BlaiseIS at Statistics Netherlands

Gerrit de Bolster, Statistiscs Netherlands

1. Introduction

At Statistics Netherlands the first BlaiseIS questionnaires, created in Blaise version 4.7, were put into production in 2007. In December 2007 a start was made to develop a complete integrated and automated environment for BlaiseIS 4.8 questionnaires. This environment, operational since 1 September 2008, contains a content management system and a dissemination system and supports personalized questionnaires. Additional a development environment for BlaiseIS 4.8 questionnaires including a questionnaire generator was developed. Questionnaires created with this development environment can be easily exported to the content management system to be put on the web server. As a result at Statistics Netherlands no IT-knowledge is needed anymore to create and deploy BlaisIS questionnaires. Recently functionality is added to support the creation of download portals in 3 different types.

1.1 Some figures

From 1-9-2008 until 15-2-2009 13 questionnaires for 11 household surveys with a total of 21 reporting periods were put into production. For all these questionnaires 138.685 persons/households were invited to fill in their data. In the same period 2 questionnaires for only 1 enterprise survey with a total of 5 reporting periods were put into production. The total sample for these 5 reporting periods is 30.626 enterprises.

The enterprise survey is compulsory and has a response of about 68%. The persons/households surveys are voluntarily and the response is depending on the size and complexity of the questionnaire. It ranges between 14% and 51%.

2. Design criteria

Although we had already some questionnaires running under Blaise 4.7 it was clearly not the solution we wanted for the future. Blaise 4.7 had too many limitations for that. At Statistics Netherlands the survey departments had a set of demands a system for Internet questionnaires should comply with. Some of them scanned the market for a standard product that supported their demands. None were found. Even they looked into several solutions of other governmental organizations without any success.

2.1 Question types and house-style

First of all it should be possible to apply a wide set of question types. The question types include simple numeric questions as well complicated tables and everything else in between. The texts should be dynamic and a set of fonts, text decoration and colors should be possible. Based upon the experiences with a off-line Blaise/Basil questionnaire for the yearly Production Survey a concept house-style for CAWI questionnaires was created. This house-style included also a predefined set of background colors and a division of the basic screen setup in panels.

2.2 Preview of the questionnaire

Questionnaire developers should be able to see and test their questionnaires before releasing for production. This should be possible without the need of other staff or complicated processes.

2.3 Predefined data

Another important demand is that the Internet questionnaires could be pre-filled with user dependent (personalized) data. This could be T-1 data as well information obtained from other registers or surveys. The data must not be included in the questionnaire itself so the data could be updated without the need of updating the questionnaire.

2.4 Identified access

As a consequence of the previous criteria only the respondent involved should be able to access the copy of the questionnaire with his (confidential) data (pre-filled or not). Therefore a secure login function should be available creating identified access.

2.5 Safe storage of data

Obvious all the confidential data, pre-filled or filled-in by the respondent, should be stored at a safe place were it can not be reached through the Internet by unauthorized persons.

2.6 Low-threshold questionnaire management

Furthermore it should be possible to add or remove questionnaires and reporting periods without the intervention of IT-staff. Especially for short-term statistics this process should be fast (on the fly). Besides, statisticians do have the habit to come up with last minute changes. This flexibility is also needed for the pre-filled data as sometimes samples are created just before sending out the invitations with the access information.

2.7 Regular collection of completed statements

To feed the statistical processes following the collection of the data (like cleansing) on regular bases it should be possible to collect the completed statements from the Internet questionnaires at all times. Therefore the moments this collection process is running should be flexible. Moreover it must be possible at design time to decide if a questionnaire allows correction of already completed statements after sending it. In the first case some kind of version indicator should be present.

2.8 Scalability

As it was hard to say how many Internet questionnaires will be created and therefore how many respondents will be filling in statements the web-server capacity should be scalable without a redesign of the Internet environment.

2.9 Infrastructure

Although this demand was not coming from the statistical departments it is a very important and obvious one. The technology involved should fit in the infrastructure of Statistics Netherlands. This means it should be Windows based.

3. Architecture

With the release of Blaise 4.8 it was possible to for fill all the design criteria described in the previous chapter. Unlike Blaise 4.7 it is based on what is called a server-park which makes it very scalable. The possibility to

create a separate data-server were all the confidential data is stored out of the reach of the users of the World *Wild* Web makes it very safe. A disadvantage is that the data-server itself is not scalable at this moment. A big advantage of using Blaise 4.8 should be that our questionnaire designers are used to work with Blaise. We soon discovered that this was not completely true. I will explain that in the next chapter.

3.1 Basic demands

At Statistics Netherlands electronic data collection as in CASI (self interviewing) is already in use since the early 1990's. With off-line questionnaire tools like CBS-IRIS, EDISENT, EDR and CBSQUEST (all built mainly in Blaise) a lot of primary data was and is collected interactively. Besides that there are more e-channels through which we collect files of all sizes from our respondents as well as other governmental organizations. To manage these flows we have two intertwined applications called MesDesk and QuestManager. These applications are also mainly built in Blaise. Naturally these applications should also support the Internet questionnaires. Therefore they were already extended to support the Blaise 4.7 questionnaires but in a limited way. Installing (and de-installing) the questionnaires was done by staff of the IT-infrastructure department and for collecting the finished statements a process of that same department was used running on predefined moments. For the Blaise 4.8 solution that should change radically. The department responsible for the IT-infrastructure, suffering under severe cut-backs in staff, should only be involved in case a new server should be added to the server-park or a new Blaise version should be installed. All the other processes should be managed by the (non-IT) staff of our data collection department using MesDesk and QuestManager.

3.2 Different Blaise versions

There are currently different Blaise versions in use at Statistics Netherlands. On our production network the default version is still Blaise 4.6. Recently new projects are now beginning to use the Blaise 4.8 version. For the Internet environment Blaise 4.8 is a must. However, on the production network both the applications MesDesk and QuestManager are running under 4.6, but, to interact with that environment from another server, Blaise 4.8 is needed. The IT-architecture group was afraid that introducing Blaise 4.8 on the production network could cause some conflicts, especially if the Blaise services are installed. So it was decide to introduce a separate server, called the Control server, were Blaise 4.8 Developer is installed. Processes that would communicate with the Internet environment should run on that particular server. To do so these processes are written in ASP and IIS is installed on the Control server. The MesDesk and QuestManager applications have full access to a certain folder on the Control server by which the can communicate with it. The ASP's on the Control server are invoked By MesDesk and QuestManager using de GNU-licensed (free) WGET.EXE program. This is some kind of command-line Internet browser.

3.3 Server-parks

Currently the Blaise server-park for our Internet questionnaires consists of a Management server, a Data server and 3 combined Web/Rules servers. On each of it is Windows 2003 Server installed as well as on the Control server mentioned before. For the download portals there will be an additional server-park created using another Data server and another Web/Rules server but the same Management server. Both server-parks (named "onderzoek" and "bestanden") have their own URL's (https://onderzoek.cbs.nl and https://bestanden.cbs.nl) and therefore their own certificates. The Web/Rules servers are connected to the Internet through a Proxy server. Figure 1 shows this configuration. The bars called DMZ, BackEnd and Production are different networks separated by firewalls.
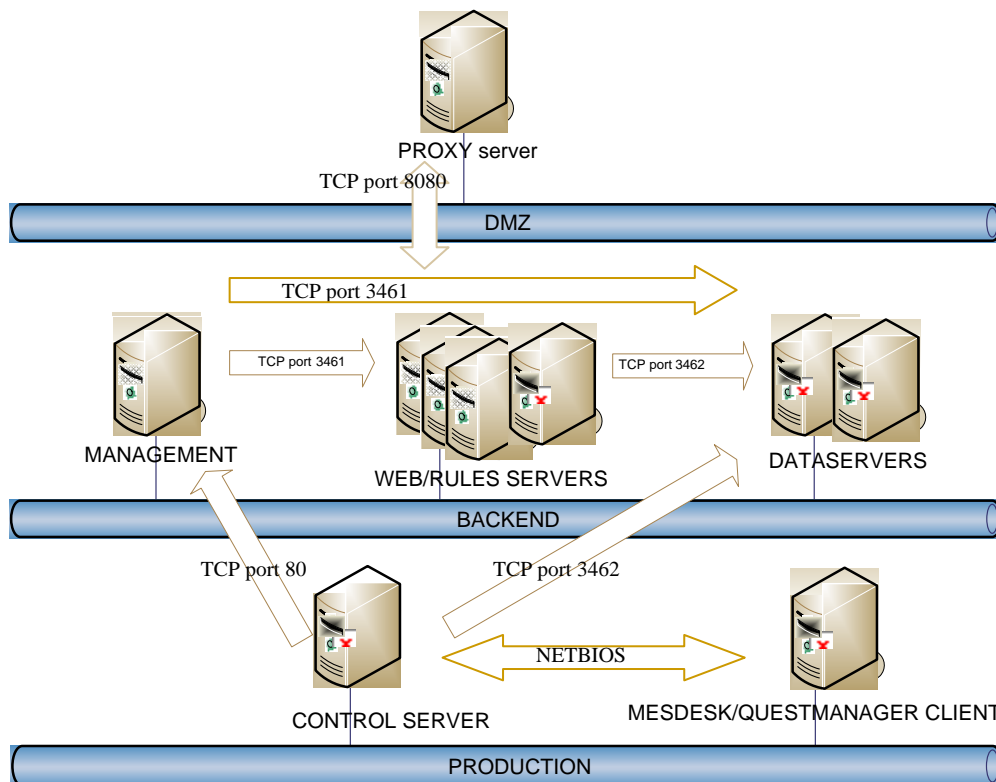
Figure 1.

3.4 Blaise(IS) components

The main processes are written in ASP running Blaise(IS) components for the final actions. This is possible as since version 4.8 all the Blaise Internet programs are in fact components. For interactive use interfaces are added. Without this it would not have been possible to create our BlaiseIS infrastructure. Only the documentation is not very sophisticated. As I am sitting next to the Blaise group in our office this was for me not a real problem.

3.5 Meeting the criteria

Introducing the server-park concept with the separate Data server and the extendible number of Web/Rules servers the criteria about safe storage and scalability was automatically met. Of course, using Blaise we also fit in the infrastructure of Statistics Netherlands. In the next paragraphs the criteria that are met by the chosen architecture are described. The remaining criteria are covered by the development environment as explained in the next chapter.

3.5.1 Identified access

The identified access was obtained by introducing a special questionnaire called "Login". This questionnaire is not called directly by the respondents but invoked through the starter page of every questionnaire. For that we had to change the standard Active Server Page (BiInterviewStarter.asp) that comes with Blaise. Starting an interview the respondent is automatically redirected to the Login questionnaire. This general Login questionnaire searches in the first place if there are more than one periods available for the questionnaire involved. If that is the case the respondent is requested to select the period he wants to fill in. In case only one period is available this period is automatically selected. The information about these periods is stored in a small standard Blaise file that

is available on the Data server with every questionnaire installed. Adding or removing periods this file is automatically updated.

After the selection of the period the respondent is requested to fill in his user number and access code. Using the open HOTP (HMAC-Based One-Time Password) algorithm (RFC 4226) included in a DLL the Login questionnaire checks if the given access code is valid for this questionnaire/period/user combination. This check is programmed as a hard Blaise check. Therefore a respondent cannot pass this part of the Login questionnaire without filling in a valid combination. If he does so, the user number is registered in a log file on the Data server (the" journal") together with a unique session-id supplied by IIS, the browser used, the screen resolution at the respondents' side and the login date and time. After that the respondent is returned to the starter page of the questionnaire. Before starting up the questionnaire the starter page checks if the call was coming from the Login questionnaire. As we even think that these checks are not safe enough we also check in the questionnaire itself if the Login information is registered in the log file at the Login questionnaire (accessing it as an external) to be sure the respondent logged in according to the rules.

Besides the questionnaires we also have open anonymous forms for which a respondent does not have to pass through the Login questionnaire. As soon the data has been filled in the statement is stored and cannot be retrieved by any respondent.

As we do have a general solution for identified access the questionnaire designer is not bothered any more by that issue.

3.5.2 Low-threshold questionnaire management

The questionnaires are installed by the MesDesk application. They are created by the designer using the development environment described in the next chapter. As a result a zip-file is created using an export option and delivered to the department in control of the MesDesk and QuestManager applications. Activating some menu options in MesDesk the questionnaire packed in the zip-file is automatically installed in the right server-park. After that action one more active periods can be added to the questionnaire applying the QuestManager application. To do so a file containing the user numbers has to be imported in the databases of QuestManager. This file must be produced by the statistical department. The corresponding access code are automatically generated by QuestManager and sent to the respondents in a letter. It is possible to divide the group of respondents for one specific period in subgroups. This is done so we can send the invitations out on different moments in time not overloading our reproduction department and the Web/Rules servers.

Using QuestMananager a period can be closed for a subgroup of respondents or removed completely. The whole questionnaire can be removed too using MesDesk. If a period is closed and a respondent still selects it the questionnaire will produce a page informing the respondent that he cannot fill in this period for this questionnaire anymore. If the period is removed completely the Login will not be offering this period anymore for selection by the respondent. If it was the only period left the Login questionnaire will inform the respondent that this questionnaire is inactive. After removing the complete questionnaire the respondent gets the standard error (HTTP 404) that the page cannot be found.

The closing and/or removing of the periods are done automatically by registering the closing and removing dates in QuestManager. All the processes can be scheduled so it is even possible to invoke them e.g. every weeknight.

3.5.3 Predefined data

With the QuestManager application it is also possible to transfer predefined data to the Data server. This can be done initially when a period is added to the installed questionnaire or later additionally. This data can consist of different Blaise files used as externals in the questionnaires or can be included in pre-filled statements in the Blaise result file. Of course, this data should be created by the statistical department to which the questionnaire belongs to. The pre-filled statements (with added data from the respondents) for a certain period are removed from the Blaise result file when a period is removed. Of course, the completed statements have then already been collected by the system.

3.5.4 Regular collection of completed statements

Another function of the MesDesk system is to collect the filled-in statements from the Data sever(s). This process checks the server-park(s) registered in MesDesk and retrieves the information about the installed questionnaires from the corresponding Management server. Next all the questionnaires are checked on

completed statements which are copied to the input folder of MesDesk to be transferred further on. The copied statements are not removed but a status field is changed in value so those statements are not retrieved again in the next run. Furthermore the records from the log file are copied giving us information about the browser used, the screen resolution at the respondents' side and the login/logout date and time. The logout date and time is written to the log file when the respondents submits the statement or interrupts the session. The collection process can be scheduled as well or invoked interactively.

The statements are not removed for a specific reason. We do support 2 types of questionnaires related to the times a respondent can send its statement. The type is defined at design-time. In the first type a respondent can send its completed statement only once. If, after sending it, the respondent logs in again he or she is informed that the statement has already been completed and cannot be accessed again. In case of the second type it is possible to open the questionnaire again after sending it. The data can then be edited and submitted again. In the statement a hidden version number is included that is increased automatically. In both cases we need the data to be stored on the Data server to be able to retrieve it after logging in and determine its status or even show the data.

4. Development environment

When we started to develop BlaiseIS 4.8 questionnaires we soon discovered that it was not so easy using the basic Blaise facilities as the BlaiseIS workshop. They are quite technical and Statistics Netherlands is aiming at questionnaire development by specialists not being IT-staff. Another policy of Statistics Netherlands is to apply methodology standards designing the questionnaires. These standards include not only the colors and looks of the questionnaires but also the basic behavior. Besides that the questionnaire designers need to be able to preview the questionnaire under development at any time. They could not use the preview facilities of Blaise caused by the fact that we were using a general Login questionnaire and some specific standard auxiliary files to store information about the questionnaire and its respondents on the Data server. To be able to do that these "externals" must be accessed through Blaise BOI-files that are opened in the rules. The standard Blaise environment is not able to detect that and therefore will not give an accurate preview of the questionnaire. And finally these BOI-files should be created separately and they include Internet host-dependant information.

4.1 BISmenu

For all these reasons we already started creating several auxiliary processes while developing the CAWI environment. Then we grouped them together in a Manipula menu to be able to invoke then easily. The development environment named "BISmenu" was born. To preview the questionnaires IIS was installed on our PC's and BISmenu was extended to be able to install the questionnaires on this so-called "localhost" Web/Rules server. We also included functions to create the necessary auxiliary files and copy them to the "localhost" Data server. The standard configuration that should be used for all the questionnaires (the "house style") was stored in a set of files. From these files emerged the so-called templates.

4.2 Design documents

The example questionnaires we developed to test the CAWI environment were still made by hand programming Blaise sources. In March 2008 it appeared that both the directors from the division for household surveys and the division of surveys for establishments signed contracts already in 2007 with other (semi-)governmental organizations involving large amounts of money to develop and exploit combined Internet questionnaires without being aware that we still did not have the environment to support these kind of questionnaires. The first questionnaire should be operational on September 1, 2008! This discovery put a lot of pressure on the development of the CAWI environment. We contacted the questionnaire designers to get information about these important questionnaires and to inform them about the technical standards that were chosen. The designers for household surveys were using a Word-document for the question texts and types and a Visio-document for the flow to define the questionnaires, until then still being CAPI. Checks and imputations could be found in both documents. Based on these documents their programmers created the Blaise sources by hand adding a lot of design definitions left out in the 2 documents. There were several designer dependant versions of these documents although they used mainly the same principals. As we were involved we tried to create a CAWI

questionnaire from that type of documents. Soon it appeared to be a burdensome task copying and pasting e.g. question texts from the Word-document into the Blaise source. As we are IT-minded the idea of a questionnaire generator using these design documents awoke.

4.3 The questionnaire generator

As I explained in the previous paragraph the first thoughts about a questionnaire generator emerged because of the burdensome copying of questionnaire texts from the Word-document. Because of that the first step was to create a Manipula set-up that created the FIELDS-section of a Blaise source using the Word-document saved as a text file. The next step was aimed at creating the RULES-section. We first examined how we could extract information from a Visio-document. The standard exports just created unusable graphic files. Then Lon suggested that being part of Microsoft-Office it probably could be accessed by an object-model. And so it was (thanks, Lon). Next we examined the Visio-documents created by the questionnaire designers. It appeared that they used it more like a package for drawing (like PowerPoint) than anything else. We soon discovered that Visio could be used in a much better way than that. Finally, in consult with the questionnaire designers, we designed a more standard and structured way how to fill-in a Visio-document as well as a Word-document so we could retrieve through the object model the necessary information to generate Blaise sources for CAWI questionnaires. The questionnaire generator, part of BISmenu, is still growing today as well the functionality of BISmenu itself.

4.4 Templates

Templates play a very important role within BISmenu. They define on a high level the type of questionnaire a designer wants to produce. Although we started with a lot of templates currently we reduced them to a basic set. Several differences were taken out and included in the Word-document as a setting. Thanks to the flexibility of the Blaise components all Blaise (configuration) files (BOI, BMF, BDP, BIP etc.) can be altered without bothering the questionnaire designer with difficult technical oriented interfaces. A template consists of a basic BIS-file (Blaise Internet Specification file), a BLA-file (Blaise source) and several configuration files.

The current set of templates supports the following types of Internet questionnaires:

01   – Normal questionnaire with identified access
A1   – Questionnaire with answers separated from identification
D1   – Download portal with identified access
D2   – Download portal with password protected zip-files
D3   – Public accessible download portal
F1   – Questionnaire with public access
M1   – Master/detail questionnaire with identified access (under development)

4.4 Creating a questionnaire

The basic process to create a CAWI questionnaire using BISmenu is as follows:

- Create a Visio- and a Word-document defining the questionnaire.
- Start-up BISmenu.
- Select the menu option "create new questionnaire" typing in the code of the template, the code of the questionnaire (part of the URL), its readable name and (needed in Statistics Netherlands) the size of the user number. The latter is only needed in case of identified access.
- Import the Visio- and Word-document and parser the resulting Blaise source.
- Select the menu-option "update BIS-file".
- Select the menu-option "add questionnaire" to install the questionnaire on your "localhost".
- Select the menu-option "period-info", create one or more new periods (if not yet there) and add the information to the "localhost".
- Select the menu-option "user-info", create one or more new users (if not yet there) and add the information to the "localhost".
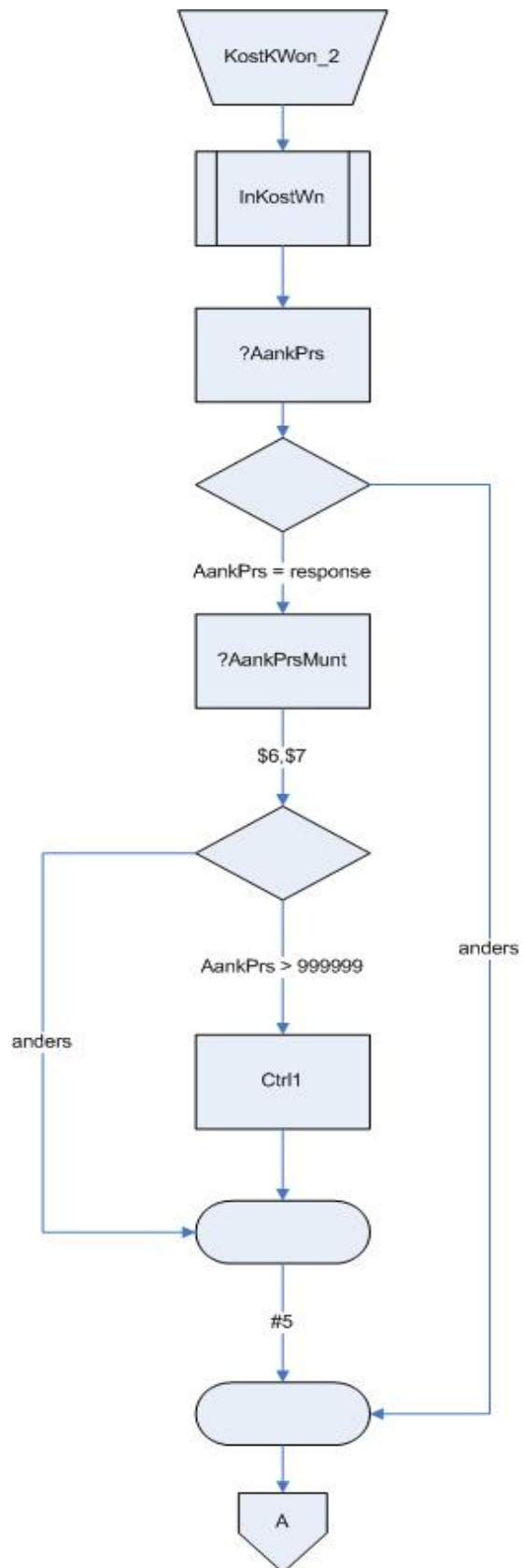
- Optional: select the menu-option "add data" to add pre-filled data to the "localhost". This data should be created on forehand; some functionality for this is included in BISmenu.
- Start up your Internet browser and check your questionnaire.
- If OK, select the menu-option "export questionnaire" to create a zip-file and send it to the production department.

4.5 The technology of BISmenu

BISmenu is build in Blaise (4.8.1). It consists of a set of Manipula set-ups and VB-script files (Visual Basic Script or VBS) to invoke all the necessary components (Blaise or Microsoft) and uses a set of adapted BlaiseIS files as ASP's (Active Server pages) and XSL's (Extended Style Sheets). To create a similar solution you need in-depth knowledge of Blaise/Manipula and BlaiseIS as well of ASP (which is VBS for IIS) and VBS (both very simple script languages) or bribe the author of this paper.

5. Some samples

5.1 Visio-document

```
                    KostKWon_2


                    InKostWn


                    ?AankPrs



                    ◇ ────────────────┐
                                      │
         AankPrs = response           │
                                      │
                    ?AankPrsMunt      │
                                      │
                    $6,$7             │
                                      │
       ┌──────────  ◇                 │
       │                              │
  anders    AankPrs > 999999     anders
       │                              │
       │            Ctrl1             │
       │                              │
       │                              │
       └──────────▶ ╭─────────╮       │
                    ╰─────────╯       │
                                      │
                    #5                │
                                      │
                    ╭─────────╮◀──────┘
                    ╰─────────╯

                    ⬠ A
```

5.2 Word-document

Blok: KostKWon

Condities:
$1: WonType <> [Boerdery] | [Kantoor]
$2: WonType = [Boerdery]
$3: WonType = [Kantoor]
$4: EenMeerH = [Een]
$5: EenMeerH = [Meer]
$6: AankPrsMunt = [Euro]
$7: (<> $6) en (AankPrsMunt <> DK|RF)
$8: RenteMunt = [Euro]
$9: RenteMunt = [Gulden]
$10: <> GemVerh(Gemeente,Verh)

Berekeningen:
#1: Trim(WonGem)
#2: Naw.Naam = 'Jan'
#3: Gemeente = Plaats(WonGem).Gemcodel
#4: WonCode = 15
#5: (= $7): VerkPrys = AankPrs + 1000
#6: (= $1): WonHulp = N_Hypoth + 10

Controles:
?1: = Plaats(WonGem) "Deze plaats komt niet voor in de zoeklijst."
?2: = GemVerh(Gemeente,Verh) "Deze verhuurder komt niet voor in uw gemeente."
?3: (= $6) of (AankPrs = response) en (VerkPrys = response) en (AankPrs <= VerkPrys)
"De aankoopprijs mag niet groter zijn dan de verkoopprijs."

Hulpgegevens:
WonHulp
[0..999]

Vragen:

WonGem	&RB80H1

In welke plaats <u>woont</u> u?
>>Klik op de knop "Zoeken" om de zoeklijst te openen.<<
STRING[35]=Plaats

WerkGem	&RB80H1

In welke plaats <u>werkt</u> u?
>>Klik op de knop "Zoeken" om de zoeklijst te openen.<<
STRING[35]=Plaats

WonType
 Type van de woning:

1.	Flat, appartement	[Flat]
2.	Eengezinswoning	[Huis]
3.	Boerderij	[Boerdery]
4.	Kantoor aan huis	[Kantoor]

Verh	&RB80H1

Van welke verhuurder heeft u de woning?
STRING[80]=GemVerh.Verhuurder?Gemcodel=Gemeente

5.3 Screen captures of CAWI test-questionnaires

6. Remarks

Microsoft, Windows, Microsoft-Office, Visio, PowerPoint, Visual Basic (and all the other ones I forgot) are registered ® trademarks of the Microsoft Corporation.

Blaise is a registered ® trademark of Statistics Netherlands.

MesDesk, QuestManager and BISmenu are unregistered names of our applications. No one should even think about it to copy © them!

I want to thank the Blaise team as they helped us a lot to get this CAWI show on the Internet road.