

# Controlling the Design of Reusable Components Using Fuzzy-Logic Approach

R. Kamalraj\*  
Dr. A. Rajiv Kannan\*\*  
P. Ranjani\*\*\*

## Abstract

To reduce the efforts on building software systems have different methods. Among them 'Fuzzy Logic' with 'if-then' rules can be applied on the various conditions required for defining the characteristic of a component in the systems as 'Reusable-one' or not. From the existing methods it can give a 'Rigid Condition Based' system to lead the development team to shape a requested component to reach the reusable metrics level. 'Fuzzy-Logic' with 'if-then' rules should not sacrifice any condition to give liberty to development team to build a 'reusable component' after some time. So it is fully controlled activity on software personnel involved in the development activities. This approach can give reduction of time and resources after selecting the 'Reusable Component' those are designed using Fuzzy-Logic if-then rules. It may give a great improvement of 'Software Quality' also.

**Keywords:** Software Engineering, Reusable Components, Software Quality, Fuzzy-Logic

## 1. Introduction

Software Engineering is a collection of systematic tasks to develop software systems in an effective manner. In Software Process Model different phases like 'Requirement Analysis', 'Software Design', 'Developing Source', 'Testing' and 'System Maintenance' are doing their task to give outcome as a 'Quality Product'. Before begin development phase 'Effort Estimation' should be done on available resources to develop the software systems in a short period. But, to gain from the development of a product it should be built in a short period with attractive features. To construct the software systems in a very low development cost the reusable term can be applied. Reuse the existing 'Design Documents', 'Tools' and 'Technology' can give full support for constructing systems in a rapid manner with less resources and less time [Jim-Min Lin (1997)]. To design such reusable components while developing a system is a difficult one. To reduce the difficulty on designing those reusable components 'Fuzzy Logic' can be applied. The method of constructing reusable patterns using Fuzzy logic is discussed in following chapters.

### 1.1 Model for Software Reuse

To develop a system, 'Requirement Analysis' should be done on customer requirements to find any loop hole like similar

\*SNS College of Technology, Coimbatore, Tamilnadu, India

\*\*KSR College of Engineering, Tiruchengode, Tamilnadu, India

\*\*\*Avinashilingam University For Women, Coimbatore, India

component code or designs available in knowledge management systems [Caldiera, G (1991)]. There is prescribed steps should be followed during 'Requirement Analysis' other than 'Requirements Collection and Validation'. They are listed and discussed below.

**1.1.1 Problem Domain Analysis:** In this step, the given problem domain will be compared with existing solved problem domains [Basili V.R (1994)]. So, by matching the model of proposed system with available one can provide constructed components already to construct the new one. So finding similar design document may reduce the time on 'Designing new system' using those same components. In step by step and in coming process model phases the time and effort can be reduced.

**1.1.2 Components Analysis:** During this step, different procedures will be used to identify the fitness of procured reusable components and elements. Because suppose mistakenly identified elements not helped in prompt manner that may collapse the remaining system construction phase tasks. So thorough analysis will be done on components' metrics to find the suitable and supportable one.

**1.1.3 System Analysis:** This step focusing on the impact of reuse available items in the product development. Reusing already constructed items may or may not fulfill the quality factors given in requirement. So different features of the reuse components will be verified that whether there is any quality violation in the system development.

## 1.2 Fuzzy Logic

Fuzzy Logic is a 'Conditional Based Concept' to do certain tasks as per valuation of conditions in any system [S. V. Kartalopoulos (1996)]. It is based on '0' or '1' to represent 'false' or 'true' respectively. By using True-False methods, checking the constraints easily to start or reach to a state. In Fuzzy Logic System three different activities should be performed to derive an output.

Those are

- a) Selection for Input
- b) Process and Find the branch for appropriate conditions
- c) Provide Output

**1.2.1 Selection for Input or Fuzzy Perception:** Many data may be collected while analyzing a task, but from the collected data suitable data should be selected for giving input to get output according to that [M. Halkidi (2011)]. Input may be value of a field or condition on a particular field to process them.

**1.2.2 Process and Find the branch:** As per given input, the value or condition have to be processed to know in which path the control has to transferred for activating other conditions or deriving values for other fields in different places.

**1.2.3 Provide Output:** In this step, the output should be derived from the given input value and different values reached in different states. It indicates that the job system will be over in this step.

## 1.3 Fuzzy Logic in Software Engineering

Software Engineering is a systematic approach with different tasks or activities to develop a system. It consists different management activities to manage the abrupt situations reached in the product development. Putting Fuzzy Logic on software engineering activities that can give the outcome as a simplified development structure control flow to track the development tasks and situations easily. By applying the Fuzzy logic in SE that will give a connected neurons (states) in software development to build a software system [Parvinder S.(2006)].

In developing a system with specified needs they should be valued in each stage in the product development phases. So in each stage there is a constraint to be checked to get into a next phase or another phase or enter into the previous stage. Different states of the system can be identified already in the system design phase stage. But the activation of a particular task is fully based on the satisfaction of a condition. Results should not be adjusted for doing next tasks. So it is a 'Strict Task Force' to develop systems when Fuzzy Logic is applied.

## 2. Metrics of Reusable Modules

'Reusability' characteristic of a module or Component is fully depend on the design metrics of the module in the system. In Design Phase the 'Software Designer' is doing the job for design the 'High-Level' and 'Low-Level Design' of the proposed system to satisfy customer needs. Designer will analyze the connections or communications among modules to derive a solution. The following metrics should be analyzed to decide whether the designed elements can be reused in future.

### 2.1 Coupling of Modules

Coupling means that kind of relation such as 'weak' or 'strong' connection between different modules in the proposed design. To find that strength of connectivity the 'High-Level' Design patterns will be scanned. Based on the types of connection the coupling in various forms. They are 'Loosely Coupled', 'Highly Coupled', 'Data Coupling', 'Functional Coupling', 'Import Coupling' and 'Export Coupling'. In 'Loosely Coupled', a component may be independent from other components in the system. It indicates that that corresponding component can work with itself to reach the system tasks. But in 'High Coupled' the life-time of a component is fully dependent on other modules present in the problem domain. Then, if two components are shared a message then they are in data coupling. If they are communicated through passing controls among them is known as 'Functional Coupling' and it one component takes the service of other components then it lies under 'Import coupling' and components gives data to other components is in group of 'Export Coupling'. By measuring this and value this can give the idea of copying a component and place it another system design to finish the system development as much as possible. productivity

### 2.2 Cohesion of Component's Elements

'Cohesion' states that the logical relationship among the elements in a component to do their service to fulfill their component's services. A single element may have a dependency other elements in the same module represents that it has a 'Internal Dependency' to complete its duty. In another direction, an element may be based on the function of an element in another

module. So it is in 'External Dependency' to complete its functions. For getting communication with external module's element it will use its component interface.

**2.3 Degree of Complexity**

This design metric is checking the cross communication among the modules and occurrence of them. A component having a very high degree of coupling and cohesion and it types then it may be difficult to separate the channels of them. High degree of coupling may lead to develop the system in a very complex manner. So while designing a system the 'Partitioning' approach may be applied in a sufficient level to reduce the system complexity.

**2.4 Depth of Inheritance**

Inheritance concept increases the level of reusable elements to build different systems in a rapid manner. In this concept the code segment or element of a component or the entire component may be hidden by new one identified in the proposed problem domain. The parent-child relationship should not be so depth to access their attributes. The nominal range of inheritance depth is 0-4.

**2.5 Range of Stability Factor of a Component**

Stability Factor of component can be measured using the two types of Coupling such as 'Import' and 'Export Coupling' and its range is '0 to 1'. A component can act as 'Server' component when it has one and only 'Export Coupling'. When this type of component is selected for 'Reuse' it would not need any additional effort. In another case, a component has only 'Import' coupling it cannot be preferred for reuse in another system. Because they need so much of effort to remove its dependency or acquiring its related modules from the total system. A component having 'Import' and 'Export' Coupling may act as an 'Dealer' package in the system. If this type of component identified for further use then they may need some amount of effort to remove the external dependency or it may increase the system memory space when its connected modules also selected for reuse. Only stable elements or component can give high degree of performance and better utilization.

**3. Fuzzy Logic on Designing Reusable Component**

Designing of a reusable component can be found within the 'Design' phase itself when checking or calculating the above metrics. To announce a component as a 'Reusable' one may need to complete different steps or stages. Those steps can be depicted as 'if-then' rules of Fuzzy Logic.

The different constraints on designing a 'Reusable Component' are:

- \*Requirement may be needed many platform
- \*Amount of Function points satisfied by the component
- \*Type of Coupling (Import or Export Coupling)
- \*Degree of Complexity (Low or High)
- \*Reliability (Stability of Component)
- \*User-Interface design quality (if any)

The shaping procedure of a reusable component can be initiated when needed customer requirement is a standard with other problem domains.

**3.1 Fuzzy Logic Design for Reusable Component**

To design a reusable component the fuzzy logic design can be considered as 'Tracking System' to lead that selected component into 'Reusable Component' level. For that, 'if-then' rules with '0' or '1' can be applied for structuring the process flow in shaping the 'Reusable Component'.

```

Start:
  IF (Requirements == Standard )
  THEN
    IF (No. of Functions > 1)
    THEN
      IF (COUPLING == EXPORT)
      THEN
        IF (COMPLEXITY == LOW)
        THEN
          IF (UI == BEST)
          THEN
            REUSABLE LEVEL REACHED
    
```

The above structure show that only basic flow or 'true' flow in all levels. Suppose there is any condition is false then it may go in another direction.

- a) IF (Requirements == Standard)
  - THEN
  - .....
  - ELSE
    - Stop the shaping procedure and go to previous step
- b) IF (No. of Functions > 1)
  - THEN
  - .....
  - ELSE
    - Stop the shaping procedure and go to previous step
- c) IF (COUPLING == EXPORT)
  - THEN
  - .....
  - ELSE
    - Stop the Procedure and avoid that component
- d) IF (COMPLEXITY == LOW)
  - THEN
  - .....
  - ELSE
    - Stop the procedure
- e) IF (UI == BEST)
  - THEN
  - .....
  - ELSE
    - To spend effort to give Good Quality of GUI

The above 'if-then' rules on reusable component construction can represent like in following control flow pattern. In the following figure the basic flow alternate flow are represented as nodes and their outcomes to reach a destination. In the below figure 'C' means that different conditions given in the basic flow and numeric values indicate the state or step in development phase.

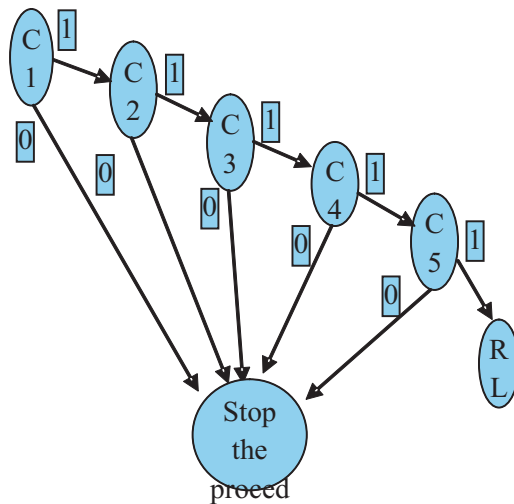


Figure 1. Fuzzy-Logic Based Reusable Component Design

From the above figure, if first condition i.e the requirement is a standard one then only the remaining conditions may be valid to verify. Else they would not participate to give the output. After the first stage, the remaining stages may give 'false' result as per the condition evaluated in the appropriate stage and it may lead to stop the shaping of the component. But in last stage, suppose 'GUI' is in poor quality then that component UI may be recommended to change the appearance of that. Or having different GUI standards for the same component may be useful to reach the product quality into specified level.

#### 4. Results and Discussions

Applying the concept Fuzzy Logic on designing a reusable package can give a great confidence for selecting the components in the design phase using the same estimated effort. Estimated Effort would not be disturbed at any cost due to verifying conditions can take a little amount of time or that would not consume any time [Nancy Bazilchuk (2005)]. It is a 'Rigid Conditions Based' on checking the metrics level of the components and their corresponding values. It would not be adjusted the present situations in the components status. So tracking of a component to reach 'Reusable Level' is a simple one and anyone involved in the software development can do this activity and record the result in the 'Knowledge Management System' with appropriate data.

#### 5. Conclusion

Designing reusable elements using 'Fuzzy Logic' is a strict condition based to give a best reusable component. Through applying 'Fuzzy Logic' concept, the data from various personnel will be collected and weighted for finding the 'Reusability' characteristic of a component in the proposed design. From the construction phase, designer can recommend some components have to be implemented carefully to achieve the goal of 'Reusable Component' with best quality. This proposed methodology can reduce the problems in selecting 'Reusable Component' from KMS. Also it reduces required effort and time and resources for building further projects. By selecting good quality reusable elements may improve the productivity of the development team [Nunamaker J.F (1989)]. And it is a another way of classifying the software components as 'Reusable' while designing the system. The proposed methodology is a common to all types of programming languages.

#### 6. References

1. Nunamaker J.F.; Chen M (1989) "Software productivity: a framework of study and An approach to reusable components", *System Sciences, Software Track, Proceedings of the 22nd Annual Hawaii International Conference on* vol. 2, 3-6.
2. Nancy Bazilchuk; Parastoo Mohagheghi, (2005) "The Advantages of Reused Software Components". *R&D and Technology Transfer*.
3. Caldiera, G.; Basili, V.R, (1991) "Identifying and qualifying reusable software components", *IEEE Computer Society*.
4. Parvinder S. Sandhu; Hardeep Singh, (2006) "Automatic Reusability Appraisal of Software Components using Neuro-fuzzy Approach", *International Journal of Information Technology* Vol 3, No. 3.
5. S. V. Kartalopoulos, (1996) "Understanding Neural Networks and Fuzzy Logic-Basic Concepts and Applications", *IEEE Press*, pp. 153-160.
6. Basili V.R; Bri L.C. and Thomas W.M (1994), "Domain Analysis for the Reuse Of Software Development Experiences", *Experimental Software Engineering Group (ESEG), Vol. 11*, pp. 86-95.
7. Jim-Min Lin (1997), "Cross-Platform Software Reuse by Functional Integration Approach", *21st International Computer Software and Applications Conference*, Vol. 1, pp.402.
8. Halkidi; D. Spinellis; G. Tsatsaronis; M. Vazirgiannis, (2011) "Data mining in software engineering", *Intelligent Data Analysis*.