Hengtian Reengineering Expertise Overview

1. Overview of Reengineering

Reengineering, also known as renovation or reclamation, is the examination and alteration of a subject system to reconstitute it in a new form and to subsequently implement it. Reengineering generally includes some form of reverse engineering followed forward engineering or restructuring. This may include modification with respect to new requirements not met by the original system. The following graph depicts the general model of reengineering.

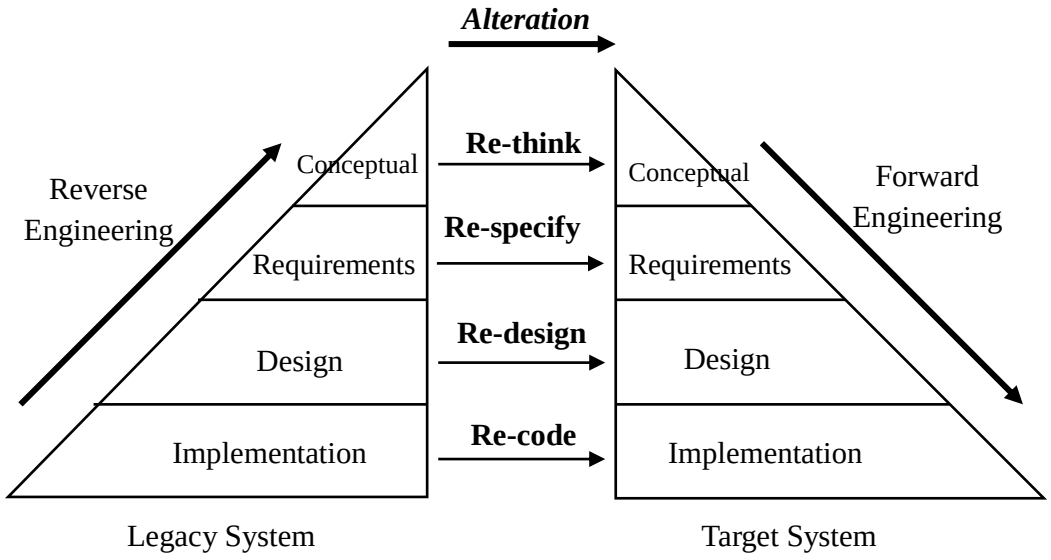

Fig.1 General Model of Software Reengineering

2. Business Drivers

Although many enterprises have reengineered or replaced their legacy systems since the 1990s, there are still many legacy systems in various domains, like the financial domain, that remain to this day. On the other hand, as businesses develop, a newly developed system will gradually become incapable of satisfying the new market, and the system itself will become a "legacy system". Developing new systems to replace the legacy system will require software investments, but after the economic crisis many enterprises have cut their IT facilities' budgets. Compared to developing a completely new system, the cost of using reengineering approaches to renew the legacy system would be decreased by 60% . Therefore, reengineering would yield more opportunities in further software engineering. The advantages of reengineering are:

- Improved reliability, maintainability, performance, etc
- Complete comprehension of the legacy system

- Decreased development costs and risks
- Shortened system development cycle
- Undisrupted business continuity

3. Challenges for Reengineering and Our Outstanding Solutions

***The Challenge** (1)*

Reverse Engineering
Difficulties include outdated techniques, architecture regression, and a lack of documents and developers. IT staff could not fully comprehend the whole system. There are four critical difficulties: business comprehension from program language, design recovery from architecture-degenerated software, the gap from programming to human thought, and combination of bottom-up and top-down analysis.

Too Much Human Effort
Although there are many software clustering, program slicing and pattern matching approaches that facilitate reverse engineering, most tools require a great deal of human interaction. It often takes more than 60% of total resources and project time.

Our Solution

Program Comprehension Framework
Program comprehension and analysis is one of the critical difficulties for reverse engineering. We proposed a framework of business rules extraction and comprehension of legacy systems. The framework is composed of five parts: legacy code slicing, domain variable identification, data analysis, business rules representation and verification. It has been applied to analyze a large financial legacy system.

***The Challenge** (2)*

Migration
Migrating the legacy system with outdated techniques to a new platform is difficult for almost every system.

Our Solution

Reengineering Legacy System into J2EE Partition-Based Distributed Environment
Compared to previous target environments, a Partition-Based Distributed Environment has several advantages, including high performance, scalability and availability. The target system would take advantages of the J2EE environment. The framework consists of code conversation, component identification, component interface modeling and target system deployment. This model has been applied into one equity trading system.

***The Challenge** (3)*

Efforts and Risk Assessments

Since assessments cover various factors like business requirements, software process, platform, techniques and project staff, it is difficult to provide a standard assessment model. Most current models can only be used for legacy systems of specify domains.

Software Maintenance

Software maintenance is inevitable during reengineering. Therefore, the legacy system for reengineering may not be the latest version, which leads to great challenges and risks.

Our Solution

Spiral Model

In this model, the legacy system is reengineered into new platform iteratively. In each cycle, only a part of the system is alternated and deployed in the new environment. Compared to the big-bang approach, this model could decrease reengineering risk.

Since each module alteration or adding new requirements sub-procedure is performed on a trustable stable system, the risk is decreased each time, which reduces the total risk of the legacy system reengineering.

4. Our Research on Reengineering

From the industry experience, some reengineering models including Partition-Based Software Reengineering Framework, Incremental Software Reengineering Model and Global Cooperative Software Reengineering Model were proposed. Additionally, a reverse engineering framework on recovering business rules was brought out. Now we are conducting research on the reengineering of large-scale software systems. All our research achievements have been published on the international conferences or journals including ICSE2006 and ICSM2007.

Following are part of the published papers related to reengineering.

1. "Reengineering Standalone C++ Legacy Systems into the J2EE Partition Distributed Environment", In Proceedings of the 28th International Conference on Software Engineering (ICSE'06 Far East Experience Track), Shanghai, China, May 20-28, 2006, 525-533.
2. "Experience Report: Reengineering Standalone System into the Service-Partition Distributed Environment", In Proceedings of IEEE International Conference on Software Maintenance (ICSM'07), Paris, France, Oct 2007, pp. 477-480.
3. "Business Rules Extraction from Large Legacy Systems", In Proceedings of the European Conference on Software Maintenance and Reengineering (CSMR'04), Tampere, Finland, Mar. 24-26, 2004, 249-258.
4. "Inter-procedural Static Slicing of Concurrent Programs Based on threaded System Dependence Graph (tSDG)", WSEAS transactions on information science and applications, vol. 4, Issue 10, Oct 2007, pp. 1249 – 1254.

5. "A Framework of Business Recovery from Large Legacy Systems", WSEAS Transactions on Information Science & Applications, vol. 3, Issue 3, Mar. 2006, pp. 576-583.

6. "A New Approach of Component Identification Based on Weighted Connectivity Strength Metrics", Information Technology Journal, Jan. 2008, 7(1):56-62.

5. Our Achievements on Reengineering

(1) Customer Feedback and Media coverage

"What started as a small, offshore R&D facility for State Street Corp is now a fully fledged technology center in China that has re-engineered 100 of its legacy applications. The increased productivity has delivered more benefits than the labor arbitrage alone ever could."

——Jerry Cristoforo, State Street EVP and CTO

"The system is now in production with four times the previous volume and it took less than six months" -- The client's CIO Joseph C. Antonellis described the success of this project.

——From "An IT Flower Blooms in China", CIO Magazine

"Labor savings have been significant (approximately 25% of the cost for U.S.-based staff), but savings through revamped legacy applications have been even greater, avoiding expensive replacements. Re-engineering has been achieved for less than 2% of the cost of system replacement and has saved millions of dollars."

——From "Case Study: State Street Corp. Takes a Chinese Road to Supercharged Applications", Gartner.

(2) Case Study

Equity Trading System Reengineering
Fund Administration System reengineering