Verifying the design of an outsourced COBOL system with IntensiVE

Andy Kellens
Carlos Noguera
Theo D’Hondt

Luc Jorissen
Bart Van Passel

Software Languages Lab
Vrije Universiteit Brussel

inno.com
Context

- Academic-industrial collaboration
- Large system for Belgian bank
- Started development in 2005; currently in production
- COBOL

Modern design:
- Components
- Services

Old language

Verifying the design of an outsourced COBOL system with IntensiVE

Friday 17 September 2010
Problem

- Large investment
- Outsourcing

Verifying the design of an outsourced COBOL system with IntensiVE

Friday 17 September 2010
Problem

- Large investment
- Outsourcing

Prevent drift

Verifying the design of an outsourced COBOL system with IntensiVE

Friday 17 September 2010
Problem

- Large investment
- Outsourcing

Prevent drift

Verifying the design of an outsourced COBOL system with IntensiVE

Friday 17 September 2010
Mapping OO concepts to COBOL

Implementation guidelines
- Patterns
- Conventions
- Idioms
- Naming schemes
- Layering

Sequence diagrams
- Programs
  - Sections
  - Paragraphs
  - Statements

Friday 17 September 2010
Verifying the design of an outsourced COBOL system with IntensiVE
IntensiVE in a nutshell

Software Implementation

Specify

Verify

Architectural Design Documents

LMP

Model

Tool suite

Verifying the design of an outsourced COBOL system with IntensiVE
IntensiVE for COBOL

<table>
<thead>
<tr>
<th>Island-based parsing</th>
<th>Static analyses</th>
<th>Predicate library</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Island-based parsing image" /></td>
<td><img src="image2.png" alt="Static analyses image" /></td>
<td><img src="image3.png" alt="Predicate library image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicate library examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>?program isProgramWithIdentifier: ?id</td>
</tr>
<tr>
<td>?program callsProgram: ?callee</td>
</tr>
<tr>
<td>?program writesToTable: ?table</td>
</tr>
<tr>
<td>?section sectionInProgram: ?program</td>
</tr>
<tr>
<td>?section sectionPerforms: ?section</td>
</tr>
<tr>
<td>?section sectionWithName: ?name</td>
</tr>
<tr>
<td>?program usesCopybook: ?copybook</td>
</tr>
</tbody>
</table>
Example: layering naming convention

“A section can *only* perform sections that have section name with the *same* or *later* beginning letter”

Sections with callees

```plaintext
if ?caller sectionPerformsSection: ?callee
```
Example: layering naming convention

“A section can only perform sections that have the same or later beginning letter”

```coql
if ?caller sectionPerformsSection: ?callee
```

Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)
Example: layering naming convention

“A section can only perform sections that have section name with the same or later beginning letter”

Sections with callees

\[
\text{if } \text{?caller sectionPerformsSection: } \text{?callee}
\]
Example: layering naming convention

“A section can only perform sections that have section name with the same or later beginning letter”

Sections with callees

if ?caller sectionPerformsSection: ?callee

∀ ?invocation ∈ Sections with callees:
  ?invocation.caller isSectionWithName: ?callerName,
  ?invocation.callee isSectionWithName: ?calleeName,
  [?callerName <= ?calleeName]

Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)
Example: layering naming convention

“A section can only perform sections that have section name with the same or later beginning letter”

Cognac: a framework for documenting and verifying the design of Cobol systems (CSMR 2009)
Verifying the design of an outsourced COBOL system with IntensiVE

Verifying sequence diagrams (1)

Custom extension

Map to code

Verify mappable diagrams

Rational Rose

Scripting

Design

COBOL

IntensiVE

Call graph
Verifying the design of an outsourced COBOL system with IntensiVE

Verifying sequence diagrams (2)

SystemBorder  EntryPoint  Program 1  Program 2

- perform use case
- invoke sub-operation 1
- invoke sub-operation 2
- invoke sub-operation 3

Dedicated tool

Tool identifies best match

Section RETR-DATA-Feed

P1  P2  P3  P4  P5  P6  P7
Experimental setup

<table>
<thead>
<tr>
<th>Version</th>
<th>LOC</th>
<th>#diagrams</th>
<th>parsing (sec)</th>
<th>analysis (sec)</th>
<th>verification (sec)</th>
<th>total (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1</td>
<td>548 560</td>
<td>895</td>
<td>105</td>
<td>150</td>
<td>192</td>
<td>447</td>
</tr>
<tr>
<td>Version 2</td>
<td>665 220</td>
<td>917</td>
<td>134</td>
<td>228</td>
<td>204</td>
<td>566</td>
</tr>
<tr>
<td>Version 3</td>
<td>1 053 381</td>
<td>1366</td>
<td>413</td>
<td>479</td>
<td>288</td>
<td>1180</td>
</tr>
</tbody>
</table>

- 3 versions
- Increase in size
- Scalable tool
Results

<table>
<thead>
<tr>
<th>Version</th>
<th>#diagrams</th>
<th>mappable</th>
<th>unmappable</th>
<th>% mappable</th>
<th>consistent</th>
<th>inconsistent</th>
<th>% inconsistencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1</td>
<td>895</td>
<td>408</td>
<td>487</td>
<td>45.59%</td>
<td>326</td>
<td>82</td>
<td>25.15%</td>
</tr>
<tr>
<td>Version 2</td>
<td>917</td>
<td>476</td>
<td>441</td>
<td>51.90%</td>
<td>386</td>
<td>90</td>
<td>23.33%</td>
</tr>
<tr>
<td>Version 3</td>
<td>1366</td>
<td>763</td>
<td>603</td>
<td>55.85%</td>
<td>637</td>
<td>126</td>
<td>19.78%</td>
</tr>
</tbody>
</table>

- **Half mappable:**
  - Mismatch documentation/implementation
  - Informal parts of documentation

- **Over time, less inconsistencies**

- **Inconsistencies:**
  - Diagrams without implementation
  - Order of calls
  - Calls missing
Lessons learned

**Academic**

- Required pragmatic solutions:
  - Island-based parsing
  - Customized tool
  - Reporting vs. integrated tool

**Industrial**

- Quality control when outsourcing development
  - Early-on detection of conformance problems
  - Outsourcing firms: demonstrate quality guarantees
Conclusions

- Outsourced COBOL system
- Using IntensiVE tool
  - dedicated support for COBOL
- Verify mapping of design onto COBOL code
- Verify high-level design documentation
Verifying the design of an outsourced COBOL system with IntensiVE

http://soft.vub.ac.be
http://www.intensional.be
http://www.inno.com