Introduction

– Since the beginning of this year Colectica and Statistics Netherlands have entered into a long term partnership to build software linking Blaise, Colectica, and the DDI Lifecycle standard

– During this presentation some background will be given on Blaise in relation to DDI and some details will be given on what is being worked on
Blaise

- Blaise is a package for computer-assisted data collection and survey processing
- It is being developed by Statistics Netherlands
  - Not for profit organization
  - SN is also a big user of Blaise (eat your own dog food)
  - This year we celebrated the 30th anniversary
- Blaise is used both for social & business surveys
  - But most often used for (very complex) social surveys
- Two versions of Blaise are being used
  - Blaise 4 and Blaise 5
Blaise in a nutshell

– The Blaise datamodel language (a DSL) is used to describe the survey instrument
  - Very complex instruments can be defined
– The prepared survey instrument definition is used to run the system
  - Metadata driven
– Data storage definition generated automatically
– Metadata and data accessible using Blaise tooling or Blaise components (API’s)
  - Offers possibility to integrate Blaise into other systems
– Extensive paradata (data about the data collection process) is available
DDI and Blaise

- DDI is much more than Blaise
- DDI does not cover all survey related information that Blaise has to offer
  - As DDI evolves perhaps the intersection can become larger
From Blaise to DDI (1): MQDS

– MQDS (Michigan Questionnaire Documentation System) can export Blaise meta data to DDI
  - MQDS reads Blaise metadata & data using Blaise API’s
  - It can generate DDI 3.1 compliant DDI output
– Computes GOTO’s and text fill’s.

– For more info on MQDS see the following paper presented at the 17th International Blaise Users Conference (IBUC), The Hague, NL, 2016:
  The Michigan Questionnaire Documentation System (MQDS) Version 5 Update
From Blaise to DDI (2): Colectica

- Colectica Designer can import an existing Blaise survey instrument and once imported, then export to DDI (2.x or 3.x).
- Import is based on parsing of the Blaise survey instrument source code
  - Using the instrument source code is not the most optimal solution: using the API is to be preferred
- Colectica Blaise to DDI Metadata Converter is also available as open source
From DDI to Blaise

- Requires a Blaise source code generator that uses a DDI instrument definition as input
- Various implementations within the Blaise user community that generate Blaise source code, some DDI related, some not:
  - RTI, INSEE, ABS, Statistics Netherlands, ...
- Colectica Designer can be used to export a Survey Instrument definition to Blaise 4
  - One (major) drawback: it cannot be guaranteed that it is a valid Blaise 4 survey instrument
For Blaise to be successful with DDI:
- Requires in-depth knowledge of DDI
  • DDI is not simple and straightforward
  • Where to store Blaise concepts that do not have a logical / obvious place in DDI?
- When DDI evolves, so must the provided links to DDI

We decided to search for a partner and found one: Colectica

Plus: Statistics Netherlands decided to become a member of the DDI Alliance
The following products are being developed:

- **Blaise Colectica Questionnaires**
  - To generate a valid Blaise datamodel
  - Wizard to define conditional routing expressions

- **Blaise-DDI Mapping Paper**
  - To describe in detail how each Blaise metadata element is mapped to DDI

- **Blaise Colectica DDI Connector**
  - To export Blaise metadata
Benefits for Blaise

- Visual Survey Designer
  - As an alternative for programming
  - Easier for non-programmers / less training required
- Question Bank / Colectica repository
  - Re-use!
  - Import questions from existing Blaise instruments
- Documentation and Reports
  - PDF, publish metadata on the web, ...
- DDI Standards Support
  - Becomes more and more a requirement
## Blaise-DDI mapping

### High Level Mapping

<table>
<thead>
<tr>
<th>DDI</th>
<th>Blaise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument</td>
<td>DATAMODEL</td>
</tr>
<tr>
<td>Sequence</td>
<td>BLOCK</td>
</tr>
<tr>
<td>StatementItem</td>
<td>FIELD and RULE</td>
</tr>
<tr>
<td>QuestionConstruct + QuestionItem</td>
<td>FIELD and RULE</td>
</tr>
<tr>
<td>QuestionConstruct + QuestionGrid</td>
<td>-</td>
</tr>
<tr>
<td>QuestionConstruct + QuestionBlock</td>
<td>-</td>
</tr>
<tr>
<td>IfThen</td>
<td>IF THEN</td>
</tr>
<tr>
<td>Loop</td>
<td>FOR DO</td>
</tr>
<tr>
<td>RepeatWhile</td>
<td>-</td>
</tr>
<tr>
<td>RepeatUntil</td>
<td>-</td>
</tr>
</tbody>
</table>
## Blaise-DDI mapping

### As example: Question Mapping

<table>
<thead>
<tr>
<th>DDI</th>
<th>Blaise</th>
</tr>
</thead>
<tbody>
<tr>
<td>ItemName</td>
<td>FIELD {ItemName} {QuestionText} {Type} RULE: {ItemName}.ASK</td>
</tr>
<tr>
<td>QuestionText</td>
<td></td>
</tr>
<tr>
<td>CodeDomain</td>
<td>TYPE T{CodeListItemName} =</td>
</tr>
<tr>
<td>If MultipleChoiceType == SelectAllThatApply</td>
<td>SET OF</td>
</tr>
<tr>
<td>If MultipleChoiceType == SelectUpTo</td>
<td>SET [{Max}] OF</td>
</tr>
<tr>
<td>Code + Category</td>
<td>{CategoryName} {CodeValue} &quot;{CategoryLabel}&quot;</td>
</tr>
<tr>
<td>TextDomain</td>
<td>Type = String[{MaxLength}]</td>
</tr>
<tr>
<td>NumericDomain</td>
<td>Type = {Low}..{High} OR Integer</td>
</tr>
<tr>
<td>DateTimeDomain</td>
<td></td>
</tr>
<tr>
<td>If type = Date</td>
<td>Type = DateType</td>
</tr>
<tr>
<td>If type = Time</td>
<td>Type = TimeType</td>
</tr>
<tr>
<td>If type = Year</td>
<td>Type = 1..9999</td>
</tr>
<tr>
<td>If type = Month</td>
<td>Type = 1..12</td>
</tr>
<tr>
<td>All other types</td>
<td>Type = String</td>
</tr>
</tbody>
</table>
Blaise to DDI Connector Application

– Simple applications to create DDI metadata from a Blaise instrument.
– Connector tools will extract information from a prepared Blaise instrument (BMIX) using the Blaise Meta API.
– All information extracted from the Blaise BMIX is stored in a ResourcePackage.
The Blaise – Colectica – DDI link will benefit from some new concepts in DDI 3.3:
- Types on IfConditions, TypeOfComputation, Attributes on references, QuestionItems referencing RepresentedVariables
- Plus there are some things that can be considered (for 3.3 if possible):
  - Soft/Hard Edits/Checks (a relation between questions)
    • Too young to be married / have children
- Future: support for paradata & support for layout in DDI & direct support for pre-load and reference data?
Thank you for your attention!

Questions?