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)

C# .Net code

```csharp
case BLAPI3A.BlFieldType.blftAll :
    break;

case BLAPI3A.BlFieldType.blftBlock:
    break;

case BLAPI3A.BlFieldType.blftClassification:
    break;

case BLAPI3A.BlFieldType.blftExternal:
    break;

case BLAPI3A.BlFieldType.blftString:
    break;

case BLAPI3A.BlFieldType.blftUnknown:
    type = "STRING";
    break;

case BLAPI3A.BlFieldType.blftEnumeration:
    break;

case BLAPI3A.BlFieldType.blftInteger:
    type = "INTEGER";
    break;

case BLAPI3A.BlFieldType.blftFloat:
    type = "REAL";
    break;

case BLAPI3A.BlFieldType.blftDate:
    type = "DATETYPE";
    break;
```

Blaise source

```
if (Count_USE.Once + USE.Count_USE_Twice + USE.Count_USE_FourTimes + USE.CountUSE_FiveTimes > 0) {
    Blaise.chkPt := GoToDIWE1;
    E.Count_USE.Never = 16 THEN
    IWD3_ = RESPONSE AND DIWD3_[1] IN [LightK

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_ = NONRESPONSE

ND ((None IN DIWD2_) OR DIWD2_ = NONRESPONSE

ND ((None IN DIWD3_) OR DIWD3_ = NONRESPONSE
```
Blaise Datamodel Information

- Delta (1.6), Manipula (1.7)
Blaise Datamodel Information

- Cameleon (1.4)
- Datamodel Properties (1.5)
Blaise Datamodelf 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

25 April 2012
New Tricks with Old Tools
Metric Counts

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

```plaintext
# 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

<table>
<thead>
<tr>
<th>Metric</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td># Fields Defined</td>
<td>27</td>
</tr>
<tr>
<td># Fields Used</td>
<td>26</td>
</tr>
<tr>
<td># Fields ASK</td>
<td>25</td>
</tr>
<tr>
<td># Fields KEEP</td>
<td>5</td>
</tr>
<tr>
<td># Fields SHOW</td>
<td>2</td>
</tr>
<tr>
<td># Fields Assigned</td>
<td>2</td>
</tr>
<tr>
<td># Fields Fill</td>
<td>3</td>
</tr>
<tr>
<td># Fields Calculated</td>
<td>2</td>
</tr>
<tr>
<td># Fields Texts</td>
<td>26</td>
</tr>
<tr>
<td># Fields Text length</td>
<td>1199</td>
</tr>
<tr>
<td># Fields Description</td>
<td>6</td>
</tr>
<tr>
<td># Fields Descr length</td>
<td>66</td>
</tr>
<tr>
<td># Fields Codes</td>
<td>144</td>
</tr>
<tr>
<td># Fields Code text length</td>
<td>1873</td>
</tr>
<tr>
<td># Fields Tags</td>
<td>26</td>
</tr>
<tr>
<td># Fields DK</td>
<td>27</td>
</tr>
<tr>
<td># Fields RF</td>
<td>27</td>
</tr>
<tr>
<td># Fields Empty</td>
<td>0</td>
</tr>
</tbody>
</table>
Crosswalk

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

<table>
<thead>
<tr>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fieldname</td>
</tr>
<tr>
<td>Line#, Kind</td>
</tr>
<tr>
<td>HowOldWant</td>
</tr>
<tr>
<td>109, Field</td>
</tr>
<tr>
<td>142, ASK</td>
</tr>
<tr>
<td>NumBiscuitsEaten</td>
</tr>
<tr>
<td>119, Field</td>
</tr>
<tr>
<td>145, ASK</td>
</tr>
<tr>
<td>151, Layout</td>
</tr>
<tr>
<td>StateWalkCode</td>
</tr>
<tr>
<td>59, Field</td>
</tr>
<tr>
<td>89, KEEP</td>
</tr>
<tr>
<td>90, Assigned</td>
</tr>
<tr>
<td>91, SHOW</td>
</tr>
<tr>
<td>UnusedQuestion</td>
</tr>
<tr>
<td>114, Field</td>
</tr>
</tbody>
</table>
Crosswalk

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

<table>
<thead>
<tr>
<th>Fieldname</th>
<th>Line#, Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>HowOldWant</td>
<td>109, Field</td>
</tr>
<tr>
<td></td>
<td>142, ASK</td>
</tr>
<tr>
<td>NumBiscuitsEaten</td>
<td>119, Field</td>
</tr>
<tr>
<td></td>
<td>145, ASK</td>
</tr>
<tr>
<td></td>
<td>151, Layout</td>
</tr>
<tr>
<td>StateWalkCode</td>
<td>59, Field</td>
</tr>
<tr>
<td></td>
<td>89, KEEP</td>
</tr>
<tr>
<td></td>
<td>90, Assigned</td>
</tr>
<tr>
<td></td>
<td>91, SHOW</td>
</tr>
<tr>
<td>UnusedQuestion</td>
<td>114, Field</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Date</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/04/12</td>
<td>297.5</td>
</tr>
<tr>
<td>08/04/12</td>
<td>250.1</td>
</tr>
<tr>
<td>07/04/12</td>
<td>256.3</td>
</tr>
</tbody>
</table>

Complexity score = \((Q + F + A + R + P + (2^D) \ldots + C - B + (1.5^K)\ldots) \cdot L / Q\)
Reducing errors

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

```plaintext
Inadvertent check/signal
A1
IF A1 = Yes THEN
    xA2Fill = 'anyone'
ELSE
    xA2Fill := 'someone'
ENDIF
A2
```
Reducing errors

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