PUBLISHED IN THE PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON INTERNET SURVEY, STATISTICS KOREA, DAEJEON, SOUTH KOREA, 9-10 SEPTEMBER 2009.

Getting the technology right for online surveys with nationally representative samples: ensuring the tools live up to your demands

By Tim Macer, University of Southampton

Abstract

Internet surveys are born of technology. Pages, forms, buttons and the other components of the Web survey are so commonplace in the everyday language of the Internet that Web programmers and designers can overlook the actual needs of the researcher wishing to conduct surveys online for the first time, and provide researchers with tools that are inadequate for the scale and complexity task. Fortunately, there is a considerable range of software available for online research today for all levels of need and complexity. However, it is important to know what to look for, when deploying software for Internet surveys.

In this paper, we will outline the typical architecture of an online survey tool; identify its different components and key areas of functionality. We will consider which advanced features are desirable for the professional researcher, and emphasize those which are particularly beneficial from a methodological perspective, in terms of controlling survey error, particularly measurement and nonresponse error.

Where nationally representative samples are required, an online sample is usually inadequate on its own, and a mixed mode approach is considered necessary by most institutes working with national statistics. We will identify the key requirements of survey software capable of supporting a mixed mode approach, and discuss some of the technical and methodological considerations that can help to reduce effort and potential error in what is usually a highly complex operational task.

Introduction

Since Internet surveys were first deployed in market research and social research in the late 1990's, a great deal has been written by scholars and practitioners about the methodology and practice of what has now grown to become the predominant interviewing channel by professional research companies (ESOMAR, 2009: 107). It is a growth that seems likely to continue for several years to come.

Computer software, and particularly, specialist software designed around the survey process has been available for at least four decades now, and has colonised every aspect of the research process.

Yet unlike the methods that preceded it, such as mail, telephone or face-to-face interviewing, it is not possible to conduct Internet research without the use of a computer, or to create a survey without using computer software. The software system provides a dual role in providing the context in which the survey instrument is delivered, and in systematizing the process, including the design, execution and management of the survey.

The specific survey software used to design an Internet survey therefore occupies a central role in defining that survey and ultimately is defining the limits of what the survey designer is capable of achieving. Despite this, there has been relatively little discussion of online survey software or its role in supporting online survey methodology in either publications or at conferences.

The aim of this paper is to identify the main concerns or considerations of the Internet survey researcher, particularly when working at an advanced level (such as within large research institutes, national statistics services and so on). I will do this by exploring the following five themes:

1. System architecture and design. Here, I will describe the constituent parts of typical Web survey systems and highlight how system design and capabilities can be instrumental in supporting operational efficiency in the survey process.

2. State of the art technology. I will consider some of the more recent advances in survey technology and how these can be used to improve the Internet survey.

3. Methodological concerns. Here, I will review a number of the more important methodological concerns in relation to Internet surveys, emphasizing recent learning, and relating these observations to the kinds of support that are demanded from survey software.

4. Mixed mode considerations. Then, I will extend the discussion to include a number of specific software requirements for mixed mode research, where the Internet is one of a range of methods being deployed.

5. Deployment. Finally, I will discuss a number of observations with regard to research institutes either using standard software packages or developing their own.

I will use the terms Internet survey, Web survey and online survey interchangeably to describe self-completion research surveys delivered over the Internet in real time and conducted on a personal computer (or standard laptop computer) with a standard Web browser with an assumed resolution of at least 1024 by 768 pixels. I will similarly use Internet research and online research to describe survey research carried using an Internet survey, and Internet researchers to describe practitioners engaged in designing and interpreting the results of Internet surveys.

I will not be discussing surveys delivered to small format Internet devices such as mobile PCs and Smartphones, where requirements and practice are very different from online surveys.

Just as online research is now a subject, which has been extensively discussed and delineated by survey practitioners, it is also now well understood by specialist software vendors. There is a surprisingly broad choice of off-the-shelf software packages available for online research and also for mixed mode research. The functions offered by a great many Web survey software applications reflect a largely consensual view on the basic needs of the survey researcher. The observations I will be making are generalised from a broad range of commercially available software packages, though this paper does not seek to provide specific recommendations on individual software packages.

In any case, it is my experience that software vendors are adept at borrowing ideas from one another, as well as enhancing their software to make up for any perceived lack, which tends to make comparative reviews out-of-date within a very short period of time.

A possible framework I have found useful conceptualise and analyse software packages for online surveys is as follows:

1. **Efficiency**: the extent to which the system provides a highly efficient working environment for the administration of online surveys by automating processes, for example, by allowing the re-use of elements of previous surveys and by facilitating collaboration between colleagues involved in the research activity;

2. **Flexibility and scalability**: the extent to which the software supports all that the Internet researcher needs to do, without imposing limits on the complexity of the survey instrument being designed, the methods that may be applied or the amount of data that may be collected.

3. **Accuracy and reliability**: the extent to which the system allows for the creation of robust and methodologically sound survey instruments that do not unwittingly increase the scope for survey error, but provide support for the active control and management of survey error.

This framework also reflects the first three of the paper's themes outlined above, with good system design facilitating efficient working; recent technological advances offering improvements in flexibility and scalability, and the over-arching methodological concerns of the survey researcher with respect to the control and management of survey error.

However, methodological considerations go beyond the topic of survey error, and must include an awareness of the experience from the participant's perspective, notably respondent burden and engagement. Increasingly, researchers are expressing a desire to make surveys 'engaging' for participants, which is not merely an ethical consideration, as it is also an important means to improve response.

In the final section on deployment, I reflect on the desire that often exists in research institutes to save money by building their own survey software, and why, in my view, this is false economy and a misguided endeavour.

I will now conclude the introduction with a definition and summary of survey error, as this will provide a context for much of the subsequent discussion.

Survey error

Couper (2000) and others describe the major sources of survey error in Internet research using the models elaborated earlier by Groves (1989) for face-to-face, postal and telephone surveys. These are: sampling, coverage, nonresponse, and measurement error. Sampling and coverage error arise prior to the execution of the survey instrument, but nonresponse and measurement error principally arise during the execution of the survey instrument. Yet the very close relationship that exists between software and the survey instrument in online research means that the software, its capabilities and its limitations, are likely to be highly influential on its execution, and thereby on nonresponse and measurement error. It is therefore ironic that, in the discussion of error in online research, attention has largely focused on sampling and coverage as sources of error, rather than errors arising from the execution of the survey (e.g. Taylor, 2000; Poynter, 2001). Indeed, there is often an implicit assumption that Internet surveys handle nonresponse and measurement error rather better than more traditional methods, because they are more rigorous in their data capture.

Until relatively recently, where nonresponse and measurement error have been considered in online research, it was usually in comparison with a 'conventional' method such as telephone or paper-based self-completion, as attempts were made to understand and explain differences between online and traditional methods. It is only in recent years, discussion has started to focus on the influence that the online survey instrument can have as a source of measurement error, and how adjustments to the survey instrument can reduce error. It is an area where best practice is still being explored.

As the focus of this paper is online survey software, and ensuring that the technology employed is fit for purpose, I shall concentrate principally on the influence of nonresponse and measurement error, and how software can be engaged to improve the design of survey instruments in order to minimize both. I shall make reference to coverage effects when considering the extent to which a survey instrument is universally accessible to the entire subpopulation of Internet users, including those still using older generations of web browsers or communications links.

However, sampling and coverage issues can also be addressed by adopting a mixed methodology, as they often are now in national statistics or social statistics surveys where exacting nationally representative samples are a necessity. A mixed-mode survey raises another set of demands on the software to support additional research channels and adds to their complexity, and I will discuss these in a specific section.

Operational considerations: efficiency in practice

The majority of the operational considerations described below address the area of efficiency, though there are some aspects of system design and architecture, which can offer scope to address measurement and nonresponse error. First, I will explain how systems are typically organized and some improvements that are now possible, using contemporary database and web-based technologies.

Metadata and paradata

In the following sections I will be discussing two data constructs: *metadata* and *paradata*. It may be useful to start with a definition of both of these terms.

> **Metadata** is a term widely used in survey research to mean the data which describe or define the survey and associated case data, presented in a computer readable form. Metadata usually describes each question, the wording of the question, the type of data it holds, textual labels associated with the answers and other information on the format and structure of the data.
>
> **Paradata** is a terms to describe incidental data collected about the progress of the survey instrument through each interview. It is less widely used as a term than metadata. Paradata might include such information as the time of day the interview started, the IP address of the computer, how long the participant took to complete each question or even record whether the participant went backwards in the interview, and where they went back to.

Surveys, survey instruments and questionnaires

The majority of Internet survey systems tend to divide the work and the data into discrete surveys. A survey will typically be considered by the system to be made up of the following components:

- A questionnaire: the survey instrument, which also provides the survey metadata;
- Sample: the set of invitations to be issued;
- Survey data: the case data collected, usually one record per interview;
- Activity data: information on each individual's participation, such as the time of day, length of interview, browser and computer configuration used, and so on: the paradata.

The majority of Internet survey software packages tend to use the term survey and questionnaire (the survey instrument) interchangeably. A small number of systems separate the functions, and allow a questionnaire to be developed and then applied to multiple surveys.

Within the survey instrument, and reflected in the case data, are the individual questions. Systems will often use the terms 'question' and 'variable' interchangeably.

Sample and case data

If the survey is being administered to an invited sample of participants, e.g. by email invitation, or by mailing them a unique code to access the survey, then the case data will contain an entry for each invitation or potential participant. As participants progress through the survey instrument, their case data record will be supplemented with their responses.

It is convenient to consider the case data that exists before the start of the survey to be the sample (and some systems explicitly give it this name), even though in a strict sense, these records represent the *potential sample*, and it is only the completed interviews that make up the *achieved sample*. These records may also contain known characteristics or demographic data, which could be a generalised item such as a geographic region code or it could be specific information on that respondent from a previous phase of research. Either way, it is useful to be able to access this information within the survey instrument for the purpose of verifying responses provided in the interview, or to control the questions being asked. It also means that these data will be available alongside the collected data when analysing the final results.

Activity records

A good system will also generate an activity record for each attempt by a participant to participate in the survey. Note that there will not be a one-to-one correspondence between case data and activity data, as some participants may make more than one attempt to complete the survey while others will make no attempt, and there is no activity to record.

Activity records can provide a useful source of data when validating the data or seeking to investigate issues such as poorer than expected participation. Some systems will only provide minimal data, such as the time and duration, while the more advanced ones will provide paradata on each individual question, such as how long it was displayed for, the order in which items were selected, and even items which were selected then discarded. They may also allow for the researcher or system administrator to customize the paradata being recorded. Paradata can be especially useful in the detection of interview fraud, which is discussed later.

It can often be more convenient to record some key paradata values within the case data record, as this will simplify any analysis the researcher wishes to perform on the influence of external factors on the case data, such as the time to respond, speed of response, time of day and so on. If the system allows this to be automated or incorporated into a standard survey template, then so much the better.

Status and outcome

The system must allow researchers to differentiate easily between complete and incomplete interviews, as normally only completed interviews will be used for analysis. Systems will normally provide a range of "outcome codes" to signify the status of that interview. In Internet surveys, there are usually at least three principal statuses or resting states for an interview record: no response, incomplete (the participant started the interview but did not reach the end) and complete. While the survey is live, there may also be records which are active.

Because of the nature of Internet communications, a survey will be completed in a series of disconnected bursts of activity, as a response is provided and the survey server transmits the next question. As the participant reads the question and chooses his or her responses, the system is unaware of whether they remain active or not. This is different to a telephone call, or in research terms, a telephone interview, where an interviewer would be aware immediately if the respondent hung up on them. Indeed, a participant may leave a question on screen for a very long time, if interrupted by a phone call, for example, and may appear to have taken 30 minutes or more just to answer one question. This means that interview timings for Internet surveys always need to be treated with some caution. It also means the decision as to when an interview is no longer active is also somewhat arbitrary. Usually, the system will set a time-out period after which the participant would need to formally start again, and a new attempt would be recorded. This should make no difference to the recording of the case data, however. The convention is that, when a break-off does occur, the participant will resume from the same point in the interview, though the survey designer must be able to override this, and in some situations, force the interview always to recommence at the first question.

More advanced systems offer considerable flexibility over defining custom outcome codes and defining survey-specific statuses and even time-out values.

These elementary components are illustrated in Figure 1.

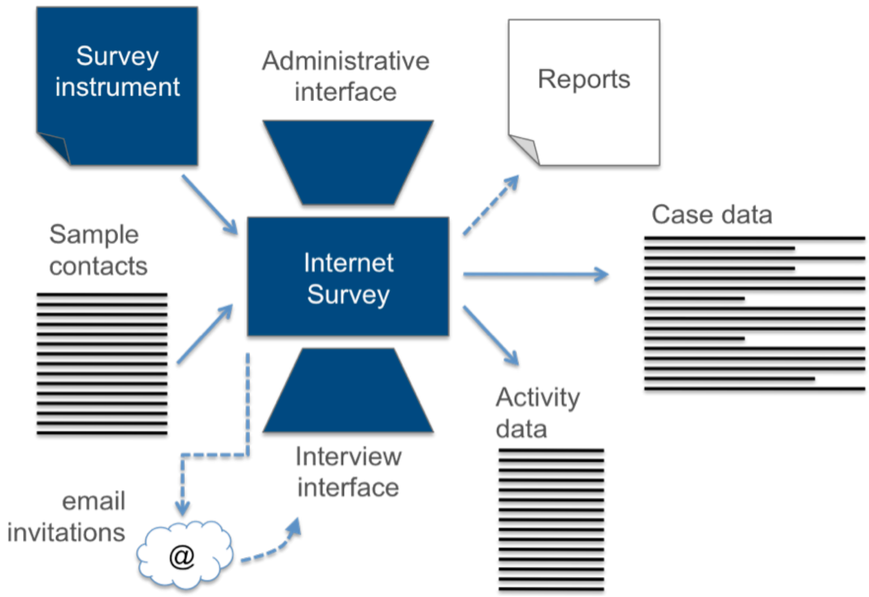

Figure 1 Components of an Internet survey

Sub-records

Some research designs will contain repeating sub-sections of questions. This is commonly referred to as a *hierarchical* design. Examples of this include studies in which participants are requested to keep a diary of activities – journeys made or radio programmes heard. Each sub-record contains the same set of questions (i.e. it has the same structure) but a varying number of these sub-records may occur. This capability is a key differentiator in the capabilities of Internet survey software applications, as many do not allow for hierarchical survey questions or sections of questions. The more sophisticated systems allow for questions or sections to repeat, either by explicitly declaring a sub-record or level, or implicitly, by declaring a loop.

If the software does not support loops or hierarchical sub-records, a compromise design is to define a section of questions and then explicitly to repeat or duplicate them, to allow, for example, for up to ten or even fifty repetitions. This can lead to inefficiencies in processing, as the data records will be sparsely completed, or may lead to loss of data if more repetitions are required than have been anticipated at the design stage. Creating these structures in this manual and repetitive way is a painstaking and error-prone process, as is analysing the data afterwards. Integrated support will not only save time but also virtually eliminate the scope for error.

Research designs are also possible where there is more than one level in the hierarchy, e.g. a survey of household travel, where data may be collected for the household as a whole (level 1), individual household members level 2) and each separate journey taken by each person in the household (level 3). Multi-level designs are too complex to be handled manually, and require the system to support either hierarchical records or nested loops.

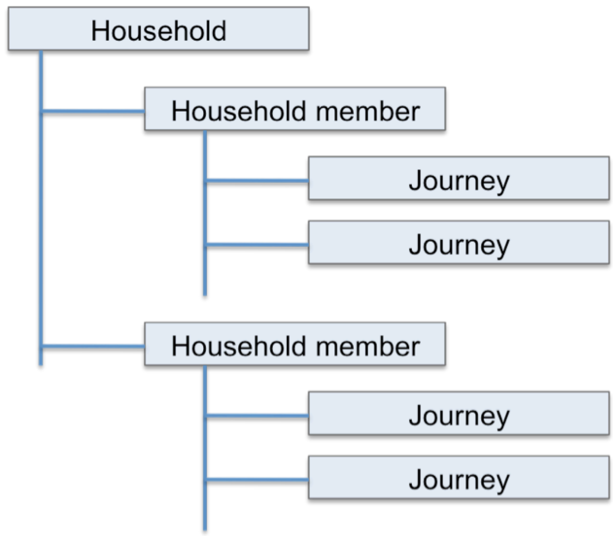

Figure 2 Example of a hierarchical or repeating data models

Survey metadata

Survey metadata can be considered to exist at two levels:

1. For the survey as a whole, where it may describe such characteristics as the fieldwork period, methodology, sample frame etc.

2. For the survey case data, where it is used to describe each field, usually providing at least a name and a textual label or annotation for each field, as well as the characteristics of that data in that field.

I will only be dealing with metadata in this paper in its second definition – as a computer-readable description of the case data.

An essential function of the survey instrument, in the context of an Internet survey system, is to provide the metadata for the survey. This will consist of both the characteristics of each field, as described above, and also the execution logic applied by the survey engine when presenting a survey to a participant.

It can be very useful if the survey system will also make this metadata available to other software, particularly when transferring case data to another program for statistical analysis, as this will ensure that the data can transfer automatically, without the need for any manual intervention or recoding of data. Recoding the data manually should be avoided, as it is a significant source of error, and the quality control processes that become necessary to deal with this are costly and time-consuming.

Metadata standards

There is a survey-specific metadata standard, the Triple-S standard (Hughes, Jenkins and Wright 2000), which has been adopted by over 60 different software packages. This is a relatively simple XML-based standard that deals with most common types of survey case data, though not with survey logic. However, it will recreate all of the original questions or variables automatically, with their appropriate names and text labels within the receiving software tool.

Two other de facto standards (though not independent standards) for data transfer are the proprietary data formats used by the SAS and SPSS statistical packages. Many Web survey packages will output data in this format, and many survey analytical tools will read data represented in this format, and use the metadata to recreate the variables, labels and values ready for analysis.

There are no agreed metadata standards for the transfer of survey instruments from one system to another, with all of their questions and execution logic.

System modules

An Internet survey tool is likely to comprise a series of different modules to handle each stage of the research process. A typical set of functional modules, which would be necessary for most professional survey applications, is set out below (and illustrated in Figure 3):

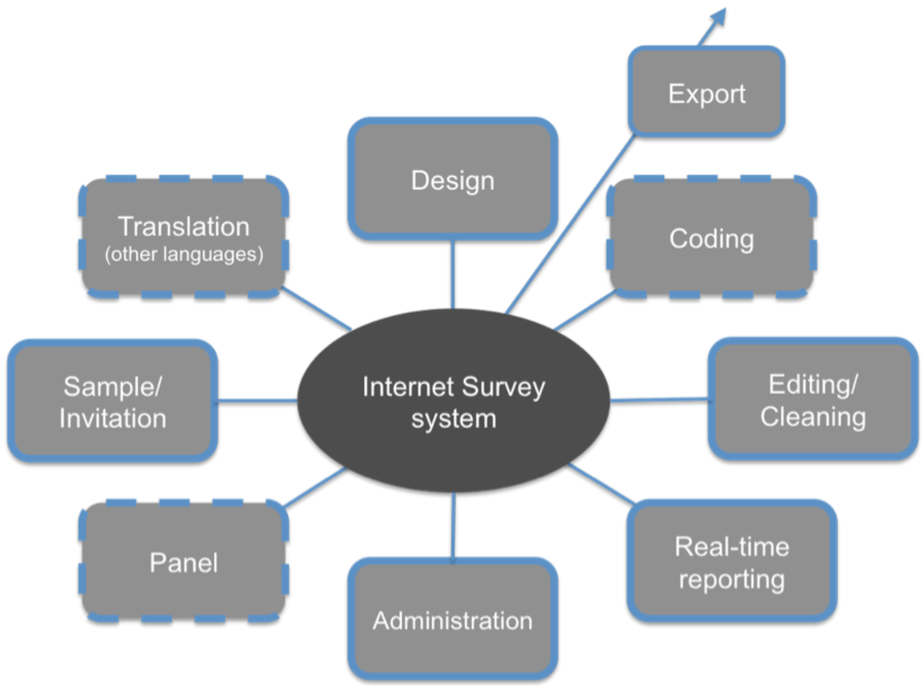

Figure 3 Principal modules of an Internet survey system

Survey designer

This is an essential component of every Internet survey system, in which the survey is created. In older systems, this was achieved by writing syntax, but most modern systems use a graphical interface and allow the survey designer to see how the question will be presented at the time it is created. Though appearance is important, so too is the integrity of the data, and the survey design module should avoid the need for the designer to be concerned about how or where the data are stored or validated, but should provide a menu of different question types and related validation conditions that will ensure the integrity of the data.

Translation module

Where a survey instrument is to be administered in more than one language, it is simpler to complete the design in one language alone, and then translate it into the other languages.

The simplest survey systems require that the survey instrument be duplicated in order to allow for other languages to be accommodated. This is unsatisfactory, as any subsequent amendment to the survey must be applied systematically to each language copy, which provides scope for inconsistency and error. This method also does not allow the participant to choose their preferred language.

More advanced systems tend to recognise that the survey instrument needs to incorporate all required languages and allow participants to switch from one to another. They tend to handle translation by one of two methods:

1. All of the texts are exported into a structured document such as a Microsoft Excel workbook, and the translator then sends back a version of the document that includes the new language within the same structure, which is then systematically imported into the survey instrument

2. A translator's interface is provided to the system to allow direct access to the survey instrument.

Administration/management

For larger organisations, it will not necessarily be the survey author who will decide when the survey is to be deployed, or when sample will be released. A separate administration or management function will allow for survey deployment and control to be handled independently of the survey authoring stage.

Sample and Invitation management

This module is used to organise the sample, create customised invitations, despatch invitations by email to the selected participants, and should also allow for reminders to be created and despatched automatically only to those who have not responded. Where large samples are involved, the release of the sample may be staged in batches. The more sophisticated systems allow for this process largely to be automated. In some cases, this is a function of the survey administration interface; in others, it is a separate module.

Panel management

Internet surveys, particularly in the world of commercial market research, are typically fielded to respondents who have opted to join a research panel. Some Internet survey systems offer sophisticated functionality for panel administration, whereas in others, these features barely reach beyond the management of the survey invitations.

Real-time reporting

Access to reports showing the current status of the survey, the response rate being achieved and other related data is essential. Respondents tend to respond to survey invitations almost immediately, and it is therefore important to have as complete a view as possible about the performance of the survey once it is live. In particular, it is important to know if there are any problems with refusals or non-response, e.g. if participants are abandoning the survey without completing it, and if so, which are the questions where the termination is most likely to occur. This may offer scope for a partial redesign, or at least, to clarify wording, to support or reassure participants better at that point.

Real-time reports may also provide useful early indications into trends emerging from the data, though care must be taken, when sharing these reports with non-researchers, such as internal or external clients, that they understand the provisional and incomplete nature of the data.

Editing and cleaning

Surprisingly few systems offer sophisticated tools for editing and cleaning data. These tools need to detect suspected errors, detect outliers and either flag these for individual attention, or clean or purge them systematically, according to rules which can be configured for each survey.

Coding

An integrated coding module within the Internet survey tool will allow for verbatim responses to be analysed within the survey software, can offer advantages of speed, accuracy and efficiency over other methods, particularly those that involve exporting the unclassified data and then re-importing the classified data.

The majority of systems, however, tend to mirror manual processes in the coding facilities they provide. The coding module simply provides an interface to gather together all of the classification work to be done, then presents each verbatim response for classification on a case-by-case basis. Further automation may be provided by allowing coders to search for items and even to classify a mass of responses where there is an exact match, e.g. the name of a place or a product. These tend not to cater for close matches deriving from spelling or typing mistakes, and until recently, classification of verbatim responses has remained a time-consuming and labour-intensive activity. This can be particularly problematic with online surveys, where verbatim responses can be both lengthy and plentiful compared to other interviewing modes.

There have been some interesting developments in recent years, however, in the application of computerised text categorisation based on machine learning models. Machine learning appears to be particularly well suited to the task of coding verbatim responses in surveys (Macer, Pearson and Sebastiani 2007) because the system can be trained for each specific question or research topic with a relatively small set of training examples in a similar way to conventional coding. From these examples it will then build automated classifiers which will take very similar coding decisions to those taken by manual coders. Unlike manual coders, however, large amounts of data can be processed very rapidly, its performance can be monitored closely, and if omissions or mis-classifications start to emerge, the system can be retained through providing additional examples, or performing corrective action upon wrongly classified cases, and the categorisation model then rebuilt and reapplied to all of the data. By contrast, manually recoding data is extremely costly and is therefore very rarely carried out.

Export

Export capabilities will allow the user to obtain data and metadata, typically in a choice of formats, so that they can be transferred to other systems for data processing, analysis and reporting. The Triple-S format already discussed will allow files to be transferred to a range of other systems. Other useful formats are SPSS and SAS, as these are also likely to involve creating comprehensive metadata to describe the data.

While most exports will be concerned with obtaining case data and the associated metadata, it is also useful to be able to export other data from the system, such as:

- A listing of the survey instrument, as a text or Word document.
- Activity records or survey paradata.
- Sample, including any unused sample.

Advanced system architecture

Web-based architectures

The majority of commercially available Web survey tools these days are entirely web-based systems; with all of the 'back office' modules typically being operated via a Web-browser interface. Systems which cater for multiple modes (e.g. CATI, CAPI and even paper) are less likely to be completely web-based, but there are still many to choose from that use this model.

An important consideration for the larger research organisation is whether the software is available for it to install within its own data centre, or on a web server under its own control. Many of the Web-based products are only offered on a 'Software-as-a-Service' (SaaS) basis, whereby the surveys and data all reside on the provider's server, although readily accessible to registered users. This does not necessarily compromise security, however. Some of the high-end SaaS providers have been subject to, and satisfied, extensive security audits from very large clients who are extremely conservative from a security standpoint, such as banks, police authorities and the military. Most SaaS providers would probably not be able to meet this level of security, however.

The larger research institute may also be interested in installing the software on their own servers for reasons of cost-economy too, though it cannot be assumed that this approach will reduce cost in every case, even for the largest operators.

Relational databases

It is relatively recently that Internet survey systems have started to exploit relational database platforms such as Microsoft SQL Server, Oracle or MySQL. Traditionally, case data, sample data and so on were collected, held within the system and processed as simple sequential files, and this is still a very common model in survey system architecture. Relational databases are not particularly well suited to survey data. They have developed in response to the needs of large-scale business systems, which will often require random and unpredictable access to any record from among thousands or millions held. Consider, for example, how data records may be used within an airline reservation system, a corporate asset management system or, at a smaller scale, a Web content management system, compared to the orderly progress of the records though a web survey system, where samples are typically of thousands and very rarely hundreds of thousands.

Access to survey data is more predictable than that in most enterprise-type systems, and the volume of records is often much smaller. Survey metadata is also complex than is typical in business systems and is also different for each survey. Until recently, there has been little motivation to move from a predominantly sequential file-based model to a relational database model, but that is now changing, and a new generation of Web survey tools has emerged on the market in the last few years which make full use of the capabilities of the relational database.

Scalability

Relational database systems are generally more scalable, and can handle large volumes of records without any noticeable degradation in performance. This is important when a survey does have a very large volume of records, such as some of the multi-million record surveys carried out by national statistics services. For this type of work, nothing less than a relational database system should be considered.

Commercial market researchers are also finding a relational database model useful in the context of panel administration, as a respondent panel can consist of many hundreds of thousands of records, possibly millions of records.

Open systems

Web survey tools built upon relational databases also tend to be more open, allowing the survey tool or the data to integrate with other software or sources of data: for example, to perform real-time validation of addresses, or even to interpret textual data.

Although a database such as an Oracle or Microsoft SQL Server database can often be read and even updated directly from another application, this is far from good practice. An open system should allow other software to communicate with it through an application program interface (API), which provides a standardised and documented protocol that other programmers can use to allow their programs to interact with the survey system.

A very useful development for Internet survey software is the Web Services Architecture, which "provide[s] a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks" (Booth, Haas and McCabe 2004) and is a standard endorsed by W3C, the Worldwide Web Consortium. The Web Services architecture takes principle of the API and applies it to the Internet, allowing Web-based software to communicate directly with other Web-based programs that may be located anywhere, either to control functions of the software or provide or exchange data.

This means the capabilities of any system with a Web Services interface can be enhanced either by writing a 'plug in' application, or by using an off-the-shelf third-party enhancement, and that the survey researcher is effectively no longer limited by the native capabilities of the survey system being used.

There are already third-party plug-in applications available for survey research activities for such diverse tasks as panel management, verbatim response coding, automated text classification, fraud detection and advanced questioning techniques such as Flash-based sortboards and conjoint models. There are likely to be many more such plug-ins in the future.

Survey silos and question databases

A disadvantage of many Internet survey systems is the rigid approach they apply in segregating one survey from another. This makes it difficult to access questions or data from other pre-existing surveys when working on a new survey, or to access all the other data that may relate to a particular participant in the context of a longitudinal study or where that participant may have taken part in multiple surveys as a part of a panel. This is largely due to the architecture inherited from the pre-database era where the survey data and metadata would typically be represented as separate sequential or indexed sequential access files. However, this rigid separation of surveys remains a persistent architectural model with many of today's Internet and even multi-modal survey systems.

The relational database model does make it possible to break down the barriers that result in the different survey data being held within different 'silos' (see Figure 4) by allowing the system to create linkages between surveys, question and participants at different levels, wherever there is a common point of reference (Figure 5). At a respondent case level, this could be the respondent identifier, or it could be other proxy identifiers; at a survey level, it could be an entire question and the previous characteristics of how that question has been answered.

Currently very few survey systems support this true relational model, but a few are starting to emerge, particularly in the area of systems optimised for use with large respondent panels, where the benefits of being able to link all responses from an individual across all surveys they have taken are most obvious.

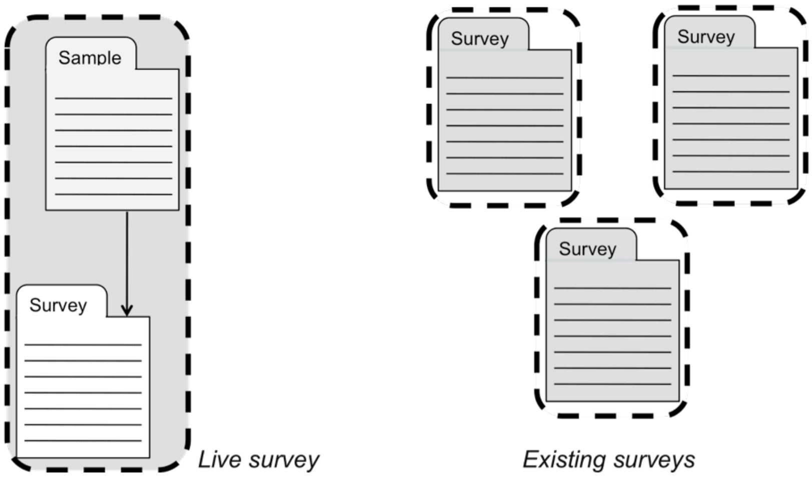

Figure 4 Non-integrated 'Silo' survey model

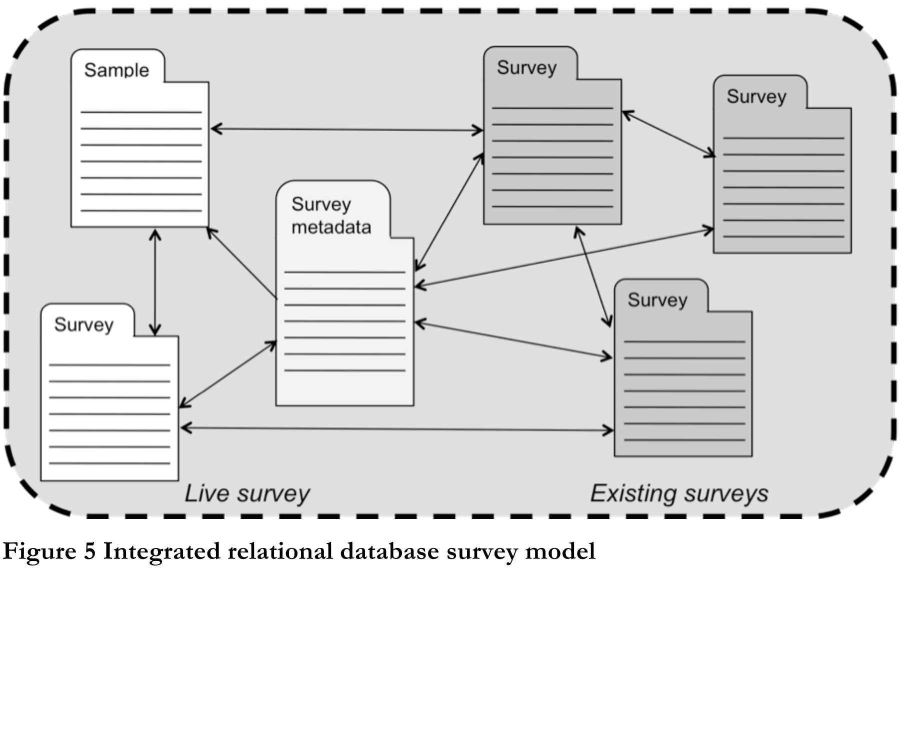

Figure 5 Integrated relational database survey model

Methodological considerations

There are enormous variations in what different practitioners mean by an Internet survey. It is a spectrum which range from, at one end, simple scrolling forms as a direct online counterpart of a self-completion paper questionnaire such as those originally advocated by Dillman (2000) in his 'total design method' through to rich multi-media surveys containing audio, video and highly interactive components (e.g. Fuchs and Funke 2007).

Many practitioners advocate simplicity in online survey design. The key design criteria set out by Poynter, a pioneer in online research in the UK, in 2001 are equally relevant today, despite greatly increased sophistication in both Web survey software capabilities and the capabilities of the Internet in general:

- Clear instructions
- Engaging design
- Error-free questionnaires
- Short interviews
- Treating the respondent with respect

Source: Poynter (2001)

It is also observed by many that survey researchers over the years have not been imaginative in designing survey instruments for the Internet, often simply putting forms online which were the direct equivalent of a paper questionnaire (Terhanian 2003), and Couper (2007) has observed more recently that a great many web surveys still look and feel like paper surveys. In commercial market research, particularly in the United States, it is not the paper survey, but the telephone survey (typically the CATI survey) that the Web survey has been used to replace, and more than a few researchers have been guilty of treating the Internet survey as a kind of self-completion CATI survey.

Yet there are considerable methodological perils in migrating. For example, Smyth, Christian and Dillman (2008) have demonstrated that several very common questioning techniques, such as a multiple response question presented as a tick box list, yield very different responses from the equivalent question asked within the context of a CATI interview, when each answer will be prompted individually.

It is this writer's view that the survey software, and particularly the survey design modules of many Internet survey tools have acted to constrain survey design, and produce a certain 'sameyness' (i.e. are isomorphous) in many surveys in use today. At an institutional level, this foregrounds the importance of selecting appropriate software that will not impose constraints on survey design, or force survey designers to use inappropriate templates or 'wizards' to design their surveys. At an individual user level, it highlights the importance of both initial and ongoing training to ensure that the survey designer can make full use of the capability of the software.

There is now a considerable canon of published work identifying modal effects, or measurement errors that are characteristic of the Internet survey. It is probably fair to summarise that the Internet is no better and no worse than any other mode as a source of measurement error, but as with all other modes, there are certain characteristic measurement errors that arise, which Groves (1989) describes as modal effects.

There is not the space for this paper to summarise all that is known about modal effects characteristic of the Internet survey, and there are many others engaged in this discussion. However, I shall identify some key areas that any Internet survey system needs to address, in order to allow survey designers the freedom to counteract these known effects.

Progress bars

Progress bars rapidly became virtually ubiquitous in Internet surveys from the outset, largely in response to the feeling among researchers such as Poynter (ibid) that instructions should be clear, surveys short and the respondent treated with respect, and to some extent because they were easy for software engineers to provide. Fortunately, research carried out Conrad et al (2003) found that progress bars in surveys are indeed beneficial in reducing dropouts and encouraging the participant to stay with the survey to the end. However, they also found their effectiveness is also dependent upon the survey designer providing realistic explanation of the scale and length of the task to the participant at the start of the interview.

It is not possible to acquire a survey system without progress bars, but some refinements are often necessary. Progress bars can be compromised if they are completely automatic, as some routings on a complex questionnaire can result in uneven progress being displayed as the bar jumps forward, or creeps very slowly forward initially. Progress bars should be fully customizable with respect not just to look and feel (colours, image etc) but also to their incrementation as the survey progresses.

Scale questions

It has long been observed that there are many subtle differences between how rating scales are asked in Web surveys compared to paper self-completion and CATI interviews.

Recently, Christian, Parsons and Dillman (2009) have examined the influences of some of the many possible design permutations when presenting scalar questions, e.g. with or without graphical or textual indications, with positive items on the left or right, and so on. Influences do exist and tend to be both context dependent and subtle in their variations. They observe that that there is also considerable variability in what different Internet survey software will permit, and this can be a limiting factor. Consideration is needed in the placement of text around the screen for scales, and also in allowing scales to be presented in different ways.

It is therefore essential that the survey software should provide the survey designer with a wide range of different ways of presenting and collecting scale and rating type questions, including:

- Likert scales with one discriminating text for each scale
- Anchored (or semantic differential) scales with text on both sides
- Sliders as an alternative to radio button scales
- The ability to calibrate sliders, and show or hide the calibrations
- The ability to set the initial or default slider position

- Ability to allow scales or sliders to be tolerant or intolerant of missing values
- Ability to randomise or rotate scales, in order to prevent order effects
- Ability to retain the same randomised order whenever the same set of scales is presented again, e.g. in the context of a different question.

In essence, the survey designer should have no constraints imposed by the system over the positioning of scales on screen, on their relative size, on the positioning or size of text above, below or on either side of them.

Open-ended questions

There is often considerable variation between surveys in the frequency and length of verbatim responses typed into the box presented in an open-ended question. Smyth et al (2009) have observed that a larger open-ended box tends to result in longer answers, and rather curiously, that this is most noticeable among those that are slower to respond to the survey invitation.

Some systems determine the size of the box or fix its size automatically. Clearly, the system should allow the survey designer to control the size of the box directly, and to vary this easily from question to question, so that any influence can be controlled explicitly.

Images

Light & Oxley (2008), point out that the Internet is essentially a visual medium, and there are merits in seeking ways to ask questions in a visual way. At the simplest level this can mean adding images, and images are indeed very common in Internet surveys today.

Couper, Conrad and Tourangeau (2007) researched effect of images and their placements. They observe that any image in a Web survey forms stimulus material and therefore is going to influence the responses, however there position of the image is also highly influential. There are some positions to be avoided if the purpose of the image is to elicit a response, such as in the header area, and other placements or placement combinations, which can optimise recognition and response. It is therefore important that the software provides the survey designer with fine control over the sizing and placing of images on screen in relation to other objects such as text, buttons, progress bars etc, rather than automating these decisions.

Rich media

With the Internet being increasingly used to stream video, and for the online experience to include animations, moving images, sound and particularly spoken word, survey researchers are looking to include multimedia capabilities into their online surveys. However, controlled experiments (e.g. Fuchs and Funke 2007) have found that this approach is still fraught with technical difficulties, and that the computers, browsers and Internet connections in use by many participants are often insufficiently powerful and offer too much variability in the survey experience actually delivered to make this method viable at the time being.

Despite this, many research practitioners are seeking ways to incorporate the interactivity and rich media of today's Internet (one aspect of what is often referred to as the 'Web 2.0' environment, the other principal aspect being social media phenomena).

If the past can be used to predict the future, frustrations with speed and technical capabilities were the concern of Internet researchers a decade ago, who wisely at that time cautioned against introducing any images into surveys, or dividing a single scrolling page into individual screens (Dillman, 2000; Flatley, 2001), as these were likely to introduce too great a delay for the participant to endure. Happily, these are not concerns today, and the difficulties with rich media are also likely to ease in the next few years, and may already be acceptable in some contexts.

Survey systems need to be able to provide the survey designer with both flexibility and control over the use of rich media,

Video introduces a unique set of problems, largely connected with the very large file sizes and need for fast communication links all the way to the participant, and the participant having all of the right software and plug-ins on their computer, which it is not safe to assume they will have.

I list below the major requirements for survey software when handling video:

1. Freedom to place the video in any position on screen, and to be able to hide other 'clutter' from view.

2. Ability to embed the video within the survey instrument and stream the video directly from the web server.

3. The option to use third party streaming services (e.g. YouTube or better, subscription-paid services).

4. Automatic detection of the respondent's browser environment and available bandwidth, and the ability to determine the optimum format to use.

5. Detection that the video has played to completion and the option to prevent the participant from answering the next question until the video has completed.

6. The means to handle situations where the participant does not have enough communications bandwidth with grace, e.g. to allow for a DVD of the video clips to be sent to the participant, and for the survey to operate using local rather than streamed media.

Interactivity

Some of the interactivity that Light and Oxley were referring to, such as sortboards and discriminant game-type questions can also provide benefits in terms of cognition and ease of response. Visual types of questions can be better at eliciting a response, such as to allocate or slide items into place, rather than rely on a purely textual response (Delavande and Rohwedder 2008). Moreover, visual questions, when used appropriately, are highly effective in increasing respondent engagement and can yield better quality data (Reid et al, 2007) as a result.

Survey systems tend not to provide the means to create these questions directly, and instead, they are typically created as Java or Flash animation, which is then embedded in the survey. Clearly, if considering this approach, it is essential that the software will support embedded Flash, and be capable of collecting the data received back from the Flash animation and recording this in questions or variables within the survey instrument, so the responses are available alongside all the other data during analysis.

Some systems are now providing APIs or even Web Services interfaces, as discussed earlier, to make it easier for survey designers to integrate any Java or Flash animations into their surveys, or even to access libraries of read-made animations as plug-ins.

Respondent fraud, satisficing and response quality

Internet fraud has been much discussed within the field of commercial market research, whereas in the field of social statistics, the equivalent discussion has been around detecting poor data quality from misleading responses provided by participants (Baker and Downes-Le Guin 2007). Fraud is most prevalent where any kind of incentive is offered, which is the norm in much commercial survey research, or where panels are involved. Although it now would appear that the level of fraud is much less than at first was feared (Cape 2008, Bain 2009), nevertheless, even surveys by public authorities, social research institutes and national statistical offices are not immune from some of these issues.

Internet survey researchers need to be concerned about the tendency for online respondents to find the easiest route through a survey instrument – a phenomenon often called 'satisficing' (Holbrook, Green and Krosnick 2003). In the context of an Internet Survey, this can lead to more respondents selecting a don't know answer, when provided, and failing to provide accurate responses to grids or scale questions (Heerwegh and Loosveldt 2008). Satisficing behaviour is prone to arise during longer surveys, as observed by Galesic and Bosnjak (2009).

Fraud and error detection capabilities can help to identify speed cheats, multiple survey takers, those outside the country who should be ineligible, 'satisficers', 'straightliners' and inattentive respondents who answer the questions without thinking, or constantly pick an easy answer.

A range of software-based detection methods are available, and the principal methods and applications are discussed in Table 1. It is worth remembering that, at the simplest level, fraudulent or poor-quality cases can be identified and eliminated by analyzing the data. However, this may then mean the overall sample size is drastically reduced, and may require additional interviewing to compensate. Commercial researchers have therefore become interested in being able to detect suspicious behaviour from participants, such as straightlining through grids, speeding through questions or other forms of satisficing, as it occurs.

Behaviour	*Observed by*	*Methods*
Speeding (fast survey takers)	Participants answer questions too quickly, rushing through without allowing sufficient time to consider each question	Measure the time taken to respond to all questions or set a speed trap at key questions; compare time taken with expected normal time; challenge respondent or terminate interview
Inattentiveness	Participant is not paying sufficient attention to answer questions accurately	Inconsistent answers – can set traps to look for contradictory answers to questions.
Straightlining	With batteries of grid or scale questions, the survey taker takes an easy route through the questions by selecting the same answer each time, or possibly a simple pattern (e.g. a diagonal line)	Logic checks on common patterns resulting from this behaviour.
Hyperactive survey takers	The same PC is used multiple times in a survey or across surveys. This is a particular concern with members of panels who may have registered many times with different identities	Digital fingerprinting – several vendors offer methods to identify a PC from a range of unique characteristics provided as a part of the normal HTTP communication between server and browser
Location fraudsters	Participant is not resident in the country of study and should be ineligible.	Validation of the IP address of the survey taker

Table 1 Software-based fraud and error detection methods

There are now some third-party or plug-in applications in the market that allow survey designers to refer completed interview records to a checking service, which will then engage a number of methods including digital fingerprinting, locational IP checking, speed-based heuristics, and assign a probabilistic 'cheat score' to the case. Using a service like this has the advantage that each survey institute will benefit from other recorded attempts by participants in other surveys from other research organizations. This is especially beneficial in identifying hyperactive survey takers through digital fingerprinting of their PC, as otherwise, each research organisation can only be aware of a survey taker's over-activity in respect of their own survey invitations.

Mixed mode interviewing

Many have argued, though few more persuasively than Blyth (2007), that a mixed mode approach is already (or if not, will soon be) the only means by which representative samples can be achieved. From the software perspective, this immediately causes a dilemma in choosing or designing a system. Only a minority of Web survey packages support other modes, and very few support all four major interviewing modes, namely telephone (CATI), face-to-face (CAPI), paper and Web.

A great many survey organisations conducting mixed mode research do so using different software for each mode. However, this is operationally inconvenient and it also increases the cost of running the survey considerably, as the survey instrument has to be programmed and tested on each different system. It is far more cost effective and efficient if the same system can be used (Rippen 2003), and response quality can be improved if the switch from one mode to another can be performed instantly.

I undertook a research project in 2003 (Macer, 2003A) to examine the benefits of using an integrated system for mixed mode research, and at the same time, examined what specific mixed-mode capabilities a roster of ten different survey providers had built into their systems. From that research, I identified twelve key requirements for any mixed mode interviewing system. Since publication, these have been examined and adopted by several international survey software developers.

Today, I consider the first eleven of these remain important, but I have substituted a new requirement for the twelfth item. These requirements are:

1. A common survey authoring tool that generates a single survey instrument for all modes.

2. Independence between design and execution, with mode-specific templates and rules.

3. The ability to define mode-specific texts in addition to foreign language alternatives

4. A single, consolidated database for all survey data, updated in real time.

5. Ability to determine the mode of initial contact from the sample subject's stated preference.

6. Efficient switching between modes, initiated by the script or by the respondent.

7. Ability to conceal all interviewer-recorded data when switching to self-completion modes.

8. Support for reminders and fall-back strategies to revert to a prior mode if still incomplete.

9. Single view management and reporting, which identifies response by mode.

10. Quota controls implemented across all modes in real time.

11. Question constructs for mixed mode, e.g. unprompted questions for CAWI, and the ability to have mode specific answer categories (e.g. "don't know and "not stated") present or absent in different modes, as required.

12. The ability to perform a warm handover from CATI to web, so that the participant can confirm they are able to start the Web interview while the interviewer remains in contact with the participant.

The original twelfth item was 'Recording of mode applied, at a datum level not just a case level.' However, I have learned from practitioners and software providers that this is considered unnecessary by many, and increases data size unnecessarily for the few occasions where it is required. Furthermore, the information can often be reconstructed from activity records, where they exist.

I have replaced this with support for a so-called warm handover, because a number of researchers have reported to me subsequently that there can be a very high level of dropout when handing over from telephone to web. Some of this can be accounted for by limited motivation by the participant to continue, and their agreement on the telephone actually concealing what is a polite refusal. However, another reason for dropout is that the interviewer does not record the email address accurately, when asked of the participant (or the participant does not state it accurately) and therefore the survey invitation fails to reach the participant. A warm handover, while the interviewer and respondent remain in telephone contact, can overcome this and also provide a level of encouragement from the interviewer that seems to reinforce the motivation to continue. For whatever reason, a warm handover is a very useful device in boosting the response rate.

Deployment

As can be seen from the foregoing discussion, the needs of the Internet researcher are complex and highly advanced. Not only are there the operational and efficiency considerations to take into account, but the exactitudes of survey methodology and controlling survey error too. Furthermore, these requirements can vary considerably over time, and new requirements can emerge rapidly when a new research project is being commissioned.

It is a major constraint on the research institute if the demands of new surveys require re-engineering and programming of the software in order to accommodate legitimate research requests. Most high quality, well-provisioned survey software packages will offer the flexibility to accommodate virtually any survey instrument or interviewing technique, and allow for the researcher to make adjustments in order to minimise known interview effects and sources of survey error, as discussed earlier.

Furthermore, although the features may be untried within that research institute, they will most likely have been tested and used extensively by other researchers elsewhere.

There is a surprisingly broad range of software readily available for purchase. Software directories such as those maintained by Quirk's (www.quirks.com), the Association for Survey Computing (www.asc.org.uk) or meaning ltd (www.meaning.uk.com) each list over a hundred Internet survey solutions at varying levels of cost and complexity.

The difficult matter of cost

For any research institute, selecting and deploying a new survey system (or a replacement system) is a major capital investment and a case will need to be made. Cost is always a major factor when considering which software to deploy, and arriving at an accurate comparison of cost can be difficult when different vendors use widely differing cost models. Some will charge a one-off purchase fee, followed by a smaller annual maintenance fee; others charge a monthly or annual subscription; others charge solely on the number of interviews conducted. In addition, most models include a sliding scale of charges according to how many users there are, how many surveys exist, how many interview are collected and so on. Just as each research institute's own set of requirements for their software will be different, so too is the cost equation – and it is largely unpredictable.

However, the cost of the software licences and hardware purchases must be considered in the context of the total cost of ownership, for which the following questions may be useful to consider:

How much will the system cost to run?

For example, if the software is deployed on your own servers, there may e additional internal charges or procurement costs from the central IT department for hardware, communications and maintenance, some of which will be recurring costs. Another important factor is whether upgrades to the core system are included in the cost, or whether these need to be purchased separately in the future.

How do costs compare over time?

It is reasonable to expect a software system to have a lifetime of several years, and it is often realistic to model the costs over five years, though, in reality, the software is likely to be used for much longer than that. Systems that may appear expensive, when acquisition costs are largely incurred in the first year, can emerge as being much less expensive when compared to systems where there are annually recurring charges.

How will improvements in efficiency translate into actual cost reduction?

The kinds of operational efficiencies offered by some systems, for example, by not requiring a skilled programmer to set up a project, or through automating what were previously labour-intensive tasks can deliver tangible savings – and these will vary from one system to another.

How will the costs vary if the level of work changes?

The costs that a software provider charges are usually related to the volume of work or size of the operation in some way, and it is important to understand the effect of increased (or even decreased) work on the costs payable to them.

The desire for custom built software

With such an abundance of choice, and particularly with the level of sophistication offered by the more 'professional' tools, I consider there is little justification in creating a custom built system, and conversely, a high level of risk.

Despite this, a surprisingly large number of commercial research companies appear to develop their own survey software. In an international survey of market research companies carried out by meaning ltd (Wilson and Macer, 2009), 25% of companies reported they used Web survey software they had developed themselves. There is clearly strong motivation among survey research organizations to develop their own Internet survey software, despite the effort involved.

With such an abundance of choice, and particularly with the level of sophistication offered by the more 'professional' tools, there is almost no justification for developing yet another tool to administer surveys online. The risks in developing an in-house solution are:

- that the requirements are not all anticipated by the research team or the software designers, so the system does not fully support the full range of survey applications or questionnaire complexity that it may need to cover;
- the developed system is not fully tested and contains undetected programming errors that affect the resulting data;
- that the system is difficult to enhance, and becomes out-of-date as Internet technology and research techniques advance;
- the knowledge about the system is lost to the organisation when one or more individuals are no longer available to work on the software.

From a methodological perspective, it also means that there is a very real risk that undetected errors in the software will give rise to survey error. It is an unfortunate fact of software engineering that changes to software are very likely to introduce new software errors, and in the case of a live survey, this can have a catastrophic effect on the data, if the error goes undetected. It is difficult to imagine that an in-house team or a busy research department will have the time or resources to devote to testing each update to the software to the level required to isolate programming errors, or even in a comparable way to a high quality software producer. A reputable software developer will have expertise and dedicated resources for software quality control.

I have encountered several situations where a request from a survey team or research department to purchase specialist Web survey software has been met by a response from the in-house IT or Web development team that they could provide such a tool easily and inexpensively either by developing it themselves, or building it into an existing development project. This situation typically arises because the in-house team is only concentrating on one small aspect of the process: the creation of the pages of the online survey instrument, and because all of the other related processes and complexities discussed in this paper simply have not been considered.

Conclusions

In this paper, I have set out five areas of exploration, in considering the needs of the Internet researcher to ensure that the software being utilised is truly fit for purpose.

In *Operational and efficiency considerations*, I described the typical modular structure of a Web survey system. The separation of design, deployment, execution and management is beneficial in that it recognises the different roles and responsibilities that a large research team will have for different parts of the research process.

In *Advanced system architecture,* I described some of the recent advances and how these can be beneficial to Internet research:

- Web based software for design and management that can simplify the process, reduce the cost of ownership and facilitate collaboration by different team members on a project.

- Database technology that can make survey systems more scalable, particularly when dealing with very large volumes of data, and also survey software to connect and reuse data and other components across surveys, link together panel data, and so on; all of which is difficult to do in systems where there is a rigid separation between surveys.

- APIs and Web Services interfaces that can make it easier to extend the capabilities of the survey software, and allow plug-ins to be written, or third party plugs-ins to be used very easily.

In *methodological considerations*, I addressed some of the more important identified sources of measurement and nonresponse error in Internet surveys, and suggested ways in which Internet survey software can allow researchers to produce reliable survey instruments that will minimise survey error, as well as address concerns of respondent burden and engagement. These included having great flexibility in the placement and presentation of question wording, images, progress bars and the size of verbatim boxes. We also considered some of the measures possible to combat poor survey quality due to fraud, inattentiveness and so on though a combination of careful design and also measures that can be used to detect unsatisfactory or inconsistent data.

In *mixed mode interviewing* I described the twelve key criteria I had earlier developed for mixed mode research, which have been adopted by several specialist software suppliers.

Finally, in Deployment, I set out a case that survey researchers may find useful in justifying any investment in specialist survey software for Internet or mixed mode research, and explained the risks of developing software in-house, from both a technical and methodological standpoint.

It is typical for software decisions within organizations to be based upon important considerations such as efficiency, maintainability and the use of components or protocols and that are compatible with a organization's wider IT standards. Survey methodology and the system's effectiveness in allowing researchers to control survey error is less frequently discussed. With Internet research, software and methodology are inextricably connected, and it is vital that this becomes a regular part of the dialogue between survey designers and software developers.

About the author

Tim Macer is the founder of meaning ltd, an international research technology consultancy based in London, England, and has worked for over 26 years in the field of information technology for marketing and opinion research. In 2006, he was appointed Visiting Senior Fellow at the University of Southampton in the field of market research.

Tim is widely published as a specialist writer exploring the application of technology to research. In 2008, with Dr David F. Birks, he jointly edited "Marketing Research: Critical Perspectives", a new four-volume anthology of definitive essays and papers on the subject, published by Routledge.

References

Baker, R. and T. Downes-Le Guin. 2007. "Separating the Wheat from the Chaff: ensuring data quality in Internet samples", in *The Challenges of a Changing World: Proceedings of the Fifth International Conference of the Association for Survey Computing*, eds. M. Trotman et al., pp.157–165. Berkeley, UK: ASC.

Banks, R., J. Currall, J. Francis, L. Gerrard, R. Khan, T. I. Macer, M. Rigg. E. S. Ross, S. Taylor, A. J. Westlake, 2003. *Survey and Statistical Computing IV: The impact of technology on the survey process*, Chesham, UK: ASC.

Bain, R. 2009. Multiple panel membership 'doesn't affect quality', Research Live [online], London: MRS [Accessed on 14 August 2009]. Available from: http://www.research-live.com/news/4000195.article

Booth, D., H. Haas, F. McCabe, 2004. Web Services Architecture: W3C Working Group Note, World Wide Web Consortium [Accessed 14 August 2009]. Available from: http://www.w3.org/TR/ws-arch/#id2260892

Cape, P. 2008. "Multiple Panel Members: saints or sinners", Conference Notes, ASC Conference: "Have we lost touch with reality: re-engaging with the respondent", *International Journal of Market Research* **50**(5), pp.702-704.

Christian, L. M., N. L. Parsons, and D. A Dillman, 2009. 'Designing Scalar Questions for Web Surveys", *Sociological Methods Research* **37**(3), pp.393-425

Conrad, F. G., M. P. Couper, R. Tourangeau, A. Peytchev. 2003**.** "Effectiveness of Progess Indicators in Web Surveys: It's what up front that counts", in *Survey and Statistical Computing IV: The impact of technology on the survey process*, eds R. Banks et al., pp.333-342. Chesham, UK: ASC.

Couper, M. P. 2000. "Web surveys: a review of issues and approaches", *Public Opinion Quarterly* **64**(4), pp.464-494.

Couper, M. P., F. G. Conrad and R. Tourangeau. 2007, "Visual context effects in Web Surveys", *Public Opinion Quarterly* **71**(4), pp.623-634.

Couper, M. P. 2007. "Whither the Web: Web 2.0 and the Changing World of Web Surveys", in *The Challenges of a Changing World: Proceedings of the Fifth International Conference of the Association for Survey Computing*, eds. M. Trotman et al., pp.7–16. Berkeley, UK: ASC.

Dillman, D. 2000. Mail and Internet Surveys: the tailored design method, New York: John Wiley.

ESOMAR. 2008. Global Market Research: ESOMAR industry report, Amsterdam: ESOMAR.

Flatley, J. 2001. "The Internet as a mode of data collection in government social surveys: issues and investigations", paper presented at the ASC International Conference on Survey Research Methods, Chesham, UK.

Delavande, A., and S. Rohwedder. 2008. "Eliciting subjective probabilities in Internet surveys", *Public Opinion Quarterly* **72**(5), pp.866-891.

Fuchs, M., and F. Funke. 2007. "Video Web survey: results of an experimental comparison with a text-based Web survey", in *The Challenges of a Changing World: Proceedings of the Fifth International Conference of the Association for Survey Computing*, eds. M. Trotman et al., pp.63–80. Berkeley, UK: ASC.

Galesic M., and M. Bosnjak. 2009. "Effects of Questionnaire Length on Participation and Indicators of Response Quality in a Web Survey", *Public Opinion Quarterly* **73**(2), pp.349-360.

Groves, R. M. 1989. *Survey Errors and Survey Costs.* New York: Wiley.

Heerwegh, D., and G. Loosveldt. 2008. "Face-to-face versus web surveying in a high-Internet-coverage population: differences in response quality", *Public Opinion Quarterly* **72**(5), pp.836-846

Holbrook, A. L., M. C. Green, and J. A. Krosnick. 2003. "Telephone versus face-to-face interviewing of national probability samples with long questionnaires. comparisons of respondent satisficing and social desirability response bias." *Public Opinion Quarterly* **67**(1), pp.79–125.

Hughes, K., S. Jenkins, and G. Wright. 2000. Triple-S XML: A standard within a standard, Social Science Computer Review, **18**(4), pp.421-433.

Krosnick, J. A., A. L. Holbrook, M. K. Berent, Richard T. Carson, W. M. Hanemann, R. J. Kopp, , R. C. Mitchell, S. Presser, P. A. Ruud, V. K. Smith, W. R. Moody, M. C. Green, and M. Conaway. 2002. "The impact of 'no opinion' response options on data quality: non-attitude reduction or an invitation to satisfice?" *Public Opinion Quarterly* **66**(3), pp.371-403.

Light, B., and M. Oxley. 2008. "Unlocking the real potential of web-based market research", *Admap* **494** (May): pp.24-26.

Macer, T. I. 2003 A. "We seek them here, we seek them there: how technical innovation in mixed mode survey software is responding to the challenge of finding elusive respondents", in *Survey and Statistical Computing IV: The impact of technology on the survey process*, eds R. Banks et al., pp.231-242. Chesham, UK: ASC.

Macer, T. I. 2003 B *Research Software Review*, London: The Market Research Society.

Macer, T. I., M. Pearson, and F. Sebastiani. 2007. "Cracking the code: what customers say in their own words", paper presented at The MRS Golden Jubilee Conference, Market Research Society, Brighton, UK.

Oosterveld, P., and P. Willems. 2003. "Two modalities, one answer? Combining Internet and CATI surveys effectively in market research", paper presented at ESOMAR Technovate conference, Cannes, France.

Poynter, R. 2001. "A Guide to Best Practice in Online Quantitative Research" in *The Challenge of the Internet*, ed. A Westlake et al, Chesham, UK: ASC.

Reid J., M. Morden, and A. Reid. 2007. "Maximizing respondent engagement: the use of rich media", paper presented to ESOMAR Annual Congress, Berlin, Germany.

Rippen, H. 2003. One project, three methods, in *Research Software Review*, ed. Macer, T. I., London: The Market Research Society.

Smyth J. D., Christian, L. M. and Dillman, D. A. 2008. "Does 'Yes' or 'No' on the telephone mean the same as 'check-all-that-apply' on the Web?" *Public Opinion Quarterly* 72 (1): 103-113.

Smyth J. D., D. A. Dillman, L. M. Christian, and M. Mcbride. 2009. "Open-Ended Questions in Web Surveys: Can Increasing the Size of Answer Boxes and Providing Extra Verbal Instructions Improve Response Quality?", *Public Opinion Quarterly* **73** pp.325-337.

Taylor, H. 2000. "Does Internet Research Work? Comparing online survey results with telephone survey", *International Journal of Market Research* **42**(1) pp.51-63.

Terhanian, G. 2003. "The unfulfilled promise of Internet research", paper presented at Research 2003, The Market Research Society Annual Conference, Birmingham, UK.

Trotman M., T. Burrell, L. Gerrard, K. Anderton, G. Basi, M.P. Couper, K. Morris, D.F. Birks, A.J. Johnson, R. Baker, M. Rigg, S. Taylor, A.J. Westlake 2007, *The Challenges of a Changing World: Proceedings of the Fifth International Conference of the Association for Survey Computing*, 7–16. Berkeley, UK: ASC.

Westlake, A. J., R. Banks, J. Francis, A. Manners, M. Rigg, and W. Sykes 2001. *The Challenge of the Internet*, Chesham, UK: ASC.

Wilson S. C., and T. I. Macer. 2009. "A report on the Confirmit Market Research Software Survey" in *Quirk's Marketing Research Review*, May: 74.