W-Shaped Framework for Component Selection and Product Development Process

Vinay, Manoj Kumar and Prashant Johri

SCSE, Galgotias University, Noida, India

Abstract: The IT industry started software development by using software components to reduce the efforts, the cost of the product and enhance the productivity and quality of the product. The quality and cost of a software product are highly dependent on the quality and cost of the components assembled to produce the product. There exist a variety of parameters for component selection such as usability, maintainability, reliability, security, functionality, cost of customization and integration. Hence, component selection is a critical, search driven and NP-hard problem. For a better component selection different component selection meta-models are available. Context Driven Component Selection (CDCS) provides a better way to evaluate different characteristics of a software component which is important for component selection from the developer’s point of view. There is a strong need to devise a component selection model and strategy to fulfill the needs of industry. In this paper we propose the W-Shaped model for component selection. This paper provides insight into the steps required for component selection and product development process. It helps the developers to systematically manage, customize and updating of different software components.

Key words: Data · Component Repository · Faceted Classification · Context driven approach · Component Selection, Quality factor.

INTRODUCTION

Today, the software industry is suffering from the problems of high development cost, low productivity and quality compromised end products. To counter these problems, Component Based Software Engineering (CBSE) provides a way to develop software systems in cost and time efficient manner with minimum effort [1]. This approach to software system development is based on the selection of efficient software components and assembling them using a well-defined architecture [2]. Component based software development is meant for system development using off-the shelf components, in-house components and components acquired from third-party developers. A software component provides a specified functionality in the system and help to minimize the time, cost and effort for software system development. Component selection plays a vital role in the CBSD process and faces several challenges. CBSE has some open issues in front of it like how to select the most effective component? What is the basis of component selection? What is the quality of a component? How to measure the fitness of the component? How to measure the quality of the software system developed by integrating different components? What is the impact of the quality of components on the quality of the end product? Different researchers tried to answer the above mentioned questions but those solutions are not fulfilling the objectives of the software industry. Earlier proposals provide the strategy for component selection based on single aspect but the component selection using single aspect is a biased approach. Earlier component selection models are based on functional and non-functional attributes of the software component [3].

Related Work: Various researchers proposed models for component selection based on specific aspects. Component selection requires further exploration based on different criteria, hence it attracts researchers to work in this field. OTSO [4] (off-The-Shelf Option) is an AHP based approach for component selection, progressive filtration and evaluation. It considers functional and non-functional attributes for component selection. The PRISM model [5] (Portable, Reusable and Integrated Software Module) focuses on generic component architecture followed by integration evaluation and field

Some of the challenges came out during the development of Component based system and we need to fix those challenges to enhance the quality of the system. Component filtration, evaluation and selection are a vital part of CBSE. Different challenges are related to ensure genericity of the component, evaluation of candidate components, criteria for component selection and multi-criteria decision making. Most of the Component selection meta models use the Analytical Hieratical Process (AHP) for evaluation and ranking of candidate components. AHP shares some disadvantages during the ranking of candidate components. AHP follows a hierarchical approach for weight calculations and dependencies among different selection criteria are neglected or ignored. To resolve this problem an illustrated approach for CBSE is explained below.

**Proposed Framework:** Kumar et al. [23] proposed the W-shaped metaphor for multi-faceted test case classification and selection. Challenging issues in component based software development process and ref. [23] have motivated us to propose W-shaped framework for component selection and product development process. The COTS-based software development (CBSD) process has five main phases and research challenges are associated with these phases. These research questions are described as follows:

RQ1) How Domain analysis provides better benefits before requirement analysis?
RQ2) Whether to take single-criteria or multi-criteria decision making for efficient component selection.
RQ3) How to divide responsibilities among different stakeholders?
RQ4) How to select and evaluate candidate components?
RQ5) How to make a software component reusable?
RQ6) Problems arising during the integration of software components in a complete system.
RQ7) How to handle mismatches between non-functional requirements for components?

Hot issues of CBSD include domain analysis process, component selection process, repository updating process, component customization and adoption process, system testing and documentation process. We require five key points to draw the W-shape. Above questions motivated us to propose a W - shaped framework for component selection and product development process.

The W-shape of the development process emphasizes five main sub-processes of the complete development process. These five important sub-processes are i) Domain Analysis ii) Component Filtration iii) Component repository which gives the required component iv) Customization or Adaptation Process v) Documentation to ensure quality of the final product. The W-shaped software development model solves different research questions described above. Each point of the W-shaped model solves above problems in an efficient manner with the help of existing tools and techniques.
Domain Analysis: In the components selection and customization, domain analysis plays a vital role. The conceptual study of application domain of the system provides an outline of the final product. RQ1 and RQ2 are well justified in this phase for better understanding of service domain and required functionality of components. Domain analysis consists of well-structured and intense study for collection of application areas. Domain analysis help developers to find which software they can reuse or how to make software component reusable? During domain analysis, analysts try to characterize and understand the problem space and factor out similarity in problems to understand solution space.

Requirement Analysis: Requirement analysis is a crucial phase of software development. This phase decides different functionalities required by the software components. It also provides snapshots for necessary customization in selected components. It is an important phase to answer RQ3 and RQ4 as requirement elicitation creates a baseline for component selection based on multiple attributes of quality. Requirements are gathered from different stakeholders and on the other hand responsibilities are divided among stakeholders. It encompasses activities to determine the requirements of a new or altered system and conflicting requirements of various stakeholders. This phase includes extensive surveys and interviews with stakeholders to accumulate different requirements.

Component Selection and Filtration: The component selection plays an important role in CBSD and efficient component leads to a good quality software product at the end of CBSE. It is the most important phase of this framework as the filtration and selection is done from component repository. RQ5 is answered in this phase of the W-shaped model. The filtration and selection of the components are based on the keywords and different aspects which describe the functional and non-functional characteristics of the components. The functional and non-functional attributes set the criteria for component filtration and selection. Evaluation of candidate component is based on different quality attributes.

The quality of a component depends on functional and non-functional aspects of the component. Dependencies among different quality attributes should be developed after conducting a survey, the survey results are useful to establish relationships between quality attributes and to compute weight values of different attributes. Pairwise comparisons setup can be designed to conclude weight values using Analytical Network Process (ANP). ANP is a classical computing and decision making technique in which dependencies among different evaluation criteria are considered as it overcome some disadvantages of AHP. Component classification [3] is of three different types that is Enumerated classification, Faceted classification and Attribute-value classification.

Use of ANP: ANP is determined as a first mathematical theory which illustrates all kinds of dependence and feedback. Different quality attributes form a network in the form of cluster of sub attributes. Different quality attributes and sub attributes reserve some kind of direct or indirect dependency among themselves. This may generate a kind of network which is shown in Figure 2. The relative importance of different quality factors is depicted after a critical survey, interview and brainstorming sessions. Survey results lead to the development of pairwise comparison matrix to calculate the weight value of different factors.

Customization Process: Customization is a process of accommodating required attributes within the component as per requirement of the system. This phase of W-shaped model assist to answer RQ6, RQ7 and RQ8. Customization can be implemented in different level as per functionality or behavioral requirements. The component retrieved from the component repository in the initial phases of system development but sometime direct integration of component in the system is not possible and it requires some mechanism to make it adaptive as per operating environment. In principle adaptation of software
components is a process to modify them in accordance with changed conditions. Conditions mean the different operating environment or adaptation of software component behavior to assemble applications [16]. Adaptation means the change in the system to accommodate changes in its environment. An adaptation of a software system is causing the changes in the system, hence system will able to meet the requirements of new environment which is unable to fulfilled by old environment [16]. Adaptation is further subdivided into two different types, which is shown in Figure 3.

Input to this process is the selected component from the component repository. Information regarding any discrepancy in the selected component and required component is gathered. The approach is to minimize the mismatch among requested component requirements and selected component. For this purpose different strategies are proposed which are given as follows:

**Developer Handling Approach:** The developer will determine the mismatches between requirement specifications and selected component. Mostly this technique is used for White box components whose source code is available to developers. Development team makes necessary changes in the code structure of the component to remove any discrepancy. This process is time consuming and requires manual efforts to determine the quality and functionality of the component.

**Use of Automated Tools/ Patches:** Automated tools provide adapter services. A tool which generates adapter software component is given below [7]. The descriptive process is shown in the Figure 4. Different Integration problems may arise which should be solved before final integration of the components together to form a single software system. Problems that can arise during integration are mentioned in Table 1.

Integration of software components in system software leads to different problems, these problems are solved using glue code mechanism which acts as a patch to integrate component in system software. A common way of resolving these issues is to use wrappers, bridges or mediators as intermediaries [17].
Different characteristics of mentioned techniques are given as follows:

- **Wrapper characteristic**: it yields a new component in different ways in the same system, components are encapsulated behind a covering of abstraction and interface repair code is available.
- **Bridge characteristics**: it is an assumption based approach which requires external control. It has independent repair code and consists of the data format convertor.
- **Mediator characteristic**: Bridges are encapsulated into autonomous components.

**Component Repository**: Integration of component repository with W-shaped model fulfills an important role for faster component selection. Component repository helps in implementing primary filtration of components. Design of component repository is an important and critical issue which needs careful consideration. Proper and efficient component repository will help in the fast component retrieval and updating. It provides a GUI to store, delete and update component and their functional and non-functional information in the repository. The attributes of repository comprise of structure, function, primary data, interfaces, platform, operations and time. The following points specify above mentioned attributes in detail and also important with respect to quality certification of components. The important attributes of component log file are follows:

  Figure 5 shows the online component repository available to derive software component which is referred from [18].

  Figure 6 is referred from our previous work which shows the template of Component Entry form which can be used to save components in component repository and to update existing component in repository.
**Table 2: Attributes of Component Log**

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Includes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structure</strong></td>
<td>Source code, anything beyond software and hardware that is the part of product etc.</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>Includes arithmetic function, functions that modify or transform something, test suits etc.</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>Input or Output</td>
</tr>
<tr>
<td><strong>Interfaces</strong></td>
<td>An element that mediates the exchange of data with the users, stubs and drivers.</td>
</tr>
<tr>
<td><strong>Platform</strong></td>
<td>External software component and configuration that includes the OS</td>
</tr>
<tr>
<td><strong>Operations</strong></td>
<td>Patterns and sequences of input</td>
</tr>
</tbody>
</table>

**Fig. 5: Component Repository Template**

**Component Filter:** Component filtration performs according to requirement specifications. On the basis of requirement repository is searched. It is assumed that more the number of selected attributes, it becomes easy to refine the components from the repository. Sample repository template and selection template is given below:

![Component Entry Sample](image)
**System Assembly:** After the process of successful adaptation and customization, the next phase is System Assembly and System Testing. The component technology should provide support for integration of component on a continual basis. After the successful integration whole system is tested according to desired requirement. If there is any mismatch found it is reported to the Test Team to mitigate that problem.

**Documentation:** An important phase of software development for quality concerns. Buddy QE can be implemented to monitor the process of development as per DFD’s [19]. It is required to consider all phases of development under written scripts and it is necessary to monitor that the development work is done under given guidelines. Documentation process involves writing scripts of test results, training and troubleshooting steps to better understand the software product.

**CONCLUSION**

Component selection is a multifaceted NP-hard, search optimization and context driven problem. The proposed framework for component selection and adaptation is an efficient, flexible and interactive method for component elicitation and customization. The proposed framework speeds up the component selection and system assembly process and identifies components.
that closely match user requirements. The proposed method saves time of component selection and helps in customization, assembling the product and system testing. Keyword based retrieval plays a significant role in faster selection. It will minimize the risk of component mismatch. If a mismatch is identified direct interaction of developer support us in resolving issues related to component selection, adaptation and system assembling process. In future, soft computing techniques such as fuzzy logic, neural network, support vector machine, swarm intelligence approaches can be explored and validated for multifaceted component selection.

REFERENCES