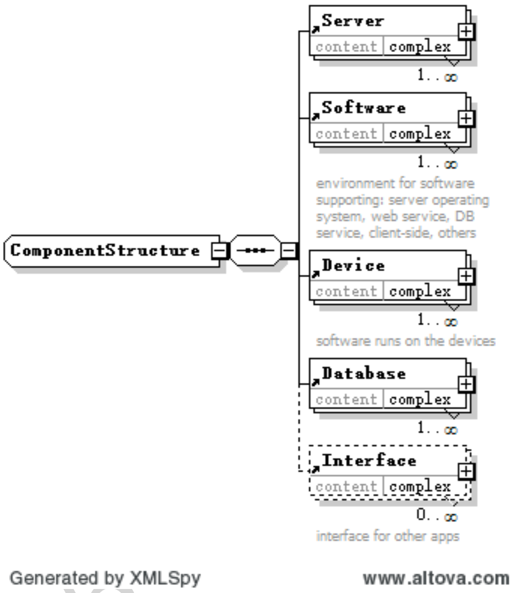


Fig. 5-7 ComponentStructure Diagram

This complex type is used to specify the system's components. There are 4 required children elements and 1 optional children elements in a sequence order. In principle, software executes on the device, and server supports the software and connect with database. As to interface, it connects with other apps. In some complex situations such as a java applet run on a web browser as the software, it is necessary to provide 2 severs at least.

5.3.8 DataStructure

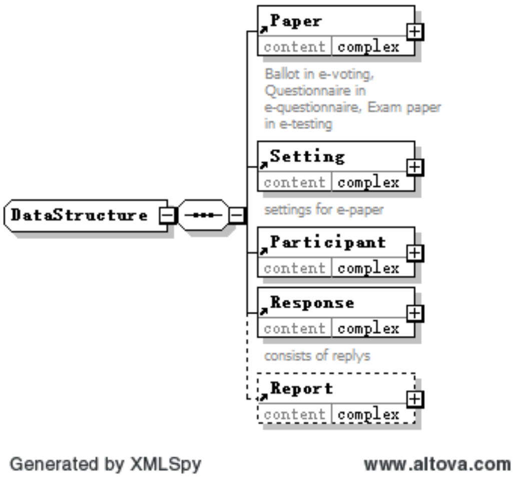


Fig. 5-8 DataStructure Diagram

This complex type is used to specify the data for data exchange. The reused data are following child elements. In this type, there are listed schema 220, 230, 250, 260, and 270. In data exchange specification, we provide only a set of single paper sheet with its necessary data. If questioner or sponsor intend to import and reuse the data, he can reuse it one by one set.

5.3.9 DatabaseStructure

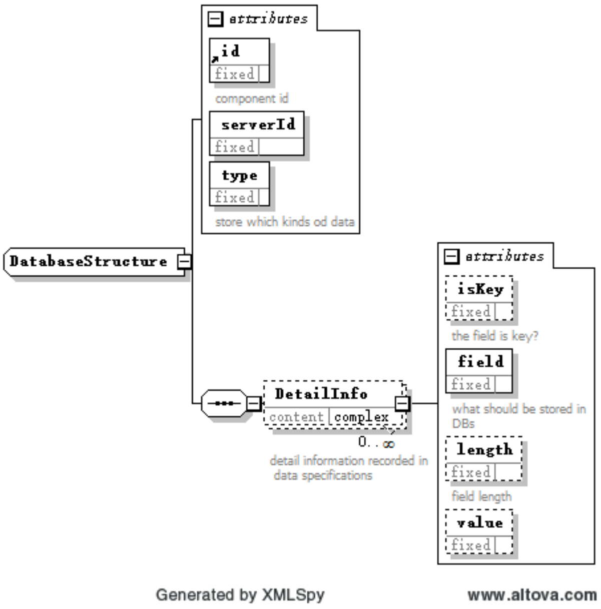


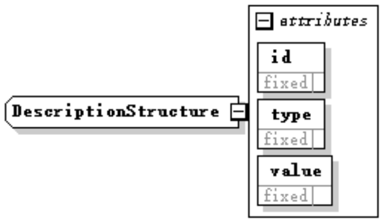
Fig. 5-9 DatabaseStructure Diagram

Element	Attribute	Type	Use	Comment

DatabaseStructure	id	IdType	required	
	serverId	IdsType	required	
	type	Union: DataType & PaperType	required	
DetailInfo	isKey	YeNoType	optional	database detailed info
	field	FieldType	required	
	length	xs:nonNegativeInteger	optional	
	value	xs:NMTOKEN	optional	

This complex type is used to specify the databases. Database connects with server, usually, (server-to-database) one-to-one or one-to multi relation. There is a situation that a database for different servers provide service, please repeat the database elements used same id. DetailInfo element is an optional child element. It is used to record the data in database.

5.3.10 DescriptionStructure



Generated by XMLSpy www.altova.com

Fig. 5-10 DescriptionStructure Diagram

Element	Attribute	Type	Use	Comment
DescriptionStructure	id	IdType	required	
	type	DescriptionType	required	
	value	xs:string	required	title or explanation, or ending page

5.3.11 DeviceStructure

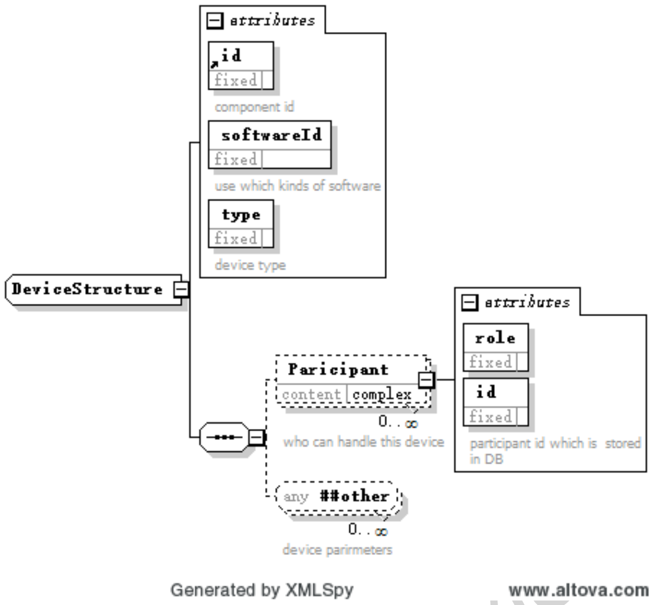


Fig. 5-11 DeviceStructure Diagram

Element	Attribute	Type	Use	Comment
DeviceStructure	id	IdType	required	
	softwareId	IdType	required	
	type	DeviceType	required	

This complex type is used to specify the device. This provide platform to execute software.

5.3.12 FunctionStructure

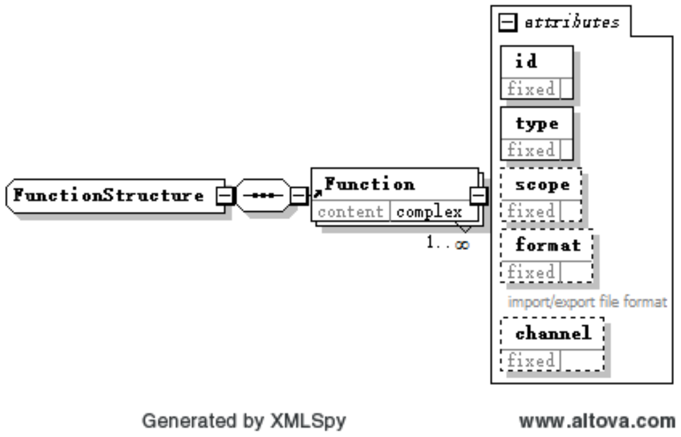


Fig. 5-12 FunctionStructure Diagram

Element	Attribute	Type	Use	Comment
Function	id	IdType	required	
	type	FunctionType	required	
	scope	Union: DataType & PaperType	optional	
	format	FormatType	optional	
	channel	ChannelType	optional	

This structure is used to specify the functions of the systems during each phase. The following table lists the phase and its corresponding functions and scopes.

Phase	**Function types**	**other attributes**
setting up	create	scope (value: questionnaire, testing, voting, setting)
	edit	
	copy	
	add	
	import	scope (all values) format (all values)
	export	
	delete	scope (value: questionnaire, testing, voting, setting)
	search	
	generate token	format (all values)
registering	login	
	monitor	
	authenticate	
distributing	distribute	channel
submitting	answer	
	submit	channel
	export	scope (response) format (all values)
	import	
	monitor	
	ping	
	integrate	scope (response) format (all values)
	back	
	internalStop	
collecting	collect	channel
counting	integrate	scope (response) format (all values)
	count	
marking	mark	
	blind mark	
analysing	gap analysis	

	trend analysis	
	basis analysis	
	report	
	integrate	scope (report) format (all values)
auditing	audit	

5.3.13 InterfaceStructure

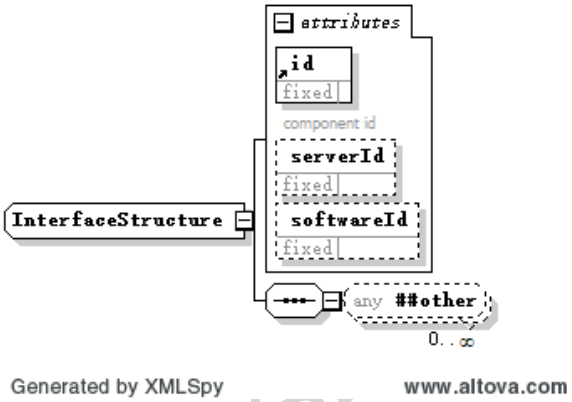


Fig. 5-13 InterfaceStructure Diagram

Element	Attribute	Type	Use	Comment
InterfaceStructure	id	IdType	required	
	serverId	IdType	optional	
	softwareId	IdType	optional	

This complex type is used to specify the interface with other apps or software.

5.3.14 LimitationStructure

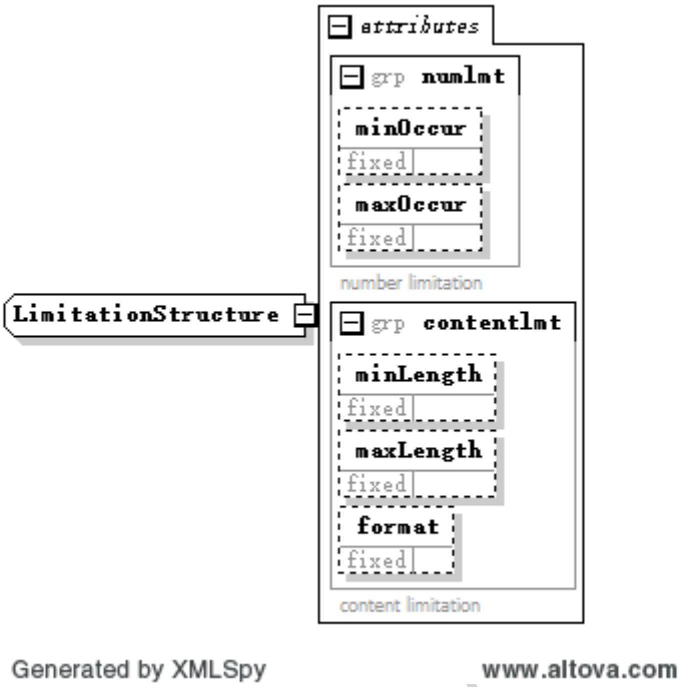


Fig. 5-14 LimitationStructure Diagram

Element	Attribute	Type	Use	Comment
LimitationStructure	minOccur	xs:nonNegativeInteger	optional	number limitation
	maxOccur	xs:nonNegativeInteger	optional	
	minLength	xs:nonNegativeInteger	optional	text area limitation
	maxLength	xs:nonNegativeInteger	optional	
	format	FieldType	optional	

To limit the selection numbers, this provides *numlmt* attribute group to restrict. To limit the text length, this provides *contentlmt* attribute group to restrict.

5.3.15 LogicStructure

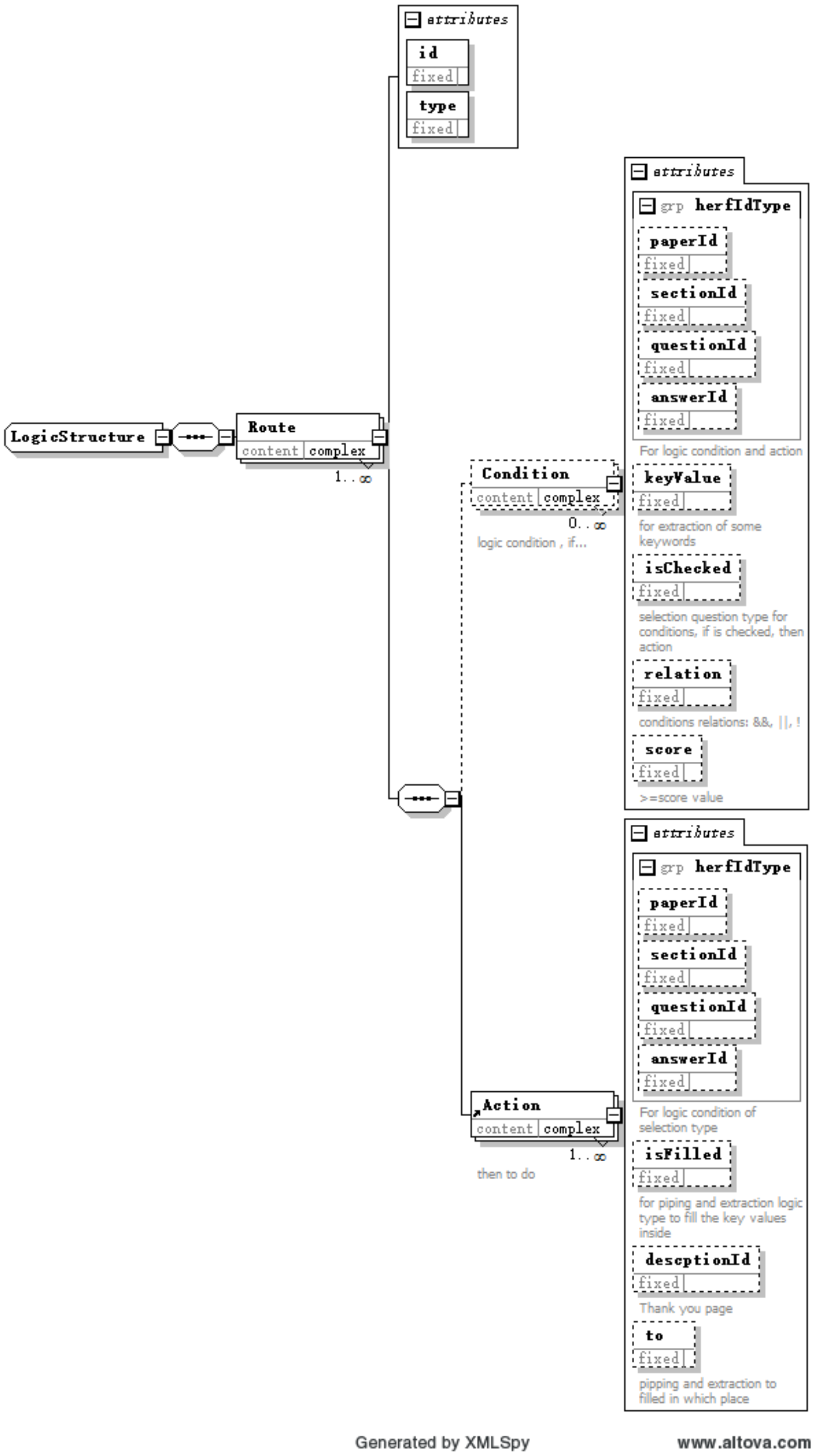


Fig. 5-15 LogicStructure Diagram

Element	Attribute	Type	Use	Comment
Route	id	IdType	required	
	type	LogicType	required	

Condition	paperId	IdType	optional	
	sectionId	IdType	optional	
	questionId	IdType	optional	
	answerId	IdType	optional	
	isChecked	YesNoType	optional	condition if question type selection is selected, then..
	relation	restriction of xs:token	optional	conditions relations: &&, ||, !
	score	xs:double	optional	>=score value
Action	paperId	IdType	optional	
	sectionId	IdType	optional	
	questionId	IdType	optional	
	answerId	IdType	optional	
	paperId	IdType	optional	
	isFilled	YesNoType	optional	
	descriptionId	IdType	optional	for jump to conclusion
	to	union: xs:anyURI& restriction of xs:token pattern:" $('pip[0-9]*'), $('ext[0-9]*')	optional	to url or piping| extract content

5.3.16 MediaStructure

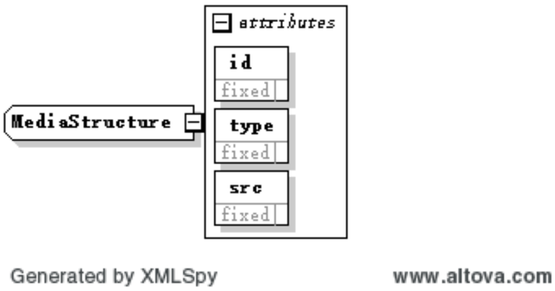


Fig. 5-16 MediaStructure Diagram

Element	Attribute	Type	Use	Comment
MediaStructure	id	IdType	required	
	type	MediaType	optional	
	src	xs:anyURI	optional	relative path

5.3.17 PaperStructure

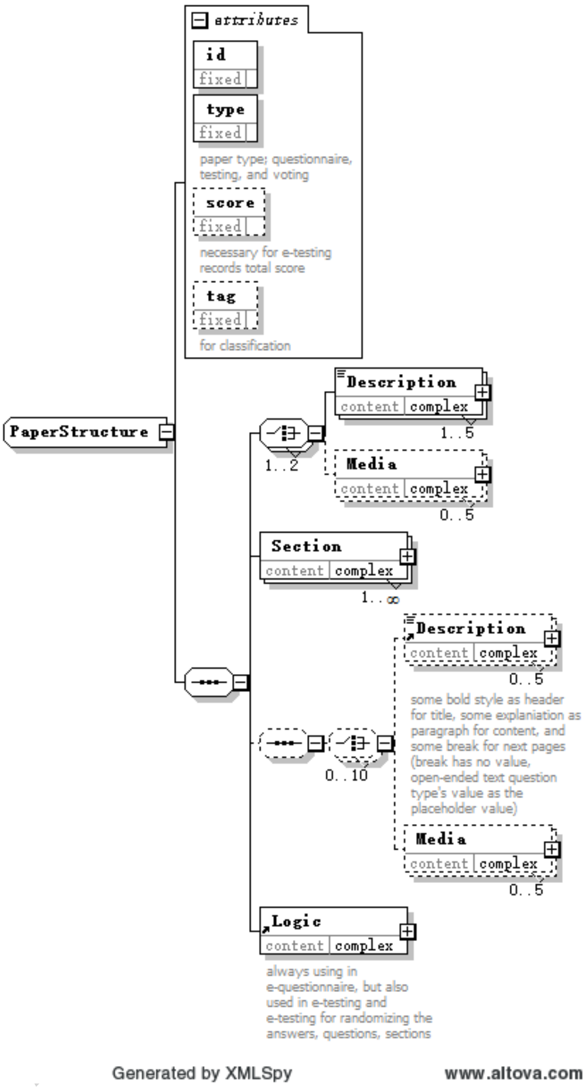

attributes
id
fixed
type
fixed
paper type: questionnaire, testing, and voting
score
fixed
necessary for e-testing records total score
tag
fixed
for classification
PaperStructure
Description
content complex
1..5
1..2
Media
content complex
0..5
Section
content complex
1..∞
Description
content complex
0..5
some bold style as header for title, some explanation as paragraph for content, and some break for next pages (break has no value, open-ended text question type's value as the placeholder value)
0..10
Media
content complex
0..5
Logic
content complex
always using in e-questionnaire, but also used in e-testing and e-testing for randomizing the answers, questions, sections
Generated by XMLSpy
www.altova.com

Fig. 5-17 PaperStructure Diagram

Element	Attribute	Type	Use	Comment
PaperStructure	id	IdType	required	
	type	xs:token	required	value: questionnaire, testing, voting
	score	xs:positiveInteger	optional	paper total score
	tag	xs:token	optional	for classification

This complex type is specified the structure of a paper. This paper consists of title defined as **Description**, and some media supporting it. Paper consists of sections and some conclusions.

5.3.18 PaperSheetStructure

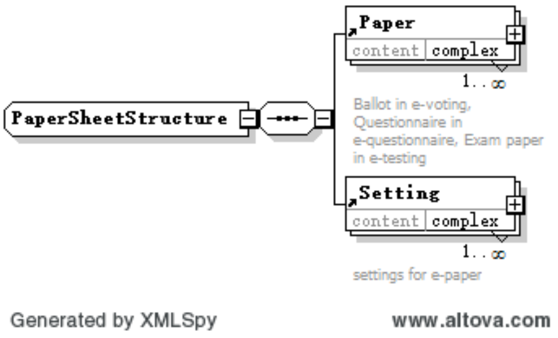


Fig. 5-18 PaperSheetStructure Diagram

PaperSheet is the schema 120. It consists of lots of papers and settings for corresponding papers.

5.3.19 ParticipantStructure

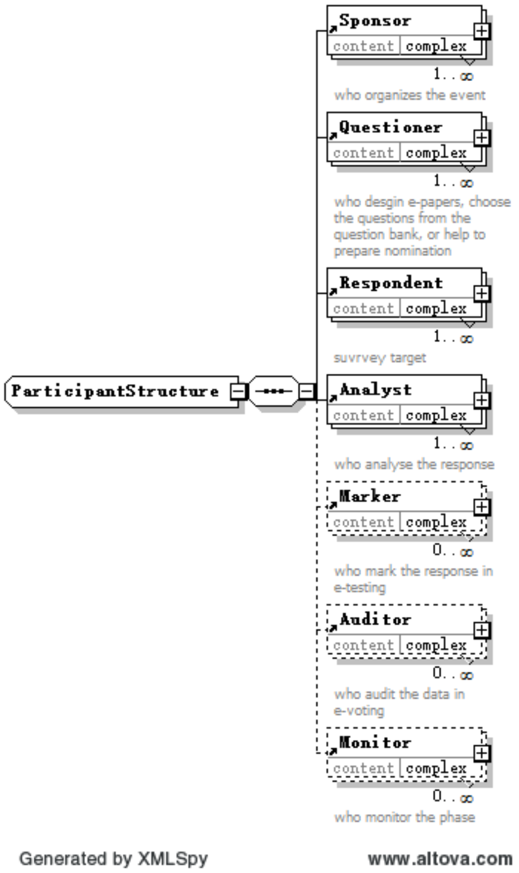


Fig. 5-19 ParticipantStructure Diagram

This complex type is used to specify the participant roles. There are 4 required children elements and 3 optional children elements in a sequence order. To distinguish the participant roles of e-questionnaire, e-testing, and e-voting, we list the necessary roles for each kind of events.

Role	Questionnaire	Testing	Voting
Sponsor	required	required	required
Questioner	required	required	required
Respondent	required	required	required
Analyst	required	required	required
Marker	optional	required	optional
Auditor	optional	optional	required
Monitor	optional	required	optional
Candidate	×	×	required
Proposal	×	×	optional

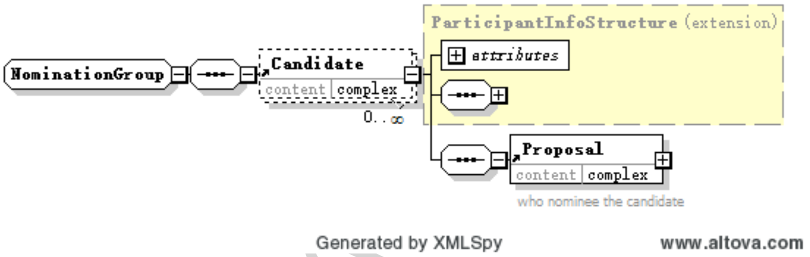


Fig. 5-20 NominationGroup Diagram

This is an extension of ParticipantStructure for element **Participant**. It records the information in an e-voting event, especially in the options nomination (Candidate info: approving the options to be presented to respondents in a referendum) and candidate nomination (Proposal and candidate info: approving nominees as eligible candidates for certain positions in an election by whom proposal).

5.3.20 ParticipantInfoStructure

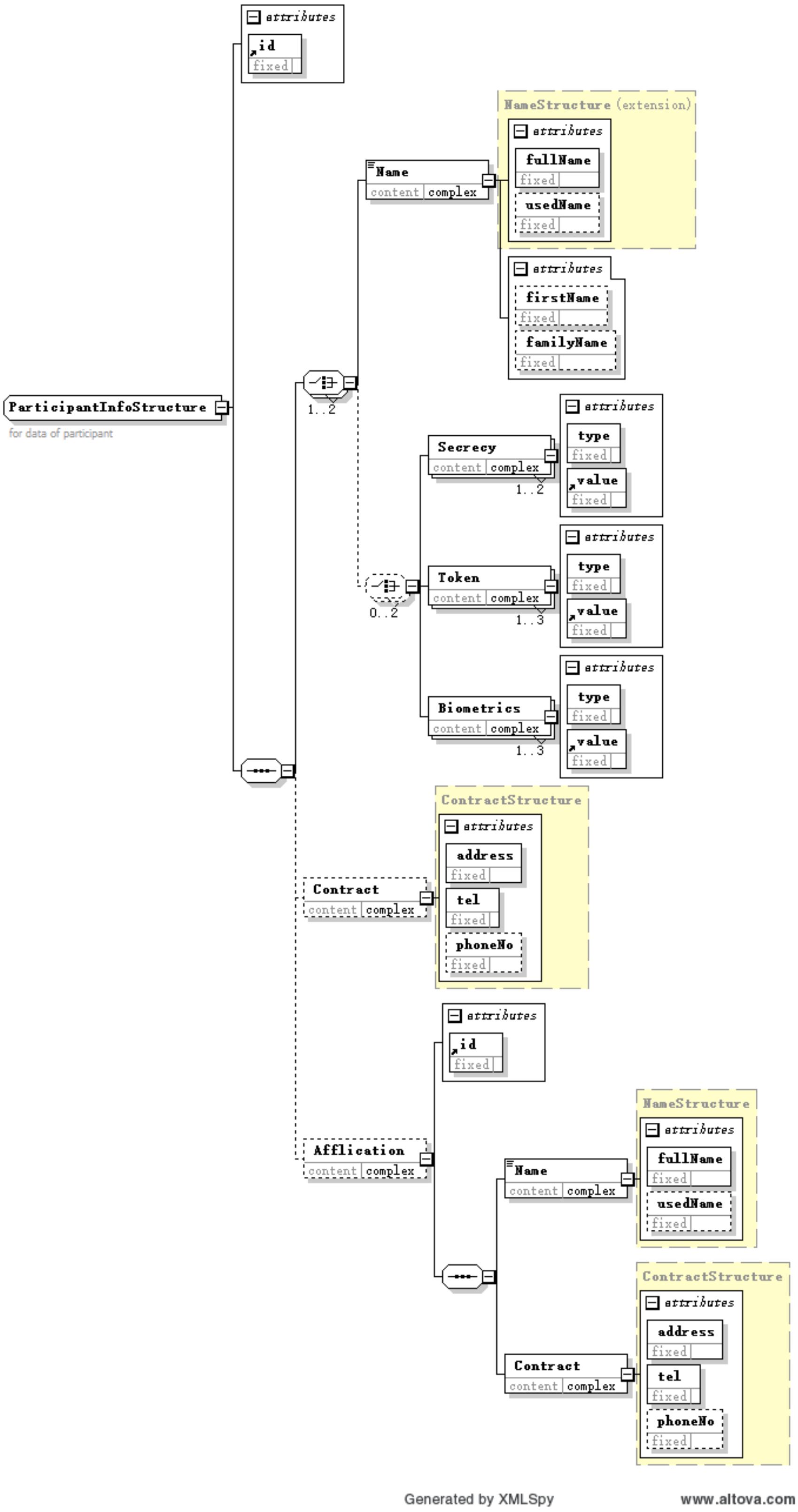


Fig. 5-21 ParticipantInfoStructure Diagram

Element	Attribute	Type	Use	Comment
ParticipantInfoStructure	id	IdType	required	
Name	fullName	NameType	required	
	usedName	NameType	optional	
	firstName	NameType	optional	
	familyName	NameType	optional	
Secrecy/Token/Biometrics	type	AuthType	required	
	value	xs:token	required	general security value
Contract	address	AddressType	required	
	tel	TelNumType	required	
	phoneNo	TelNumType	optional	
Affiliation	id	IdType	required	

This complex type is used to specify the participant's detailed information to record in database. It applies to each kind of participant roles with some extensions. In addition, it allows to records the general security values of each participant.

The necessary information about a participant are the identification and full name. There is a situation is a respondent change his name and the registered name becoming a used name when in a national election event. Even to register and record much more detailed information for security, but compared with simple survey just need a name with a necessary identificaiton.

5.3.21 PhaseStructure

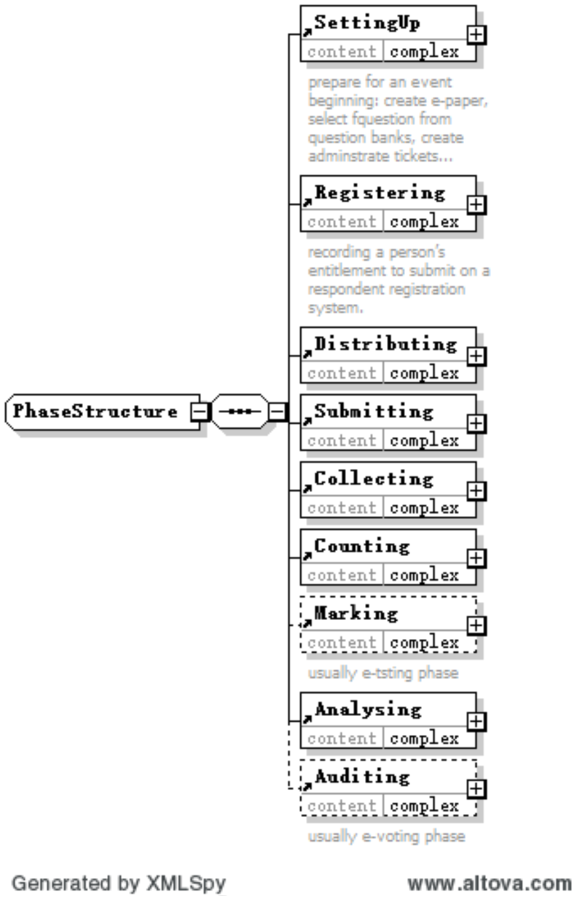


Fig. 5-22 PhaseStructure Diagram

This complex type is used to specify the phase list. There are 7 required children elements and 2 optional children elements in a sequence order. To distinguish the necessary phases of e-questionnaire, e-testing, and e-voting, we list them for each kind of events.

Phase	Questionnaire	Testing	Voting
SettingUp	required	required	required
Registering	required	required	required
Respondent	required	required	required
Distributing	required	required	required
Submitting	required	required	required
Collecting	required	required	required
Counting	required	required	required
Marking	×	required	×
Analysing	required	required	required
Auditing	×	×	required

5.3.22 QSLStructure

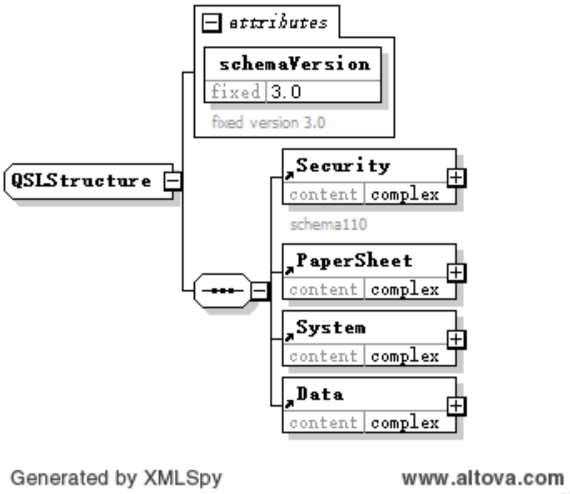


Fig. 5-23 QSLStructure Diagram

Element	Attribute	Type	Use	fixed
QSLStructure	schemaVersion	xs:token	required	3.0

This complex type is used to specify the QSL document (specification). There are 4 required children elements (schema 110, 120, 130 ,140) in a sequence order.

5.3.23 QuestionStructure

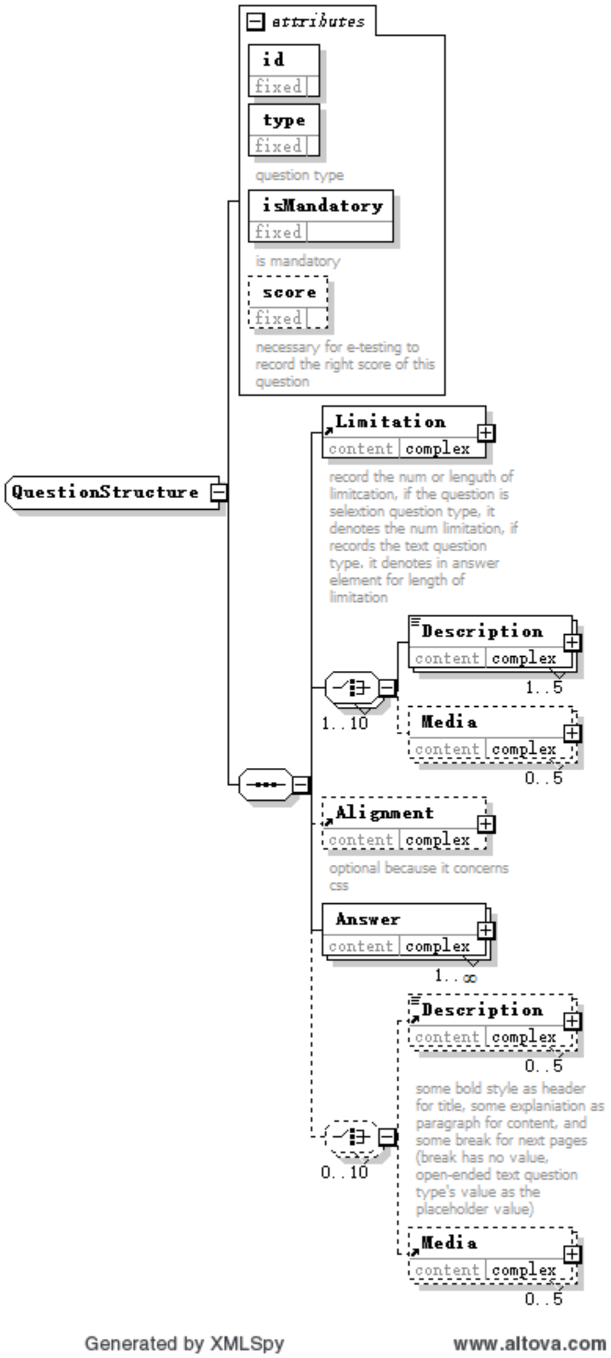

attributes
id
fixed
type
fixed
question type
isMandatory
fixed
is mandatory
score
fixed
necessary for e-testing to record the right score of this question
QuestionStructure
Limitation
content complex
record the num or length of limitation, if the question is selection question type, it denotes the num limitation, if records the text question type, it denotes in answer element for length of limitation
Description
content complex
1..5
1..10
Media
content complex
0..5
Alignment
content complex
optional because it concerns css
Answer
content complex
1..∞
Description
content complex
0..5
some bold style as header for title, some explanation as paragraph for content, and some break for next pages (break has no value, open-ended text question type's value as the placeholder value)
0..10
Media
content complex
0..5
Generated by XMLSpy
www.altova.com

Fig. 5-24 QuestionStructure Diagram

Element	Attribute	Type	Use	Comment
QuestionStructure	id	IdType	required	
	type	QuestionType	required	if section type chosen, then question type is similar with

				section question type
	isMandatory	YesNoType	required	* necessary to answer
	score	xs:positiveInteger	optional	question total score

5.3.24 ReportStructure

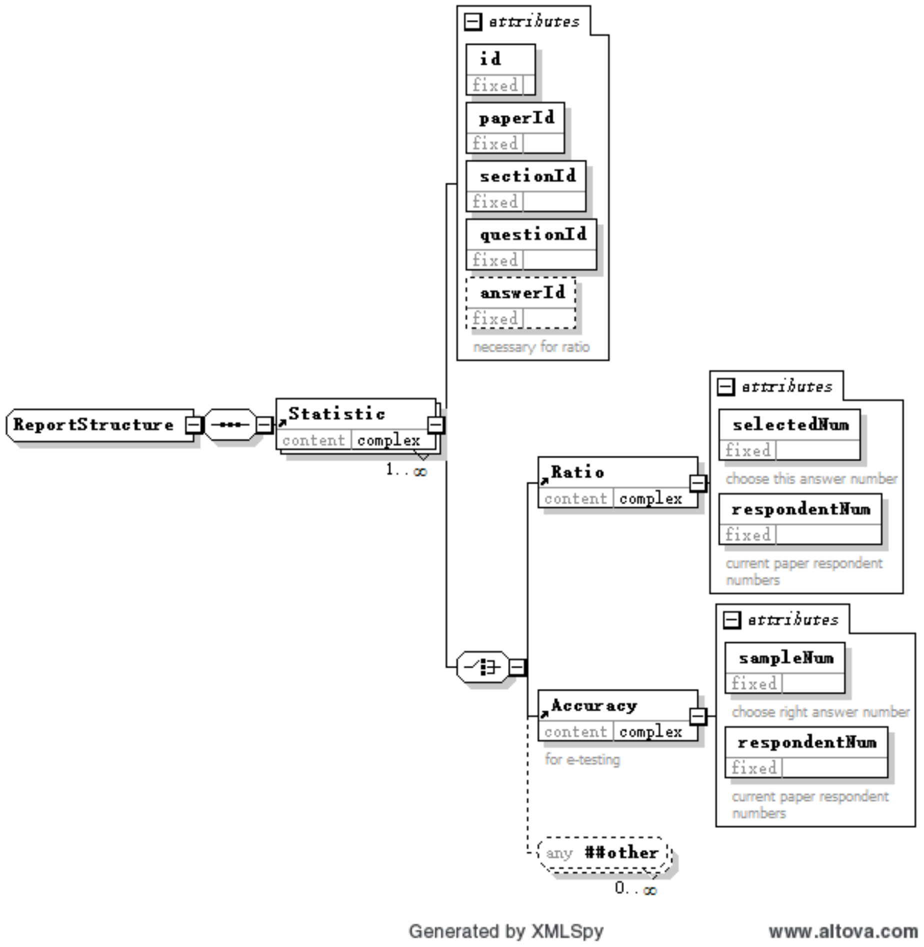


Fig. 5-25 ReportStructure Diagram

Element	Attribute	Type	Use	Comment
Statistic	id	IdType	required	main of report
	paperId	IdType	required	
	sectionId	IdType	required	
	questionId	IdType	required	
	answerId	IdType	optional	necessary for ratio

Ratio	selectedNum	xs:nonNegativeInteger	required	selected<= respondent
	respondentNum	xs:nonNegativeInteger	required	
Accuracy	sampleNum	xs:nonNegativeInteger	required	for e-testing; sample<= respondent
	respondentNum	xs:nonNegativeInteger	required	

5.3.25 ResponseStructure

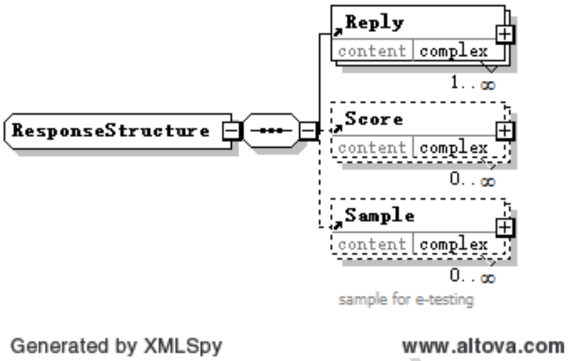


Fig. 5-26 ResponseStructure Diagram

This complex type is used to specify the respondents' responses in detail. There are 1 required child element and 2 optional child elements in a sequence order. In an event for testing, It concerns the score data and sample answer data.

5.3.26 SecrecyStructure

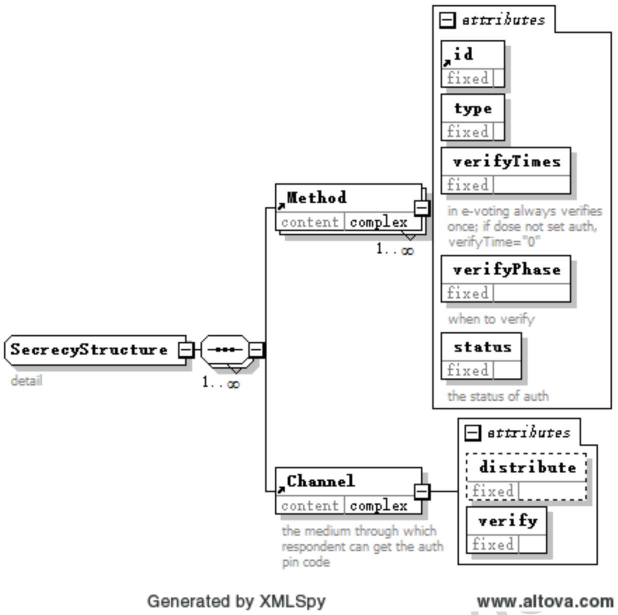


Fig. 5-27 SecrecyStructure Diagram

Element	Attribute	Type	Use	Comment
Method	id	IdType	required	
	type	AuthType	required	
	verifyTimes	xs:nonNegativeInteger	required	
	verifyPhase	PhaseType	required	
	status	restriction of xs:token	required	
Channel	distribute	ChannelType	required	
	verify	ChannelType	required	

This complex type is used to specify the structure of Secrecy, which consists of secrecy method and communication channel. It support only password and randomized password as the secrecy send to respondent by the channel, and verify by same or different channel. It allows the respondent to reuse the password to interrupt to answer and login during different phase. As well recording the status of respondent who has this password. (not only for respondent, also can for other participant role)

5.3.27 SectionStructure

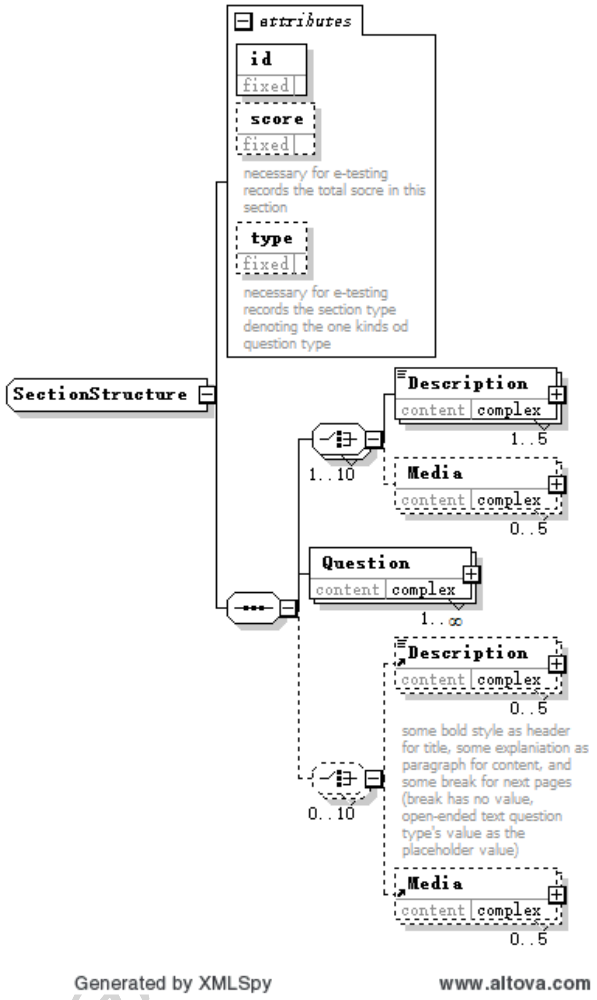

attributes
id
fixed
score
fixed
necessary for e-testing records the total socre in this section
type
fixed
necessary for e-testing records the section type denoting the one kinds od question type
SectionStructure
Description
content complex
1..5
1..10
Media
content complex
0..5
Question
content complex
1..∞
Description
content complex
0..5
some bold style as header for title, some explaniation as paragraph for content, and some break for next pages (break has no value, open-ended text question type's value as the placeholder value)
0..10
Media
content complex
0..5
Generated by XMLSpy
www.altova.com

Fig. 5-28 SectionStructure Diagram

Element	Attribute	Type	Use	Comment
SectionStructure	id	IdType	required	
	type	QuestionType	optional	classify section in an unified question type usually in e-testing
	score	xs:positiveInteger	optional	section total score

5.3.28 SecurityStructure

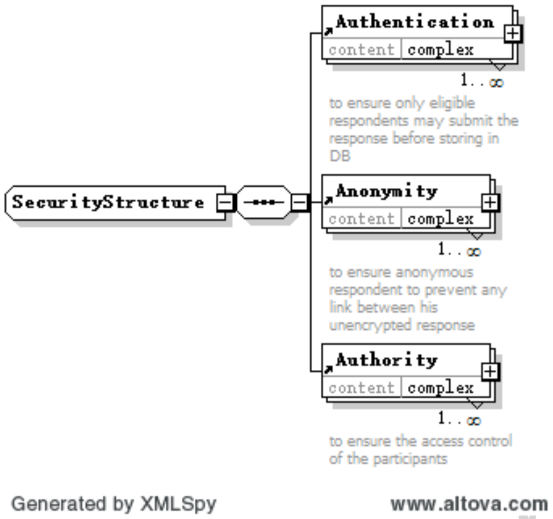

SecurityStructure
Authentication
content complex
1..∞
to ensure only eligible respondents may submit the response before storing in DB
Anonymity
content complex
1..∞
to ensure anonymous respondent to prevent any link between his unencrypted response
Authority
content complex
1..∞
to ensure the access control of the participants
Generated by XMLSpy
www.altova.com

Fig. 5-29 SecurityStructure Diagram

This complex type is used to specify security structure what a system should have. It consists of schema 510 and schema 520. As to element **Authority** is public, necessary element for e-questionnaire, e-testing, and e-voting. Schema 510 and 520 are necessary for e-voting, but also optional for e-questionnaire and e-testing. Especially in an election, there is a serious limitation to ensure fairness, justice, open, and transparency.

5.3.29 ServerStructure

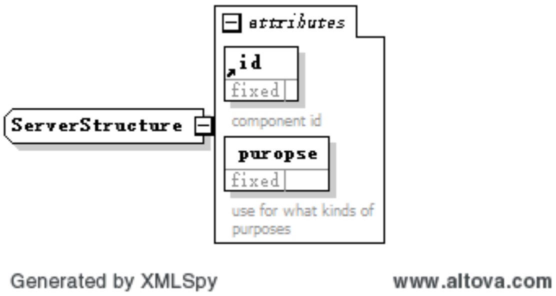
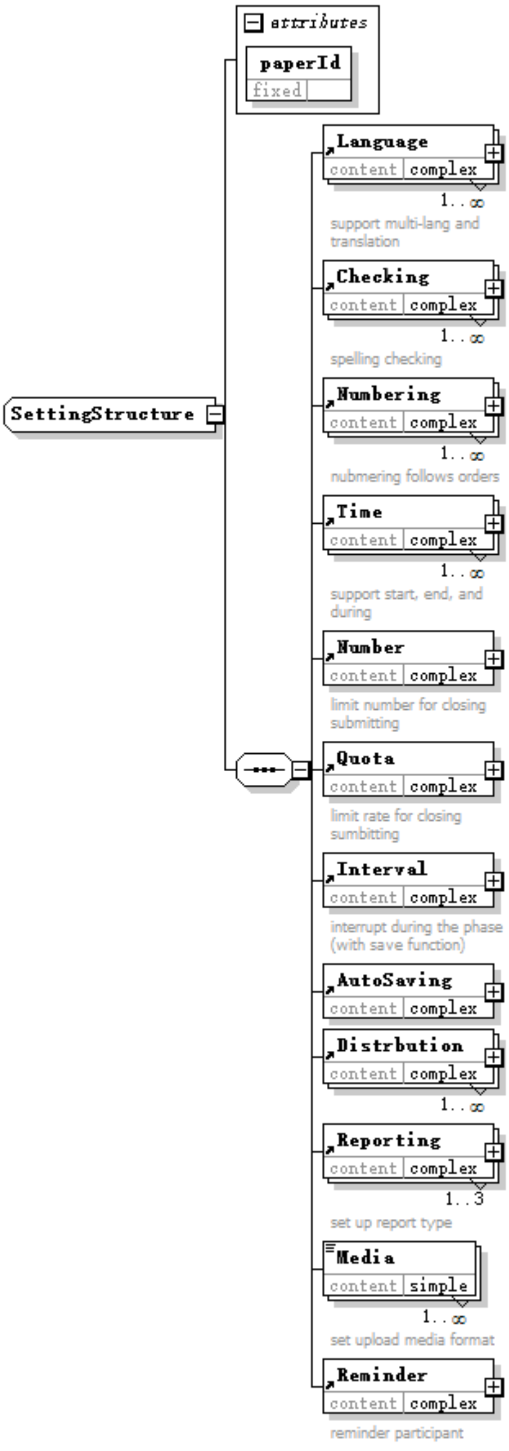


Fig. 5-30 ServerStructure Diagram

Element	Attribute	Type	Use	Comment
ServerStructure	id	IdType	required	
	purpose	PhaseType	required	

This complex type is used to specify which kinds of servers of the system should provide to support an event. In an election, it is necessary to provides much more registering servers, submitting servers, and independent auditing servers for security. The server id is used to connect with database if it is necessary, and to support software. As to the purpose of the server, this is used to define the server is used for which phase.

5.3.30 SettingStructure



Generated by XMLSpy www.altova.com

Fig. 5-31 SettingStructure Diagram

Element	Attribute	Type	Use	Comment
SettingStructure	paperId	IdType	required	

5.3.31 SoftwareStructure

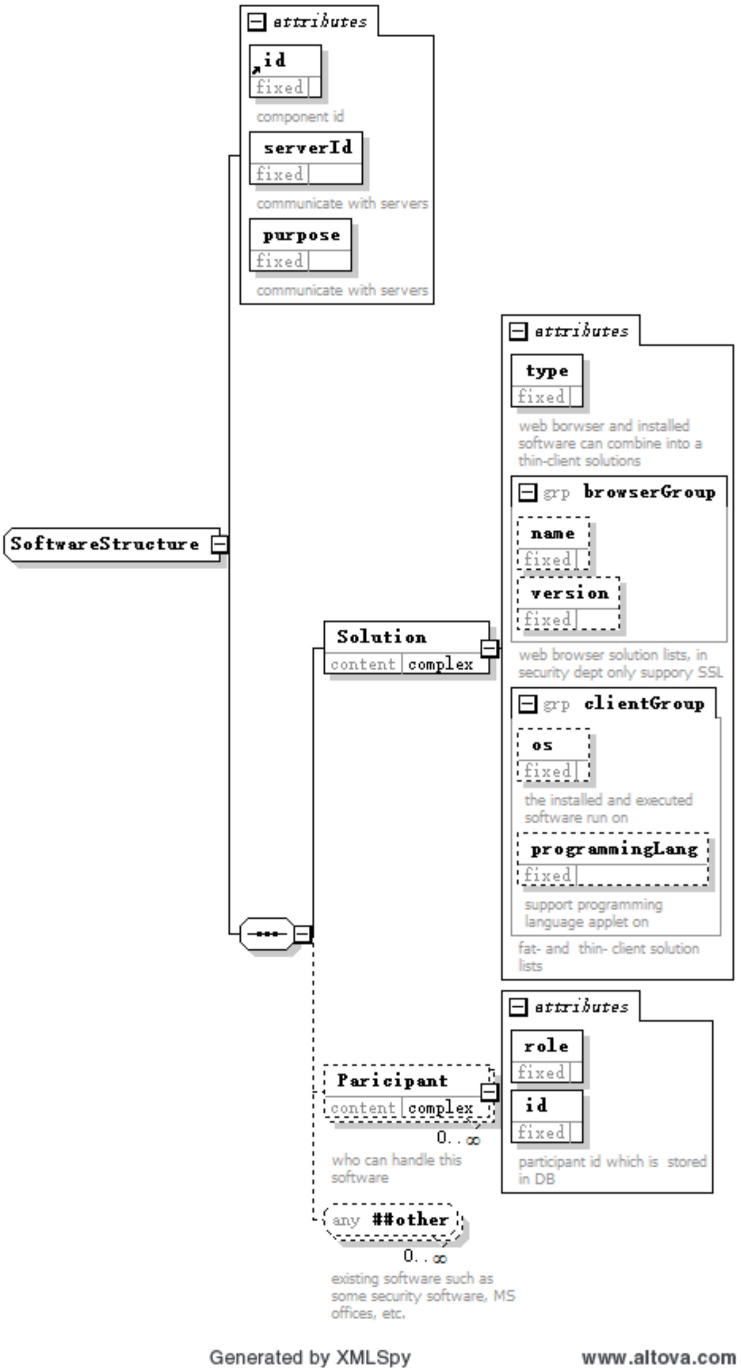

attributes
id
fixed
component id
serverId
fixed
communicate with servers
purpose
fixed
communicate with servers
SoftwareStructure
attributes
type
fixed
web borwser and installed software can combine into a thin-client solutions
grp browserGroup
name
fixed
version
fixed
web browser solution lists, in security dept only suppory SSL
grp clientGroup
os
fixed
the installed and executed software run on
programmingLang
fixed
support programming language applet on
fat- and thin- client solution lists
Solution
content complex
attributes
role
fixed
id
fixed
participant id which is stored in DB
Paricipant
content complex
0..∞
who can handle this software
any ##other
0..∞
existing software such as some security software, MS offices, etc.
Generated by XMLSpy
www.altova.com

Fig. 5-32 SoftwareStructure Diagram

Element	Attribute	Type	Use	Comment
SoftwareStructure	id	IdType	required	
	serverId	union:IdType|xs:token (pattern:0)	required	connect with server
	purpose	PhaseType	required	

Element	Attribute	Type	Use	Comment
Solution	type	ClientSideSolutionType	required	
	name	BrowserType	optional	browser solution
	version	xs:token	optional	
	os		optional	fat|thin solution
	programmingLang	ProgrammingLangType	optional	
Participant	id	IdType	optional	
	role	ParticipantType	required	

This complex type is used to specify the software information, which connect with server, and execute on the device. Attribute id is used to connect with device, and serverId is used to connect with server, purpose is used to define this software is for which phase. There is a possibility that a software concerns with a lot of same purpose's server. Obviously, an attribute serverId is not enough, thus, please choose *serverId="0"* and set *purpose=same phase with server*. This complex type has a required element **Solution** and an optional element **Participant**. There are 3 kinds of solution, which are web browser solution, fat-solution (installed software), and thin-solution (combination of the first and the second, that means applet runs on browser). Meanwhile, there are two group for defining different solutions. Participant who can handle this software are recorded.

5.3.32 SystemStructure

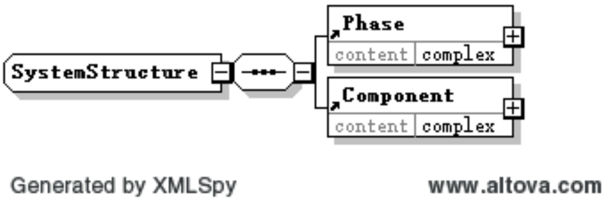


Fig. 5-33 SystemStructure Diagram

This type consists of 2 required child elements. In addition, these child elements are schema 210 and schema 240. System specification is considered from software, hardware, functions.

5.3.33 TokenStructure

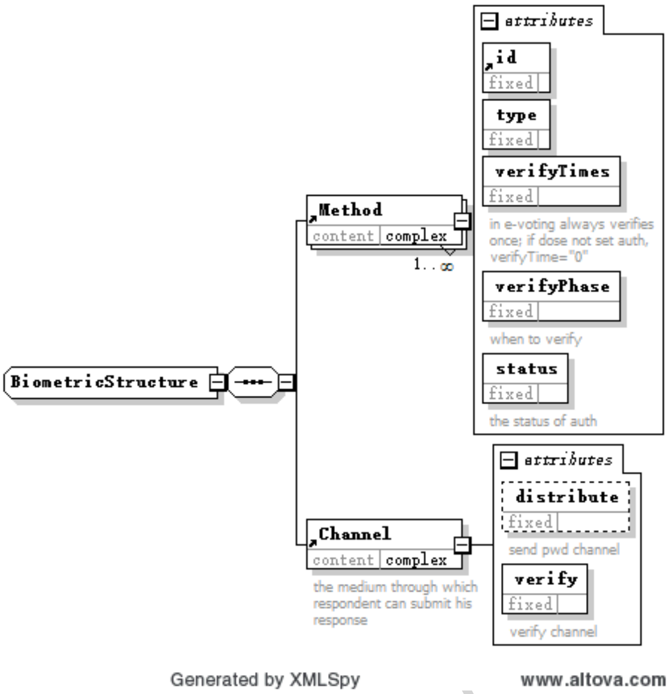


Fig. 5-34 TokenStructure Diagram

Element	Attribute	Type	Use	Comment
Method	id	IdType	required	
	type	AuthType	required	
	verifyTimes	xs:nonNegativeInteger	required	
	verifyPhase	PhaseType	required	
	status	restriction of xs:token	required	
Channel	distribute	ChannelType	×	
	verify	ChannelType	required	

This complex type is used to specify the structure of Token, which consists of token method and communication channel. It support ID card, job card, library card, and identifier as the token verify by the channel. It allows the respondent to reuse the token to interrupt to answer and login during different phase. As well recording the status of respondent who has this token. (not only for respondent, also can for other participant role)

6 QSL Schema Descriptions

6.1 Overview

The following table lists a overview of QSL schemas. Further explanations are given in the following sub paragraphs.

Schema Name	**Purpose**
QSL-100 QSL	declares the core definitions of QSL schema
QSL-110 Security	provides security schema for secure system
QSL-120 PaperSheet	provides schema of paper and setting
QSL-130 System	provides schema of phases and component
QSL-140 Data	provides schema of data of participant, paper, setting, response, and report
QSL-210 Phase	provides schema for complete phases with fulfilled functions
QSL-220 Paper	provides schema of a complete paper construction with logic
QSL-230 Setting	provides schema of settings that questioner sets up before distributing
QSL-240 Component	provides schema of server, software, device, database, interface
QSL-250 Participant	provides schema of participant roles and information
QSL-260 Response	provides schema of response list
QSL-270 Report	provides schema of report list
QSL-310 Logic	provides schema of logic of paper, section, question and answer
QSL-410 Marker	provides schema of marker and information
QSL-420 Score	provides schema of scoring rules
QSL-430 Sample	provides schema of sample answers
QSL-440 Formula	provides schema of math formula
QSL-510 Authentication	provides schema of authentication for eligible respondents
QSL-520 Anonymity	provides schema of anonymous method in different phases for different participant role
QSL-530 Auditing	provides schema of auditing and corresponding server, participant role named auditor
QSL-540 Candidate	provides schema of candidate and information
QSL-610 simpleType	provides general simpleType definitions for reusing attribute values

6.2 110-QSL

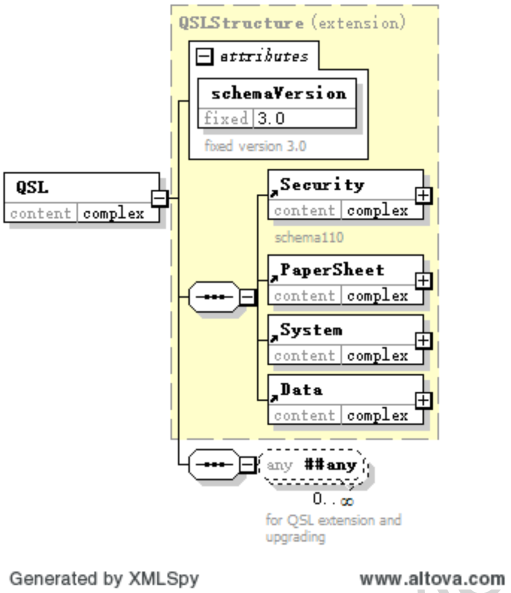


Fig. 6-1 QSL Diagram

Element	Attribute	Type	Use	fixed
QSL	schemaVersion	xs:token	required	3.0

6.2.1 Description of Schema

This schema is used for declaration of a specification, as well as the identifier for QSL structure editor. Defined any QSL-format specification must define this schema. This schema is based on the extension of complex type QSLStructure. Because we provide the extension for QSL.

6.2.2 Source of Schema

```
<xs:schema xmlns="http://www.aise.ics.saitama-u.ac.jp/qsl"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
targetNamespace="org:aise:name:zy:schema:qsl" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="3.0" id="qsl-100" xml:lang="en">
    <xs:include schemaLocation="110-security-v3.0.xsd"/>
    <xs:include schemaLocation="120-papersheet-v3.0.xsd"/>
    <xs:include schemaLocation="130-system-v3.0.xsd"/>
    <xs:include schemaLocation="140-qsldata-v3.0.xsd"/>
    <xs:element name="QSL" id="qsl100">
```

```
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="QSLstructure">
                    <xs:sequence minOccurs="0">
                        <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded">
                            <xs:annotation>
                                <xs:documentation>if there is any
other</xs:documentation>
                            </xs:annotation>
                        </xs:any>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="QSLstructure">
        <xs:sequence>
            <xs:element ref="Security"/>
            <xs:element ref="PaperSheet"/>
            <xs:element ref="System"/>
            <xs:element ref="Data"/>
        </xs:sequence>
        <xs:attribute name="schemaVersion" type="xs:token" use="required"
fixed="3.0">
            <xs:annotation>
                <xs:documentation>fixed version 3.0</xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>
</xs:schema>
```

6.3 110-Security

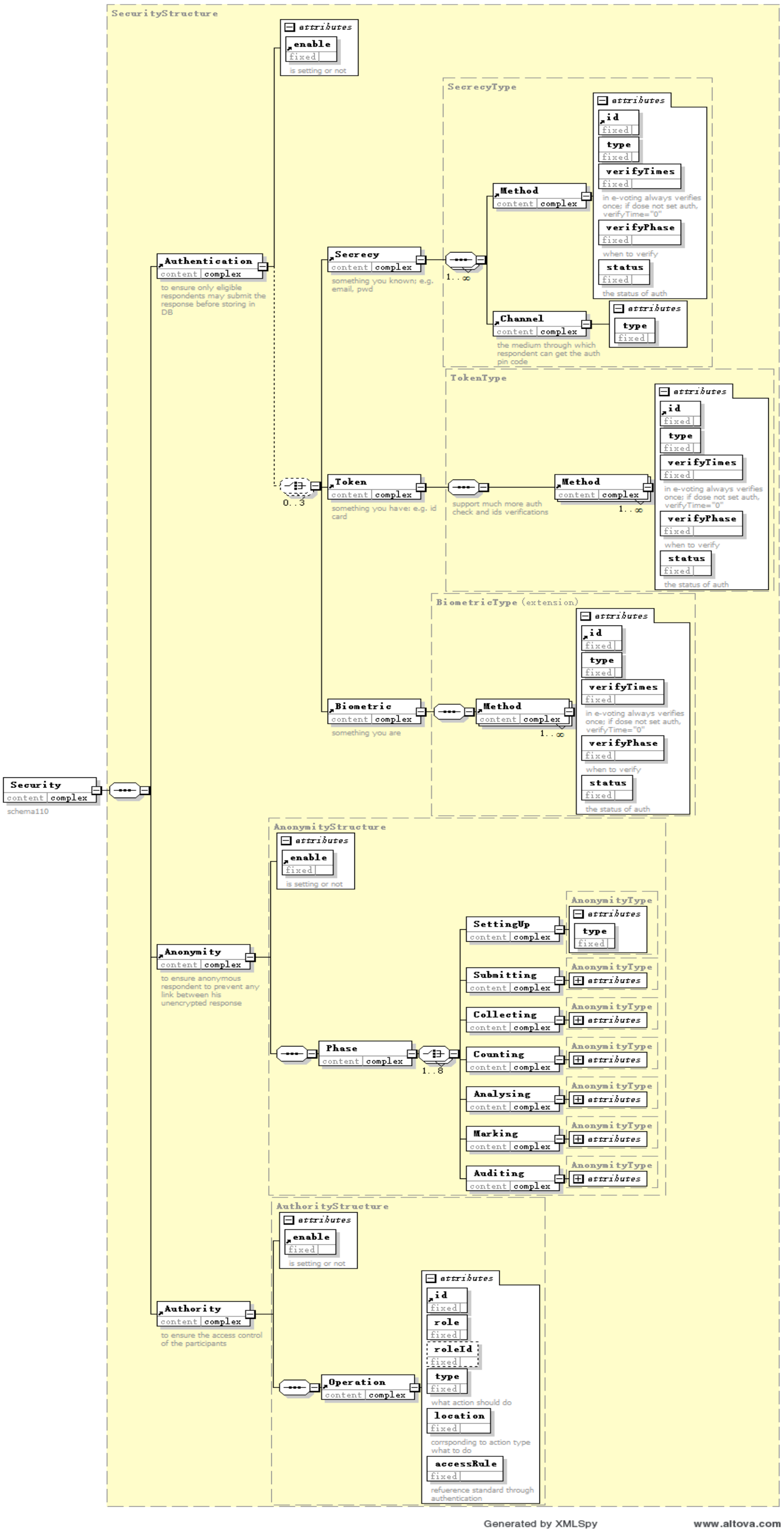


Fig. 6-2 Security Diagram

6.3.1 Description of Schema

This schema is used for specify the security of e-questionnaire, e-testing, and e-voting systems. It specifies the authentication to ensure only eligible respondents may submit the response before storing in database, anonymity to ensure anonymous respondent to prevent any link between his unencrypted response, authority to ensure the access control of the participants. This schema is based on the complex type of SecurityStructue.

6.4 120-PaperSheet

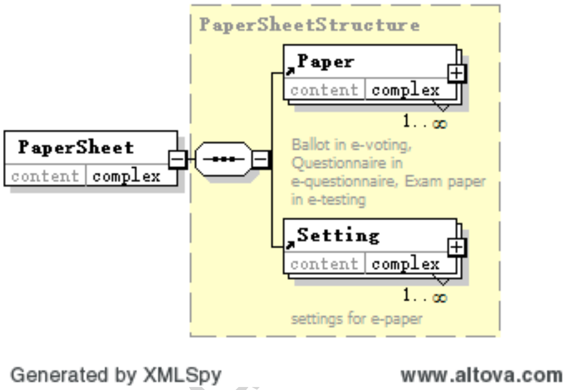


Fig. 6-3 PaperSheet Diagram

Please refer to PaperSheetStructure. This schema records the paper sheets consisting of lots of sections, questions, settings for them. Questioners and sponsors can directly use its specifications to define papers to respondents. It is the core component of the question bank for e-questionnaire and e-testing.

6.5 130-System

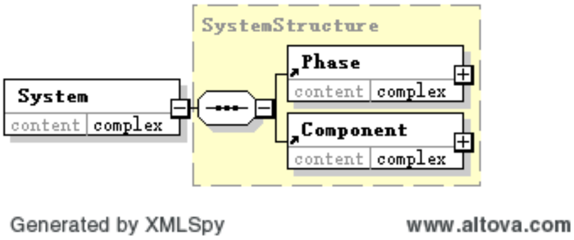
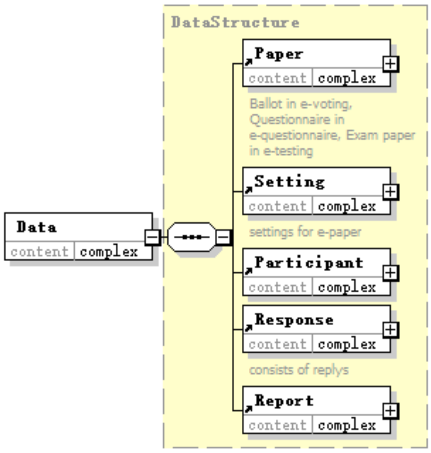


Fig. 6-4 System Diagram

This schema is used for specify the system, which provide services to do e-questionnaire, e-testing, and e-voting. It specifies the functions during each phase, and component information such as server, interface, database, device, etc. This schema is based on the complex type SystemStructure.

6.6 140-Data

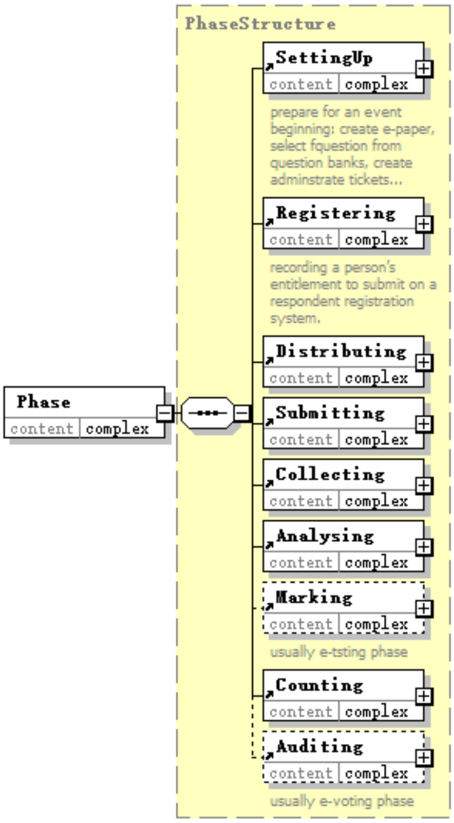


Generated by XMLSpy www.altova.com

Fig. 6-5 Data Diagram

This schema is used for specify the detailed information for data of e-questionnaire, e-testing, and e-voting systems. This provides the specification of e-paper, participant information, response, and report in detail, for data exchange and reusability. This schema is based on the complex type DataStructure.

6.7 210-Phase

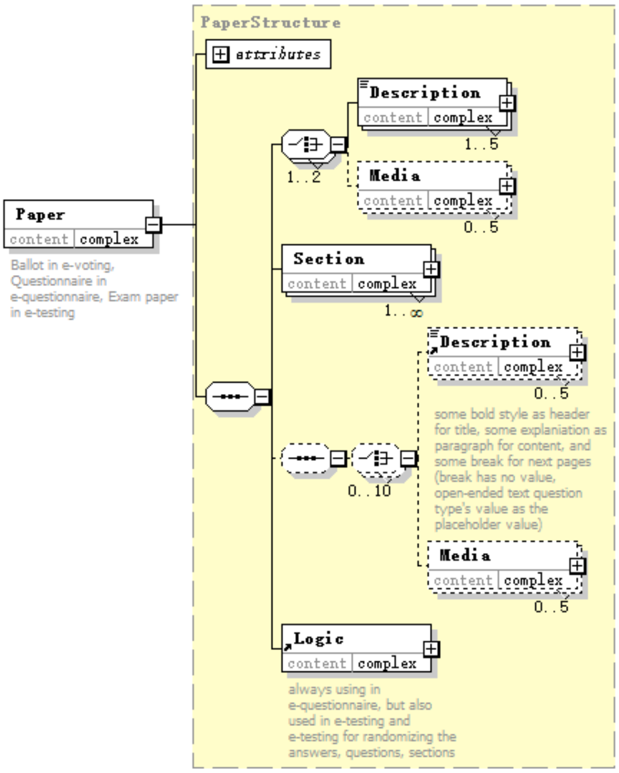


Generated by XMLSpy www.altova.com

Fig. 6-6 Phase Diagram

This schema is used for specify the functions during each phase. This schema is based on the complex type PhaseStructure.

6.8 220-Paper


PaperStructure
attributes
Description
content complex
1..5
1..2
Media
content complex
0..5
Paper
content complex
Ballot in e-voting, Questionnaire in e-questionnaire, Exam paper in e-testing
Section
content complex
1..∞
Description
content complex
0..5
some bold style as header for title, some explaniation as paragraph for content, and some break for next pages (break has no value, open-ended text question type's value as the placeholder value)
0..10
Media
content complex
0..5
Logic
content complex
always using in e-questionnaire, but also used in e-testing and e-testing for randomizing the answers, questions, sections

Generated by XMLSpy www.altova.com

Fig. 6-7 Paper Diagram

The child element of **PaperSheet** (schema 120), it is the core component for an e-questionnaire and e-testing, as we all known that they consists of questions. Questioner and sponsor design a paper with lots of questions. It is the precondition of an event. Detailed information please refer to PaperStructure.

6.9 230-Setting

Setting is the schema 230 based on SettingStructure complex type. There are 12 child elements. Due to the space restriction, we divide each child element to explain.

6.9.1 Language

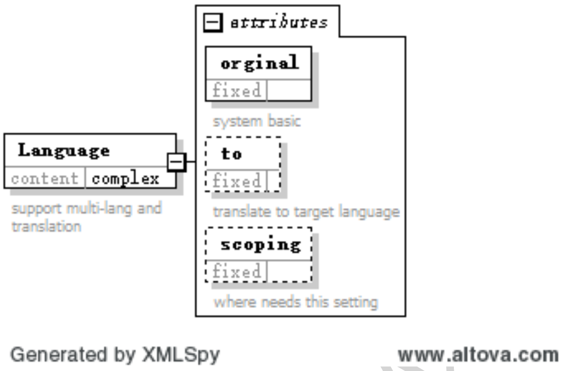


Fig. 6-8 Language Diagram

Element	Attribute	Type	Use	Comment
Language	original	LanguageType	required	
	to	LanguageType	optional	translate to
	scoping	Union: PaperType& DataType& SystemType	optional	if need translate; value: system, paper, response, report

This element defines the original and translation language in which scope.

6.9.2 Checking

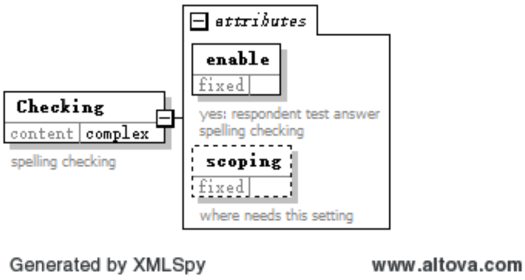
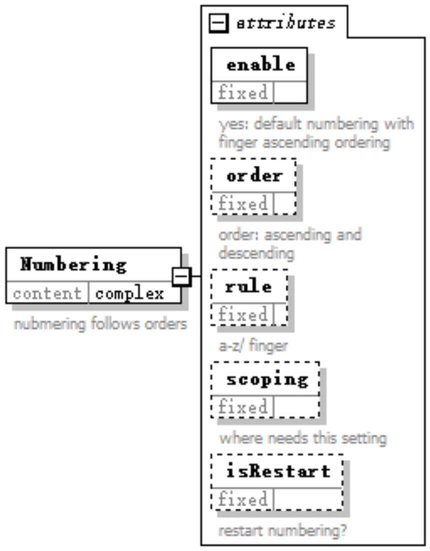


Fig. 6-9 Checking Diagram

Element	Attribute	Type	Use	Comment
Checking	enable	YesNoType	required	
	scoping	PaperType	optional	value: answer

This element defines the spelling checking setting for providing conveniences for respondents to answer questions.

6.9.3 Numbering



Generated by XMLSpy www.altova.com

Fig. 6-10 Numbering Diagram

Element	Attribute	Type	Use	Comment
Numbering	enable	YesNoType	required	
	order	OrderType	optional	default: ascending
	rule	LimtitationType	optional	
	scoping	union: PaperType & DataType	optional	value: paper, section, question, answer, response, report
	isRestart	YesNoType	optional	default: no

6.9.4 Time

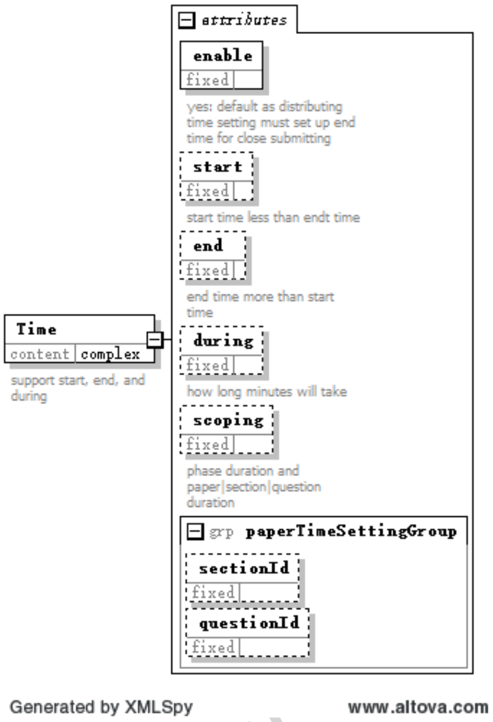

attributes
enable
fixed
yes: default as distributing time setting must set up end time for close submitting
start
fixed
start time less than endt time
end
fixed
end time more than start time
Time
content complex
support start, end, and during
during
fixed
how long minutes will take
scoping
fixed
phase duration and paper|section|question duration
grp paperTimeSettingGroup
sectionId
fixed
questionId
fixed
Generated by XMLSpy
www.altova.com

Fig. 6-11 Time Diagram

Element	Attribute	Type	Use	Comment
Time	enable	YesNoType	required	
	start	TimeType	optional	*during* exclusive
	end	TimeType	optional	
	during	xs:double	optional	*start|end* exclusive
	scoping	union: PhaseType &PaperType	optional	
paperTimeSettingGroup	sectionId	IdType	optional	
	questionId	IdType	optional	

6.9.5 Number

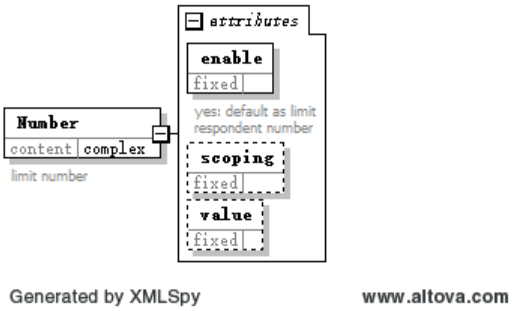


Fig. 6-12 Number Diagram

Element	Attribute	Type	Use	Comment
Number	enable	YesNoType	required	limit how many respondents can submit
	scoping	union: ParticipantType &DataType	optional	value: respondent, response
	value	xs:nonNegativeInteger	optional	*scoping* inclusive

6.9.6 Quota

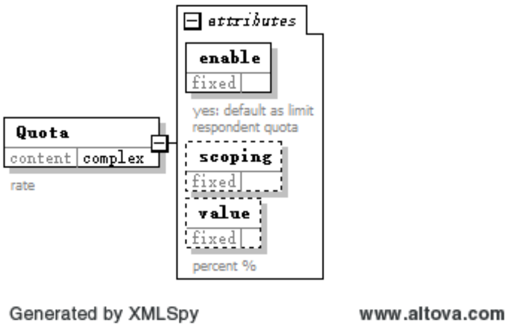


Fig. 6-13 Quota Diagram

Element	Attribute	Type	Use	Comment
Quota	enable	YesNoType	required	limit how much percents of respondents can submit
	scoping	union: ParticipantType &DataType	optional	value: respondent, response
	value	xs:nonNegativeInteger	optional	*scoping* inclusive

6.9.7 Interval

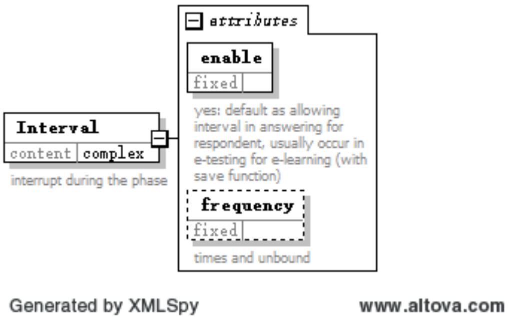

attributes
enable
fixed
yes: default as allowing interval in answering for respondent, usually occur in e-testing for e-learning (with save function)
frequency
fixed
times and unbound
Interval
content complex
interrupt during the phase
Generated by XMLSpy
www.altova.com

Fig. 6-14 Interval Diagram

Element	Attribute	Type	Use	Comment
Interval	enable	YesNoType	required	interrupt during the phase
	frequency	union: xs:nonNegativeInteger & xs:token (bound)	optional	

6.9.8 AutoSaving

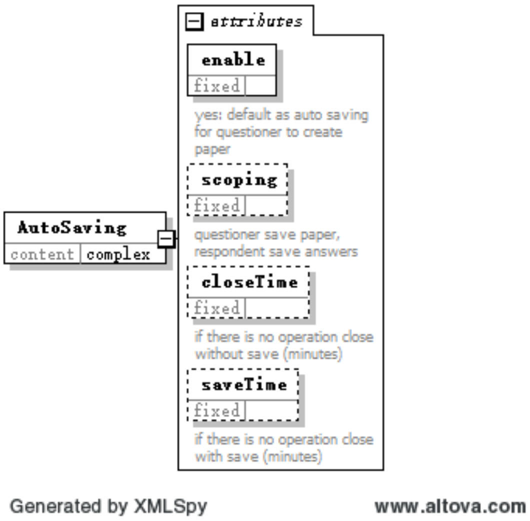

attributes
enable
fixed
yes: default as auto saving for questioner to create paper
scoping
fixed
questioner save paper, respondent save answers
closeTime
fixed
if there is no operation close without save (minutes)
saveTime
fixed
if there is no operation close with save (minutes)
AutoSaving
content complex
Generated by XMLSpy
www.altova.com

Fig. 6-15 AutoSaving Diagram

Element	Attribute	Type	Use	Comment
AutoSaving	enable	YesNoType	required	
	scoping	PhaseType	optional	
	closeTime	xs:nonNegativeInteger	optional	*saveTime* exclusive
	saveTime	xs:nonNegativeInteger	optional	*closeTime* exclusive

6.9.9 Distribution

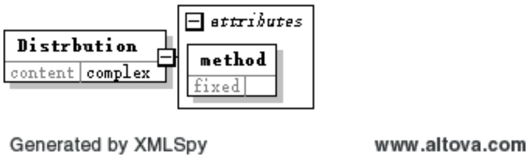


Fig. 6-16 Distribution Diagram

Element	Attribute	Type	Use	Comment
Distribution	method	ChannelType	required	

6.9.10 Reporting

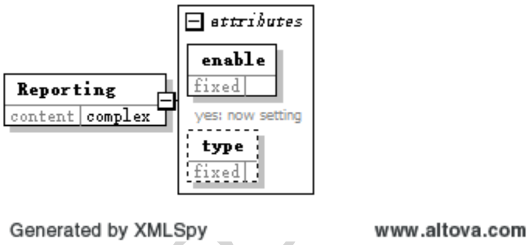


Fig. 6-17 Reporting Diagram

Element	Attribute	Type	Use	Comment
Reporting	enable	YesNoType	required	
	type	ReportType	optional	

6.9.11 Media

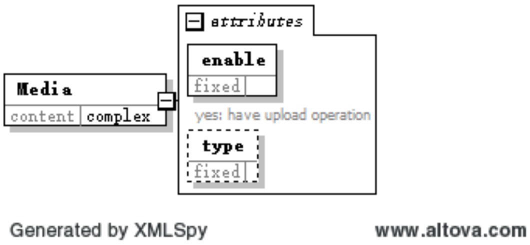


Fig. 6-18 Media Diagram

Element	Attribute	Type	Use	Comment
Media	enable	YesNoType	required	

	type	MediaType	optional	

6.9.12 Reminder

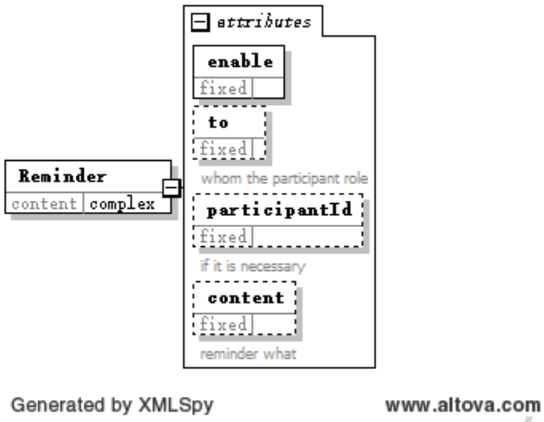


Fig. 6-19 Reminder Diagram

Element	Attribute	Type	Use	Comment
Reminder	enable	YesNoType	required	
	to	ParticipantType	optional	notify whom (role)
	participantId	IdType	optional	notify whom (in detail)
	content	ReminderType	optional	notify values

6.10 240-Component

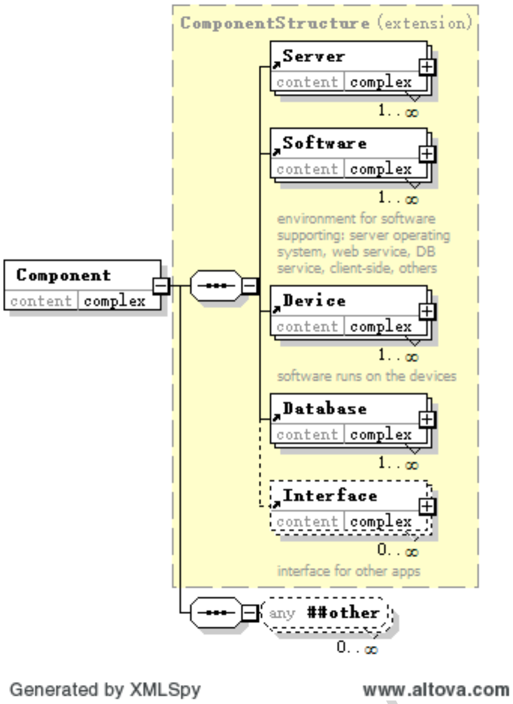


Fig. 6-20 Component Diagram

6.10.1 Description of Schema

This schema is used for specify the components especially for the environments of the e-questionnaire, e-testing, e-voting. This schema is based on the complex type ComponentStructure.

6.11 250-Participant

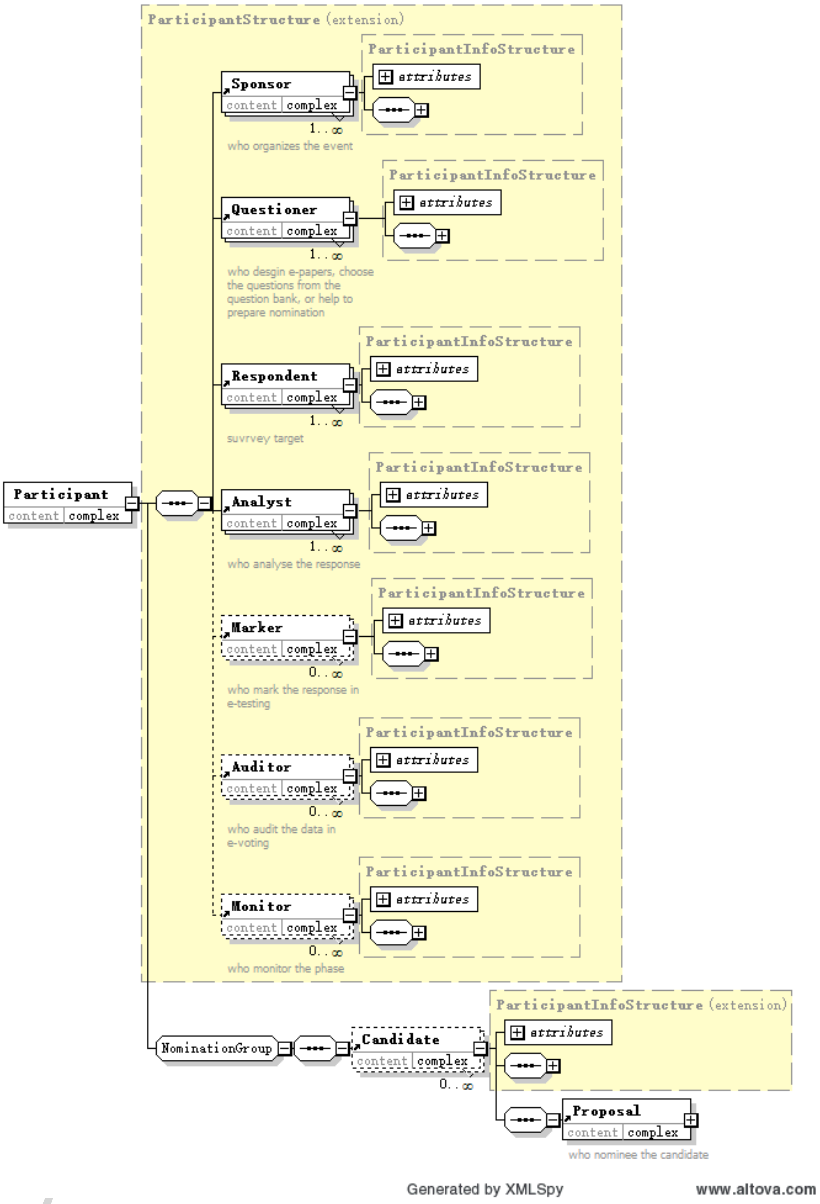


Fig. 6-21 Participant Diagram

6.11.1 Description of Schema

This schema is used for specify the participant in an event. This schema is based on the complex type ParticipantStructure.

6.12 260-Response

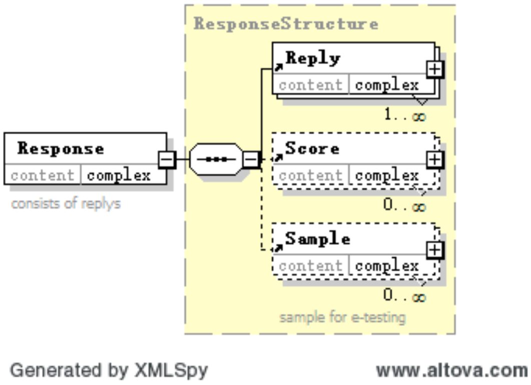


Fig. 6-22 Response Diagram

Response data consists of each respondent's reply, and score if the respondent took part in a testing event, and the questioner provides the sample answer. Element **Response** refers to ResponseStructure. We introduce child element **Reply** in details. As to **Score** and **Sample**, please refer to schema 420 and schema 430.

6.12.1 Reply

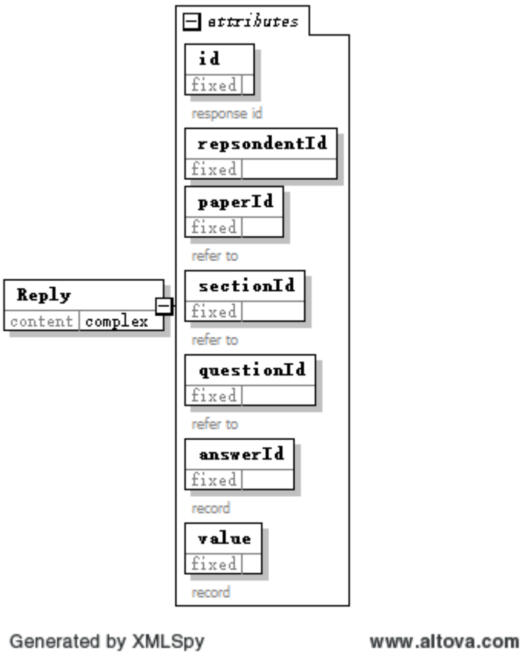


Fig. 6-23 Reply Diagram

It provides the specification for response data in details.

Element	Attribute	Type	Used	Comment
Reply	id	IdType	required	
	respondentId	IdType	required	
	paperId	IdType	required	
	sectionId	IdType	required	
	questionId	IdType	required	
	answerId	IdType	required	
	value	xs:string	required	

6.13 270-Report

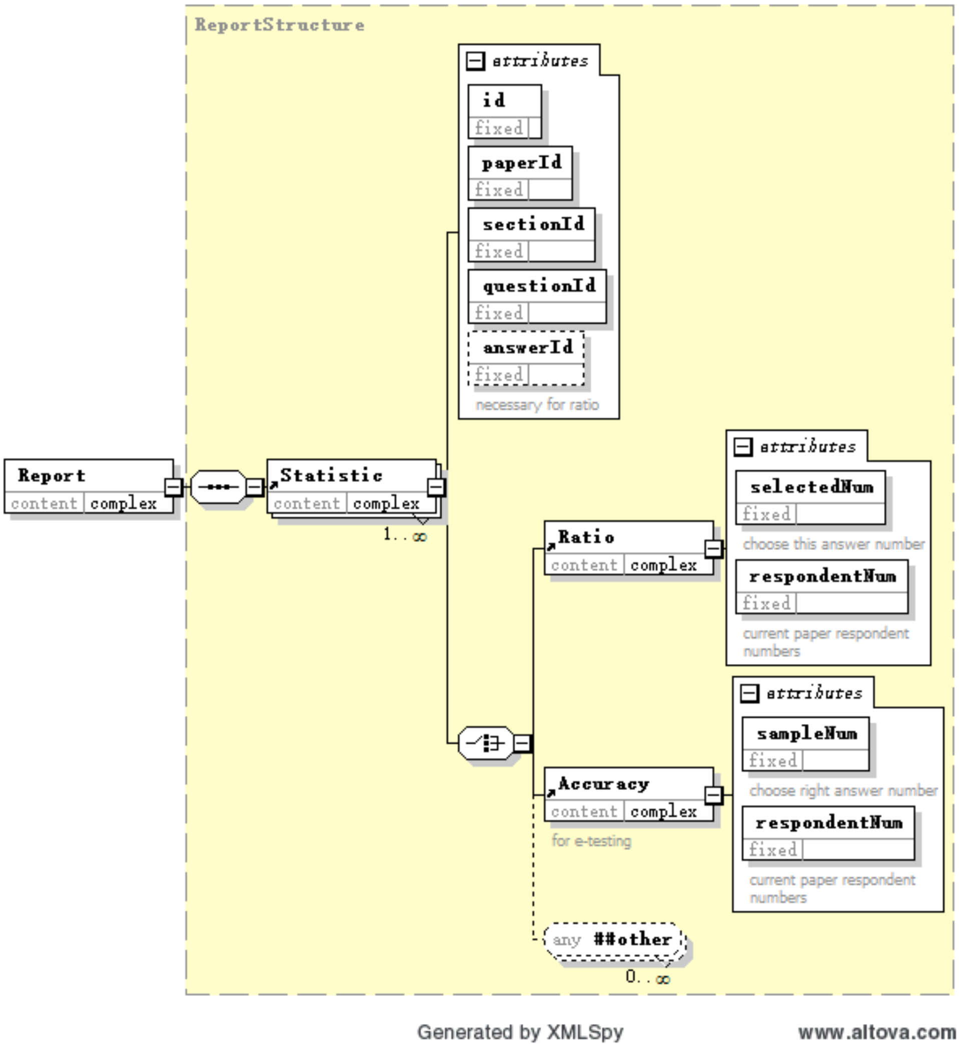


Fig. 6-24 Report Diagram

Please refer to ReportStructure. **Report** is used to define the report data in details. Its child element **Statistic** used to specify a lots of report style. It determines by setting for report types. This provides the reference to detailed answers identifier, and provides the **Ratio** to completeness and **Accuracy** for e-testing. Both child elements of statistic are in a choice order. As to other element for extension.

6.14 310-Logic

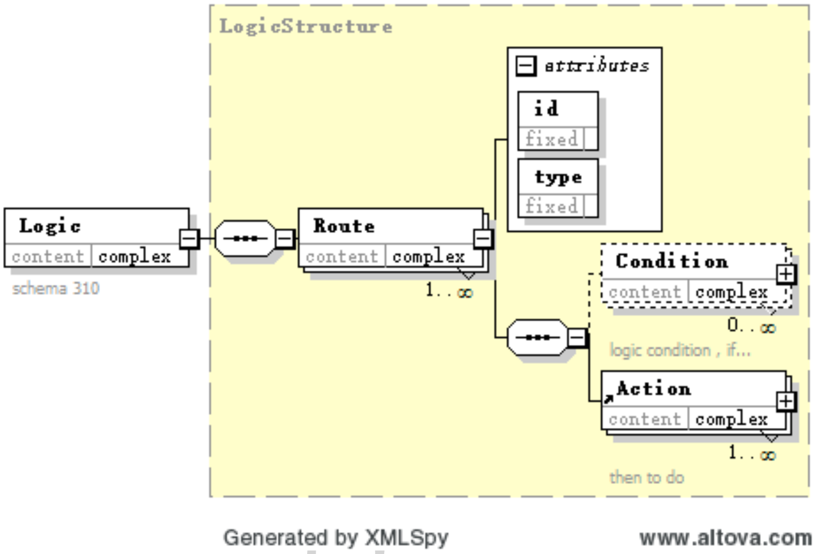


Fig. 6-25 Logic Diagram

Please refer to LogicStructure. **Logic** has a child element **Route** to record the routes of the logic from which **Condition** and go to which **Action**.

Much more logic types and the templates for guiding how to use it in details are listed in Chapter 8.

6.15 410-Marking

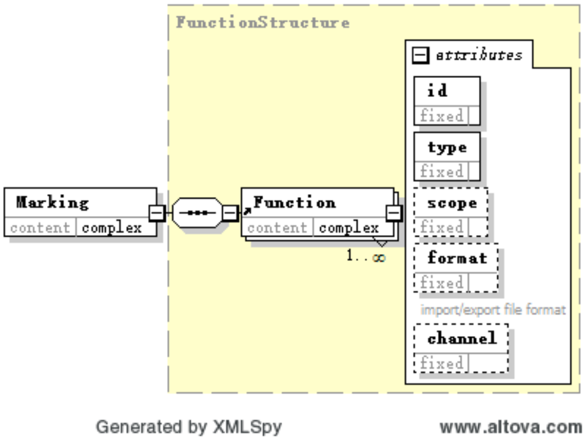


Fig. 6-26 Marking Diagram

Marking is the phase in e-testing. This schema is based on FunctionStructure.

6.15.1 Marking Derivatives

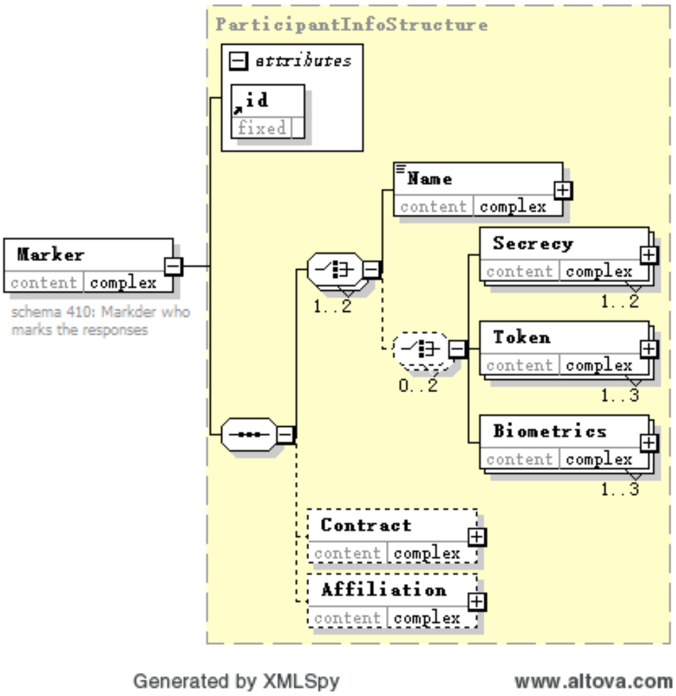


Fig. 6-27 Marker Diagram

Designing QSL is based on the primitive elements. Before figuring out primitive elements, we found the similarities and differences of e-questionnaire, e-testing, and e-voting systems, and concluded that the lots of items are based on the phases.

Marking is the phase in e-testing. According to the combination regulation, it combine with participant can derive a element named **Marker**, who can marks the responses. This element is the child element of **Participant**, and based on ParticipantInfoStructure. In addition, the simple type PhaseType has a value named *marking*. The simple type ParticipantType has a value named *marker*.

6.16 420-Score

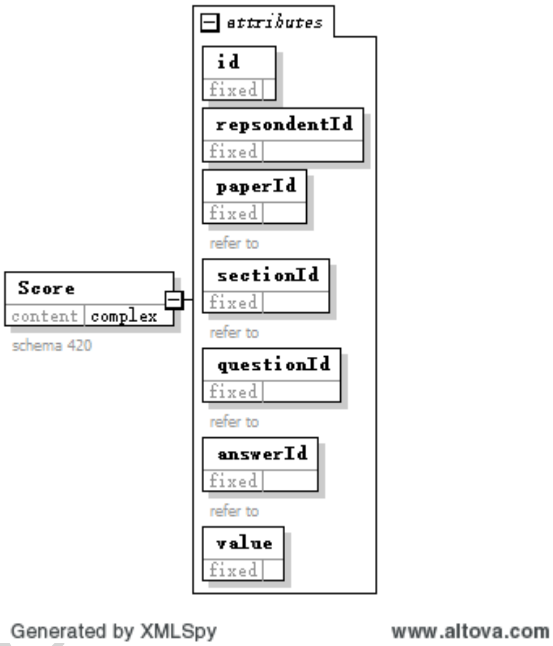


Fig. 6-28 Score Diagram

Schema 420 provides the specification for score data in details. As to the score setting rule is specified in schema 230.

Element	Attribute	Type	Used	Comment
Score	id	IdType	required	
	respondentId	IdType	required	
	paperId	IdType	required	
	sectionId	IdType	required	
	questionId	IdType	required	
	answerId	IdType	required	
	value	union: xs:double& xs:token	required	score

6.17 430-Sample

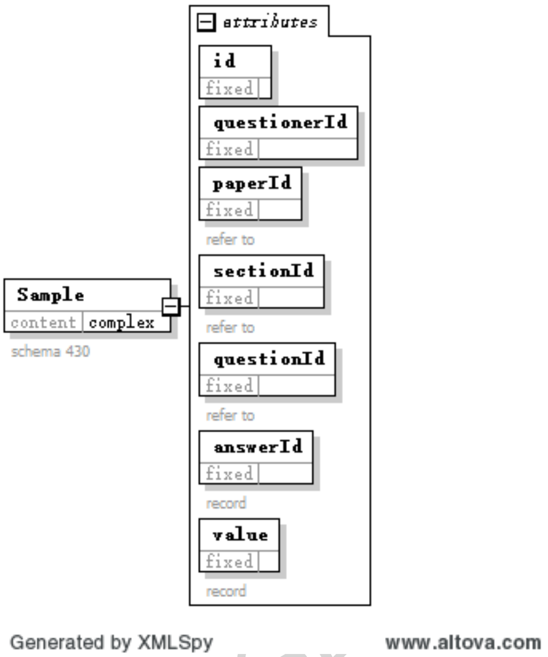


Fig. 6-29 Sample Diagram

Schema 430 provides the specification for sample data in details. It only occurs in e-testing.

Element	Attribute	Type	Used	Comment
Sample	id	IdType	required	
	respondentId	IdType	required	
	paperId	IdType	required	
	sectionId	IdType	required	
	questionId	IdType	required	
	answerId	IdType	required	
	value	xs:token	required	sample answer

6.18 440-Formula

Now, formula is the necessary item always used in e-testing. There is a XML-based Language named MathML to specify the mathematic. Our developed offline e-testing environment based on QSL and borrowed MathML to support math test during designing questions.

6.19 510-Authentication

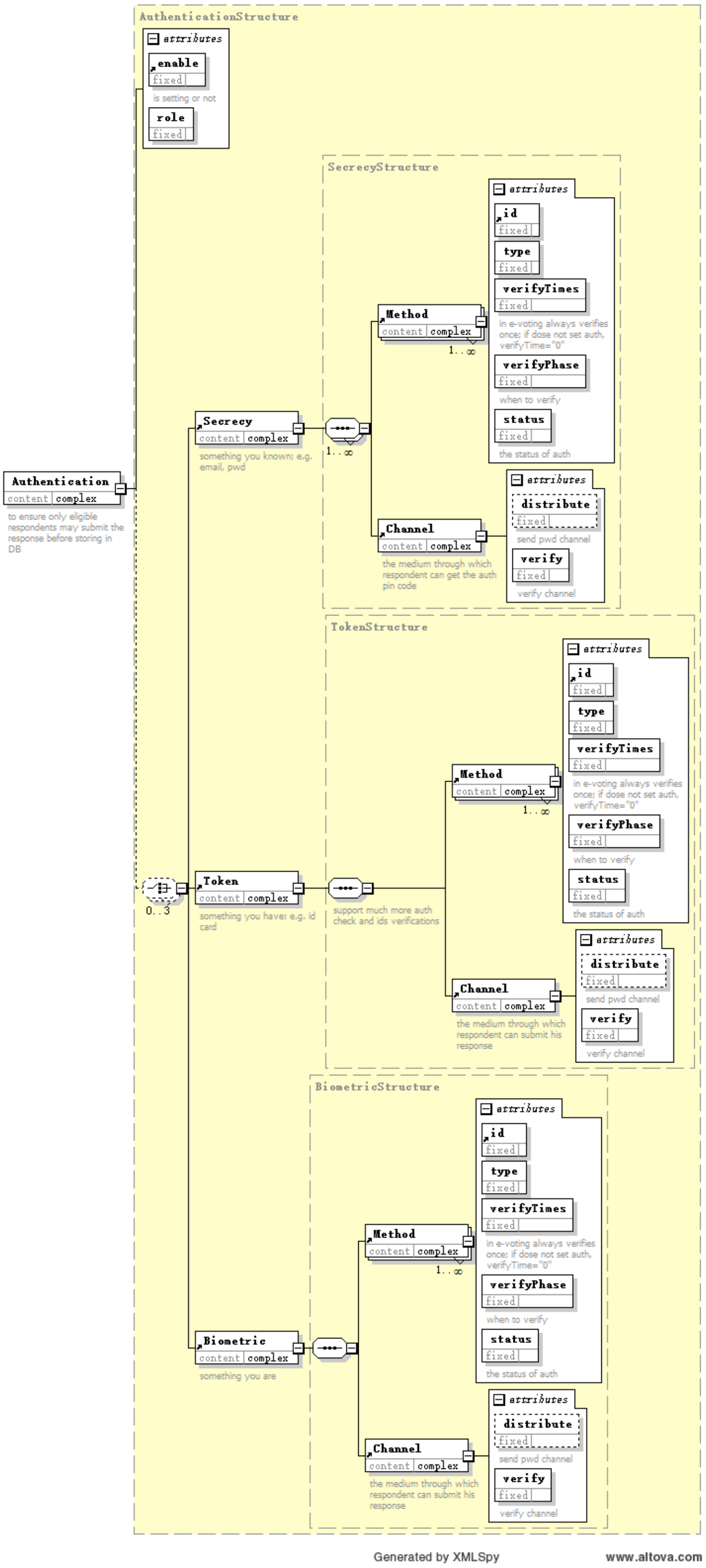


Fig. 6-30 Authentication Diagram

Please refer to complex types: AuthenticationStructure, SecrecyStructure, TokenStructure, BiometricStructure.

6.20 520-Anonymity

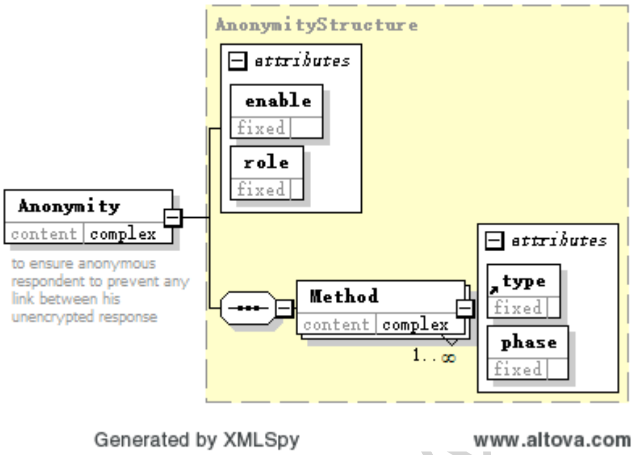


Fig. 6-31 Anonymity Diagram

Please refer to AnonymityStructure.

6.21 530-Auditing

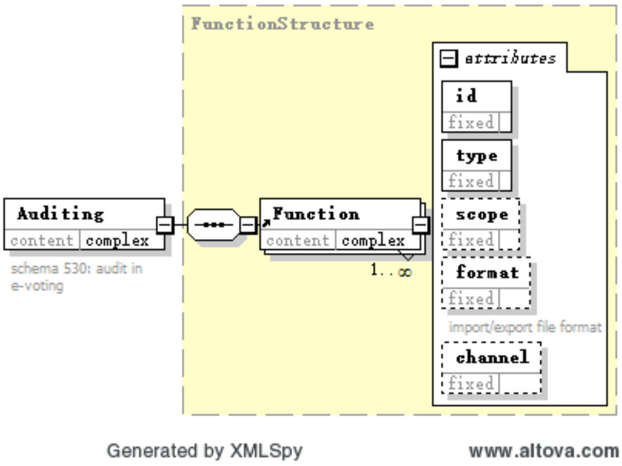


Fig. 6-32 Auditing Diagram

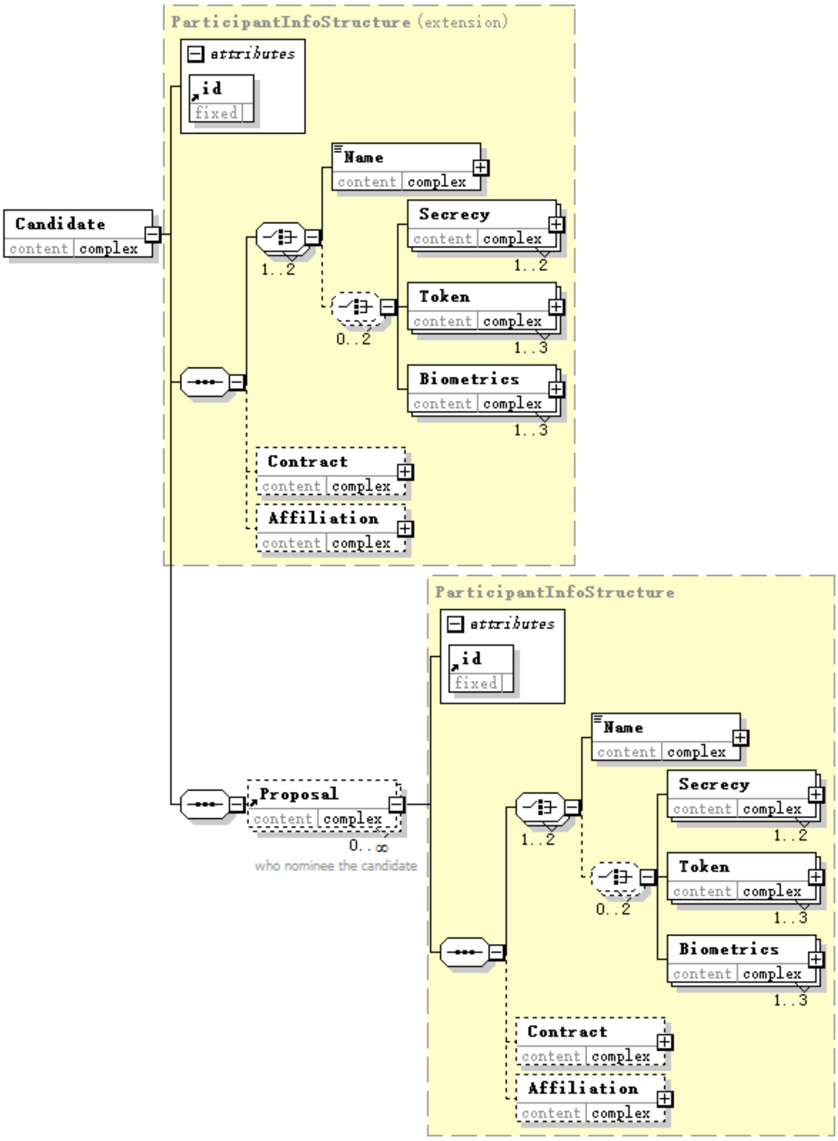
Auditing is the last phase in e-voting, especially in election. This schema is based on FunctionStructure.

6.21.1 Auditing Derivatives

Designing QSL is based on the primitive elements. Before figuring out primitive elements, we found the similarities and differences of e-questionnaire, e-testing, and e-voting systems, and concluded that the lots of items are based on the phases.

Auditing is the last phase in e-voting. According to the combination regulation, it combine with participant can derive a element named **Auditor**, who can audit the results. This element is the child element of **Participant**, and based on ParticipantInfoStructure. In addition, the simple type PhaseType has a value named *audit*. The simple type ParticipantType has a value named *auditor*.

6.22 540-Candidate



Generated by XMLSpy www.altova.com

Fig. 6-33 Candidate Diagram

Candidate often occurs in an election, not only in a simple voting event. In this schema, **Candidate** is based on extension of ParticipantInfoStrucutre, and it has an optional child element named Proposal who nominate the candidate, always parties and organizations, as well as someone respectability. In this situation, it doesn't need an authentication for proposal.

7 Question Types Reference Guide

The whole things of QSL: QSL structure, QSL Schema, elements, attributes, simple types, and complex types are listed previously. In this chapter, we will see how we can use the important but difficult parts of QSL usage: the question types which determine a complete e-questionnaire, and an e-testing. The reason is that an e-questionnaire or an e-testing is mainly composed of some questions.

Although many e-questionnaire and e-testing systems provide widely variety of question types particularly in the various classification methods, we concluded four basic types and aimed at gaining various question types though combination of basic types.

We generally talked about the type attribute of question element in previous chapter. Not enough is known about type attribute to use it well. Therefore, we will deeply illustrate the question types and how to use them.

This chapter provides a reference to all of question types as follow:

- Basic question types;
- Possible combinations;

All content and diagrams in this chapter refer to QuestionPro[1].

7.1 Basic Question Types

Four basic question types are listed as follow:

- Multiple Choice;
- Open-ended Text;
- Matrix;
- Ranking;

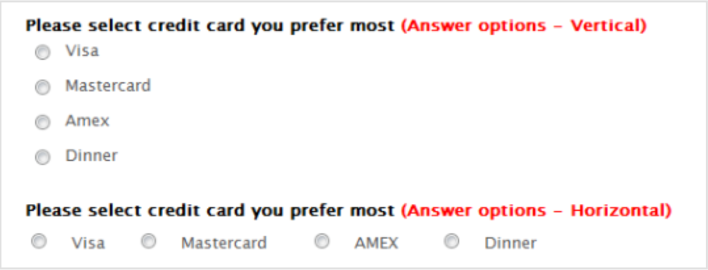
7.1.1 Multiple Choice

What is multiple choice question type?

Multiple Choice is tacit recognition as a radio button or a checkbox that that allows the respondents to choose only one or multiple selections from a predefined set of options. In general, there are two types of multiple choice question type, which are single choice and multiple choices. Some screenshots

[1] Question Pro is an online survey software tool, one of the most popular online survey websites all over the world.

help to deeply understand it.



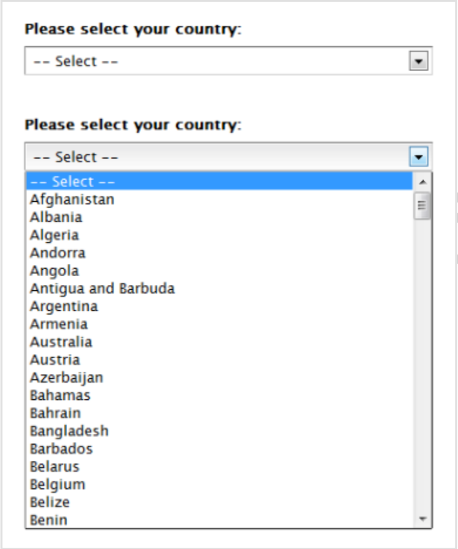
Please select credit card you prefer most (Answer options – Vertical)

○ Visa
○ Mastercard
○ Amex
○ Dinner

Please select credit card you prefer most (Answer options – Horizontal)

○ Visa ○ Mastercard ○ AMEX ○ Dinner

Fig. 7-1 Screenshots of Radio Button Single Choice



Please select your country:

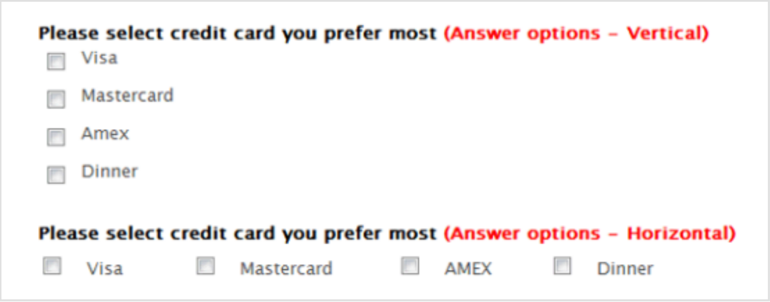
-- Select --

Please select your country:

-- Select --

-- Select --
Afghanistan
Albania
Algeria
Andorra
Angola
Antigua and Barbuda
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin

Fig. 7-2 Screenshots of Drop-down Single Choice



Please select credit card you prefer most (Answer options – Vertical)

□ Visa
□ Mastercard
□ Amex
□ Dinner

Please select credit card you prefer most (Answer options – Horizontal)

□ Visa □ Mastercard □ AMEX □ Dinner

Fig. 7-3 Screenshots of Checkbox Multiple Choices

How to distinguish single choice and multiple choices?

With Designing QSL based on combination of primitive elements, both single choice and multiple choices are the multiple choices question type (Drop-down list belongs to single choice). We distinguish both of them by **Limitation**[1].

[1] Bold contents in this chapter are defined as QSL elements.

How to set up multiple choice question type?

Step1: to choose question type for **Question**, which has an attribute *type*[1] and to choose its value "selection".

Step 2: to distinguish the single choice and multiple choices, and to control it by **Limitation**.

- **Tips 1:** As to a radio button single choice type, questioner should choose the **Limitation** *minOccur* attribute value as "1", *macOccur* attribute value as "1", which means to restrict respondent to select only one answer.
- **Tips 2:** As to a checkbox multiple choices type, questioner should choose the **Limitation** *minOccur* attribute value more than 1, and *macOccur* attribute value less than total number of answers, which means to restrict respondent to select multiple answers more than one less than total number of answers.

Step 3: to set up **Answer**, which has its child element **Description**, which should be written the content of each answer between start-tag and end-tag of text.

As an example to let you deeply understand how to set up a multiple choice question type (radio button single choice),

```
<Question isMandatory="yes" type="selection" id="q0001">
    <Limitation minOccur="1" maxOccur="1"></Limitation>
    <Description type="paragraph" value="Please selected credit card u prefer most"
id="de0001"/>
    <Answer id="a0001">
        <Description type="paragraph" value="Visa" id="de0002"/>
    </Answer>
    <Answer id="a0002">
        <Description type="paragraph" value="Visa" id="de0003"/>
    </Answer>
    <Answer id="a0003">
        <Description type="paragraph" value="Visa" id="de0004"/>
    </Answer>
    <Answer id="a0004">
        <Description type="paragraph" value="Visa" id="de0005"/>
    </Answer>
</Question>
```

To distinguish the arrangement of Fig. 7-1 and Fig. 7-2, we can set up **Alignment** element.

```
<Question isMandatory="yes" type="selection" id="q0001">
    <Limitation minOccur="1" maxOccur="1"></Limitation>
    <Description type="paragraph" value="Please selected credit card u prefer most"
id="de0001"/>
    <Alignment vertical="center"/>
```

[1] Italics content in this chapter are defined as QSL attributes.

```
    <Answer id=″a0001″>...

<Alignment dropDown=″listBox″>
```

As another example to let you deeply understand how to set up a multiple choice question type (checkbox multiple choices),

```
<Limitation minOccur=″1″ maxOccur=″4″></Limitation>
<!--at least choose 1 option and at most choose 4 options-->
<Limitation minOccur=″3″ maxOccur=″3″></Limitation>
<!--Exactly choose 3 options-->
```

To distinguish the arrangement of Fig.7-3, we can set up **Alignment**.
Attention: there is no need to set up a drop-down list for a checkbox selection.

```
<Alignment horizontal=″left″/>
```

There is a special question type named "True-False". The template snippet is shown below.

```
<Question isMandatory=″yes″ type=″selection″ id=″q0001″>
    <Limitation minOccur=″1″ maxOccur=″1″></Limitation>
    <Description type=″paragraph″ value=″True-False″ id=″de0001″/>
    <Answer id=″a0001″>
        <Description type=″paragraph″ value=″true″ id=″de0002″/>
    </Answer>
    <Answer id=″a0002″ isDefault=″yes″>
    <!--sample answer-->
        <Description type=″paragraph″ value=″false″ id=″de0003″/>
    </Answer>
    <Answer id=″a0002″ isOtherField=″a0003″ score=″-0. 5″>
    <!--allow respondent to add a other answer, usually in e-questionnaire -->
    <!--if respondent adds other answer, then minus 0. 5 score-->
        <Description type=″paragraph″ value=″other″ id=″de0004″/>
    </Answer>
</Question>
```

7.1.2 Open-ended Text

What is open-ended text question type?
Open-ended Text is a text type question where respondents can input long text, single row text, numeric text, or an email address.

How to distinguish long text, single row text, numeric text, email address?
Because Design of QSL is based on combination of primitive elements, so we

proposed that all of these types are the Open-ended Text Question Type. To help understand deeply, Fig. 7-4, Fig. 7-5, Fig. 7-6, and Fig. 7-7 are presented below.

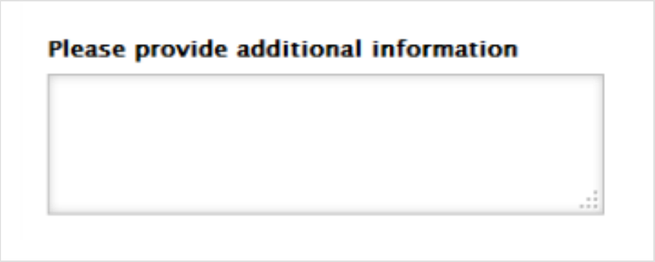


Fig. 7-4 Screenshot of Long Text

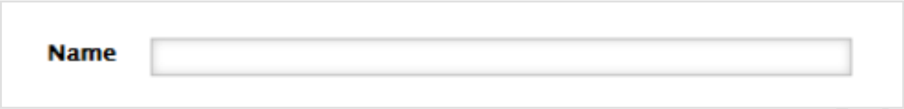


Fig. 7-5 Screenshot of Single Row Text

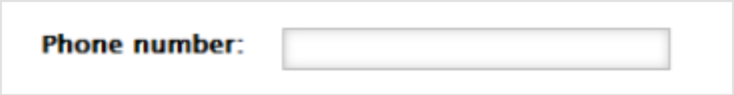


Fig. 7-6 Screenshot of Numeric Text

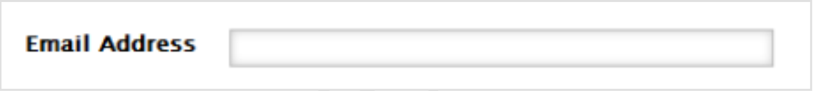


Fig. 7-7 Screenshot of Email Address

Questioners can use **Limitation** element to restrict.

How to set up open-ended text question type?

At first, questioner should choose the question type for the **Question** element, which has an attribute *type*. Questioner should choose the type attribute value "text". Attention: If the character text is presented up above the input box, please write in the **Description** element of **Question** element. If the character text is presented in placeholder of the input box, please write in the **Description** element of **Answer** element.

In addition, to set up the **Answer** element, **Answer** element has its child element **Description**.

At last, to distinguish the long text, single row text, numeric text, and email address, questioner can control it through **Limitation** element. However, it is different with Multiple Choice Question Type, the **Limitation** element is as a child element of **Answer** element.

- As to a long text type (developer should set up the default size as 80px×320px), questioner should choose the limitation attribute *minLength* value and *maxLength* value less than "255"as default.
- As to a single row text type (developer should set up the default size as

400px, but height cannot be increased), questioner should choose the **Limitation** *minLength* value and *maxLength* value less than ”255” as default.

- As to a numeric text type (developer should set up the default size as 80px, but height cannot be increased), questioner should choose the **Limitation** attribute *minLength* value and *maxLength* value less than ”255”as default.
- As to an email address type (developer should set up the default size as 320px, but height cannot be increased), questioner should choose the **Limitation** attribute. In common, the *most* attribute value and the *minLength* attribute are not necessary.
- There are 2 **Limitation** locations, first, as the **Question**'s child element, second, as the **Answer**'s child element. The first one in the outer layer to control line numbers. The second one in the inner layer to control character numbers, and the *format* value to control content format.

As an example to let you deeply understand how to set up a long text type,

```
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="0" maxLength="5"/>
    <!--outer layer: limitation control line numbers -->
    <Description type="paragraph" value="Please provide additional information"
id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="5" maxLength="255" format="long text"/>
        <Description type="paragraph" value="" id="de0002"/>
        <!--inner layer: limitation control character numbers -->
    </Answer>
</Question>
```

As an example to let you deeply understand how to set up a single row text type,

```
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1"/>
    <!--outer layer: limitation control line numbers -->
    <Description type="paragraph" value="Name:" id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="2" maxLength="129" format="char"//>
        <Description type="paragraph" value="" id="de0002"/>
        <!--inner layer: limitation control character numbers -->
    </Answer>
</Question>
```

As an example to let you deeply understand how to set up a numeric input type,

```
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1"/>
    <!--outer layer: limitation control line numbers -->
    <Description type="paragraph" value="Phone Number:" id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="5" maxLength="25" format="int"//>
        <Description type="paragraph" value="" id="de0002"/>
        <!--inner layer: limitation control character numbers -->
    </Answer>
</Question>
```

As an example to let you deeply understand how to set up an email address type,

```
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1"/>
    <!--outer layer: limitation control line numbers -->
    <Description type="paragraph" value="Email:" id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="10" maxLength="129" format="char"//>
        <Description type="paragraph" value="" id="de0002"/>
        <!--inner layer: limitation control character numbers -->
    </Answer>
</Question>
```

There is a special question type named "fill in the blank". The usual presentation in HTML is:`<p>my name is<input type="text">, I am a student</p>`

The template snippet is shown below. It concerns the piping logic type and its grammar.

```
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1"/>
    <!--outer layer: limitation control line numbers -->
    <Description type="paragraph" value="My name is $(’ a0001’), I am a student"
id="de0001"/>
    <Answer id="a0001">
        <Limitation minLength="2" maxLength="5" format="char"//>
        <Description type="paragraph" value="" id="de0002"/>
        <!--inner layer: limitation control character numbers -->
    </Answer>
</Question>
```

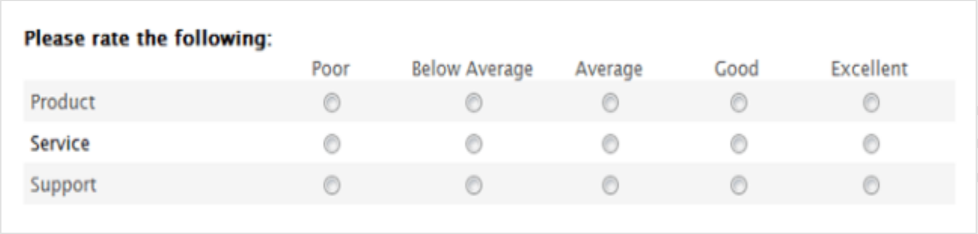
7.1.3 Matrix

What is matrix question type?

Matrix Question Type is a series of questions that share the same answer choices. There are two types, which are one selection type and many selections type.

How to distinguish one selection type and many selections type?

One Selection Type: This question can be used when we need to group questions that have the same answer option scale. Consolidated reporting can be done on Matrix questions (see Fig. 7-8).

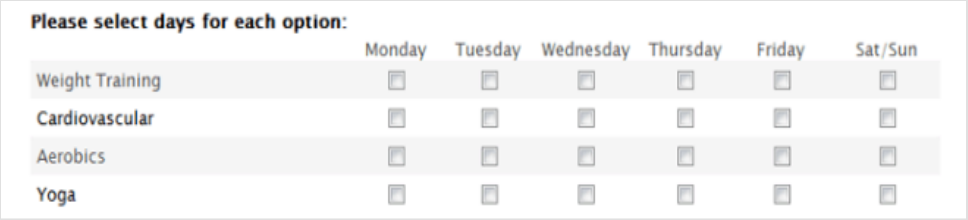


Please rate the following:

	Poor	Below Average	Average	Good	Excellent
Product	○	○	○	○	○
Service	○	○	○	○	○
Support	○	○	○	○	○

Fig. 7-8 Screenshot of One Selection Type

Many Selection Type: Matrix question where respondents can select multiple options (see Fig. 7-9).



Please select days for each option:

	Monday	Tuesday	Wednesday	Thursday	Friday	Sat/Sun
Weight Training	□	□	□	□	□	□
Cardiovascular	□	□	□	□	□	□
Aerobics	□	□	□	□	□	□
Yoga	□	□	□	□	□	□

Fig. 7-9 Screenshot of Many Selections Type

How to set up a matrix question type?

At first, questioner should choose the question type for the **Question** element, which has an attribute "*type*". Questioner should choose the type attribute value "matrix".

In addition, to distinguish the one selection and many selections, questioner can control it through **Limitation** element.

- As to a one selection type, questioner should choose the **Limitation** *maxOccur* attribute value as "1", *minOccur* attribute value as "1", which means to restrict respondent to select only one option in each row.
- As to a checkbox multiple choices type, questioner should define the **Limitation** *maxOccur* attribute value not less than 1, *minOccur* attribute value less than total number of answers, which means to restrict respondent to select multiple answers more than one less than total number of options in each row.

At last, to set up the **Answer** element, **Answer** element has its child element **Description**, which should be written the *value*. In addition, its attributes *Column.Grid* and *Row.Grid* to control the descriptions of column and row.

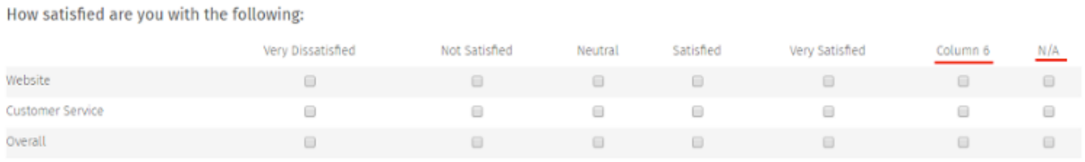
As an example to let you deeply understand how to set up one selection type,

```
<Question isMandatory=″yes″ type=″matrix″ id=″q0001″>
    <Limitation minOccur=″1″ maxOccur=″1″/>
    <Description type=″paragraph″ value=″Please rate the following″ id=″de0001″/>
    <!--Column-->
    <Answer id=″a0001″ Column.Grid=″0″>
    <!--Grid begins from 0 -->
        <Description type=″paragraph″ value=″Poor″ id=″de0002″/>
    </Answer>
    <Answer id=″a0002″ Column.Grid=″1″>
        <Description type=″paragraph″ value=″Below Average″ id=″de0003″/>
    </Answer>
    ...
    <!--Rows-->
    <Answer id=″a0003″ Row.Grid=″0″>
        <Description type=″paragraph″ value=″Product″ id=″de0004″/>
    </Answer>
    <Answer id=″a0004″ Row.Grid=″1″>
        <Description type=″paragraph″ value=″Service″ id=″de0005″/>
    </Answer>
    <Answer id=″a0005″ Row.Grid=″2″>
        <Description type=″paragraph″ value=″Support″ id=″de0006″/>
    </Answer>
</Question>
```

As an example to let you deeply understand how to set up many selections type, it only need to change the **Limitation**'s attributes.

```
<Limitation minOccur=″1″ maxOccur=″3″/>
```

There is a special example that is a Not Applicable Option.



How satisfied are you with the following:

	Very Dissatisfied	Not Satisfied	Neutral	Satisfied	Very Satisfied	Column 6	N/A
Website	□	□	□	□	□	□	□
Customer Service	□	□	□	□	□	□	□
Overall	□	□	□	□	□	□	□

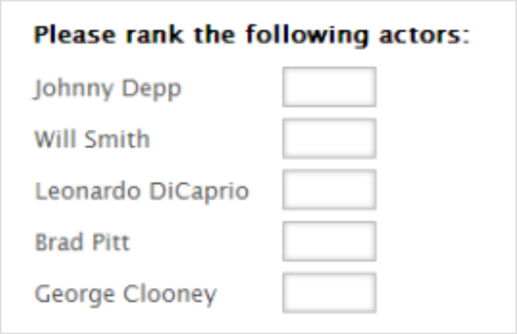
Fig. 7-10 Screenshot of N/A Option in Matrix

```
<Answer id=″a0002″ Column.Grid=″6″ isNAOption=″yes″>
<!--set NA Option-->
    <Description type=″paragraph″ value=″ ″ id=″de0007″/>
    <!--omit the value, automatically populate-->
</Answer>
```

7.1.4 Ranking

What is ranking question type?

Ranking Question Type allows a certain set of brands or products to be ranked based upon a specific attribute or characteristic. In common, some e-questionnaire systems present rank order type (see Fig. 7-11), which is similar with numeric text type to enter the order number.



Please rank the following actors:

Johnny Depp □

Will Smith □

Leonardo DiCaprio □

Brad Pitt □

George Clooney □

Fig. 7-11 Screenshot of Ranking Question Type

How to set up a ranking question type?

At first, questioner should choose the question type for the **Question** element, which has an attribute "*type*". Questioner should choose the type attribute value "ranking". Attention: If the value is selected as "ranking" that means the order number cannot be repeated and from 1 to the number of total options. In addition, questioner can use **Limitation** element to restrict *maxOccur* and *minOccur*.

Furthermore, to set up the **Answer** element, **answer** element has its child element **text**, which should be written the content of each answer between start-tag and end-tag of **text**.

At last, do not forget to give it a restriction for its **Alignment** and set up *vertical*.

As an example to let you deeply understand how to set up one selection type,

```
<Question isMandatory="yes" type="ranking" id="q0001">
    <Limitation minLength="1" maxLength="5"/>
    <Description type="paragraph" value="Please rank the follow actors:" id="de0001"/>
    <Alignment vertical="stretch"/>
    <Answer id="a0001">
        <Description type="paragraph" value="Johnny Depp" id="de0002"/>
    </Answer>
    <Answer id="a0002">
        <Description type="paragraph" value="Will Smith" id="de0003"/>
    </Answer>
    <Answer id="a0003">
        <Description type="paragraph" value="Leonardo DiCaprio" id="de0004"/>
    </Answer>
    <Answer id="a0004">
        <Description type="paragraph" value="Brad Pitt" id="de0005"/>
    </Answer>
    <Answer id="a0005">
        <Description type="paragraph" value="George Clooney" id="de0006"/>
    </Answer>
</Question>
```

However, there is a difficult that is how to set the sample answer for the e-testing. Even we always set the sample in a Boolean type, but in ranking situation, the sample should be a order result. Therefore,

```
<Question isMandatory="yes" type="ranking" id="q0001">
    ...
    <Answer id="a0005" defaultValue="4">
        <Description type="paragraph" value="George Clooney" id="de0006"/>
    </Answer>
</Question>
```

7.2 Possible Combinations of Question Types

After we explained the basic question types, we will present the possible combinations of basic question types. In this sub-chapter, we will describe the possible combinations as such examples.
At first, we explain some common possible combination question types.

7.2.1 Contact Information

What is Contact Information?
Contact Information adds in a properly formatted and consolidated question asking users for their contact information (see Fig. 7-12). It is similar with open-ended text question type, like a set of text questions.

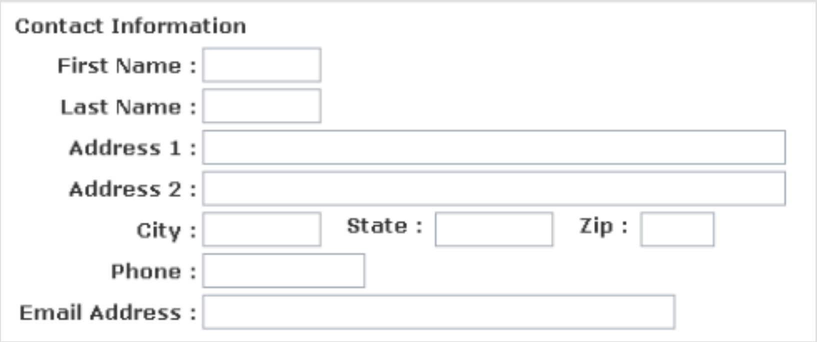


Fig. 7-12 Screenshot of Contact Information

How to set up a contact information question type?

At first, questioner should choose the question type for the **Question** element, which has an attribute "*type*". Questioner should choose the type attribute value "text", because it likes a set of open-ended text questions.

In addition, to set up the **Answer** element.

Furthermore, to distinguish the long text, single row text, numeric text and email address, questioner can control it through **Limitation** element. However, it is different with Multiple Choice Question Type, the **Limitation** element is as a child element of **Answer** element.

At last, the **Alignment** element is necessary to distinguish each line and arrangement.

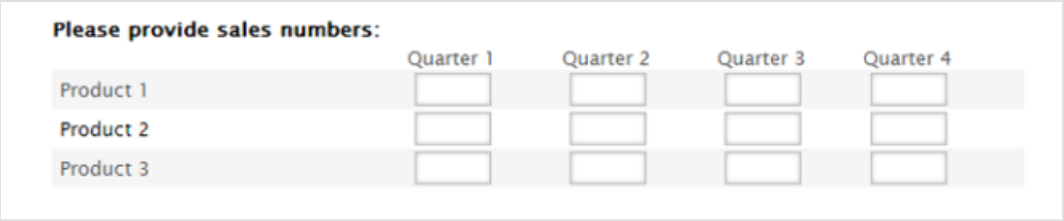
As an example to set up a contact information presented in Fig. 7-12,

```
<Question isMandatory="yes" type="text" id="q0001">
    <Limitation minLength="1" maxLength="1" />
    <Description type="paragraph" value="Contact Information" id="de0001"/>
    <Alignment vertical="stretch"/>
    <!--each answer arrangement-->
    <Answer id="a0001">
        <Limitation minLength="1" maxLength="15" format="char"/>
        <Description type="paragraph" value="First Name:" id="de0002"/>
    </Answer>
    <Answer id="a0002">
        <Limitation minLength="1" maxLength="15" format="char"/>
        <Description type="paragraph" value="First Name:" id="de0003"/>
    </Answer>
    <Answer id="a0003">
        <Limitation minLength="5" maxLength="129" format="char"/>
        <Description type="paragraph" value="Address 1:" id="de0004"/>
    </Answer>
    <Answer id="a0004">
        <Limitation minLength="5" maxLength="129" format="char"/>
        <Description type="paragraph" value="Address 2:" id="de0005"/>
    </Answer>
    ...
</Question>
```

7.2.2 Matrix Spreadsheet

What is matrix spreadsheet?

Matrix Spreadsheet is a multiple questions in spreadsheet style with text input boxes (see **Fig. 7-2-2-1**). It is the combination of open-ended text type and matrix. In QSL, we sort it as one of the matrix (easy to remember because of the name of it).



Please provide sales numbers:

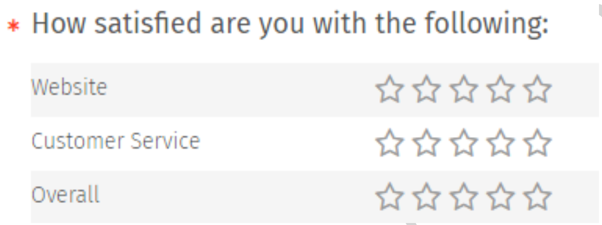
	Quarter 1	Quarter 2	Quarter 3	Quarter 4
Product 1	□	□	□	□
Product 2	□	□	□	□
Product 3	□	□	□	□

Fig. 7-13 Screenshot of Matrix Spreadsheet

How to set up a matrix spreadsheet?

```
<Question isMandatory="yes" type="matrix" id="q0001">
    <Limitation minLength="1" maxLength="5" format="int"/>
    <!--length control the character -->
    <Description type="paragraph" value="Please rate the following" id="de0001"/>
    <!--Column-->
    <Answer id="a0001" Column.Grid="0">
        <!--Grid begins from 0 -->
        <Description type="paragraph" value="Quarter 1" id="de0002"/>
    </Answer>
    <Answer id="a0002" Column.Grid="1">
        <Description type="paragraph" value="Quarter 2" id="de0003"/>
    </Answer>
    <Answer id="a0003" Column.Grid="2">
        <Description type="paragraph" value="Quarter 3" id="de0004"/>
    </Answer>
    <Answer id="a0004" Column.Grid="3">
        <Description type="paragraph" value="Quarter 4" id="de0005"/>
    </Answer>
    <!--Rows-->
    <Answer id="a0005" Row.Grid="0">
        <Description type="paragraph" value="Product 1" id="de0006"/>
    </Answer>
    <Answer id="a0006" Row.Grid="1">
        <Description type="paragraph" value="Product 2" id="de0007"/>
    </Answer>
    <Answer id="a0007" Row.Grid="2">
        <Description type="paragraph" value="Product 3" id="de0008"/>
    </Answer>
</Question>
```

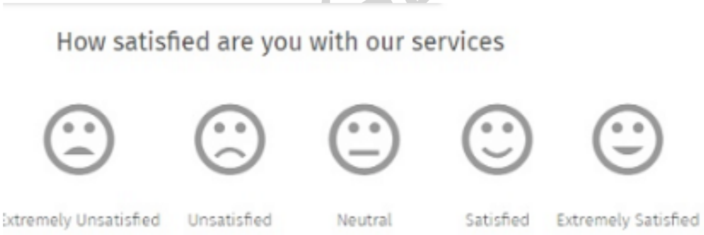
7.2.3 Slide and Rating



* How satisfied are you with the following:

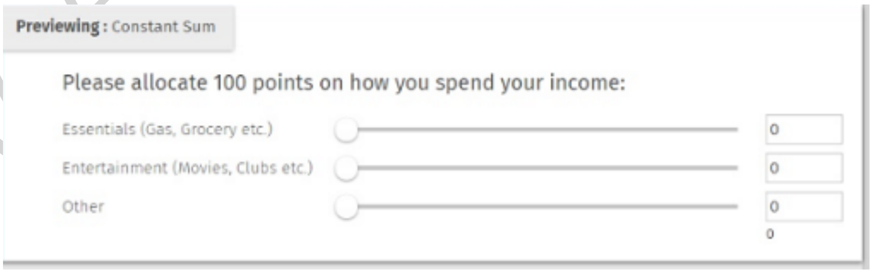
Website	☆☆☆☆☆
Customer Service	☆☆☆☆☆
Overall	☆☆☆☆☆

Fig. 7-14 Screenshot of Star Rating



How satisfied are you with our services

Fig. 7-15 Screenshot of Smiley Rating

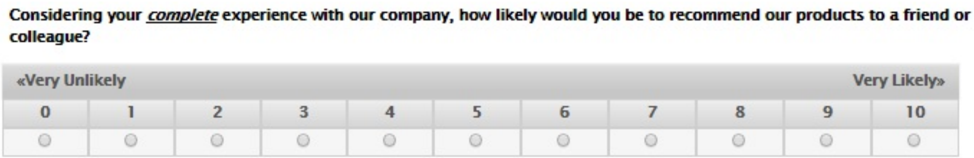


Previewing : Constant Sum

Please allocate 100 points on how you spend your income:

Essentials (Gas, Grocery etc.)	0
Entertainment (Movies, Clubs etc.)	0
Other	0
	0

Fig. 7-16 Screenshot of Slide



Considering your *complete* experience with our company, how likely would you be to recommend our products to a friend or colleague?

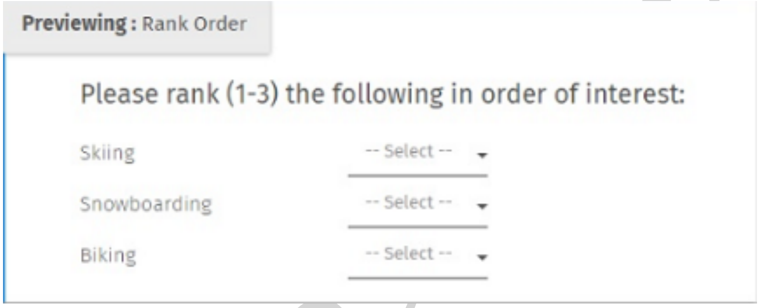
«Very Unlikely Very Likely»

0	1	2	3	4	5	6	7	8	9	10
○	○	○	○	○	○	○	○	○	○	○

Fig. 7-17 Screenshot of NPS

These 4 kinds of question types are similar. From the respective of basic question type, we consider them as single selection. They have different presentations. However, QSL does not consider the front-end presentation.
Star Rating can be designed as a single selection question with 5 answer options, which are weights from 1to 5. Smiley Rating can be designed as a single selection question with 5 answer options and their corresponding images. As to Slide question, it also can be designed as a single selection with a 100 options. In this situation, questioner just need to specify the limitation *maxOccur* as 100, and omits the answer values and ids.

7.2.4 Rank Order and Drag



Previewing : Rank Order

Please rank (1-3) the following in order of interest:

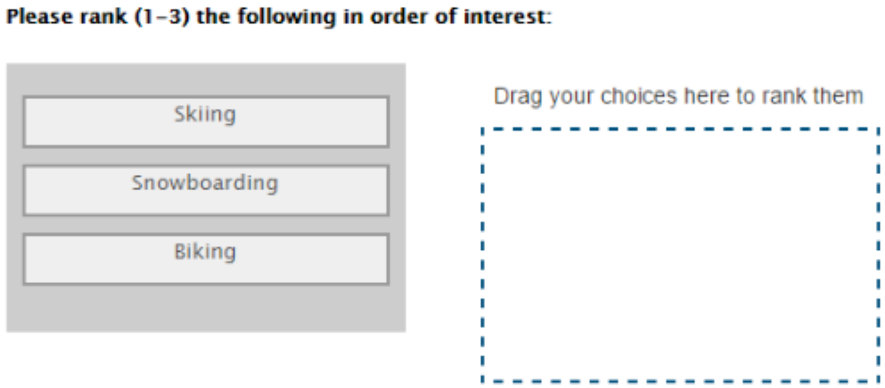
Skiing -- Select --

Snowboarding -- Select --

Biking -- Select --

Fig. 7-18 Screenshot of Rank Order

This is based on the single selection (drop-down list), but concerns exclusive when choose an option, the others cannot choose the selected options. As the prototype, It likes a matrix: grid row (skiing, snowboarding, biking), and grid column (1, 2, 3), and mutually exclude.



Please rank (1–3) the following in order of interest:

Skiing

Snowboarding

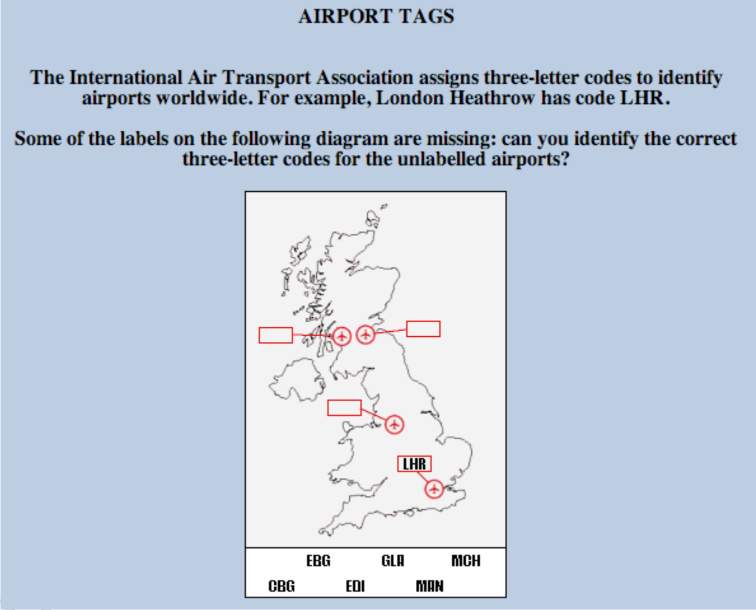
Biking

Drag your choices here to rank them

Fig. 7-19 Screenshot of Drag Object

This is based on ranking question type. Drag is a presentation, which QSL does not concern until now.

```
<Question isMandatory="yes" type="ranking" id="q0001">
<!--question type: ranking-->
    <Description type="paragraph" value="Please rank..." id="de0001"/>
    <Answer id="a0001">
        <Description type="paragraph" value="Skiing" id="de0002"/>
    </Answer>
    <Answer id="a0002">
        <Description type="paragraph" value="Snowboarding" id="de0003"/>
    </Answer>
    <Answer id="a0003">
        <Description type="paragraph" value="Biking" id="de0004"/>
    </Answer>
</Question>
```



AIRPORT TAGS

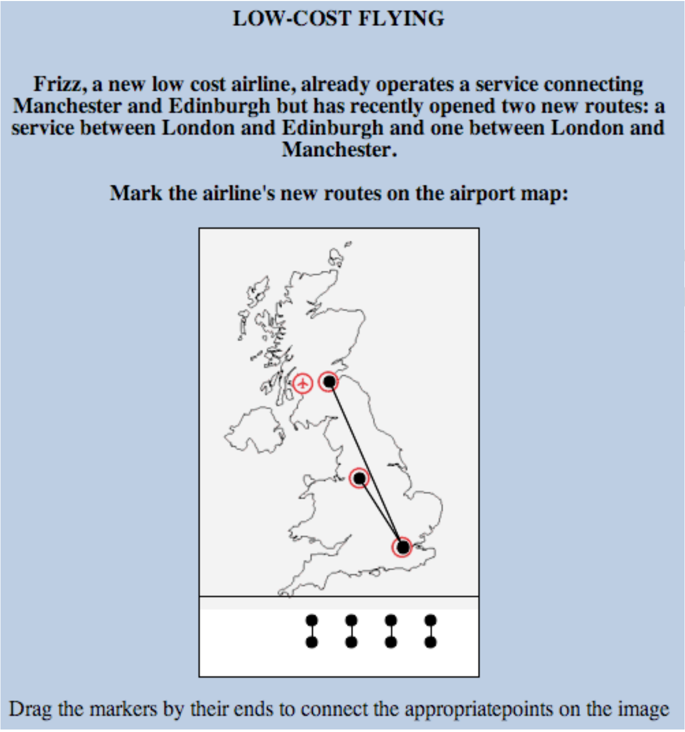
The International Air Transport Association assigns three-letter codes to identify airports worldwide. For example, London Heathrow has code LHR.

Some of the labels on the following diagram are missing: can you identify the correct three-letter codes for the unlabelled airports?

Fig. 7-20 Image of Drag Target

This image refer to QTI. It is based on ranking (unrepeated options), but this question concerns the image and location. The questioner can mark the locations on the map and let respondent to choose. It is a extension of match item or Connect the Point question type.

```
<Question isMandatory="yes" type="ranking" id="q0001">
    <Limitation minOccur="5" maxOccur="5"/>
    <Description type="header" value="AIRPOER TAGS" id="de0001"/>
    <!--header -->
    <Description type="paragraph" value="The International Air Transport..."
id="de0002"/>
    <!--explanation-->
    <Media id="me0001" type="img/png" src="img/airport.png"/>
    <!--Image with marks -->
    <Answer id="a0001">
        <Description type="paragraph" value="EGB" id="de0003"/>
    </Answer>
    ...
</Question>
```



LOW-COST FLYING

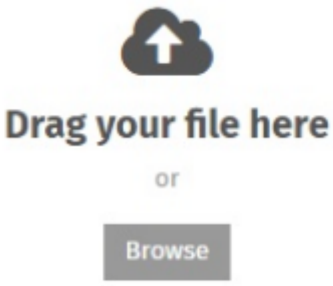
Frizz, a new low cost airline, already operates a service connecting Manchester and Edinburgh but has recently opened two new routes: a service between London and Edinburgh and one between London and Manchester.

Mark the airline's new routes on the airport map:

Drag the markers by their ends to connect the appropriatepoints on the image

Fig. 7-21 Image of Connect the Point

7.2.5 Upload File

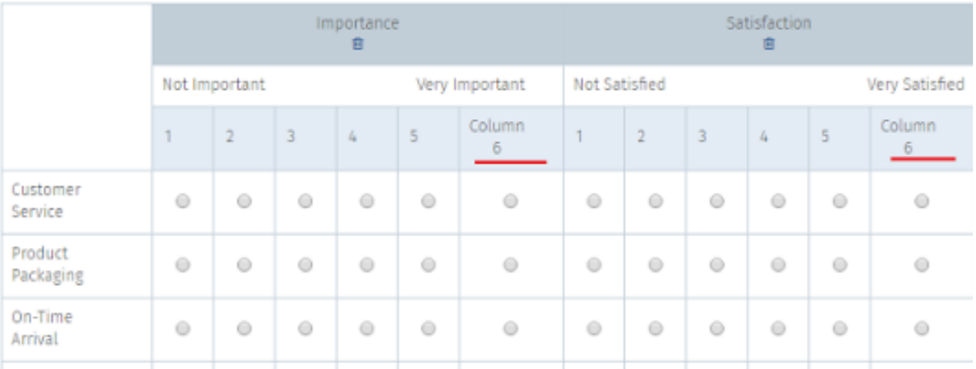


Drag your file here

Fig. 7-22 Screenshot of Upload File

```
<Question isMandatory="yes" type="upload" id="q0001">
<!--question type: upload-->
    <Description type="paragraph" value="Drag your file here" id="de0001"/>
    <Answer id=″a0001″>
        <Description type=″paragraph″ value=″ ″ id=″de0002″/>
        <!--omit value -->
    </Answer>
</Question>
```

7.2.6 Sid-by-side Matrix



	Importance						Satisfaction					
	Not Important				Very Important		Not Satisfied				Very Satisfied	
	1	2	3	4	5	Column 6	1	2	3	4	5	Column 6
Customer Service	○	○	○	○	○	○	○	○	○	○	○	○
Product Packaging	○	○	○	○	○	○	○	○	○	○	○	○
On-Time Arrival	○	○	○	○	○	○	○	○	○	○	○	○

Fig. 7-23 Screenshot of SBS-Matrix

This is a 3D-matrix question type. X.Grid (Importance, Satisfaction), Y.Grid (1,2,3,4,5), Z.Grid (Customer Service, Product Package, On-Time Arrtival)

```
<Question isMandatory=″yes″ type=″matrix″ id=″q0001″>
    <Limitation minOccur=″1″ maxOccur=″1″/><!--single selection-->
    <Description type=″paragraph″ value=″Please rate the following″ id=″de0001″/>
    <!--X-->
    <Answer id=″a0001″ X.Grid=″0″>
        <Description type=″paragraph″ value=″Importance″ id=″de0002″/>
    </Answer>
    <Answer id=″a0002″ X.Grid=″1″>
        <Description type=″paragraph″ value=″Satisfaction″ id=″de0003″/>
    </Answer>
    <!--Y-->
    <Answer id=″a0003″ Y.Grid=″0″>
        <Description type=″paragraph″ value=″1″ id=″de0004″/>
    </Answer>...
    <!--Z-->
    <Answer id=″a0004″ Z.Grid=″0″>
        <Description type=″paragraph″ value=″Customer Service″ id=″de0005″/>
    </Answer>...
</Question>
```

8 Logic Types Reference Guide

So far, we have introduced the creation of static e-questionnaires. In this chapter, we will see how to create dynamic e-questionnaires; we also called “intelligent” e-questionnaires to ensure that only relevant questions are displayed to the appropriate respondents.

Previously, we described the **logic** element and its attribute *type* at the end of chapter 6. However, only scratching the surface, for this large field, was not enough. For reason, it owes to various manifestations of e-questionnaires.

From all the minor details, many e-questionnaire systems provide it a variety of classifications and definitions. And thereupon we proceeded with the thoughts of trichromatic combination principle, in other words, a multiplicity of separate elements according to some standards. As to uses, this chapter provides a reference to all of logic types (basic logic types and possible combinations) to define them.

All content and diagrams in this chapter refer to *QuestionPro*®.

8.1 Basic Logic Types

There are four basic logic types:

- Skipping,
- Piping,
- Extraction,
- and Randomization

8.1.1 Skipping

What is skipping?

Skipping gives questioner the ability to create “intelligent” e-questionnaires. For example, it wouldn’t make sense to ask questions about married life to someone who is not married. Using skipping questioner can ensure that only relevant questions are displayed to the appropriate respondents.

How to set up skipping?

- The **Logic** is the child of **Paper**, so that questioner must write logic that should be included in tags of paper.
- **Logic** has a child element **Route**. **Route**'s attribute *type* is written as “skipping”.
- **Route**'s child elements **Condition** and **Action** to control if satisfies

conditions and then jump to where.

```
<Logic>
    <Route type=″skipping″ id=″ro0001″>
        <Condition answerId=″a0001″ isChecked=″yes″/>
        <Action questionId=″q0007″/>
    </Route>
</Logic>
```

which means that if selected a0001, then jump to question qu0007.

- Sometimes, between start-tags and the end-tags, records the URL of destination, as to the to attribute, as the sample value as “url”.

```
<Logic>
    <Route type=″skipping″ id=″ro0001″>
        <Condition answerId=″a0001″ isChecked=″yes″/>
        <Action to=″www.url.com″/>
    </Route>
</Logic>
```

which means that from this answer skipping to a website. There is a situation that if choose an option then jump to Thank you page usually occurs in e-questionnaire. `<Action descriptionId=″de0010″/>`

What are the different skip to options?

Based on the answer option selected, respondents will be directed to the jump to location that is set on the survey. Following Skipping To options are available:

- **Papers**: select from any e-questionnaire and e-testing that is permitted to enter.
- **Questions**: select from any question that is after the source question.
- **Sections**: can select from any of the section that are after the source question.
- **Answers**: usually skip to answer when question types is chosen as “text”.
- **Terminate**: go to finished page, if questioner design its setting that respondent can see the result, then show the statistics, otherwise, jump to:
- **Go to Conclusion Page (Description)**: Select this option if questioners wish to direct respondents to the Thank You page. Response will be marked as a complete response. In other words, select this option if you wish to terminate the survey for respondents.
- **Chain URL**: Select this option to take respondents to a different website.

What is Default Destination skipping?

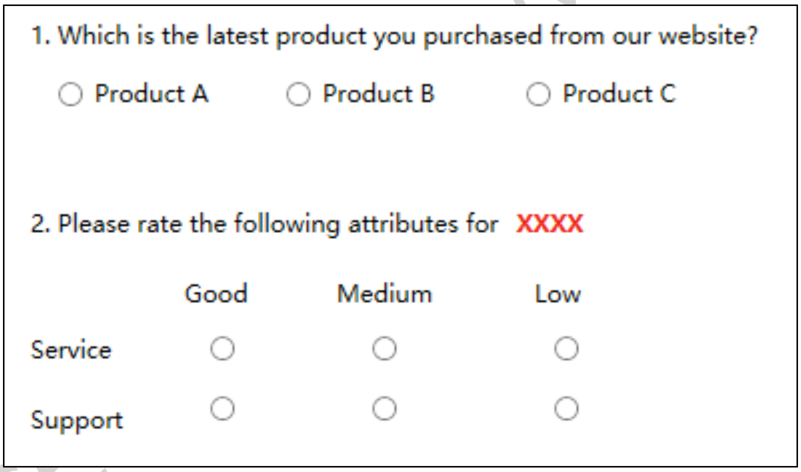
The Default Destination skipping is the fallback logic that gets executed when no other logic gets triggered. If respondents do not choose any option, the default skipping destination is used when NO OTHER logic gets triggered. This is the fallback logic that gets executed when no other logic gets triggered.

```
<Logic>
    <Route type="skipping" id="ro0001">
        <Condition questionId="q0001" isChecked="no"/>
        <!--if there is no choice of question-->
        <Action descriptionId="de0010"/>
        <!--Then finish-->
    </Route>
</Logic>
```

8.1.2 Piping

What is piping?

Piping enables you to carry text from one question to the next depending on the options selected by the respondent. For example if you have a scenario as described below (see Fig. 8-1).



1. Which is the latest product you purchased from our website?

○ Product A ○ Product B ○ Product C

2. Please rate the following attributes for XXXX

	Good	Medium	Low
Service	○	○	○
Support	○	○	○

Fig. 8-1 Scenario of Piping

Now questioner would like to replace the text XXXX with the actual product that the respondent had selected for question 1. Piping will enable you to replace XXXX with the appropriate text.

How to set up Piping?

- **Logic** is the schema 310 element. Its child element **Route** to define a route of logic. It is necessary to define route id, and type value as "piping".
- **Route**'s child elements: **Condition** and **Action**.
- **Condition** sets the question's reference, and the condition trigger (*isChecked*). **Action** sets the piping route and the filled location *to* attribute that is written as dollar mark as braces including piping Ids,

underscore and id which is like a function name, such as “$('piping Ids')”. We restricted the value of it as “**$('pip[0-9]*')**”. Then, in the destination question. As the example,

```
<Question id=″q0002″>
    <Description id=″de0002″ type=″paragraph″ value=″Please rate the following
attributes for $(’pip0001’)″/>
<!--question2 title add the piping location-->
    …
</Question>
…
<Logic>
    <Route type=″piping″ id=″ro0001″>
        <Condition paperId=″p0001″ sectionId=″s0001″ questionId=″q0001″
isChecked=″yes″/>
        <Action paperId=″p0001″ sectionId=″s0001″ questionId=″q0002″ isFilled=″yes″
to=″$(’pip0001’)″/>
<!--action: filled the piping location into checked values-->
    </Route>
</Logic>
```

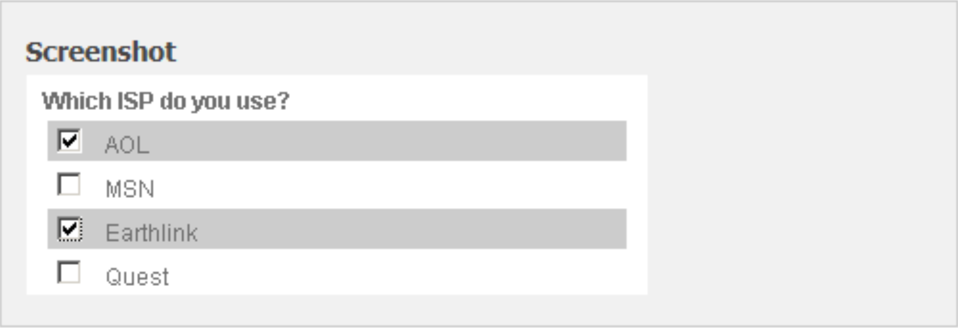
In this example, at the first question, there are 3 choices using piping logic, whichever be chosen, the question answer text will be piping to next question as one part of next question text.

Can I carry over the Piping Text to multiple questions?
Piping text can only be carried over to the next question. If you wish to carry over text to multiple questions you can get more information in possible combinations of logic types below.

Is there any limitation on the number of characters that can be piped?
Piping text has a limitation of 255 characters (Including Spaces/Special Characters).

8.1.3 Extraction

What is extraction?
Extraction enables questioners to display the choices of a question based on the choices selected for the previous question. It differs from piping. Piping is used to carry out the selected options to fill in some descriptions. However, Extraction is used to carry out the selected options to compare the existing question's texts. For example, (see Fig. 8-2, Fig. 8-3, and Fig. 8-4)



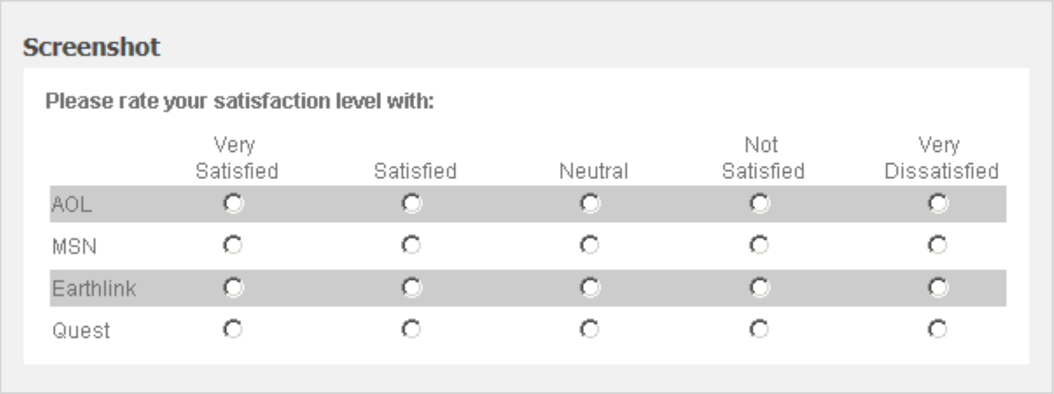
Screenshot

Which ISP do you use?

☑ AOL
□ MSN
☑ Earthlink
□ Quest

Fig. 8-2 Screenshot of Extraction

The following question without extraction will be displayed as follows:



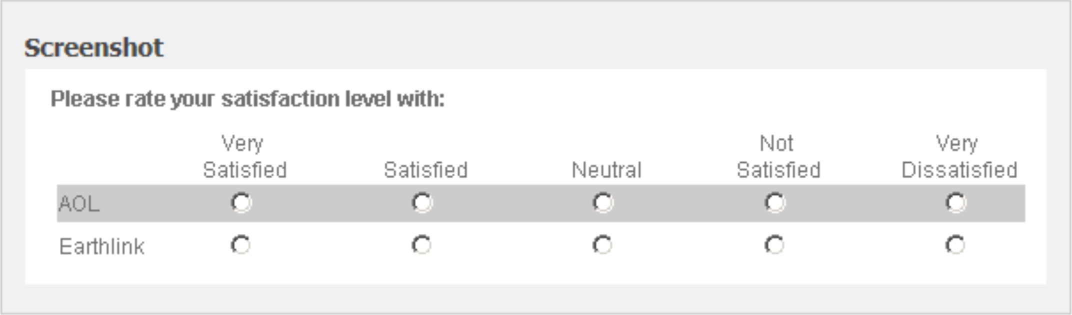
Screenshot

Please rate your satisfaction level with:

	Very Satisfied	Satisfied	Neutral	Not Satisfied	Very Dissatisfied
AOL	○	○	○	○	○
MSN	○	○	○	○	○
Earthlink	○	○	○	○	○
Quest	○	○	○	○	○

Fig. 8-3 Screenshot of Extraction 2

The following question without extraction will be displayed as follows:
If the respondent selects options **AOL** and **Earthlink** for question 1 then the extracted question (i.e. question 2) will be displayed as follows:



Screenshot

Please rate your satisfaction level with:

	Very Satisfied	Satisfied	Neutral	Not Satisfied	Very Dissatisfied
AOL	○	○	○	○	○
Earthlink	○	○	○	○	○

Fig. 8-4 Screenshot of Extraction 3

(Only the options selected by the respondent will be displayed on e-questionnaire for respondent).

How to set up Extraction?

- **Logic** is the schema 310 element. Its child element **Route** to define a route of logic. It is necessary to define route id, and type value as "extraction".

- **Route**'s child elements: **Condition** and **Action.**
- **Condition** sets the question's reference, and the condition trigger (*isChecked*). **Action** sets the piping route and the filled location *to* attribute that is written as dollar mark as braces including piping Ids, underscore and id which is like a function name, such as "$('piping Ids')". We restricted the value of it as "**$('ext[0-9]*')**". Then, in the destination question.

```
<Paper id=" p0001" type="questionnaire">
    <Section id=" s0001">
        <Question isMandatory="yes" type="selection" id="q0001">
            <Limitation minLength="1" maxLength="4"/>
            <Description type="paragraph" value="Which ISP do you use?" id="de0001"/>
            <Answer id="a0001">
                <Description type="paragraph" value="AOL" id="de0002"/>
            </Answer>
            <Answer id="a0002">
                <Description type="paragraph" value="MSN" id="de0003"/>
            </Answer>
            <Answer id="a0003">
                <Description type="paragraph" value="Earthlink" id="de0004"/>
            </Answer>
            <Answer id="a0004">
                <Description type="paragraph" value="Quest" id="de0005"/>
            </Answer>
        </Question>
        <Question isMandatory="yes" type="matrix" id="q0002">
            <Limitation minOccur="1" maxOccur="1"/>
            <Description type="paragraph" value="Plaese rate your satisfaction level
with:" id="de0006"/>
            <!--Column 0-4-->
            <Answer id="a0005" Column.Grid="0">
                <Description type="paragraph" value="Very Satisfied" id="de0007"/>
            </Answer>
            <Answer id="a0006" Column.Grid="1">
                <Description type="paragraph" value="Satisfied" id="de0008"/>
            </Answer>
            <Answer id="a0007" Column.Grid="2">
                <Description type="paragraph" value="Neutral" id="de0009"/>
            </Answer>
            <Answer id="a0008" Column.Grid="3">
                <Description type="paragraph" value="Not Satisfied" id="de0010"/>
            </Answer>
            <Answer id="a0009" Column.Grid="4">
                <Description type="paragraph" value="Very Dissatisfied"
id="de0011"/>
            </Answer>
            <!--Normal situation begin: Row 0-3-->
            <Answer id="a0010" Row.Grid="0">
                <Description type="paragraph" value="AOL" id="de0012"/>
            </Answer>
            <Answer id="a0011" Row.Grid="1">
                <Description type="paragraph" value="MSN" id="de0013"/>
            </Answer>
            <Answer id="a0012" Row.Grid="2">
                <Description type="paragraph" value="Earthlink" id="de0014"/>
            </Answer>
            <Answer id="a0013" Row.Grid="3">
                <Description type="paragraph" value="Quest" id="de0015"/>
            </Answer>
            <!--Normal situation end: Row 0-3-->
            <!--Extraction situation begin: Rows-->
            <Answer id="a0010" Row.Grid="*"> <!--rows auto-fill |add-->
                <Description type="paragraph" value="$('ext0001')" id="de0012"/>
            </Answer>
            <!--Extraction situation end: Rows-->
        </Question>
    </Section>
    <Logic>
        <Route type="extraction" id="ro0001">
            <Condition paperId="p0001" sectionId="s0001" questionId="q0001"
isChecked="yes"/>
            <Action paperId="p0001" sectionId="s0001" questionId="q0002"
isFilled="yes" to="$('ext0001')"/>
        </Route>
    </Logic>
</Paper>
```

8.1.4 Randomization

What is randomization?

In order to avoid cheating, some questioners want sections, questions, or answers can be randomized to be presented. Randomization enables you to display without ordering in advance.

How to use randomization?

Because randomization does not need to piping to a destination, so if the questioner chooses a randomization type as a attribute value of **Route**. The reference attribute group support the paper randomization, section randomization, question randomization, and answer randomization.

What is the conflict?

There is a schema 230 Setting. This schema controls the order setting (ascending, descending, and default). If questioner does not set this setting for the same loaction (paper, section, question, answer), then he can set this logic route.

With which question type does the answer display order works?

- Multiple Choice - Select One, Drop Down, Image Chooser, Select Many, Image Chooser (Select Many)
- Rank Order / Drag N Drop

How to set up Randomization?

- **Logic** is the schema 310 element. Its child element **Route** to define a route of logic. It is necessary to define route id, and type value as "randomization".
- **Route**'s child elements: **Condition** and **Action**. In randomization logic situation, it does not need **Condition**.
- **Action** sets the locations should be randomized.

As the example,

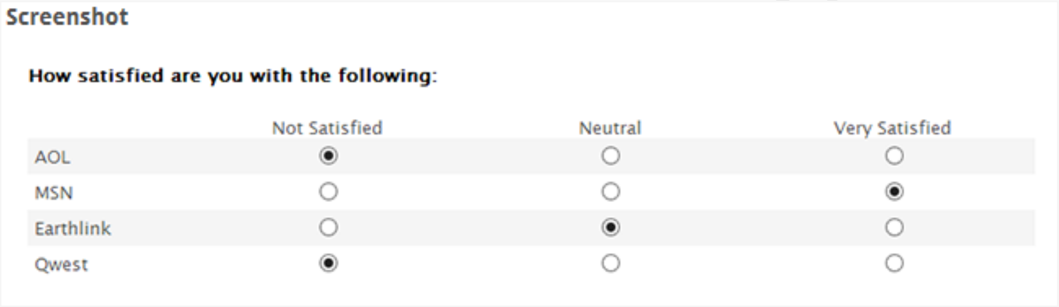
```
<Logic>
    <Route type="randomization" id="ro0001">
        <!--The answers of location:( paper 1 section 1 question 1) should be randomized
-->
        <Action paperId="p0001" sectionId="s0001" questionId="q0001"/>
        <!-- The questions of location:( paper 1 section 2) should be randomized ->
        <Action paperId="p0001" sectionId="s0002"/>
        <!--gfgf-->
        <Action paperId="p0001" sectionId="s0001" questionId="q0001"/>
    </Route>
    <Route type="randomization" id="ro0002">
        <!-- The sections of location:( paper 2) should be randomized -->
        <Action paperId="p0002"/>
    </Route>
</Logic>
```

8.2 Possible Combinations of Logic Types

After we explained the basic logic types, we will present the possible combinations of basic logic types. In this sub-chapter, we will describe the possible combinations as such examples.
At first, we explain some common possible combination logic types.

8.2.1 Matrix Extraction

What is extraction from matrix question type?
Extracting from a matrix question is a very useful feature, especially for surveys such as (see Fig. 8-5):



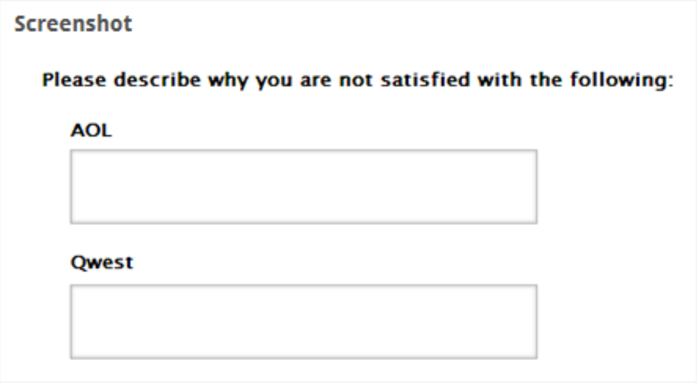
Screenshot

How satisfied are you with the following:

	Not Satisfied	Neutral	Very Satisfied
AOL	●	○	○
MSN	○	○	●
Earthlink	○	●	○
Qwest	●	○	○

Fig. 8-5 Screenshot of Matrix Extraction

Now, suppose questioner who would like to ask the respondent why he/she selected Not Satisfied for a particular service provider. Questioner can use Extraction to set this up.
If the respondent selects Not Satisfied for AOL and Qwest for question in Fig. 8-5 then the extracted question (i.e. in Fig. 8-6) will be displayed as follows:



Screenshot

Please describe why you are not satisfied with the following:

AOL

Qwest

Fig. 8-6 Screenshot of Matrix Extraction

In this scenario above, it is similar with the example presented in simple extraction type, just the question type changed. Therefore, the specification is presented below,

```
<Paper type="questionnaire" id="p0001">
    <Section id="s0001">
        <!--the first question-->
        <Question isMandatory="yes" type="matrix" id="q0001">
            <Limitation minOccur="1" maxOccur="1"/>
            <!--single choice for matrix each row-->
            <Description type="paragraph" value="How satisfied are you with the
following:" id="de0001"/>
            <Answer id="a0001" Column.Grid="0">
                <Description type="paragraph" value="Not Satisfied" id="de0002"/>
            </Answer>
            <Answer id="a0002" Column.Grid="1">
                <Description type="paragraph" value="Neutral" id="de0003"/>
            </Answer>
            <Answer id="a0003" Column.Grid="2">
                <Description type="paragraph" value="Very Satisfied" id="de0004"/>
            </Answer>
            <Answer id="a0004" Column.Grid="0">
                <Description type="paragraph" value="AOL" id="de0005"/>
            </Answer>
            <Answer id="a0005" Row.Grid="1">
                <Description type="paragraph" value="MSN" id="de0006"/>
            </Answer>
            <Answer id="a0006" Row.Grid="2">
                <Description type="paragraph" value="Earthlink" id="de0007"/>
            </Answer>
            <Answer id="a0007" Row.Grid="3">
                <Description type="paragraph" value="Qwest" id="de0008"/>
            </Answer>
        </Question>
        <Description type="break" value="" id="de0009"/>
        <!--break to create a new area for the second one-->
        <!--the second question-->
        <Question isMandatory="yes" type="text" id="q0002">
            <Limitation minLength="1" maxLength="1"/>
            <Description type="paragraph" value="Please describe why you are not
satisfied with the following:" id="de0010"/>
            <Answer id="a0008">
                <Limitation minLength="5" maxLength="129"/>
                <Description type="paragraph" value="$('ext0001')" id="de0011"/>
            </Answer>
        </Question>
    </Section>
    <Logic>
        <Route type="extraction" id="ro0001">
            <Action paperId="p0001" sectionId="s0001" questionId="q0002"
isFilled="yes" to="$('ext0001')"/>
        </Route>
    </Logic>
</Paper>
```

8.2.2 Compound Branching

What is compound branching logic?

With simple branching of skipping type, questioners cannot set up logic based on responses to multiple questions. Also with simple branching, the logic is executed immediately. **Compound Branching** means that with help of compound Logic you can set up branching based on responses to multiple questions in the questionnaire. Sometimes, there is another name called **Delayed Branching** that means questioners can set up Branching Logic based on a Question which was asked much earlier in the questionnaire. But, there are no differences between both of them.

How to use Compound Branching?

In compound branching, we provide logic *type* "skipping" to jump to other questions, *and* attribute to connect with relative answers.

Consider this example:

Q1: Gender
Q2: Age
Q3: General question to be answered by everybody
Q4: General question to be answered by everybody
Q5: General question to be answered by everybody

Now after Q5 respondents should branch to different questions based on their selection in Q1 AND Q2

Q6: To be answered only by respondents who selected Gender=Male and Age=18 & above

Q7: To be answered only by respondents who selected Gender=Female and Age=18 & above

Now all respondents should again answer rest of the questions.

Q8: General question for all respondents
Q9: General questions for all respondents.

Specification for setting up this case:

```
<Paper type=″questionnaire″ id=″p0001″>
    <Section id=″s0001″>
        <Question isMandatory=″yes″ type=″selection″ id=″q0001″>
        <Limitation minOccur=″1″ maxOccur=″1″/>
            <Description type=″paragraph″ value=″Gender″ id=″de0001″/>
            <Answer id=″a0001″>
                <Description type=″paragraph″ value=″F″ id=″de0002″/>
            </Answer>
            <Answer id=″a0002″>
                <Description type=″paragraph″ value=″M″ id=″de0003″/>
            </Answer>
        </Question>
        <Question isMandatory=″yes″ type=″selection″ id=″q0002″>
            <Limitation minOccur=″1″ maxOccur=″1″/>
            <Description type=″paragraph″ value=″Age″ id=″de0004″/>
            <Answer id=″a0003″>
                <Description type=″paragraph″ value=″less than 18″ id=″de0005″/>
            </Answer>
            <Answer id=″a0004″>
                <Description type=″paragraph″ value=″greater than or equal to 18, less
than 60″ id=″de0006″/>
            </Answer>
            <Answer id=″a0005″>
                <Description type=″paragraph″ value=″greater than or equal to 60″
id=″de0007″/>
            </Answer>
        </Question>
        <!--Question 3-5 omitted-->
        <Question isMandatory=″yes″ type=″text″ id=″q0006″>
            <Limitation minLength=″1″ maxLength=″1″/>
            <Description type=″paragraph″ value=″Reason″ id=″de0013″/>
            <Answer id=″a0010″>
                <Description type=″paragraph″ value=″″ id=″de0014″/>
            </Answer>
        </Question>
        <Question isMandatory=″yes″ type=″text″ id=″q0007″>
            <Limitation minLength=″1″ maxLength=″1″/>
            <Description type=″paragraph″ value=″Reason″ id=″de0015″/>
            <Answer id=″a0011″>
```

```
                <Description type="paragraph" value="" id="de0016"/>
            </Answer>
        </Question>
    </Section>
    <Logic>
        <Route type="skipping" id="ro0001">
            <Condition paperId="p0001" sectionId="s0001" questionId="q0001"
answerId="a0002" isChecked="yes" relation="and"/>
            <Condition paperId="p0001" sectionId="s0001" questionId="q0002"
answerId="a0003" isChecked="yes" relation="or"/>
            <Condition paperId="p0001" sectionId="s0001" questionId="q0002"
answerId="a0004" isChecked="yes"/>
            <Action paperId="p0001" sectionId="s0001" questionId="q0006"/>
        </Route>
        Route type="skipping" id="ro0002">
            <Condition paperId="p0001" sectionId="s0001" questionId="q0001"
answerId="a0001" isChecked="yes" relation="and"/>
            <Condition paperId="p0001" sectionId="s0001" questionId="q0002"
answerId="a0003" isChecked="yes" relation="or"/>
            <Condition paperId="p0001" sectionId="s0001" questionId="q0002"
answerId="a0004" isChecked="yes"/>
            <Action paperId="p0001" sectionId="s0001" questionId="q0006"/>
        </Route>
    </Logic>
</Paper>
```

Attention:

In theory, we can set up a delayed branching, such as use piping function in question 2 in above example, but, it is difficult to read and compile. Therefore, please try to avoid skipping back.

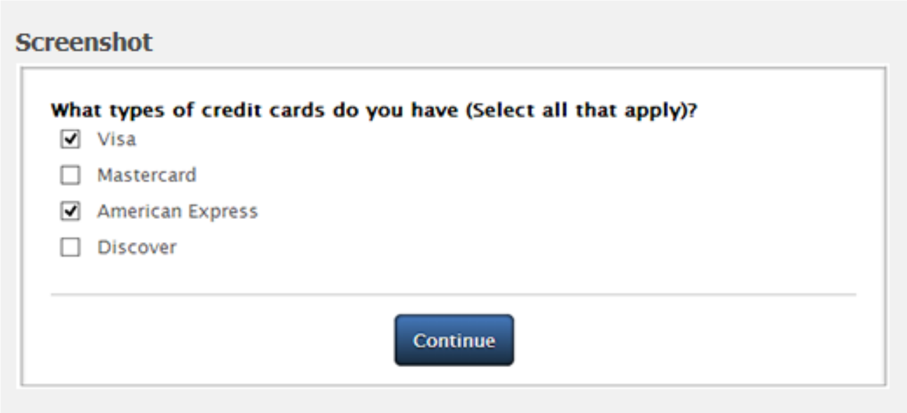
8.2.3 Looping with Text Piping

What is looping with text piping?

Simply put, considering the following example: Questioners are asking respondents to select credit cards they have and for each credit card, and wish to collect further information using follow up questions (see Fig. 8-7).

If the respondents select the options “Visa” and “American Express”, they should only be asked follow up questions for “Visa” and “American Express” cards. They should skip questions for “Master” and “Discover” cards.

To set this up, first set up the entire survey with all questions (see Fig. 8-7).



What types of credit cards do you have (Select all that apply)?

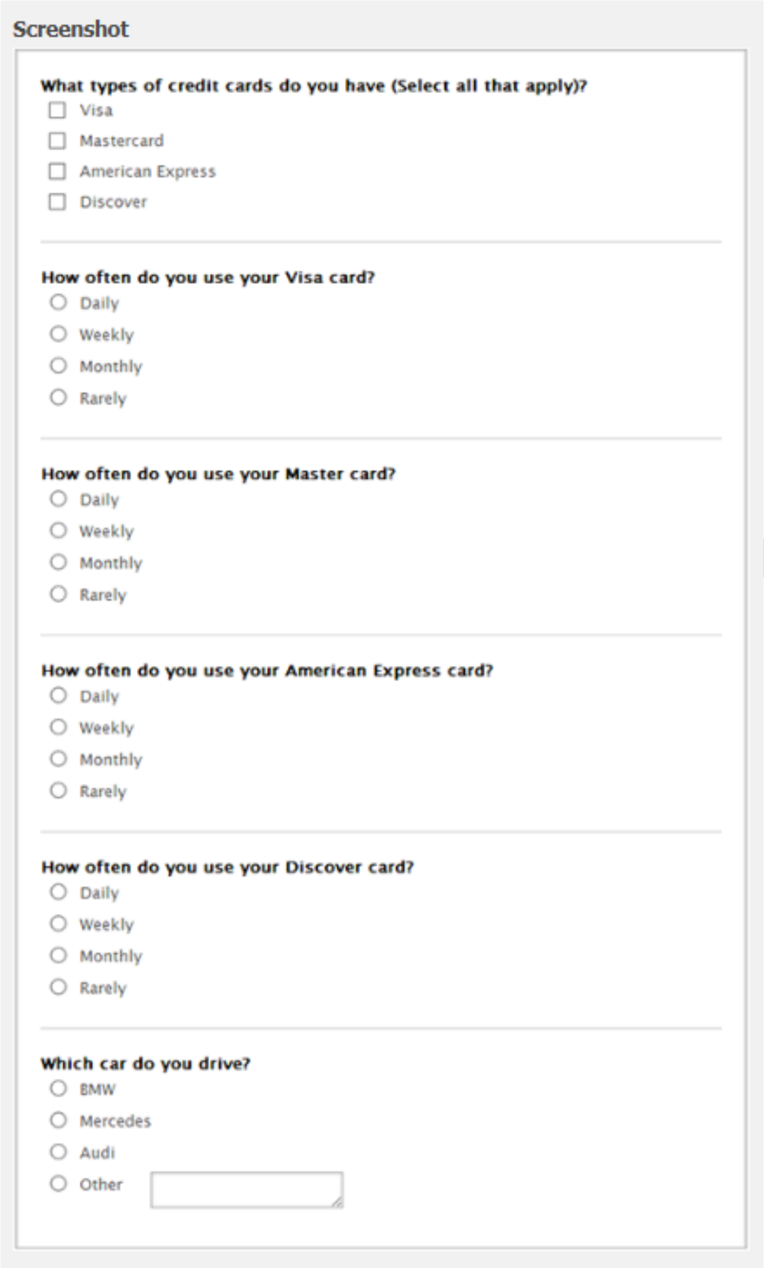
☑ Visa

□ Mastercard

☑ American Express

□ Discover

Fig. 8-7 Screenshot of Looping with Text Piping



Screenshot

What types of credit cards do you have (Select all that apply)?

□ Visa
□ Mastercard
□ American Express
□ Discover

How often do you use your Visa card?

○ Daily
○ Weekly
○ Monthly
○ Rarely

How often do you use your Master card?

○ Daily
○ Weekly
○ Monthly
○ Rarely

How often do you use your American Express card?

○ Daily
○ Weekly
○ Monthly
○ Rarely

How often do you use your Discover card?

○ Daily
○ Weekly
○ Monthly
○ Rarely

Which car do you drive?

○ BMW
○ Mercedes
○ Audi
○ Other

Fig. 8-8 Screenshot of Looping with Text Piping

How to use looping with text piping?

Looping with Text Piping is derived from Piping. Therefore, the logic *type* value is “piping”. Specification on above example:

```
<Paper type="questionnaire" id="p0001" >
    <Section id="s0001">
        <Question isMandatory="yes" type="selection" id="q0001">
            <Limitation minOccur="1" maxOccur="4"/>
            <Description type="paragraph" value="What types of credit cards do you
have (Select all that apply?)" id="de0001"/>
            <Alignment vertical="stretch"/>
            <Answer id="a0001">
```

```
                <Description type="paragraph" value="Visa" id="de0002"/>
            </Answer>
            <Answer id="a0002">
                <Description type="paragraph" value="Mastercard" id="de0003"/>
            </Answer>
            <Answer id="a0003">
                <Description type="paragraph" value="American Express" id="de0004"/>
            </Answer>
            <Answer id="a0004">
                <Description type="paragraph" value="Discover" id="de0005"/>
            </Answer>
            <Description type="break" value="" id="de0006"/>
        </Question>
        <Question isMandatory="yes" type="selection" id="q0002">
            <Limitation minOccur="1" maxOccur="1"/>
            <Description type="paragraph" value="How oftern do you use your
$('pip0001')" id="de0006"/>
            <Answer id="a0005">
                <Description type="paragraph" value="Daily" id="de0006"/>
            </Answer>
            <Answer id="a0006">
                <Description type="paragraph" value="Weekly" id="de0007"/>
            </Answer>
            <Answer id="a0007">
                <Description type="paragraph" value="Monthly" id="de0008"/>
            </Answer>
            <Answer id="a0008">
                <Description type="paragraph" value="Rarely" id="de0009"/>
            </Answer>
        </Question>
        <Question isMandatory="yes" type="selection" id="q0003">
            <Limitation minOccur="1" maxOccur="1"/>
            <Description type="paragraph" value="Which car do you dirve?"
id="q0003"/>
            <Answer id="a0009">
                <Description type="paragraph" value="BMW" id="de0010"/>
            </Answer>
            <Answer id="a0010">
                <Description type="paragraph" value="Mercedes" id="de0011"/>
            </Answer>
            <Answer id="a0011">
                <Description type="paragraph" value="Audi" id="de0012"/>
            </Answer>
            <Answer id="a0012" isOtherField="yes">
```

```
                <Description type="paragraph" value="Other" id="de0013"/>
            </Answer>
        </Question>
    </Section>
    <Logic>
        <Route type="piping" id="ro0001">
            <Condition paperId="p0001" sectionId="s0001" questionId="q0001"
isChecked="yes"/>
            <Action paperId="p0001" sectionId="s0001" questionId="q0002"
to="$(’pip0001’)"/>
        </Route>
    </Logic>
</Paper>
```

8.2.4 Quota and Scoring Logic

Refer to Fig. 8-7. For instance, questioner sets each answer option has different score.

```
    <Question isMandatory="yes" type="selection" id="q0001">
            <Limitation minOccur="1" maxOccur="4"/>
            <Description type="paragraph" value="What types of credit cards do you
have (Select all that apply?)" id="de0001"/>
            <Alignment vertical="stretch"/>
            <Answer id="a0001" score="2.5">
                <Description type="paragraph" value="Visa" id="de0002"/>
            </Answer>
            <Answer id="a0002" score="-2 ">
                <Description type="paragraph" value="Mastercard" id="de0003"/>
            </Answer>
            <Answer id="a0003" score="-1">
                <Description type="paragraph" value="American Express" id="de0004"/>
            </Answer>
            <Answer id="a0004" score="2.5">
                <Description type="paragraph" value="Discover" id="de0005"/>
            </Answer>
            <Description type="break" value="" id="de0006"/>
        </Question>
```

The total score of this question is 5 points. But each answer option score is different (2.5 point, -2 point, -1 point, 2.5 point). If questioner sets that if this question reach 3 point then jump to Thank you page.

```
<Logic>
        <Route type=″skipping″ id=″ro0001″>
            <Condition paperId=″p0001″ sectionId=″s0001″ questionId=″q0001″
isChecked=″yes″ score=″3″/>
            <Action paperId=″p0001″ descptionId=″de0016″/>
        </Route>
    </Logic>
```

Appendix A — Case Studies

A.1 URIs for Templates

Due to the space restrictions, please refer to the file:

- System: ../qsl/templates/system
- Questionnaire: ../qsl/templates/questionnaire
- Testing: ../qsl/templates/testing
- Voting: ../qsl/templates/voting
- Data: ../qsl/templates/data
- Security: ../qsl/templates/security

Appendix B – Use Cases

B.1 ENQUETE-BAISE: a General-Purpose E-Questionnaire Server for Ubiquitous Questionnaire

ENQUETE-BAISE is a general-purpose e-questionnaire server developing for ubiquitous questionnaire that can be used as a readymade e-questionnaire server component in various web service systems as well as an alone e-questionnaire server with general-purpose for various questionnaires. **ENQUETE-BAISE** can also be used as an e-testing server and an e-voting server with general-purpose by restricting its general functions and strengthening its security functions. **ENQUETE-BAISE** has been used since 2007.

Why we mention **ENQUETE-BAISE** here as a use case? Firstly, it is a general-purpose e-questionnaire server for ubiquitous questionnaire, thus it is able to contain and support services for various e-questionnaires and e-questionnaire systems. Secondly, it belongs to our lab achievement, so we can get the whole information well. That means we can analyze inside and outside, so that specify by QSL very well. Last but not the least, the ideal state of QSL is, implementing a QSL generator system to automatically generate e-questionnaire, e-testing, and e-voting systems. **ENQUETE-BAISE** is a general-purpose server for supporting e-questionnaire, e-testing, and e-voting, also is the *best candidate* as the QSL generator systems, owning to its universality and comprehensive.

In summary, QSL is endorsed as the specification to be used for data exchange, and specified input format to be used for automatically generating e-questionnaire, e-testing, and e-voting system on **ENQUETE-BAISE**. All the functions are specified by QSL.

B.2 A General-Purpose Offline E-Testing Environment

We implemented a general-purpose offline e-testing environment based on QSL to provide users with offline e-testing service to execute various offline e-testing. This environment has been applied to execute final test of Discrete Mathematic in University of Japan since 2015.

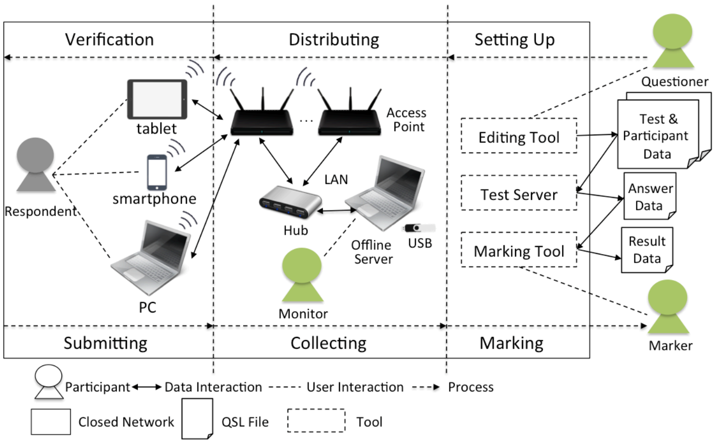
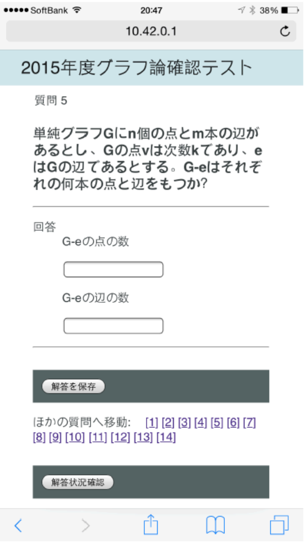


Fig. B2-1 Offline e-testing environment.

The overview of general-purpose offline e-testing environment is illustrated in **Fig. B2-1** below. An editing tool is set into the USB flash memory. A questioner uses the editing tool to specify all participants by QSL. After that, the editing tool will generate admission ticket for each respondent automatically. A questioner uses the editing tool to prepare all questions, which files are specified by QSL. An offline test server distributes questions to respondent through wireless LAN. The environment uses access point to support to test large number of the respondents easily and conveniently. The respondents confirm the questions from offline server. During the test, the monitor monitors the connection state of all respondents through offline server for online cheating activities, and the closed network will also block accesses from outside to connect to the offline server. After the test, all answers will be collected and integrated by offline server as QSL-format files. The offline server distributes and collects offline e-testing. Questioner only need to stick the USB flash memory into a PC, the PC can be the offline server to execute offline e-testing. The marking tool is also set into the USB flash memory. A marker uses the marking tool to mark the collected answers and give a result of the test as a QSL-format file. An Analyst uses the marking tool to analyze the result of the test automatically. All the functions and data in the environment are specified by QSL.



•••••SoftBank 20:47 38%

10.42.0.1

2015年度グラフ論確認テスト

質問 5

単純グラフGにn個の点とm本の辺があるとし、Gの点vは次数kであり、eはGの辺であるとする。G-eはそれぞれの何本の点と辺をもつか?

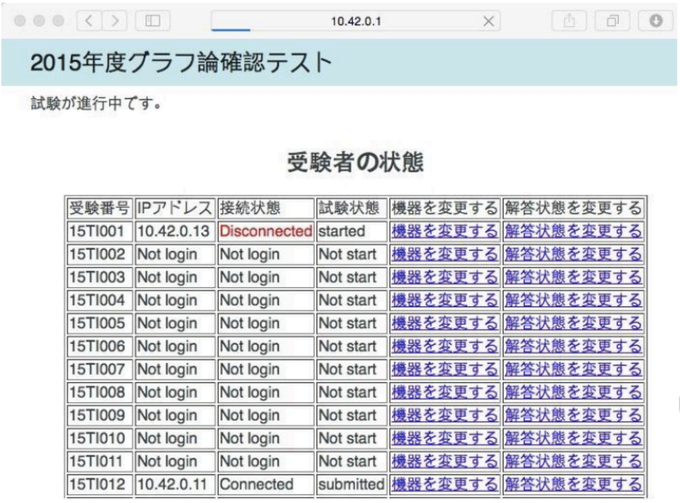
回答

G-eの点の数

G-eの辺の数

ほかの質問へ移動: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]

Fig. B2-2 Answer function.



10.42.0.1

2015年度グラフ論確認テスト

試験が進行中です。

受験者の状態

受験番号	IPアドレス	接続状態	試験状態	機器を変更する	解答状態を変更する
15TI001	10.42.0.13	Disconnected	started	機器を変更する	解答状態を変更する
15TI002	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI003	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI004	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI005	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI006	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI007	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI008	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI009	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI010	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI011	Not login	Not login	Not start	機器を変更する	解答状態を変更する
15TI012	10.42.0.11	Connected	submitted	機器を変更する	解答状態を変更する

Fig. B2-3 Monitor function.

As to the **selling point** of our environment, is implement the constructing server function as a bootable USB flash memory to make the environment can be easily and conveniently constructed even by teachers who are non-specialist of computer.

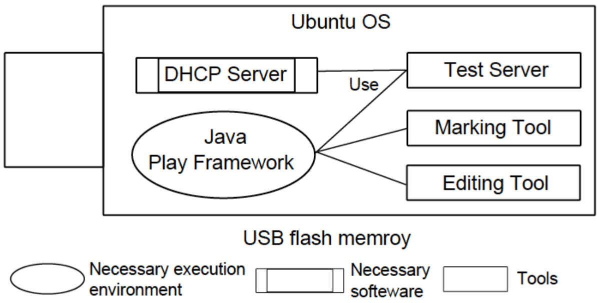


Fig. B2-4 Construction of USB flash memory.

The usefulness of the general-purpose offline E-testing environment is confirmed in the test. It worked well in the test, although the classroom is large and the number of respondents is many. All of the respondents did the test smoothly from start to the end of the test. All of steps in our test can be easily repeatedly. Besides, all data in the environment were specified by QSL well, so we can use the environment to do various tests in lots of middle schools and universities.

In general, QSL used in this situation, shows that QSL can be used as a standardized specification to support a large test and complex question types. The design of our environment is imaginative, but it needs QSL to support it and helps it to clear up what are necessary in the whole environment. QSL can specify all the functions and even the environment. Furthermore, QSL is a practicable data format to support data exchange.

B.3 An E-Voting Application of an Enterprise Information System on WeChat Public Platform

We developed an e-voting application as a web plugin of an internal enterprise information system, and extended on the WeChat public platform, as we all know that WeChat is one of the most-used smartphone applications in the world.

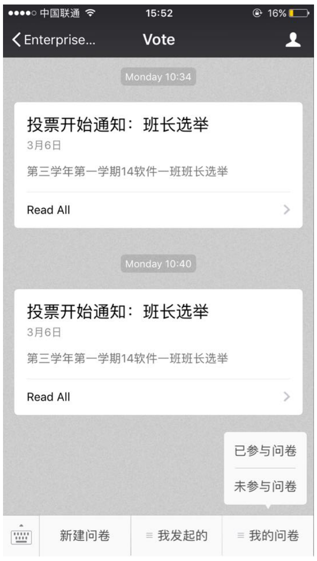
This e-voting application provides services on both Web and WeChat to create and edit e-ballot, to vote and check analyzed results if settings qualified voters to do, to distribute and collect by both e-mail and WeChat, and to make a list of all the drafts, distributed e-ballots as a role of a questioner, and all the submitted and uncommitted e-ballots as a role of a respondent. The e-voting application provides authentication service by the method of secret, and limits the channel on WeChat. WeChat is a platform execute on smartphone and tablet, helping send and reply messages to notify employees who are eligible respondent according to department, branch, or preference even chosen by sponsor or questioner based on internal process.

We specified by QSL to make a detailed requirement list of the e-voting application when we design it. QSL has been used directly. Moreover, all the data are specified by QSL. This e-voting application is used for 377 staffs in a Chinese software company since 2016. Because the company database keeps track of its staffs, departments, and projects for internal use only, we used NAT-DDNS tech to access WeChat, and did test on a WeChat enterprise account to do e-voting. The application is designed to run under Apache and Tomcat. We have made QSL the cornerstone of its development and likely to draw upon QSL.

Comparing with above-mentioned two use cases (even they are published), this is a real application in a real development in a software company; it is not just a research (the development is so-called a project not a product). Under developing, we are effort to implement according to a standardized specification and instructing under the design mind of our product manager. QSL devotes our pre-design stage and API. A so-called project may design under our imagination, but a real product is designed by our team (me is a developer for front end and our team members as developers for after end). In the process of development, we understood that both PM and developers need compromises (PM is too idealistic and has a lot of imagination but does not understand development but, developers have technology but does not understand business field).

Some screenshots of our application during development and test stage are shown as follow:

At first, the application totally divided into three main buttons, which have different functions. 1) to create a new poll, 2) to check the list of poll created by ourselves, 3) to check the lists of poll what we took part in (contains we have already participated or have not done yet). In addition, some notifications push to us, which want us to poll (we are as the role of the respondents). All the notification will be clicked and jump to the real poll page if you do not poll yet. If you poll for it, it will jump to the result page.

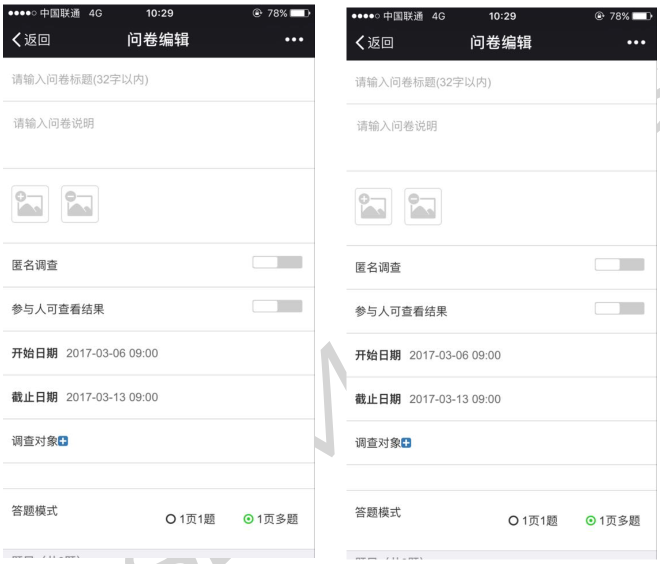


Monday 10:34

投票开始通知：班长选举
3月6日
第三学年第一学期14软件一班班长选举
Read All

Monday 10:40

投票开始通知：班长选举
3月6日
第三学年第一学期14软件一班班长选举
Read All

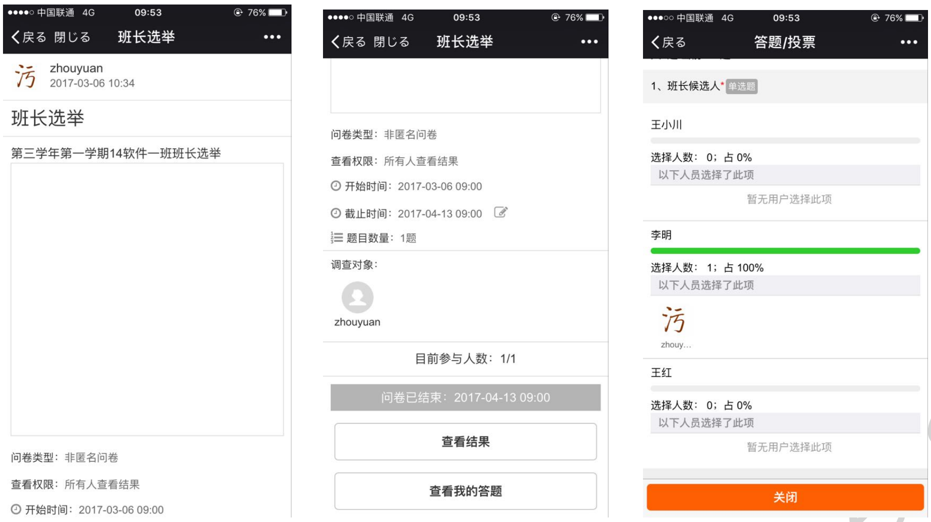


请输入问卷标题(32字以内)
请输入问卷说明
匿名调查
参与人可查看结果
开始日期 2017-03-06 09:00
截止日期 2017-03-13 09:00
调查对象
答题模式 ○ 1页1题 ⊙ 1页多题

请输入问卷标题(32字以内)
请输入问卷说明
匿名调查
参与人可查看结果
开始日期 2017-03-06 09:00
截止日期 2017-03-13 09:00
调查对象
答题模式 ○ 1页1题 ⊙ 1页多题

Fig. B3-1 Main buttons & notifications. **Fig. B3-2** Polling sheet.

Secondly, in the creating phase, our application allows you questioner to write poll title; explanation likes some subtitle or information about this poll; some pictures; settings like anonymous respondents, invisible results for respondents, beginning and ending time; respondent ranges; response modes like one page one question, one page multiple questions. As to the question types, there are three kinds types (radio button for single choice, checkbox for multiple choices, and text area for open-ended text). Each question and answer support to add, remove, force or not, choose to write other fields, and add pictures. Each polling sheet supports to edit, immediately publish, and just save.
Thirdly, as to the lists of polling as a questioner or a respondent, both roles are in two kinds of phases, which are before publish or not, and before poll or not. Therefore, as a role of a questioner, the lists show draft list and publish list. And as a role of a respondent, the lists show the stages of being already took part in and before taking part in.



●●●●○ 中国联通 4G 09:53 76%
く戻る 閉じる 班长选举 •••

zhouyuan
2017-03-06 10:34

班长选举

第三学年第一学期14软件一班班长选举

问卷类型：非匿名问卷
查看权限：所有人查看结果
开始时间：2017-03-06 09:00

●●●●○ 中国联通 4G 09:53 76%
く戻る 閉じる 班长选举 •••

问卷类型：非匿名问卷
查看权限：所有人查看结果
开始时间：2017-03-06 09:00
截止时间：2017-04-13 09:00
题目数量：1题
调查对象：
zhouyuan
目前参与人数：1/1
问卷已结束：2017-04-13 09:00
查看结果
查看我的答题

●●●○○ 中国联通 4G 09:53 76%
く戻る 答题/投票 •••

1、班长候选人* 单选题

王小川
选择人数：0；占 0%
以下人员选择了此项
暂无用户选择此项

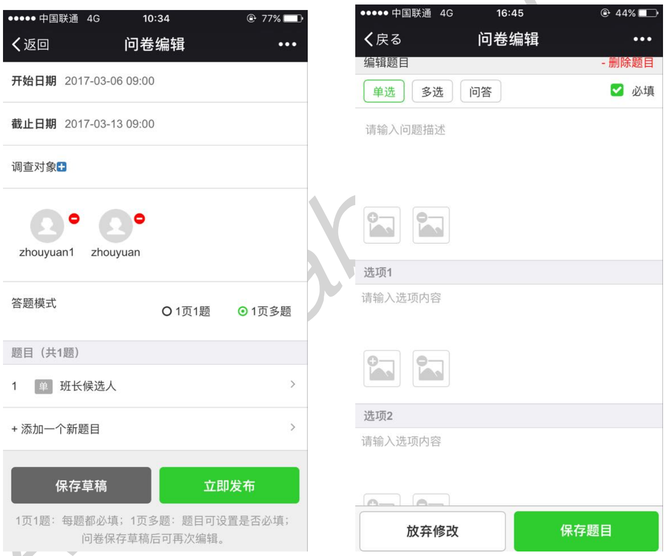
李明
选择人数：1；占 100%
以下人员选择了此项
zhouy...

王红
选择人数：0；占 0%
以下人员选择了此项
暂无用户选择此项

关闭

Fig. B3-3 Result and in detail.

Above-mentioned functions and screenshots are still in the development and testing stages based on WeChat development platform. We used a public WeChat account named “HaveFun” to do develop and test. After testing and before really publishing this application, all the information is private because it belongs to the Star-net company and concerns its departments and staffs.

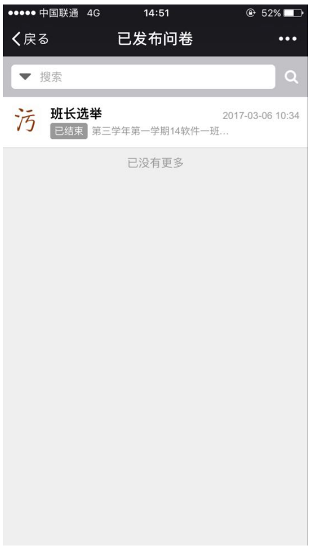


●●●●● 中国联通 4G 10:34 77%
く返回 问卷编辑 •••

开始日期 2017-03-06 09:00
截止日期 2017-03-13 09:00
调查对象
zhouyuan1 zhouyuan
答题模式 ○ 1页1题 ● 1页多题
题目（共1题）
1 单 班长候选人 ›
+ 添加一个新题目 ›
保存草稿 立即发布
1页1题：每题都必填；1页多题：题目可设置是否必填；问卷保存草稿后可再次编辑。

●●●●● 中国联通 4G 16:45 44%
く戻る 问卷编辑 •••

编辑题目 - 删除题目
单选 多选 问答 ☑ 必填
请输入问题描述
选项1
请输入选项内容
选项2
请输入选项内容
放弃修改 保存题目



●●●●● 中国联通 4G 14:51 52%
く戻る 已发布问卷 •••

搜索
班长选举 2017-03-06 10:34
已结束 第三学年第一学期14软件一班...
已没有更多

Fig. B3-4 Polling sheet in written. **Fig. B3-5** Publishing list.

It is worth mentioning that, the respondent list shows the departments within the company, concerns **security**. Except anonymous function during creating phase, doing an e-voting or polling event by this application, we need to verify the respondents’ identities. With the development of this application, we used to verify the email addresses or telephone numbers based on WeChat platform, because a WeChat account is registered by a real address or a telephone number. All the information about the staffs are stored in the internal database. We were coding to connect and compare these two fields between our database and

WeChat account based on WeChat development API. If the fields do not match the internal database, the respondent cannot join the company list. What's more, the eligible respondents only have one chance to poll.

There are some screenshots about respondent lists in a testing stage and a real condition.

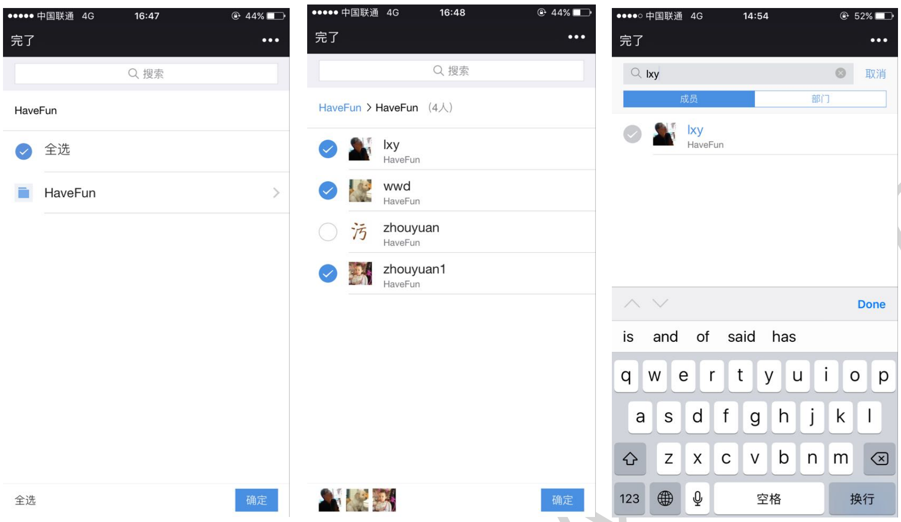


Fig. B3-6 Respondent list in testing stage.

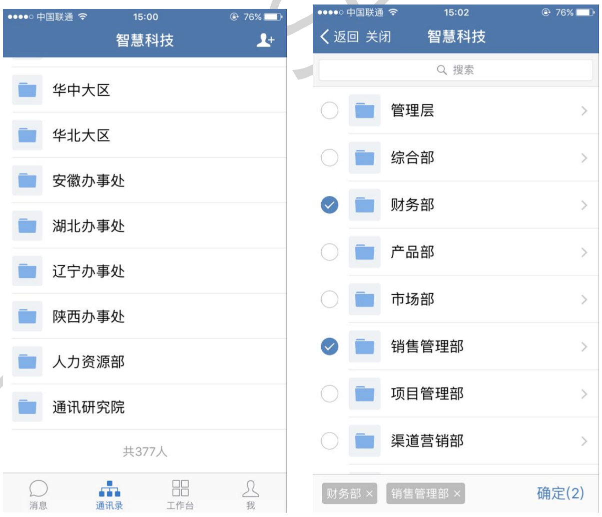


Fig. B3-7 Respondent list in real.

Appendix C – A List of Primitive Elements

category	entity		representation	remarks
phase	1	creating	direct-nominating	1)
			option-nominating	
			question-bank-option	
			direct-creating	
	2	verifying	registration	2)
	3	distributing	paper-delivery	3)
	4	submitting	answering	4)
			casting	5)
	5	collecting	collecting	--
	6	marking	marking	--
	7	analyzing	analysis	--
	8	counting	counting	--
	9	auditing	auditing	--
participant	10	sponsor	name contract age affiliation	6)
	11	respondent		--
	12	questioner		7)
	12	distributor		8)
	13	verifier		--
	13	analyst		--
	14	monitor		--
	15	marker		--
	16	counter		--
	17	auditor		--
	18	candidate		--
paper	19	description	title	--
			paragraph	--
			break	--
	20	section	--	--
	21	question	open-ended text	--
			matrix	--
			multi-choices	--
			ranking	--
	22	logic	skipping	--

			randomization	--
			piping	--
			extraction	--
	23	answer	NA-option	--
			other filed	--
			sample answer	--
	24	aligment	horizontal	--
			vertical	--
			drop-down list	--
	25	limitation	number-limited	--
			character-limited	--
			rate-control	--
	26	score	--	--
	27	formula	--	--
setting	28	time	start	--
			end	--
	29	media	--	--
	30	spelling-check	--	--
	31	language	--	--
	32	analysis	basic	--
			gap	--
			trend	--
	33	rate	--	--
	34	number		
component	35	server		
	36	device		
	37	interface		
	38	software		
	39	database		
security	40	authority		
	41	authentication		
	42	authority		
	43	anonymity		
	44	channel		
data	45	sample		
	46	result		
	47	report		
total number	--		--	--

Remarks:

1) Questioners create a question sheet, or choose questions from a question bank; questioners help to approve nominees as eligible candidates of a party or an organization, or help to approve the options to be presented to respondents.
2) Respondent registration is to identify the eligible respondents.
3) Paper delivery is to make available e-paper to overseas and uniformed respondents.
4) Respondents answer the question sheet, also it need blind *verification* because sometimes should authentication again in an e-voting event.
5) Respondents cast an individual answer sheet.
6) Sponsor is the person in charge to organize or support an event, contains to divide tasks to executors.
7) Questioner is a role of an executor, works for creating phase. He may design contents of question sheet, or verify nominees and options of candidates, or pick some questions from a question bank.
8) Distributor is a role of an executor, works for distributing phase. He may just trigger distribution event. He may be a teacher to distribute test paper. But sometimes he is not necessary.

Appendix D – Terminology

Analyst is one kind of participants who processes the collected answers.

Anonymity in security is to prevent any link between the voter and his unencrypted vote.

Auditing is a phase to provide services of recording, monitoring, and verification of audit data to make authenticity and accuracy of voting results, for the security of e-voting.

Auditing software is used to communicate with submitting system.

Authentication in security is to ensure only eligible respondents may cast submit only once before storing, and must ensure the casted papers are clearly separated from the identity of the respondent.

Certification server is used to provide services to validate the respondent s and staffs we will explain below to prevent any possibility of affecting results.

Client-side software is used to communicate with submitting server.

Collecting is a phase to collect the answers from respondents.

Counting is a phase to calculate the collected answers and get results, usually called tallying phase in e-voting.

Counting software is used to communicate with counting server.

Distributing is a phase to distribute e-paper to respondents. Submitting is to answer e-paper and send to submitting server, usually called voting phase in e-voting.

E-paper consists of settings, questions, and options. An e-questionnaire, an e-test paper, and an e-ballot have the extremely similar contents used to express a choice preference, collectively called e-paper.

Marker is one kind of participants to mark the responses of respondents and gives the results.

Marking is a phase to mark the answer sheet and get the results.

Monitor is one kind of participants who monitors whether illegal or dishonest behavior occurs or not.

Phase is the process of doing an e-questionnaire, e-testing, or e-voting event.

Questioner is one kind of participants who designs an e-paper, usually called examiner in e-testing,

Respondent is one kind of participants who answers the e-paper, usually called examinee in e-testing and voter in e-voting.

Server is used to store the collected results and server to provide registration services for respondents.

Setting Up is a phase to prepare software communicating with server and e-paper needed for an event.

Sponsor is one kind of participant who organizes and supports an event.

Appendix E – Papers and References

E.1 Papers

[1] Y. Zhou, Y. Goto, J. Cheng: QSL: A Specification Language for E-questionnaire Systems, the 5th IEEE International Conference on Software Engineering and Service Science (ICSESS 2014), IEEE, Beijing, China, pp. 224—230, 2014.

[2] Z. Wang, Y. Zhou, B. Wang, Y. Goto, J. Cheng: An Extension of QSL for E-Testing and its Application in an Offline E-Testing Environment, in J. J. Park, et al. (Eds.), Advanced Multimedia and Ubiquitous Engineering}, Lecture Notes in Electrical Engineering (LNEE), Springer, Vol. 352, pp. 7—14, 2015.

[3] Y. Zhou, H. Gao, J. Cheng: QSL: A Specification Language for E-Questionnaire, E-Testing, E-Voting Systems, in J. J. Park, et al. (Eds.), Advanced Multimedia and Ubiquitous Engineering, Lecture Notes in Electrical Engineering (LNEE), Springer, Vol. 393, pp. 255—261, 2016.

[4] Y. Zhou, H. Gao, J. Cheng: An Extension of QSL for E-voting Systems, in J. J. Park, et al. (Eds.), Advances in Computer Science and Ubiquitous Computing}, Lecture Notes in Electrical Engineering (LNEE), Springer, Vol. 421, pp. 87—96, 2016.

[5] H. Gao, Z. Wang, Y. Zhou, J. Cheng: Development of a General-Purpose Offline E-Testing Environment, the 12th International Conference on Computational Intelligence and Security (CIS 2016), IEEE, Wuxi, China, pp. 603—607, 2016.

E.2 References

[1] AddPoll, http://www.addpoll.com

[2] Ai Diao Yan (in Chinese), http://www.idiaoyan.com

[3] AISE Lab, "ENQUETE-BAISE," http://www.aise.ics.saitama-u.ac.jp/enquete/index.html

[4] AskForm (in Chinese), http://www.askform.cn

[5] A. D. Bethke: Representing procedural logic in XML, Journal of Software, Vol. 3, No. 2, pp. 33—40, 2008.

[6] A. D. Bethke: Using XML as a questionnaire specification language, IEEE Proceedings SoutheastCon 07, Richmond, pp. 127—131, 2007.

[7] Council of Europe: Electronic Democracy, https://wcd.coe.int/ViewDoc.jsp?id=141 0627

[8] Council of Europe: Legal, Operational and Technical Standards for E-Voting, https://wcd.coe.int/ViewDoc.jsp?id=778189

[9] J. Cheng, Y. Goto, M. Koide, K. Nagahama, M. Someya, Y. Utsumi, and A. Shionoiri: ENQUETE-BAISE: A General-Purpose E-Questionnaire Server for Ubiquitous Questionnaire, IEEE Asia-Pacific Services Computing Conference, IEEE—CS, pp. 187—194, 2007.

[10] Constant Contact, http://www.constantcontact.com

[11] CubeQuery (in Japanese), http://cubequery.jp

[12] Datagle, http://www.datagle.com

[13] Diao Cha Quan (in Chinese), http://www.diaochaquan.cn

[14] Dounano (in Japanese), http://www.dounano.jp

[15] EnableQ, http://www.enableq.com/cn/index.html

[16] Enq-maker (in Japanese), http://enq-maker.com

[17] eSurv, http://eSurv.org

[18] ExamSoft, http://learn.examsoft.com

[19] Examcoo (in Chinese), http://www.examcoo.com

[20] FC2 Vote (in Japanese), http://vote.fc2.com

[21] Fluidsurveys, http://fluidsurveys.com

[22] FormSite (online), http://www.formsite.com

[23] Google Form, http://docs.google.com

[24] Y. Goto and J. Cheng: Information Assurance, Privacy, and Security in Ubiquitous Questionnaire, 4th International Conference on Frontier of Computer Science and Technology, IEEE—CS, Shanghai, China, pp. 619—624, 2009.

[25] Helio, https://vote.heliosvoting.org

[26] Imitsu, https://imitsu.jp

[27] QTI: IMS Question and Test Interoperability Specification, http://www.imsglobal.org/question/

[28] Ischool Random Testing System (ver. 3.7.1), http://down.chinaz.com/soft/26226.htm

[29] Key Survey, http://www.keysurvey.com

[30] LoveKao (in Chinese) (ver. 1.2.1), http://down.chinaz.com/soft/29038.htm

[31] U. Madise and T. Martens: E-Voting in Estonia 2005. The first Practice of Country-wide Binding Internet Voting in The World. In: Krimmer, R. (Eds.), Electronic Voting 2006, Proc. Of the 2nd International Workshop, LNI GI Series, Bonn, Germany, pp.15—26, 2006.

[32] Mobo Survey, http://www.mobosurvey.com

[33] Moodle, http://www.moodle.org

[34] My3Q (in Chinese), http://mysurveylab.com

[35] MySurveyLab, http://enq-maker.com

[36] National Education Examinations Authority, http://www.neea.edu.cn

[37] OASIS Election Markup language (EML) Specification Version 7.0, http://docs.oasis-open.org/election/eml/v7.0/cs01/eml-v7.0-cs01.html

[38] OCR, http://www.ocr.org.uk

[39] oExam (in Chinese), http://www.orivon.com

[40] OQSS (online), http://www.oqss.com

[41] Opinion Stage, http://www.opinionstage.com
[42] Opoll (in Chinese), http://www.opoll.com
[43] PHPEMS Online Testing System (ver. 2.0.1), http://www.phpems.net
[44] Poll Every Where, http://www.polleverywhere.com
[45] POLYAS, https://www.polyas.com
[46] ProProfs, http://www.proprofs.com
[47] Qi Bo Testing System (in Chinese) (ver. 1.0), http://down.chinaz.com/soft/29998.htm
[48] Qualtrics, http://www.qualtrics.com
[49] QuestionPro, http://www.questionpro.com
[50] A. Schmidt, L. Langer, J. Buchmann, and M. Volkamer: Specification of a Voting Service Provider, In Requirements Engineering for E-voting Systems (RE-VOTE), First International Workshop on, Atlanta, GA, pp. 9—18, IEEE, 2010.
[51] Simple Voting, https://www.simplyvoting.com
[52] Smart Survey, http://www.smartsurvey.co.uk
[53] Smaster (in Japanese), http://www.smaster.jp
[54] Snappy Poll, https://www.snappypoll.com
[55] So Jump (in Chinese), http://www.sojump.com
[56] Sogo Survey, http://www.sogosurvey.com
[57] Statistic Bureau, Ministry of Internal Affairs and Communications, http://www.stat.go.jp
[58] Stone Poll (in Chinese), http://www.stonepoll.com
[59] SurveyMonkey (in Japanese), http://www.surveymonkey.com
[60] Survey Moz, http://www.surveymoz.com
[61] Surveyi, http://www.surveyi.com
[62] Tian Hui Diao Yan Bao (in Chinese), http://www.diaoyanbao.com
[63] Tomexam Online Testing System (in Chinese) (ver. 2.7), http://www.tomexam.com
[64] Tou Piao Wang (in Chinese), www.toutoupiao.com
[65] Vizzual Forms, http://www.vizzualforms.com
[66] E. van der Vlist: XML Schema, O' Reilly , New York (2002).
[67] Voice Poll, https://voicepolls.com/publisher
[68] M. Volkamer: Evaluation of Electronic Voting: Requirements and Evaluation Procedures to Support Responsible Election Authorities. Lecture Notes in Business Information Processing, vol. 30, Springer, 2009.
[69] Votenet, http://www.votenet.com
[70] W3C: Extensible Markup Language (XML) 1.0 (Fifth Edition), http://www.w3.org/TR/2008/REC-xml-20081126/
[71] W3C: XML Schema, https://www.w3.org/XML/Schema
[72] Yong Dao Offline Testing (in Chinese) (ver. 1.0), http://www.onlinedown.net/soft/49716.htm
[73] YouPoll, http://www.youpolls.com

Appendix F – Revision History

Date	Version	Contents and Changes
2013.05	v0.0	Prototype
2014.02	v1.0	First QSL Proposal
2014.09	v1.3	QSL Foundation in a frame
2014.11	v1.4	Add some comment, question type, logic type
2014.12	v1.5	Change some comments, formula, interface, functions
2015.01	v1.6	Change QSL structure into 3 parts
2015.05	v1.7	Add some details of phases, security, etc.
2015.11	v.1.8	Change new QSL structure in circle model
2016.08	v.2.0	Extend e-voting elements and attributes
2017.05	v.2.1	Revise some details of e-voting part
2017.07	v.3.0	Desperate the simpleType and ComplexType for resue
2017.08	To be continue…	

Acknowledgements

First of all, I would like to extend my sincere gratitude to my supervisor, Prof. Jingde Cheng, for his instructive advice and useful suggestions on my research. I am deeply grateful of his help in this research. Without his consistent and illuminating instruction, this research could not have reached its present form.

High tribute shall be paid to Prof. Yuichi Goto, whose profound knowledge of Computer Science and Software development. I am deeply indebted to him in English triggers for this beautiful language and rigorous scientific attitudes, which helps me a lot in the past five years.

Special thanks should go to my lab members who have put considerable time and effort into their comments on the draft.

Finally, I am indebted to my parents and my husband for their continuous support and encouragement.