

**Implementing Concept of Case Base Reasoning for Component Selection in
Component Based Software Engineering using Rating Factor**

Gagandeep Sethi^{*1}, Amandeep Kaur²

^{*1}M.Tech CSE, Lovely Professional University, India

²Assistant Professor, CSE, Lovely Professional University, India

gagandeepsethi.lpu@gmail.com

Abstract

Performing good component selection in Component Based Software Development plays a critical role in the success of a software project. Many techniques have been developed to select the component and to handle uncertainties and mismatches with the user's requirement. Incorporating and implementing the knowledge from the previous success or failures of the projects developed using CSBD approach must be followed to acquire appropriate ranking and usefulness of the component selected from the candidates or the alternatives of the desired requirement component. In this context I am going to develop an interface for the selection and retrieval of the component in the form of a web portal using Ajax technique and various decision parameters used to rank a particular component from the list of the candidate components.

Keywords: Component, component selection, previous knowledge, Ajax, web portal.

Component Based Software Engineering

From the last several years software development has been influenced by Component Based Software Development. Component Based Software Engineering is a branch of Software Development that deals in developing software systems by reusing the self dependent and cohesive components of existing software system. This approach is different from the traditional software development models. Using this approach by using components of loosely coupled software systems, developers tend to make a new system with less time and efforts by reusing existing compositions of a software. A software component can be defined as-“Any executable part or composition of existing software”. It can be a subroutine, class file, jar file or a module, a web service or a web resource that is executable composition of software, a self contained entity and can be a minimal unit of large software or it can be COTS component i.e. Commercial Off-the shelf component that are ready made available to incorporate them with the new software. A minimal example would be an online chat application component for a website to provide online help.

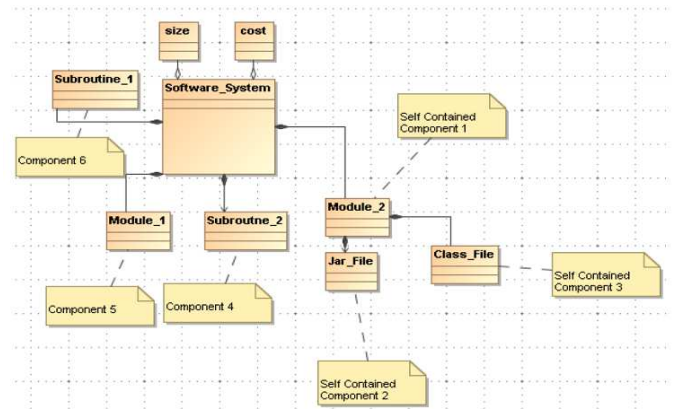


Figure 1.1 Componentized view of a software system

Reusable Component-A Minimal Illustration

Final software is a composition of reusable components. A reusable component is a part of the final software that is an independent part that can be executed separately and added or extended to another part or class of the module/component to make it a final part.

```

package Default;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ImagePanel extends JPanel {
    private Image img;
    public ImagePanel(String img) {
        this(new ImageIcon(img).getImage());
    }
    public ImagePanel(Image img) {
        this.img = img;
        Dimension size = new Dimension(img.getWidth(), img.getHeight());
        setPreferredSize(size);
        setMinimumSize(size);
        setMaximumSize(size);
        setSize(size);
        setLayout(null);
    }
    public void paintComponent(Graphics g) {
        g.drawImage(img, 0, 0, null);
    }
}

package Background;
import java.awt.*;
import java.awt.Graphics;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class ImageTest {
    public static void main(String[] args) {
        ImageIcon panel = new ImageIcon("koola.jpg").getImage();
        JFrame frame = new JFrame();
        frame.getContentPane().add(new ImagePanel(panel));
        frame.pack();
        frame.setVisible(true);
    }
}
    
```

Figure 2.1 Reusable Component example in Java

Figure 2.1 depicts a composed component i.e. ImageTest.class with the reusable component ImagePanel.class. Here ImagePanel is reused to set background image and is acting as base to set image to another frame. ImageTest has use composition/aggregation relationship to use that component i.e. In aggregation relationship, a method delegation is used with member object of a class. Aggregation can also be done in same manner just the ImagePanel has a strong relationship with ImageTest thus the composition relationship has been established between the two.

The Component Based Software Development Process

A Component Based Software Development process is different from traditional software development approach in the sense that the phases are different from the traditional software development processes.

The CBSD Process involves:

- 1) **Requirement Analysis:** In this phase requirement elicitation is done for the new software system to develop. They can be Functional or Non-Functional.
- 2) **Component Repository:** Every new software is a variety of existing software. Considering the same concept a repository is made including all the deployable compositions or parts of earlier developed software systems. These components are stored in a repository called Component Repository
- 3) **Component Selection:** Component selection phase incorporates selecting components to achieve required functionality in the new development.

4) **Design:** Accordingly partial software is developed and the design for the same is developed and tested.

5) **Development of new software components:** Rest of the desired additional components are designed to achieve full functional product.

6) **Integration:** Reusable components/COTS components are then integrated and adapted to new developed functionality using Middleware like Java Beans, COBRA, COM(Component Object Model)

7) **Test, Release and Maintenance:** The developed system needs to go through different types of tests like unit testing, integration testing, acceptance testing etc. Then if the test is successful then beta release is declared and accordingly maintenance is performed.

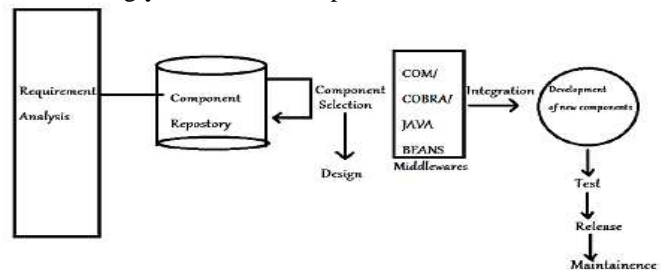


Figure 3.1 Component Based Software Development Process

Component Selection

Component selection plays the important role in CBSD. As the selection is based solely on the requirements and component should so be selected that it fulfils the user requirement with less cost and time. Many techniques for component selection has already purposed to select component with lesser mismatches but only few of them like PORE (Procedure Oriented Requirement Engineering) and MiHOS are on focus to deal the components with lesser mismatches.

PORE (Procedure Oriented Requirement Engineering)

In Procedure Oriented Requirement Engineering Technique, the component is selected using requirement engineering with following characteristics:

- a) PORE has a process model that identifies undertaken fundamental process like requirement acquisition and validation, supplier selection, software package and management of the system procurement.
- b) A method box that indicates methods, techniques and tools that is available to help and achieve each of the processes.
- c) A product model that enables effective product evaluation and selection using the use case modeling, goal-based requirement methods and architecture modeling techniques.

- d) It selects products by rejection that is if any product has not fulfilling the user needs then it is removed from the candidate products list.
- e) It uses techniques like card sorting, MDCM, AHP and laddering for requirement acquisition.

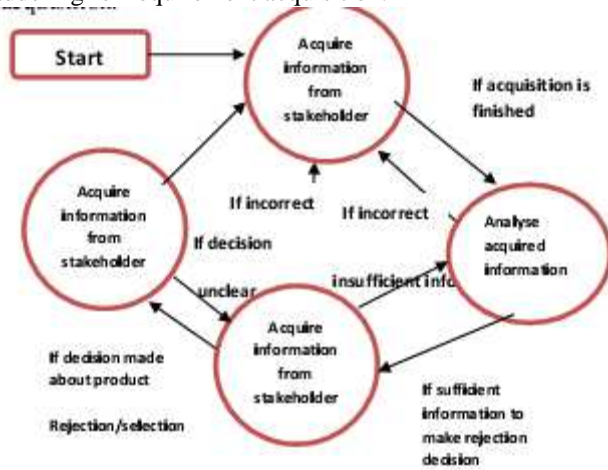


Figure 4.1 PORE Process

OSTO (Off the Shelf Option)

OSTO is the first COTS selection technique purposed in 1995, which presents a method for evaluating and selecting the components to be reused in software development.

- a) It provides a well defined, systematic and detailed evaluation criteria definition based on functional requirements, product quality characteristics, strategic concerns and architecture compatibility.
- b) It provides a method for estimating the relative efforts and Cost benefit analysis of different alternatives.
- c) It uses weighted score method (WSM) for better decision making criteria.

The idea is defined in the following figure:

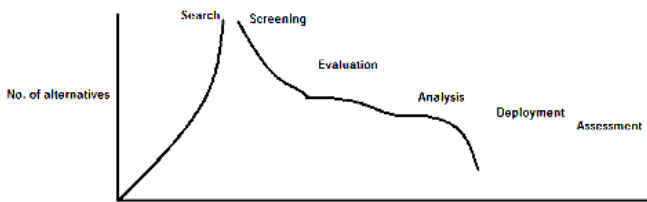


Figure 4.2 OSTO Process

Problem Formulation and Proposal

As the technology and requirements are growing, the time and cost is necessary for a successful project. The millions lines of code can't be written in limited time period. In context of this problem, CBSD can be followed instead of traditional software development process but if appropriate selection of the component is not performed and mismatches occurs

again and again then this approach has no use. To solve this different selection techniques are developed. Thus the area lies between the selection phase of the CBSD process and accordingly retrieval of the appropriate component with best match. All approaches follows the some kind of ranking and evaluation of component to make it in the candidate list and no technique has yet follows the previous knowledge of CBSD developed project so that the new project can be developed with the knowledge that if the component we are selecting with the specified requirements has failed or succeed the previous cases. This can be creating using a ranking algorithm to rank a particular component along with its different attributes and its previous cases.

An Interface can be developed known as "INTERFACE FOR COMPONENT RETRIEVAL AND SELECTION" using where we can apply the methodology as follows:

- Step 1: Create a repository of components
- Step 2: Add details and parameters for each component like it is an information system, or a widget
- Step 3: Add number of functions it contains like update record, online chatting, report generator etc.
- Step 4: Link it with previous projects ad create some cases based on that
- Step 5: While selecting create a logic to rank the component based on number of user requirements it is fulfilling, factor of success or failure from other projects, and user liking.
- Step 6: If component is suitable, select it and update Case if not then print Component not found.

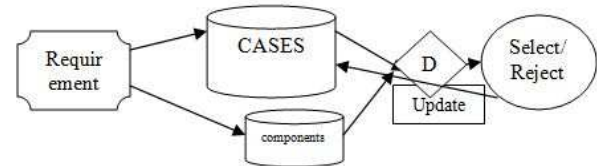


Figure 5.1 Purposed Methodology

References

- [1] Andrea Vescan, Crina Grosan (2008), "Evolutionary Algorithm for the Component Selection Problem", 19th International Conference on Database and Expert Systems Applications (IEEE Computer Society), pp. 509-513
- [2] Ardhendu Mandal and S.C. Pal (2012), "Emergence of Component Based Software Engineering", International Journal of Advanced Research in Computer Science and Software Engineering, pp. 311-315
- [3] D. Jeyamala et al.(2013), "FAULT-PRONE COMPONENTS IDENTIFICATION FOR REAL TIME COMPLEX SYSTEMS BASED ON

CRITICALITY ANALYSIS”, International Journal of Computer Science and Informatics, ISSN (PRINT): 2231 –5292, pp. 17-23

- [4] Hamdy Ibrahim, et al.(2011), “ UnHOS- A Method for Uncertainty Handling in COTS Selection”, Internation Journal of Energy Information and Communication Vol. 2, pp 21-48
- [5] Jeetendra Pande, et al. (2013), “ Optimal Component Selection for Component Based Software Development using Paliability Metrics”ACM SIGSOFT Software Engineering Notes, pp. 1-6
- [6] Miguel A Martinez and Ambrosio Toval (2008), “COSTRE: A Component Selection Method based on Requirement Engineering”, IEEE Seventh International Conference on Composition Based Software Systems, pp. 220-223
- [7] Pradeep Tomar et al. (2013), “New Algorithm for Component Selection to Develop Component-Based Software with X Model”, Lecture Notes on Software Engineering, Vol. 1, No. 3, pp. 298-302
- [8] Sayed Ahsan Fahmi and Ho-Jin Chu (2009), “A Study on Component Selection Methods”, ICACT 2009, Korea, pp. 288-292
- [9] Zahid Javed et al. (2102), “An Empirical Study of COTS Components Persuasion, Evaluation and Selection and Integration in Software houses Faisalabad, Pakistan”, IJCSI International Journal Vol. 9, Issue 6, pp.165-169