International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 13 (2014), pp. 1245-1250
© International Research Publications House
http://www. irphouse.com

A Survey on Selection Techniques of Component Based Software

[1]Nitish Madaan and [2]Jagdeep Kaur

[1] Student, SE- M.TECH, ITM University, Gurgaon (Haryana)
[2] Department of CSE/IT, ITM University, Gurgaon, (Haryana)

Abstract

Component-based software engineering (CBSE) is an approach to develop software that relies on software reuse. The success of the final system is completely based on the component based software engineering that sometimes depends on the previous successful or failed case experience, previous decision and helps us to select the component that leads to the final system. It may reduce time for software development. In software development, Component-Based Software Development (CBSD) is a new phase that helps in development of complex software from the integration of pre-build components instead of developing everything from scratch. There are several problems (For example: Integration, Maintenance, Testing, etc.) occurred in the selection of components for integration. In this paper we discuss about the several component selection techniques that helps in the selection of the components.

Keywords: CBSE, CBSD, CLG, COTSRE.

1. Introduction

A component is the unit of a system that offers predefined service, and must be able to communicate with the other components. Component based software engineering (CBSE) mainly focuses on building large software systems with integrating previously-existing software components/modules [1].To improve software reusability and maintainability so many techniques have explored by software engineers for easing distress and pains that are given by complexity of software system, i.e. Modularization of software, object oriented techniques. Object-oriented technique is an approach to designing modular, reusable software systems. The goal of Component based software development is to reuse components based on previous conclusion and experience. But, the main goal of Component based software development is that it reduces the time for software development [2]. By using previous experience and the knowledge of failure can definitely reduce the overall development time.

Difficulties faced by the developers are not only to overcome the common properties are sufficient, but enable a software developer to make such confidence to select suitable components to be used.

The main problem we find is that how to select the component from the available list of components which satisfy the requirements of the system [3]. In this paper, we describe the component selection techniques that help to select the components which can satisfy the requirements.

2. Cluster Based Selection Process

The cluster based component selection process, as shown in Fig.1. It consists of 3 stages: that is dependency analysis, goal-oriented specification, and cluster analysis [4, 5].The First stage, goal-oriented specification, uses a goal-based requirements elicitation method to specify Component based Selection (CBS) requirements. The CBS requirements are having two levels of abstraction, that is, high-level goals and concrete-level goals, based on the refinement rules defined in. The second stage, dependency analysis, investigates the interaction between concrete-level goals. The third stage, cluster analysis, taking into concern the availability of suitable components and organizes mutually dependent concrete-level goals into clusters and creates consolidated goals. As a result, a selection of candidate components is identified for each consolidated goal. These Stages of the cluster based component selection are discussed in detail in next Sections.

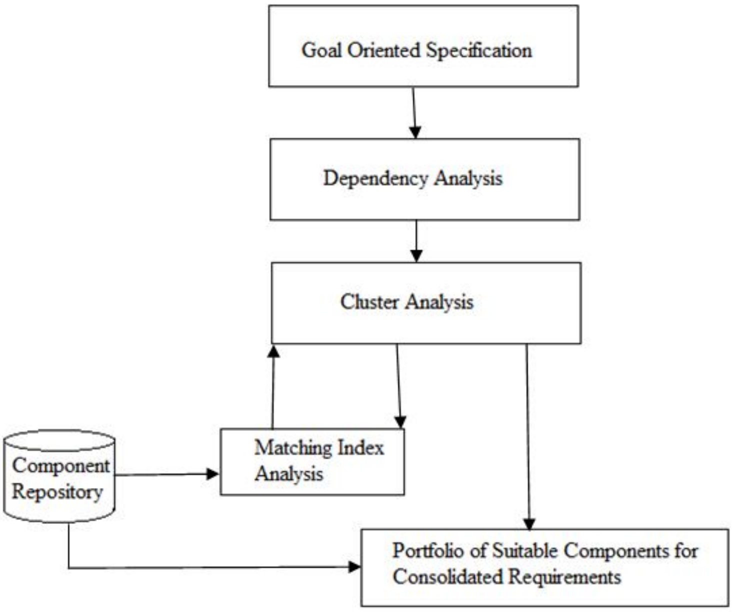
Goal Oriented Specification
Dependency Analysis
Cluster Analysis
Component Repository
Matching Index Analysis
Portfolio of Suitable Components for Consolidated Requirements

Fig. 1. Cluster Based Component Selection Process Overview.

2.1 Goal Oriented

Goal oriented specification in component based software development(CBSD) required a strategy that provides an opportunity for a software developer to shortlist the component that is either fully or moderately provide the preferred functionalities. In this CBS requirement flexibility is mainly essential for the process of component selection as it increases the possibility to find strongly matched components.

2.2 Dependency Analysis

However, the component selection process needs to consider the dependencies between Concrete Level Goals (CLG). We classify the relationships between these goals into three types of dependencies, i.e., usage, non-functional and threat, as shown in Fig. 2. A signed graph is used to model the dependencies as described in with every node representing a CLG and every edge representing a semantic dependency.

Each edge is assigned a positive or negative weight to show the nature and the strength of the corresponding dependency. In the following, we discuss the semantic dependencies and their graphical representation.

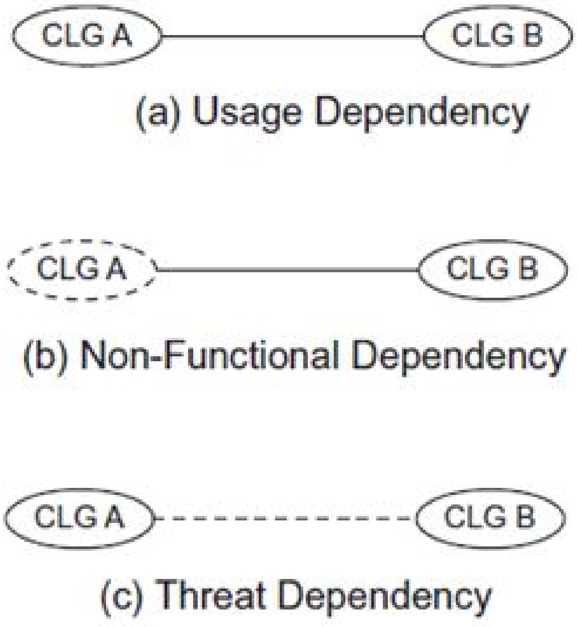

Fig. 2. Semantic Dependency Relationship

2.2.1 Usage Dependency

Usage dependencies states the usage relation between functional concrete level goals likewise, a concrete level goal require other concrete level goal for the correct implementation. Fig. 2 (a), Illustrate a usage dependency relationship in between two concrete level goals A and B. An edge is used in between two functional concrete level goals with an associated weight to represent graphically the usage dependency.

2.2.2 Non-functional Dependency

Non-functional dependency is a relationship between a nonfunctional Concrete level and functional concrete level goals. Representation shown in fig. 2(b) of a non-functional dependency as an edge between a non-functional concrete level goals and non-functional CLGs is described as a dotted node.

2.2.3 Threat Dependency

Similar to conventional software systems, a Component Based Software helps in terms of optimistic goals. This helps to neglect potential unwanted system risks (for example – security risk) that are undesirable for a Stakeholder. So, some general (common) risks need context precise solution any point in a system. Hence, it is essential to study the risks in between dissimilar concrete level goals. The representation of a treat dependency is shown by a dotted edge in fig.2(c).

2.3 Cluster Analysis

Cluster analysis is having two steps:-

1. Formation of Cluster
2. Selection of Component based on the index matched of each cluster.

2.3.1 Formation of Cluster

In this step, the goal is to cluster concrete level goals and non functional association.

When separating the concrete level goals with risk dependencies.

The formation of cluster helps in combining the concrete level goals within a cluster which works mutually to achieve the functionality of the component based system.

2.3.2 Selection of Component based on the index matched of each cluster

In this section we represent the measure to check the probability of the best clustering obtained in the above section. For each cluster we can classify a record of candidate components using progressive filtering technique. Each concrete level goal in the cluster is recognized using a keyword and it is use to search all the off the shelf components that can satisfy all the concrete level goals in a cluster.

To match index of cluster CL we define it as:

$$\text{M(CL)} = \frac{\text{Total Number Of Components that Satisfying all CLGs in CL}}{\text{Total Number Of Components that Satisfying at least one CLGs in CL}}$$

In this whenever a clustering of a non functional CLG with the related functional CLG is happened then it acts like as a standard for the selection of the component for the functional CLG. So, both the functional and non functional concrete level goals are necessary to satisfy a cluster. When the size of the cluster is increased then the matching index is decreases. Developer may face the problem because of the low matching index because there is not enough choice of candidate is available to select.

Research Gap

There are some gaps that are given below: -

1. In Goal Cluster Analysis, during cluster formation only two clusters are formed with positive edges on one side and negative edges on the other side. It can be improved by allowing both types of edges to appear anywhere in the graph.
2. In goal oriented specification and model it is assumed that a non functional CLG cannot be associated with more than one function CLG.
3. The threat dependencies in Dependency Analysis can be explored further & remedial measures can be thought for these.

3. Components Selection Method Based on Requirements Engineering (COTSRE)

To provide the software development team with design and components to search and verify the requirements that are matched as the requirements given in the software requirement specification document (SRS), there is an approach named "agile", in which the SRS is having the properties that shows the source from where one or more components can be found that are used in future in the software system that is being developed. Here some values are added in the properties that depend on the location of the components (URL, Component Repository, etc.). When the requirements are written in srs document, it may contain some references to components that are used in future [6]. If there is no reference written in the srs document then it means that developer has one and only option to start the project from the scratch. In this, we are having an initial list of the links to the components which does not means that these all the components are the valid components for the current project, we just confirm that by using these components we can achieve the functionality. In the selection of component we must keep in mind that the rest of the linked requirements, especially those non functional requirements that force some type of constraint in the final system (operating system, memory, programming language, etc...). Here for each of the requirement we make a Selection matrix, which contains a column that is used to identify each of the requirements in it, and a row with each of the identified components.

Table 1: EXAMPLE OF SELECTION MATRIX

Component	Requirement A	Requirement B	Requirement C	Requirement D
Catalyst Internet Mail v4.4	X		X	X
Rebex Mail .NET v1.0.2537.1	X	X	X	
Easy Mail .Net v3.2		X		X
Active Mail Professional v1.4	X	X		

The criteria followed to decide which of the requirements are used to consider and which are not it's all depends on the developer [7]. In this selection method we also take into account that other requirements in the final Product Requirements Document contains not only the technical requirements but can also impose more limitations (in cost, available licenses, development team qualities) until even the selection procedure finishes in an empty list of components.

Research Gap

In selection matrix, we can associate weights with the requirements by considering the component selection process as a multi objective optimization problem. The objectives can be reduced cost and time & the non-functional requirements (performance, security) can act as the constraints.

4. Conclusion

This paper presents the role of Requirement analysis in component selection process that helps the software developers to model their requirements. In the First technique we identified some gaps which can be removed by application of Fuzzy clustering techniques or signed graph approach from the social network analysis. In the second technique, Multi Criteria Decision Making algorithms can be applied for calculating the weights to be associated with the requirements. There is a need to strongly integrate requirement analysis phase with the component selection stage. In future we plan to validate the existing techniques with the large datasets as well as improve the efficiency of the current techniques by using Multi Criteria Decision Making algorithms (Fuzzy Analytic Hierarchy Process, Goal Programming).

References

[1] Wes J. Lloyd *"A Common Criteria Based Approach for COTS Component Selection"*, in *Journal of Object Technology*, April 2005.

[2] Syed Ahsan Fahmi, Ho-Jin Choi *"A Study on Software Component Selection Methods"* Feb. 15-18, 2009 ICACT 2009.

[3] Arvinder Kaur, Kulvinder Singh Mann, *"Component Selection for Component based Software Engineering, International Journal of Computer Applications" (0975 – 8887) Volume 2 – No.1, May 2010.*

[4] Muhammad Ali Khan , Sajjad Mahmood *"A graph based requirements clustering approach for component selection"*, 2012.

[5] Ed Mancebo, Anneliese Andrews, *"Strategy for Selecting Multiple Components"*, *SAC'05* March 1317, 2005, Santa Fe, New Mexico, USA

[6] Miguel A. Martinez and Ambrosio Toval. *"COTSRE: a Components Selection method based on Requirements Engineering"*, 2008

[7] Jeetendra Pande,*"On Some Critical Issues in Component Selection in Component based Software Development", International Journal of Computer Applications (0975 – 8887) Volume 46– No.4, May 2012*