Software Reuse: Domain Analysis and Design Process

Keywords:
- Object-Oriented
- Ada Language
- Software Engineering
- Software Reuse

Extended Abstract:

Note: This was originally submitted as a tutorial; parts have been modified; A paper submission is pending.

This paper addresses the basic software reuse principles and goals, various domain analysis and design processes, important ways to reduce cost, reduce schedule, and produce quality software. It lays out an innovative, comprehensive, and unified approach to software projects - one that integrates object-oriented components, design patterns, and conversion tools that make software written for one purpose serve other uses. It will help managers, analysts, and developers to halt the rapidly rising curve of software costs, obtain higher project success rates, increase product maintainability and reliability, cut development delays, improve application performance, contribute to the evolution of a common component warehouse, and gain greater portability, scalability, understandability, and standardization.

Intended audience: This paper is intended for software programmers, software developers, project engineers, software quality engineers, quality assurance managers, testers, engineering managers, computer engineers, system analysts, system engineers, and program managers who are involved or likely to be involved with software.

Upon completing this tutorial participants should be able to:

- Halt the rapidly rising curve of software crisis
- Obtain higher project success rates
- Cut development delays
- Improve application performance
- Increase products maintainability and reliability
- Gain knowledge of the software reuse processes and techniques for eliminating excessive and unnecessary costs, increasing productivity, and other beneficial effects
- Achieve significant cost reductions while simultaneously increasing quality and reliability by utilizing the software reuse
- Have an excellent understanding of the software requirements for reuse process
- Use the latest techniques in domain engineering to identify potential specifications and modules for reuse
- Use the Internet for economical software development
- Reuse software management caveats - lessons learned for improving practices and processes

Detailed tutorial outline:

The tutorial will address the following main topics:

1. Software Reuse Strategies
2. Software Reuse Paradigm
3. Software Reuse Processes
4. Planning for Software Reuse
5. Software Reuse Management
6. Distributed Object Technology
7. Domain Models
8. Domain Analysis and Design Process Method
10. Domain Implementation
11. Application Development with Reuse
12. Repositories and Assets

The tutorial topics will be covered in 6 sessions as outlined below:

1. Summary of Course and Objectives
   A Live Scenario
   Software Crisis
   Importance of Software Reuse
   Software Engineering Goals and Principles
   Object-oriented Software Development Iceberg
   Life Cycle Importance of Object-oriented Software Reuse
   Object-oriented Software Reuse Processes
   Domain, Boundary, Environment, Specific Architecture
   Software Reuse Classification
   Formalizing Software Reuse Process
   Identifying Reusable Assets
   Software Reuse Maturity Model Concept

2. Planning for Software Reuse
   Preparing for a Culture Change
   Assessing Potential Problems
   Standards & Practices
   Reuse Issues
   Establish Domain
   Case Study
   Configuration Management
   Quality Management
   Operation and Maintenance of Reuse Center
   Customers and Users Participation
   Repository System Establishment and Maintenance
   Reuse Life Cycle Phases
   Integrate Reuse into Software Development and Maintenance
   Checklist: Suggestions for not failing a Reuse Project

3. Domain Engineering
   Application Engineering
   System Life Cycle Models
   Domain Repository and Application Development
   Selection Criteria of a Suitable Reusable Software
   Distributed Object Technology
   Common Object Request Broker Architecture
   Interface Definition language
   CORBA Architecture
   Quiz
   Domain Models
   Examples

4. Domain Analysis Methods
   IDEF Process Model
   Quiz
   Feature-oriented Domain Analysis
   Joint Object-oriented Domain Analysis Method
   Domain Analysis and Design Process Method
Domain Design Methods
Example

5. Domain Implementation
Asset Collection
Asset Submitting Schema
Asset Accepting Schema
Asset Certification
Level of Certification
Repository Mechanism
Reuse Asset Anomaly Process
Collection of Metrics
Building a System with Reuse Assets
Exercise

6. A Practical Case Study "Distributed Object Technology in a Three Tier Client Server Environment" - Discussion
Workshop Evaluation

Tutorial description for catalog, proceeding, and brochure:

Software Reuse gives software development projects a realistic roadmap to a reachable goal that will pay enormous dividends for many, many years to come. Ever since the first code was written for the first computer, software reuse has been both a sought-after opportunity and an intransigent problem. This course lays out an innovative, comprehensive, and unified approach to software development and maintenance projects—one that integrates object-oriented components, design patterns, and conversion tools that make software written for one purpose serve other users. Large-scale software systems used by government and corporations—they're anything but simple, but object-oriented programming has made these systems far easier to manage through software reuse.

Some of the highlights are software reuse taxonomy, paradigm shift, and importance of software reuse, reuse objectives, and software reuse evolution. Software reuse is a process of implementing or updating software systems using existing software assets. Assets can be defined as software components, objects, software requirement analysis and design models, domain architecture, database schema, code, documentation, manuals, standards, test scenarios, and plans. Software reuse may occur within a software system, across similar systems, or in widely different software systems. The object-oriented approach provides convenient ways that pre-tested software can be reused. List of repositories and assets that are easily available and accessible on Internet will be discussed.