Implementation of Blaise 5 web questionnaires into an existing Business Survey Management System

Leif Bochis Madsen, Statistics Denmark

Abstract

For more than ten years Statistics Denmark has used Microsoft Infopath as tool for development of web questionnaires for our Business Survey Platform. Because this tool is going to be discontinued in a few years, we have considered alternative tools and settled for Blaise 5 as the main tool for developing web questionnaires in the coming years.

However, the entire Business Survey Platform also consists of a number of frameworks and systems in order to support automated procedures and effective work procedures in the data collection process. Implementing Blaise 5 as tool for web questionnaire development therefore implies a range of adaptations in order to support exchange of data as well as metadata between the various sub systems.

Important parts comprise exchange of data to and from our Business survey data store (XIS = Xml-based Input System), incorporation of Blaise questionnaires into our Business survey portal (VIRK) and our backend survey administration system (IBS).

The work has implied a large number of decisions with respect to details of communication between the various parts, i.e. which specific constructs should be needed to transfer data and metadata to and from the Blaise questionnaire. As a result, a basic template for questionnaires has been developed alongside a standard resource database including – beside layout standards – also constructs supporting the communication between Blaise and our backend system.

In order to make it possible to move a portfolio of approx. 65 questionnaires to Blaise in a few years, we also need to consider possible ways to auto-generate Blaise code from existing metadata in various formats.

1. Introduction

1.1 Statistics Denmark Business Survey Platform

In 2006 Statistics Denmark launched a new, centralized data collection system for all business surveys comprising data collected by paper forms, web forms, system-to-system transfers or – occasionally – telephone interviews.

The system consist of common practices and components for the entire data collection process, including a central data store and common administrative procedures like follow up on the data collection progress for each survey, providing user support, handling non-response and reminders, etc.

Furthermore, the survey portal is a part of the Danish governmental enterprise portal – the so-called VIRK portal[1] – including full integration with its login and authentication processes using digital signatures.

[1] See ref. [VIRK]

It was decided to develop the web forms using Microsoft Infopath and during the following years approximately 70 web forms has been developed. However, as Microsoft has announced the discontinuation of support for Infopath by 2022 it has become crucial to find a replacement in time.

This general system has proved a success by generalizing the procedures of administration and conduct of the wide range of business surveys into a common system. This system has become the unified entrance of surveys, questionnaires and sample data well known to all divisions working in the field of business surveys and this system was not ready for a replacement. It was therefore important to find a tool that could be integrated into this general framework.

After studying the market it turned out (in spring 2017) that there were only two realistic options available:

1. To develop our own tool for building html forms. Advantage: Easy to integrate into the existing setup. Disadvantage: A lot of work to build generators, standards, tools, and keep up with technology improvements.
2. To integrate Blaise 5. Advantage: It is a widely used system for building questionnaires for multiple platforms, it is maintained to cope with technology changes and Statistics Denmark had 25 years of experience using Blaise. Disadvantage: It seemed not easy to incorporate into our existing setup including security and authentication procedures.

After studying the two options it was decided to opt for the choice of Blaise 5 in the early spring 2018. However, because integration is easy as it resembles the integration of Infopath, it may still be possible to write hand-tailored html forms and use them alongside Blaise and Infopath forms. This option makes it possible to collect data even for surveys where Blaise 5 for some reason may not be suitable.

1.2 Data storage

The Business Survey data collection system comprises a central database for storage of all collected survey data whether from web surveys, system-to-system transfers or scanned paper forms as well as internal data editing. The XIS database has been in use for the last 12 years and form the backbone for all Business survey data. I.e., all forms – whether from Infopath or Blaise questionnaires – should be stored in XIS. The entire Business Survey system operates on this database with standardized procedures for upload of sample and background data (aka. “prefill”), dissemination to data editing and analysis, sample management, etc. The questionnaire engines draw updated information from this database upon request from a respondent to fill a form. This comprises sample data and background information as well as partly filled form data from e.g. previously interrupted attempts to fill the form.

The data in XIS are stored in a relational database in a structure similar to the Blaise “In depth” data partitioning option, i.e. the data from the collected forms are all stored as field-value pairs. However, there is a logical layer – the Data Report Definition – that may describe these data in a simple, hierarchical structure.

The data report definition consists of a collection of fields, each of which may be of type integer, real, string or “group”. The latter type consists of another collection of fields and may be repeated. From this simple model data may be transferred in various formats: as objects, as xml documents or in a relational format where each defined group is represented as a table.

2. Implementation of Blaise 5

2.1 Architecture of the Business Survey Portal

The business survey setups are based on pairs of frontend web servers and backend data control servers as shown in fig. 1 below.

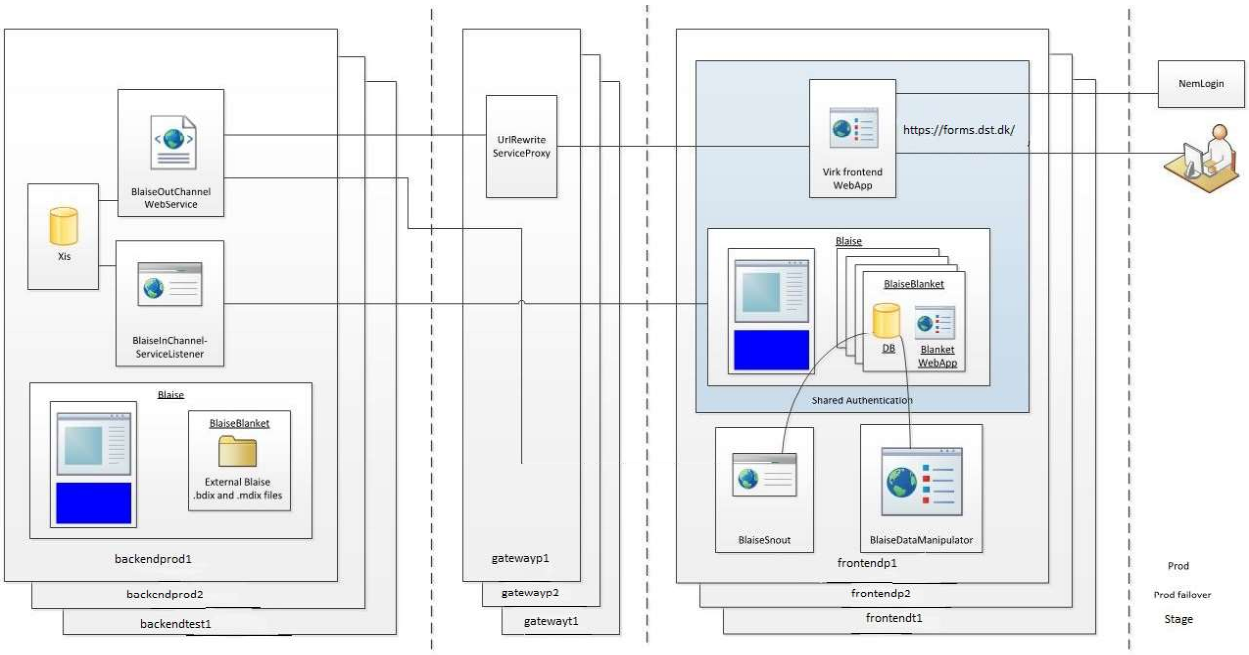

Fig. 1. Architecture of Business Survey Portal

The servers described on the figure comprise three frontend web servers, briefly named *prod*, *prod failover* and *stage*. The latter is a test server configured as an exact copy of the production servers and used for acceptance test. Each of the frontend servers are paired to an internal server used for controlling data retrieval and store.

The respondents are authenticated via the login procedure ("NemLogin"[2]). The authentication is carried out by an external identity provider which is part of the Danish Public Key Infrastructure. Via the authentication process the user passes an authentication token according to the SAML 2.0 standard.

All data transfers (via BlaiseInChannel and BlaiseOutChannel services) are controlled by the internal servers and operate through a relative Blaise data interface to the database on the frontend web server.

Beside these three server setups, there is also a parallel internal test setup, configured in the same way, except that the frontend server is not accessible from outside.

[2] See ref. [NemID]

2.2 Blaise as a form engine

Blaise is used in this system as a mere form engine. Originally, the requirement was that data should not be stored on the web server, but only exist as internal data in the web application. Due to the main architecture of Blaise we could not achieve that, but the database is filled with the necessary data upon start of the session and is removed again after completion, quit or timeout. The Blaise database thus works as a temporary database[3].

Data are transferred between XIS and Blaise questionnaires through the standardized procedures BlaiseOutChannel (from XIS to Blaise) and BlaiseInChannel (from Blaise to XIS) as showed in fig. 2 below.

These procedures are C# programs using the Blaise API for accessing Blaise data while they share the solutions for accessing XIS with similar procedures handling MS Infopath forms.

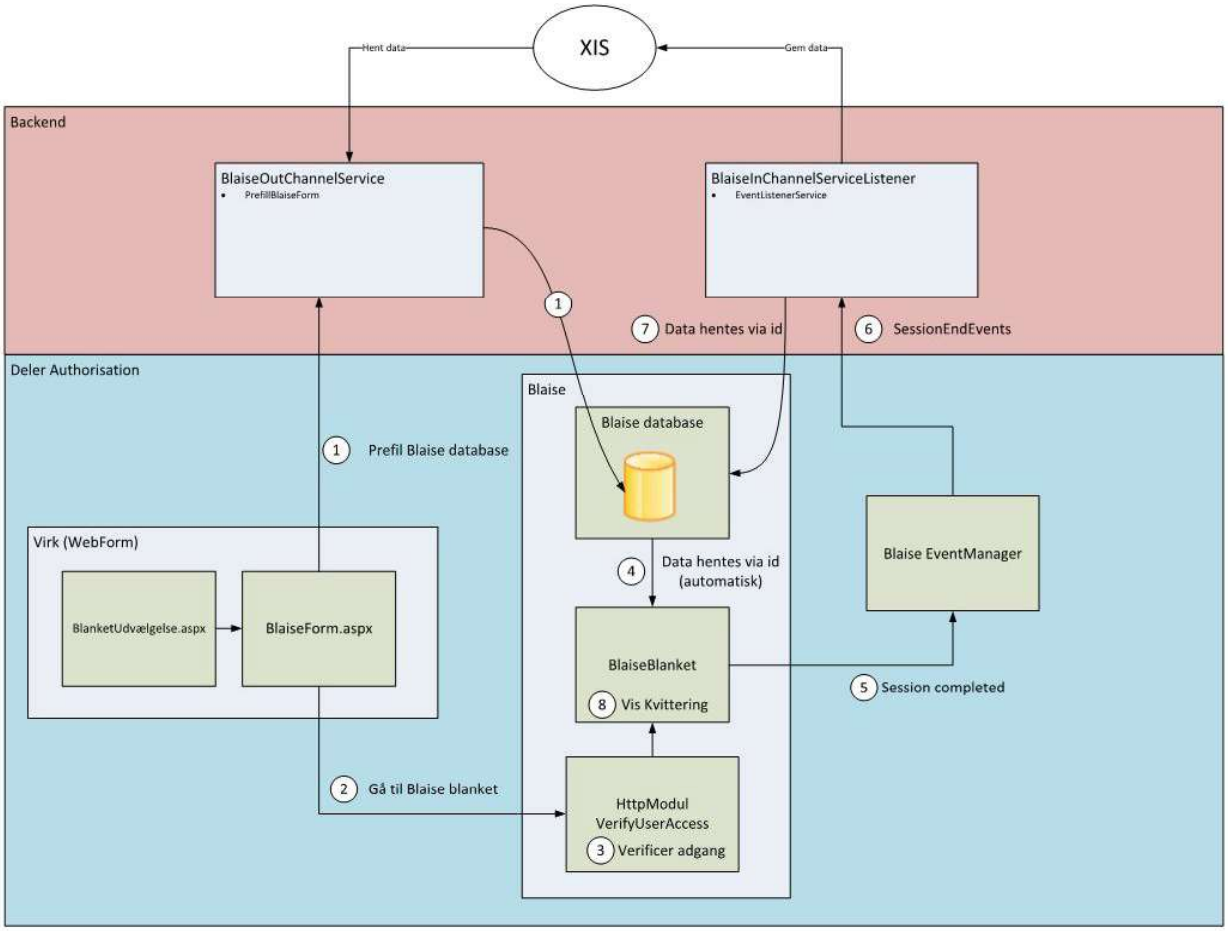

Fig. 2. Access to a Blaise 5 form

When the authenticated respondent requests a web form from the VIRK form overview (left in figure, "BlanketUdvælgelse.aspx") the web application transfers control to the BlaiseForm controller (similar procedures are available for forms developed in MS Infopath). The Blaise form controller asks the

[3] We are considering the option of storing Blaise data in an external database.

BlaiseOutChannel service to fetch sample data and possibly partly filled form data from the XIS data store and put it into the Blaise database identified by an ID (primary key) generated as a hash from the respondent ID and the authenticated name (steps 1). When this is done control – including verification of the provided ID – is transferred to the Blaise questionnaire ("BlaiseBlanket", steps 2 to 4). When the form has been completed the receipt page is displayed ("Kvittering", step 8) and meanwhile, the session is closed and within three seconds[4] an event is raised through the EventManager (step 5). This event is caught by the BlaiseInChannelServiceListener and the filled form data are fetched and deleted from the Blaise database and stored in the XIS data store (steps 6 and 7). Also, an email receipt is submitted to the respondent.

At last, control is returned to the form selection pane and the respondent may possibly fetch another form to fill.

A similar procedure is carried out when the respondent quits, though the status of the form will be set to "Draft" in the Business Survey Portal. Closing the browser will trigger a timeout in due time and meanwhile the form will be closed to other users, but may be reopened by the same user.

A pdf copy of the filled form may be generated from the receipt page, which is a standard feature in Blaise. Also, we have implemented a way to generate a copy of an already completed form from the form selection pane by launching the same process, though with an extra parameter to inform the Blaise questionnaire to jump to a separate page in the end of the questionnaire.

2.3 Necessary constraints to developed datamodels

In order to adapt Blaise data models to fit into the existing data collection scheme some limitations are implied.

1. An overall structure of the datamodel through a general template for the .blax file
2. Certain constraints to the definition of the questionnaire that allows a generic exchange of data between Blaise and XIS, i.e. storing Blaise Datafields in XIS
3. A general resource database to support the functionality required for proper communication between the Blaise Questionnaire and the Business Survey Platform

```
Datamodel example
Primary
   RespID
FIELDS
   RespID : string[100]
      // Form ID generated from respondent ID and authentication ID

   Stamdata : BStamdata
      // a block of data with info from the Business Survey Portal

   Metadata : BMetadata
      // a block of data about the respondent and the form, period etc.

   Indberetningsdefinition : BIndberetningsdefinition
      // a block of data defining the questionnaire
```

Fig. 3. Datamodel template

[4] The default of 10 seconds was considered too long time to wait for updates to take place.

The BStamdata block defines fields necessary for the handling of the form by the BlaiseInChannel service. The BlaiseInChannel service takes care of removing the data from the Blaise database and store the contents in XIS, and in order to do this basic information like Survey ID, Report Form ID, Report period, Entity Key (unique key per respondent per report period), version number of form, data collection period, etc. is needed.

The BMetadata block defines information needed for – mostly – personalization of the form. This includes information to publish on the receipt page, a series of links that should be used upon returning from the form to the Business Survey Portal, a help link specific for the survey, plus a couple of fields containing the state of the form – briefly: prefilled form, partly filled form, completed form. The form status information is also passed on to the Business Survey Portal when the form is closed.

The BIndberetningsdefinition block merely contains the definition of the survey questionnaire.

This template is accompanied by a template layout file, which define field and type mappings, selection of master page template and a couple of new page instructions[5].

2.4 Questionnaire definition constraints

As described earlier, data should be stored in XIS using a simple, hierarchical structure. Compared to the constructs available in Blaise the Form Report Definition for XIS is rather primitive, so in order to support a generic exchange of data a few constraints and decisions has been made.

Also, because the coming Blaise questionnaires will be replacing MS Infopath forms with already existing counterparts in XIS Form Report Definitions it might be necessary to allow for several conversion schemes. First priority, however, is that we can avoid mapping lists for specific survey and strictly keep to a generic conversion.

Basically, we took the Blaise metadata definitions as starting point, as these are by far the mostly detailed of the two metadata systems. As we while moving data back and forth are always going to have access to both metadata definitions, we may simply check whether it is possible to store the contents of the Blaise field into the type of the Form Report field. The basic mapping should map a Blaise field into a field in the Form Report Definition based on equal names (non-case-sensitive in both languages). For the simple fields we could set up a mapping scheme as follows:

Table 1. Mapping of field types

Blaise type	Form Report Type
Enumeration	String or integer: always store the code
Set	String: comma-separated list of codes
Datetype	String: YYYY-MM-DD format
Timetype	String: HH:MM:SS format
Real, Real ranges	Float
Integer, Integer ranges	Integer
String, Open	String
Classification	String
Block	Group
Arraytype	Group, repeated

[5] We have not yet started to use the layout language available from Blaise 5.3.0.

Integers and reals have been implemented in the same way as in Blaise, but strings are restricted to a maximum length of 2000 characters, due to storage in an Oracle database

A Blaise BLOCK is equivalent to an XIS Group, however, an XIS Group is the only XIS type that may be repeated. Therefore, it is necessary to define Blaise data fields comprising an array of e.g. strings in a surrounding block type, e.g.:

```
BLOCK BStringInBlockToSupportArray
FIELDS
     StringToBeRepeated : string
ENDBLOCK
FIELDS
    MyStringArray : array[1..] of BStringInBlockToSupportArray
```

These constraints may of course lead to avoidance of certain constructs that are "natural" to Blaise. To overcome this limitation we have defined a naming convention, so that we can define a data field with a postfix of "NONXIS", that will not be stored (alternatively, using auxiliary fields may also solve this problem). Then the rules of the datamodel must take care of the proper assignments of field values, e.g.:

```
Rules
   DatafieldInXIS.keep
   // move value(s) from DatafieldInXIS to Datafield_NONXIS
   // (once, at form initialization)
   Datafield_NONXIS.ask
   // move value(s) from Data_NONXIS to DatafieldInXIS
```

2.5 Standard resource database templates

The resource database should of course define the standard layout settings as decided by the design team. More important, however, is that the templates support the communication between the Blaise form and the general Business Survey web application.

First, we have defined a set of field references in order to pass background information about the respondent or survey to the form. For example, URL info that is used to direct the user back to the portal page, a help URL concerning specific help info, information needed for personalizing the receipt page and form status info that can be passed back.

Second, page templates (Master page, Receipt page, Abort page) are adapted to support the design chosen for our business surveys. On these pages are also defined a set of buttons needed to carry out the actions decided by the user. The user has the options of saving a draft or (if the form has been completed) to submit the form. When choosing the latter, the field containing the status info must be updated before continuing to the receipt page. This is defined as part of the events attached to the submit button.

As mentioned earlier, it is also possible to start a session only for printing a copy of an already submitted form. This feature is supported by checking a field reference: If the field has a value "Print form", the usual navigation buttons are disabled and only the options of printing the form and returning to the portal are available.

2.6 Verification

A module has been built in order to check concordance between the Blaise metadata and the XIS structure. The aim of the program is to verify that the data defined in the Blaise questionnaire definition can also be stored in the XIS data store. The first version of the program is ready and we are planning further releases.

In the backlog are improvements of the user interface so the questionnaire designers may easily verify questionnaires during development and possibility to plug it into the deployment procedures so installation of non-compatible questionnaires may be avoided.

2.7 Deployment of questionnaires

In order to ease deployment of questionnaires some scripts and programs have been developed comprising the necessary tasks for automating installation and adaptation to the Business Survey Portal scheme.

Automated installation of questionnaires on the web servers was the easy part as it just involved supplying the ServerManager.exe with the proper parameters like server name, Blaise server credentials and installation package.

Generation of a relative Blaise Data Interface was implemented by developing a C# program using the data interface API. The program is supplied with the installation package (.bpkg) which is actually a zip file. The meta (.bmix) and the manifest (xml format) files are extracted, the latter in order to extract the survey ID which is necessary in order to automate generation of the data interface.

Also, a powershell script was developed in order to include elements necessary to integrate the questionnaire with the Survey Portal Authentication setup into the survey web.config file. This script copies common elements and elements specific to the web servers (test and production servers) into the web.config file. Some of these settings might have been included in the installation package, but it was easier to manage these changes at survey installation and just generate a general installation package to use in any of the environments including development and pre-test (outside the portal environment).

2.8 Status, right now

September 3rd, 2018 we have succeeded to deploy the first Blaise questionnaire into the new version of the Business Survey Platform in Test, Stage and Production environments, where we are now supporting Infopath- as well as Blaise 5-based web forms[6].

We have started working on the implementation of further 3-4 Blaise questionnaires, which should all be ready for production by the end of 2018.

[6] Upon writing this paper we have not yet started to use this for real production, i.e. invited business respondents to fill the forms.

3. Future development

3.1 Blaise Questionnaire Generation

The existing Infopath questionnaires have been generated partly from metadata and descriptions stored in Excel files, one Excel project per survey. Of course, we want to use this information for generating the Blaise source code, too.

Therefore, we have started a project in order to define the requirements for developing a Blaise source code generator using this information. To make it work it will be necessary also to supplement the existing information with metadata from other sources.

The project aims at releasing the first version of a generator in the beginning of 2019.

3.2 Procedures for future deployment of newer versions of Blaise 5

Blaise 5 is constantly improving with new features added to the system twice a year and these features we certainly want to apply for our coming surveys.

Unfortunately, it often implies certain incompatibilities between Blaise versions. Therefore, we will have to upgrade the Blaise installations as well as the programs for integration in order to use the proper versions of the APIs so we may exploit the new possibilities.

As we still we need to run existing surveys without the need to constantly upgrade and redeploy them, we also need to refine and automate our test and deployment operations in order to support new and coming features of Blaise 5.

3.3 Load balancing

At last, we shall decide how to implement load balancing into the system. It would be possible to carry it out at login by the Virk Frontend Web app. An alternative might be to apply the Blaise built-in scheme for load balancing. We should make this decision soon in order to implement it in due time before the vast majority of the questionnaire portfolio has been moved to Blaise 5.

4. Conclusions

Blaise 5 has reached a level of maturity that makes it an obvious choice for setting up stable, scalable data collection solutions for complex questionnaires.

The resource database definition has also been improved considerately, and we are now able to develop and apply a standard. The making of a stable, organization-wide resource database is one of the most important and time-consuming tasks in order to set up an effective, professional environment for managing business surveys.

Still, upgrading from one version of Blaise 5 to another has been a challenge and we shall need to test thoroughly before jumping into the exploitation of neat, newer features.

5. Acknowledgements

Thanks to Lon Hofman and Roger Linssen from the Blaise team for discussing the possible ways to make integration work. Special thanks to Eric Munsters, CBS for kindly sharing info and documentation describing the architecture of the CBS Phoenix project. Also thanks to my colleagues, who took part in the development team, without which this paper would never have been written. Especially, Maja Kirchhoff Krølner and Nicolai Zangenberg, who implemented the crucial parts and provided me with nice illustrations.

6. References

[VIRK] https://indberet.virk.dk – portal for reporting forms to municipalities and government agencies etc. A brief explanation in English can be found at https://danishbusinessauthority.dk

[NemID] NemLogin ("nem" = Danish for "easy") , https://digst.dk/it-loesninger/nemlog-in/ - brief info in English: Agency for Digitisation, https://en.digst.dk