DigitalAssets Discoverer: Automatic Identification of Reusable Software Components

Eduardo Machado Gonçalves
DigitalAssets Innovation Lab
Est. Telebras, km0,97 - Inovasoft
13081-970 – Campinas/SP – Brazil
+55 19 3705.5779
eduardo.machado@digitalassets.com.br

Marcílio da Silva Oliveira
DigitalAssets
Rod. SP340, km118,5 Prédio 9C
13086-902 – Campinas/SP – Brazil
+55 19 3705.5777
marcilio.oliveira@digitalassets.com.br

Kleber R. Bacili
DigitalAssets
Rod. SP340, km118,5 Prédio 9C
13086-902 – Campinas/SP – Brazil
+55 19 9605.6465
kleber.bacili@digitalassets.com.br

Abstract

DigitalAssets Discoverer is a tool that implements a group of indicators for automatic identification of software components that can be reused in the development of new applications and Web Services. This tool brings into light the J2EE applications portfolio developed in-house, increasing productivity and anticipating the ROI in companies.
The process of components harvesting and analysis uses an interactive user graphical interface that enables the tuning of selected indicators, visualization of the results and publishing the identified components into a reusable software development assets repository [1].

Categories and Subject Descriptors. D.2.13 [**Software Engineering**]: Reusable Software – *Domain engineering, Reusable libraries and Reuse models.*

General Terms: Management, Measurement, Economics, Human Factors, Standardization.

Keywords: Software components, reuse, services-oriented architecture (SOA), software assets, reusable assets identification, reverse engineering.

1. Introduction

The component-based development, combined with services-oriented architecture (SOA), is seen as a way to increase productivity and quality in software development. This is achieved by reusing already available parts (components) as building blocks for the new applications.

However, the task of tracking the reusable assets from a legacy applications portfolio, as they are consolidated and populated into digital libraries across environments, is highly dependable on supporting tools. [4]

According to this, DigitalAssets Discoverer [3] is a tool developed to provide this support by using innovative mechanisms to scan the legacy applications and identify the components eligible for future reutilization. This process enables the return on investment already made in software development.

An intuitive graphical interface was created to help the analyst easily inspect a group of applications, configure and trigger the identification mechanisms, tune and reapply them in the analysis process.

In order to enable tool integration and results sharing, the representation was based on the Reusable Asset Specification – RAS model [2].

2. Identification Methodologies

The reusability indicators were developed using approaches based in reverse engineering, legacy systems reengineering, and architectural analysis. The result was an implementation of three indicators following a set of heuristics:

- Analysis of cohesion and coupling of components;
- Recovery of the application architecture according to the dependency matrix;
- Pre-defined components in distribution packages such as JAR or ZIP.

In order to give a better visual representation of the candidate to be a reusable component, the tool applies a set of heuristics suggesting the possible names for the identified assets. The suggestions are based on the artifacts (classes, packages, etc.) that better translate their functionalities.

3. Identification Tool

The DigitalAssets Discoverer integrates source code analyzers with their indicators. Figure 1 presents the methodology for this tool subdivided in four stages:

i. *Scanning* of the existing JEE applications selected for analysis;
ii. *Creation of a knowledge base* starting with the analysis of the scanned source-code and the identification of the internal and external references;

OOPSLA'07 October 21–25, 2007, Montréal, Québec, Canada.
ACM 978-1-59593-865-7/07/0010.

iii. *Execution of the indicators*. This stage can be done repeatedly considering the reconfiguration of the indicators in each interaction aiming an optimized result;

iv. *Harvesting*. The tool presents a group of artifacts suggested as reusable assets where the analyst decides the relevance of the suggestions. The tool gives an option of capturing and exporting the components as a RAS [2] package.

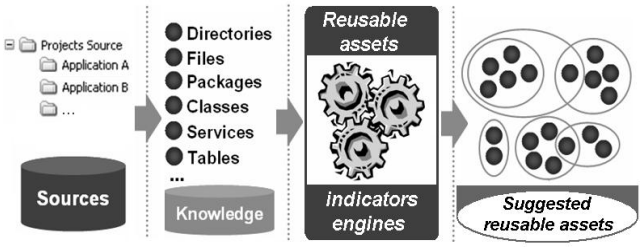

Figure 1. Tool Identification Cycle

4. Interface Features

The DigitalAssets Discoverer offers a graphic-based visualization of the knowledge foundation and the components suggested as reusable (figure 2). The visualization of the relationships within the classes is flexible and scalable. This interface offers resources that help the understanding and validation of the component identified as a candidate, highlighting its related classes.

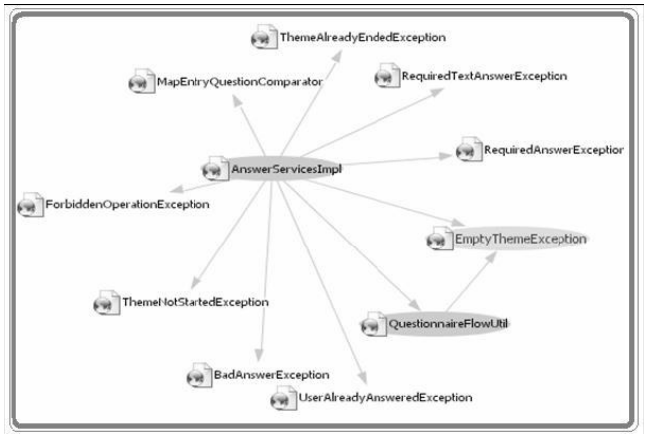

Figure 2. Assets' Artifacts and the Relationships Visualization**.**

5. Pratical Results

Various tests for proof of value were applied in typical web applications in order to validate the tool and its methodologies. The analyzed applications include component-based solutions as well as legacy applications with little architectural planning in terms of modularization.

The results pointed to a good component automatic identification rate. As expected, the results were more successful in architecturally consistent applications. In the case of chaotic architecture applications the tool pointed various quality features and some suggestions to improve its design before identifying reusable components. This shows that each structure requires a specific set of indicators.

6. Conclusion

This paper presents a tool for automatic identification of software components focusing in some of the main applications in modernization scenarios [4] and component-based development. Through R&D efforts in the DigitalAssets/Unicamp Software Innovation Lab some methodologies were integrated with the capability of identifying software assets (components, services, procedures etc.) with reuse potential.

Component-based development aims to increase productivity and quality while reducing costs and time-to-market of new applications. One of the main concerns in software reuse is to optimize the process of finding suitable components for a given need and provide enough information for the proper and efficient use of these components. The tool for Automatic Identification of Software Components called DigitalAssets Discoverer presented in this work aims to help companies in their reuse and SOA initiatives anticipating the ROI (Return on Investment).

7. Future Work

As a future work we intend to apply these techniques in a wider variety of existing applications through several technological platforms to determine more precisely the impact of this approach supporting SOA and reuse initiatives.

Various other resources are being put into practice. Some of the features we are working on are: reading the source-code directly from version managers; increasing the number of programming languages supported by the tool, developing new indicators and evolving the current ones.

Acknowledgments

Special thanks to the DigitalAssets Innovation Team at Unicamp. Thanks for all the research and development work that made possible the integration between usability and innovation.

References

[1] Bacili, K., Oliveira, M., "DigitalAssets Manager: sharing and managing software development assets", OOPSLA'06 Demo Session, ACM, NY, 2006, pp. 700-701.

[2] RAS (2006), "Reusable Asset Specification, OMG Available Specification". Object Management Group, Available at: ttp://www.omg.org/technology/documents/formal/ras.htm. Last view: 2007-03.

[3] Oliveira, M., Gonçalves, E.M., Bacili, K., "Automatic Identification of Reusable Software Assets: Methodology and Tool". Information Reuse and Integration IRI'07 (to publish).

[4] The Object Management Group (OMG), Architecture-Driven Modernization Task Force (ADMTF), "Architecture-Driven Modernization Scenarios", white paper, 2006, available at http://adm.omg.org/ADMTF_Scenario_White_Paper(pdf).pdf . Last view: 2007-07.