Advanced Editing: Integrating Blaise with a Management System

Peter Stegehuis and Seth Benson-Flannery, Westat

Presenter: Peter Stegehuis

1. Introduction

During the stage of cleaning interview data, it is good practice to use the same Blaise datamodel as used in the field for reasons of consistency and efficiency. This paper will touch on the challenges presented by special demands during this phase. Some of the challenges are in the Blaise datamodel and the data itself, for instance, the need for additional checks and being able to reach "behind walls" set during the field interview. We also look at how best to present data issues to staff in an easy-to-use user interface, how to automatically categorize interviewer comments (Blaise remarks), and how to integrate this editing system within a comprehensive Home Office System.

2. What Is There to Clean?

2.1 A Different Focus

During a field interview, especially for a longer and more complex survey, the focus is, of course, on getting high-quality responses, but also on ensuring the continued engagement of the respondent. To keep the interview going, and not frustrate respondents or give them an easy opportunity to stop their cooperation, interviewers and the (Blaise) interview program need to work well without significant pauses. It puts emphasis on good interviewer training on the one hand, and on a well-designed and well-functioning instrument on the other.

For the design, that could mean choosing to not add too many checks, balancing the need for the highest data quality with the need to not slow down the interview or frustrate the respondent to the point where they stop their cooperation. For the CAPI instrument, it means things have to work well and flow well, without glitches or long pauses.

This means that during the cleaning phase, some additional Blaise soft checks may be applied to scrutinize situations that purposefully were not flagged with the respondent during the interview.

In very complex instruments, there may be many places where variables get assigned a value behind the scenes, meaning not in rules that will get reevaluated, which should be undone in case the interviewer backs up and changes answers in a way that leads to a different route through the questionnaire. This cleaning up can get very complicated, and it may be a realistic option to do this cleaning up after completion of the interview, in the data cleaning phase, especially if it is a rare scenario and the data that is not being cleaned up has no impact on the remainder of the interview with the respondent.

These are a few examples of additional checks that can be applied during data cleaning, either by adding checks in the Blaise rules that are only active during the data cleaning phase or by running separate Manipula setups on the Blaise data when loading the case into the data cleaning system and creating new data cleaning issues for the case when needed.

2.2 Categories of Issues

In the data cleaning phase, the focus shifts to fixing specific issues. Issues to be looked at during the data cleaning phase may roughly be divided into four categories:

a) Interviewer comments/Blaise remarks

b) Known issues, like additional checks in the cleaning stage
c) New problems with the fielded CAPI questionnaire
d) Issues coming from the HelpDesk, reported by field interviewers after (partially) completing an interview
e) Issues discovered during the data cleaning phase by a Data Quality Control (DQC) Data Technician

Interviewer comments can be very useful, but at the same time they are very time consuming to handle, so we have developed some special ways of dealing with them. During data collection, instead of the standard Blaise remark screen, we bring up a Manipula dialog that asks for a category before the interviewer can make the comment itself, and then when the category is known, we remind the interviewer what specific information to include. This was described in a separate 2018 IBUC paper (Stegehuis, 2018).

We store the comment in a separate Blaise data file—with the category—but still use the Blaise remark paperclip for visibility of and easy access to the comment.

When the interview data gets received at Home Office, our DQC process will send these comments through a Natural Language Process to parse the comment itself and—separate from the category assigned by the interviewer—assign the most likely three categories for the comment. These categories are stored with the comment in the DQC issue that gets created and will be visible during data cleaning, so it is easier to assign the right specialist Data Technician to deal with the issue.

For category b), we have a folder where we have Manipula setups ready to run against all incoming data and run the necessary checks. Each issue that gets flagged during this process gets turned into its own data cleaning item in the SQL Server database that will guide the DQC stage.

In case any problems are found in the data, for instance, based on newly found CAPI program issues, as in category c) above, new Manipula setups can be added to the same folder. The overall system is set up to execute all Manipula setups in the folder, and flexibility lets us check all incoming cases automatically. Any discovered issues from these setups will be recorded in the DQC SQL Server database. This flexibility allows us to react very quickly to any new problems that may be found, even during the field period itself.

HelpDesk reports and issues discovered by a DQC Data Technician, categories d) and e) above, may be turned into DQC issues as well, so they can be addressed quickly.

3. Datamodel Changes

As mentioned before, one of the strengths of Blaise is that the same code can be used both for the original field interview and for the data cleaning stage. The two big advantages of that approach to cleaning are:

- Any changes made during the data cleaning stage will undergo the same routing and rules checking as the original interview, ensuring that no new data problems are introduced during the cleaning stage.
- Re-use of the Blaise code for this stage is a big cost saver compared to creating and maintaining a separate code base that uses different software, especially for complex surveys and for panel surveys.

However, there are differences in how we'd want to work with the Blaise application based on the difference in what the main goal is: completing an interview from start to finish versus fixing one issue, or maybe a few issues, in that entire case.

So, we do want to have one code base but also have slightly different behavior in field interviews than when used at the Home Office for data cleaning purposes. There are different ways to achieve that, but the way we have implemented this is by assigning a value to a datamodel-level auxfield on the command line when a data cleaning "interview" session gets started.

We will highlight just a few behavior changes that this use of the command line parameter enables.

3.1 Walls

The first main section of our questionnaire establishes the household composition, and at the end of the section we know, based on preload and the answered questions, who will be part of the remainder of the questionnaire. At the end of this section, we have programmatically erected a "wall," so that interviewers cannot back up after proceeding past it. This is a common strategy to ensure that the data collected in the main interview will not be invalidated by an interviewer backing up and changing the household composition later on.

In the data cleaning phase, however, it may be important to have access to those first questions, for instance, to change a typo in a person's name or correct a birth date or age. Determining whether a section gets an "ASK" or a "KEEP" in Blaise language based on an auxfield value is easy enough.

The challenge here is what needs to happen at the end of that first section, the code in the wall, if you will. In the field interview, this is the spot where the person roster, plus all that needs to be carried over from any preload data, gets put into place. During the cleaning phase, we do not want to overwrite that person roster (for the same reason we don't allow backing up during the field interview), so we have to check at the wall whether any changes were made during the data cleaning session that would have a negative impact. This is complicated code but is needed to ensure the quality of the data. If all is fine, the DQC Data Technician can proceed; otherwise, a warning will be put up detailing the problematic circumstance that has arisen and a decision needs to be made by project staff about what needs to be done.

3.2 Skipping a Section

An easier change in instrument routing during the data cleaning phase occurs when a section can be skipped as a whole, because it is outside the bounds of DQC staff to make changes there. An example of this is when we ask household members for signatures or to complete additional questionnaires, either online or on paper. This is only relevant during the field interview, and we do not need nor want DQC staff to ever make any changes here, so we use the auxfield parameter value in the datamodel rules to skip such sections.

4. Data Flow

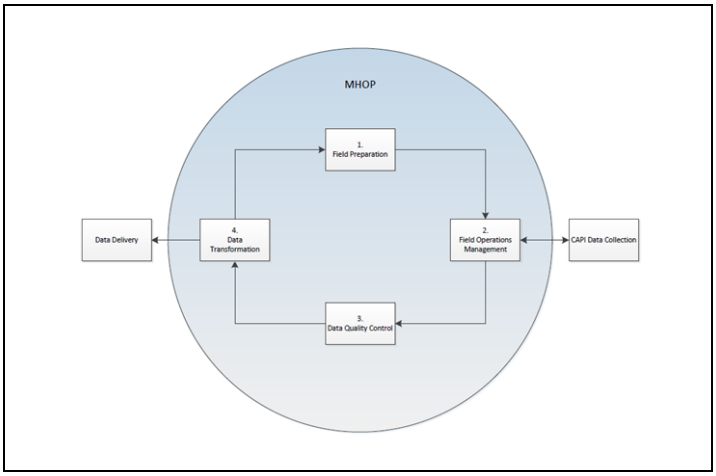
A somewhat simplified step-by-step description of the data flow and the place for data cleaning within it:

- Data transmissions with completed interviews come in from the field
- Daemon automatically picks up those transmissions and divides parts of the data to deposit them in the right place on the network (e.g. management data, CARI data, Blaise interview data)
- Load Blaise data for new cases into a consolidated database to always have the Blaise data exactly as it was before any data cleaning

- Run any interviewer comments (Blaise remarks) through a special process to determine the most likely category for each of them, making it easier for work distribution later
- Run any necessary checks on cases with Manipula setups

- Log any issues, either from these checks or from interviewer comments into the SQL Server database that is the backbone of our DQC (the "cleaning stage")

Figure 1. High-Level Overview of Data Flow for Our Survey



Of course, some cases will have issues that need to be cleaned, or at least looked at, and others will not need any additional scrutiny at this stage. In our experience, approximately three-quarters of cases sail through, meaning they have no interviewer comments and no issues from additional checks or from HelpDesk tickets.

Data Technicians work on the issues for the cases that do need attention using the Blaise CAPI instrument, with a special command-line parameter that is used to alter the DEP behavior for this stage where needed:

```
IF auxEditModeToggle = Yes THEN
  auxSlashQ:= auxSlashQ + 'EditMode=Yes;'
ENDIF
auxCommandLineString := auxCommandLineString + auxSlashQ + ' /X'
AuxRslt := EDIT(auxCommandLineString)
```

A straightforward skip of a section in this EditMode, or cleaning mode, would simply become:

```
IF EditMode <> Yes THEN
    RF_Main
ELSE
    RF_Main.KEEP
ENDIF
```

Note that we are using an auxfield for this designation. The value does not get saved at the end of the session, so that the desired behavior—interviewing or cleaning—gets set for just the current session and not for all future sessions.

5. Home Office System Integration

On interviewer laptops, we have an Interviewer Management System (IMS) that shows the field interviewers their assigned cases and tasks, one of which is starting the Blaise interview. When data gets transmitted back to the Home Office, parts of the data package go in different directions: Blaise CARI data goes one way, management data from the IMS goes to a central SQL Server database, and the Blaise CAPI data will go yet another way—to be loaded overnight into our DQC system.

Just like the IMS is controlling the field interviewers' options and tasks, the Home Office System is controlled by the management data in SQL Server. Subprocesses, like DQC, have their own SQL Server tables and statuses for tasks, but the overall control lies with the central system. This includes the overall status for each case, as well as status for additional items like separately collected (online or paper) forms.

The DQC system runs its own tasks and also checks the consistency of cases and status within the larger system.

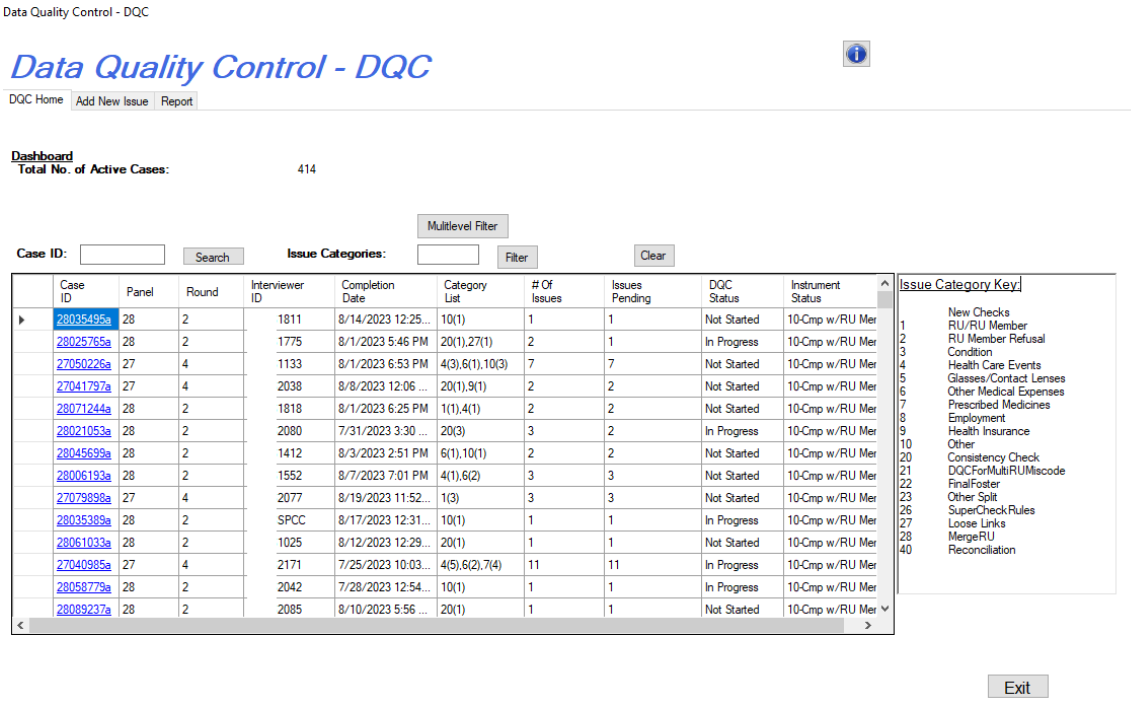
6. The User Interface

The DQC Data Technicians and DQC Supervisors start their work using our C# application that serves as the graphical user interface. It shows their workload based on role, and for each assigned issue, they can see the status overview plus a detail screen with the history of the issue and if needed, they can start the Blaise instrument from there.

After a Blaise data entry session, we run the same Manipula setups that were executed during the original loading of the case to ensure no new issues were introduced. They can close out issues and when a case has no more issues left can clear it for further processing.

When the user first starts the C# program, they will see the cases available to them.

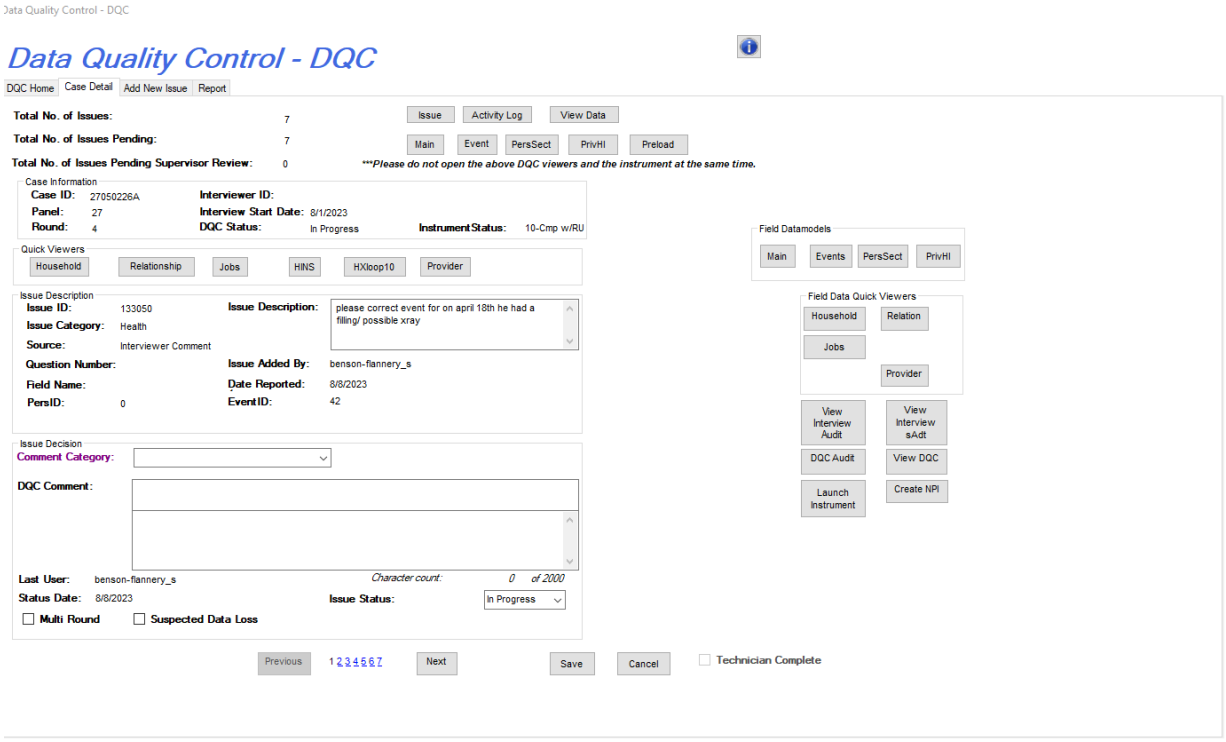
Figure 2. DQC Overview Screen



After selection of the case, the user will be taken to the Case Details Screen, as seen below. As you can see, this page allows the user to see any information we have about this specific case. In addition to the data provided on the page, the user can launch the Blaise instrument via a button click. The user may also select a data viewer to see the case data in its entirety or a predetermined subset of the data.

This screen is also where the user will enter any information about the case they feel is relevant to the issue. They can also record that all work on the case has been completed.

Figure 3. DQC Case Detail Screen



7. Conclusions

The use of Blaise for data cleaning as well as the original field interviewing can lead to a very powerful and efficient system, fitting seamlessly between fielding of cases and following tasks like data delivery and creating preload data for the next round of field interviewing.

Using scheduled automatic processes on (virtual) daemon machines, we can have a steady stream of new cases to work for our DQC Data Technicians and create a very efficient process using the same Blaise code for field interviewing and data cleaning.

8. References

Stegehuis, P., & Westat. (2018, October 22). *A different approach to Blaise remarks*. 18th International Blaise Users Conference, Baltimore.