Business Rules Extraction for Legacy Brokerage System

Client Overview
The Client, North America’s leading customs broker, offers a range of services and solutions to companies with customs brokerage, compliance, trade consulting and freight needs. The Client is expected to grow revenues by 50% over the next four years, both organically and through strategic acquisitions. The brokerage operation runs on a proprietary system, Online Customs System (OCS), which is written in VAX COBOL utilizing ISAM, VSAM and RDB databases. OCS is the account data repository for all of the Client’s lines of business.

Client Challenge
- OCS was highly customized, contained 25 years of business intelligence and approximately 2,500,000 lines of code
- Skilled resources with mainframe COBOL application development experience are retiring
- The OCS is based on legacy technology, which is at or near its end of life
- The business operations required a strong, robust and cost effective technology that supported the business need of information processing services
- Ability to meet evolving business needs, competitive opportunities and successful acquisition integrations were hindered
- Rising costs in technology maintenance and support
- Development inefficiencies resulted in continuous accumulation of lines of code
  - These inefficiencies created a large backlog of system enhancements (the backlog equated to multiple years of development resources)

Trinity’s Solution
Trinity provided software engineering analysis of the current (legacy) applications’ source code. This analysis created functional, high level business documents that described the current source code in quantitative terms.
Summary Scope of Work

The Client asked Trinity to extract all business rules based on logic currently embedded within the OCS platform, as well as other integrated systems. These business rules were then output into technical design documentation. The Client also requested documentation and verification of these business rules, including the following:

- All existing interfaces, data analysis and reporting tools
- Verification of performance levels within OCS
- Analysis and documentation of OCS’ business rules, business functions and business activities

Abstract Syntax Tree (AST)

The core element of Trinity’s business rules extraction process (Trinity's BR Plus™) is the AST. The AST represents the most fundamental element of logic contained with the millions of lines of application code. Trinity parsed the OCS codebase, which is the act of interrogating and decomposing code to its most elementary state. This allowed Trinity to identify and extract business rules utilizing automation. Operating at the AST level, Trinity performed the following:

- Identified and extracted existing rules
- Identified pattern aggregation candidates for more efficient disposition
- Abstracted logical code elements in preparation for analysis and target code refactoring
Business Alignment

Trinity collaborated with the Client to identify the Business Requirements (BRs), those essential enterprise functions that OCS supported. The Business Functions (BFs) and the Business Activities (BAs) of those requirements were captured and documented. Trinity developed a Business Alignment Document (BADOC), which depicted supporting code elements in outline form, including Entry Points for each BF and BA. The BADOC analyzed OCS and aligned code elements with the functional organizational departments that they supported.

Business Alignment and Entry Points

Business Requirement Definition

Entry Points serve as the basic connection between the actions of a business and the IT systems and applications that support those actions. The Client identified Entry Points for OCS and delivered the completed BADOC to Trinity for the next step in the process, Business Topology.
Business Topology

With Entry Point elements identified in the BADOCS, Trinity utilized its proprietary technology to organize and form the Topology of the codebase. Topology refers to the layout and isolation of specific code segments directly against business functionality. Trinity harnessed all knowledge derived from OCS’ Patterns and Aspects library. The goal of the process was to isolate all code uniquely relevant to each BR, BF, and BA, regardless of overlap. Topology enabled the isolation and identification of dead and null code as well. A sample of Code Topology is shown below:

Variable Translation Process

The purpose of the Variable Translation process is to make output documents as English-like and understandable as possible. The Client’s subject matter experts defined the legacy variables and resolved conflicts between variable meaning usage and/or assignment. For example, one variable might have been used in two different ways or used in two different places with different meanings. An overview of the Variable Translation Process is shown below:
Business Rules Extraction
Trinity used its proprietary Business Rules Engine to extract the active and relevant business rules within OCS. Business Rules Extraction included the identification and archiving of code slices, which are particular paths of execution through the code. Candidate Rules were extracted and identified as either business, technical, validation or other types of rules. Candidate Rules were extracted based on the following:

- Conditionals such as IF, Evaluate, When, etc.
- Calculations
- Variable usage
- Hard-coded literals

Trinity exported these rules to an Excel spreadsheet based on the Client’s specifications. Execution order and flow are critical for a procedural language such as VAX COBOL. Trinity documented all data access activities and all specified application artifacts were used as inputs to Use Case or User Story documentation. The format of the Business Rules document was customized according to the Client’s specifications. Samples of a Business Rules Engine and a Business Rules Document are shown below:
Use Cases

Trinity provided Use Case documents to demonstrate the interaction between an actor and OCS. The actor triggered a specific activity that resulted in a particular system response. These Use Cases captured OCS’ behavioral requirements by detailing scenario-driven threads through the functional requirements. The format of the Use Case was customized according to the Client’s specifications. A sample Use Case is shown below:

Use Case

Results

Trinity’s proprietary BR Plus™ process revealed that 25% of the legacy code was either dead or obsolete and did not require modernization. Further, Trinity identified a considerable amount of duplicate code that resulted in significant cost savings for the Client. Also, Trinity delivered very detailed documentation of the Client’s application portfolio. The Client used Trinity’s deliverables to verify that all functionality of the VAX COBOL was captured in its target state environment, C#.

We invite you to contact us to learn more about our expertise, proven processes and proprietary technology. We would welcome the opportunity to provide an overview of our solutions. Call (877) 615-1606 or email sales@tringroup.com today.