Using Blaise 5 in CAWI

Rogier Hellenbrand, Statistics Netherlands

Introduction

The presenter, Rogier Hellenbrand works for CBS (Statistics Netherlands) in the IT Department and is currently involved in the Phoenix Program as information analyst. His team is directly involved in development and maintenance of the Computer Assisted Web Interviewing (CAWI) channel.

This paper will cover the following topics:

- First I will give you an introduction on the Phoenix+ program and why we chose to use Blaise 5 as our instrument for our main survey method CAWI
- After that I will provide you with a quick overview of our solution to integrate Blaise surveys in our business process
- Then I will elaborate on some (fairly) new features of the Blaise suite for which we built specific functionality within our CAWI channel, as well as discuss some challenges we encountered along the way.
- And finally I will give you our judgement on the Blaise product.

Phoenix+ and CAWI

The Phoenix+ program was launched in 2015 with the expressed goal of replacing several data collection legacy systems by one, new application landscape. These legacy systems are technically fragmented, outdated, file based and often offline. In other words: they are end of life.

CBS' policy is to use CAWI as preferred interviewing method, because it is the most cost efficient. Philosophy is to start with CAWI and to use other modes only if necessary to get statistically valid results. So the CAWI channel is of the utmost importance. The architectural guidelines for our new CAWI channel were:

- Fully web based (multi platform, multi device) → for his purpose we decided to switch to Blaise 5 questionnaires
- Storage in database (SQL) instead of file based
- Generic for all types of surveys (social and economic)
- Message-based communication with other domains

Our first experience with using Blaise 5.x was in 2014. For a large production survey (80.000 companies were sent a questionnaire) we created a new CAWI channel, called PS-online. Special in this case was that we generated 12 different templates with in total 250 different routings using meta data from an old legacy system. Automated generation of templates based on meta proved to be possible, but it was quite challenging at the time.

In 2015 we started with the Phoenix+ program and reused lots of the PS-online software. But because the CAWI channel had to fit into the overall domain landscape, numerous alterations were necessary.

The first stage of Phoenix+ was completed august 2016. At present our KPI's are as follows:

- 30 of 125 surveys migrated to Blaise 5 surveys
- 600K+ records per year, response rate around 20 - 40% (for social surveys. For economical surveys where companies are obliged to respond, the response rate is much higher, up to 100%)
- Only 4 surveys need CAWI functionality which is not yet fully developed

In the future we will expand to other modes, probably using Blaise 5:

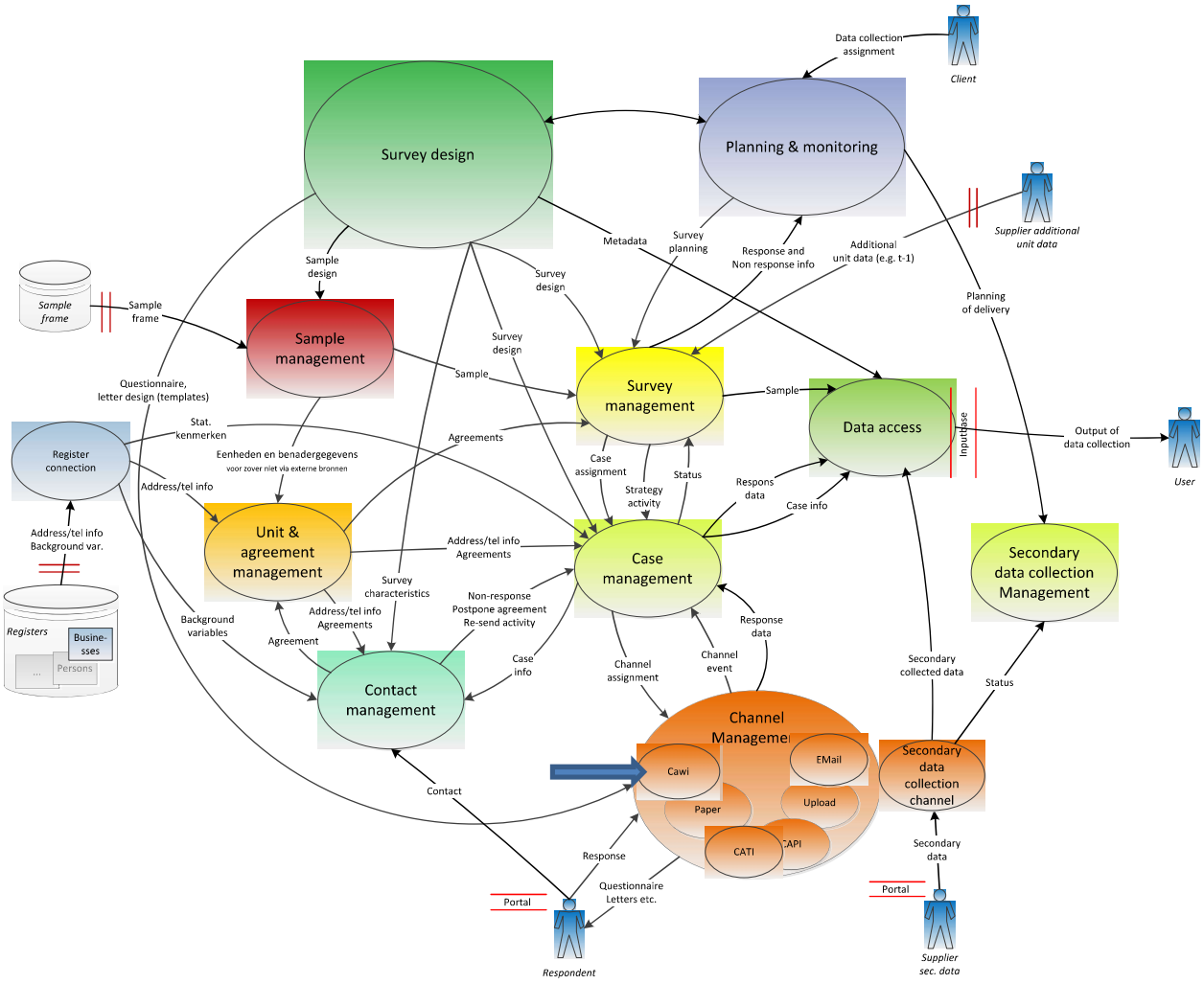
- Telephone
- Face to face
- Paper input (less and less)

And apart from this we are looking at an easier way to generate surveys (VLOP i.e. Survey template development process). Phoenix+ is scheduled to be completed during 2020.

Blaise 5 surveys within Phoenix+

As already said, Phoenix+ is an application landscape based on different domains and consistent messaging between these domains. See figure 1 for an impression of the complexity of this application landscape.

Figure 1: Phoenix+ domain architecture



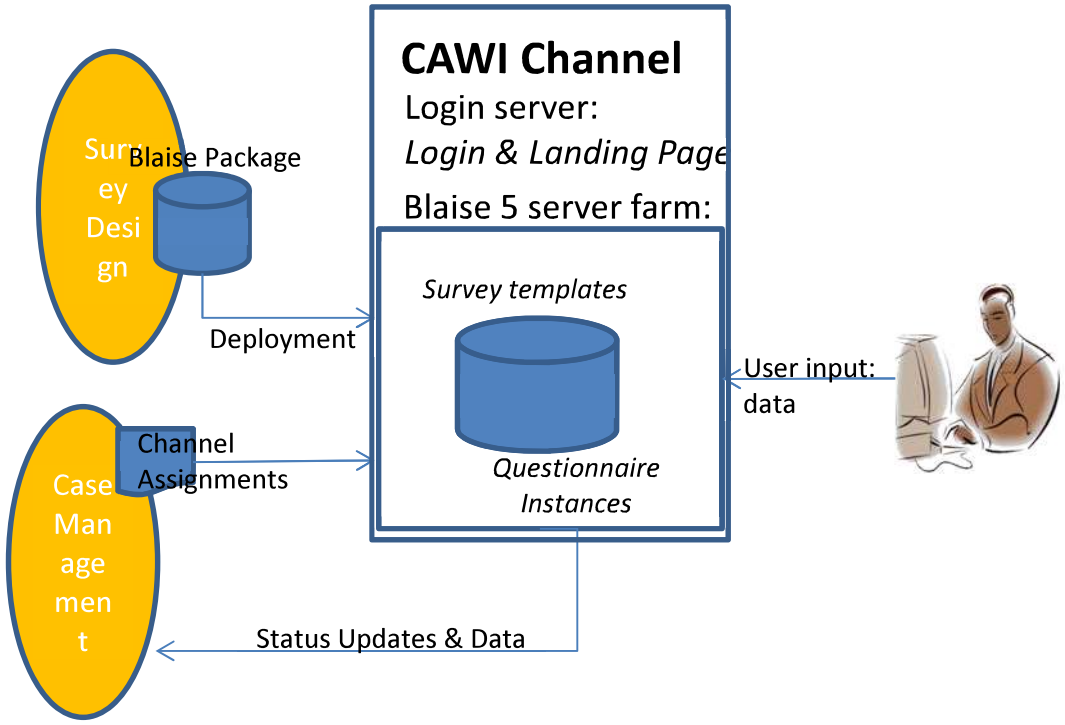
Within this picture the CAWI channel (blue arrow) seems to be only a small part, but in reality it is of extreme importance, as it is the main primary data collection channel.

In figure 2 you will find a simplified version of the interaction with and function of the CAWI channel in Phoenix+ context. The main functionalities are:

- Deployment of Blaise 5 questionnaire templates. All specific channel requirements are added to a package during deployment. This includes Master/Detail questionnaires, a switch for single or multiple reply possibility etc.

- Receive Channel assignments from other domain (Case Management). These channel assignments are transformed using the Blaise API to prefilled Blaise records
- Transforming Login credentials to a start survey command, opening the right instance of the questionnaire. The login functionality itself is generic and also used in other modes (like the upload channel or designated survey software).
- Possibility to fill in the survey safely (using standard Blaise functionality, but augmented with extra software to ensure safety)
- Re-enter survey session in a safe way after interruption/abort
- Acknowledge completed response (including possibilities to print a PDF with the response)
- Receive response and transferring this response to other domains.

figure 2: CAWI interactions with Phoenix+



In addition to these main functionalities the CAWI channel also provides:

- Heartbeat solution
- Audit trail solution
- Centralized logging

Newly incorporated Blaise 5 features

To be as real-time as possible, we incorporated Blaise events to trigger the distribution of response from the Blaise DB to other domains in Phoenix. In particular the EndQuestionnaire event is used by a windows service to read the newly sent-in record, to convert this to the desired format and to send it as a JSON message to the next domain. This functionality is purely on the background.

Although it should not be necessary, it is a fact of life that people make mistakes which are discovered when it is too late: a misspelled sentence, a glitch in the layout. So there is a need to be able to change a questionnaire template while the survey is already in production. Blaise knows whether the data model of the questionnaire is the same by comparing the checksum of the package. We used this knowledge in our own application: if name and checksum are the same, then allow the change, if not then do not allow. See figure 3 for the screenshot of this functionality

Figure 3: Changing the template in production

In our testing environment we noticed that starting the first instance of a newly deployed package can take up to 30 seconds. We call this the 'cold start up'. One of the most time consuming actions in this cold start up is loading the Blaise Metadata file BMIX. In earlier versions of Blaise this BMIX is kept in server memory for 1 hour. If the questionnaire is not used within this hour, the BMIX is cleansed out of memory. In Blaise 5.4 the time-out of keeping the BMIX in memory is made configurable. First tests did not lead to the expected results. We are still analysing this, but as we do not get any complaints from respondents, it is not a high priority issue.

Challenges with Blaise 5

In the earlier versions of Blaise 5 we experienced some difficulties regarding the backwards compatibility of the web layout after an upgrade. And because of the major technical changes from ASP.NET to MVC we even implemented a second CAWI channel to be able to support both Blaise 5.0.5 templates and Blaise 5.2.5 templates. But the transition from Blaise 5.2.5 to Blaise 5.3 went much more smoothly and the transition from Blaise 5.3 to Blaise 5.4 didn't even create ripples. It is safe to say that Blaise as an organisation has grown substantially in this field.

The second challenge we faced (and still face partly) concerns load testing. CBS uses virtual hardware and the results of our load tests were quite different from the results of similar tests done by the Blaise organisation on physical hardware. The point on which additional concurrent users clog the system lies substantially lower using virtual hardware. In our case we saw congestion on the webserver starting at around 80 concurrent users, whereas on physical hardware congestion didn't start until 300 concurrent users. We tried lots of different test scenarios, but in all of them we faced congestion at approximately the same number. And when the Blaise team tried their own test on our environment they faced the same. As of yet, we have not figured out why virtual hardware has such a bad influence on concurrency, but fortunately we can upscale the numbers of webservers to meet the demand (in the short run at least).

One other thing stands out in the performance area: Blaise 5.3 MVC in combination with Microsoft browsers is much slower than with other browsers. This seems to be caused by the JavaScript engine used. In one survey in particular this presented us with a big challenge, which we faced by compiling this template in Blaise 5.2.5, so the size tree functionality is not present in the build-up of the webpage. This resulted in a performance improvement of nearly 50 per cent on certain pages with a lot of elements on the page.

Evaluation using Blaise 5

In general Blaise 5 ticks the following boxes:

- Receiving of questionnaire instances (channel assignments) event driven
- Distributing response event driven
- Deployment of survey templates based on interface from survey design domain
- Stability of the whole process
- Backwards compatibility

Apart from this we have seen substantial growth in the Blaise organisation over the last couple of years. The one thing still bothering us is the performance of web based surveys containing tables in the combination Blaise 5.4 and Internet Explorer browsers.