

Exploration of Blaise Instrument Generation from Metadata

Fred Wensing, Juanita Pettit, Australian Bureau of Statistics

1. Introduction

Blaise instruments consist of source code which contain all the descriptive and definitional information about each data field (or question), as well as relationship information between data fields, such as sequencing, edits and derivations. Combined with layout attributes, the compiled source code becomes an instrument which can be used interactively or in batch mode to collect data, edit data, display data and apply amendments.

Instruments are often scripted from specifications, which define the instrument content, data flow, derivations and constraints (edits) on collected data. The specifications from which instruments are prepared can be considered to be the detailed metadata which defines the instrument.

The task of scripting an instrument involves transcribing the metadata content of each data element, adding various presentation standards and conventions, as well as program logic to define the instrument flow and behaviour.

If sufficient rigour were applied to the specification metadata, would it be possible to build a system that could generate the source code for Blaise instruments? This question is explored in this paper. It commences with a detailed analysis of the elements which make up an electronic questionnaire, and leads into the development of a prototype facility which is expected to deliver Blaise instruments from metadata.

2. Context

Blaise has been used at the Australian Bureau of Statistics (ABS) for the development of electronic questionnaires for population surveys since 1995. Since that time, virtually all population survey questionnaires at the ABS have been converted from pen-and-paper mode to computer assisted interviewing (CAI).

The early development of processes to support CAI for population surveys at the ABS had a focus on Case Management (Wensing and Martin, 2001) and defining standards for screen layout (Wensing, Barresi and Finlay, 2003). These were obviously essential for the successful implementation of CAI and took priority over any plans for generation of instrument source code.

For instrument preparation, early emphasis was placed on conventions and programming standards to ensure that there was consistency between instruments, and to ensure that the source code was well managed. The Blaise source code was mostly scripted by hand but drew on standard instrument shells and shared modules. Some developments were undertaken to create utilities to assist with instrument assembly and release (Wensing, 2004) but otherwise there was no generation of source code.

To support the transition from the pen-and-paper forms used in population surveys to electronic forms, a repository was developed to record the specifications of data items and survey questions in a structured format. The main purpose of the repository, known as the Survey Development Tool, was to organise the specifications and make them readily available to all persons working on the survey. The repository was widely used but remained rudimentary, with no particular features for generation of source code. A replacement facility is currently under development and is expected to provide some support for code generation in the future.

Since 2003, Blaise instruments have also been used in business surveys at the ABS for computer-assisted telephone interviewing (CATI) for data collection and sample maintenance (Farrell, 2006). More recently, Blaise has also been used in business surveys to provide a mechanism to apply edits to collected data and to provide an interface to assist with the investigation and resolution of failed edits (Wensing, 2006).

The specifications for the data elements to be included in business survey instruments are prepared in database facilities or spreadsheets. In the same way as for population surveys, the specification facilities for business survey instruments did not include features for generation of source code. Once again, the Blaise source code was mostly scripted by hand, drawing on standard instrument shells and some shared modules.

Over the past decade the ABS has developed elements of a Corporate Metadata Repository which will provide a central store containing the definitions of the data elements used in all statistical collections. The repository is largely based on *International Standard ISO/IEC 11179: Information Technology - Metadata Registries*.

In conjunction with the developments associated with the Corporate Metadata Repository, a project was sponsored to examine the feasibility of supporting Blaise instrument code generation from data element definitions and other metadata. An expected output of the project was a prototype system and explanatory material to prove the concept of generating Blaise instruments from metadata. Aspects of that project and prototype system are described in this paper.

3. Why generate instrument source code?

The act of scripting a Blaise instrument involves taking the detailed specifications, assigning names and types to field definitions, transcribing the specified texts into the field definitions, adding text formatting, and then adding structure, sequence logic, derivations and edits. The work involves knowledge of the Blaise language syntax, knowledge of local programming practices and code storage, familiarity with instrument design and the systems into which it will be deployed, as well as some understanding of the subject matter being covered.

The main reasons for contemplating the generation of source code are:

- to take advantage of infrastructure developed to manage specifications;
- to avoid duplication of effort;
- to minimise the possibility of error;
- to give more control over the content to survey designers;

- to ensure that specification and instrument are consistent with each other;
- to maximise reuse of data element specifications;
- to enable speedy update of an instrument after changes to specifications;
- to facilitate the implementation of standards; and
- to save time.

Code generation will also free the survey expert to concentrate on content and design issues rather than the technical scripting process.

The concept of Blaise instrument code generation is not new and other agencies are also working on this. It could be that the generator just does the initial copying of static text into an instrument structure with further change(s) being made directly to the source code (Egan, 2003). Alternatively, the generation could be achieved by using a specification interface which guides the instrument development one question at a time (Vreugde, de Groot, and van 't Hof, 2003). A more systematic approach would be to have a metadata store which records the attributes of many data elements and provides an interface to enable users to select required elements and generate the code for any instrument (De Bolster, 2004). It is this latter approach which more closely aligns with metadata developments at the ABS.

4. Analysis of questionnaire metadata

In order to understand the complexity of instrument code generation, analysis was carried out to define the full extent of metadata that could be needed to support a questionnaire. This analysis was based on current data modelling for a Corporate Metadata Repository supplemented by current practices in questionnaire construction for both population surveys and business forms and Blaise instrument development practice.

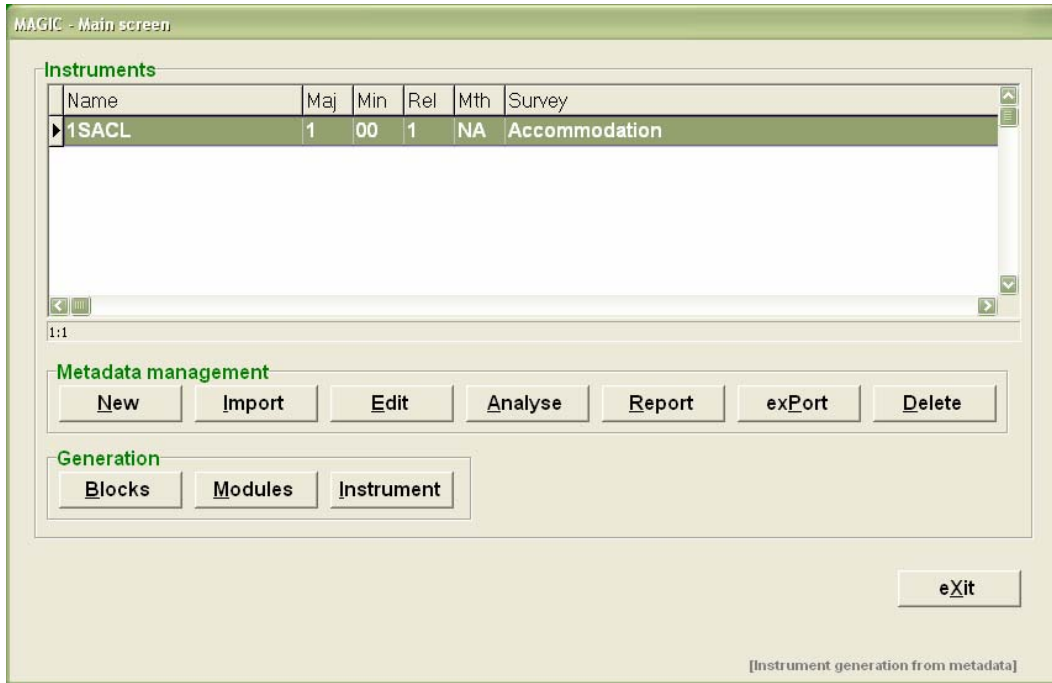
Given that one objective of the project was the generation of a Blaise instrument, a logical starting point for gathering the metadata was the Blaise datamodel and all its elements. This approach also ensured that the Blaise elements and behaviour were accommodated.

Figure 1 shows the data model developed to represent the basic metadata needed to display the questions for an electronic instrument. The basic metadata is restricted to the data elements, questions, text construction, type definitions, response sets and display features. Control events, such as conditional routing, derivation and the application of edits are to be given more attention in the next iteration of the analysis.

All questions and other fields in an instrument are related to Data Elements which are expected to be registered in the Corporate Metadata Repository. Due to ABS questionnaire design practice, particularly for population surveys, the main building block of a questionnaire (or instrument) is the question module. The use of question modules makes it easier to define self-contained sets of fields with their own internal relationships. A question module often translates to a block of source code in the Blaise instrument making this a useful technical distinction as well.

metadata would at least assist in the instrument preparation process through this system. Figure 3 shows the main control screen for the MAGIC prototype.

Figure 3. Main screen of the prototype code generator



The main underlying feature of the prototype system was a table of metadata element definitions. Each row of this table corresponds to one instrument element which in turn may relate to a real data element or to a text object that is displayed on the instrument screen. The columns of the metadata table identify the various attributes that describe each element. Figure 4 shows a list of some of these attributes.

Figure 4. List of attributes recorded for each instrument element

Attribute	Description
System name	A unique system name for the data element
Description	Description of this element (optional)
Question number	Question number (text)
Element type	Type of data element (free text, containing key words relating to type)
Element code	Code letter(s) representing the type of element (e.g. Positive integer; Enumerated; Date etc)
Type specification	Block specification for this type
Minimum value	Lowest value for numeric data
Maximum value	Highest value for numeric data
Category definition	Category definition to use with this data element
Display method	The display method to be used for this element: Ask, Show, Keep
Column	The main column to be used for this element
Lines	Number of line spaces to provide for text
Block name	The block presentation to be used for this element (derived unless a selection is made)
Prefix	Unit of measure to be used as a prefix

Attribute	Description
Suffix	Unit of measure to be used as a suffix
Header	Header for question text
Main text	Main text for the question
Instruction text	Instructions to be displayed
Help text	Help text to be displayed
Alternate text	Alternate text to be used for question

In order to manage the instrument metadata table some tools and functionality were added to the prototype MAGIC facility. These included:

- interface to record the identity and main characteristics of a new instrument;
- utility to import metadata from a comma-separated text file, with flexibility to identify which columns relate to particular attributes;
- ability to edit the attributes for individual questionnaire elements and make corrections or adjustments (see Figure 5);
- utility to analyse the questionnaire elements and produce a summary report of the key attributes;
- utility to export metadata to a comma-separated text file.

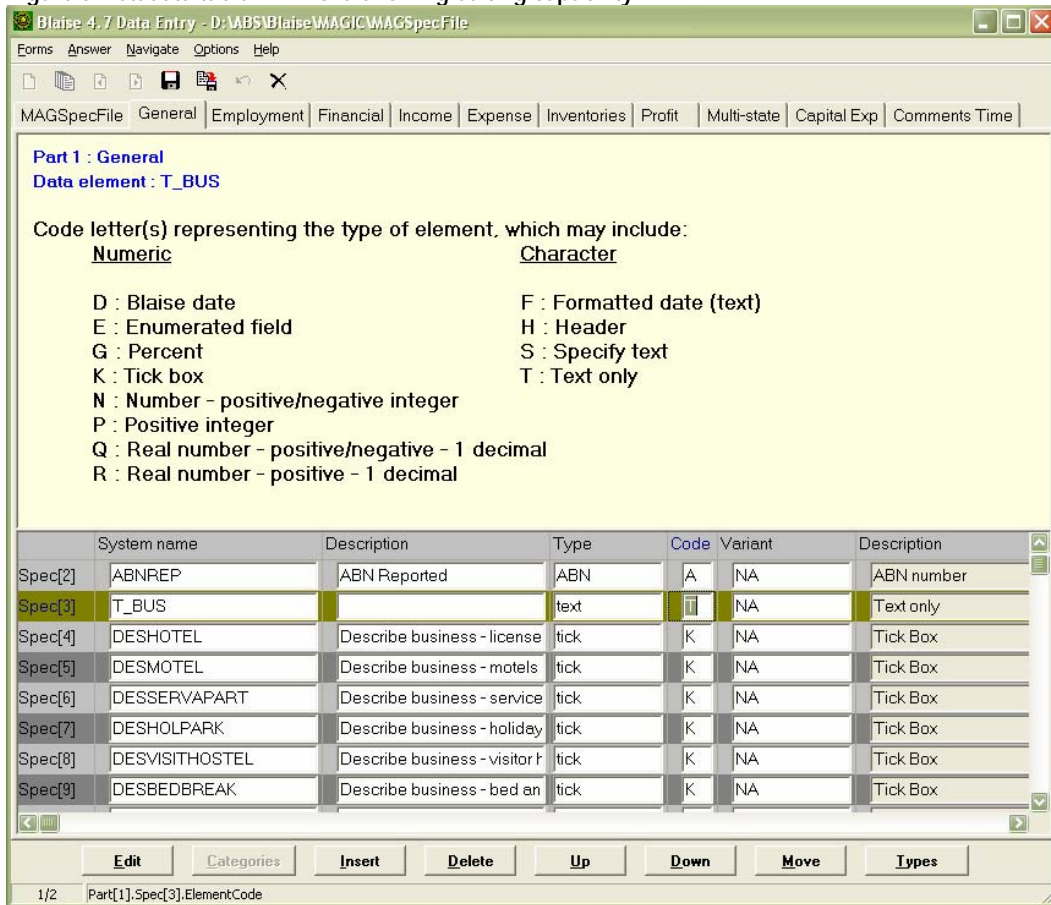
Initial experimentation with the prototype facility looked at supporting the production of business survey instruments to be used in data editing. There were a number of reasons for this:

- the instrument structure used for business survey instruments was well defined, consisting of standard control fields, and a number of sections consisting of a table structure and data elements in columns and rows;
- a shell instrument containing standard control fields and basic operational source code already existed, providing a good starting point for instrument generation;
- question texts in business survey instruments tend to be static (as for paper forms);
- the order of questions in many business survey forms do not require conditional routing to be applied, making it possible to postpone dealing with this issue;
- the specifications of business survey data elements were often prepared in a spreadsheet making it easier to import the metadata.

Requirements for CATI instruments were investigated but project resources were exhausted before generation of CATI instruments could be included in the prototype. A shell instrument was also not available for immediate use as was the case with editing instruments.

A complimentary tool to build template blocks for use in business survey instruments was linked to the MAGIC prototype so that the operator could define new (template) blocks if required. This tool, also written in Maniplus, was developed previously and delivers blocks of code that can be used to display a pattern of data elements in a 'row' on the screen. When applied, each block uses parameters to accept text strings that are part of the row display. Such blocks can therefore be used multiple times in the same instrument.

Figure 5. Metadata table in MAGIC showing editing capability



Once the metadata has been examined and adjusted, and required template blocks are defined, the three-stage generation process can be activated. Firstly, the required template blocks are generated, then the source code for the sections, and then the source code for the main instrument. The last stage draws on the above-mentioned shell instrument containing the standard control fields and basic operational code and inserts the additional source code to identify and activate the sections of the instrument. This final step also makes use of techniques that were previously developed to assist with instrument assembly at the ABS (Wensing, 2004).

6. Discussion and conclusions

The MAGIC prototype has been tested by current instrument programmers at the ABS and has been shown to successfully generate the Blaise source code for business survey (editing) instruments from metadata. It was possible to do this by importing the original specifications (prepared in a spreadsheet) as metadata into the MAGIC prototype. Through the iteration of analysis, adjustment of the metadata and activating the generation steps, successive versions of the same instrument could then be produced. An important feature of the prototype facility was the ability to modify and manipulate the metadata.

The immediate benefit of using code generation, even with this prototype, was the productivity gain for preparation of a first-cut instrument. Furthermore, the ability to

regenerate the instrument following changes to the metadata was also seen as a time-saver.

While the generated instruments were operational, they represented only about eighty percent of the finished product. Additional hand-written code needed to be added in order to provide the edit functionality which these instruments required. It is intended to add functionality to the MAGIC prototype to enable the hand-written code to be extracted from an existing instrument and transferred to the next iteration of that instrument. This functionality will once again draw on the techniques previously developed to assist with instrument assembly (Wensing, 2004).

The formatting of text for screen presentation was largely handled through the application of ABS standards and conventions for screen layout incorporated in the Blaise mode library component. The application of standard formatting was assisted by ensuring that the question text was broken down into small enough parts (or subtexts) each of which have their own standard format. The only way to add text highlighting to particular words within a question, however, was to use the Blaise language mark-up symbols within those texts which may be a problem if the metadata is to be used for other purposes such as the production of paper forms. This problem may be alleviated when the Blaise software enables the use of more conventional mark-up tags (e.g. HTML) within instrument texts.

Although the prototype only supports one mode of instrument at this time, namely editing, consideration has been given to the extent that instruments used in other modes of operation (e.g. CATI) may be generated from the same metadata. While the general layout aspects of different modes could be handled by changing the instrument structure and some of the formatting components, the mode differences may in fact require different wording or question format for the same data element (Pierzchala, 2006). Conversion of questions from an editing instrument (based on a self-administered paper form) into questions that are suitable for CATI is not straight-forward (Farrell, 2006) and will therefore work against the hope of using the same metadata.

There are some other aspects of instruments such as conditional routing and context-specific text fills that have not been tackled in the development of this prototype. These aspects are important for CAI and CATI instruments and will also need to be explored fully before such instruments could be generated.

In conclusion, the development of this prototype facility has demonstrated that it is possible to generate basic instrument source code from appropriately defined metadata, even though some hand-written source code may need to be added. The generation of instrument code is assisted by having standardised instrument designs and some pre-programmed questionnaire elements. A good, flexible metadata management facility is also fundamental to the success of any code generation system. Handling the construction of instruments for different modes of operation and conditional routing will be the next challenge.

7. Acknowledgements

The authors would like to acknowledge the ABS Forms Consultancy Group who sponsored and provided valuable input to the prototype development.

8. References

- Wensing F., Martin B. (2001): Case management for Blaise Using Lotus Notes, Proceedings of the 7th International Blaise Users Conference, Washington DC, USA, September 2001.
- Wensing F., Barresi J., Finlay D. (2003): Developing an optimal screen layout for CAI, Proceedings of the 8th International Blaise Users Conference, Copenhagen, Denmark, May 2003.
- Vreugde C., de Groot J., van 't Hof C. (2003) : Blaise Survey Generator, Proceedings of the 8th International Blaise Users Conference, Copenhagen, Denmark, May 2003.
- Egan M. (2003): CodeBuilder2 – An Initial Coding Tool, Proceedings of the 8th International Blaise Users Conference, Copenhagen, Denmark, May 2003.
- De Bolster G. (2004): Generating Blaise with Blaise, Proceedings of the 9th International Blaise Users Conference, Gatineau, Québec, Canada, September 2004.
- Wensing F. (2004): Instrument assembly, documentation and release, Proceedings of the 9th International Blaise Users Conference, Gatineau, Québec, Canada, September 2004.
- Farrell E. (2006): Design and testing of CATI instruments for Business Surveys, Proceedings of 2nd Telephone Survey Methodology Conference, Miami, Florida, USA, January 2006.
- Wensing F. (2006): Using Blaise to Apply Edits to Data held in an Input Data Warehouse, Proceedings of the 10th International Blaise Users Conference, Arnhem, Netherlands, May 2006.
- Pierzchala M. (2006): Disparate Modes and Their Effect on Instrument Design, Proceedings of the 10th International Blaise Users Conference, Arnhem, Netherlands, May 2006.
- <http://www.metadata-stds.org/11179/> : ISO/IEC 11179, Information Technology - Metadata Registries.