A Business Process Explorer: Recovering and Visualizing E-Commerce Business Processes

Jin Guo[1], King Chun Foo[2], Liliane Barbour[3], and Ying Zou[3]

School of Computing[1]
Queen's University
Kingston, Ontario, Canada
guojin@cs.queensu.ca

Department of Electrical and Computer Engineering[2]
Queen's University
Kingston, Ontario, Canada
3kcdf@qlink.queensu.ca

Department of Electrical and Computer Engineering[3]
Queen's University
Kingston, Ontario, Canada
{4lb3, ying.zou}@queensu.ca

ABSTRACT

A business process is composed of a set of interrelated tasks which are joined together by control flow elements. E-commerce systems implement business processes to automate the daily operations of an organization. Organizations must continuously modify their e-commerce systems to accommodate changes to business processes. However, modifying e-commerce systems is a time consuming and error prone task. To correctly perform this task, developers require an in-depth understanding of multi-tiered e-commerce systems and the business processes that they implement. In this paper, we present a business process explorer tool which automatically recovers business processes from three tier e-commerce systems. Developers can explore the recovered business processes and browse the corresponding source code. We integrate our tool with IBM WebSphere Business Modeler (WBM), a leading commercial tool for business process management and modeling. Business analysts could then visualize and analyze the recovered processes using WBM. The business process explorer eases the co-evolution of business processes and their e-commerce system implementation.

Categories and Subject Descriptors

D.2.7 [**Distribution, Maintenance, and Enhancement**]: Documentation, Reverse Engineering

General Terms

Documentation, Design.

Keywords

Business process, process recovery, e-commerce, program comprehension

1. INTRODUCTION

A business process is composed of a set of interrelated tasks which are joined together by control flow elements. Control flows specify the order of execution using various elements such as sequential, parallel, or alternative order. For example, the business process for ordering a book may consist of a number of tasks such as "check the availability of the book", "restock the inventory if needed", and "validate the buyer's credit card". E-commerce systems implement business processes and provide users with convenient electronic services, such as banking, investment and purchasing. Organizations use e-commerce systems to automate their daily business operations.

Business processes are constantly optimized to achieve rapidly changing business goals, such as reducing operation cost and providing faster services. A change, such as adding a new task to an existing business process, or merging two tasks into one can be labor-intensive and requires an in-depth understanding of multi-tiered e-commerce systems and the business processes that they implement. All too often, there are no explicit links between the tasks specified in process definitions, and the implementation of these tasks in the e-commerce systems. The lack of links between specification and implementation leads to difficulties in identifying the appropriate code segments which must be modified to further expand the functionalities of a system in response to changing business goals. To facilitate the evolution of e-commerce systems, we developed a business process explorer tool that recovers business processes from their e-commerce implementation and establishes up-to-date documentation and links between business process definitions and implementation.

A typical e-commerce system consists of three-tiers: User Interface (UI), business logics, and databases. The tiers react to users' requests and generate results according to user's selections using UI, business rules and status data stored in databases. Due to the dynamic nature of e-commerce systems, the execution of certain tasks can be only determined at run-time. Relying only on static analysis by tracing the source code would miss such tasks and would result in the recovery and documentation of incomplete processes. Therefore, we use static and dynamic tracing techniques for recovering business processes.

To allow a business analyst to visualize and refine the recovered business processes, our process explorer tool integrates into the IBM WebSphere Business Modeler (WBM) [5], a leading commercial business process modeling tool. Recovered processes can be visualized, analyzed and enhanced by leveraging the capability of WBM. The techniques for recovering business processes from e-commerce systems are presented in our previous work at WCRE 2004 [7], ICPC 2006 [6], ICPC 2007 [2], and PCODA 2007[1]. In this paper, we present the overall steps for recovering business processes from e-commerce systems, and major features of the tool.

2. RECOVERING BUSINESS PROCESSES

As illustrated in Figure 1, the procedure for recovering business processes consists of static analysis of the source code of all tiers, dynamic analysis of the execution, verification of the results of the static analysis using the results of the dynamic analysis, and visualization of the recovered processes for developers using Eclipse IDE, and for business analysts using IBM WBM.

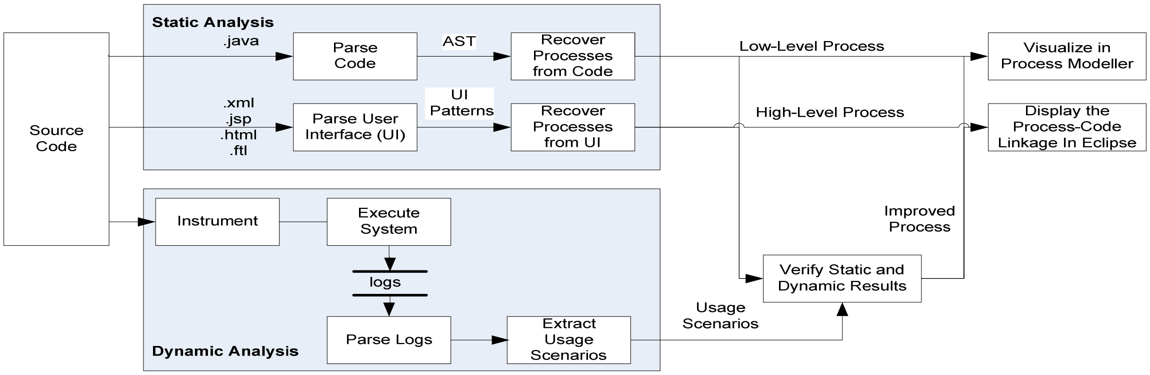


Figure 1: Steps for Recovering Business Processes from E-Commerce Systems

Static Analysis

The UI of e-commerce systems can be implemented using a variety of languages. Our tool can parse UI pages written in HTML, JSP, XML and FTL (Freemarker Template Language). Our tool can parse business logic tiers which are implemented in Java. Databases accesses are captured in the business logic tier, which issues queries and updates to the database tier. Our tool traces the code in all tiers to reason over all possible behaviors that might arise at run-time. An e-commerce system typically implements a large number of tasks. To reduce the complexity of representing the recovered business processes, our previous work [2] represents the business processes in terms of two abstraction levels: high-level and low-level business processes: (1) High-level business processes give an overview of business operations and contain tasks which require user interactions or which are executed by back-end components in the business logic tier; (2) low-level business processes contain the detailed steps performed in the back-end components. Separating complex processes into two levels provides a clearer view of the overall structure of an e-commerce application while hiding processing details.

A UI page can be decomposed into different UI widgets and each widget may contain other widgets. For example, an input form can encapsulate input fields, links and submit buttons. Each widget can be mapped to a particular task. However, we cannot make each widget a separate business task, otherwise the recovered business processes will be too difficult to understand and analyze since the processes will contain too many trivial tasks. To recover useful business tasks while dealing with the complexity of UI designs, such as hierarchical widget structures and arbitrary page transition structures, we make use of UI design patterns to locate tasks at the appropriate level of granularity and to identify the control flow between these tasks from the UI implementation. A UI design pattern describes a single unit of functionality delivered through a group of UI widgets [3]. For example, a UI search pattern is composed of a text field for entering search criteria, a submit button for triggering the search functionality, and a table for displaying the search results. Each UI screen or webpage implements several UI design patterns. We capture business tasks by identifying UI design patterns implemented in a page, and the control flows between tasks by tracing the navigation flow among pages [1].

To recover low-level processes in back-end components, we first parse the Java source code implementing the business logics. The parsed source code is stored as an abstract syntax tree (AST). We explore the AST and trace the flow of execution in the AST. Instead of simply mapping each Java method to a business task, we developed criteria for detecting the business relevance of an identified method and for separating business relevant methods from programming specific methods, such as utility methods. For example, "search for product" is a business relevant method. "establish a network connection" is a business irrelevant utility method and provides the foundation to build more sophisticated functionality.

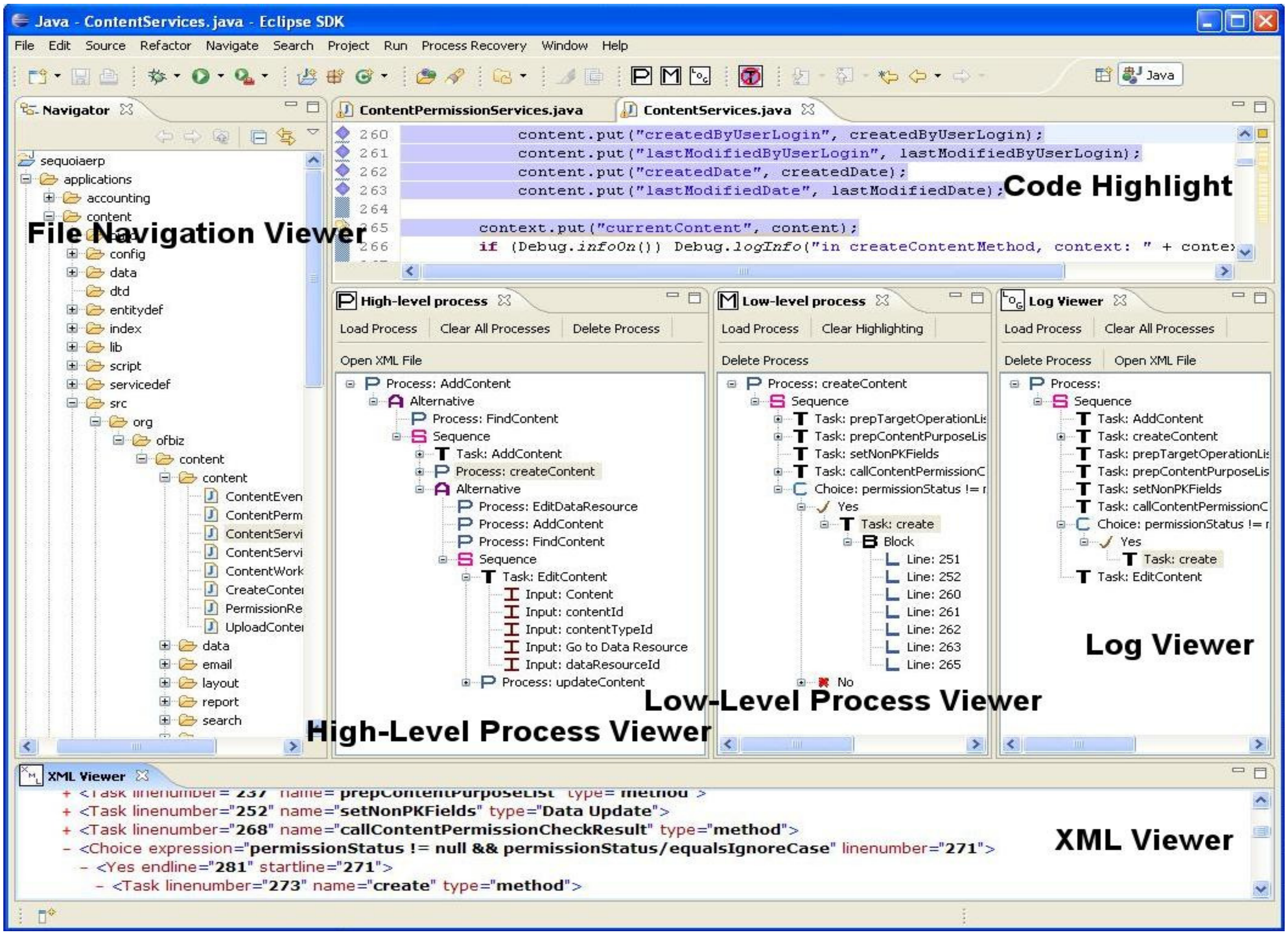
For an e-commerce website, the data described in a business process supports all the functions related to business operations. For example, for a customer to make a purchase, the e-commerce system must contain a catalog of products for sale (i.e., catalog data), and the data associated with processing orders (i.e., tax and shipping data). Such data is directly or indirectly created or changed by customers of the e-commerce website as a result of their interactions with the website. For example, customer orders are considered as business data, which affect the inventory levels. The data is also populated and stored in the database tier. A task takes business data as inputs and generates business data as output. We use business data as criteria to locate code blocks which manipulate business data and consider such code blocks as business relevant tasks [6]. Business data is detected by tracing databases accesses in the business logics tier.

Dynamic Analysis

To improve the accuracy of the recovered business processes, we use dynamic analysis techniques to observe the behavior of the system at run-time. We instrument the UI and business logic tiers at various points, such as method invocations, UI requests and responses, page transitions and the flow of business data. During the system execution, we record the tasks performed throughout the three tiers when a user interacts with the systems to fulfill business processes. When a user conducts a particular usage scenario, information regarding each step in the scenario is recorded in logs. We analyze application logs to generate a sequence of business tasks which represent a usage scenario for fulfilling a business process.

Results Verification and Visualization

Static analysis cannot correctly recover business processes when run-time information is needed, therefore, we use dynamic analysis to match statically recovered tasks with tasks recovered



```
            content.put("createdByUserLogin", createdByUserLogin);
            content.put("lastModifiedByUserLogin", lastModifiedByUserLogin);
            content.put("createdDate", createdDate);
            content.put("lastModifiedDate", lastModifiedDate);

        context.put("currentContent", content);
        if (Debug.infoOn()) Debug.logInfo("in createContentMethod, context: " + conte
```

```
+ <Task linenumber="237" name="prepContentPurposeList" type="method">
+ <Task linenumber="252" name="setNonPKFields" type="Data Update">
+ <Task linenumber="268" name="callContentPermissionCheckResult" type="method">
- <Choice expression="permissionStatus != null && permissionStatus/equalsIgnoreCase" linenumber="271">
  - <Yes endline="281" startline="271">
    - <Task linenumber="273" name="create" type="method">
```

Figure 2: Overview of the Business Process Explorer

dynamically through usage scenarios. Depending on the matching results, we can improve the accuracy of the recovered processes by removing misidentified tasks from the static results and adding missing tasks to the static results. To keep the links between elements in the recovered processes and artifacts in the e-commerce system, we record the code range of a task implemented in an e-commerce system. The links can be viewed in the Eclipse IDE. To visualize the recovered business processes, we developed a converter which transforms the recovered processes into the format recognized by the IBM WBM.

3. TOOL FEATURES

The current version of our process explorer tool provides the following features: traceability between the code and the recovered processes, consistent views for displaying the processes recovered from various sources, and integration with the IBM WBM, a leading commercial business process modeling tool.

Traceability between the code and the recovered processes: To allow developers to quickly locate the code corresponding to a task, a control flow element or business data, we track the relevant lines in the code relating to a process element. For example, as shown in the low-level process viewer of Figure 2, the Block icon in the tree structure presents the lines of the code that implement the task “create”. When a user double clicks on the task node in the low-level process viewer, the Java source file which implements the business process is automatically loaded into the Eclipse text editor. In addition, the lines of code corresponding to the clicked task are highlighted. A developer can refine the recovered process by manually examining and changing the code blocks highlighted. We define an XML schema for representing the different levels of the recovered processes and the usage scenarios. An XML viewer (as shown in Figure 2) is used to show the underlying representation of the recovered high-level processes, low-level processes and the usage scenarios. A developer can directly edit the XML representation of a process to refine a recovered process.

Consistent views for displaying the processes recovered from various sources: A high-level business process contains a collection of tasks and sub-processes which can be further broken down into a collection of tasks or sub-processes. A task in the high-level business processes is recovered from a UI design pattern. A sub-process in a high-level process hides detailed tasks performed by a back-end component. Such a back-end component is implemented using Java. A low-level process corresponds to a sub-process in a high-level process, and expands the detailed tasks performed in a back-end component. As a back-end component may further interact with other back-end components, the low-level process can contain tasks and sub-processes. To provide a hierarchical view of business processes, we represent the process as a tree structure in the viewers (e.g., high-level process viewer). Nodes annotated with different icons in the trees represent various process elements, including control flow elements (i.e., alterative, choices, and sequence), tasks and business data as parameters to tasks. As shown in Figure 2, the high-level process viewer illustrates a high-level process in a tree structure. A sub-process can be loaded and viewed as a separate process using the same tree structure in the low-level process viewer. The usage scenarios extracted from the logs can be visualized in the log viewer using the same tree structure, as shown in the log viewer in Figure 2. Consistent presentations of the recovered information from various sources allow developers to compare the results of static and dynamic analysis of the same business process.

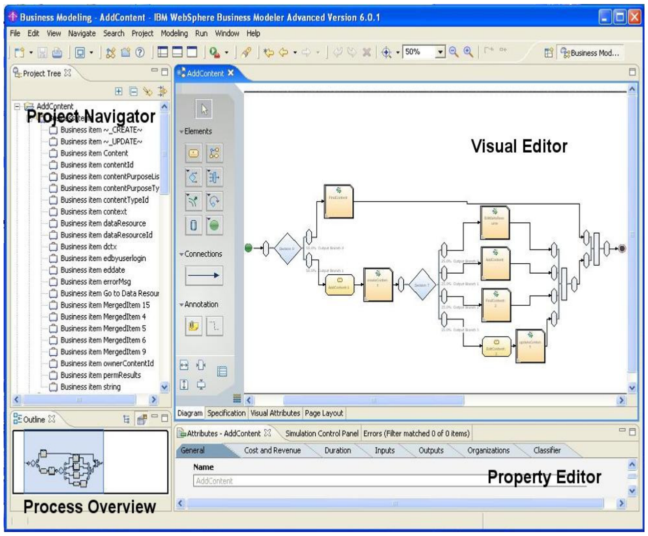

Business Modeling - AddContent - IBM WebSphere Business Modeler Advanced Version 6.0.1
File Edit View Navigate Search Project Modeling Run Window Help
50%
Business Mod...
Project Tree
AddContent
Project Navigator
Business item ~_CREATE~
Business item ~_UPDATE~
Business item Content
Business item contentId
Business item contentPurposeLis
Business item contentPurposeTy
Business item contentTypeId
Business item context
Business item dataResource
Business item dataResourceId
Business item dctx
Business item edbyuserlogin
Business item eddate
Business item errorMsg
Business item Go to Data Resour
Business item MergedItem 15
Business item MergedItem 4
Business item MergedItem 5
Business item MergedItem 6
Business item MergedItem 9
Business item ownerContentId
Business item permResults
Business item string
AddContent
Elements
Connections
Annotation
Visual Editor
Diagram Specification Visual Attributes Page Layout
Outline
Process Overview
Attributes - AddContent Simulation Control Panel Errors (Filter matched 0 of 0 items)
General Cost and Revenue Duration Inputs Outputs Organizations Classifier
Name
AddContent
Property Editor

Figure 3: Visualizing Recovered Processes in IBM WBM

Integration with a commercial business process modeling tool: To ease the adoption of recovered processes in practice, we automatically transform the process definitions from our proprietary schema to the XML schema recognized by the IBM WBM. Our recovered processes can be visualized in WBM. As shown in Figure 3, a business process recovered by the business process explorer is being visualized in WBM. Each element type in the process is represented by a different shape in the visual editor. Properties of each element can be modified by adjusting the values in the property editor. In the project navigator illustrated in Figure 3, a user can view the list of business data currently defined in the process. The visualization allows a business analyst to navigate through the recovered processes and to give more meaningful names to the tasks in recovered processes. Using WBM, business processes implemented in the source code can be kept up-to-date. We can also leverage the powerful features provided by WBM to analyze the recovered processes. For example, WBM provides a simulation tool that allows the user to simulate the execution of a process to locate possible design problems. Consequently, the process can be optimized to improve the performance of the organization which owns the e-commerce systems.

4. DEMONSTRATION

In this tool demonstration, we will give an overview of business processes. We will use SequoiaERP, an open source business application, from OFBiz Project [4], to demonstrate our business processes recovery procedure. We will then give a tour of our tool as an Eclipse plug-in. We will demonstrate the features of our tool as discussed in Section 3. We will illustrate the high-level processes in the high-level process viewer and the low-level processes in the low-level process viewer. We will show the recovered usage scenarios from the execution logs in the log viewer. We will demonstrate how to verify the results of static analysis using dynamic analysis. Finally, we will introduce IBM WBM and visualize the recovered processes using IBM WBM.

5. CONCLUSION

Rapid changes to business requirements are forcing organizations to adapt their business processes and to evolve their supporting e-commerce systems. However, the design and implementation of software systems rarely evolve consistently with the changing business needs. By using our business process explorer tool, we can automatically recover business processes using static and dynamic analysis techniques. Establishing links between the implementation and the specification of tasks in e-commerce systems can reduce the cost of maintaining and enhancing such systems. Furthermore, business analysts can visualize the recovered processes in IBM WBM and simulate the processes to better understand their performance characteristics and identify design problems. With the help of our tool, business analysts and developers can reference the same business processes. Therefore, our tool supports the co-evolution of the business processes and the e-commerce system, improves the responsiveness of the software system to changing business requirements, and reduces the cost for the software evolution.

6. ACKNOWLEDGMENT

We would like to thank Mr. Maokeng Hung at ASUS Tek Computer Inc. and Mr. Justin Everett at the Queen's University for their feedback on the initial prototype of the business process explorer tool.

7. REFERENCES

[1] K.C. Foo, J. Guo, and Y. Zou, "Verifying Business Processes Extracted from E-Commerce Systems Using Dynamic Analysis", In Proceedings of International Workshop on Program Comprehension through Dynamic Analysis, 2007.

[2] M.K. Hung and Y. Zou, "Recovering Workflows from Multi Tiered E-commerce Systems", In Proceedings of International Conference on Program Comprehension, 2007. pp.198-207.

[3] D. Sinnig, "The Complexity of Patterns and Model-Based UI Development", Mater thesis in Department of Computer Science, University of Concordia, Montreal, Canada, 2004.

[4] The Apache Open for Business Project, http://ofbiz.apache.org/.

[5] WebSphere Business Modeler, http://www-306.ibm.com/software/integration/wbimodeler/.

[6] Y. Zou, M.K. Hung, "An Approach for Extracting Workflows from E-Commerce Systems", In Proceedings of International Conference on Program Comprehension, 2006. pp.127-136.

[7] Y. Zou, T. C. Lau, K. Kontogiannis, T. Tong, and R. McKegney. "Model Driven Business Process Recovery". In Proceedings of Working Conference on Reverse Engineering, 2004. pp. 224-233.