The aim of SQBL – the Simple or Structured Questionnaire Building Language – is to provide a system-independent, open-source XML format for describing questionnaires. This is achieved by capturing key information in a form that can generate the content and logic for dynamic electronic surveys and rich paper forms.

By implementing the principles of Structured Questionnaire Design, SQBL is able to do this in a way that promises to provide rich metacontent for archives, while simultaneously reducing costs for researchers and questionnaire designers.

SQBL is Simple

One of the core principles is to maximise reuse without sacrificing usability. In practise, SQBL promotes reuse at a ‘questionnaire module’ level, rather than at a question level. SQBL also encourages an approach where common tasks for questionnaire design – such as question grids, response types and logic – have a single way to be documented. While in some respect these choices are limiting, for reuse or choice, they lower the barrier to entry when writing user-friendly tools that read and write SQBL and help ensure consistency.

As a document-structured XML format, SQBL stores one module in one file. This simple approach supports traditional HTTP URLs and XML fragment IDs, making referencing and RESTful access easier for software development.

SQBL is Structured

SQBL is the first format based on the principles of Structured Questionnaire Design encouraging the use of structured logic to manage question flow. Rather than being a burden, this change helps researchers more accurately describe questionnaire intent and articulate exactly which respondents will answer particular questions.

This structured approach also means that a SQBL module can be ‘compiled’ to provide the content and logic for any other format for questionnaire specification.

SQBL is a Questionnaire Building Language

SQBL fills a gap in current information technology by focusing on the creation of questionnaire specifications. Currently, there are few software solutions that are metadata-rich, open, interoperable and easy both for designers to use and developers to implement.

Rather than capturing metadata across the whole data lifecycle, SQBL focuses on the needs of traditional data collection. By taking a narrow focus, the SQBL XML schemas will remain small and approachable, ensuring speedy development and review, while also being a powerful tool for questionnaire-based data collection.

As such, SQBL aims to be a specialised tool in the documentation of social sciences and statistical information alongside and in support of existing standards, such as DDI and SDMX.

SQBL is Incomplete!

While many sections are stable, SQBL is still undergoing requirements gathering and further input being sought for a planned version 1.0 by early 2014. Additionally, developer support to convert SQBL documents into formats such as Blaise, PDF and HTML, or store and link SQBL alongside standards such as DDI will be needed. To find out more about how to contribute head to the SQBL website - http://sqbl.org

About Structure Questionnaire Design

Structured Questionnaire Design is the theory that questionnaire logic, such as skip patterns and filtering questions, is identical to that of computer programs and as such questionnaires can be designed more efficiently by eliminating skips in favour of other forms of control flow.

The paper detailing this theory, A Case against the Skip Statement was awarded 3rd place in 2013 IAOS Statisticians Prize and is awaiting publication. However, copies can be made available on request by emailing sam@sqbl.org
<QuestionModule xmlns="sqbl:1" name="IASSIST2013_survey" version="1">
  <TextComponent xml:lang="en">
    <Title>#IASSIST2013 Survey</Title>
    <TargetRespondent>All IASSIST Attendees</TargetRespondent>
  </TextComponent>
  <ModuleLogic>
    <Question name="Is_Best_IASSIST">
      <TextComponent xml:lang="en">
        <QuestionText>Is this the best IASSIST Ever?</QuestionText>
      </TextComponent>
      <ResponseType>
        <CodeList>
          <CodePair code="Y">
            <TextComponent xml:lang="en">Yes</TextComponent>
          </CodePair>
          <CodePair code="N">
            <TextComponent xml:lang="en">No</TextComponent>
          </CodePair>
        </CodeList>
      </ResponseType>
    </Question>
    <ConditionalTree name="Best_IASSIST_Branch">
      <SequenceGuide>
        <Condition resultBranch="Not_best_IASSIST">
          <ValueOf question="Is_Best_IASSIST" is="equal_to">N</ValueOf>
        </Condition>
      </SequenceGuide>
      <Branch name="Not_best_IASSIST">
        <BranchLogic>
          <Question name="Which_is_better">
            <TextComponent xml:lang="en">
              <QuestionText>Which IASSIST was better than Cologne 2013?</QuestionText>
            </TextComponent>
          </Question>
        </BranchLogic>
      </Branch>
    </ConditionalTree>
    <Question name="Favourite_moment">
      <TextComponent xml:lang="en">
        <QuestionText>What was your favourite #IASSIST2013 moment?</QuestionText>
      </TextComponent>
    </Question>
  </ModuleLogic>
</QuestionModule>

SQBL and Canard in practise

To the left is a short SQBL document with a 3 questions and some logic controlling the survey. We ask if this is the “best IASSIST ever”, and if the respondent says no, we want to know which one was better. Then we ask everyone their favourite moment as tagged with the conference Twitter hashtag.

The images below show this XML as previewed in the SQBL-based Canard Question Module Editor (http://bit.ly/CANARD).

When ‘yes’ is selected, question 2 is skipped or hidden...

IASSIST 2013 Survey

1. Is this the best IASSIST Ever?
   ☑ Yes
   ☐ No

2. Which IASSIST was better than Cologne 2013?

3. What was your favourite #IASSIST2013 moment?

... but when ‘no’ is selected, question 2 is available...

IASSIST 2013 Survey

1. Is this the best IASSIST Ever?
   ☑ Yes
   ☐ No

2. Which IASSIST was better than Cologne 2013?

3. What was your favourite #IASSIST2013 moment?