Internet interviewing using Blaise API

Bas Weerman, CentERdata, University of Tilburg

Abstract:

CentERdata is a survey research institute, specializing in Internet-based surveys. CentERdata is affiliated with Tilburg University, the Netherlands. CentERdata's most important activity is to carry out panel surveys through a telepanel: the CentERpanel. The CentERpanel is a panel of some 2500 households in the Netherlands. The members of those households fill in a questionnaire on the Internet every week. The CentERpanel is representative of the Dutch population.

In May 2000, CentERdata started doing Internet surveys using the Blaise for Internet system. It soon became clear that this system was not functioning optimally. CentERdata started to develop its own interview system based on the Blaise API calls. This system is now running for several months without any problems.

This paper will discuss (in short) our experience with Blaise for Internet and the possibilities of the Blaise API tools. It will describe the technical layout and the position of the Blaise API tools in our system.

Paper:

1. Introduction

CentERdata is a survey research institute, specializing in Internet-based surveys. CentERdata is affiliated with Tilburg University, the Netherlands. CentERdata's most important activity is to carry out panel surveys through a telepanel: the CentERpanel. The CentERpanel is a panel of some 2500 households in the Netherlands. The members of those households fill in a questionnaire on the Internet every week. The CentERpanel is representative of the Dutch population.

2. Conversion to a new system

In 1990, CentERdata started with a system of computer-assisted interviewing. All households had a small computer which was provided by CentERdata. With this computer the respondents phoned in to the CentERdata main server to receive their interviews. After the respondents filled in their questionnaires, they contacted the server again to deliver their answers. The system was built in-house by CentERdata programmers.

After 10 years it became clear that the interview system needed replacement. There were new requirements and the (now) old computers at the respondents' homes began to drop out and were not replaceable. In March 2000, CentERdata switched to a new system based on the Blaise for Internet package. The old system had been in use for almost 10 years, so the new system had to face a number of conditions:

- The system should look and feel almost the same as the old system. The conversion should not cause any problems for the respondents to fill in the questions (specially for the elderly).
- CentERdata needed an instrument before the actual questionnaire. This instrument was needed for identification of the respondents, by placing additional buttons on the screen for each respondent. Multiple buttons per respondents where possible (for more questionnaires per respondent). The Blaise questionnaire was then called with the appropriate household number and respondent number.

- CentERdata needed real-time logs of the users activities so online help could be provided.
- CentERdata uses setop boxes for panel members without computer. Those setop boxes don't support javascript, so all the intelligence should be on the server side.

In order to comply with those conditions, CentERdata decided to write a program between Blaise and the client browser. This program was called C2B (CentERdata to Blaise).

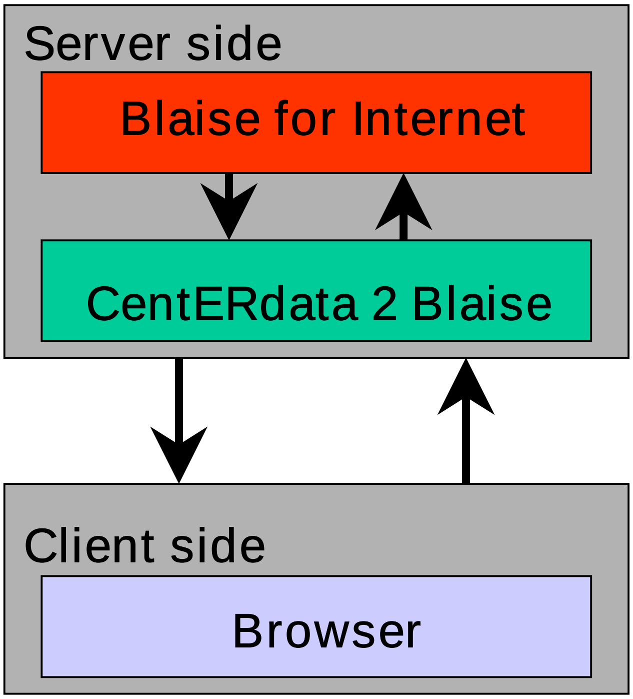


Figure 1: Layout C2B system

3. The layout of the C2B system

The C2B system was written in Delphi and installed at the server next to Blaise for Internet. The general structure of the program is as follows:

The communication between Blaise for Internet and C2B is completely in HTML. When a user logs in and requests an interview, C2B sends an html form to Blaise for Internet. The output of Blaise for Internet (in html) is transferred and sent to the browser. The question is shown in the client browser. The answer is sent back from the client browser to C2B, which transfers the HTML code with the answer from the browser into something understandable for Blaise (also in HTML code). There is no checking of the answer. A possible wrong answer can be sent to Blaise. If an answer is wrong (for example a range error), the output of Blaise is the same answer (because BIS didn't accept the wrong answer and in reply asks for the same question). C2B resends the question to the client browser, together with an error message. The error message is nothing more then: "There is an error in your answer or maybe you didn't answer.".

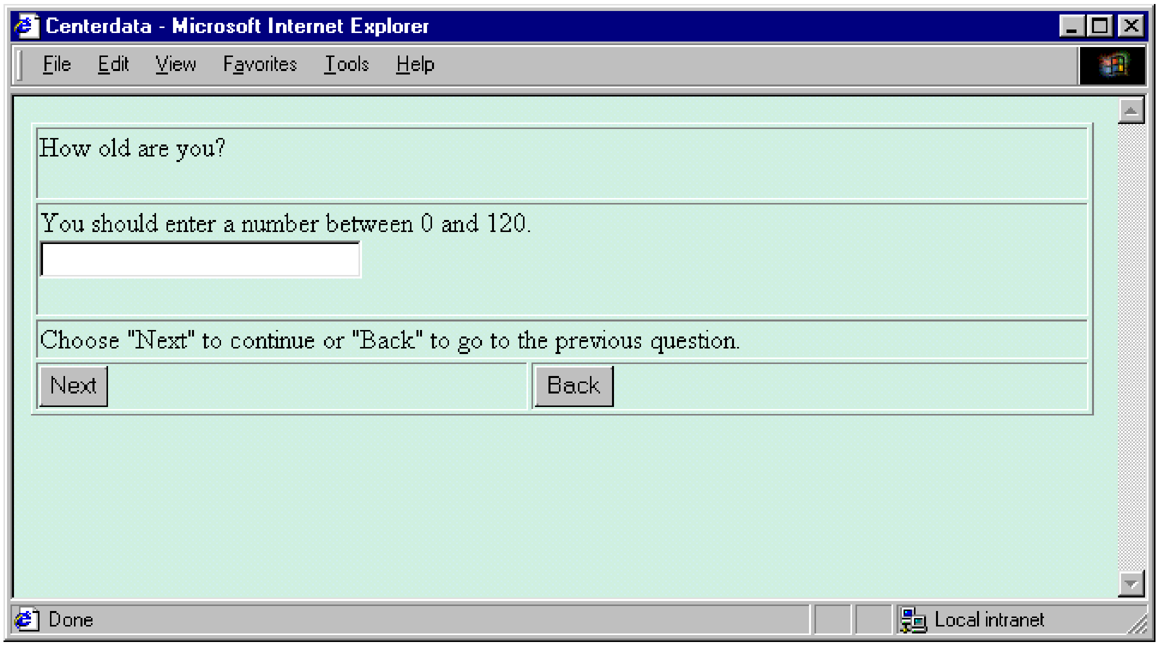
The Blaise for Internet package does not support tables. CentERdata needed the table support. In order to view tables on the screen, CentERdata has built a parser inside the C2B service. This parser gets the Blaise question text and looks for some special code. In this way, CentERdata has built a number of new features.

4. C2B features

C2BBack()

Normally, only a ‘Next’ button is placed on the screen, sometimes accompanied by a ‘Refusal’ or ‘Don’t know’ button (if those options are assigned to the field). To add a ‘Back’ button, the text C2BBack() is added to the Blaise question text. When the C2B parser sees this code in the Blaise question text, it knows that a back button should be placed on the screen. The Blaise code looks like:

```
Question2 “How old are you? C2BBack() “: 0..120
```



How old are you?

You should enter a number between 0 and 120.

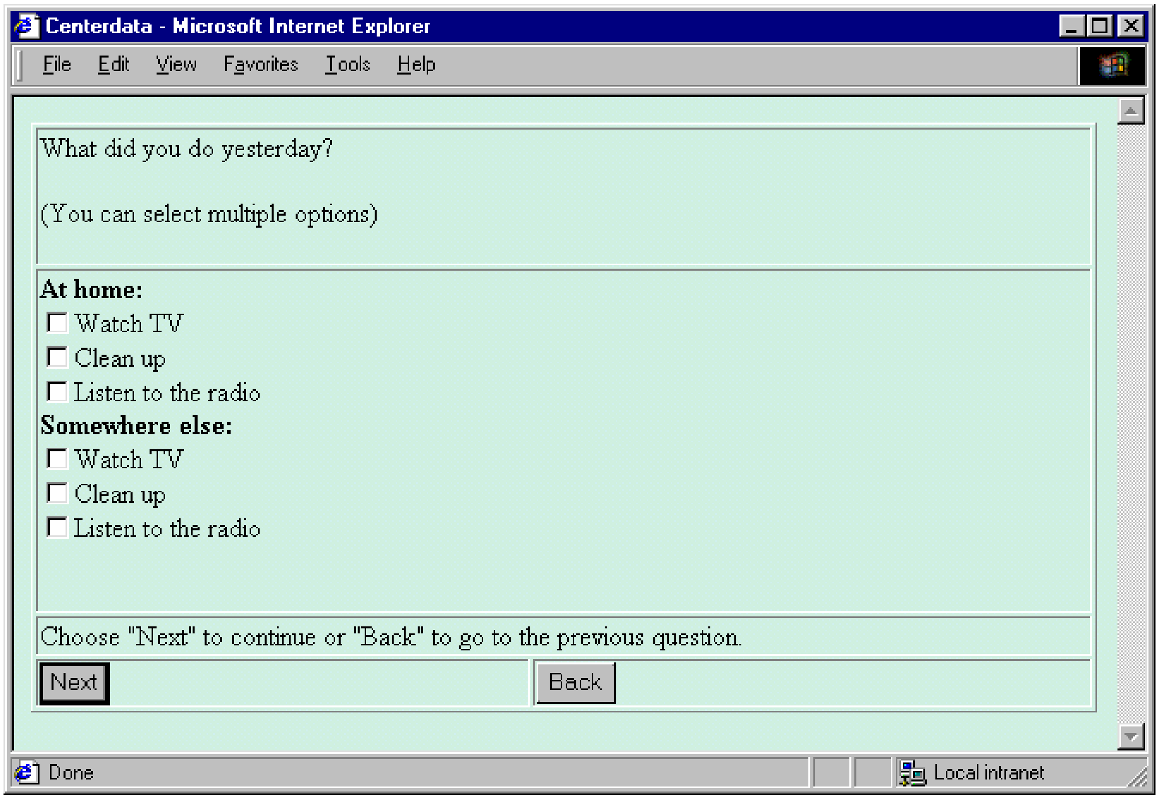
Choose "Next" to continue or "Back" to go to the previous question.

Screen 1: The use of C2BBack() in the question text

C2BNoOption()

Normally, all the answer possibilities of a set question are shown on the screen. To avoid a checkbox to appear before an option, it is possible to use C2BNoOption() in the set type.

```
Question3 “What did you do Yesterday?@/@/(You can select multiple options) C2BBack() “:
(           a1 “@b At home @b C2BNoOption()”,
            a2 “Watch TV”,
            a3 “Clean up”,
            a4 “Listen to the radio”,
            a5 “@b Somewhere else:@b C2BNoOption()”,
            a6 “Watch TV”,
            a7 “Clean up”,
            a8 “Listen to the radio”)
```


Centerdata - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Next
Back
Done
Local intranet

What did you do yesterday?

(You can select multiple options)

At home:
□ Watch TV
□ Clean up
□ Listen to the radio
Somewhere else:
□ Watch TV
□ Clean up
□ Listen to the radio

Choose "Next" to continue or "Back" to go to the previous question.

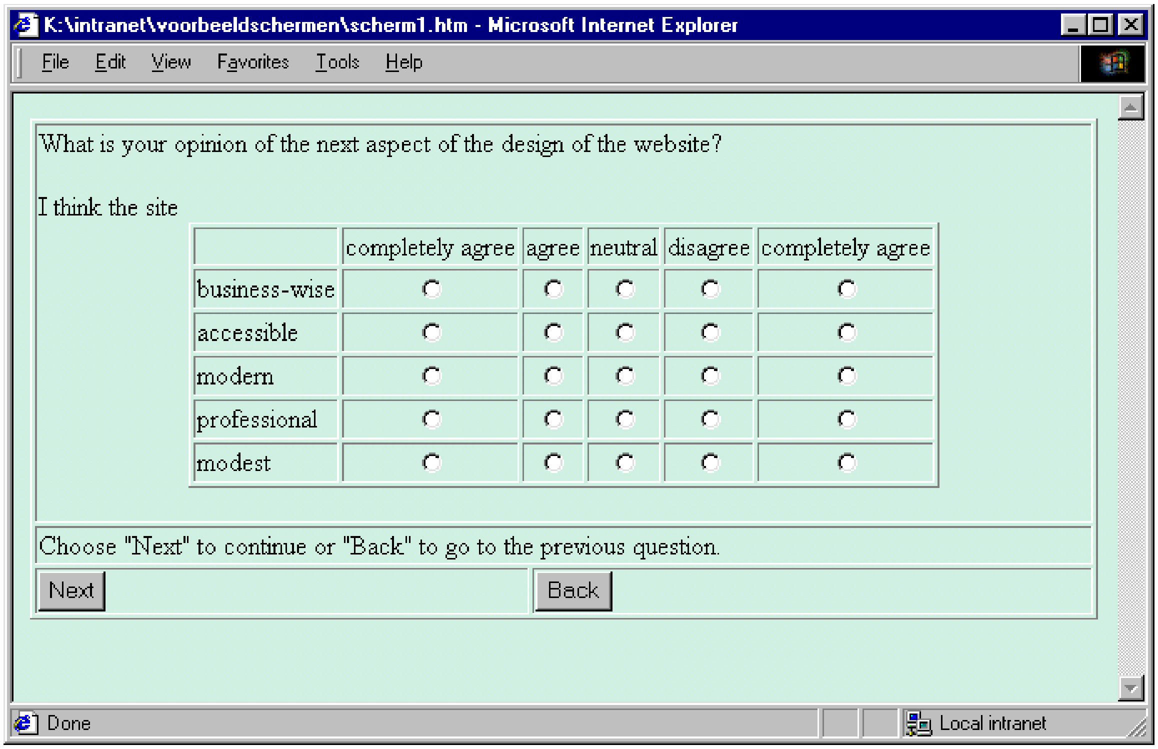
Screen 2: The use of C2BNoOption() in the question text

C2BList()

To support tables on the Internet, CentERdata has built the C2BList() option. With this option it is possible to place tables on the screen. The output of a question built with C2BList is of type String. The String is then transformed in the appropriate fields. This is done entirely in Blaise. Normally, the C2BList question could be of type Auxfield.

```
Fields
        v091 "The website is business-wise?   (o1 "completely agree",  o2 "agree",  o3
                "neutral",  o4 "disagree",     o5 "completely disagree")
        {…}
        Question4 " What is your opinion of the next aspect of the design of the website?
                @/@/I think the site
                C2BList(5, 5, 55555, 11111, “”, “”, ‘completely agree’, ‘agree’, ‘neutral’, ‘disagree’,
                ‘completely disagree’, ‘business-wise’, ‘accessible’, ‘modern’, ‘professional’,
                ‘modest’) C2BBack()”: String

Rules
        Question4
        Str2Arr (Question4, 5, TempArray, right)
        right = 1 INVOLVING(Question4) "You forgot to answer a question"
        v091 := TempArray[1]
        v092 := TempArray[2]
        v093 := TempArray[3]
        v094 := TempArray[4]
        v095 := TempArray[5]
```


K:\intranet\voorbeeldschermen\scherm1.htm - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Next
Back
Done
Local intranet

What is your opinion of the next aspect of the design of the website?

I think the site

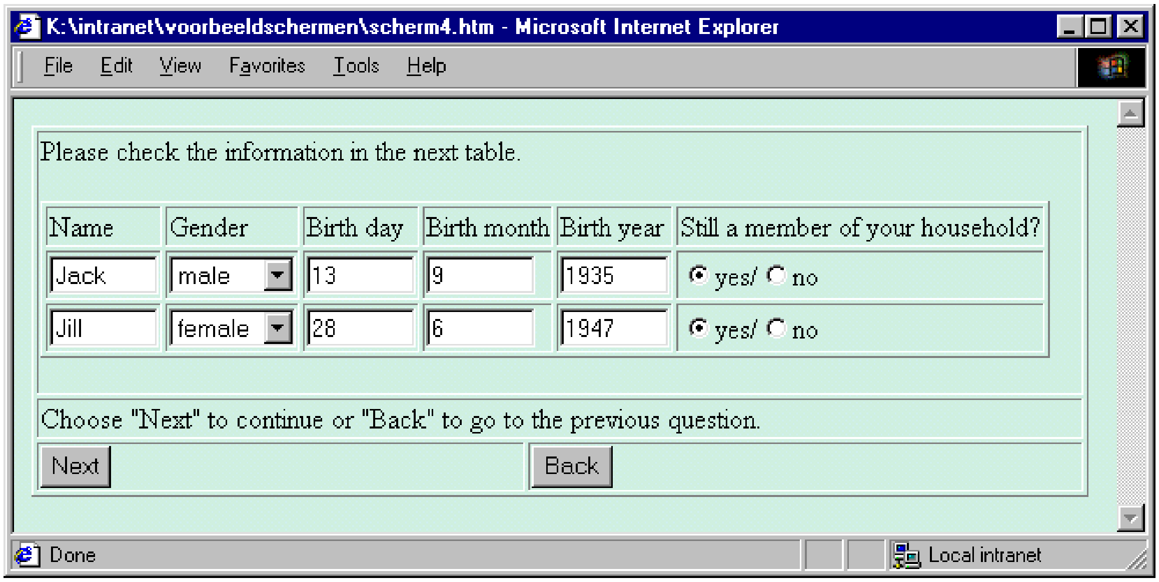
	completely agree	agree	neutral	disagree	completely agree
business-wise	○	○	○	○	○
accessible	○	○	○	○	○
modern	○	○	○	○	○
professional	○	○	○	○	○
modest	○	○	○	○	○

Choose "Next" to continue or "Back" to go to the previous question.

Screen 3: The use of C2BList () in the question text

The procedure Str2Arr in Blaise breaks the string into an array of integers. With this array, the original fields are filled. The procedure takes the original question text (the answer to Question 4), a number of expected answers (5), a TempArray which stores the array of integers and a right field which is filled with 1 if there are 5 answers given.

The use of C2BList is very advanced. Multiple answer types are supported, next to preload and advanced checking of the answers.


K:\intranet\voorbeeldschermen\scherm4.htm - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Next
Back
Done
Local intranet

Please check the information in the next table.

Name	Gender	Birth day	Birth month	Birth year	Still a member of your household?
Jack	male	13	9	1935	● yes/ ○ no
Jill	female	28	6	1947	● yes/ ○ no

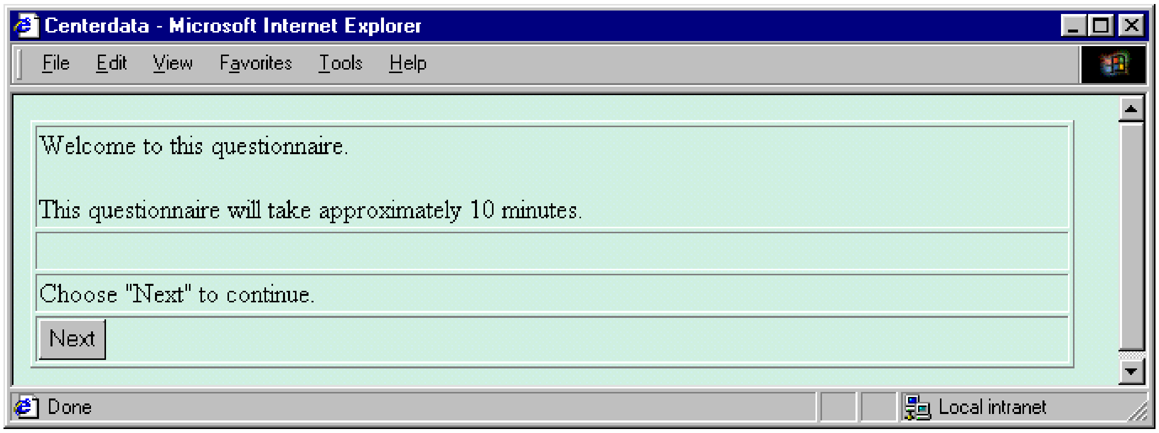
Choose "Next" to continue or "Back" to go to the previous question.

Screen 4: The use of C2BList() in the question text

C2BEmpty()

Sometimes a question is not really a question, but just used for information for the respondent. In that case it is possible to use C2BEmpty() in the Blaise question text. The normal input box is not shown.

Question2 "Welcome to this questionnaire. @/@/ This questionnaire will take approximately 10 minutes. C2BEmpty() ": String[1], empty


Centerdata - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Next
Done
Local intranet

Welcome to this questionnaire.

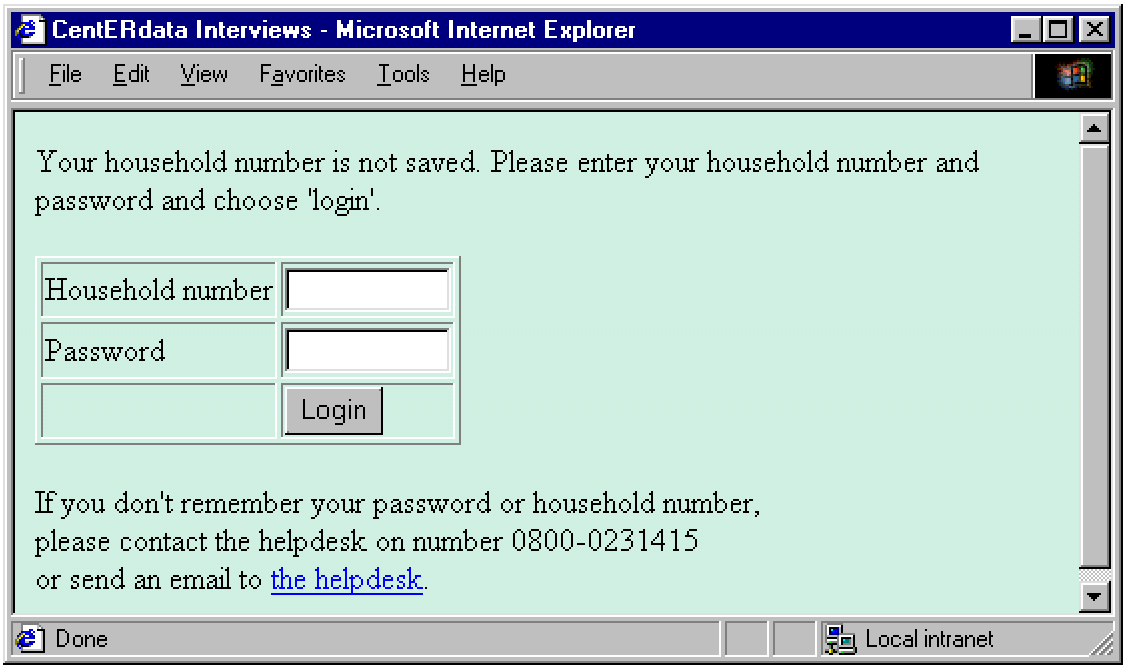
This questionnaire will take approximately 10 minutes.

Choose "Next" to continue.

Screen 5: The use of C2BEmpty() in the question text

5. The interview registration system

Next to C2B, CentERdata has built an interview registration system. This system, built in PHP and MySQL, handles the administrative part of the interviews, the questionnaires to be presented to the respondents in the current week, the attempts of the respondents and the allowance for the telephone costs. When the user first logs on to the system, the next screen will be shown:


CentERdata Interviews - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Login
Done
Local intranet

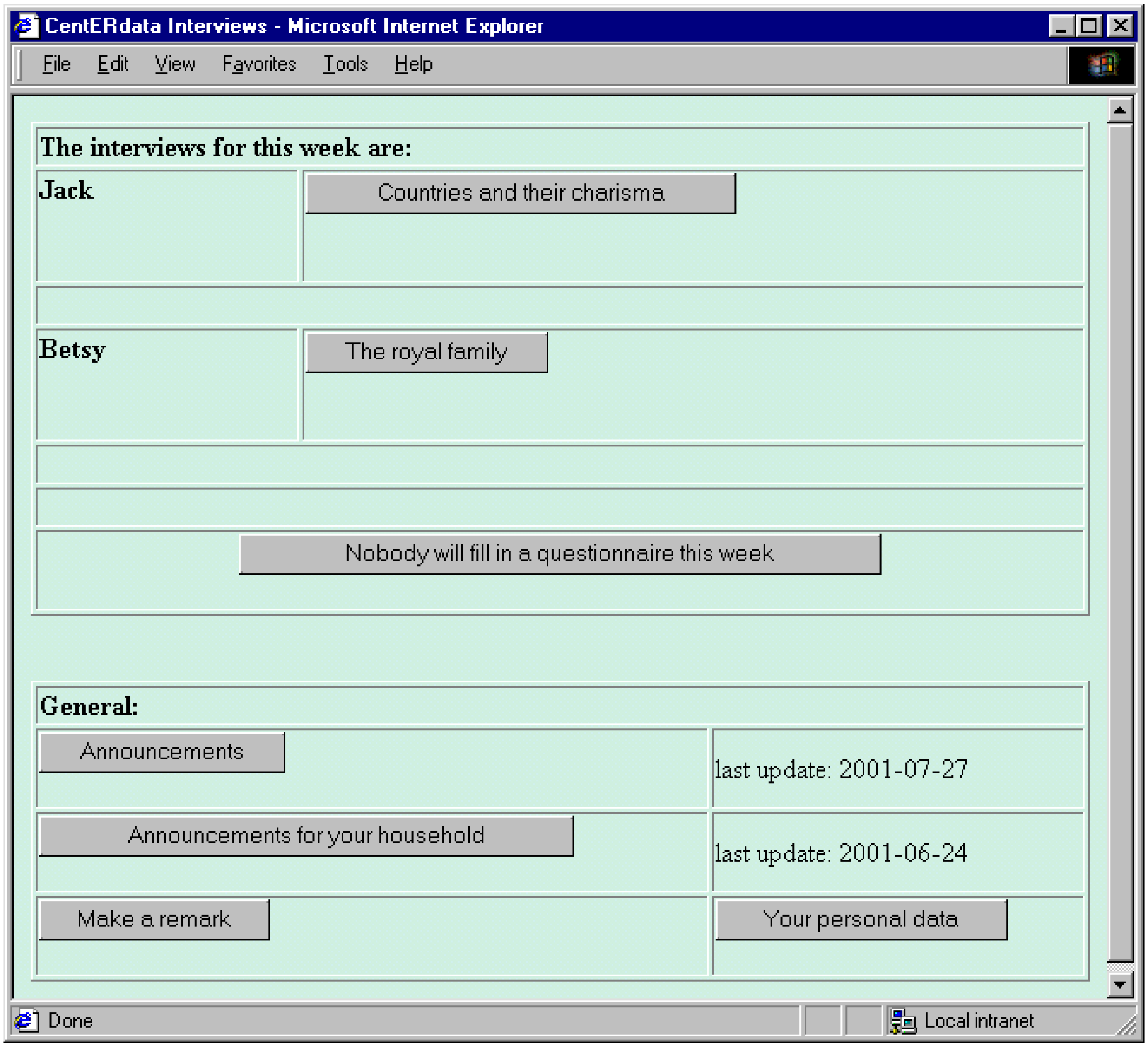
Your household number is not saved. Please enter your household number and password and choose 'login'.

Household number	
Password	

If you don't remember your password or household number, please contact the helpdesk on number 0800-0231415 or send an email to the helpdesk.

Screen 6: Logging on to the CentERdata interview system

Once the user enters the household number and password (which are sent by regular mail), the login screen disappears and the interview start page is shown. The next time the household logs on, the interview start page will be shown directly, because the household number is stored on the local machine. At the interview start page, the interview start page is shown. Here all the members of the household are listed, together with buttons for the interviews for the week. It is also possible to make remarks and view announcements. The personal data (the allowance) is behind the ‘Your personal data’ button.


CentERdata Interviews - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Countries and their charisma
The royal family
Nobody will fill in a questionnaire this week
Announcements
Announcements for your household
Make a remark
Your personal data
Done
Local intranet

The interviews for this week are:

Jack

Betsy

General:

last update: 2001-07-27

last update: 2001-06-24

Screen 7: The start screen of the CentERdata system

It is clear that Blaise for Internet is not capable of handling administrative data like this. After clicking on an interview button, a screen is shown with additional information about this interview and a possibility to go back to the interview screen (in case a respondent accidentally presses the wrong interview button). After this screen, a PHP script (interview.php) is called which handles the communication with the client browser and C2B. This script is necessary to ensure browser independency. The post information is transformed into a standard input line which is sent to C2B. In this way, C2B does not have to worry about different browsers.

In C2B a list is created with all the current users. A unique ID is created with the household and member number. Every unique ID has its own object, which handles the communication with Blaise for Internet. If the ID is not yet in the list, it is created and the start interview command is sent in HTML form to Blaise. The response of Blaise for Internet is analyzed and if it contains a *nohouse* or *nomem*, the appropriate household number and member number are sent back to Blaise for Internet (in HTML format). After this procedure, the actual interview is started. The respondent never sees that the first two questions are answered and gets the first question on the screen the moment the start interview button is clicked.

For every question, the respondent has 20 minutes to answer. If this time has passed, the quit interview command is sent to Blaise in HTML format. Internally the Blaise timeout is set to 30 minutes, so C2B is always earlier than Blaise for Internet with the timeout command. If the respondent wants to continue the interview, he has to start all over again, because Blaise for Internet only supports the New command.

6. Problems using Blaise for Internet

CentERdata started using the new system in March 2000. After a few weeks it became clear that there were a few problems concerning Blaise for Internet.

Problems regarding the content

Blaise for Internet does not support the ‘get’ method. Every time a questionnaire is started, a new record in the database is created. If a respondent wants to stop the questionnaire and continue at a later time, a new record is created. The questionnaire starts again from the beginning. This problem caused a lot of inconvenience for the respondents. A lot of questionnaires had to start all over again because the server crashed several times during the weekend, because of Blaise for Internet. This problem was presented at Statistics Netherlands. A new version of Blaise for Internet was developed. This version supports the ‘get’ method, but this version turned out to be so unstable that CentERdata never used or tested it.

Another problem is the table support. Blaise for Internet does not support tables. CentERdata had to develop its own table support by creating a parser in between Blaise for Internet and the Client browser. The development of this system and the training of the Blaise programmers was very time-consuming.

Technical problems

The first version of the Blaise for Internet system was not capable of handling more than 10 respondents. After the 10th respondent, the system started to assign wrong unique numbers. The result was that respondents started to fill in questions for other respondents and were getting the information meant for other respondents. Statistics Netherlands quickly acknowledged and solved this problem.

The Blaise for Internet system was delivered as an add-on to the Microsoft Information Server. The whole system turned out to be very unstable. The only solution to a hang up, was a hard reboot of the server. In some weekends, the server was (automatically) rebooted over 20 times. In the beginning the workload of the system was approximately 45 respondents filling in questionnaires at the same time. The maximum number of interviews filled in during the weekend was 4000. In a few months, CentERdata started to use the system for the whole panel. The upper limit of 50 respondents working on the same Blaise for Internet was quickly reached. CentERdata installed a second server. C2B redirected the respondents to the second server when the limit of 50 was reached. Later on, Statistics Netherlands changed the upper limit of the concatenating users.

In spite of the full cooperation of Statistics Netherlands, it soon became clear that this system was not working for CentERdata. A lot of respondents decided to quit their association with the panel due to the technical problems. CentERdata decided to build its own interview system based on the Blaise API calls.

7. The new CentERdata 2 Browser service

After all the effort invested in the new system, it was decided to use the old C2B service as a base for the new system. Instead of the communication with Blaise through HTML code, it would be much easier if it were possible to communicate directly with the Blaise database and metadata. Fortunately, the beta version of the Blaise API calls was available for testing. The API calls made it possible to connect directly to the Blaise database. After all, CentERdata only extracted the question text and answer possibilities from the HTML code Blaise for Internet generated. With the API calls it is possible to extract the question text and answer possibilities from a Blaise database.

8. The layout of the new CentERdata 2 Browser service

The new CentERdata 2 Browser system now also decides which question is shown on the screen. An overview is shown below.

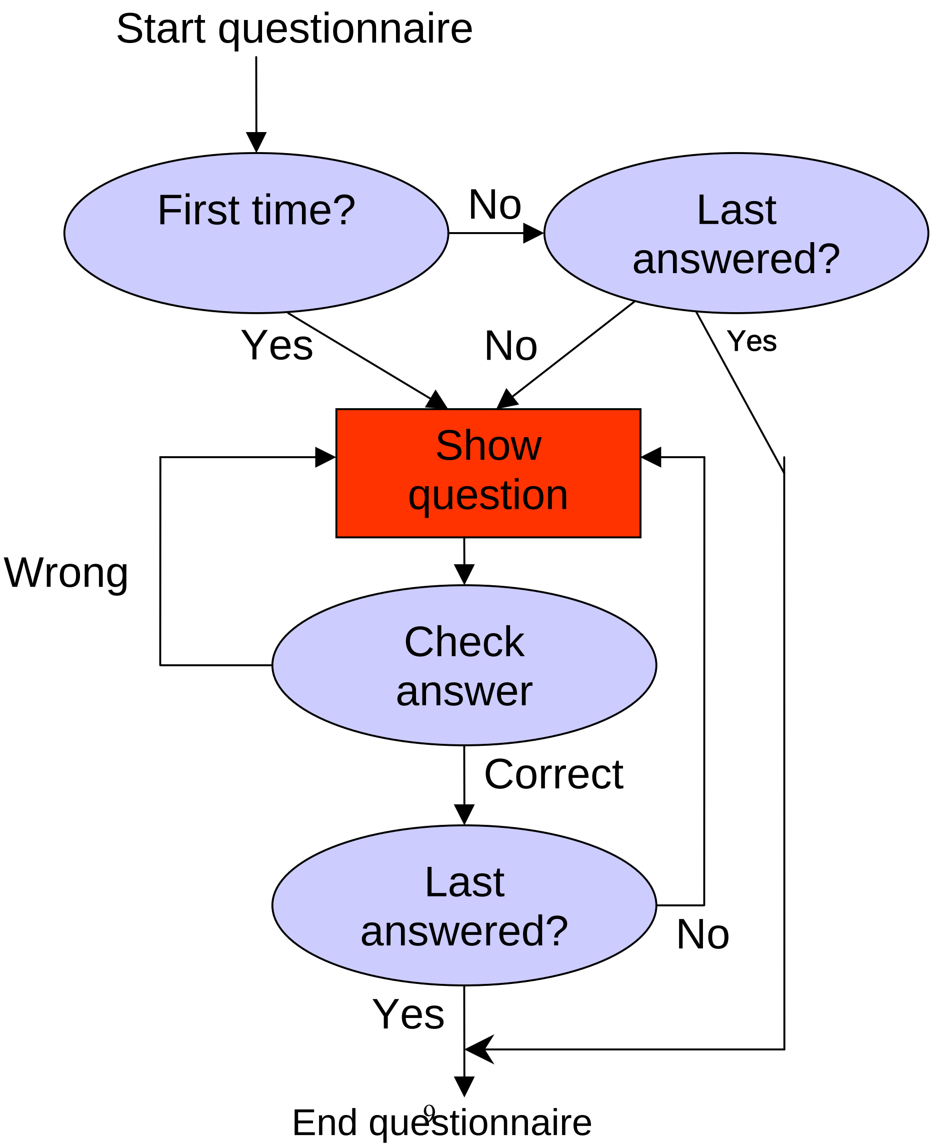


Figure 2: An overview of the Show question decision

If a respondent starts an interview, a new entry in the respondent list is made. If the respondent is already in the list, he continues with the last question. Every respondent stays in the active list for about 30 minutes. After 30 minutes of inactivity, the respondent is deleted, and the next time, a new entry should be made in the respondent list.

After the registration in the respondent list, a 'get' command is sent through the Blaise API calls to the database based on the household and member number. If this command fails, a 'new' command is sent, creating a new record for this respondent. In case of a 'new' command, the first question is shown on the screen (together with some additional information like 'You should enter a number between 0..100). If the 'get' method is successful, the 'getlastquestion' command is sent, and the first question the respondent has to answer is presented. The respondent is now capable of continuing a questionnaire.

Once a respondent has answered a question, a 'checkanswer' command is sent through the API calls. If 'checkanswer' fails, the question is resent to the client browser, together with an error message ('You should enter a number between 0 and 120.'). If the 'checkanswer' command accepts the answer, the answer is stored and the next question is shown on the screen.

This new system supports all the Blaise for Internet features. The programmers at CentERdata did not need to change the way they were programming. The new C2Bservice based on the Blaise API still used all the C2B commands. Tables are shown question by question.

9. C2B with API calls in practice

Since the complete transition to the new C2B system in January 2001, CentERdata has few to no technical problems. The number of times the server crashed is reduced to almost zero in a normal interview weekend from Friday till Wednesday. The possibility for respondents to continue their questionnaire has proved to be very powerful. The CentERpanel has grown to around 3000 households. The maximum workload so far is around 110 panel member filling in a questionnaire at the same time. The maximum number of questionnaires handled in one weekend is 10,000.

10. Future developments and hosting of Blaise questionnaires

In the near future CentERdata will try to be fully Blaise compatible. We are now working on the implementation of the tables.

From January 2001 CentERdata started to host Blaise questionnaires on its server. The results so far are very promising. Also a few other institutes are testing C2B on their own server. The C2B service is very simple. There is no intelligence in the client browser. The error checking is completely done on the server side. There is no need to reinvent the Blaise system in Java or XML. C2B service is browser-independent. The layout part of C2B is placed in a separate database. It is possible to generate a different layout for every questionnaire or even every question in an interview.

11. Conclusion

Although CentERdata faced a lot of problems during the conversion of its old interview system to the new Blaise-based system, the Blaise system behind C2B proved to be a major success. The incapability of Statistics Netherlands to solve a number of essential bugs in the Blaise for Internet package was not caused by incompetence or a lack of interest. CentERdata was just too far to the edge. The Blaise component pack is a huge step forward for everybody who wants to control Blaise and does not want to be dependent on the Blaise development team.