



New Tricks with Old Tools

Peter Sparks
25 April 2012

London, United Kingdom

Existing Tools & New Information

- Rich source of information
- Many features
- Analysis of datamodels
- Efficiency gains



Overview

- Blaise Datamodel Information
- Crosswalk
- Metric Counts
- Code complexity
- Finding unused items
- Reducing errors
- Conclusion



Blaise Datamodel Information

- BCP & source code most complete (1.1)

DatabaseManager

Databases

OpenDatabase (Database)

GetDataFileVersionInfo (VersionInfo)

GetDictionaryFileVersionInfo (VersionInfo)

OpenDataSharedDatabase (Database)

Database

C# .Net code

Blaise source

```
Count_USE_Once + USE.Count_USE_Twice + USE.
SE.Count_USE_FourTimes + USE.Count_USE_Fiv
hkPt := GoToDIWE1
E.Count_USE_Never = 16 THEN
IWD3_ = RESPONSE AND DIWD3_[1] IN [LightK
WDChkPt := GoToDIWE1
F ((DIWD1_ = RESPONSE AND (DIWD1_[1] IN [L
OR (DIWD2_ = RESPONSE AND (DIWD2_[1] IN [L
ND (DIWD3_ = NONRESPONSE OR (None IN DIWD3
IWDChkPt := GoToDIWE11a
F ((None IN DIWD1_) OR DIWD1_ = NONRESPON
ND ((None IN DIWD2_) OR DIWD2_ = NONRESPON
ND ((None IN DIWD3 ) OR DIWD3 = NONRESPON
```

```
case BLAPI3A.BlFieldType.blftAll :
case BLAPI3A.BlFieldType.blftBlock:
case BLAPI3A.BlFieldType.blftClassification:
case BLAPI3A.BlFieldType.blftExternal:
case BLAPI3A.BlFieldType.blftString:
case BLAPI3A.BlFieldType.blftUnknown:
    type = "STRING";
    break;
case BLAPI3A.BlFieldType.blftEnumeration:
case BLAPI3A.BlFieldType.blftInteger:
    type = "INTEGER";
    break;
case BLAPI3A.BlFieldType.blftFloat:
    type = "REAL";
    break;
case BLAPI3A.BlFieldType.blftDate:
    type = "DATATYPE";
    break;
```



Blaise Datamodel Information

- Delta (1.6), Manipula (1.7)

The screenshot shows the Delta software interface. The title bar reads "Delta - [C:\Projects\Conference\IBUC2012\SampleDatamodel.bmi]". The menu bar includes File, Edit, View, Compare, Tools, Windows, and Help. The toolbar contains various icons for file operations and navigation. The Tree View on the left shows a hierarchical structure of questionnaire elements:

- root. QUESTIONNAIRE SampleDatamodel
 - 1. QUESTION SampleID
 - 2. SUBQUESTIONNAIRE SubQs
 - 2.1. QUESTION DoWalking
 - 2.2. SPLIT
 - 2.2.1. SPLITIF IF DoWalking = Yes THEN
 - 3. QUESTION HowOld
 - 4. QUESTION HowOldWant
 - 5. SUBQUESTIONNAIRE AddNumbers
 - 6. QUESTION NumBiscuitsEaten

The Statement Details pane on the right shows the following code:

```
- <nodedef>  
- <qnode caption="SampleD:  
imagepath="C:/PROGRAM  
- <cat languages>
```

The screenshot shows the "Blaise 4.8 Manipula - AscToBla" dialog box. The title bar includes a question mark icon and the text "Blaise 4.8 Manipula - AscToBla". The dialog box contains the following information:

ASCII to BLAISE
Start time: 10:00:51 AM End time: 10:00:58 AM

Ident	File	Type	Read	Written	Deleted
INPU...	C:\blproj\STARRS\A-STARRS\NSSA\Data\Preload.asc	Ascii	300	-	-
OUTP...	C:\blproj\STARRS\A-STARRS\NSS... \A_STARRS_NS...	OleDB co...	-	300	-

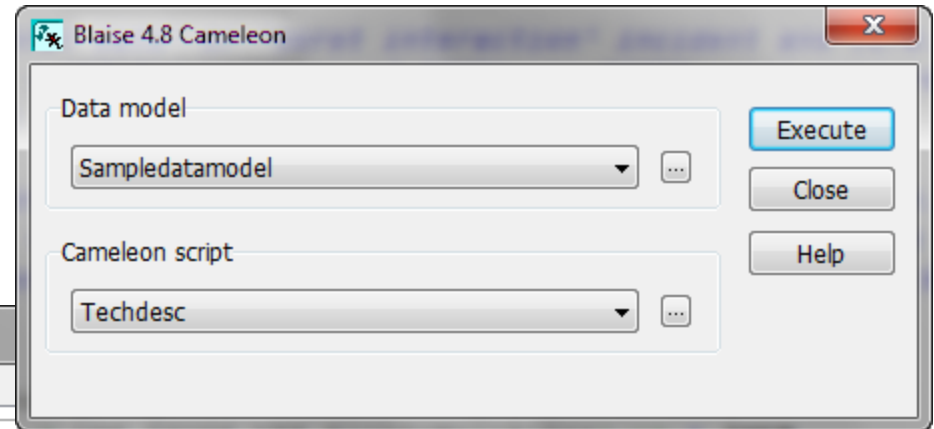
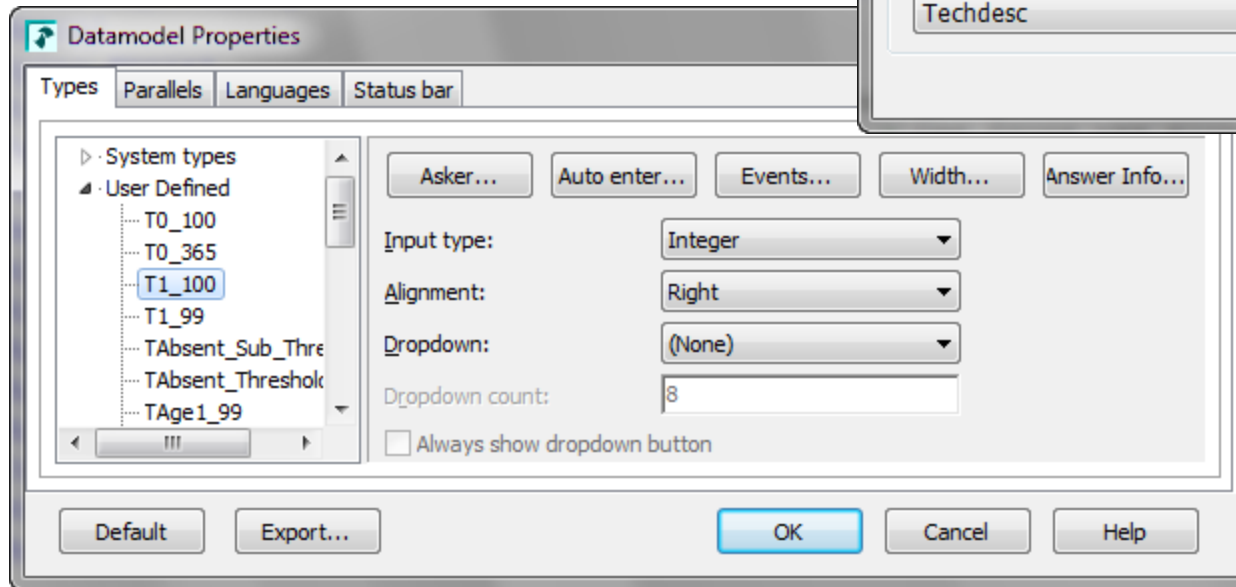
Execution successful

OK



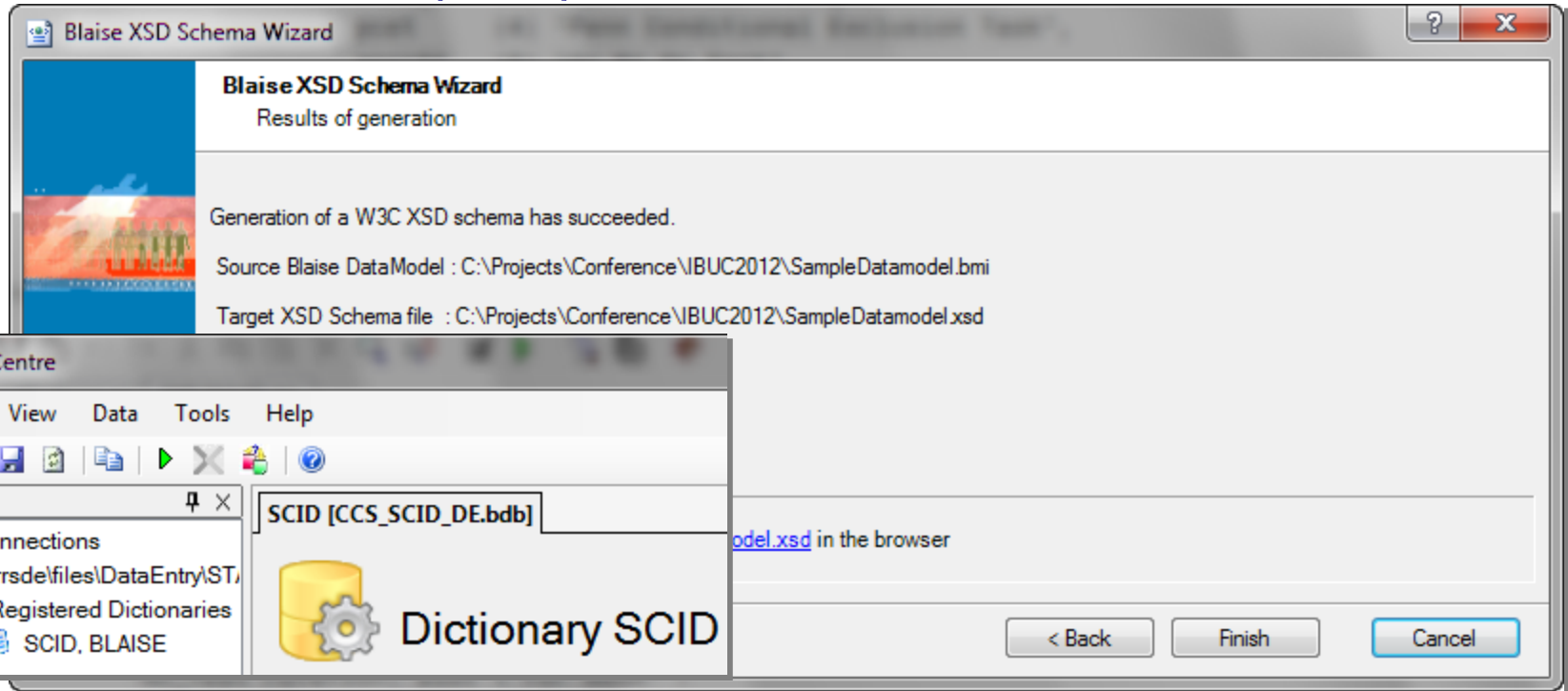
Blaise Datamodel Information

- Cameleon (1.4)
- Datamodel Properties (1.5)



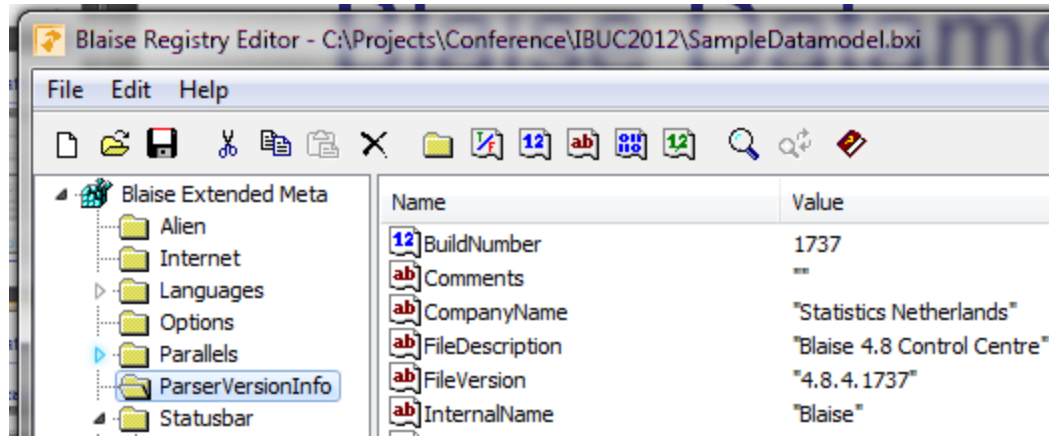
Blaise Datamodel Information

- XSD Schema (1.8)
- Data Centre (1.2)



Blaise Datamodel Information

- Blaise Registry Editor (1.3)
- Export to INI format
- Potential input to metrics/crosswalk



Metric Counts

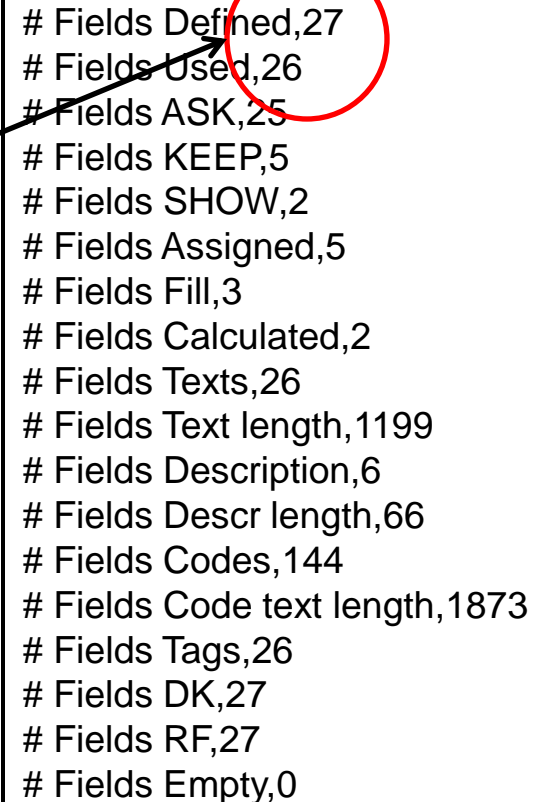
- Counts are the raw data for later calculations
- Can discover missing fields/unused items
- Emphasis on number of times used

```
# Fields Defined,27
# Fields Used,26
# Fields ASK,25
# Fields KEEP,5
# Fields SHOW,2
# Fields Assigned,5
# Fields Fill,3
# Fields Calculated,2
# Fields Texts,26
# Fields Text length,1199
# Fields Description,6
# Fields Descr length,66
# Fields Codes,144
# Fields Code text length,1873
# Fields Tags,26
# Fields DK,27
# Fields RF,27
# Fields Empty,0
```



Metric Counts

- Counts are the raw data for later calculations
- Can discover missing fields/unused items
- Emphasis on number of times used



```
# Fields Defined,27
# Fields Used,26
# Fields ASK,25
# Fields KEEP,5
# Fields SHOW,2
# Fields Assigned,5
# Fields Fill,3
# Fields Calculated,2
# Fields Texts,26
# Fields Text length,1199
# Fields Description,6
# Fields Descr length,66
# Fields Codes,144
# Fields Code text length,1873
# Fields Tags,26
# Fields DK,27
# Fields RF,27
# Fields Empty,0
```



Crosswalk

- List of different items (fields, auxfields, locals, procedure calls, texts, ...) and their use
- Helps discover missing code/unused items
- Emphasis on where used

<u>Fields</u>	
Fieldname	Line#, Kind
HowOldWant	109, Field 142, ASK
NumBiscuitsEaten	119, Field 145, ASK 151, Layout
StateWalkCode	59, Field 89, KEEP 90, Assigned 91, SHOW
UnusedQuestion	114, Field



Crosswalk

- List of different items (fields, auxfields, locals, procedure calls, texts, ...) and their use
- Helps discover missing code/unused items
- Emphasis on where used

<u>Fields</u>
Fieldname Line#, Kind
HowOldWant 109, Field 142, ASK
NumBiscuitsEaten 119, Field 145, ASK 151, Layout
StateWalkCode 59, Field 89, KEEP 90, Assigned 91, SHOW
UnusedQuestion 114, Field



Finding Unused Items

Answer "what" (metric counts) and "where" (crosswalk)

Look for ...

- Fills, Fields, Auxfields, Locals, Types
- Texts (per language)
- Assumed/default values
- Declared versus used items



Code complexity

- Metrics (from counts)
- Formulas
- Comparison of results

Average field text length = 46
Average field description length = 11
Average code text length = 13

Datamodel complexity history
10/04/12 297.5
08/04/12 250.1
07/04/12 256.3

Metric	Weight	Symbol
Number of fields asked/showed	1	Q
Fills displayed	1	F
Number of assignments	1	A
Lines of code (rules)	1	R
Number of preload variables (used/displayed)	1	P
Depth of each decision point	2	2 ^D
Number of procedure calls	1	C
Number of fields in blocks	1	B
Depth of blocks	1.5	1.5 ^K
Number of languages	1	L

Complexity score = $(Q + F + A + R + P + (2^D) \dots + C - B + (1.5^K) \dots) * L / Q$



Reducing errors

- Inadvertent checks and signals
- Locate by check/signal without custom error text (metrics/crosswalk)

Inadvertent check/signal

```
A1
IF A1 = Yes THEN
    xA2Fill1 = 'anyone'
ELSE
    xA2Fill1 := 'someone'
ENDIF
A2
```

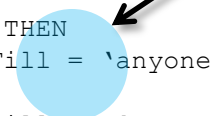


Reducing errors

- Inadvertent checks and signals
- Locate by check/signal without custom error text (metrics/crosswalk)

Inadvertent check/signal

```
A1
IF A1 = Yes THEN
    xA2Fill1 = 'anyone'
ELSE
    xA2Fill1 := 'someone'
ENDIF
A2
```



Reducing errors

- Default values (i.e., code frames, question text, initial values)

Default values

```
LANGUAGES = ENG "English",
              SMG

TYPE
  TBiscuitCount = (None           ENG "No biscuits",
                  B1_2            "1 - 2",
                  B3_4            "3 - 4",
                  B5_9            "5 - 9",
                  B10More (10)    ENG "10 or more biscuits",
                               SMG "10 ore moure biscuites")

AUXFIELDS
  xHeShe : STRING

FIELDS
  NumBiscuitsEaten
    "FR: R thinks that ^xHeShe will live to be ^DeathAge.

    @/@/How many biscuits have you eaten during the break?" :

    TBiscuitCount

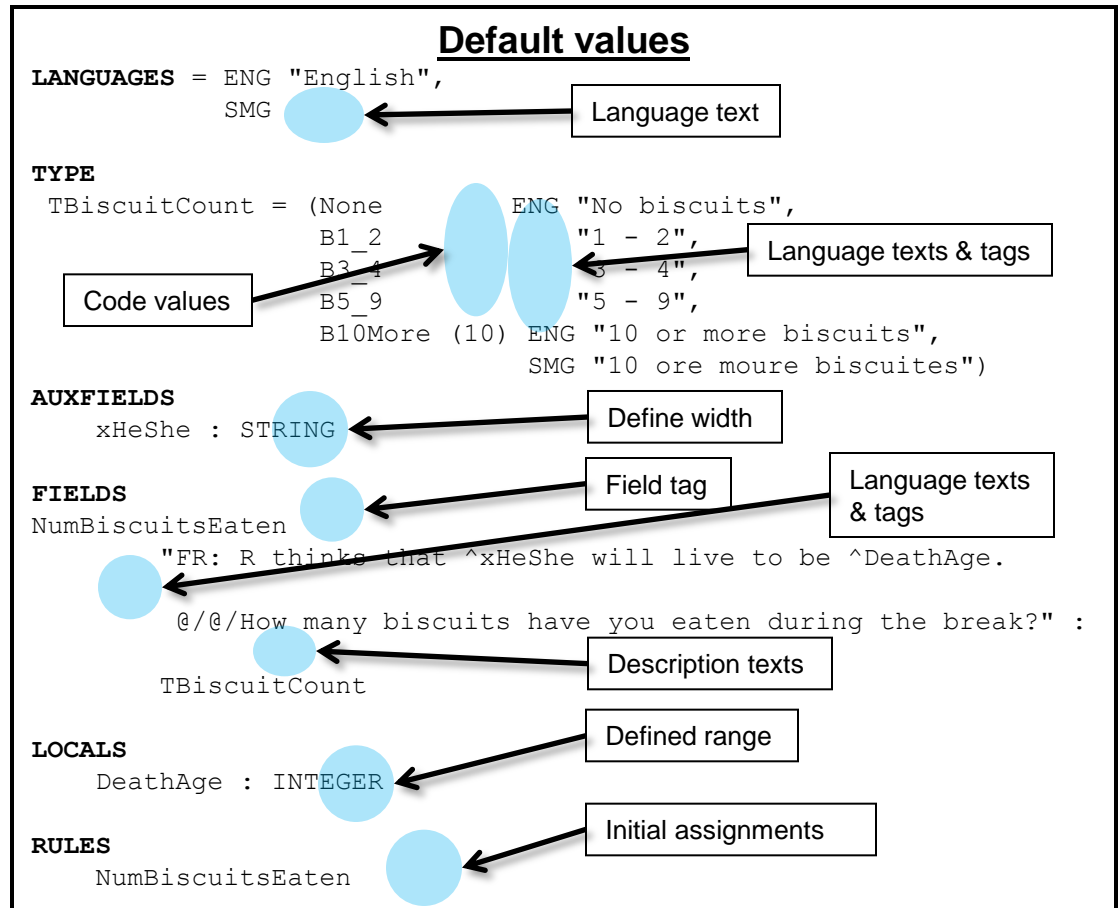
LOCALS
  DeathAge : INTEGER

RULES
  NumBiscuitsEaten
```



Reducing errors

- Default values (i.e., code frames, question text, initial values)



Reducing errors

- Fill variable lengths from multiple parts

Fills from parts

FIELDS

```
RName : STRING[20]
```

AUXFIELDS

```
xHeSheOr : STRING
```

RULES

```
...
```

```
xHeSheOr.KEEP
```

```
IF RSex = Male THEN
```

```
    xHeSheOr := 'his'
```

```
ELSEIF Rsex = Female THEN
```

```
    xHeSheOr := 'her'
```

```
ELSE
```

```
    xHeSheOr := RName + 's'
```

```
ENDIF
```



Reducing errors

- Fill variable lengths from multiple parts

Fills from parts

```
FIELDS
  RName : STRING[20]

AUXFIELDS
  xHeSheOr : STRING[?]

RULES
  ...
  xHeSheOr.KEEP
  IF RSex = Male THEN
    xHeSheOr := 'his'
  ELSEIF Rsex = Female THEN
    xHeSheOr := 'her'
  ELSE
    xHeSheOr := RName + 's'
  ENDIF
```

What should be this width?



Future work

- Continued exploration of metrics
- Better/revised code complexity formulas
- Source code revisions based on results
- Datamodel analysis tool
 - Metric counts
 - Crosswalk
 - Complexity score
 - Incomplete statements
 - Potential programming errors
 - Unused items



Contact Information

Peter Sparks, zebulon@umich.edu

