Customizable .NET Event History Calendar: Looking to the Future

Roberto V Picha, Daniel Moshinsky, Mecene Desormice, Seth Benson-Flannery, U.S. Census Bureau

1. Introduction

The U.S. Census Bureau's survey of Income and Program Participation (SIPP) provides monthly information about the nation's income, wealth, and program usage. Currently, SIPP administers three interviews per year to each sample member; each interview's reference period covers the preceding four calendar months.

In 2006, the Census Bureau initiated a SIPP re-engineering effort, a key component of which is a shift to a single annual interview covering the preceding calendar year. An Event History Calendar (EHC) is employed to help address the concern that the longer reference period may impact respondent recall. To study the implications of the switch to the EHC method of data collection and the expansion of the survey's reference period, it was decided to conduct several field tests prior to the final production implementation.

In the beginning of 2010, the first field test of the re-designed SIPP instrument was conducted. There were 7,982 sample cases for this field test. This paper will describe some of the initial findings of the field test and lessons learned as they relate to the use of the Re-engineered SIPP (ReSIPP) instrument and its EHC for a second field test which is scheduled to take place in early 2011.

2. Background

ReSIPP contains some 40 sections. This includes sections relating to employment history, government program use, health insurance coverage, assets, utilization of health care services and demographic information. The ReSIPP is a survey that interviews and collects data for each household member, directly or by proxy for people over 15 years old and by proxy for children under 15 years old. Not all of SIPP questions are well-suited for being asked in the EHC.

It has been shown that the Event History Calendar method of interviewing is more effective than the standard question list method in improving recall of dates of autobiographical events for most topics (Belli et. al., 2001). Therefore, the types of ReSIPP sections that are asked in the Event History Calendar contain questions involving the recall of dates for changes in status throughout the reference period. For example, questions about the dates of job changes or changes in receipt of benefits are collected through the EHC.

On the other hand, questions relating to topics such as assets pertain to the reference period as a whole (e.g. "What was the combined value of your assets in the previous calendar year?") and do not involve recall of dates. Topics such as these have no transitions and are included outside the EHC section of the instrument. Thus, these sections can still be asked in the standard question list method and can be programmed in Blaise.

3. Features of the 2010 Instrument

For the field test in 2010, we designed a customizable EHC, programmed in C# .NET, and invoked it from Blaise 4.8 as a COM object DLL.

Twenty-five of ReSIPP's forty content topic sections were collected in the EHC. The interview began with introductory and demographic questions collected in Blaise. This was followed by the launching of the EHC. The respondent would answer the next 25 sections (containing several hundred questions) in the EHC environment. Then, the respondent would return to the Blaise DEP to complete the remainder of the interview.

Parameters from the Blaise DEP were passed into the EHC via the Blaise API, and data collected in the EHC were saved via the API in the Blaise database. One common Blaise database was used to store all data for the case. Question text, answer lists, topics, rules, and screen layouts within the EHC were created dynamically based on the Blaise datamodel and several external metafiles. Nearly all survey-specific content was described in the Blaise metafiles without rules – the C# EHC functioned as a user-interface and a shell for displaying that content. This design is described in full detail in the paper presented at the Riga Conference (Daniel Moshinky, Mecene Desormice, Seth Benson-Flannery, 2009).

The result was an easily customizable instrument that yielded a robust data collection system that not only included all the features of an Event History Calendar, but also aimed to mimic the question flow and error checking that our users have come to expect from Blaise instruments.

4. Results from the 2010 Field Test

Feedback from the 2010 field test has come from direct observations of live interviews, as well as from written and oral comments from field representatives and their supervisors who were queried during a number of debriefing sessions.

On the whole, the majority of debriefed interviewers have rated the Event History Calendar as a very positive experience and have commented on how much they enjoyed the greater freedom and conversational style of data collection with the EHC. The interviewers also offered a number of criticisms and constructive suggestions of the 2010 tests.

4.1 Survey content

The overwhelming consensus of the Field Representatives (FRs) was that the survey was much too long, with interview times of several hours being common. In addition, FRs complained that many of the questions were repetitive and redundant. For example, many of the questions asked who in the household was covered by various benefit and health insurance programs. When subsequent household members were interviewed, the instrument re-asked those same questions again, instead of pre-populating the fields based on the answers already collected. As a result of these shortcomings, respondent engagement waned, with many respondents becoming restless, annoyed, and uncooperative. It became clear that the survey will need to be shorter and “more intelligent” if it is to maintain the cooperation of the respondents.

4.2 Usability

Overall, contrary to our concerns about the FRs acceptance of this new technology, most did not report any problems using the EHC. Of the FR comments made available to us, none commented about problems within the Event History Calendar environment. Also, no problems navigating the EHC were

reported to our technical support staff during the interview period. This does not mean that the EHC was problem-free. It is more likely that the FRs focused their comments on the problem that was most pressing to them – that of the survey length and content.

4.3 Use of EHC techniques

Because audit trails and audio recordings are still being analyzed, full results about the patterns of the FRs' use of EHC interviewing techniques to aid respondent recall are not available yet. However, early feedback shows that the FRs were often confused about the meaning and purpose of certain topics such as the "Landmarks" line (some suggest it be renamed to something more descriptive, such as "Life Events"). The FRs used other topic lines for cross referencing more than Landmarks.

4.4 Data

Data output from the 2010 field test is still being reviewed. The initial results show good data integrity resulting from our single-database design approach, but it is too early to draw any further conclusions about the quality of the data collected using the EHC method.

In summary, the FRs we have debriefed strongly warn that the response rates in subsequent interview waves will drop significantly if the length of the interview is not shortened.

5. New requirements for the next field test

A number of major requirement changes were proposed to us to accommodate FR feedback, reduce respondent burden and improve data consistency. Firstly, the EHC reference period was to be extended to the interview month – resulting in a reference period ranging from 13 to 18 months, depending on the interview date. This was done in order to be able to feed data into subsequent interview waves that would jump-start interviews and reduce the number of questions asked.

Secondly, access to a number of EHC topics that depend on the respondent's income level would depend on a few income screeners asked in the beginning of the interview. This reduces the number of questions asked of respondents whose income is too high to be eligible for the programs involved.

Thirdly, common information would be copied across household members. For example, answers about marital status would be copied from one spouse to the other, and answers about residency would be copied across all household members who have lived and moved together as a unit.

Lastly, we were asked to create the "Time without a job" line that would be automatically computed based on the dates provided on the "Job" lines– rather than asking the FR to enter the "begin" and "end" months manually for respondents that reported being out of work as it was done in the 2010 field test.

In addition to these new requirements, we needed to address some technical shortcomings of the 2010 EHC instrument. As the number of questions asked in the EHC grew tremendously over the course of the EHC development, so did the complexity of the universe statements and edit checks involved. For the most part, the rules functioned as specified, but in some situations there were problems with rule re-execution when the FRs backed up and changed values inside the EHC. With the ever-growing complexity of requirements, our existing mechanism of using a metafile and some auxiliary functions (as described in our 2009 IBUC paper) became quite difficult to maintain. It was decided that a much more efficient and easier way would be to delegate all rule processing to the Blaise engine.

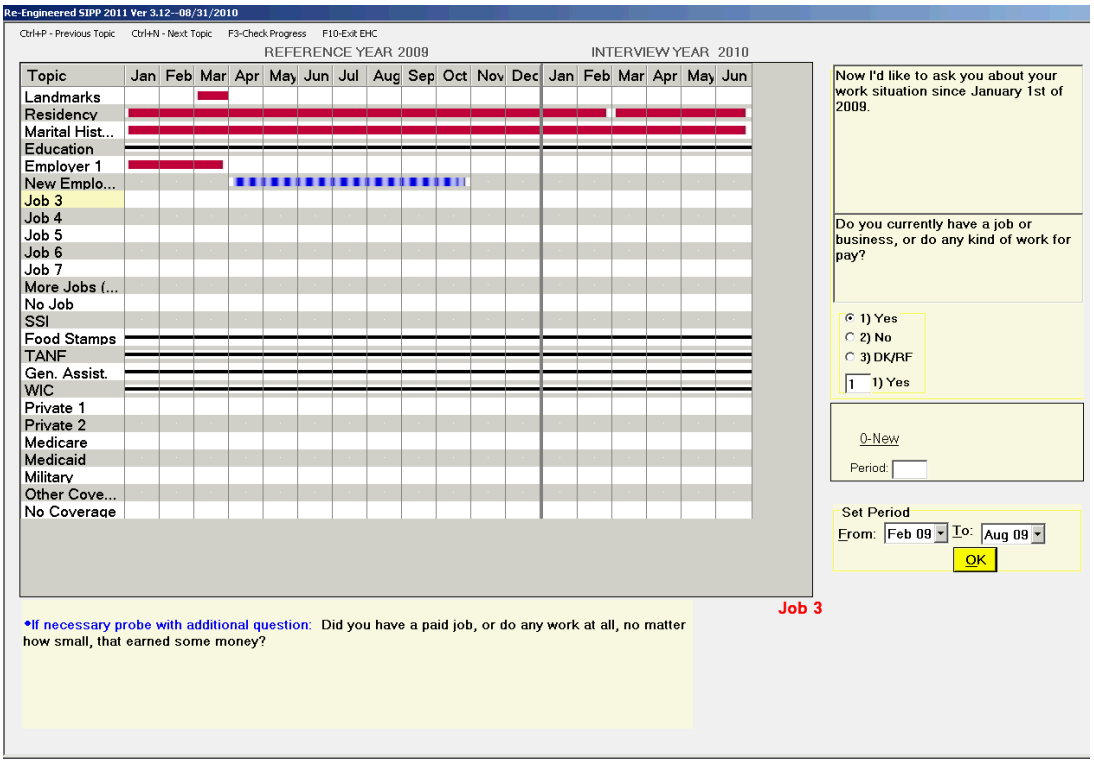
At first, we attempted to display the topics' detailed follow-up questions in the EHC by using the BCP and the Blaise API to re-execute the rules, evaluate universe conditions, and determine the next field on path. However, this approach would require us to store values to the Blaise database and force rules re-execution after every single entry in the EHC. Considering the size of the instrument, this approach carried a considerable performance burden.

We decided that the most straightforward approach would be to collect the "begin" and "end" months in the EHC, while moving the follow-up questions into the DEP and simply toggling between the two environments during the interview. Interview time would be reduced by expanding the use of screener questions to screen out respondents who are not eligible for certain sections (e.g. people with household income above a certain threshold), copying available data from one household member to another, automatically creating periods of unemployment, and improving the way the user interface guides the interviewer to the next action.

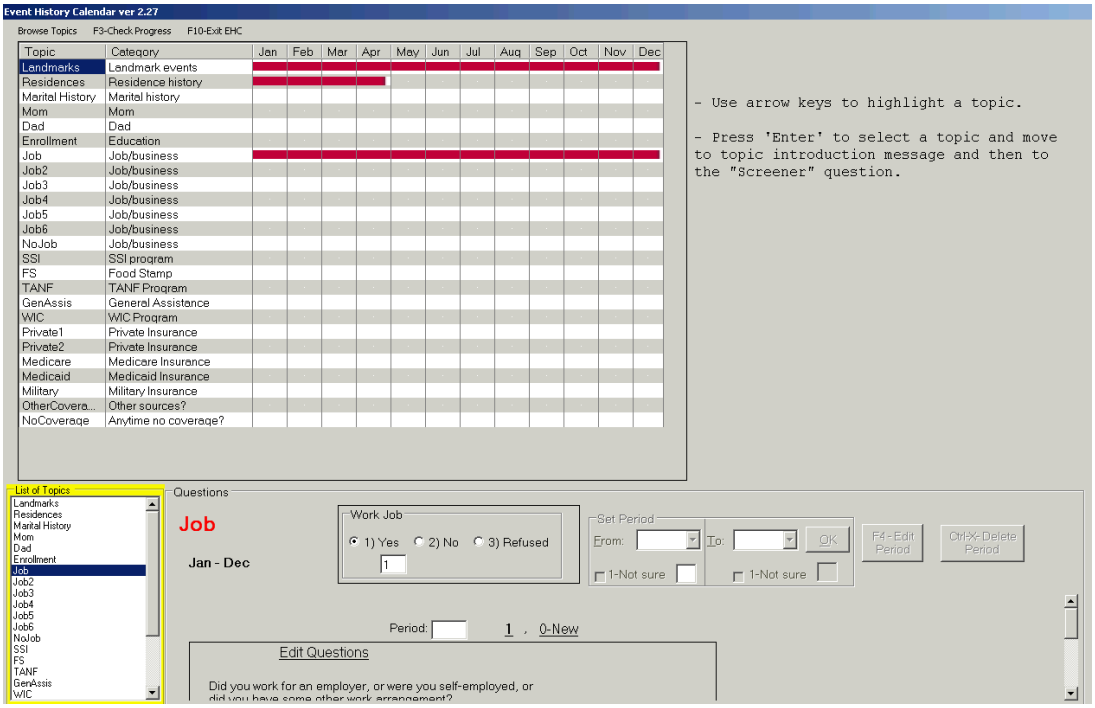
6. Design of the 2011 ReSIPP EHC

After the EHC is first launched, the user is taken to its main screen, which contains a list of topics and timelines. After a topic is selected, the user is asked one or two screener questions, and asked to create a spell. After the user creates a spell, the EHC main screen closes, and the user is taken to the DEP to answer the detailed follow-up questions for that spell period. After the last question in that topic (or earlier if a function key is pressed), the EHC appears on the screen again, and the user can either enter another spell period for the same topic, or begin data collection for a new topic. Thus, whereas in 2010 all of the follow-up topic questions were asked within the EHC, in 2011 they are being asked in the Blaise DEP.

The graphic below shows the new EHC screen for the 2011 field test.



The graphic below shows the EHC screen used in the 2010 field test.



6.1 Entering data in the DEP from the EHC

Within the ReSIPP instrument, each topic is stored as a block. (Depending on the topic, each of these blocks may also contain additional blocks.) All of the topic level blocks are stored in our TEHC block.

Upon leaving the EHC to go into the DEP portion of the instrument, three auxfields are populated via calls in the C# code. In the code presented below, these are **showQs** , which represents the topic selected in the calendar; **prdSelected**, which represents the instance of that topic (and spell period) that has been selected; and **AskReturning**, which is a flag set to "TRUE" if this is not the first time entering a particular period and "FALSE" otherwise.

If both **showQs** and **prdSelected** have a value, then we are to ask the detailed follow-up questions for a particular topic and period. In the case where **AskReturning** has been set to "TRUE," we will first ask an auxfield named **Returning** that only allows a value of "1." The purpose of this field is both to notify the user that they are about to enter a spell in which they have already entered data and to act as a gateway into a previously entered spell. When we leave the EHC, we need an empty question on route to land on. In the case where data have already been entered, **Returning** assumes that role. After **Returning** is answered, or if **AskReturning** = "FALSE", the user is taken into the **TEHC** block, passing in **showQs** and **prdSelected** as parameters.

```
AUXFIELDS
 {...}
  prdSelected  : INTEGER
  ShowQs       : STRING[200]
  AskReturning : STRING[5]
  Returning
             ENG "@/@Zs@Z @LYou are entering this period to edit your data.
                  Press '1' to continue.@L" / "Returning"
              : ARRAY[1..20] OF TContinue, NODK, NORF
{...}
RULES
{...}
 IF showQs = RESPONSE AND prdSelected = RESPONSE THEN
    IF UpperCase(AskReturning) = 'TRUE' THEN
       Returning[I]
    ENDIF
      TEHC[I]({import} CTRLNUM,I, {... more parameters} )
 ENDIF
{...}
```

Within the EHC block, these parameters (called **IN_showQs** and **IN_prdSelected** in the code below) are used to place the spell level questions on route. Using this method, only one instance of one topic can be on route at any time. (Please note that, although it is not displayed in the code below, when using this technique, all of the spell level blocks need to have a KEEP statement associated with them.)

```
PARAMETERS
    IMPORT In_ShowQs : STRING
    IMPORT IN_prdSelected : INTEGER
    {...}
RULES
  IF (uppercase(IN_showQs) = uppercase('BLandmark') OR uppercase(IN_showQs) = uppercase('BNoJob') OR
    uppercase(IN_showQs) = uppercase('BMedicaid') OR uppercase(IN_showQs) = uppercase('BNoCoverage')) THEN
    FOR I := 1 TO 6 DO
      IF I = IN_prdSelected THEN
        {...}
        IF uppercase(IN_showQs) = uppercase('BLandmark') THEN
          BLandmark[I]({Import} CTRLNUM, LNO, I,{... more parameters})
        ELSEIF uppercase(IN_showQs) = uppercase('BNoJob') THEN
          BNoJob[I]({Import}CTRLNUM, LNO, I, {... more parameters})
        ELSEIF uppercase(IN_showQs) = uppercase('BMedicaid') THEN
          BMedicaid[I]({Import}CTRLNUM, LNO, I, {... more parameters})
        ELSEIF uppercase(IN_showQs) = uppercase('BNoCoverage') THEN
          BNoCoverage[I]({Import} CTRLNUM, LNO, I, {... more parameters})
        ENDIF
      ENDIF
    ENDDO
  ELSEIF uppercase(IN_showQs) = uppercase('BResidency') THEN
    FOR I := 1 TO 5 DO
      IF I = IN_prdSelected THEN
        {...}
        BResidency[I]({Import} CTRLNUM, IN_LNO, I{... more parameters})
      ENDIF
    ENDDO
```

6.2 Returning to the EHC from the DEP

When the spell period level detailed questions have been answered, the interview is directed back into the EHC calendar. This is done by using an event to call a COM-object DLL associated to fields at the end of the block of detailed questions.

Because each spell period may contain many different variables that could be the last one on route, we opted to create a new shared **EndSpells** block that comes on route at the end of detailed questions for all topics. Within this new block, we have three questions that could be the last one on route (e.g. "Do you have any more periods to report?"). The last variable triggers the event to call the EHC DLL, which will determine whether to open up the EHC calendar or do nothing. If instrument logic dictates that the EHC is to be launched after the first question, then code within the DLL control will launch the calendar. Otherwise – if we are not ready for the calendar to be launched -- the code within the DLL control will simply return control back to the DEP without launching the calendar, and the focus will go to the next field on route. By coordinating the Blaise code and the C# code, we are assured that we always return to the EHC calendar at some point in this new block.

After the EHC is re-launched, DEP fields are used to guide the focus on the EHC screen and help guide the interviewer's transition. For example, if the interviewer indicated in the DEP that there are more time periods to report within the current topic, the focus is guided to the EHC field at which a new period can be created. On the other hand, if it was indicated that there are no more periods to report (or if the entire reference period has been covered), then the focus is guided to the calendar grid and onto the subsequent topic. Still, FRs maintain the power to navigate freely within the EHC and can enter any topic at any time. The intention is to aide the flow of the interview and to reduce the number of FR key strokes while preserving the flexibility to enter data in a free form way consistent with the conversational style of data collection within the EHC.

The transfer of follow-up questions to the DEP offers several key advantages. Firstly, the full power of the Blaise rule-checking can be used to route the instrument. Secondly, the inclusion of the EHC questions within the DEP provides users with a consistent user interface and makes the instrument appear more seamless and easier to use. In addition, the DLL launches quicker than it did in the 2010 instrument because less data now needs to be passed via the API.

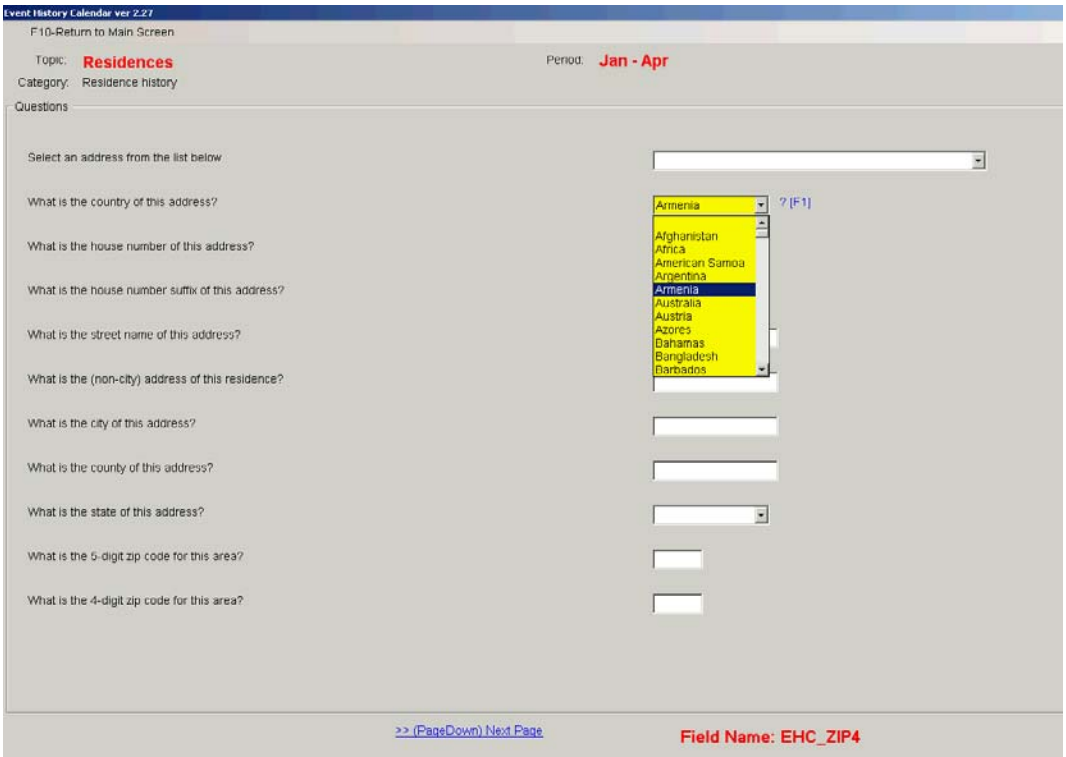
7. Challenges

7.1 Transition to the new EHC Design

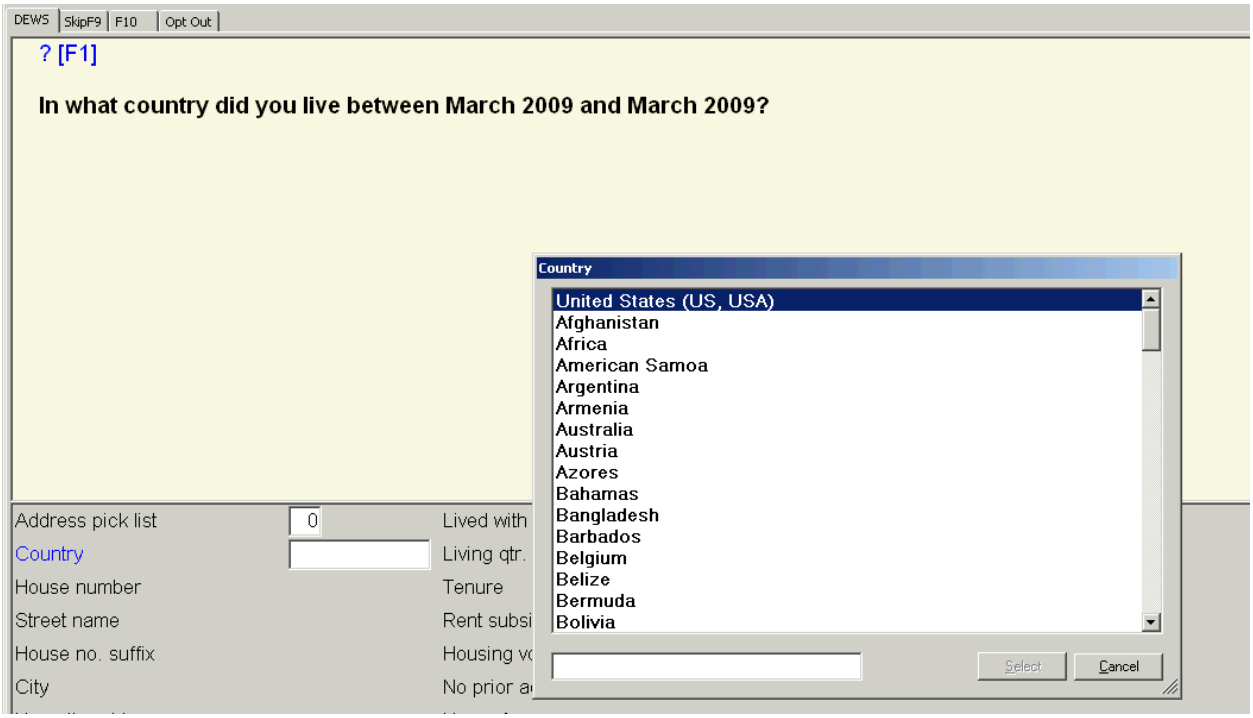
From a technical standpoint, the transition to the 2011 approach was fairly painless. All of the Blaise fields and blocks already existed in Blaise and were being read in by the EHC in the 2010 instrument. Rules had to be added, and the EHC had to be modified to return to the Blaise DEP rather than launch a window to ask the follow-up questions. The transition took us no more than a couple of weeks.

This was achieved because of the loosely coupled object-oriented design of the EHC, and our focus on making the EHC flexible and customizable. Because all of the instrument content had been driven by the Blaise database and the metafiles, very minimal changes were needed to accommodate the new approach. The balance of our development time has been spent incorporating additional requirements, making enhancements to the GUI of the EHC, and testing. A number of technical problems had to be overcome.

The graphic on the next page shows the Residency topic detail questions as they appeared in the EHC used for the 2010 field test. Note that the country combo box covered some other items below it. In order to browse through the combo box, the FR had to select the item again which was always difficult. Another issue we encountered was selecting the respondent's country as part of his address. In most cases this was the United States, and that option was not available as the first option.



The graphic below is the Blaise side of the data collection for the 2011 field test. A lookup table is now used to display countries list defaulting to the United States as the first option displayed.



7.2 Copying data

One of our biggest challenges revolved around copying data, both to populate data and to populate enumerated fields. For example, in our topic section on residency in our 2010 instrument, we were copying address information to individual fields (detail questions regarding the current address), as well as copying the entire address to an enumerated list so subsequent respondents could simply choose an address from a dropdown list rather than retype all the information. In 2010, this was done using a combination of Blaise and C#. However, the dropdown list was not as user friendly as expected. As a result, FRs were instructed to handle these items with extra care (extra keystrokes) since the dropdown automatically expanded as soon as the focus was active for this question blocking some other questions on route and confusing the FRs at times.

For our 2011 instrument, we have chosen to take advantage of Manipula for these tasks - evoking events in the DEP to call Manipula programs to populate our data. Our Manipula program is required to loop through numerous arrays, unduplicating addresses to populate an enumerated answer list to mimic the dropdown list we had in C#. We had concerns that this may add considerable time whenever this was called, but we found that the program worked quickly enough where no significant delay was detected. Impressed by the speed of this, we added another Manipula program, called more frequently, to populate data that would remain the same from person to person in a household (e.g., fields based on the residence). Again, we found that this did not create a significant delay to the program, and we have extended the use of this routine to many areas of the instrument.

7.3 Look-up tables

Another new requirement was to use lookup tables working in conjunction with one another to narrow choices down. The best example of this is our state and county lookup tables. When a state is chosen from a lookup table, we create another lookup table consisting of counties (or their equivalents) from that state. Because of the familiarity with the C# code, we opted to build these in C#, although we could have achieved this programming in Manipula had we opted to do so.

7.4 Audit Trail

In the 2010 instrument, we had one audit trail file for the DEP portions of the instrument and another audit trail file for the EHC portion of the program. Because the interview only transitioned to and from the EHC a few times per interview, it was relatively easy to reconstruct the flow of the instrument even with two audit trail files. However, in the 2011 instrument there will be many more transitions between the EHC and the DEP, and analyzing two audit trail files would not be practical.

To rectify this, we are now only creating one audit trail file. In order to do this, we have switched the "CloseFile" toggle in the .aif file so that the audit trail file is opened and closed every time a line is written to it (i.e. CloseFile=1). This removed the Blaise generated lock file after every Blaise entry, meaning that there is now nothing preventing us from writing to the Blaise generated audit trail file when we are in C#. Therefore, we are still able to use the techniques we were using to write to a C# audit trail file in our 2010 instrument, but we are now writing directly to the Blaise generated audit trail file.

In the future, we are planning to use a Manipula script in conjunction with the CloseFile setting at various points during the interview, and write additional information (i.e. paradata) to the audit trail file, such as the FR identification code, respondent name, outcome codes, and so on.

8. Conclusions

In the enhancements made for the upcoming 2011 field test, we worked to help our sponsors address the biggest concerns of the 2010 field test – that the survey was too long, and that it contained too many redundant questions. We have tried to create a smarter, more responsive and agile instrument with the help of the Blaise DEP and Manipula, using the Event History Calendar as a user-friendly shell for graphically selecting chronological information.

9. Disclaimer

This document is to inform interested parties of ongoing research and to encourage discussion of Blaise instrument design issues. The views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.