Blaise Source Code Editing System

Presenter: Danilo Gutierrez
Co-author: Sheila Deskins
Health and Retirement Study (HRS)

The 11th International Blaise Conference
Annapolis, Maryland
September 2007
Presentation Overview

How Big is Big?
What Does a Source Editor Do?
The System and Updating a New Language
Current Use & Future Plans
Questions
How Big is Big?
<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program lines</td>
<td>175,624</td>
</tr>
<tr>
<td>Include files</td>
<td>61</td>
</tr>
<tr>
<td>Procedures</td>
<td>518</td>
</tr>
<tr>
<td>Blocks</td>
<td>344</td>
</tr>
<tr>
<td>Tables</td>
<td>10</td>
</tr>
</tbody>
</table>

HRS CAI Size

- Datamodel Source Code (.bla, .inc)

Survey Research Center • Institute for Social Research • University of Michigan
HRS CAI Size

• Fields
  – 5,818 Fields
  – 1,773 Auxfields
  – 1,691 Locals
  – 5,754 Parameters
HRS CAI Size

• Type Definitions
  – 8,962 USER-DEFINED
  – 1,390 ENUMERATED
  – 2,366 STRING
  – 481 RANGE
  – 366 OPEN
  – 309 INTEGER
  – 126 ARRAY
  – 141 SET
  – 11 REAL
What does a “Source Editor” do?
System Core

- Parses source code files (.bla and .inc)
- Merges in update information
- Writes updated source code files
Statement:
Q1 (One) "Are you ready to answer questions?" : (y,n)

 Parses into tokens of:
Q1
(One)
"Are you ready to answer questions?"
:
(y
, n
)
Definition of Token

A **token** is part of a program statement consisting of characters identified as meaningful syntax.
Merging

- There are two inputs in merging
  - The user update request information
  - The parsed tokens from original source code
User Request

User update request information:

fieldname: Q1

token type: descriptor

language: default

edit instruction: add (or delete)

update: “This is the new label”
Source code in tokens:

Q1
(One)
“Are you ready to answer questions?”
“This is the new label”

Problem: the descriptor looks like question text!
How to fix the problem?

- Blaise Data Object (BDO) contains all possible parts of a Blaise syntax statement.

Blaise Syntax for Fields

Q [ Q1, [ ... ] ] [ ( Tag ) ] [ [ Lid ] "Text" ] [ ... ]
[ / [ Lid ] "Description" ] [ ... ] : T
Using the BDO

Syntax
Q
[ Q1, [ ... ] ]
[ ( Tag ) ]
[ [ Lid ] "Text" ]
[ ... ]
[/ [ Lid ]
  "Description" ]
[ ... ] : T

BDO with update
Q1
<blank>
(One)
[[ Lid ] "Are you ready to answer questions?" ]
[ ... ]
[/ [ Lid ]
  "This is the new label" ]
[ ... ] : T
Q1 (One) “Are you ready to answer questions?”

/ “This is the new label” : (y , n)
Writing

• Writing is simpler when the database is already organized by the parsing and merging processes
• Need to write out whitespace, comments, file names, etc.
• Write Spanish language diacriticals
• Write the same number of files as were parsed
System Considerations
## System Considerations

<table>
<thead>
<tr>
<th></th>
<th>Editing requires</th>
<th>.BMI (BCP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parse Blaise source files (.bla, .inc) into tokens</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Keep Comments</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Keep include file structure</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Keep whitespace</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
Why would we want a Source Editor System?

• HRS is longitudinal study.

• It’s a ‘big’ application.

• Most of the large scale (bulk) changes have to do with fields. ‘Small’ changes usually involve a few hundred fields (10% = 581 changes).

• There often are several “small” changes that take place during the CAI preparation for a field period.
What We Learned

- For the 2004 Descriptor update task
  - The few descriptors mentioned turned out to be a 2,700 descriptors change request
  - The merge key that was provided with the descriptor update request information was the DEP field name
Translator Functions

- Convert DEP field names to defined block and field name
- Report duplicate requests for same defined block and field name
Need to ‘translate’ DEP fieldname paths to defined block name.

<table>
<thead>
<tr>
<th>Block Name (Def)</th>
<th>#</th>
<th>ind</th>
<th>DEP Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>41</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>BB_Born</td>
<td>42</td>
<td>1</td>
<td>SecB.Born</td>
</tr>
<tr>
<td>BB_ShowStateList</td>
<td>43</td>
<td>1</td>
<td>SecB.Born.B076_B003_</td>
</tr>
<tr>
<td>BB_ShowStateList</td>
<td>43</td>
<td>2</td>
<td>SecB.LivedArea.B078_B047_</td>
</tr>
</tbody>
</table>
Several DEP fields update request with one define field

<table>
<thead>
<tr>
<th>Block (Defined)</th>
<th>Name (Defined)</th>
<th>Existing Descriptor</th>
<th>User Descriptor Request</th>
<th>DEP Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB_Marry</td>
<td>B066_</td>
<td>MARR YEAR BEG</td>
<td>FIRST MARRIAGE YEAR BEGAN</td>
<td>SecB.Marry[1].B066_</td>
</tr>
<tr>
<td>BB_Marry</td>
<td>B066_</td>
<td>MARR YEAR BEG</td>
<td>SECOND MARRIAGE YEAR BEGAN</td>
<td>SecB.Marry[2].B066_</td>
</tr>
<tr>
<td>BB_Marry</td>
<td>B066_</td>
<td>MARR YEAR BEG</td>
<td>THIRD MARRIAGE YEAR BEGAN</td>
<td>SecB.Marry[3].B066_</td>
</tr>
<tr>
<td>BB_Marry</td>
<td>B066_</td>
<td>MARR YEAR BEG</td>
<td>MARRIAGE YEAR BEGAN -4</td>
<td>SecB.Marry[4].B066_</td>
</tr>
</tbody>
</table>
The System and
Updating a new language
New Language Update

• The early system that handled updating descriptors needed to be expanded to handle the ‘update’ addition of a new language.
System Conceptualization

- Parser Application
- DEP Field Name Translator
- BDO Creation
- Merger Application
- Writer Application
- User Interface
System Design Considerations

- Encapsulated routines and procedures for each function
- Reusable code versus ad hoc routines
- Tokens described in more meaningful terms
- Language order option
- Parsing whitespace
<table>
<thead>
<tr>
<th>Before</th>
<th>Quick</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE English</td>
<td>CORE English</td>
<td>CORE English</td>
</tr>
<tr>
<td>CORE Spanish</td>
<td>CORE Spanish</td>
<td>CORE Spanish</td>
</tr>
<tr>
<td>PROXY English</td>
<td>PROXY English</td>
<td>PROXY English</td>
</tr>
<tr>
<td>EXIT English</td>
<td>EXIT English</td>
<td>PROXY Spanish</td>
</tr>
<tr>
<td>EXIT Spanish</td>
<td>EXIT Spanish</td>
<td>EXIT English</td>
</tr>
<tr>
<td>MEDIA</td>
<td>MEDIA</td>
<td>EXIT Spanish</td>
</tr>
<tr>
<td></td>
<td>PROXY Spanish</td>
<td>MEDIA</td>
</tr>
</tbody>
</table>

Survey Research Center • Institute for Social Research • University of Michigan
BDO and Language

- Looking at all statements that can contain language.
  - Field question text
  - enumerated data type code descriptions, and
  - descriptors

- Coming up with a new way to describe all involved tokens, and merge keys for the token types.
Other considerations

• Update request for enumerations (defined at the field)
  – Merge keys needed
  – BDO for an enumeration type statement

Fields
TP50 "What type of fish do you have?": (n "none", f "fresh water", s "salt water" )
Other considerations

- Update request for enumerations (defined as a type)
  - Merge keys needed
  - Look and find the type

TP60 "What type of mammals aside from dogs or cats do you have?“ : TMammals

<table>
<thead>
<tr>
<th>Blkname</th>
<th>Field Name</th>
<th>Type</th>
<th>Type No</th>
<th>Blk path</th>
<th>Blk No</th>
<th>Bk NameEnd</th>
<th>Type NameEnd</th>
<th>TokenType</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE_illustration</td>
<td>tmammals</td>
<td>25</td>
<td>1..</td>
<td>1</td>
<td></td>
<td>SE_illustration</td>
<td>tmammals</td>
<td>TName</td>
</tr>
<tr>
<td>B2_Other</td>
<td>TP50</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2_Other</td>
<td>TP60</td>
<td>41</td>
<td>8.1..</td>
<td>1</td>
<td></td>
<td>SE_illustration</td>
<td>tmammals</td>
<td>FTName</td>
</tr>
</tbody>
</table>
Other considerations

- HRS doesn’t use LID – need to put in “” to keep correct relative positioning for languages
Before

TPosition =

(STANDING (1) "Standing" "Parado",
SITTING (2) "Sitting" "Sentado",
LYINGDOWN (3) "Lying down" "Acostado")
After

TPosition =

(Stan **D**ing (1) "Standing" "Parado" {SE Added text} "Standing" {LS} "Parado" ,
S **i**ting (2) "Sitting" "Sentado" {SE Added text} "Sitting" {LS} "Sentado" ,
L **y**ing **d**own (3) "Lying down" "Acostado" {SE Added text} "Lying down" {LS} "Acostado" )
Other considerations

- Report when update request not found in the data
- Report duplicate update request for the same defined field with option to process/not process those records
Current Use & Future Plans

• The time difference between machine versus manual work.

• Is it worth the effort?
Current System Performance

- 8 hours - Parse 175,600 lines of code
- 2 hours - Type Cross Reference
- 6 hours - Create BDO Table
- 6 hours - Merge 6,500 updates
- 3 minutes - Write files

Total Time: 22 hours
Other Uses

• Analysis of Datamodel – being able to look at data within context of tokens

• An alternative way of making updates, i.e. one doesn’t necessarily need to use the merger. The system allows for direct edits to the tables in the database.
Tasks for the Current System

• Strip out obsolete or dated comments from prior years.
• Update tags. Modify tags to be more descriptive.
• Update descriptors. Modify labels for data out.
• Update data types. Modify field size, field ranges, etc.
• Update language text. Text provided by another system such as a product from the HRS translation group.
System Benefits

• Time saving, resulting in faster turn-around of tasks.
• Hundreds of changes can be made at one time.
• More accurate placement of updates and therefore better quality.
• May reduce repetitive-use injury.
System Benefits (2)

- Robust enough to handle applications as large as HRS.
- Generic enough to handle other non-HRS Blaise CAI applications.
- The application can add or re-order languages.
- The application has features to help handle scale issues.
• Acknowledgments
  – HRS Instrument Development Team
Blaise Source Code Editing System

Questions?

Contact Information:
Danilo Gutierrez
danilog@umich.edu

Sheila Deskins
sld@umich.edu

Survey Research Center • Institute for Social Research • University of Michigan
FAQ

- Q: In the paper you didn’t implement a **translator**?
- A: For the language update, *we gave* the user a table for fields and a table for enumeration *with keys needed* so we wouldn’t have to develop the translator, or worry about multiple update request for the same defined field, for this system development round.
- A: We’ll look at how we want to develop this implementation in the future.
Question

- Q: What technology did you use?

- A:
  - Parser in VB6 (Access or SQLServer)
  - Merger, Writer in VB.NET and ADO.NET
  - Source Editor Database in SQLServer
  - Proof of concept in C# .NET

Miscellaneous:
- Ad hoc routines in SAS
- BCP