A Short History of Blaise Code Generation

Leif Bochis Madsen, Statistics Denmark

1. Abstract

The history of Blaise is also a history of finding easy ways to make specifications for questionnaires without needing to write Blaise code—either because the subject matter specialists who are designing the surveys are not familiar with the writing of code or because the writing of code is simply a tedious and unwelcome task. Besides, many surveys originate from specifications already described in, for example, database tables or metadata repositories—and nobody wants to write similar specifications twice in different syntaxes.

This paper describes various ways of generating Blaise code from external sources—or via alternative ways of describing questionnaires—primarily from work done in Statistics Denmark, but with reference to experience from other organizations. It includes a discussion of benefits and limitations of the different approaches.

2. Introduction

Recently, I had the pleasure of introducing a group of IT staff members from the Jordanian Department of Statistics to the Blaise Survey system and showed them how to use Blaise to implement questionnaires.

When I started to make some simple examples of Blaise code and continued to show the famous Flight Survey demo code in order to exemplify all the nice features and the structure of Blaise code, I was met with a polite, but unmistaken, disappointment.

It was clear that an easier way of "putting the questions into Blaise" was requested, and I remembered the old fact that IT professionals don't want to write code when it is so much easier to (make other people) generate it.

Luckily, I could—after a few hours of work—get back and successfully demonstrate one of our Blaise Code Generators picking a Word document and nicely producing a Blaise data model ready to prepare.

In the end, I was asked if we could provide the generator for use in the Jordanian Department of Statistics, and I promised to do so, of course, while also starting to think of how much work it would take to liberate the tool from our own IT environment.

Afterwards, I started thinking about all the various Blaise Code Generators that people wrote in order to ease "putting questions into Blaise" and realized that a summary of these efforts and approaches to the subject would be a good topic for an IBUC paper and presentation.

3. History of Blaise Code Generators in Statistics Denmark

The history of generating Blaise code in Statistics Denmark goes back to the past century.

A variety of attempts to make the "right" and "(almost) complete" generator comprise a long list of approaches and tools:

At first, various Excel spreadsheets were used to define the questions and answers, with a number of different structures in order to get as much metadata as possible described. Scripting languages, such as VB or Manipula, were used to generate Blaise code. Merely copying the contents of a sheet and pasting it into a Blaise source code would also make sense.

Secondly, a structured Word document supplied with macros could form the basis of a Blaise questionnaire.

A third and more successful attempt is the Word-based Blaise Code Generator for Household Surveys, which is still used. It is described further in the next section.

The fourth generator is the so-called Business Survey Configurator, also described below.

Finally, attempts have been made to build a Blaise Code Generator in Blaise (i.e., to define a data model describing a Blaise questionnaire) using the data entry program to describe blocks, fields, datatypes, etc., and using Manipula scripts to generate a target Blaise data model. This generator worked fine as a proof of concept, but due to lack of interest among the targeted users, it was never finished or used for any production.[1]

4. Blaise Code Generator for Household Surveys

The aim of this code generator is to provide an easy-to-use way to describe questionnaires. Thus, the description tool is merely a Word document (or any text-processing document that can be converted into Word) and is aimed at any customer (internal or external to the organization) with limited or no knowledge of Blaise.[2]

It defines a simple "description language" based on tables (blocks), rows (fields), and columns (name, question text, type name, type definition, filter condition, and comments).

Figure 1. Fragment of Sample Word Document (from [Madsen et al., 2013])

BlockB	Discrimination			Citizenship <> Danish	The questions in BlockB are not asked of Danish citizens
IntroB	The following questions deal with various types of discrimination you may have experienced in Denmark due to your ethnic background				
B1	Within the past year, have you experienced, because of your ethnic background, being denied access to places which other people were allowed to enter? (such as a bus, taxi, nightclub or swimming baths)	YNDR	1 Yes 2 No 3 Do not know 4 Prefer not to answer		

[1] I presented this generator at the BCLUB meeting in Copenhagen, January 2016, under the code name "Blaise Monkey." I might be able to dig up some remains of the code if someone is interested. I was inspired by [Vreugde, 2003].

[2] [Madsen et al., 2013] contains a detailed description of this code generator.

An automated process carries out the following tasks:

1) The Word document is converted into its XML equivalent.
2) The XML document is extracted by an XSLT process, generating Blaise code.
3) Standard templates suited for the overall system architecture are applied.[3]
4) A version of the data model without rules is prepared in order to facilitate the checking of filter instructions (the rules part is the difficult part of the data model to produce correctly).
5) This version of the metadata is applied in a process that generates valid rules instructions.
6) Often, a preparable (though not necessarily complete) source code is the result of this process.

The generator was built using XSLT, C#, and Manipula for different processes. It has been slightly maintained over the years (e.g., updated to possibly produce Blaise 5 code) and is considered a "90 percent generator" (approximately).

This tool has met the primary aims of being an easy-to-use tool for describing questionnaires and is still in use in our household surveys division. It is mainly efficient for the generation of first versions of questionnaires, ready for refinements like layout and more complex rules.

5. Business Survey Configurator

The Business Survey Reporting System started in 2006 using Microsoft InfoPath as a questionnaire development tool. Due to the end of support for this tool, from 2018, Statistics Denmark started examining and later migrating Business Reports to Blaise 5. In the beginning, we were using the Word/table-based generator to generate Blaise 5 code, but soon we realized that we needed to address compatibility between the XIS[4] repository and the Blaise instrument.[5]

The aim of the Business Survey Configurator is to generate a Blaise project ready to prepare—and compatible with the metadata from the XIS repository. Field names in the XIS repository (as opposed to Blaise) are treated as case sensitive, so in order to automatically fetch data from a Blaise database via its metadata and store these data in XIS, it is important that fields in the Blaise instrument use the same casing as in the XIS database.

Because the XIS repository contains relatively poor metadata—and therefore, only a poor Blaise instrument could be generated directly from the XIS repository—the Configurator was developed, so supplementary metadata could now be added to the code generation. These metadata comprise auxfields, role texts, and precise data types (the XIS repository only defines the basic datatypes integer, real, and string).

Due to the limited functionality implemented in the Configurator, we are talking about a one-way generator, and it is only used the first time for an instrument. Needed modifications (e.g., layout instructions) are all carried out with the Blaise Control Centre. When subsequent changes need to be implemented for repeated or periodic surveys, maintenance will be done via the existing Blaise solution.

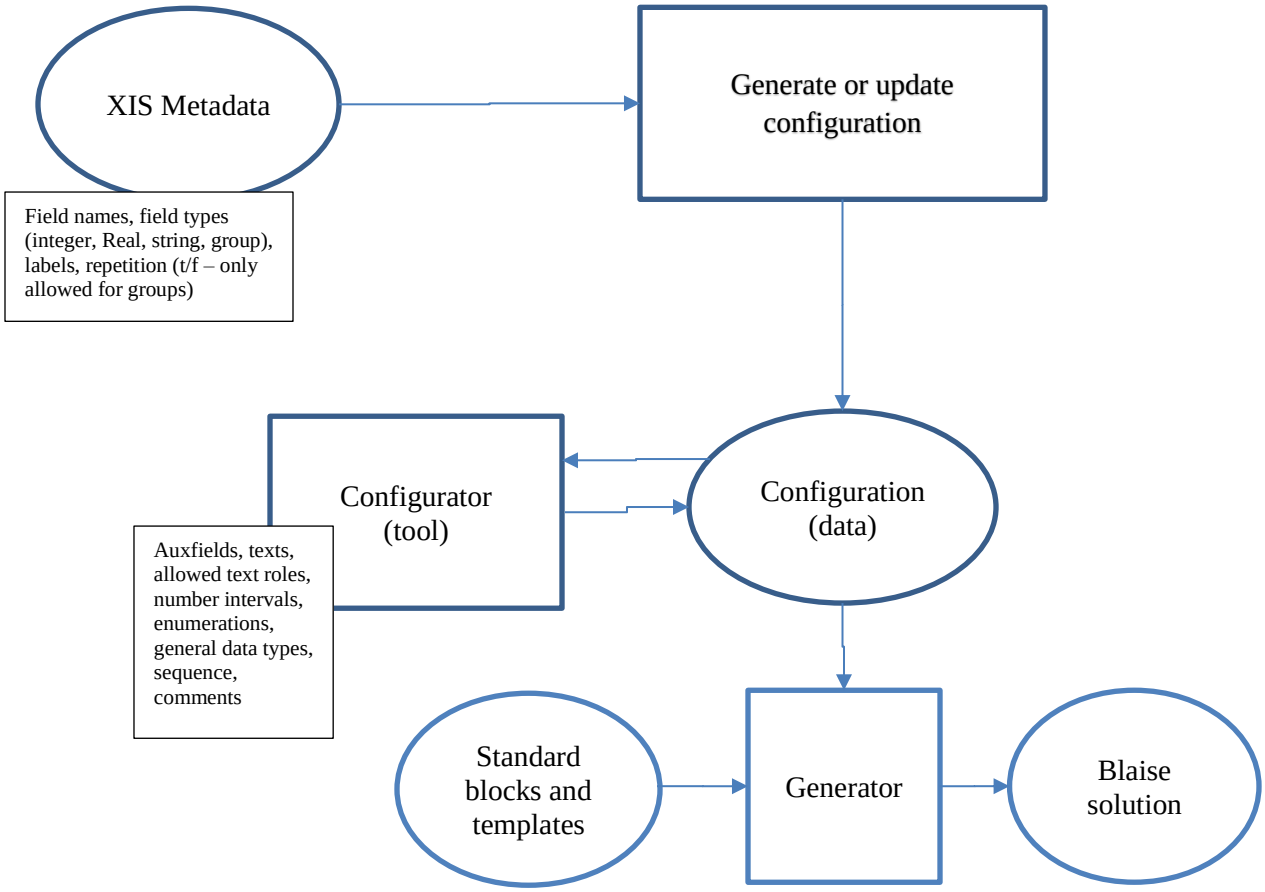
[3] This comprised a template for a standard blax file for Blaise 5 and included standard blocks for taking care of, for example, preloaded data about the survey and/or respondent. Thus, it is possible to use this generator for different setups by changing a reference to the templates.

[4] XML Input System (XIS) is a general system for storage of business survey data at Statistics Denmark.

[5] The Business Survey Framework is described in detail in [Madsen, 2018], along with some related tools in [Madsen, 2020].

Possibly, if a large number of fields or entire blocks are added, a new version may be generated and new elements copied into the existing Blaise solution.

Figure 2. Process Diagram for the Business Survey Generator



Soon, the problem of synchronization between the metadata in the configuration and the actual metadata in the instrument became visible and important to handle.

In order to facilitate use of the Configurator as the first choice of description—and as maintained storage of survey documentation—we launched an experiment. The experiment was based on the idea that we could automate the process of adding new elements and retain the changes (primarily rules and layout instructions):

1) Generate a new solution and prepare it.
2) Extract the metadata as XML.
3) Extract the metadata from the previous solution as XML.
4) Compare the two XML structures, add new elements, remove elements (fields and blocks only) not present in the new version, and let existing elements stay unchanged (including rules).
5) Generate new source code from the changed XML document (via XSLT process).

Actually, we succeeded in making a useful source code that could replace the previous version, with rules and layout instructions concerning the previous questions retained. Proof of concept—with one exception: All comments are removed from the source due to steps (2) and (3) because they are not part of the metadata.

Keeping the comments would require that they are included in the metadata (e.g., by adding annotations as a text role), but that would not help with keeping general comments attached to rules instructions (i.e., apart from conditions and checks). Thus, this grand idea still needs some further care in order to be practically useful.

All in all, the Configurator and Generator make a good beginning, and the instruments may be refined further by editing rules and adding layout instructions in the Blaise Control Centre. It has been an important tool in order to create Blaise instruments to replace Infopath report forms during our conversion process. It is, however, not clear how important it will be in the future, where the focus will instead be to maintain and make changes to new versions of the mainly periodically repeated business surveys.

6. Blaise Colectica Questionnaires (BCQ)

The combination of Colectica and Blaise has gained popularity among NSIs in the recent years. Many organizations have a policy of documenting their data in DDI, and Colectica offers a user-friendly access to DDI.

At Statistics Denmark, we have one example of using Colectica-generated questionnaires, as we designed a questionnaire in 2018 for the survey *European Statistics Code of Practice* (COP) in Colectica and generated the Blaise code. However, a lot of changes had to be carried out afterwards, especially concerning constructs for improving layout. The COP survey is annual, and minor changes need to be implemented for each version. We realized that the easiest way to accomplish this was to adapt these changes by editing the Blaise source in the Blaise Control Centre, and that is how we have maintained the COP questionnaire since the first version.

Apparently, other users of BCQ have the same experience using BCQ to generate the first version and implement subsequent changes in BCC, though it requires double editing—in Colectica as well as in BCC [CSO Ireland, 2023].

This autumn, however, we have initiated a project with the aim of modernizing the Labour Force Survey (LFS). Among the goals are replacement of older software and legacy code (e.g., replacing Blaise 4 with Blaise 5) and usage of Colectica/DDI for documentation. One of the tasks of this project is about defining design features in BCQ that may allow us to regenerate LFS questionnaires for every new data collection period.

7. Conclusions

Many have tried to find their best and most efficient way to generate Blaise code. For example, the data collections department of Statistics Netherlands uses a very impressive generator based on a Word document comprising descriptions of questions and a Visio document containing a graphical representation of the rules, including conditions.[6]

[6] Described by [Bolster, 2009], [CBS Netherlands, 2023], and [Pisiren, 2023].

But also, there are many others, as suggested by the vague selection of papers about the subject in the references.

Most successfully, probably, are solutions with emphasis on the local culture, language, and environment. Also, reuse of already available data from different kinds of data sources, including existing metadata or repositories, is an important foundation for efficient generation of instruments. Existing frameworks for processing survey data may be necessary to take into account, and cultural views or design traditions may have impact on the success of a solution.

Maybe the Blaise Colectica Generator will be a good choice for the future of Blaise Code Generation with its firm base on the DDI standard. However, not all organizations are using Colectica, so there might still be a need for other kinds of generators.

In general, Blaise Code Generators tend to be deeply rooted in one organization—its culture, language, design traditions, and habits of documenting and storing metadata. This is definitely also true about the generators we developed in Statistics Denmark.

Therefore, the implementation of a generator in an organization is a delicate task, and a number of questions should be taken into account when doing it. Among the questions to address are:

1) Who is going to use the generator and how much knowledge of the survey tool (i.e., Blaise) do they need to possess?
2) Level of ambition: How much of the code needs to be generated automatically?
3) Are (some of) the data already described in a repository of some kind?
4) Which kind of "description language" can we provide in order to define a questionnaire?
5) How much time may the user spend in order to learn how to describe a questionnaire?
6) How can we automate processing from a description to (part of) a solution?
7) How would we manage regeneration in case of repeated surveys?

The references include a noncomprehensive list of papers from recent Blaise conferences dealing with this topic.

8. References

[Bolster, 2009] BlaiseIS at Statistics Netherlands, Gerrit de Bolster; Statistics Netherlands, in: *Proceedings of the 12th International Blaise Users Conference*, Riga 2009, https://blaiseusers.org/2009/papers/4d.pdf

[Bolster, 2013] Generating Blaise from DDI, Gerrit de Bolster; Statistics Netherlands, in: *Proceedings of the 15th International Blaise Users Conference*, Washington DC 2013, https://blaiseusers.org/2013/papers/3a.pdf

[CBS Netherlands, 2023] Unrecorded presentation on Zoom, May 10th, 2023

[CSO Ireland, 2023] Unrecorded presentation on Zoom, June 6th, 2023

[Filippenko et al., 2007] Questionnaire Specification Database for Blaise Surveys, Lilia Filippenko, Joe Nofziger and Valentina Grouverman; RTI International, USA, in: *Proceedings of the 11th International Blaise Users Conference*, Annapolis 2007, https://blaiseusers.org/2007/papers/E3%20-%20Questionnaire%20Specification%20Database.pdf

[Filippenko et al., 2013] Web-based CAI System for Blaise Instruments Development, Lilia Filippenko and Sridevi Sattaluri; RTI International, in: *Proceedings of the 15th International Blaise Users Conference*, Washington DC 2013, https://blaiseusers.org/2013/papers/3d.pdf

[Joyal et al., 2013] Blaise Code Generator, Eric Joyal, Jason Gray and Sam Threinen; Statistics Canada, in: *Proceedings of the 15th International Blaise Users Conference*, Washington DC 2013, https://blaiseusers.org/2013/papers/3c.pdf

[Madsen et al., 2013] Automatic Generation of Blaise Data Models, Leif Bochis Madsen, Saliha Zayoum, and Lars Peter Jørgensen; Statistics Denmark, in: *Proceedings of the 15th International Blaise Users Conference*, Washington DC 2013, https://blaiseusers.org/2013/papers/3b.pdf

[Madsen, 2018] Implementation of Blaise 5 Web Questionnaires into an Existing Business Survey Management System, Leif Bochis Madsen; Statistics Denmark, in: *Proceedings of the 18th International Blaise Users Conference*, Baltimore 2018, https://www.blaiseusers.org/2018/papers/5_5.pdf

[Madsen, 2020] Managing a Complex Infrastructure for the Collection of Business Report Forms, Leif Bochis Madsen; Statistics Denmark, in: *Proceedings of the 19th International Blaise Users Conference*, Limassol (Zoom) 2020, https://blaiseusers.org/2020/papers/IBUC2020_S2_4.pdf

[Ostergren, 2013] A New Tool for Visualizing Blaise Logic, Jason Ostergren; University of Michigan, Survey Research Center, in: *Proceedings of the 15th International Blaise Users Conference*, Washington DC 2013, https://blaiseusers.org/2013/papers/8c.pdf

[Pisiren, 2023] Coskun Pisiren: Presentation at BCLUB meeting. London, May 2023 (available from BCLUB basecamp site)

[Seymour, 2010] Moving to a Metadata Driven World, Chris Seymour, Statistics New Zealand, in: *Proceedings of the 13th International Blaise Users Conference*, Baltimore 2010, https://blaiseusers.org/2010/papers/3a.pdf

[Vakhaetov, 2010] MetaDEx - From Data Dictionary to Complete Blaise Data Model Code Generation, Farit Vakhaetov, Department of Population Health Research, Alberta Health Services - Cancer Care, in: *Proceedings of the 13th International Blaise Users Conference*, Baltimore 2010 (abstract), https://blaiseusers.org/2010/slides/p3c%20Vakhaetov%20MetaDEx.pdf (presentation)

[Vreugde, 2003] Blaise Survey Generator, Carlo J.C. Vreugde, Jacques J.M.J. de Groot, Christiaan van 't Hoft, VNG, SGBO, StimulansZ, of the Netherlands, in: *Proceedings of the 8th International Blaise Users Conference*, Copenhagen 2003, https://blaiseusers.org/2003/papers/Blaise_survey_generator.pdf

[Wensing et al., 2007] Exploration of Blaise Instrument Generation from Metadata, Fred Wensing and Juanita Pettit; Australian Bureau of Statistics, Australia, in: *Proceedings of the 11th International Blaise Users Conference*, Annapolis 2007, https://blaiseusers.org/2007/papers/F3%20-%20Exploration%20of%20Blaise%20Instrument%20Generation%20from%20Metadata.pdf