A Random Walk Application for Blaise Instruments

Gilbert Rodriguez & Jay R. Levinsohn, RTI International

1. Introduction

Social science data collection has seen a significant movement toward the use of automated data collection procedures, using increasingly sophisticated tools. The Blaise programming language and data software are the kind of tools used in this movement. At RTI International we have been using Blaise to develop large, complex survey questionnaires for several years. While there have been significant advances in the power of the tools for development and deployment there has not been the equivalent development of tools to provide for quality assurance. For the last two years we have been seeking improved methods for Blaise questionnaire validation and quality assurance. We have developed a system for automating some aspects of the testing process, a software package we call RoboCAI (Levinsohn and Rodriguez, 2001). The RoboCAI process allows one to automatically execute predefined test scripts, compare the data from the executed scripts, and log and report results. A significant issue in this process is the level of effort required to develop meaningful scripts in sufficient quantity to provide adequate test coverage. As an extension to the capabilities afforded by RoboCAI we have developed a second process that will automatically generate scripts with relative little user effort. This automated procedure uses the concept of the random walk.

A random walk can be defined as a process consisting of a sequence of steps each of whose characteristics is determined by chance (Merriam-Webster online dictionary). The random walk as applied to a Blaise questionnaire implies that at each choice point in the questionnaire (each point where input is required) a random decision can be made to produce an input. As one steps through the questionnaire instrument based on the Blaise program and all previous responses each new choice point receives a new random input. The input is selected from the valid choices for a given questionnaire item. The output can then be examined to verify that the results agree with the questionnaire specifications and requirements. In addition, RoboWalk is designed to allow the developer to combine a mix of random inputs with pre-specified inputs. This process allows fairly rapid generation of test scripts and test scenarios. It allows the developer to build test scenarios that he feels will cover the breadth of cases that define the most likely paths as well as others.

At RTI International we have used RoboWalk to create test scenarios, which are replayed later with RoboCAI to test routing and to generate test data. The scenario logs and data are checked for correctness and consistency using utility programs. Also, by fixing values for particular gate questions in a RoboWalk script, one may create scenarios that emphasize the testing of particular blocks.

The random walk application, named RoboWalk, consists of two components that operate together, First, a WinBatch script starts a Blaise CAI instrument and feeds keystrokes to the CAI instrument and second, a Visual Basic program determines a random response for a given field. The Visual Basic program utilizes the Blaise API library and the data model files for a given questionnaire in order to determine the possible range of valid values for a particular field. It then performs a random draw based on the range of values.

An input file is used to specify a case id number and a few other items, such as questions and responses for which a random choice should not be made or in cases where a developer wants to specify part of the test script. Random choices for some questions can be problematic, since some questions may require specific types of responses in order to advance through the instrument.

An accompanying utility has been created that writes out the CAI instrument screens (in sequential order) to an ASCII text file using the Blaise data file as input. This file can then be examined for logic errors, flow errors, and range errors, as well as correctness of the screen text. Also, a random walk script file can be created from the data file so that a particular scenario can kept to be used as a "test" script in the RoboCAI application.

2. The methodology

The RoboWalk process uses the following tools:

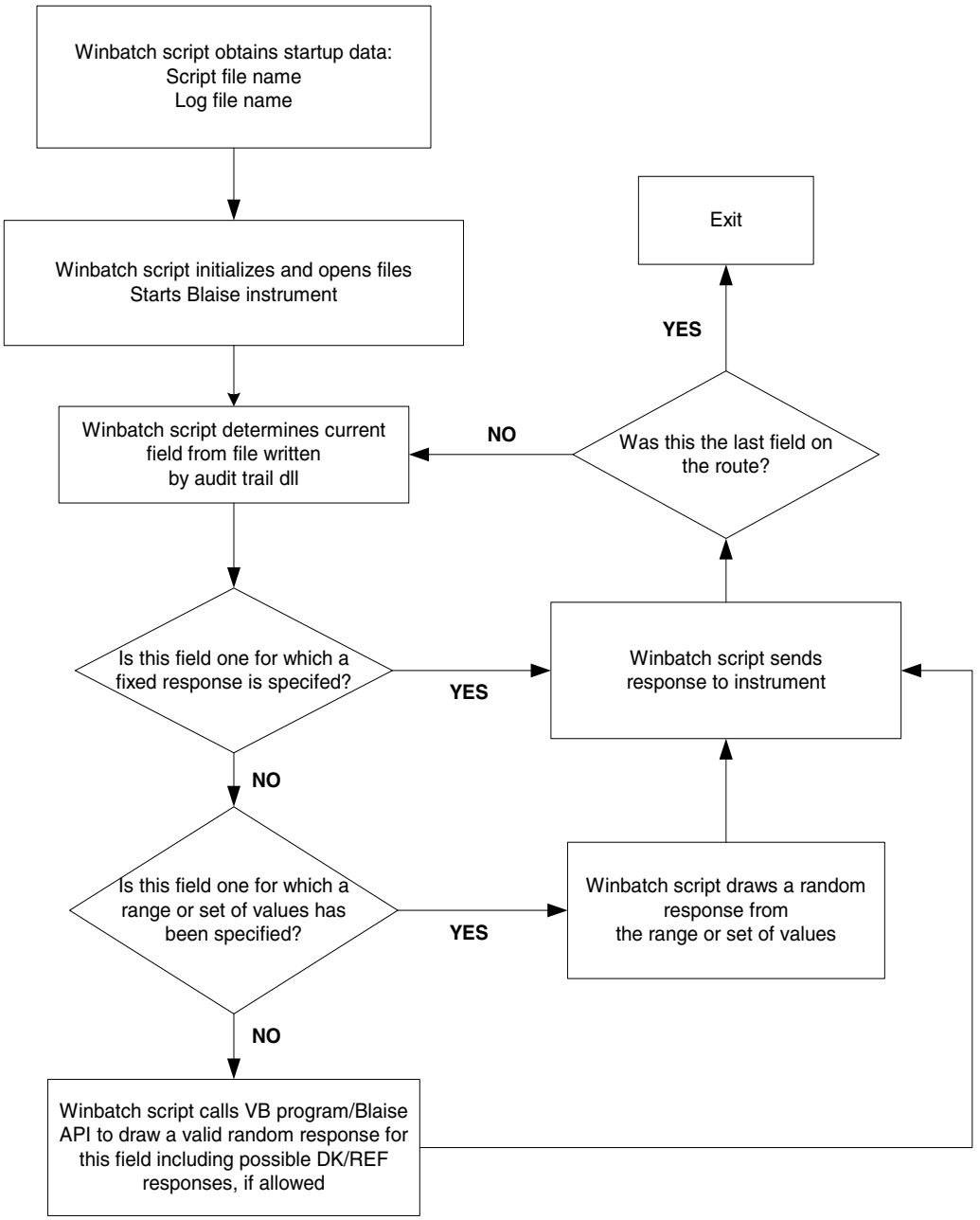
- WinBatch - a batch language interpreter for Windows. Batch files are written using WIL, the Windows Interface Language. A batch file can contain commands to run programs, read and write directly to files, send keystrokes to Windows applications, and perform a number of other tasks. Using the WinBatch compiler, an executable file can be created from a batch file. WinBatch is a proprietary product developed by Wilson WindowWare, Inc.
- Blaise 4.5 – CAI instruments created with different versions of Blaise 4.5 have been used with RoboWalk. Currently, Blaise 4.5.2.666 is being used.
- A modified audit trail DLL, audit.dll – This DLL, written in Delphi, writes out the name of the current field in an instrument to a text file where it can be read by a WinBatch batch file or executable.
- A Visual Basic application that uses the Blaise API objects to determine a valid, random response for a given field and is called as a shelled process from the WinBatch script
- A Visual Basic utility that creates print files of the path through the questionnaire and will also build a script for later use by RoboCAI.

When executed, RoboWalk prompts the user for the name of a text file containing data to be used for the case (which is called a script file) and then prompts the user for the name of a log file to which diagnostic data is written. All problems and cases where more than one random draw is required to proceed to the next screen, such as if a hard or soft error occurs, are logged in the log file. RoboWalk operates as follows:

- starts the CAI instrument,
- reads the script file,
- determines a random response,
- sends the response as keystrokes to a screen of the CAI instrument,
- proceeds to the next screen and then writes the name of the CAI screen to a text file (currentfield.txt) so that RoboWalk can determine the current field for the instrument.
- It then repeats this process for each screen of the CAI instrument.

Figure 1 presents a flow diagram of these steps.

Figure 1 - Random Walk Process



The flow and logic of this process is simple. There is one complication of coding to allow for synchronization of the activity between the Blaise audit.dll and the WinBatch code. These two processes, the random walk process and Blaise, run in parallel under the Windows operating system and RoboWalk must wait for the Blaise code to complete. The random walk process must allow the Blaise audit.dll enough time to open, write into, and close the currentfield.txt file before it can check to see what the next field is in the instrument. This is handled by the

WinBatch script. If the file currentfield.txt file does not exist, then the WinBatch script waits 1 second, and then checks again. This continues until the file exists.

An example will help to make the program operation more clear. The following sections present a short example. In Tables 1 and 2 we present the information that serves as input into RoboWalk (the script file) and the output. In this example scenario the respondent's date of birth is June 4, 1961.

2.1 Sample script

The following is an example of the contents of a typical script. In a RoboWalk script you would specify answers or a specific range of allowable answers (to be selected at random) for only a subset of the questions, the balance of the answers would be generated by RoboWalk. Table 1 presents such a script where only a part of the possible questionnaire items are given answers in the script.

Table 1 RoboWalk script

QUESTID	2000042
DKPROB	0.01
REFPROB	0.02
DOB	6-4-1961
STATE	{ 26, 41, 46, 50 }
EDUCATION	[12, 15]
ENDAUDIO	out
VERIFID	T10-0008
CASEID	TX01010111A
FIEXIT	1

Note that you can specify probabilities for Don't Know or Refusal responses which are used to randomly select a Don't Know or Refusal response for fields for which they are allowed. You may also specify a set of values or list from which a response is randomly selected. For example in the case of the STATE question

STATE {26, 41, 46, 50}

indicates that the response for STATE should be drawn as one of the items from the list 26, 41, 46 or 50 with equal selection probability. Don't Know or Refusal would not be selected for this question since a specific list is provided, only 26, 41, 46, or 50 would be selected. You may also specify a specific range of values from which a response is randomly selected. For example,

EDUCATION [12, 15]

indicates that the response for EDUCATION should be drawn only from those integer values between 12 and 15 (inclusive) with equal probability. The random draw, from a specified range or list, is performed within the WinBatch script rather than in the Visual Basic program.

Also, the WinBatch script attempts to work itself out of any hard or soft errors that may occur. If a hard or soft error dialog box pops up, then the WinBatch script will send a carriage return to clear it and then make another random draw for a response. This problem will be logged for the questionnaire author to review. The

WinBatch script will stop execution if a specific number (a user settable parameter, the default value is 20) of consecutive hard errors occur.

2.2 Sample log

The text displayed in Table 2 is the output log file generated by the random walk program file. The log file contains the responses that have been sent to the CAI instrument either from the input script or from the randomly generated selections made by RoboWalk.

Table 2 RoboWalk Log

Random Walk Log
Thu 1-30-2003 12:02:43 PM

	Screen	Value
0	QUESTID	2000042
1	DOB	6-4-1961
2	CONFDOB	2
***** HARD ERROR ENCOUNTERED *****		
3	DOB	6-4-1961
4	CONFDOB	1
5	CONFIRM	1
6	STATE	26
7	CONFSTATE	2
***** HARD ERROR ENCOUNTERED *****		
8	STATE	50
9	CONFSTATE	1
10	GENDER	5
11	CONFGENDER	4
12	HISPANIC	1
13	HISPGROUP	!
14	RACE	1 2 3 6
15	RACEASIA	1 5 6
16	MARSTAT	4
17	EDUCATION	13
18	HEALTH	2
19	INTROACASI1	1
20	HEADPHONE	
21	INTRO1	
22	INTRO2	
23	HEAROFF	
24	GOTDOG	1
25	EYECOLOR	2
26	ALLAPPLY	1 2 4
27	NUMBER	27
28	STOPLIST	1
29	DOAGAIN	2
30	BACKUP	
31	RANGEERR	2
32	INCONSIS	
33	ANYQUES	
34	ENDAUDIO	out
35	THANKR	
36	VERIFID	T10-0008
37	CASEID	TX01010111A
38	FIEXIT	1

==========
End of Log
Thu 1-30-2003 12:03:40 PM

The developer would need to review the log, shown in Table 2, to insure that the flow of the questions (the exact sequence) given the sequence of inputs is one that is consistent to the questionnaire specifications. If this review indicates any inconsistency it likely indicates an error in the Blaise program or a need to revise the questionnaire specifications.

3. Summary and conclusions

There is a growing understanding that as CAI questionnaires get larger, more complex and use increasingly complex techniques that issues of quality control become a serious issue. There is significant cost in the current testing process and there is a significant failure rate in removing all errors prior to release of the production code. The industry is looking hard for better methods. We believe that the Robo tools that we have developed at RTI International help with this problem. In the past we have presented the RoboCAI tool that will automate the testing process from user crafted "test" scripts. The RoboCAI tool is labor intensive for large applications but extremely useful for smaller ones. In an effort to help with the effort of generating large test scripts we moved on to the random walk application – RoboWalk. A random walk application can be useful as a means of generating test scenarios, especially for very large questionnaires. There is significant effort required in going back and reviewing the resulting data afterwards and this level of effort increases as the size and complexity of an instrument increases. But, we feel that these tools taken in concert make good progress toward providing the questionnaire designer with tools to increase quality. Used together the Robo tools can provide excellent testing, documentation, and regression testing tools for small (<100 items) to medium questionnaires (<250). As the questionnaires increase beyond this size more and more effort is required to create test scripts or to interpret the results of a random walk. Very large questionnaires require very large amounts of effort to understand, debug, and document. These tools certainly can help that process but still leave hard work for the developers and designers.

4. References

Levinsohn, Jay R., and Rodriguez, Gilbert (2001). Automated Testing of Blaise Questionnaires. *Proceedings of the 7th International Blaise Users Conference 2001,* Washington, D.C, USA.: Westat.