

Methodological guidelines for Blaise web surveys

Jelke Bethlehem, Statistics Netherlands & Adriaan Hoogendoorn, Free University, Amsterdam

1. Introduction

Carrying out a survey is a complex, costly and time-consuming process. Traditionally, surveys were carried out using paper forms (PAPI). One of the problems is that data collected in this way usually contain many errors. Therefore, extensive data editing is required to obtain data of acceptable quality. This consumes a substantial part of the total survey budget. Rapid developments of information technology in the last decades of the previous century made it possible to use microcomputers for computer-assisted interviewing (CAI). A computer program containing the questions to be asked replaced the paper questionnaire. The computer took control of the interviewing process, and it also checked answers to questions on the spot. Application of computer-assisted data collection had three major advantages: (1) It simplified the work of interviewers, because they did not have to pay attention any more to choosing the correct route through the questionnaire, (2) It improved the quality of the collected data, because answers could be checked and corrected during the interview, and (3) it considerably reduced time needed to process the survey data, and thus improved the timeliness of the survey results.

The rapid development of the Internet in the last decade had lead to a new type of computer-assisted interviewing: *Computer Assisted Web Interviewing (CAWI)*. The questionnaire is designed as a website on the Internet. By visiting the website, respondents can answer the questions. This type of survey is usually called a *web survey*.

It is not surprising that survey organisations use, or consider using, web surveys. At first sight, web surveys seem to have some attractive advantages:

- Now that so many people are connected to the Internet, a web survey is a simple means to get simple access to a large group of potential respondents;
- Questionnaires can be distributed at very low costs. No interviewer are needed, and there are no mailing costs;
- Surveys can be launched very quickly. No time is lost between the moment the questionnaire is ready and the start of the fieldwork;
- Web surveys offer new, attractive possibilities, such as the use of multimedia (sound, pictures and movies);

Thus, web surveys are a fast and cheap means of collecting large amounts of data. According to Couper (2000) this has opened the possibility for many organisations, other than the traditional survey organisations, to conduct surveys. Many of these organisations are not aware of potential methodological dangers of web surveys. Therefore there are many good many bad surveys on the Internet. And for the respondents it is not always easy to distinguish between both types of surveys. Do they decide to participate because of societal relevance, or are they misled by the entertainment value?

An abundance of good and bad surveys on the Internet creates a serious risk of lower response rates. Potential respondents are overcrowded, and pull out. The effect is similar to that which can be observed for CATI surveys, where the survey climate is spoiled by telemarketing activities.

This paper attempts to present a number of practical guidelines for the design of web surveys. These guidelines focus on methodological aspects. Section 2 presents a general taxonomy for survey errors, and discusses which errors are relevant for web survey design. Section 3 discusses a set of guidelines that aim at reducing survey errors as described in section 2. These guidelines are, for a large part, based on a literature review. Section 4 applies the guidelines to Blaise. It is examined to what extent they can be implemented in a Blaise web survey. A proposal is made for a default questionnaire layout that as much as possible follows these guidelines.

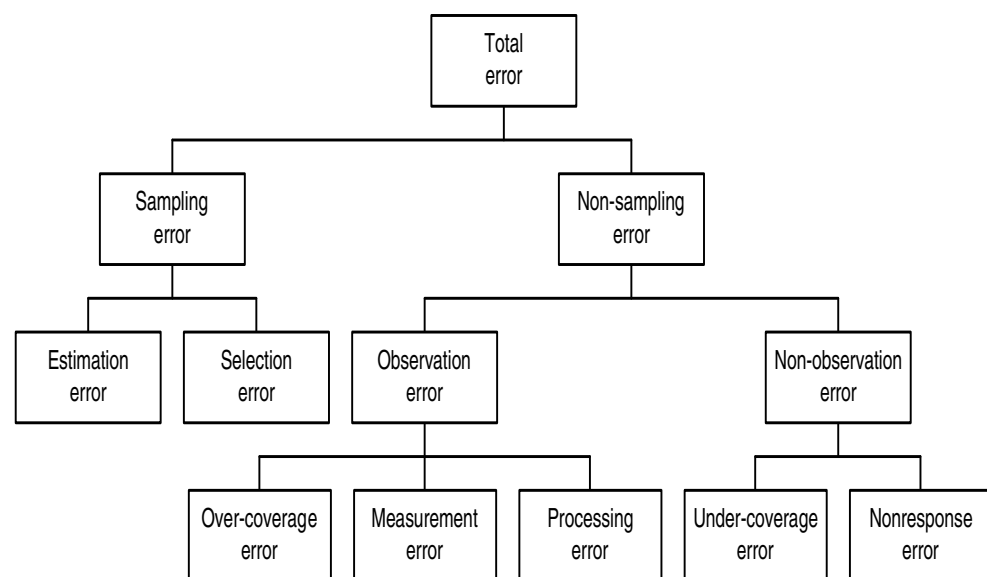
2. Survey errors

2.1. A taxonomy of errors

Usually, one of the main objectives of a sample survey is to compute estimates of population characteristics. Such estimates will never be exactly equal to the population characteristics. There will always be some error. This error can have many causes. Bethlehem (1999) gives a taxonomy of possible causes. It is reproduced in figure 2.1. The taxonomy is a more extended version of one given by Kish (1967).

The ultimate result of all these errors is a discrepancy between the survey estimate and the population characteristic to be estimated. This discrepancy is called the *total error*. Two broad categories can be distinguished contributing to this total error: sampling errors and non-sampling errors.

Figure 2.1. A taxonomy of survey errors



Sampling errors are introduced by the sampling design. They are due to the fact that estimates are based on a sample and not on a complete enumeration of the population. Sampling errors vanish if the complete population is observed. Since only a sample is available, and not the complete data set, to compute population

characteristics, one has to rely on an estimate. The sampling error can be split in a selection error and an estimation error.

The *estimation error* denotes the effect caused by using a sample based on a random selection procedure. Every new selection of a sample will result in different elements, and thus in a different value of the estimator. The estimation error can be controlled through the sampling design. For example, by increasing the sample size, or by taking selection probabilities proportional to some well-chosen auxiliary variable, you can reduce the error in the estimate.

A *selection error* occurs when wrong selection probabilities are used. For example, the true selection probabilities may differ from the anticipated selection probabilities when elements have multiple occurrences in the sampling frame. Selection errors are hard to avoid without thorough investigation of the sampling frame.

Non-sampling errors may even occur if the whole population is investigated. They denote errors made during the process of recording the answers to the questions. Non-sampling errors can be divided in observation errors and non-observation errors.

Observation errors are one form of non-sampling errors. These denote errors made during the process of obtaining and recording the answers. An *over-coverage error* means that elements are included in the survey not belonging to the target population. A *measurement error* occurs when the respondent does not understand the question, or does not want to give the true answer, or if the interviewer makes an error in recording the answer. Also, interview effects, question wording effects, and memory effects belong to this group of errors. A measurement error causes a difference between the true value and the value processed in the survey. A *processing error* denotes an error made during data processing, e.g. data entry.

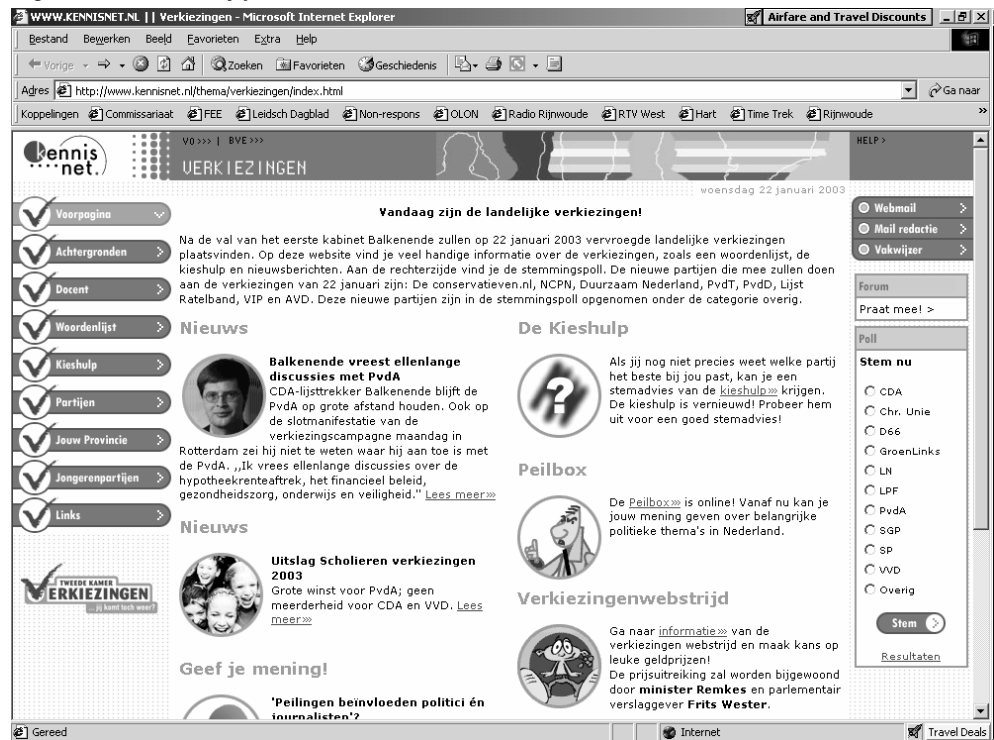
Non-observation errors are errors made because the intended measurements could not be carried out. *Under-coverage* occurs when elements in the target population do not appear in the sampling frame. So, representatives of these elements can never be contacted. Another non-observation error is *nonresponse*. It is the phenomenon that people selected in the sample do not provide the required information.

The taxonomy above makes clear that a lot can go wrong during the process of collecting survey data, and usually it does. Some errors can be avoided by taking preventive measures at the design stage. However, some errors will remain. Therefore, it is important to check the collected data for errors, and where possible, to correct detected errors. This activity is called *data editing*. Data editing procedures are not able to handle every type of survey error. They are most suitable for detecting and correcting measurement errors, processing errors, and possibly over-coverage. Phenomena like selection errors, under-coverage, and nonresponse require a different approach. This approach often leads to the use of adjustment weights in estimation procedures, and not to correction of individual values in records.

2.2. Probability sampling

In their fundamental paper, Horvitz and Thompson (1952) show that only unbiased estimates of population characteristics can be computed if every element in the population has a non-zero probability of selection, and that the researcher knows these probabilities. Furthermore, only under these conditions, the accuracy of estimates can be computed.

Figure 2.2.1. A survey just for entertainment



Many surveys on the web are not based on probability sampling. Couper (2000) distinguished three groups of such surveys:

The first group consists of surveys that are primarily intended for entertainment purposes. They often take the form of simple polls. There is often no control over who responds, and even how many times one responds. A typical example is the survey on Kennisnet (Knowledge net). This is a Dutch website for all involved in education. More than 11,000 schools and other educational institutes use this website. Figure 2.2.1 shows the page about the general elections of 22 January 2003. The right-hand part of the screen contains a small survey where visitors can vote.

Table 2.2.1 contains both the results of this poll on the morning of the Election Day, and the official results of the election.

Table 2.2.1. The results of an 'entertainment' survey

Party	Survey result	Election result	Difference
CDA (Christian democrats)	19.2 %	28.6 %	- 9.4 %
LPF (populist party)	12.0 %	5.7 %	+ 6.3 %
VVD (liberals)	16.1%	17.9 %	- 1.8 %
PvdA (social democrats)	8.8 %	27.3 %	- 18.5 %
SP (socialists)	14.4 %	6.3 %	+ 8.1 %
GL (green party)	17.0 %	5.1 %	+ 11.9 %
D66 (liberal democrats)	3.1 %	4.1 %	- 1.0 %
Other parties	9.4 %	5.4 %	+ 4.0 %

The survey figures are based on 17,574 people. This is not a small sample. Nevertheless, it is clear that the survey results are no way near the true election results.

The second group of non-probability sample surveys is self-selected web surveys. This approach uses open invitations on portals, frequently visited web sites, or dedicated survey sites.

An example of this approach is the election site of the Dutch Television channel RTL 4, see figure 2.2. It resembles to some extent the ‘entertainment’ web survey, but has a slightly more serious appearance. Again, the survey researcher has no control at all over who is voting. There is some protection, by means of cookie, against voting more than once. However, this also has the drawback, that only one member of the family can participate.

The conclusion from this analysis can be that probability sampling is an important prerequisite for making reliable inference on a target population.

Figure 2.2. A self-selection survey

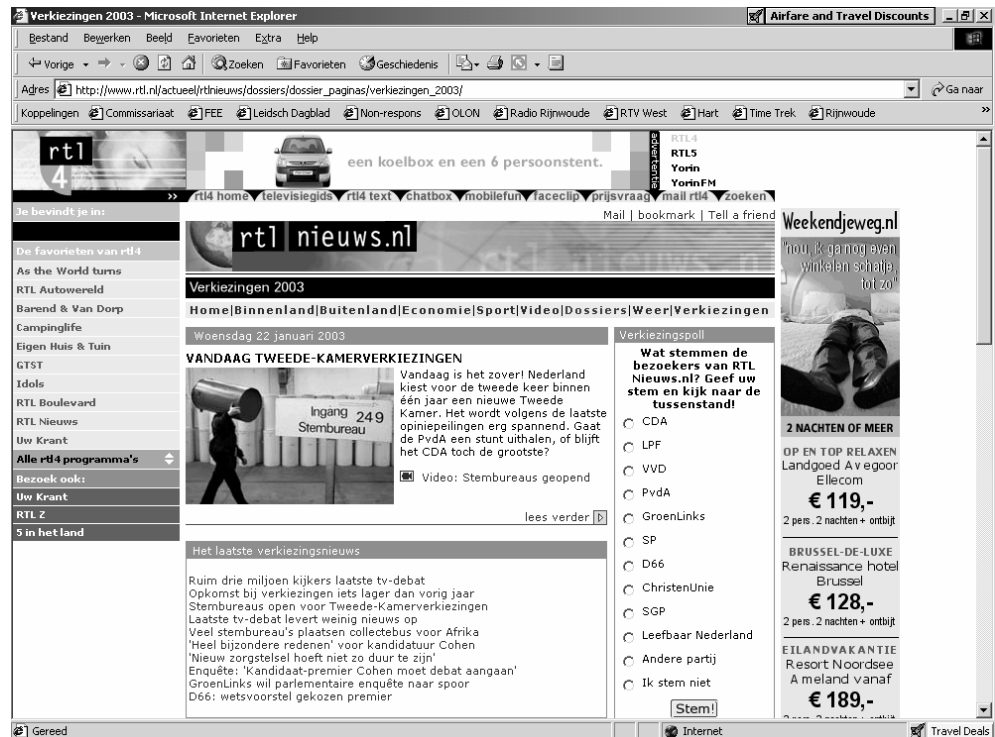


Table 2.2 shows the survey results at noon on the day of the general elections. Figures are based on slightly over 10,000 votes. Deviations between estimates and true figures are large, particularly for the large parties. Note that even the large sample size of over 10,000 people does not help to get accurate estimates.

Table 2.2.2. The results of a self-selection survey (percentages)

Party	Survey result	Election result	Difference
CDA (Christian democrats)	16 %	28.6 %	+ 12.6 %
LPF (populist party)	8 %	5.7 %	-2.3 %
VVD (liberals)	25 %	17.9 %	+ 7.1 %
PvdA (social democrats)	27 %	27.3 %	+ 0.3 %
SP (socialists)	7 %	6.3 %	- 0.7 %
GL (green party)	6 %	5.1 %	- 0.9 %
D66 (liberal democrats)	5 %	4.1 %	-0.9 %
Other parties	6 %	5.4 %	-0.6 %

The third group of non-probability sample surveys distinguished by Cooper (2000) consists of volunteer panels of Internet users. This approach creates a volunteer panel by wide appeals on well-visited sites and Internet portals. At time of registration basic demographic variables are asked. A large database of potential respondents is created in this way. For future surveys, samples are selected from this database. Only registered people can participate in these surveys.

Table 2.2.3. The results of a self-selection survey (seats in parliament)

Party	Survey result	Election result	Difference
CDA (Christian democrats)	42	44	- 2
LPF (populist party)	6	8	- 2
VVD (liberals)	28	28	
PvdA (social democrats)	45	42	+ 3
SP (socialists)	11	9	+ 2
GL (green party)	6	8	- 2
D66 (liberal democrats)	5	6	- 1
Other parties	7	5	+ 2

An illustrative example of this approach was the general election survey of the Dutch commercial television SBS6. People were invited to participate in the survey. Those visitors of the site accepting the invitation were asked a number of questions related to socio-demographic characteristics and voting behaviour in the previous election. From the set of people that was obtained in this way, samples of size 3000 were selected. Selection was carried out such that the sample was representative with respect to the social-demographic and voting characteristics. And these were asked for their voting behaviour in the coming election. Table 2.2.3 shows some results. The survey took place on the day before the general elections.

Table 2.2.4. The results of a probability sample survey (seats in parliament)

Party	Survey result	Election result	Difference
CDA (Christian democrats)	42	44	- 2
LPF (populist party)	7	8	- 1
VVD (liberals)	28	28	
PvdA (social democrats)	43	42	+ 1
SP (socialists)	9	9	
GL (green party)	8	8	
D66 (liberal democrats)	6	6	
Other parties	7	5	+ 2

Although attempts have been made to create a representative sample, the results differ still considerable from the final result. A much better would have been obtained with a true probability sample. Table 2.2.4 shows the results of a survey based on a true probability sample. The television channel Nederland 1 carried it out. A sample of size 1200 had been selected by means of random digit dialling. Compared to table 2.2.3 the results in table 2.2.4 are much better. The *Mean Absolute Deviation* (MAD) is table 2.2.3 is 1.75. In table 2.2.4 the values of this quantity has reduced to 0.75.

The conclusion from the analysis above can be that probability samples are a vital prerequisite for making proper inference about the target population of a survey. Probability sampling starts with the choice of a sampling design. This is the

definition of the selection mechanism applied. It should give a non-zero chance of selection to every element in the target population. A straightforward example of a sampling design is *Simple Random Sampling*, where each element has the same probability of selection.

Since elements are drawn at random in probability sampling, each new selection of a sample will result in a different set of elements. Therefore, the value of the estimator will also differ. This results in the *estimation error*. It only depends on the sampling design used, and not on the mode of data collection. So, the choice for a web survey has no impact on the selection error.

To actually select the sample according to the specifications of a sampling design, a *sampling frame* is required. This is a list, map, or other complete specification of all elements in the target population. The sampling frame must provide information on how to contact selected elements. Examples of sampling frames are a list of addresses, a list of phone numbers, a random digit dialling procedure, or a map containing the geographical positions of households.

If a proper sampling frame is used, the actual selection probabilities of the elements are equal to the planned selection probabilities as planned in the definition of the sampling design. However, there are situations in which is not the case. For example, when a household survey sample is selected from an address list, small business owners appear twice in the frame: once with their home address, and once with the business address. If no correction takes place for this phenomenon, these households will have a probability of being selected that is twice as big as planned. Using the wrong probabilities in the estimation procedure will result in biased estimates of population characteristics. This results in a *selection error*.

The situation is even worse if the Internet is used as a selection mechanism. This happens when surfers on the World Wide Web are invited to participate in a survey through a banner, popup window, or other announcement they happen to see on a website. The probability of being confronted with such an invitation is unknown, and the probability of accepting it is unknown. Therefore, it is impossible to compute unbiased estimates of population characteristics. The opinion poll of the Dutch television channel RTL 4 that was mentioned in section 2.2 is a good example of such a self-selection survey.

2.3. Coverage problems

The collection of all elements that can be contacted through the sampling frame, is called the *frame population*. Since the sample is selected from the frame population, conclusions drawn from the survey data will apply to the frame population, and not to the target population. Coverage problems can arise when the frame population differs from the target population.

Under-coverage occurs when elements in the target population do not appear in the frame population. These elements have zero probability of being selected in the sample. Under-coverage can be a serious problem for web surveys. If the target population consists of all people with an Internet connection, there is no problem. However, usually the target population is more general than that. Then, under-coverage occurs due to the fact many people do not have access to the Internet.

An analysis of data (source: www.cbs.nl) about the availability of an Internet connection at home indicates that this is unevenly distributed over the population in November 2001. Figure 2.3.1. shows the distribution by sex. Clearly, more males than females have access to the Internet.

Figure 2.3.1. Having an Internet connection at home by sex.

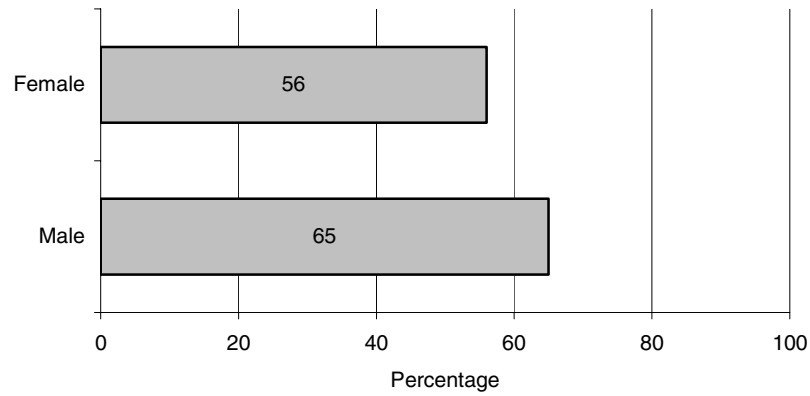


Figure 2.3.2 contains the percentage of people having an Internet connection by age group. The percentages of people having access to Internet at home decreases with age. Particularly, the people of age 55 and older will be very much under-represented when the Internet is used as a selection mechanism.

Figure 2.3.2. Having an Internet connection at home by age.

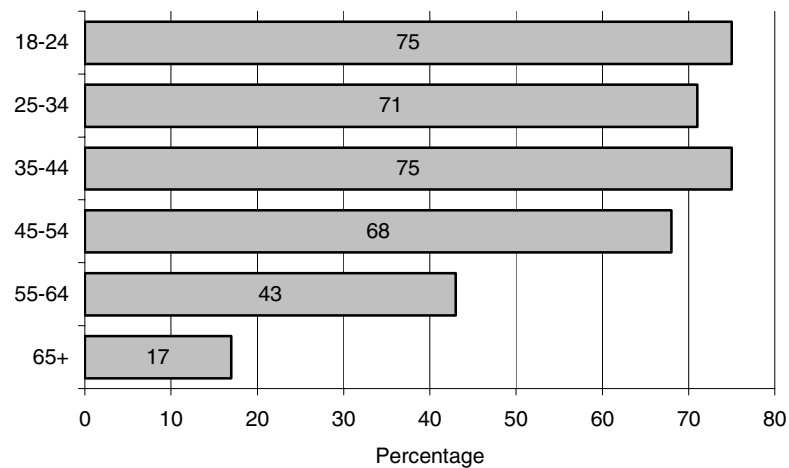
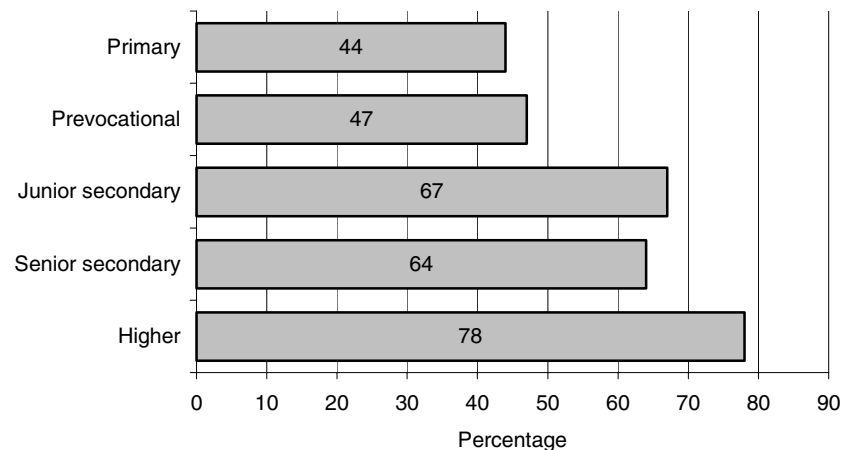


Figure 2.3.3 contains the percentage of people having an Internet connection by level of education.

Figure 2.3.2. Having an Internet connection at home by level of education.



It is clear that people with a higher level of education have more access to the Internet than people with a lower level of education.

The Social and Cultural Planning Agency (SCP) carried out a survey about the use of digital media among a sample of students in secondary education, see De Haan et al. (2002). It turned out that 97% of the students have a PC at home, and that 84% have an Internet connection. However, these percentages are much lower among students with a foreign background. 15 % of the students with a Moroccan or Turkish background have computer at home. And 12% of the students with a former Dutch Indies background have no computer at home.

The results described above are in line with the findings of authors in other countries. See e.g. Couper (2000), and Dillman and Bowker (2001). It is clear that when the Internet itself is used as a selection mechanism, certain specific groups will be under-represented.

2.4. Nonresponse problems

Nonresponse can be defined as the phenomenon that elements (persons, households, companies) in the selected sample do not provide the requested information, or that the provided information is useless. The situation in which all requested information on an element is missing is called *unit non-response*. If information is missing on some items only, it is called *item non-response*.

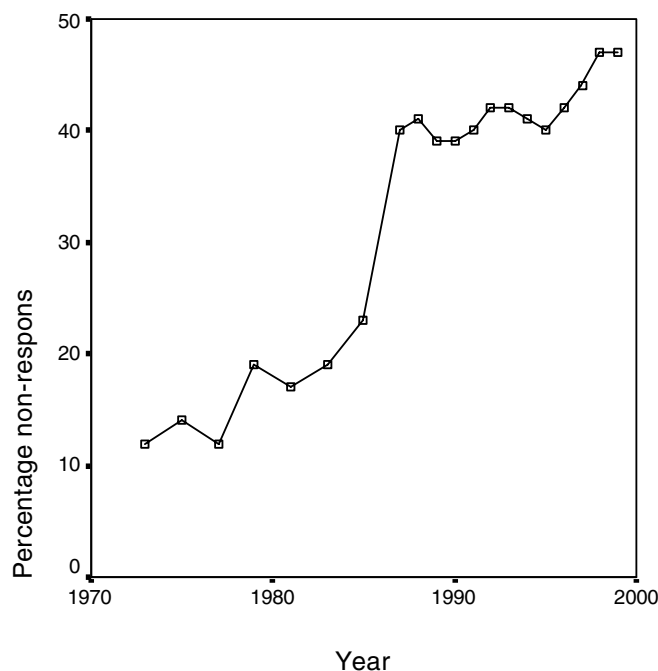
Due to nonresponse the sample size is smaller than expected. This leads to less accurate, but still valid, estimates of population characteristics. This is not a serious problem. It can be taken care of by taking the initial sample size larger. A far more serious problem caused by nonresponse is that estimates of population characteristics may be biased. This situation occurs if, due to non-response, some groups in the population are over- or under-represented, and these groups behave differently with respect to the characteristics to be investigated.

Nonresponse has a negative impact on the quality of population estimates. One of the most serious effects of nonresponse is that these estimates may be biased. This means, that in fact the wrong conclusions are drawn from the survey results. Nonresponse constitutes a serious threat for the quality of survey results, particularly when nonresponse rates are high.

Nonresponse rates are high in The Netherlands. As an example, figure shows the rate of nonresponse for the Dutch Labour Force Survey. The percentage of nonresponse has been rising ever since 1970, and now approaches the 50%.

The magnitude of the non-response is determined by a large number of factors, including the topic of the survey, the target population, the time period, the length of the questionnaire, etc. The mode of data collection is also an important factor. Generally, face-to-face surveys and telephone surveys have lower nonresponse rates than surveys in which respondent are confronted with self-administered questionnaires. The role of interviewers is crucial in this. They can play an important role in persuading people to participate, and in assisting respondents to answer questions. Such assistance is not available for self-administered questionnaires.

Figure 2.4.1. Nonresponse rates of the Dutch Labour Force Survey



A web survey questionnaire is a form of self-administered questionnaires. Therefore, web surveys have to potential of high nonresponse rates. An additional source of nonresponse problems is technical problems of respondents having to interact with the Internet, see e.g. Couper (2000), Dillman and Bowker (2001), Fricker and Schonlau (2002), and Heerwegh and Loosveldt (2002).

Slow modem speeds, unreliable connections, high connection costs, low-end browsers, and unclear navigation instructions may frustrate respondents. And this often results in respondents discontinuing the completion of the questionnaire. In order to keep the survey response up to an acceptable level, every measure must be taken to avoid these problems. This requires a careful design of web survey questionnaire instruments.

3. Design of web survey questionnaire instruments

3.1. Introduction

A proper design of a survey questionnaire instrument is crucial for the quality of the data that is collected. Layout of the questionnaire, question wording, type of expected answer, ease of navigation, speed of processing, and browser compliance are all factors that may have an impact on measurement errors and response rates. Therefore, careful attention must be paid to these aspects in the design stage of the web survey instrument.

This section addresses a number of these aspects. Based on a literature review a number of guidelines are formulated. It is shown what these guidelines would lead to in a practical example. Then it is explored to what extent such a design can be realised within the Blaise System. As an example we use a simple questionnaire that attempts to survey local radio listening behaviour. Appendix A contains a paper copy of the questionnaire form.

3.2. General design issues

Web questionnaires are created with HTML. The ongoing development of this language has provided designers with all kinds of possibilities to use colours, sound, graphics, animation, and embedded programs (in Flash, JavaScript, Java, PHP, VBScript, etc). This allows for creating very fancy questionnaires. However, there are also drawbacks. One is that it makes the questionnaire program much bigger. Downloading these programs from servers to respondents takes much more time. This may lead to dropout of respondents with slow modems, unreliable connections, and high connections costs. Moreover, not every browser in use may support all these fancy features.

Dillman et al. (1998) compared a fancy web questionnaire design with a plain one. The plain questionnaire gave better results. There was a higher response rate. And a smaller number of people dropped out of questionnaire somewhere halfway its completion. Also, it took respondents less time to complete the plain questionnaire. This all means that we should apply the KISS principle (Keep it Simple, Stupid) to web questionnaire design. The focus should be on the essential parts of the questionnaire. All kinds of *questionnaire junk* should be avoided. Here is an analogue to statistical graphics. For example, Tufte (1983) describes how too much *chart junk* makes interpretation of graphs much more difficult. He introduces the *data-ink-ratio* as a measure of the amount of chart junk in a graph. It is defined as the amount of ink used to draw the essential part of the graph divided by the total amount of ink used. For a good graph, the data ink ratio should be close to 1. Likewise, the data-ink-ratio could be applied to the design of a web questionnaire. Also here, it could be a good idea to strive for a data-ink-ratio of 1.

Application of the KISS principle, going for a data-ink-ratio of 1 means that colours, graphics, animation, sound, and embedded programs are only used where it is really functional and cannot be avoided. Particularly in mixed-mode surveys, where the web questionnaire is just one of the modes of data collection, it is important that questionnaire forms of the various modes resemble one another. Therefore, it would be a good idea to let the web questionnaire form resemble the paper questionnaire form as much as possible.

This all leads to a guideline to the guideline that web survey questionnaires should be designed plain and sober. They should look like their paper analogues. Unnecessary and distracting questionnaire junk like picture, sound and animation must be avoided.

The texts on the screen should be well readable. The choice of font is important. Generally, sans-serif fonts like Arial, Verdana and Helvetica are preferred. The font size should be large enough to keep text readable on screens of various resolutions. At least, they must be readable for the most common screen resolutions (800 × 600 and 1024 × 768).

The layout of the questionnaire screens must not depend on the screen resolution used. Dillman and Bowker (2001) describe web surveys where a different screen resolution results in different ways in which the information is presented on the screen. This may easily lead to measurement errors. One way to get a resolution independent design is to make use of the <table> tag in HTML. Simply put all questionnaire information in a fixed-width table. This trick has been applied in figure 3.2.1. The questionnaire form is designed as a white page on a grey background. The change in screen resolution from 800 × 600 to 1024 × 768 only causes the information to be displayed enlarged. It has no impact on the layout.

Figure 3.2.1. The same web questionnaire screen for different screen resolutions

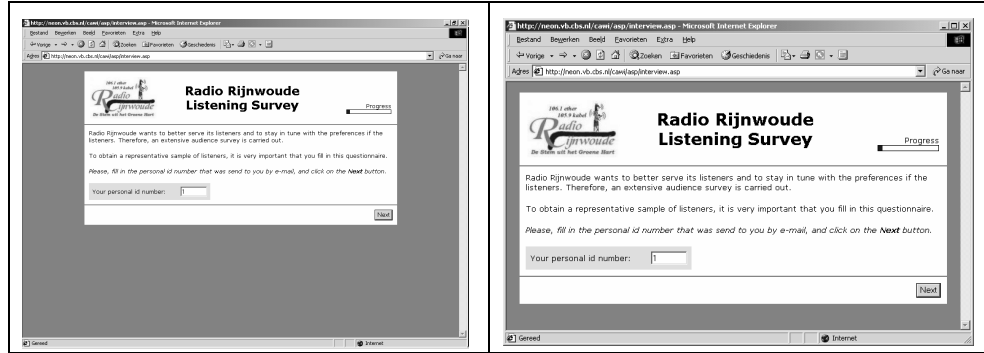


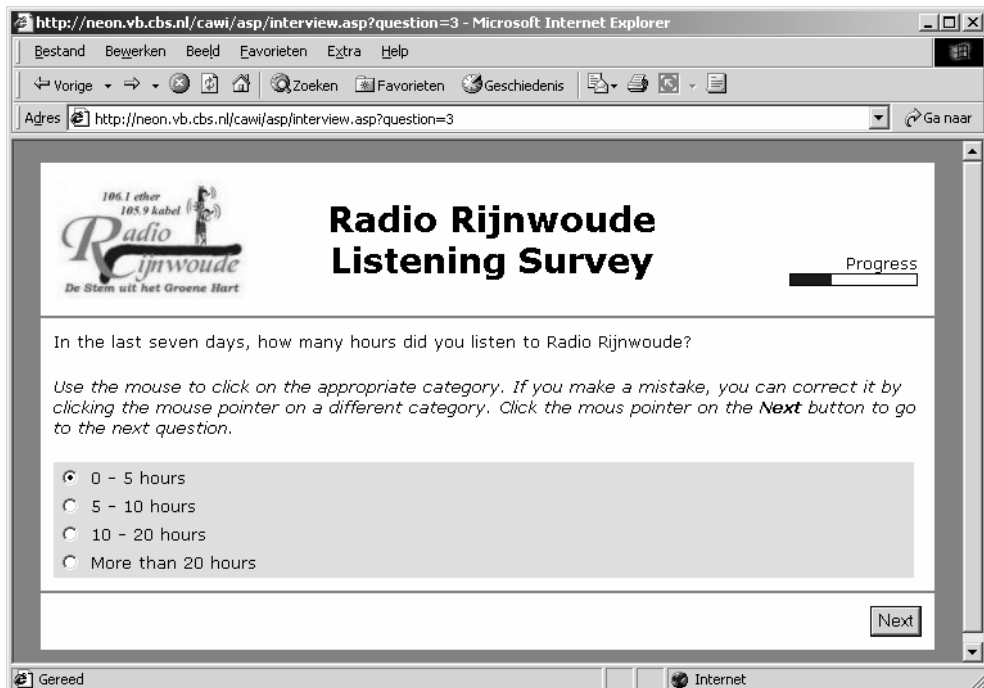
Figure 3.2.1 also shows another important guideline. That is that the use of scrollbars should be avoided as much as possible. Completion of the questionnaires should be as simple as possible. The actions required to answer a question, and to go to the next question, must be kept to a minimum. And it also must be clear what actions are required to realise this. For example, having to scroll down to find a **Next** button may lead to confusion, frustration, and even to dropping out of the session.

3.3. Design of item screens

The layout of the various screens of the web questionnaire must be clear, easy to understand, and easy to handle. The various items on the screen must be presented in a standard way. We advocate a structure that consists of three parts.

The top part contains a limited amount of general information. There can be a logo at the upper left indication the organisation that is carrying out the survey. There may be contact information (telephone number or e-mail address). The upper middle part contains the name of the survey.

Figure 3.3.1. A question screen of web questionnaire



The upper right is reserved for some kind of progress indicator. This progress indicator shows respondents where they are in the completion process. It can be a simple text like a percentage, or some kind of graphical display like a progress bar.

Couper et al. (2001) use a pie chart. They show that such a device reduces item nonresponse. Care must be taken that including fancy progress indicators do not increase downloading times, which can have a negative effect on response. Also no use must be made of embedded programs that are not supported by browsers, or require downloading and installing special viewers. In our example we use a simple progress bar, see e.g. figure 3.2.1 or 3.3.1.

The middle part of the screen contains the current question to be answered. Generally, the question information consists of three parts:

- The text of the question;
- Instructions how to answer the questions, and how to proceed to the next question;
- The answer to the question. The format of this part depends on the type of question (open, closed, check-all-that-apply, numeric, etc).

The *text of the question* is displayed in a normal font. To give the text of the question more emphasis, and to distinguish it from the rest of textual information on the screen, one may want to consider displaying the text in boldface, or the use a different colour. To emphasize certain parts (words) in the text one may consider using a different colour. However, one should always be aware of the fact that the colour-blind persons may not be able to distinguish different colours.

It must always be clear for respondents which actions they have to take in order to answer a question and to proceed to the next questions. Dillman et al (1998) describe several ways of doing this.

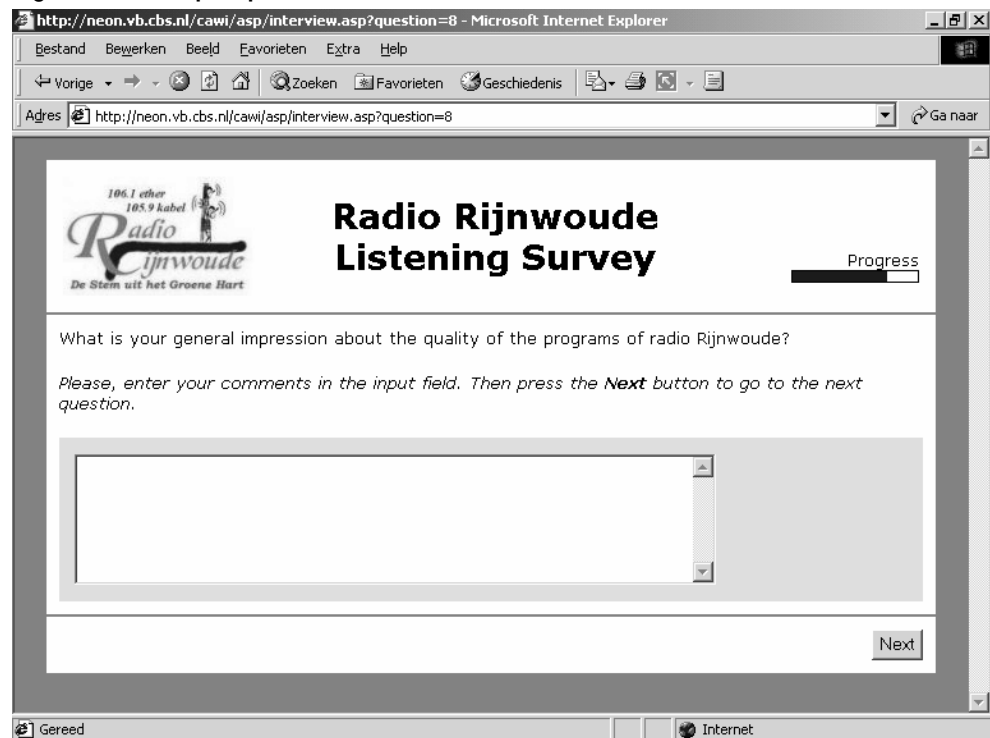
- Add instructions to every question on the screen. An example of this approach is given in figure 3.3.1. To distinguish instructions from the question text, the instructions are displayed in italics. This seems to be more or less a convention. However, one should realise that text in italics may not always be very readable on low-resolution screens. Another option could be to display the question text in boldface and the instruction in normal typeface.
- Only add instructions to the first few questions of the questionnaire. In any case, there must be instructions for any new type of question that appears for the first time. This will lead to less cluttered question screens. But it may lead to problems later in the questionnaire for those that have forgotten the instructions.
- Add an initial screen to the questionnaire where respondents are asked whether or not they are experienced computer users. For experienced users, this is followed by a questionnaire without instructions. And those who indicate they are not experienced will get a questionnaire with instructions.
- Add a window that ‘floats’ on top of the questionnaire screen, and that contains instructions for answering the question. This is a feasible option but may require (too much) additional processing time.
- Add an initial screen with general instructions on answering questions. The respondents will see this screen only once. Dillman et al. (1998) do not consider this desirable practice from the point of view of cognitive or learning practice.

The third part of the question consists of the area in which the respondents must give their answers. The structure and layout of this part depends on the type of question.

For an *open question*, there will be an area in which some text can be entered. See figure 3.3.2 for an example. It is not always clear for respondents what kind of input is expected from them: only a few words, a paragraph, or maybe even a small story. Sometimes, they only see a small input field, and they are not aware of the possibility that a lot of text can be entered by simply scrolling the text. It could help to give the input field a size (in terms of numbers of rows and columns) that more or less reflects the size of the expected answer.

In figure 3.3.2 some text is expected that may not exceed 300 characters. Since the field is 60 characters wide, the number of rows is made equal to 5.

Figure 3.3.2. An open question



The screenshot shows a Microsoft Internet Explorer browser window. The address bar displays the URL: <http://neon.vb.cbs.nl/cawi/asp/interview.asp?question=8>. The browser's menu bar includes 'Bestand', 'Bewerken', 'Beeld', 'Favorieten', 'Extra', and 'Help'. The toolbar contains icons for 'Vorige', 'Zoeken', 'Favorieten', 'Geschiedenis', and 'Ga naar'. The main content area features the logo for 'Radio Rijnwoude' with the tagline 'De Stem uit het Groene Hart' and the text '106.1 ether 105.9 kabel'. The survey title is 'Radio Rijnwoude Listening Survey'. A progress bar is visible in the top right corner. The question text reads: 'What is your general impression about the quality of the programs of radio Rijnwoude?' followed by instructions: 'Please, enter your comments in the input field. Then press the **Next** button to go to the next question.' Below the text is a large, empty text input field with a vertical scrollbar on the right side. A 'Next' button is located at the bottom right of the input area. The browser's status bar at the bottom shows 'Gereed' and 'Internet'.

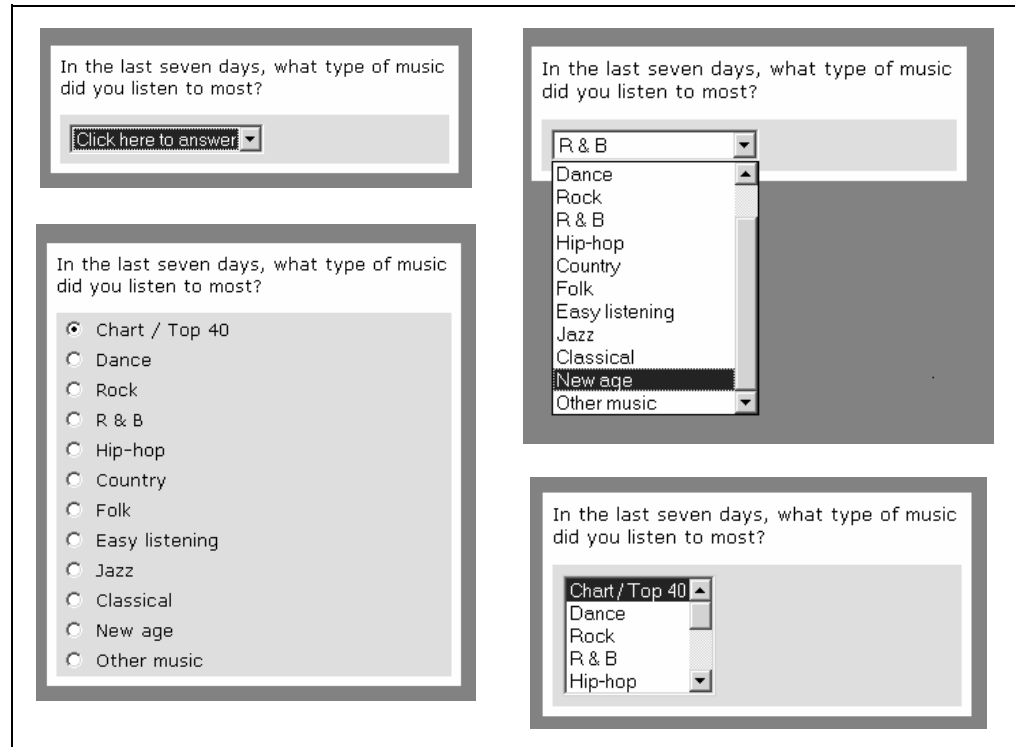
For a closed question, HTML offers various ways of displaying the possible answer categories, and selecting one of them. Examples are displayed in figure 3.3.3. The solution on the lower left uses radio buttons, and the other ones use drop-down boxes. The advantages and disadvantages of both approaches are discussed by Dillman (2000), Heerwegh and Loosveldt (2002), and Couper (1999).

The advantage of radio buttons is that a closed question looks very much like its paper questionnaire analogue. An answer is selected by clicking the appropriate radio button. Clicking another radio button can change an answer. But an answer can never be erased any more. When using radio buttons, all possible answers are visible. This helps the respondent to select the proper answer. A disadvantage of radio buttons is that they can take a lot of space when the list of possible answers is long.

Drop-down boxes usually show only one item (the first one, or the selected one). To select an answer, the respondent must click on the drop-down box. Then the list of possible answers becomes visible. If this list is very long, it only becomes partially visible. In this case, the respondent may have to use the scrollbar to display that part of the list containing the appropriate answer. This is illustrated in figure 3.3.3. The top left drop-down box initially only shows the first item of the list ('click here to answer'). After clicking the drop-down box, the list of possible answers unfolds. The result is display top right. A number of items becomes

visible, but not all of them. The bottom right show a drop-down box with a different initial state. Always five items are visible. The scrollbar must be used to show another part of the list.

Figure 3.3.3. Various ways of displaying a closed question

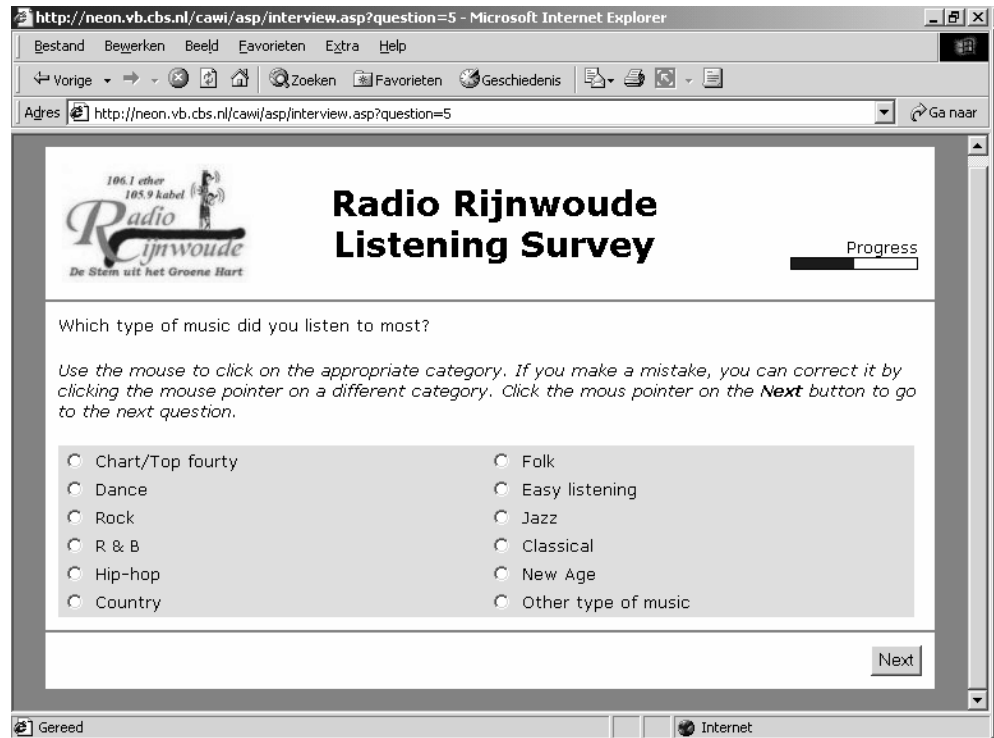


Drop-down boxes have the advantage over radio buttons that it takes less time to download them from the server. However, drop-down boxes also have severe disadvantages. One is that they are more difficult to handle by respondents. They have to perform three actions to select an answer: clicking the box, scrolling to the right answer, and clicking the right answer. Selecting the right answer from a set of radio buttons only requires one click.

Another serious disadvantage has been investigated by Couper (1999). He shows that respondents tend to select more the initially visible options than the non-visible ones. Also Heerwegh and Loosveldt (2002) express concerns about mode effects caused by list boxes. This all indicates that one should use radio buttons as much as possible.

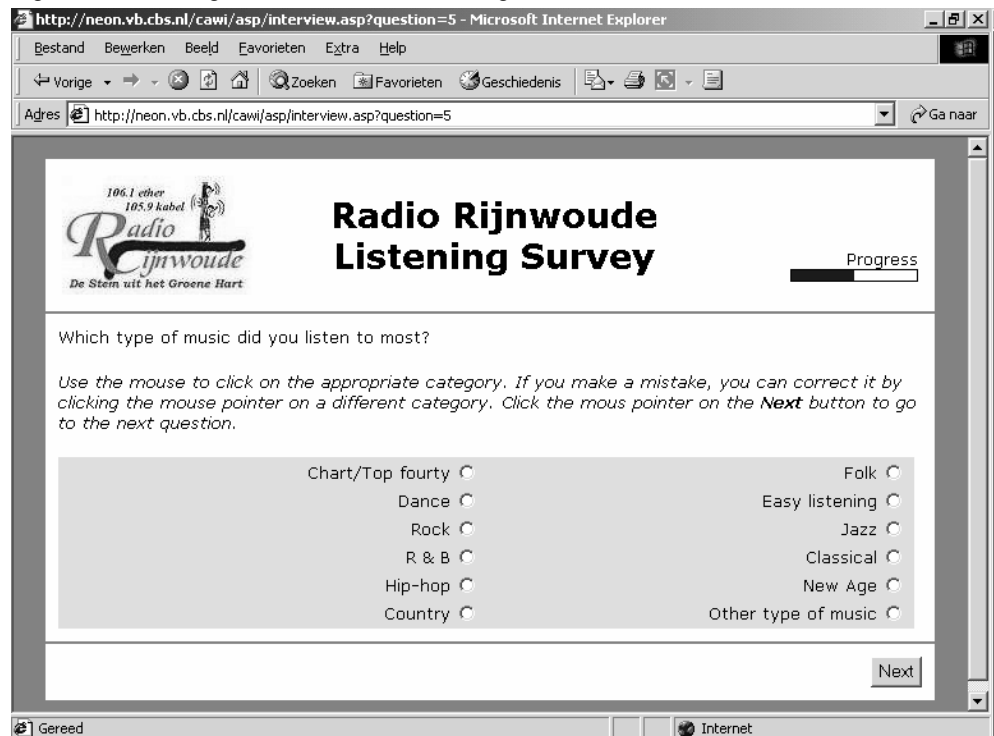
Attention should be paid to displaying radio buttons when the list of possible answers. As was mentioned earlier, all question information should be visible on the screen, and not require any scrolling. Therefore, it is better to split a long list of radio buttons over a number of columns. Figure 3.3.4 gives an example. Here the answers are distributed over two columns. To give a visual clue that these two lists belong together, Dillman et al. (1998) suggest to put them in a kind of box. This is realised in figure 3.3.4 by means of a grey background.

Figure 3.3.4. A closed question with many possible answers



It is very common for paper questionnaires to have the answer boxes to the left of the descriptions of the possible answers. This principle is also being implied in many web surveys, see e.g. figure 3.3.4.

Figure 3.3.5. Putting the answer boxes to the right



However, having to answer closed questions with a mouse in this layout requires substantial mouse movement. This not only takes time, but may also be a source of errors. Bowker and Dillman (2000) describe a small study investigating effects of placing answer boxes to the right of the text. The basic idea is that this layout would reduce the efforts required to answer the question, and to proceed to the next

question. Figure 3.3.5 shows an example of such an approach. Results of their study seem to indicate that changing the position of the answer boxes has no effect on item nonresponse rates and quality of the answers given. However, less-experienced computer users seem to be more confused about how to answer the questions. Although their study had some limitations, results show no clear advantages of putting answer boxes to the right.

A question type that is known to have measurement error problems, is the “check-all-that-apply” question. Respondents are asked to check all items that apply to them in a sometimes very long list. The problem is that often respondents stop checking items when they think they have checked enough answers. There is substantial evidence that respondents do not read all possible answers before going to the next question, see e.g. Dillman et al. (1998).

Figure 3.3.6 shows an example of a check-all-that-apply question. Particularly when the list of possible answers is long, it is easy to make mistakes, like checking the wrong category, or forgetting to check categories.

Figure 3.3.7 shows a different format for a check-all-that-apply question. Each check box is replaced by two radio buttons, one for “yes” and one for “no”. This approach forces respondents to perform an action for each category. It encourages them to go down the list item-by-item, and give an explicit answer for each item. This approach reduces measurement errors, but also has a drawback: it requires a lot more time to answer the question, and this may increase nonresponse rates.

Figure 3.3.6. A check-all-that-apply question that uses check-boxes

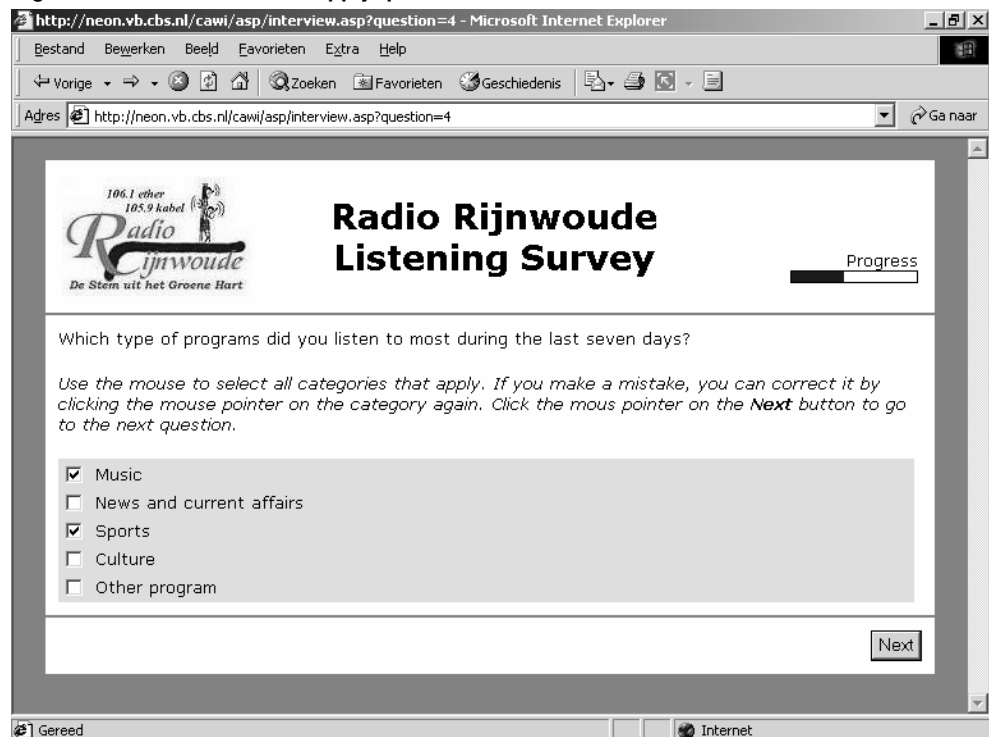
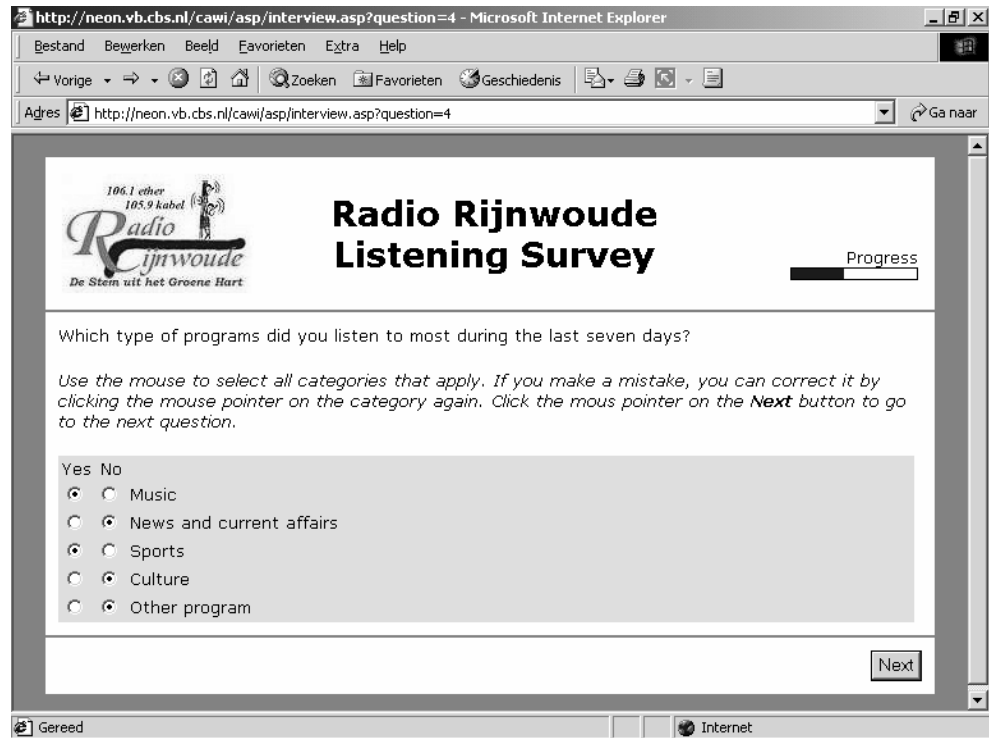


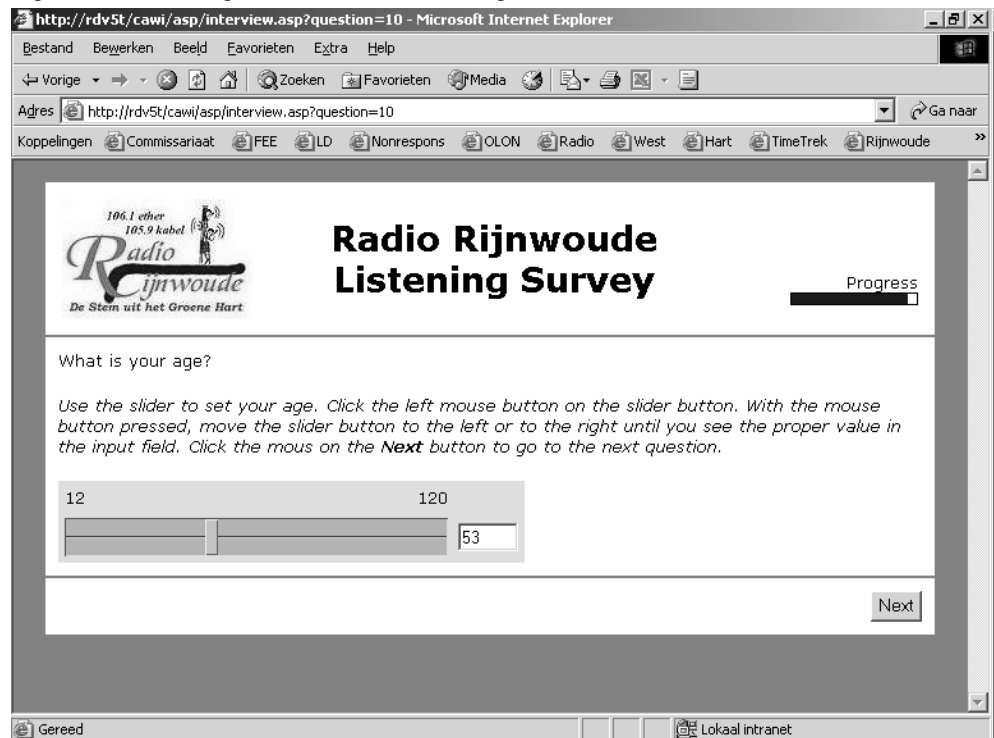
Figure 3.3.7. A check-all-that-apply question that uses radio buttons



A final question type that is briefly mentioned is the *numerical question*. The display format of such a question depends upon the range of valid answers. If the range is limited, the numerical question can be treated as a closed question. This is the easiest way to answer, and reduces measurement errors as much as possible. This approach can typically be applied for questions like the size of the household. If the value range is so large it cannot be handled by a set of radio buttons, an open question type input format has to be used. This means offering an input field in which the respondent must type the answer. Typing is a source of errors. It is easy to enter non-numerical information or values out of the defined range. This requires extensive error checking. Couper et al. (2001) showed that use of input fields causes higher item nonresponse rates.

One can think of different ways of letting respondents specify numerical answers within a specified range. One way is making use of so-called slide bars. Figure 3.3.8 contains an example of such an approach. The respondents specify their answer by dragging the slide bar button to the left or right. One advantage of this technique is that no typing errors can be made. Another advantage is that no out of range values can be entered. One disadvantage is that a starting value has already been specified. This makes it very easy for the respondent to click the Next button without paying attention to the question and its answer. Another disadvantage is that the slide bar is no standard feature in HTML. It has to be programmed, and the code has to be embedded in the HTML page. The slide bar in the example of figure 3.3.8 was programmed in JavaScript. Not every browser may support JavaScript, and it also increases transmission times.

Figure 3.3.8. Entering a numerical answer using a slide bar



In a mixed-mode survey, questionnaires for the various modes should resemble one another as much as possible. This helps to avoid mode effects. This would imply to implement numerical questions in the traditional way, using a simple input field.

The third, and final, part of the questionnaire screen is the bottom part. It contains the navigation buttons. Our example contains just one such button, and that is the Next button. If this button is clicked, the system first carries out various checks. Only if a question has been answered, and the answer is within the defined domain, the respondent is allowed to go to the next question. In case something is wrong, a clear error message will appear, explaining what is wrong, and how the error can be corrected.

3.4. Incorporating checks

Experience with various types of CAI surveys has shown that the quality of the collected information can be improved by carrying out *active error checking*. It means that all kinds of range checks and consistency checks are carried out the moment the relevant information has been entered by the respondent or interviewer. This feature should also be implemented in web surveys.

For closed questions, one radio button must have been checked. If not, it is not possible to proceed to the next question.

For numeric questions with an input field, it must be checked whether a numerical answer has been entered, and that this answer is within the valid range. For numeric questions with a slide bar or similar device it must be checked whether the bar was moved.

For check-all-that-apply questions with checkboxes, no checks can be carried. For check-all-that-apply questions with yes/no radio buttons, it must be checked that for each answer one of the two radio buttons has been checked.

For open questions, no checks are carried out. However, it is possible for specific open questions to require the respondent to enter at least something.

Experience with e.g. Blaise surveys has shown that data quality can be improved even more by carrying out consistency checks. Whenever relationships between survey variables impose restrictions on the values of answers given, this should be checked. If an inconsistency is detected, it should be reported on the screen. The respondent should be given the possibility to modify one or more of the answers of questions involved in the check. Usually, this means jumping back to the point in the questionnaire to a previous question. The interview software must have the facilities to do this. In case an error has been detected, a list of questions involved must appear on the screen. Then the respondent must be able to select the question in the list to jump back to.

3.5. Navigation issues

With respect to navigation through the questionnaire, two approaches are possible. One is called active routing, and the other one passive routing.

Active routing means that the routing through the questionnaire is forced upon the respondents. They are not free in answering the questions in the order they like. Once they have answered a question, they can only go to the next question on the routing path through the questionnaire. It is not possible to go to questions that are not on the route. The only exception is that they can go backwards to the previous question. This allows for correcting mistakes that might have been made in answering a question.

The active routing approach guarantees that respondents only answer relevant questions, and that they are not confronted irrelevant questions. This also avoids dropout of respondents just because they see the mere quantity of potential questions that can be asked. A disadvantage of this approach is that respondents may feel too limited in their movements, and they may not like that feeling.

The passive routing approach lets the respondents completely free in their movements through the questionnaire. They can go to any question they like and answer any question they like. Obviously this gives more freedom. But there also is a big risk of accidentally skipping relevant questions, and answering irrelevant ones. This may lead to item nonresponse, and also dropout because of getting the impression that too many questions must be answered.

From a data quality point of view, we advocate the active routing approach. This means that the web interviewing software must be in complete control of navigation. Going to a next question, or possibly to a previous one, should only be possible to click on the appropriate buttons at the bottom of each question screen. One should take care to de-activate other navigation facilities of the browser, like the previous and next icons in the browser toolbar.

4. Implementing web surveys in Blaise

4.1. Blaise IS

This section explores the possibility of applying the guidelines described above in developing Blaise web survey instruments. To that end Blaise IS 2.0 was installed on Windows platform running Windows 2000 Server SP3 and Microsoft IIS 5.0. The Local Radio Listening Survey example was used to test the system.

The Blaise IS 2.0 system consists of three programs: the *Workshop*, the *Package Builder* and the *Server Manager*. Next to these programs, the *Online Assistant* provides the necessary information how to use these programs in order to set up a

web survey, starting from a Blaise data model (.bmi file). According to the *Online Assistant*, the three programs correspond to three roles in the survey process.

The *Workshop* is a tool for questionnaire designers. They can use this program, together with the *Modelib editor*, as a tool to define the layout of the web survey instrument. The *Workshop* can also be used to configure data entry behaviour (Interviewing, Edit-Check, Edit-No Check), response pages (web pages that are shown to respondents when the respondent submits or aborts a survey, or when an error occurs) and server settings (e.g. the location of the data file on the server).

The *Package Builder* is a program intended for survey managers. This program deals with survey management issues. It ensures that files concerning 'respondent authentication' and the 'interview start templates' are stored on the server. The *Package Builder* creates a package with all information to run the web survey.

Finally, the program *Server Manager* is intended for the server manager who monitors the technical aspects of the web survey (disk usage, security, and performance issues). Although, as in our case, it is possible that one person takes care of all these tasks, for large surveys it may be useful to have different people for these roles.

Traditionally, as introduced in Blaise 3, the layout of a Blaise survey is controlled by *Modelib* settings. For a Blaise web survey, the layout is controlled by the combination of *Modelib* settings and a 'style sheet'. The information required for a Blaise web questionnaire is stored in an XML file. The way in which the questionnaire is displayed on the screen is controlled by means of an XSL file. XSL (*eXtended Stylesheet Language*) is a language defining layout rules for the elements in an XML file. The beta-release of Blaise IS 2.0 provides two such style sheet files: one for a default web layout (*biHTMLWebPage.xsl*), and one for a default DEP layout (*biHTMLDEPPage.xsl*). It is possible to alter the existing style sheets, or even to write your own. However, this is not an easy task. The two style sheets both contain almost 3000 lines of XML code!

The style sheet *biHTMLDEPPage.xsl* is written with the purpose of creating a web survey instrument that is as close as possible to the *Data Entry Program (DEP)* that runs under Windows. Hence, this style sheet supports the settings of *Modelib*, and supports the menu settings of the DEP menu manager. A single exception is that it does not support the speed buttons of the menu.

The style sheet *biHTMLWebPage.xsl* is written as an example of a style sheet that is suitable for the purpose of a web survey. This prototype is supposed to be changed according to the wishes of the survey organization. By changing the style sheet, the web survey could include the logo of the organization. The style sheet *biHTMLWebPage.xsl* is written such that it only handles information in *Field panes*. The information in *Info panes* is not used. This explains why the default settings in the *Rijnwoude0* example (see section 4.2) resulted into a web page that only showed field names and input lines. No question texts or answer lists are displayed. In order to include these, we have to specify this using the *Modelib Editor* (see the *Rijnwoude2* example in section 4.4).

Section 4.2 describes the default web layout, and section 4.3 the default DEP layout. It is possible to create a totally different screen layout, and this is the topic of sections 4.4 and 4.5.

4.2. The default web layout

The first step in creating a web survey instrument with the default web layout was a traditional one: We programmed the sample questionnaire in the Blaise language,

and prepared it in the *Blaise Control Centre* to create the necessary files (e.g. the *bmi* and the *bdb* file). Next, we ran the three Blaise IS programs successively accepting default settings only. In the *Workshop* tool we merely selected the appropriate metadata file (*bmi* file), and saved the interview specification as *Rijnwoude0.bis*. We used the *Package Builder* tool to create a new web survey package, selecting the proper input file (*Rijnwoude0.bis*) and saving the new package as *Rijnwoude0.bip*. Finally, in the *Server Manager* tool we uploaded the package *Rijnwoude0.bip* to the Blaise Internet Server.

The result of choosing default settings only is a bare-bone interviewing instrument that cannot be used in a regular web survey situation. Figure 4.2.1 shows an example of the screen layout. The instrument has a number of major problems:

- There is no welcoming message;
- There are no question texts;
- There are no instructions on how to answer questions and how to proceed to the next question (or previous question);
- There are no navigation buttons, or other visual navigation means;
- For closed questions and all-that-apply questions, the possible answers are not listed;
- The answer to every question, including closed questions and all-that-apply questions, must be entered by means of a text input field;
- Horizontal and vertical scroll-bars must be used to navigate to other parts of the questionnaire;
- The questionnaire does not resemble its paper analogue;
- There is no clear survey description in the top part of the screen;
- There is no progress indicator;

Figure 4.2.1. Layout of a Blaise web survey instrument in default web layout



This first version has also some positive points: domain checks are carried out on entered answers, dynamic routing is applied, and the screen display is independent of the screen resolution. However, it is clear that the default version cannot be used for a web survey. It should be seen as a starting point for developing a more

realistic survey instrument. In section 4.4 we will show how we can use this style sheet and add question texts and navigation buttons.

4.3. The default DEP layout

As a next step in developing a Blaise web survey instrument, we used the default DEP layout. This leads to something that looks like the *Data Entry Program (DEP)* in *Interview* mode. This layout can be realized with minimal efforts.

We started with the prepared questionnaire in the *Blaise Control Centre*. Then we used the *Workshop* to alter one of the *Interview Specifications*. In the sub-tab *Stylesheet* of the tab *Layout*, we changed *biHTMLWebPage.xsl* into *biHTMLDepPage.xsl*. Again we used the *Package Builder* to create a web interviewing package. This package was added to the system with the *Server Manager*.

The result of these settings was indeed an instrument that resembled the Blaise *DEP* in *CAPI/CATI* mode. Figure 4.3.1 shows an example of the screen layout.

Figure 4.3.1. Layout of a Blaise web survey instrument in default DEP layout

Rijnwoede1 - Microsoft Internet Explorer

In the last seven days, how many hours did you listen to Radio Rijnwoede?

1. 0 - 5 hours
 2. 5 - 10 hours
 3. 10 to 20 hours
 4. More than 20 hours

Enter a value

PersonID	<input type="text" value="1"/>	Impress	<input type="text"/>
Listen	<input type="text" value="1"/>	Yes	<input type="checkbox"/>
ListenRR	<input type="text" value="1"/>	Yes	<input type="checkbox"/>
Hours	<input type="text"/>	Sex	<input type="text"/>
Programs	<input type="text"/>	Age	<input type="text"/>
Music	<input type="text"/>	Edu	<input type="text"/>
Day	<input type="text"/>		
Time	<input type="text"/>		

New | 1 / 1 | Modified | Dirty | Rijnwoede1

Internet

The screen layout has improved with respect to a number of aspects. We now have question texts. There are limited answer instructions. These are Blaise system texts. They may be changed but this is not a straightforward procedure. More extensive answer instructions could also be included as part of the question text, see section 4.5.

Radio buttons are used to present the possible answers to closed questions. Closed questions can be answered in two ways: by clicking on the proper radio button, or by pushing the corresponding numerical key. A long series of possible answers is distributed over several columns.

For check-all-that-apply questions, check boxes are used. Also here, a question can be answered in two ways: by clicking on the proper check-box, or by pushing the

corresponding numerical key. A long series of possible answers is distributed over several columns.

The screen displays one question at the time. No scroll-bars are required to make different parts of the questionnaire visible.

In principle, the top part of the screen cannot be used for displaying a logo, questionnaire title and a progress indicator. However, it is possible to include the survey title as the first item in each question text.

The bottom part of the screen presents a form view of the questionnaire. This is uncommon for web surveys. This view might help in giving the respondents some sense of where they are in the questionnaire form. If it works, a progress indicator would not be needed. And if one prefers to get rid of the form pane that can be taken care of in the *Modelib Editor*.

Probably the main problem with this layout is that it gives no clues as to how to navigate through the questionnaire form. It is not possible to include buttons for going to the next or previous question.

This instrument applies dynamic routing. Both domain checks and consistency checks can be carried out.

Although the default DEP layout is superior to the default web layout, it still is totally unacceptable as a web survey. While some trained interviewers may easily find their way with this layout, the average respondent of a web survey certainly will not. In addition we found that the style sheet with the DEP layout does not support navigation buttons. It is therefore less suitable as a basis for improvements. In the next section, we will proceed by improving the web layout of section 4.2.

4.4. Improving the web layout

Both the default web layout and the default DEP layout turned out to be insufficient for use in web surveys. In this section we will show how the default web layout can be changed in an attempt to come closer to an instrument satisfying the guidelines presented in section 3. Starting from the default web layout, we will combine question text and answer fields of a question on a single web page, and add navigation buttons to it.

In order to improve the layout of the web questionnaire form, we used the *Modelib Editor* to change a limited number of *Modelib* settings in order to put question texts and answer lists on separate pages. Figure 4.4.1 contains an overview of the new settings. The Data Model must be prepared in de *Blaise Control Centre* by setting the *Mode Library* to *Rijnwoude2.bml*. This can be done with *Project | Options*.

Figure 4.4.1. Modelib settings

Style	Form pane font: 10 (was 8)
Layout / Grids / DefaultGrid	BackGround: WHITE (was GREY) Cell width: 78 (was 38) Page width: 1 (was 2)
Layout / Field panes /DefaultFieldPane	Width: 78 (was 38) Height: 12 (was 1)
Controls / Field Name	Visible = FALSE (was TRUE)
Controls / Field Text	Visible = TRUE (was FALSE) Border = FALSE (was TRUE) Width = 78 (was 0) Height = 6 (was 0) Background = WHITE (was YELLOW)
Controls / Input Line	Left = 0 (was 12)
Controls / Answer List	Visible = TRUE (was FALSE) Border = FALSE (was TRUE) Top = 7 (was 0) Width = 78 (was 0) Height = 6 (was 0) Background = WHITE (was YELLOW) Columns = 1 (was 2) Codes = FALSE (was TRUE)
Controls / Answer Name	Visible = FALSE (was TRUE)
Controls / Remark point	Visible = FALSE (was TRUE)

Blaise's navigation buttons are available in the *Speed buttons tab* of the DEP menu. The style sheet for the DEP layout of the beta-release of Blaise 2.0 (i.e. the file *biHTMLDEPPage.xml*) does not support the DEP speed bar. However, the style sheet for the web layout (i.e. the file *biHTMLWebPage.xml*) supports a series of buttons: *Exit*, *Don't know*, *Refuse*, *Make remark*, *Previous question*, *Next question*, *Sub forms*, *Next page*, *Previous page*, *First page*, *Last page*, *Form language*, *Question help* and *Info*). We will limit ourselves to using two of these buttons: one for *Next question* and one for *Previous question*.

We used the *Dep Menu Manager* to add navigation buttons. Next, we created a Dep Menu file in the *Dep Menu Manager*. On the tab *Menu items* we de-selected the menu entries *Forms*, *Answers*, *Options*, and *Help*. Also, on the tab *Menu items*, we opened the menu *Navigate*, and de-selected all sub-menu items. We saved the menu specifications under the name *Rijnwoude2.bwm*.

An example of the result is shown in figure 4.4.2. The screen layout has improved with respect to a number of aspects.

We now have question texts. This version still has no answer instructions, although these could be added in the question text.

Radio buttons are used to present the possible answers to closed questions. For check-all-that-apply questions, check-boxes are used. In this version a long series of possible answers is not distributed over a number of columns. This requires more *Modelib* settings.

The input field for an open question always consists of one line. There are no simple instructions to create an input field consisting of more lines. Another problem with open questions is that entered text does not automatically wrap to a new line.

Figure 4.4.2. An improved version of the web layout



The screen displays one question at the time. No scroll-bars are required to make different parts of the questionnaire visible.

The top part of the question screen still displays its default characteristics. There is no logo, no longer title, and no progress indicator. Moreover, we see an ugly background colour, and a wrong text font. Sans serif fonts should be preferred.

This layout contains navigation buttons: one for going to the next question, and one for going to the previous question. However these buttons are not in an ideal position. One would prefer to have them at the lower right.

This instrument applies dynamic routing. Both domain checks and consistency checks can be carried out.

4.5. An even more improved version of the web layout

The layout of the web survey instrument in section 4.4 was still not as we would like to have it. So, we explored ways to improve it even further.

On our request, the Blaise team sent us two adapted versions of the style sheet for the web layout (i.e. the file *biHTMLWebPage.xsl*): one that allowed to add graphics to the title bar, and one that placed navigation buttons at the bottom of the screen. Based on the two style sheets, we created a new style sheet that combined these two functionalities, and that picked a different font for the header text. A further improvement was obtained by adding 'answer instructions' to the question texts, and by defining more detailed *Modelib* settings. We created different *Field panes* for different types of questions and added *Layout sections* to the Blaise questionnaire.

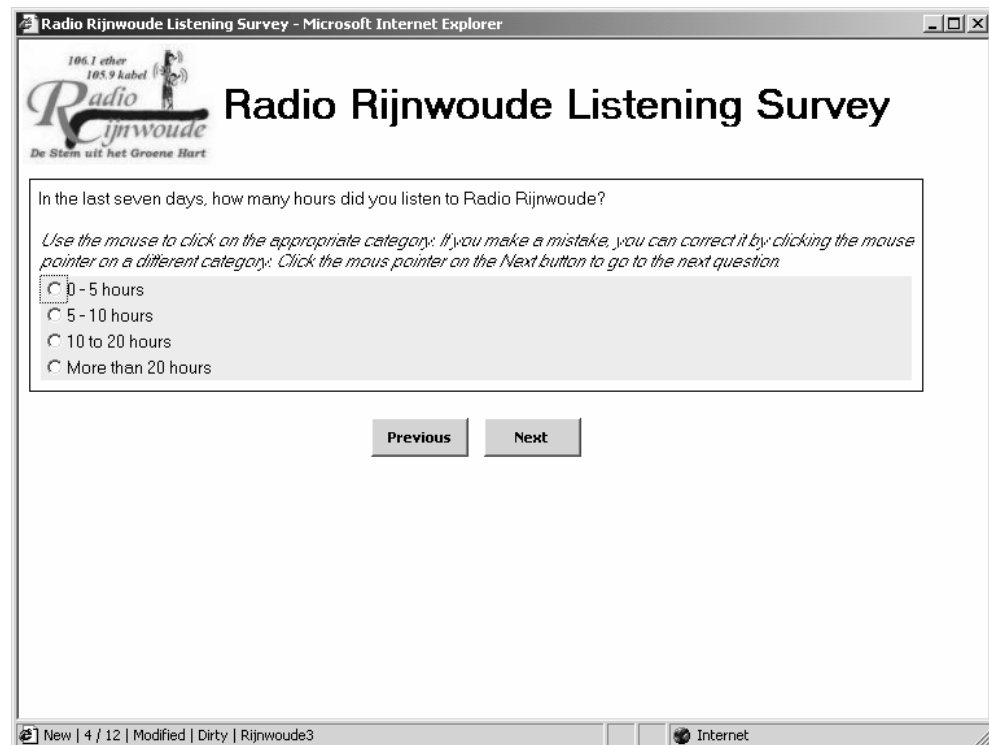
As a result, the top part of the screen has improved considerably. We now have a logo and a survey title. However, there is still no progress indicator.

Answer and navigation instructions have been included as part of question texts. These instructions are set in italics.

Radio buttons are used to present the possible answers to closed questions. It is now also possible to distribute a long series of possible answers over a number of columns. The same applies to check-all-that-apply questions.

The screen displays one question at the time. No scroll-bars are required to make different parts of the questionnaire visible.

Figure 4.5.1. An even more improved web layout



This layout contains navigation buttons: one for going to the next question, and one for going to the previous question. These buttons are neatly placed at the bottom of the screen.

This instrument applies dynamic routing. Both domain checks and consistency checks can be carried out.

4.6. Domain and Consistency checks

We mentioned that all web surveys based on Blaise IS are able to carry out domain and consistency checks. We like to comment briefly on these topics.

In the Blaise questionnaire that we programmed we have domain checks. The domain check is a result of the field definition, as in:

```
Age "What is your age? " : 0..120;
```

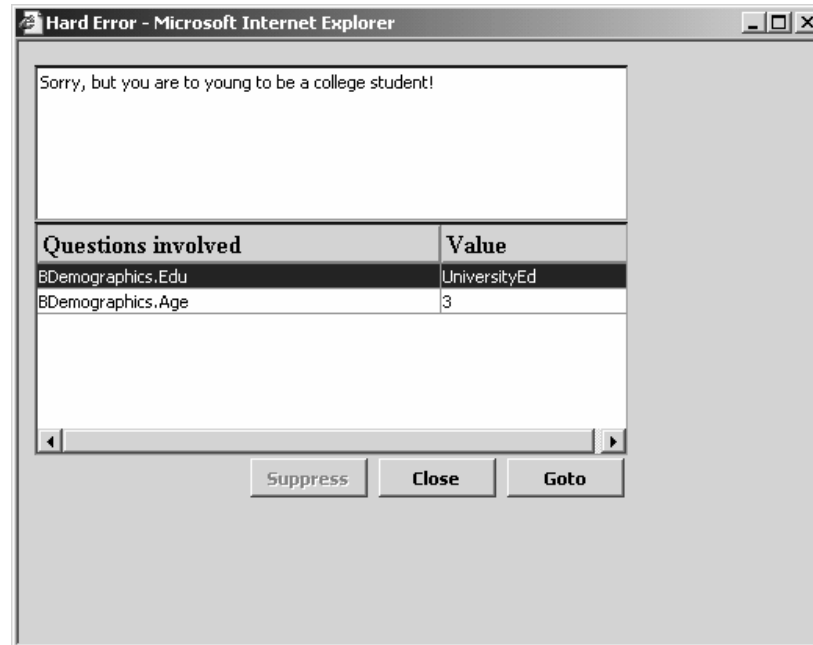
If a respondent provides an answer outside the domain, an error message pops up: "Input Invalid. Value not in range 0 to 120". The respondent can only click the *OK* button and returns to the text box that contains the out of range answer.

As an illustration we also added a consistency check:

```
NOT((Age < 16) AND (Education = EdUniversity))
"Sorry, but you are too young to be a college student"
```

In words the check means that we do not allow a person with an age below 16 to have university education. If a respondent violates this condition, an error message will pop up, see figure 4.6.1.

Figure 4.6.1. A consistency error has been detected



The top field displays the error message. The bottom field contains a list of questions involved in this error. The respondent is supposed to go to either the question *Edu* in the block *BDemographics* or to the *questionAge* in the same block *BDemographics*, and change its value.

Although the actions to be taken will be clear for experienced interviewers, we expect that naïve respondent will have large difficulty in dealing with this error message. For use in web surveys, we suggest to change this message in three respects:

- Change the name of the window (*Hard error*) in to something less frightening. Respondents will not understand what Hard error means. They could even think there is something wrong with their computer.
- The question names (e.g. *BDemographics.Edu*) and question answers (e.g. *UniversityEd*) will be cryptic for respondents. They will hardly be able to make a link between this information and questions already answered. A solution could be to use question labels and answer labels.
- The error message window should contain clear instructions about what to do. It must be clear for respondents that they must select a question, and go to that question to change its answer. Possibly, it is also a good idea to indicate that, after changing a question, the interview will continue from there. This could mean that respondent will see questions they already answered.

4.7. Conclusions

In this section we explored the possibilities of Blaise to create a layout for a web survey that satisfies as much as possible the guidelines presented in section 3. Blaise IS 2.0 contains two default layouts: (1) The DEP layout, which attempts to mimic the CAPI/CATI layout as much as possible, and (2) a default web layout.

In section 4.3 we discussed the shortcomings of the default DEP layout. Navigation is a fundamental problem here. Navigation with the Enter key and cursor keys works very well in a CAPI/CATI situation, where the interviewers are in charge of the instrument. However, for a web survey this type of navigation differs too much from web navigation. Therefore, it is by no means intuitive for respondents, and will be a source of problems, like dropping out of an interview in an early stage.

In section 4.2 we showed the default web layout. This layout is far from useful in a regular survey situation. It should be noted that the Blaise developers did not mean this layout to be used. They see it as just a starting point for questionnaire designers who will always want to adapt this layout for their own purposes.

By changing a *Modelib* file we were able to make a number of changes that made the instrument look more like a web survey instrument. This result was presented in section 4.4. To really come close to satisfying the guidelines in section 3, more substantial changes were required. The result of these changes can be found in section 4.5. This instrument comes close to what we would like to see on the web.

Finally we'd like to conclude with a set of recommendations and points of concern for the Blaise future.

Better and more style sheets

In our opinion, the adapted versions of the style sheet for the web layout we used in section 4.5, were essential for the success of our efforts. Producing a web survey instrument requires a lot of work. Although the style sheet approach allows a survey designer to tailor the web layout, adapting a style sheet involves extra work that should be done by specialists. We feel it would help many survey designers if the default web layout addresses as many problems as possible. Therefore, we would like to make a plea for an improved default web layout, or offering a number of default web layouts.

Navigation issues

We remarked that in the default web and DEP layout navigation is problematic. Although the addition of *Next* and *Previous* buttons improved navigation, the web survey still may show unexpected behaviour for naïve respondents. Problems occur when the respondent uses the keyboard to give answers. In the case of a closed question it is common Internet practise to use arrow-up and arrow-down to change the category. In the Blaise web survey, however, arrow-up and arrow-down are linked to the next or previous question. This may confuse respondents. We recommend the web survey to have the behaviour of a standard HTML form.

Open questions

We feel that the input field of an open question should have more flexibility. It would be nice if the questionnaire designer could set the number of rows and columns of the field, and thus given an indication of the size of answer that is expected. Moreover, text should wrap to a new line. We used the *Modelib Editor* to create a *Text box* by defining a *Field pane* for the open text question with an *Input line* with a height of 6. However, the resulting text box showed improper behaviour: the text typed remained on the top line of the text box, and scrolled horizontally.

Progress indicator

We did not succeed in including a progress indicator in the web survey instrument. We suggest making available a number of types of progress indicators in future versions of Blaise IS (text, bar, pie).

Check-all-that-apply questions

In section 3 it was suggested that answer possibilities of a check-all-that-apply question could be presented as a set of yes/no radio buttons. It is possible to redefine a set question in the Blaise questionnaire as a list of yes/no questions. This would however restructure the data model. It would be desirable to make it a layout matter, so that we can choose between a 'check-all-that-apply' question and a series of yes/no questions. Maybe this point reflects Blaise in general, and not only to Blaise IS.

Active routing and checking facilities

Blaise IS inherits a number of strong points from its CAPI/CATI version. One is that it applies active routing. And another strong point is the checking facilities. However, handling of error messages is a point of concern. Literature suggest using these checks sparsely. We expect that the error message caused by a domain check will not give problems, especially when it is clear to respondents why the answer they gave is not a valid one. The error message caused by a consistency check is too complex. The layout of the error message can be handled using a style sheet called *biHTML_ErrorDialog.xsl*. We recommend carrying out research in order to find out how respondents react to different forms of error dialogues. And if a suitable format is found, it should be incorporated into the error dialogue style sheet.

Some programming issues

We'd like to make some points with respect to the programming and development of a Blaise web survey. A point of concern is that the road to create a web survey is longer than the road to create a CAPI/CATI windows application. As a result, it takes more effort of a developer to find out what the effects of programming decisions are. In the case of a CAPI/CATI application, the developer can simply prepare and run the data model to find out if the application works, and how it looks. In the case of a web survey, such a test takes more steps. Although development tools as the *Modelib Editor* and the *Blaise IS Workshop* have preview facilities (we are very happy about that!), these are not always sufficient. In the case of the *Modelib Editor* the preview of the *Field pane* gives little information on how the result will look on the Web. The preview in the *Blaise IS Workshop* is much better, but requires more steps (saving *Modelib* settings in the *Modelib Editor*, and loading the bmi file in the *Blaise IS Workshop*. In case one uses dynamic text substitution, as we did to incorporate respondent instructions into the question text, the only way to see the result is to create a package of the *Interview specifications*, to upload the package to the Blaise IS server, and to test it.

Although we produced a list of points of concern and recommendations, we believe that the Blaise IS 2.0 edition is a very promising step to add web surveys to the set of data collection modes in the Blaise Family of Software.

5. References

Baker, R.P., Crawford, S. & Swinehart, J. (2002): Development and testing of web questionnaires. Paper presented at QDET.

Bethlehem, J.G. (1999): Cross-sectional Research. In: H.J. Adèr and G.J. Mellenbergh, *Research Methodology in the Social, Behavioural & Life Science*, Sage Publications, London, pp.110-142.

Bowker, D. & Dillman, D.A. (2000). An experimental evaluation of left and right oriented screens for Web questionnaires. Presentation to Annual Meeting of the American Association for Public Opinion Research.

Couper, M.P. (1999): Usability evaluation of computer assisted survey instruments. Proceedings of the Third ASC International Conference, Edinburgh, pp. 1-14.

Couper, M.P. (2000) : Web surveys: A review of issues and approaches. *Public Opinion Quarterly* 64, pp. 464-494.

Couper, M.P., Traugott, M.W. and Lamias, M.J. (2001) : Web survey design and administration. *Public Opinion Quarterly* 65 (2), pp. 230-253.

De Haan, J., Huysmans, F. and Steyart, J. (2002): Van huis uit digitaal: verwerving van digitale vaardigheden tussen thuismilieu en school. *Sociaal en Cultureel PlanBureau*, Den Haag.

Dillman, D.A. (2002): *Mail and Internet surveys. The Tailored Design Method*. Wiley, New York.

Dillman, D A., Tortora, R.D. and Bowker, D. (1998): Principles for construction web surveys. Technical report 98-50, Social and Economic Sciences Research Center, Washington State University, Pullman, WA.

Dillman, D A., Tortora, R.D. , Conradt, J. and Bowker, D. (1998): Influence of Plain vs. Fancy Design on Response Rates for Web Surveys. Presented at Joint Statistical Meetings, Dallas, Texas. August 1998.

Dillman, D A. Bowker, D. (2001): The web questionnaire challenge to survey methodologists. In: Reips, U.D. and Bosnjak, M. (eds.), *Dimensions of Internet Science*, Pabst Science Publishers, Lengerich, Germany.

Fricker, R.D. & Schonlau (2002): Advantages and disadvantages of Internet research surveys: Evidence from the literature. *Field Methods* 14, pp. 347-367.

Heerwegh, D. and Loosveldt, G. (2002): An evaluation of the effect of response formats on data quality in web surveys. Paper presented at the International Conference on Improving Surveys, Copenhagen, 2002.

Kish, L. (1967): *Survey Sampling*. Wiley, New York, USA.

Magee, Straight and Schwartz (2001): Conducting web-based surveys: Lessons learned and keys to success with known populations. Paper presented at the 56-th Annual Conference of the American Association of Public Opinion Research (AAPOR), May 17-20, 2001.

Sangster, R.L. & Kennedy, J. (2002): Structured and flexible navigation within XML web surveys. Paper presented at the International Conference on Improving Surveys, Copenhagen, 2002.

Schneidermann B. (1997): Designing the user interface -- strategies for effective human-computer interaction. Addison-Wesley (3rd edn).

Tufte, E. (1983): The Visual Display of Quantitative Information. Graphics Press, Cheshire, Connecticut.

Appendix. The sample questionnaire

Radio Rijnwoude wants to better serve its listeners and to stay in tune with their preferences. Therefore, an extensive audience survey is carried out.

To obtain a representative sample of listeners, it is very important that you fill in this questionnaire.

1. The last seven days, did you listen at least 5 minutes to the radio?

Mark one box.

- Yes
 No → Skip to 9

2. In the last seven days, did you listen at least 5 minutes to Radio Rijnwoude?

Mark one box.

- Yes
 No → Skip to 9

3. In the last seven days, how many hours did you listen to Radio Rijnwoude?

Mark one box.

- 0 – 5 hours
 5 – 10 hours
 10 – 20 hours
 More than 20 hours

4. Which type of program did you listen to most during the last seven days?

Mark all boxes that apply.

- Music
 News → Skip to 6
 Sports → Skip to 6
 Culture → Skip to 6
 Other program → Skip to 6

5. Which type of music did you listen to most?

Mark one box.

- Chart / Top 40
 Folk
 Dance
 Easy listening
 Rock
 Jazz
 R & B

- Classical
 Hip-hop
 New age
 Country
 Other music

6. At what type of day did you listen most to the radio?

Mark all boxes that apply.

- Weekdays
 Weekends

7. At what time of day did you listen most to the radio?

Mark all boxes that apply.

- Morning (6 – 12)
 Afternoon (12 – 18)
 Evening (18 – 24)
 After midnight (24 – 6)

8. What is your general impression about the quality of the programs of Radio Rijnwoude?

Print comments.

9. Are you male or female?

Mark one box.

- Male
 Female

10. What is your age?

Print number in boxes.

11. What is your highest level of education?

Mark one box.

- Primary education
 Secondary education
 Tertiary education

This is the end of the questionnaire. Thank you for participating in the survey.

