Business Process Recovery for Software Maintenance – An Empirical Approach

By Zhengong Cai
College of Computer Science,
Zhejiang University
Sep 22th, 2009
Outline

- Introduction
- The Recovery Approach
- Application in ETS
- Conclusion & Future work
- Q&A
Introduction

- The business process is composed of a sequence of business tasks. Understanding business processes is an important step for the comprehension and maintenance of the source code.

- The main thoughts to construct business processes:
  - Requirement analysis – *not suitable*
  - Recover from legacy systems

In fact, it is a great challenge to recover implemented business processes since the system documents are no more available and original developers left.
Related Works

- **Business logic extraction and concept location techniques**, including static analysis, dynamic tracing and the combined ones. They achieve good results under specific situations, but only discrete business logics could be re-acquired without proper organization.

- **Model-driven business process recovery framework proposed by Zou et al.** Static analysis is limited by the polymorphism and encapsulation which is popular in OO programming.

- **Some thoughts (Salah et al) through combining unified process and dynamic tracing to comprehend legacy systems.**

However, financial systems like equity trading have some peculiarities as following and these approaches face challenges when applying to this domain

- Complex business processes. The system workflow implementing these business processes is very complex accordingly.

- Most business processes of the systems are driven by the external actors. Thus it's not easy to trace the processes without knowing of the triggers.
The Identification Approach

- Interviewing for Use Cases
- Dynamic Tracing
- Static Analysis
- Tracing Log Analysis
- Business Process Representation
The Approach—Step 1

- Step 1: Requirement Reacquisition
  - One of the effective way is to interview the users, for lack of available documents
  - Not only for initial comprehension, but also for guiding the source code tracing
  - The interview results are usually not intact, integration them into initial business process is one of the useful ways to find missing use cases.
    - Usually only the boundary services are collected;
    - Parts of the services could be obtained
The Approach—Step 2

- **Step 2: Program analysis**
  - Bottom-up reverse engineering
  - Top-down forward engineering
  - Hybrid one – static analysis and dynamic tracing

  The right is an example.

**Need to note:**

Not all the business processes are triggered by external actors even if this is a classical input-triggered system. Some concept locating techniques could be used, like GREP.
The Approach—Step 3

- Step 3: Tracing log analysis
  - Tracing results are not real business process
    - Tracing results contain various elements for relevant functions, including business-independent elements as well as the business-dependent ones.
    - For “pure” business processes, filter the business independent elements like most of the utilities, I/O, Exception, and the language-specific implementations, etc.
  - Decouple the business process candidates
    - A metric is given to measure the similarity as follows.
      - If the similarity of two processes exceeds a given threshold, they could be merged.
      - If a sub-process is shared by more than one, it may be separated and formed as a new one.
  - After log analysis, more business knowledge is acquired. As the feedback, the tracing could be repeated and the results are refined.

\[ Sim(BP_i, BP_j) = \frac{BP_i \cap BP_j}{BP_i \cup BP_j} \]
Step 4: Business process representation

- The recovered business processes should be well represented to facilitate system maintenance.
- Layered representation
  - Top layer – high-level business tasks
  - Bottom layer – atomic business tasks
  - Middle layer – sub-tasks connecting business processes and atomic tasks
Project Introduction

- In production for more than 15 years
- Over 700 KLOC
- Outdated documents for frequently changed
- Endless bug reports after modification since lack of sufficient comprehension
Application in ETS – recovered use cases
Take the *Order Management* as an example, it contains three scenarios, *Order Entry, Order Cancel* and *Order Update*

Over 300 classes are traced:

- 41 for message parsing
- 34 classes/files for message syntax analysis
- 103 for network communication and data management
- Over 80 for others like utilities, exception, etc.
Application in ETS – process representation

For clear, only the *Order Entry* is showed here.
By reconstructing the business processes for this project, the maintenance efficiency was improved, as in Table 1.

- The efforts for system comprehension was greatly improved for new-joined maintainers by using the recovered hierarchical business processes.
- The maintenance process was accelerated since the concept location could be easier by the requirement-implementation mapping. For example, before re-organizing the business processes, it needed one maintainer for a specific system component. So when making a complex change, such as changing a field of the order, nearly all the components would be affected and all the maintainers would be involved.
- Once any fine logic was ignored, the fault would occur. The bug fix is more successful with recovered business processes.

<table>
<thead>
<tr>
<th></th>
<th>Efforts for comprehension</th>
<th>Involved maintainers</th>
<th>Fault rate of bug fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>over 3 months</td>
<td>5</td>
<td>15%</td>
</tr>
<tr>
<td>After</td>
<td>1 month</td>
<td>2</td>
<td>6%</td>
</tr>
</tbody>
</table>
Conclusion & Future work

➢ Contributions
  • The approach is concluded from the ETS application, but could be applied to other systems with peculiarities like complex business process, input-driven.
  • The hierarchical representation provides multiple views. From top to bottom, it changes from business view to implementation view.
  • The concept location is accelerated. When a new change request or bug comes, the concept could be easily located by the hierarchical business process representation.

➢ Issues & Future work
  • It still needs further refinement and validation by analyzing more cases.
  • Some metrics should be defined to guide the recovery process.
Thank you

Q&A