Review for Different Approaches for identify a Software Component

Mr. Anil k. pandey[1], Dr. Tulika[2],
Department of Computer Science & Engineering.,
SHAUTS ALLAHABAD

Abstract

Component based software engineering offers inherent benefits in software quality, developer productivity and overall system cost. There are different ways to classify the components like physical and logical type. This paper introduces existing different method for components classification. We consider the logical components aspects of a component. To identify the component is a very difficult task. There are many evaluation and non-evaluation approaches are available for evaluate the components.

Keyword: Component, CASE, Software Architecture, cluster.

Introduction

Component-based software engineering (CBSE) offers inherent profit in software quality, developer productivity, and overall software cost [1]. And yet, many problems remain to be overcome before the component based is widely used throughout the industry. In addition to software components, a variety of reusable artifacts can be acquired by a software engineer. These include technical representations of the software, documents, test data, and even process related tasks. The CBSE process encompasses two concurrent sub processes—domain engineering and component-based development. The intent of domain engineering is to identify, construct, catalog, and disseminate a set of software components in a particular application domain.

Component is a reuse based approach to defining, implementing and composing loosely coupled independent components into systems. Components play this role, for example, in web services, and more recently, in service oriented architectures (SOA), whereby a component is converted by the web service into a service and subsequently inherits further characteristics beyond that of an ordinary component as shown in figure 1.The software components divided into three categories. The Business-oriented components, Architecture-focused or Logical components and Technical components. Business-oriented components are associated with business components and require realizing business processes. Architecture-focused components concentrate on logical characteristics, i.e., a structuring is required in them. Finally, [3], Technical components focus on deployment and implementation aspects.

logical component, in contrast to a physical component, is a component representing requirements except for technology, environments, and constraints. it is meaningful that logical components provide the abstract form for designing the software architecture. Logical components are the primary abstractions of the entire design of a system. Dividing a system into logical components has a key role in defining the system architecture. In case of component based software design process, an architect is work for decomposing a system into different logical components. However, because lack of software tool support for identification of component and experienced software architect.

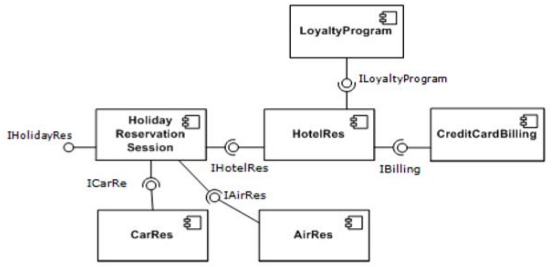

Figure 1: Example of Software component

Component based approach similar to conventional or object-oriented concept. An architectural design is established, but rather than moving immediately into more detailed design tasks, the members examines requirements to determine what subset is directly available to composition, rather than construction. That is, the members ask the components available to implement the requirement or internally developed reusable components available to implement the requirement. It is involving different parameter with unspecified values, with or without limited conditions.

Component adaptation is that software architecture represents design patterns that are composed of components, connections, and coordination. In essence the architecture defines the design rules for all components, identifying modes of connection and coordination. These components must be adapted to meet the needs of the architecture. Architectural style again plays a key role in the way in which software components are integrated to form a working system.

Software component is a dynamic package of one or more programs managed as a unit and accessed through documented interfaces that can be discovered in run time or Software component is a unit of composition with contractually specified and explicit context dependencies only. Assessed by software engineers to ensure that not only functionality, but performance, reliability, usability, and other quality factors conform to the requirements of the system or product to be built is qualified component and Adapted components to modify unwanted or undesirable characteristics.

Object oriented based architecture is used to understand the software system. Object is combination of attribute and properties and it create components. Component is a composition of individually verifiable and reusable software unit. System means collection of components and related to each other's component.

This paper concentrates on different approaches available for identifying components in given scenario. Logical definition of component as follows A

2. Component identification problem

Component identification problem is a real environmental based involving different parameter with unspecified values, with or without limited conditions. Finding and managing a component are difficult type problem because of component is a run type entity.Component identification problem is belonging to non-polynomial type problem. By the definition: A problem 'belongs to the P (Polynomial) it there exists an algorithm which can solve in polynomial time, that is, the time needed to produce an answer is a polynomial in the size of the problem. There are many problems which cannot be solved by polynomial time method. The only approach for some of these problems seems to be an exhaustive search of all probabilities. A probability is guessed and then verified, that is, it is checked to see if it is a valid solution. if this verification can be done in polynomial time, the problem belongs to the NP (Nondeterministic Polynomial). and find a component is a NP type problem. To solve components based problem is byusing both evaluation and non-evaluation approach with object oriented unified modelling techniques.

Criteria for well defined component

To Find a criteria for defined component is another issues. There are different criteria which depend on numbers factors. In this papers we emphasis that component granularity is most important factor for design. if the component is large, the structure of the system is complex. Therefore level of abstraction, complexity, coupling, cohesion and maintainable are the very important criteria. In this study we focus on coupling and cohesion.

3. Different Approaches for identifying Logical Software Component

There is different Approaches used for identifying the software components. These techniques are divided into a two categories name as evolutionary and non evolutionary.

Defined and example.

like feature-based clustering method,k-means, Hierarchal, Graph-based method, and Fuzzy C-means different methods, Graph Partitioning, Clustering Based, CRUD based and FCA and Evolutionary. Note that this approach is used to identify Technical components, i.e., software modules, from source codes. ere used for analysis to achieve good component.

In this paper we discuss both techniques one by one as below.

3.1 Component identification method with coupling & cohesion

In the paper Jong Kook lee et.al [8] had explain a technique for identified software component. The block diagram of process to identified software component is as shown in figure no2.

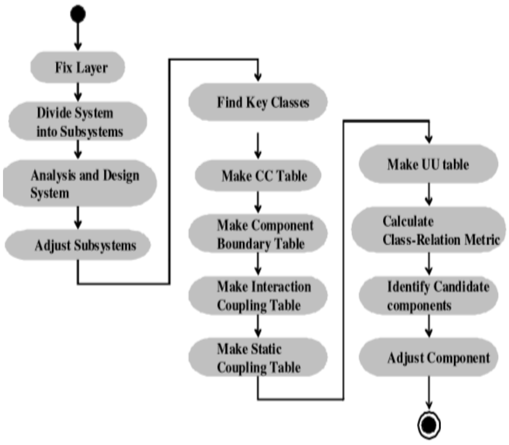
Fix Layer
Divide System into Subsystems
Analysis and Design System
Adjust Subsystems
Find Key Classes
Make CC Table
Make Component Boundary Table
Make Interaction Coupling Table
Make Static Coupling Table
Make UU table
Calculate Class-Relation Metric
Identify Candidate components
Adjust Component

Figure 2: Process for identify Component

In this method first we define a fix layer and then divided system into subsystem

Explain same

3.2 Graph Partitioning Approach.

Albani et al.[]have related to object models into vertices and edges of a graph. then, based on relation types between object model elements and programmer priority, they have assigned weights to edges.the graph is partitioned into node called components by applying the a heuristic from graph concept. Graph G =G(N,E) ,where N is a node and E is edge of graph G represent the relation of classes, classes whose relation is very strong are represent as cluster in G. G is the candidate of component and transformed the relationship model among business elements to a weighted graph. Then, they have applied a graph segmentation method is applied on the graph to identify mutually disjoint sub-graphs as components. This method has achieved cohesive components with low coupling. The disadvantage is that of this approach is that weights are manually assigned to edges according to domain expert.

3.3 Clustering-Based Approach.

Lee et al. [3] had used clustering approach to identify a software component. In this approach classes are grouped on the basis of complexity. Initially classes are selected as candidate components. then, other classes are assigned to the components that have the highest level of dependency. First step is to classes which is also a difficult task. in this paper manually determined classes by experience person. It is used hierarchical agglomerative clustering method to iteratively cluster two classes with the highest complexity.

3.4 CRUD-Based Approach.

Lee et al.[]purposed a tool called COMO for identified the component, in which “use case/class matrix” is created with respect to use case diagrams and class diagrams. It is then partitioned into blocks with tight cohesion as business components. Ganesan and Sengupta [GS01] presented a tool similar to COMO called O2BC, but it has several differences in the clustering technique and uses business events and domain objects as input. However, this method has a number of limitations similar to Clustering due to the use of clustering method.

3.5Framework Based Approach.

A framework based on the theory of FCA to partition a class diagram into logical components with several heuristics similar to clustering method. However, framework defines stability instead of cohesion and coupling as important metrics to identify components. It transformed business elements and their memberships into a lattice and simple clustering method and then identify components. They used dispersion and distance concepts to measure the cohesion and coupling,However, they used two dispersion and distance thresholds with high effect on the performance of their method, which must be manually determined by expert. This method has a number of disadvantages similar to Clustering Based approach, due to the use of classical clustering method.

3.6 Evolutionary Approach.

Evolutionary Method have been widely applied to solve the component problems. Therefore, a new field of software engineering called search based Software Engineering (SBSE) to solve software problems as optimization problems. One of the popular scopes of search-based design is module partition. In this area, source code of a software system is group into software modules with a high degree of cohesion and a low degree of coupling. Component idenfication is search-based technique, objective is to identifying components with high degree of cohesion and a low degree of coupling. In this method, the inputs of our use case and class diagrams as opposed to the inputs of search-based clustering. To find a new suitable structured group based on search-based clustering technique, there are many algorithms available like hill-climbing, simulated annealing and Meta-heuristics like genetic algorithm[2].

By using Genetic algorithm, both cohesion and coupling metrics are combined into a single objective fitness function means multi-objective approach. in exiting method has several shortcomings.

First, it yields a set of solutions, among which software architects have to select one. For example, an iterative multi-objective genetic algorithm is given to identify design classes. In this algorithm, software architects must rank a number of identified solutions in each generation. But in others a semiautomatic technique used and its performance depends on experience person.

Another difference by using evaluation approach, higher computational costs and is time-consuming. In fact, when the number of objectives is increased, other approaches are not suitable, because it needs more population members and more computations. difference between SCI-GA and all the existing search-based module clustering methods is that they aim at optimizing clustering criteria like intra-edges and inter-edges of all clusters but in Genetic algorithm objective is to maximizing cohesion and minimizing coupling and reduce complexity. K means

3.7 K Means

Component identification problem is solved by using k-means technique, which are inefficient to deal with complex search landscapes due to their simple greedy and heuristic nature. These methods are effects to optimize measurements like component cohesion, coupling, and complexity.

The K-Means approach is one of main algorithms in which Pattern recognition and Machine Learning involve. Because of random selection of group centers and the adherence of results to initial group centers, the risk of trapping into local optimality ever exists. Inspired by a genetic algorithm which is based on the K-means method, a new approach is developed, and in which group centers are selected and computed appropriately. Suggested approach by using standard data sets and comparing it with alternative methods in the literature reveals out that algorithm outperforms the K-means algorithm and other candidate algorithms in the pool.

Modified K means

The K-means algorithm is a major clustering method. Despite its simplicity it is a basic method for many other clustering. In its simple version,first some points as many as required clusters are selected randomly. Then based on the ongoing similarities and trends each data item is allocated to one relevant cluster which in turn creates new clusters as well. Accordingly, by repeating the same procedure and by averaging data items new cluster centers are calculated and then data items are reassigned to new clusters. This procedure continues as long as there is no observable change among data items. Although in this algorithm convergence is guaranteed but due to its dependence on the initial cluster centers the ultimate solution is neither unique nor necessarily optimum. Many different approaches have been suggested to overcome this shortcoming K-means algorithm such as Particle Swarm Optimization (PSO), The Colony of Ants ,Taboo Search,[5] The Metal Annealing method and some other combined solutions[6].

3.8 Quality-Centric Approach for Software Component Identification from Object-Oriented Code

Components and connectors are the main building blocks for software architectures. In the design phase of a software system, components can either be created from scratch or reused. When reused, they can either exist on component shelves or identified from existing software systems. Thus, software component identification is one of the primary challenges in component based software engineering. Typically, the identification is done by analyzing existing software artifacts. When considering object-oriented systems, many approaches have been deal with this issue by identifying a component as a high cohesive and loose coupled set of classes. However, this assumption leads to two main limitations: in one hand, the focus on simple metrics like high cohesion and loose coupling will not necessarily lead to the identification of good quality components. On the other hand, the identified components external structure (provided and required interfaces) is missing. As a result, the identified components can hardly be reused, composed, packaged and documented. To overcome these limitations, we propose in this paper an approach for identifying components based on a fitness function to measure the quality of a component.Also, we propose to identify provided and required interfaces of components.

In addition, in most of the existing software component identification approaches, the result is a set of components with high intra-component cohesion and low inter component coupling. Components identified using these approaches are not necessarily of high quality due to the lack of a semantic-correctness model to identify the best components of the system. Another shortcoming of the existing approaches is that they identify only the classes belonging to a component without specifying its external structure (i.e. provided and required interfaces). Thus, the identified components cannot be directly used, composed or packaged for future reuse. In this paper, we propose a bottom-up approach for soft- ware component identification from existing object- oriented source code that avoids the previous limitations. In our previous works , we have defined a semantic-correctness model for extracting software architectures from object-oriented source code. We rely on these results to pro- pose a software component identification technique. Our ap- proach is based on a mapping model between object-oriented and component-based concepts and a fitness function to measure the semantic-correctness of a component. Beyond identifying the internal structure of software components,

4 Conclusion and Future work

The reusability of CBSE is a common approach to improve quality, reliability and productivity as well as reduced time and cost for development. The impact of time and cost is still a challenging issue for search based software engineering in the domain-specific components. To identifying optimal component is among the main advantages of CBSE.In this research, we have reviewed different approaches for identifying the software components based methodologies. This study also compares different evaluation and non-evaluation based approaches the future requirement is to look into how components are optimized by using machine learning and integrated to encourage reusability and reduce cohesion.

5. References

1. R. J. G. B. Campello, E. R. Hruschka, "A Fuzzy Extension of the Silhouette Width Criterion for Cluster Analysis" Fuzzy Sets and Systems, vol. 157, n. 21, pp. 2858-2875, 2006
2. J.F. JIMENEZa, F.J. CUEVASb, J.M. CARPIO "Genetic Algorithms applied to Clustering Problem and Data Mining " Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization, Beijing, China, September 15-17, 2007
3. N Md Jubair Basha1 and Dr Chandra Mohan2 "A strategy to identify component using clustering approach for component reusability"Natarajan Meghanathan, et al. (Eds): SIPM, FCST, ITCA, WSE, ACSIT, CS & IT 06, pp. 397–406, 2012. © CS & IT-CSCP 2012 ,

4. Alex A. , "Computing Laboratory, A Review of Evolutionary Algorithms for Data Mining" Freitas University of Kent, UK
5. Mohammad Babrdel Bonab "Modified K--Means Algorithm for Genetic Clustering "IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.9, September 2011
6. Mark Harman and Yuanyuan Zhang" Search Based Software Engineering: Trends, Techniques and Applications" ACM, Inc., 2 Penn Plaza, Suite 701, 2007 New York,
7. O.A. Mohamed Jafar and R. Sivakumar "Ant-based Clustering Algorithms: A Brief Survey "International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010 1793-8201
8. Gholam Reza Shahmohammadia Saeed Jalilia Seyed Mohammad Hossein Hasheminejada "Identification of System Software Components Using Clustering Approach" JOURNAL OF OBJECT TECHNOLOGY Published by ETH Zurich, Chair of Software Engineering © Jot, 2010
9. Wiem Mkaouer, Marouane Kessentini" High Dimensional Search-based Software Engineering: Finding Tradeoffs among 15 Objectives for Automating Software Refactoring using NSGA-III "Conference'10, Month 1–2, 2010, City, State, Country. Copyright 2010 ACM 1-58113-000-0/00/0010
10. Mahdie khadame, Sima Emadi "Measurement of Software Reliability Based on Coupling and Cohesion Rate by External and Internal View of Classes "IJCSNS International Journal of Computer Science and Network Security, VOL.16 No.11, November 2016
11. L E Agustin Blas " A New Groping Genetic Algorithm for Clustering problem " Expert Ststem with application 38(2012) 9685-5703
12. Eleni Constantinou "Extracting resuable components : A Semi Automated Approach for complex structure " information processing letter 115 (2015) 414- 417