Challenges of Developing and Supporting Multimode Survey Instruments

Leonard Hart, Scott Reid, and Erin Slyne, Mathematica Policy Research

Over the past decade, Statistics Netherlands' Blaise survey software has evolved into the primary data collection system used at Mathematica Policy Research because of its adaptability in meeting our multimode survey instrument needs. Although computer-assisted telephone interviewing (CATI) and computer-assisted data entry (CADE), also known as high-speed data entry, were the predominant modes of computerized data collection early in the past decade, the introduction of computer-assisted web interviewing (CAWI) and computer-assisted personal interviewing (CAPI) increased our need to find ways to best integrate and leverage the advantages provided by using these other very distinct data collection modes.

This paper reviews our experiences with multimode survey data collection and the challenges inherent in working with instruments in single and multiple data collection software environments. We will discuss supporting multiple web-based data collection systems in an ever-changing internet environment and the effects they have on data collection. We will also describe the obstacles and solutions involved in coding and maintaining a single Blaise instrument that handles multiple modes of data collection in one central real-time Blaise database.

1. Background

1.1 Experience with multimode surveys

Over the years, Mathematica has conducted a growing number of multimode surveys. Currently, we believe we are one of the few in our industry successfully deploying surveys that use one Blaise instrument accessing in real-time one centralized Blaise database accepting data across distinct data collection modes. Most of these surveys combine CATI and CAWI in real time; on a couple of occasions, we have included CADE into the mix. We have also incorporated CAPI data collection into a shared centralized database, but not on a real-time basis, although that is a future goal of ours. Getting to the point of implementing single-instrument/multimode surveys did not happen overnight; rather, it developed over a few years.

Paper-and-pencil methods of data collection have been around since before Mathematica's inception and are still used today to fulfill the needs of simple self-reporting surveys or quick in-person observations. In spite of all the technological advances we have seen since our founding—namely, the proliferation of personal computing; the ability to easily share data via internal and, later, external networks; and the explosive growth of the internet—the paper survey still has its place as a valuable data collection tool.

All the advantages of the computer-assisted interviewing (CAI) systems and software developed over the years, which enable our industry to conduct computer-assisted telephone, in-person, and internet-based surveys efficiently, would go beyond the scope of this paper. From the outset, however, many organizations, including Mathematica, recognized the importance of incorporating them into a data collection arsenal. When properly applied, CAI data collection systems that implement CATI, CAPI, or CAWI instruments allow for greater accuracy of survey data, help speed up the entire data collection process, reduce the burden on our respondents, and can lower overall data collection costs. As these CAI systems and software became more robust, our capabilities to apply them across multiple modes also increased. No longer were we tied to just one method of data

collection. Having the capability to reach respondents in multiple ways enabled us to achieve high response rates for our surveys, even as it has become increasingly difficult to contact respondents in recent years using traditional methods.

CATI was used initially in a stand-alone mode for conducting instruments where utilizing the capabilities of a CATI system to handle complex questionnaire logic and systematically handling and delivering cases to respondents via a call scheduler were advantageous over the paper-and-pencil methods of having an interviewer try to calculate complex logic on the fly or handling thousands of paper contact sheets. CATI later became a tool used to supplement mail or in-person paper surveys, by enabling us to follow up quickly with respondents who failed to self-report within the time allotted for data collection.

CAPI's introduction was very similar to that of CATI's; it was initially used in a stand-alone mode because of the computer assisted advantages it provided in handling complex logic and case delivery over manual paper processes. Later CAPI was used to follow up on self-reported paper surveys or CATI. A knock on the door and the ability of a field interviewer to collect the survey data easily via a CAPI instrument and then be able to transmit that data back to a central office works extremely well when dealing with a universe of respondents who could be difficult to reach by telephone.

As the internet became a more widely accepted communication tool in society, CAWI began to grow as a valuable data collection method. CAWI was first used at Mathematica in a stand-alone mode on surveys in which the entire sampling universe had the capability to participate via the internet. It then grew into a multimode option used to supplement other data collection efforts. In multimode data collection, CAWI was first used as an alternative method of gaining respondent cooperation. For example, as a final follow-up to a CATI data collection, respondents were told they could self-report their data via an internet-based instrument and would not be burdened with a potentially lengthy telephone interview or the inconvenience of scheduling that interview. Over the years, this trend has reversed, because nearly all of our multimode surveys that use CAWI start data collection in this mode. CAWI is often the first option available to a respondent, although CATI is used only as a follow-up. As we have seen on several of our annual or longitudinal surveys, these methods have grown increasingly popular with our respondents; for these studies, CAWI is becoming a dominant mode of data collection. Another potential benefit of offering respondents a CAWI option is in the cost savings that projects see when you decrease the number of mailings, in-person visits, or telephone contacts you have to make to gain the cooperation of the respondent.

Over the years, we have conducted many multimode surveys. Surveys of which we are most proud include the 2003 National Survey of Recent College Graduates (NSRCG), which was the first real-time, one Blaise instrument with a centralized Blaise database, CATI/CAWI/CADE survey we conducted; the 2008 NSRCG, which was the first survey in which we collected CATI/CAWI and CADE data via one Blaise instrument and a central database using Blaise IS; the National Survey of Substance Abuse Treatment Services (N-SSATS), in which we have collected data using CATI, CADE, and CAWI instruments for the past ten data collection cycles and have seen the internet-based mode grow to be the predominant mode chosen by our respondents; and the Ewing Marion Kauffman Foundation's Kauffman Firm Survey, which is now entering its fifth follow-up round and has been able to track a cohort of new businesses successfully via both CATI and CAWI data collection.

1.2 CAI software used

Mathematica has used several CAI systems over the years to support our data collection efforts. We currently use Blaise for our CATI and CAPI studies, Viking Software Solutions Viking Data Entry system for CADE, and different products for web surveys.

We have tried using Blaise for CADE on a several surveys and had some success. However, due to the additional labor involved in programming for the specific requirements of a data entry instrument, while also making it work with the needs of a CATI, CAPI, or CAWI instrument, as well as our client's needs to have data double key verified, we have found it is easier and more cost-efficient to use the Viking system to meet our high-speed data entry needs.

For CAWI studies, we use three different systems, based on the capabilities of each package and how it best addresses particular survey requirements.

Blaise IS is the CAWI system we rely on for the studies in which we determine using one instrument, with one real-time centralized database supporting multiple modes (usually CAWI and CATI) with extremely similar instrument specifications, works best for our data collection needs.

ObjectPlanet, Inc.'s Opinio system is used for internet-based instruments that are short (in number of questions presented and the overall time to complete the survey) and simple in content (for example, simple skip patterns, simple question designs, and no grids or complex tables). A nonprogrammer can easily set up the screens and basic skips available in this system, which makes it cost-effective for very straightforward web surveys.

Mathematica's homegrown WebSurv system, which is an ASP.NET front-end/SQL back-end application, is used for CAWI studies in which we have an extremely specialized data collection need that Opinio or Blaise IS cannot easily meet. Using WebSurv can occasionally give us increased web page design flexibility over our other systems. Another advantage is that most of the instrument set up can be done by junior-level staff from outside the Information Systems department.

2. Support Issues

2.1 Web dominates

Based on our extensive experience supporting different modes of data collection, including CATI, CAPI, CADE, and CAWI, we have found that CAWI requires the most user and programming support. Much of this has to do with CAWI being the "new kid on the block" and with the learning curve for both programmers and respondents in dealing with a new and ever changing technology.

For this reason, we are concentrating on CAWI, but we will also discuss how other modes might be affected. CAWI support is different from the other modes of data collection because we cannot control the respondent environment as we can for an internal user's environment. As with other modes, CAWI surveys will have traditional support issues (such as logic errors or internal system issues). However, CAWI surveys open up a whole new area that we cannot control: the user desktop. We discuss some of these CAWI-specific issues next.

2.2 Authentication processes

Making sure the correct respondent accesses your CAWI instrument properly is as important as the data collection efforts of the instrument itself. Respondents must feel confident of the security of their data being transmitted, especially when it is sensitive or personally identifiable. How a respondent views the capabilities of your authentication process is the equivalent of making a guest feel welcome when entering your home. Because we use three CAWI systems, getting them all to meet our security needs, as well as representing Mathematica web systems as a "welcome place" to participate in a survey, can be a challenge.

Although Blaise out of the box offers some basic authentication solutions, we felt we needed increased security, the ability to integrate the authentication process into other systems, and greater customization than it offered. Therefore, we decided to utilize Full Content Protection (FCP) using ASP.NET. FCP forces a user to be validated when the user requests any content from the web application. Because the FCP authentication process we use for Blaise instruments closely replicates the process used by our WebSurv system and is also written as an ASP.NET application, we have a greater pool of programmers available who are familiar with ASP.NET applications. Because of the similarities of the two systems, we were also able to standardize the feel or the authentication process across both and build common standards for our login and password combinations.

Opinio's authentication process is proprietary and less customizable than FCP for Blaise or WebSurv. User authentication is optional with Opinio, as you can create surveys available to the general public; however, this is not an option we would choose to implement. You can assign user names and passwords to particular respondents. However, because of the proprietary nature of Opinio, it requires staff to learn a second set of requirements for creating and maintaining a study's list of web respondents. We investigated using the FCP authentication piece we use for Blaise CAWI instruments, but found it to be much more cost-effective to use the authentication piece provided by Opinio itself. Ideally, we would find it most beneficial if we could come up with one authentication system that we could use across all three of our products.

2.3 Multiple types of web browsers

According to the Net Applications Browser Market Share report for the period ending July 2010, the top five types of browsers in use worldwide are Microsoft's Internet Explorer (60.74 percent), Mozilla's Firefox (22.91 percent), Google's Chrome (7.16 percent), Apple's Safari (5.09 percent), and Opera Software's Opera Browser (2.45 percent) (http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0). This list covers 98.35 percent of all the browsers in use, but that still leaves 1.65 percent of the internet population using alternate, unsupported, or even obsolete browsers. Factor into that the number of versions of the top five browsers available and you can see that it would be nearly impossible to provide a comprehensive list of browsers potential respondents could use on one of our CAWI surveys.

For the most part, our three CAWI tools have been able to handle the preponderance of the browsers in use; however, as they say about the stock market, "Past performance is no indication of future returns." When we first started supporting CAWI instruments, we did come across situations in which a question would display one way in one browser and a different way in the others. However, as CAWI use grew and the tools available became more robust, these differences have become less problematic.

We are seeing an increase in the number of respondents using mobile or handheld devices, and there is a growing concern that our current web design standards might have to be adjusted further as

this segment of the browser marketplace increases. The Net Applications Browser Market Share report for July 2010 shows that handheld or mobile browsers make up 2.2 percent of the market, up from 0.98 percent in October 2009 (http://www.netmarketshare.com/report.aspx?qprid=61).

The internet is an ever-changing environment, and our mantra for the past decade when working with CAWI instruments has been "Keep it simple." Although it is possible to do many exciting and flashy things within an internet instrument, you have to make sure that almost any respondent can properly view the page you are presenting and that you are not introducing any bias into your questionnaire design because of browser behaviors.

2.4 Synchronization and merging of data between disparate systems

Getting disparate data collection tools to work together smoothly for a single questionnaire can be a challenge. The data is more than likely stored in either a proprietary format or database, and the data variables and values probably will be completely different for each because of the unavoidable idiosyncrasies of each system.

The first challenge we encountered when using disparate systems for the same instrument was trying to prevent duplication of data between the different modes of data collection. Although you would hope that the data between the systems would be identical, we found that is not always the case, especially on institutional or establishment surveys in which two people could be reporting for the same entity. For example, in the early rounds of data collection for a facilities study, a particular facility might have completed a mail survey that was marked as received in our sample management system (SMS) before being passed to data entry staff to input into the Viking data entry system the same day a telephone interviewer contacted the facility to complete the survey via the Blaise CATI instrument. Meanwhile, someone else at the same facility might have logged into the WebSurv CAWI instrument the same day and completed the survey there. Because there were differences in the data collected in each instrument, additional calls to the facility would have been needed to determine which mode of data collected was the one we wanted to keep.

One way we cut down on the discrepancies between the systems was to increase the number of real-time updates between the systems being implemented. For example, as soon as a mail survey was marked as received in our SMS database, the login for the web survey was disabled in the WebSurv system and the case status there was adjusted accordingly. For CATI, if the case was delivered by the Blaise call scheduler, or if selected for manual dialing by a CATI interviewer, a call was made to the SMS database which pulled the status into Blaise and let the interviewer know that a call was not necessary and upon closing the case it would be then statused as mail received in Blaise and no further dialing attempts would be necessary.

Because WebSurv and the SMS data are both SQL databases, the triggers and stored procedures used to make the updates between those systems were relatively straightforward; however, we had to develop a process to share data between the Blaise and SMS database, which in turn would update the WebSurv database. We developed a Dynamic-link library to use as an interface. The Blaise instrument (or the SMS database) can use this interface to pass data between these systems.

The other challenge we encountered involved merging and processing the survey data between these systems. Because each system collects and outputs (or allows you to access) its data in its own format, there needed to be a way to "crosswalk" the data into one common format. The biggest problem is that any mistake in the crosswalking leads to errors in your final data and the eventual analysis. When dealing with disparate systems, we have tried as much as possible to name the

variables the same in each system, make sure we ask the questions in the instruments in the same format, and make the answer categories or data types the same. By keeping the modes as common as possible, there is less chance of the data not lining up when combined via SAS processes.

Because of these challenges, we are strongly considering implementing Datalink on a future project, so all the survey-related data will be stored in a common SQL format, and the communication and updates between these systems should be handled easily. More important, this is why we try to use one Blaise instrument with one central real-time database when we have a multimode need.

2.5 Around-the-clock support

Multimode surveys have support needs on varying levels and timelines. For the most part, pencil and paper and the software to enter the data recorded on hard-copy instruments are the easiest to support technically and have least urgency when requests come to programmers, as there is no respondent waiting for a solution. Supporting CATI and CAPI is a little more challenging than CADE, because you have the interaction with your respondents to contend with; however, having the interviewer as part of the process can reduce the urgency when problems do arise. The CATI or CAPI field interviewer can easily explain to the respondent the issue we might be encountering and negotiate a time when we can continue the survey.

When it comes to CAWI, the level of support needed and the urgency factor increase. The CAWI instrument itself is the main point of communication between you as an organization and the respondent. Because the internet never sleeps, keeping CAWI respondents happy becomes a 24 hour a day, seven day a week, 365 (or 366) day a year proposition. Because problems with CAWI surveys can pop up anywhere, they can be the hardest to diagnose and fix. For example, if a respondent cannot access your survey, the problem could be anything from an Internet Service Provider (ISP) issue, the connection to our web servers having gone down, the web server itself having failed, an internal network issue, or a programming error putting the survey into an unstable state.

When CAWI was a data collection tool newly available to Mathematica, we encountered support issues relating to browsers, internet connections, software, and user errors; compounding these issues was that this was a new technology for end users and for us. Over time, survey support staff have gained more experience supporting CAWI instruments, and they now triage calls and emails. Before, programmers had to become involved in every CAWI issue. Now, survey staff can check that a respondent's browser is functioning properly or that the respondent can reach other sites besides our survey's site, which helps determine if the problem is on the respondent's end or ours. In addition, respondents are becoming more internet savvy and can address items such as connectivity. Also, web survey packages have gotten better at addressing some issues, and we have built new features into our supporting tools to address certain inefficiencies.

2.6 CAWI instrument troubleshooting

No matter how much instrument testing you do, occasionally a programming bug will get into the real world and create problems for your survey. One of the biggest support challenges is trying to determine what exactly a CAWI respondent is doing when that problem occurred.

Unlike a pencil-and-paper survey, in which the respondents will occasionally leave comments in the margins if a question is not clear, or a CATI or CAPI survey, in which a programmer can extract an accurate description of the problem from the interviewer, communicating with CAWI respondents is much more difficult. They may choose to tell us nothing and just complain about not

being able to complete the survey, or, if you are lucky enough to secure their cooperation in triaging the issue, they might not remember or have recorded exactly what they were doing when the instrument problem occurred. However, there is hope in trying to determine what CAWI respondents are doing when problems arise and information is limited. Occasionally, you can glean information from some of the web server logs, but one of the best tools available to the CAWI instrument programmer is a web audit trail.

With Opinio surveys, tracking respondent movements and issues that arise from those movements is less of a problem, as the logic in these surveys is less complex than what is found in our homegrown WebSurv system or in Blaise Internet instruments. In WebSurv, it is possible to trace the path a respondent takes through the survey. The system tracks the web page from which the respondent submitted data and exactly when the respondent submitted that data. The system also determines if the respondent changed the data initially submitted by backing it up on his or her own or if the instrument itself put the respondent at an incorrect point in the survey.

Trying to implement CAWI Blaise audit trails has been a hit-or-miss proposition. For the initial Blaise surveys we hosted, we used a third-party piece of software called C2B ("CentERdata to Blaise"), developed by CentERdata at the University of Tilburg. C2B provided the interface between PHP (an open-source server-side scripting language) and our Blaise instruments. As part of the C2B system, we could capture detailed audit trails that showed us the movement and data changes respondents made. This proved to be a valuable tool in finding the programming equivalent of a needle in a hay stack when trying to resolve programming bugs. When we switched from C2B to Blaise Internet several years ago, we hoped the journaling feature built into Blaise would provide us with the same level of detail we had with C2B. However, we have been unable to replicate that success using the Blaise IS journal without customizing it greatly for each project. We hope this is something that Statistics Netherlands can improve upon in future versions of Blaise, as it is an important tool to have at our disposal.

3. Blaise Single Instrument for Multiple Modes

Using a single Blaise instrument for surveys conducted for multiple modes has advantages and disadvantages. Perhaps the biggest advantage is the ability to maintain the data in one centralized database. Because there is no need for special programming to synchronize different databases, data management is easier to handle.

The major drawback of this type of single-instrument design comes from the complexity of our surveys, which have occasionally challenged the capabilities of Blaise IS. Complexities have included handling multiple arrays of survey data, addressing complicated edit checks that involve calculations using fields spanning the entire survey, sporadic issues with question display in various browsers, CATI management not recognizing web activity, incorrect delivery of cases in the day batch, and problems displaying Blaise tables.

Statistics Netherlands works with us continuously to improve Blaise IS and has provided us with periodic updates or workarounds to resolve most of these deficiencies. Updates have included solutions for browser display issues and complex table displays; in some instances, however, we still have to write additional code to overcome issues. In spite of these issues, using a single instrument for conducting a multiple-mode survey is the method we have chosen more than any of the others, as the advantages have easily outnumbered the disadvantages.

With all modes needed for a survey residing in the same instrument, multimode surveys require less redundant questionnaire programming. For most of the questions we encounter, one field can be declared for all modes. For those questions that need to be presented or asked differently in each mode, we used Blaise's multilanguage capabilities and declared each mode as a separate language (see code example 5.1 for a CATI/CAWI example).

Because the logic path (the Blaise rules) and text fills are usually identical for multiple modes, redundant code is eliminated. However, if the logic or text displayed is specified differently for each mode, then separate *IF statements* and question variables are needed in the *RULES section* to handle the differences for each mode (see code example 5.2).

Fields in Blaise group tables, which use only enumerated types and have question header text in CAWI, although needing introductory questions in CATI, can be shared by using two separate LAYOUT sections (see code example 5.3). These separate layout sections are declared in the mode library individually for each mode. Although the planning and actual programming in doing this can be more involved and time-consuming for the instrument programmer, the benefits of sharing fields and coding just one instrument outweigh the cost of the complexity.

Writing the instrument code for a single Blaise instrument to be conducted in multiple modes can complicate your programming efforts, particularly when the survey is complex and includes tables, which we use often in collecting fields with different data types. Because the programming requirements often differ between modes, challenges arise when coding to accommodate all the modes involved. Sharing frequently means compromising; although we were able to avoid trading off capabilities most of the time, there were instances in which we had to write superfluous code or forgo a request due to increased programming costs—for example, a multimode survey instrument that requires CATI to allow for nonresponse options, but not in CAWI. Because one field cannot have different field attributes, we had to write redundant edit checks in CATI for every field shared with CAWI (see code example 5.4). This inability to declare different attributes for each mode within a field particularly causes problems when programming tables. The *don't know* and *refusal* options in CATI interfere with the table display in CAWI. The solution involved creating two separate tables, one for each mode, so they could display properly in a web browser and be asked in the manner needed by a CATI interviewer. Because we had to allow for a respondent possibly to transition between modes, as the instrument could be started in one mode and then be prompted to finish in the other (usually starting in CAWI and finishing in CATI), we occasionally had to assign the collected value from one mode into the other mode's field so it would then be on path and available as part of that mode's rules. Creating two separate instances of the same question can compromise data integrity if this is not carefully programmed and then tested (see code example 5.5).

Although the flexibility to switch easily between modes is powerful, it is not always reliable. We have experienced difficulties, including delayed day batch updates and case management issues. For example, we sometimes have inadvertently called a respondent to interview just after that respondent had completed the survey in CAWI because the CATI day batch had not yet refreshed to reflect that the case had been completed and should not be delivered to a CATI interviewer. To solve this problem, we developed a Maniplus program, scheduled to run every 15 minutes during interviewing hours, that updates the day batch if necessary. However, when accessing a case via CAWI, the CATI management block is not updated; as a result, the call scheduler is still not completely up to date.

An additional advantage to using a single Blaise instrument with one centralized database for surveys conducted in CATI, CAWI, and CADE modes is in its case-locking capabilities. CAWI

respondents, CATI interviewers and CADE staff rarely encounter this feature when accessing data in their respective data collection modes. For example, when a CATI interviewer talks to a respondent on the telephone, the respondent is unlikely to log into the web survey at the same time. This feature is important, however, on surveys in which different respondents might have to complete different sections of the same survey. An example would be a survey of businesses in which we ask questions about financial records and how the business is managed. In this situation, different staff from within the business might have to address certain questionnaire sections based on their knowledge and expertise. However, we would not want more than one respondent to access the same instrument in CATI or CAWI mode at the same time, because one could potentially change the other's data. One minor issue we have noticed with Blaise in this multimode, one database concept is when a CATI interviewer, CAWI respondent, or programmer has to access the case again immediately after exiting. Occasionally they might be presented with an error message that the case is not available, when it actually should be. The message is presented because it can take from a few seconds to several minutes to clear the lock. The delay does not cause issues often, so this is not a deal breaker for us using the single instrument in this way.

4. Conclusion

4.1 Mathematica successes with multimode

Mathematica has conducted multimode surveys successfully for many years. It wasn't until the past six years, however, that we have been able to use a single real-time database containing both multimode CATI and CAWI surveys. Previously, like any other survey organization, Mathematica had one database for each mode, then either used a complex procedure to combine the data at the end of the data collection or tried to move data from one database into another while interviewing was active. Either process was time-consuming, expensive, and introduced the possibility of errors in the final data because of all the procedures moving survey information around.

By moving to a single real-time database multimode CATI and CAWI survey, Mathematica has successfully reduced the total development cost for these types of multimode surveys. By eliminating the steps moving data from one database to another, we have reduced costs and eliminated possible errors in the data movement between systems.

Moving in this direction did introduce new survey management issues that had to be addressed, however. For example, suppose you start data collection by calling a respondent and getting partway through the interview, then set up an appointment to complete the interview later. Before the appointment is delivered in the call scheduler for telephone interviewing, the respondent responds to the survey by CAWI but still does not complete it. Do you keep the original appointment that was set, do you delete the appointment, do you have the case go back into the normal pool for calling, or do you adjust the appointment to a later date hoping the respondent will complete the survey by CAWI so a call back won't be necessary? Another example: you want your respondents to respond by CAWI so you can lower your interviewing labor costs by not having to pay for telephone interviewer time. If the respondent doesn't finish the case by CAWI, how do you then manage calling the person for a follow-up?

Another beneficial aspect of using one database and one instrument for both CATI and CAWI is that it decreases the instrument development time. In theory, creating one instrument should save development time and labor costs, because you do not have to write two different instruments. One might think this would cut your development cost in half because you are writing just one instrument. In reality, however, you probably will save only about 30 percent in development costs.

This is primarily because of the complexity introduced due to mode differences, which take longer to program. For example, it is not always possible to design a question that can be displayed to a CAWI respondent or asked by a CATI interviewer in a similar manner. On the other hand, if you do have to make a change to the code, you might see additional savings because you are programming just one instrument for both CATI and CAWI. The changes would be in one instrument, whereas previously, if you made a change in one instrument for a particular mode, you would have to make that change in the other.

From our survey division's perspective, having one database that is maintained in a real-time environment offers better reporting of the data they collect, which translates into better management of the survey and decreases the overall costs. Reports available to the project are up to date and available anytime they have to be reviewed. In contrast, in the past, reports were normally created after some data combination process occurred and were prone to logic or crosswalking errors. Due to the cost and possible programming logic in combining of data, reports were basically a snapshot picture from a particular moment, whereas with one database it is much easier to create an up-to-the-minute report. Another benefit of having all the data in one location is that it enables us to write a reporting process only once, instead of possibly writing a report for one mode's database and then needing a secondary report process for the other's database.

We have conducted a few surveys in which we have done CATI/CAWI/CADE in one instrument, one database, all in real time. However, due to limitations of using Blaise as a high-speed data entry program, we have not had as much success as we would have liked. Although Blaise can do data entry, it is not very cost-efficient, cannot do double key entry verification without a lot of tricks and/or additional programming code, and cannot keep statistics on the error rates of the data entry staff. In addition, the extra coding increases the complexity of the instrument and adds greatly to the programming cost. We have found it easier and cheaper to use specialized software designed for high-speed data entry than trying to incorporate this mode into our Blaise real-time, one database concept. We will continue to look at Blaise as a high-speed data entry application if some fundamental changes happen (for example, native double key verification). As a side note, we have discussed moving to a relational database for back-end data storage, and this might allow our high-speed data entry package to write directly to the same relational database.

We have had some success with Blaise writing to a relational database, but not using the Blaise Datalink application. The product we developed uses the Blaise API and allows a Blaise instrument to write to its native database while also writing to a relational database. We have used this product, called SQL2Blaise, on multimode surveys in which Blaise is used for CATI and we use our in-house web product called WebSurv. We have thought about expanding the use of SQL2Blaise, but we will wait for additional testing of Datalink in Blaise 4.8.2 before making a final decision.

4.2 Future of Blaise and multimode surveys at Mathematica

Building upon the success we have had with multimode using one instrument for CATI and CAWI, and one database accessed in real time, we would like to begin to include CAPI in this design. Our ultimate goal is one instrument, one database accessed in real time for CATI, CAWI, and CAPI. We have tested this concept in a nonproduction environment with great success, but in the United States wireless broadband coverage is still a strong limiting factor. Coverage continues to increase, but there are still vast sections within the United States where wireless broadband doesn't work efficiently or you have to use multiple carriers to cover the areas in which you are interviewing, which increases the costs for your project. We envision this being an invaluable solution to meeting multimode data collection needs efficiently in the future, but it is still a few years away. However, when we do reach this point, it will raise some interesting questions. Should the

CAPI instrument be another version of the CATI instrument or should it be a CAWI instrument shared by all modes? Testing and client preferences will probably determine the final resolution on that issue, and there is no definite opinion in either direction. Also, if you are in the field doing real-time CAPI and writing data directly to a centralized database at the home office, what happens if the broadband connection goes down? The latest version of the .NET framework might offer some possible solutions if the connection back to the office cannot be made to store the data locally and when the connection returns a process in the background could upload the data from a laptop back to your central database. We feel we are getting closer to moving in this direction for real-time multimode CAPI surveys, but we think we are still a few years from this becoming a reality.

Because of the cost of producing a Blaise IS instrument compared with our other CAWI systems, we hope the next generation of Blaise (Blaise NG) will be a lot easier and more flexible to use for developing and deploying instruments. For this to happen, the designer portion of the system must be very easy to use (possibly to the point where a nonprogrammer could design the basic survey) and must easily address possible mode differences without creating additional fields or tricky coding to overcome these difficulties.

There has been a push in recent years toward unimode instruments, in which all modes are designed with commonality in mind. If unimode instruments do move into the forefront, that could help address the problems of differently designed questions for different modes. We are still left with the problem that field attributes could be different for certain modes, but, hopefully, Blaise NG will also address that.

Finally, we believe multimode with one programmed instrument and one real-time centralized database is here to stay. There are several systems already on the market that attempt to do this, but typically they cannot handle the complex surveys we conduct at Mathematica. We hope Blaise NG will continue to build upon this concept, while making it easier to create instruments without the additional code needed to handle mode differences and improving its ability to create CAWI screen layouts. Having seen some early beta versions of Blaise NG, it looks as if Statistics Netherlands is addressing the cumbersome process of making changes in various locations to get a screen to display across multiple modes and in various internet browsers.

5. Code Examples

5.1 Example of using Blaise's language utility to declare each mode as a separate language

```
A2
     ENG "What is the correct business name?"
       WEB "@BV2. Please enter the business name.@B@/" : STRING[50]
```

5.2 Example of using IF statements to specify different paths between modes

```
IF (A5a=NonOwner) THEN
    IF (piModeOfProcessing = CATI_) THEN
        A5a_NonOwner
    ELSEIF (piModeOfProcessing = WEB_) THEN
        A5New
    ENDIF
  ENDIF
```

5.3 Example of using two separate LAYOUT sections, one for each mode

```
AUXFIELDS
       Label
      WEB "@BF3. Equity investment is money received @U in return for
       some portion of ownership@U, and it is another way to fund
       business expenses.
       During calendar year 2009, did the business obtain @Bequity
       financing@B from any of the following sources?@B
       @/@/@I@OPlease indicate Yes or No for each item.@I@O"
      ENG “Equity investment”
  : STRING[30], EMPTY

RULES
...

LAYOUT
       BEFORE Label NEWPAGE
LAYOUT (Internet)
       BEFORE Label NEWPAGE
       AT Label FIELDPANE BISQuextextOnly
```

5.4 Example of redundant edit checks due to the inability to declare different attributes for each mode within a field

```
F3f
IF piModeOfProcessing = CATI_ THEN
    F3f <> EMPTY "You must answer."
ENDIF
F3g
IF piModeOfProcessing = CATI_ THEN
    F3g <> EMPTY "You must answer."
ENDIF
```

5.5 Example of storing the collected value into the other mode's field

```
FOR I := 1 TO 9 DO
    IF piModeOfProcessing = CATI_ THEN
      C1CATI[I](aItem[I], aLetter)
      C1[I].C1 := C1CATI[I].C1
      C1[I].C2 := C1CATI[I].C2
      C1[I].C3 := C1CATI[I].C3
    ELSE
      C1[I](aItem[I], aLetter)
      C1CATI[I].C1 := C1[I].C1
      C1CATI[I].C2 := C1[I].C2
      C1CATI[I].C3 := C1[I].C3
    ENDIF
  ENDIF
ENDDO
```