American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

Requirement Classification using Association Analysis

Swarnalatha K S[1], G N Srinivasan[2], Swetha Murali[3], Rakesh R[4], Vipin Dwivedi[5], Kishore PV [6]

[1,2,3,4,5,6]RV College of Engineering, Mysore Road, R V Vidyanikethan Post,

Bangalore, Karnataka 560059, INDIA.

Abstract*: The proposed method illustrate about how the different kinds of requirements can be classified in to pre-defined domains, according to the bee-hive model[1] the Requirement engineering process consists of background research, the areas of background research considered here are Application domain, market trend ,scope of evolution, scale/product analysis, organizational factors and safety and security this research analyzes and proposes an optimum requirements segregation process flow using proposed algorithm, the algorithm effectively classify the requirements in to different 6 domains, the classified requirements can be easily analyzed and process it for further process.*

Keywords*: Software Engineering; Requirements Elicitation and Analysis; Requirements Specification; Requirements Validation; Requirements Engineering Process Model, Association.*

I. INTRODUCTION

The success of any software project is determined by its ability to meet the goal for which it was intended. Requirements Engineering is a phase which enables the developers to identify the needs of the stakeholder and document them so that they can be analyzed and implemented efficiently. Well defined and effective requirements are really important as they form the basis of the framework of a system. The quality of the software product being developed is directly related to the quality of its requirements. The main task for software engineers is to recognize the important requirements from the unimportant ones and also to classify the multifarious requirements into distinct domains. There is a need to identify the relationships among numerous requirements as it gives a clear understanding about the requirements and also provides sufficient knowledge in prioritizing the requirements. In this paper we use data mining concepts like association analysis to obtain the important requirements and find relationships among various requirements which would aid in their classification. Classification of the requirements and prioritizing them helps the software engineers in developing products which efficiently reflects the stakeholders' needs. We classify the requirements by applying association analysis techniques on the keywords identified from the given set of requirements. Support and confidence thresholds are used to eliminate unimportant requirements and uninteresting relationships among the requirement keywords thereby minimizing the cost and time of development of the product. Documentation of the obtained classified requirements is of utmost importance as they play a critical role in determining the needs and behavior of complicated systems.

II. REQUIREMENTS ENGINEERING

Requirement engineering is the most effective phase of software development process. It aims to collect good requirements from stakeholders in the right way. It is important for every organization to develop quality software products that can satisfy user's needs. It is a complex exercise that considers product demands from a vast number of viewpoints, roles, responsibilities, and objectives. Therefore, it becomes necessary to apply requirement engineering practices in every phase of software development process.

Requirements engineering is a systematic process of developing requirements through an iterative cooperative process of analyzing the problem, documenting the resulting observations in a variety of representational formats, and checking the accuracy of the understanding gained. Requirement engineering calls for the involvement of requirements engineers, customers who commission the system, users who interact with the system and the people who introduce the system in the enterprise. A requirement is defined as a condition or capability that must be met or fulfilled by a system to satisfy a contract, standard, specification, or other formally imposed documents [8]. The requirements defined for a system should be: correct, consistent, verifiable and traceable. It identifies the technological restrictions under which the application should be constructed and run. It is an iterative and cooperative process with the objective to analyze the problem, to

document the results in a variety of formats and evaluate the precision of the results produced. Whenever a software application is built, be it for the Web or not, the development team has to acquire certain knowledge about the problem domain and the application's requirements. The elicitation and specification of these requirements is a complex process as it is necessary to identify the functionality that the system has to fulfill in order to satisfy the users' and customers' needs. Although there is a lack of a standardized process supporting requirements handling and guaranteeing the quality of the results, best practice in the development of general software applications provide a set of techniques [3].

III. REQUIREMENTS ENGINEERING PHASES

The Software Engineering Community recognizes four major activities with the Requirements Generation process: Feasibility Study, Requirements Elicitation and Analysis, Requirements Specification and Requirements Validation although they seem to be separate tasks, these four processes cannot be strictly separated and performed sequentially.

The four main phases of Requirements Engineering are:

A. Feasibility Study
B. Requirements Elicitation and Analysis
C. Requirements Specification
D. Requirements Validation

A .Feasibility Study

An estimate is made of whether the identified user needs may be satisfied using the current software and hardware technologies. The study considers whether the proposed system will be cost effective from a business point of view and whether it can be developed within existing budgetary constraints. A feasibility study should be quick and cheap. The result should inform the decision of whether to go ahead with a more detailed analysis.

B. Requirements Elicitation and Analysis

This is a process of deriving the systems requirements through the observation of existing systems, discussions with potential users and procurers, task analysis and so on. This may involve development of one or more system models and prototypes. These help the analyst understand the system to be specified [2].

C. Requirements Specification

The activity of translating the information gathered during the analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document. They are User Requirement and System Requirement. User Requirements are abstract statements of the system requirements for the customer and the end-user of the system. System Requirements are a more detailed description if the functionality to be provided.

D. Requirements Validation

This activity checks the requirements for realism, consistency and completeness. During the process, errors in the requirements documents are inevitably discovered. It must then be modified to correct these problems.

IV. RESEARCH METHOD

The requirements obtained are classified into various domains [1]. Given documents containing list of requirements and the commonly present keywords we attempt to classify the requirements into specific domains based on the keywords present employing association analysis and hashing techniques by the following procedure. In the first phase using the requirement documents and the keyword list, the requirements table is constructed containing the keywords corresponding to each requirement. The association analysis is applied to the constructed requirements table to get the hash table containing related keywords satisfying the given support and confidence thresholds. Each bucket of the hash table forms an individual domain for requirements classification. In the second phase the hash table is converted into prioritized hash table by finding the frequency of the keywords present in the table. In the third phase a new table called a domain table is created which is similar to priority hash table. This domain table is initially empty, the requirements are hashed to this table based on a given hash function. The hash function assigns requirements to their appropriate domains. Now the requirements are classified based on domains. In the fourth phase we apply the sampling technique for every domain by collecting the requirements and estimate the requirement necessity over the entire population. The requirement collection can be based on questionnaire. The questionnaire should of 'yes' or 'no' format .ie., the questions must have either of the two answers since the distribution is a binomial distribution. The final(fourth) phase is optional and can be done if there is a need for elicitation or the elicitation can be performed using a different or can be postponed to other phases of the process. Finally the requirements are classified based on their domains and elicited using the sampling technique.

V. ASSOCIATION ANALYSIS

Association rules mining is one of the data mining technique which is expected to be very useful in applications. An association expression is an expression of the form $X \rightarrow Y$ where X and Y are disjoint item sets. The strength of association rule can be measured in terms of support and confidence.

$$\text{Support} = \frac{(X \rightarrow Y)}{N}$$

$$\text{Confidence} = \frac{(X \rightarrow Y)}{(X)}$$

Association rules are required to assure a minimum support and a minimum confidence at the same time. Association rule mining is the process of finding all the association rules with the condition of minimum support and minimum confidence. Initially, the support and confidence values are computed for all the rules and it is then compared with the threshold values to prune with low value of support or confidence. A brute force approach to discovering the rules consumes large amount of time and space. Hence association rule discovery is decomposed to two major subtasks.

1. Frequent item set generation: To find all the items that satisfy minimum support threshold. The items discovered are called frequent item sets.
2. Rule Generation: Extract all high confidence rules from the frequent

Item set. These rules are called strong rules.

Brute force approach for frequent item set generation involves comparison of each candidate item set with the transaction. If a candidate is present in transaction, its support count is incremented. This method is very expensive in terms of time and storage. We can minimize the computational complexity by reducing the number of candidate by using algorithms like A priori or by reducing the number of comparisons by using efficient data structures.

Rule generation focuses on extracting association rules from the frequent item set. This can be achieved by partition the item set into two nonempty subsets X and Y-X such that X → Y-X satisfies the confidence threshold. All these rules would have met support threshold as they are drawn from the frequent item set. Confidence based pruning approach or level wise generation of rules using A priori algorithm can be used.

VI. PROPOSED ALGORITHM

PHASE 1: **Using association analysis to find related keywords**

Step 1: Scan the document one containing the list of requirements and the other one containing the common keywords.

Step 2: Create a requirements table which contains the requirement id for each of the requirement and the keywords appearing in that requirement.

Step 3: Examine the keywords in the requirements table and generate the frequent item set of keywords satisfying the support threshold.

Step 4: From the frequent item set generate those rules which satisfy the confidence threshold.

Step 5: We get a list of n rules associating the keywords.

Step 6: We map these n rules to n buckets of the hash table. Each bucket of the hash table represents the keywords appearing in the rule mapped to that bucket.

EXAMPLE:

Requirement Id	Keywords Present
R01	encryption,security,mining
R02	mining,warehouse
R03	encryption,security,privacy
R04	warehouse,mining,business,security
R05	encryption, privacy

Table 1. Requirements Table

Assuming support threshold of 0.2 and confidence threshold of 0.8 we get two rules namely Encryption → Security and Warehouse → Mining.

We get a hash table with the two buckets shown below.

Encryption, Security
Warehouse, Mining

Table 2. Hash Table

PHASE 2: **Assigning the priority to the buckets in the hash table**
Step 1: For each bucket of the hash table assign the frequency count of the keyword from the requirements table that contains keywords mapped to that bucket.
Step 2: The total value of the frequency count of all the keywords in a bucket is the priority value of the bucket.
Step 3: Sort the hash table based on the priority value of each bucket.

Encryption, Security	4
Warehouse, Mining	3

Table 3. New hash table

The first bucket has a frequency of 4 because there are 4 requirements with either one of the keywords in that bucket. By similar logic the second bucket has a frequency count of 3.

PHASE 3: **Classification of the requirements to their respective domains**
Step 1: Construct a new hash table (domain table) containing the buckets same as the priority hash table.
Step 2: Each requirement from the requirement table is hashed to the buckets of the domain table. The hash function is defined as follows.

```
        domains[]  //arrays' indexes are domain index
          for( every keyword present in the requirement )
         {
           for ( Every bucket in the prioritized table )
           {
              if(keyword present in a domain)
                domains[domain]++;
           }
maxvals[] = max( domains[]);
if( Number of maxvals[] > 1)
{
   Assign the requirement to the domain which has the highest priority;
}
else
   Assign the requirement to the domain which gives maxval;
}
```

Step 3: The domain table is now constructed.

PHASE 4: **Using Sampling Technique by applying t-distribution**
Step 1: Collect the requirements in the form of samples.
Step 2: Apply the technique for the samples collected and find the result.

Result:
Consider a set of requirements for a mobile app development
The proposed algorithm classify it into different domains
The following are set of sample requirement classification after applying the association algorithm
Application Domain:
1. Effective and easily navigable user interface
2. Minimization of user input by using auto-complete, suggestions and so on
3. Detailed error messages in case of problems

Scope of Evolution:
1. Possibility of updating to latest software
2. Constant betterment of quality
3. Same basic structure after implementation of changes, to ensure easy adaptability

Market Trends:
1. Tailoring of prices to fit the target users' budgets
2. Positive prior user experience
3. Popularity of platform

Safety and security
1. Appropriate privacy settings
2. Effective anti-virus systems
3. Adequate protection against cyber attacks

Scale/Product analysis:

1. Achieving optimum functionality
2. Use of economically viable processes
3. Holistic approach to design process

Organizational factors:

1. Identification of ownership and stakeholders
2. Effective division of labor
3. Fixed schedules and budget

The requirements are classified based on their domains and the most needed requirements are estimated using the sampling technique [2].
The flow diagram (Fig. 1)to carry out the process as below.

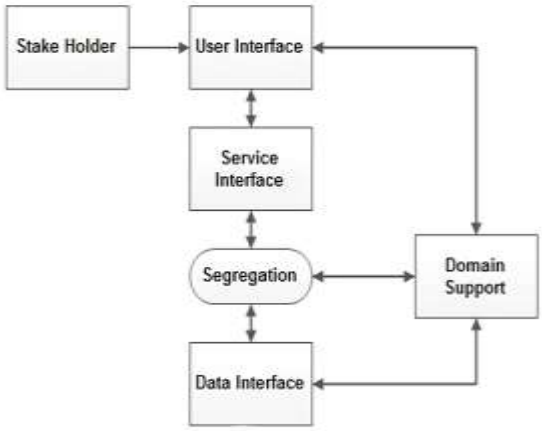

Fig. 1: Flowchart of requirement gathering

The screen shots (Fig. 2) for the above process as below.
In the below screen shot we are uploading a file and using the association algorithm the requirements are classified in to six different domains i.e. for Application Domain the number of requirements are 20 and for scope of evolution the number of requirements are 11 and so on. And on the right hand side of the screen shots shows some of the performance metrics that we will discuss in the next paper.

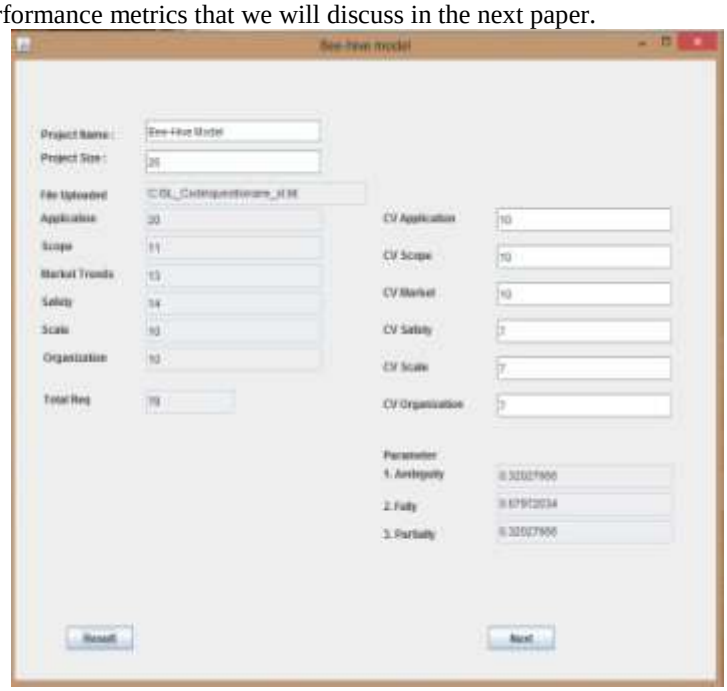

Fig. 2: Screenshot of the output

VII. CONCLUSION

There are lots of difficulties in trying to get the clear and consistent requirements. Once requirements are gathered, difficult to segregate into different domain. Our proposed association algorithm effectively segregates the requirements into six different domains by using simple techniques of association analysis and hashing. Upload the document consists of number of requirements the proposed algorithm classify it into six different domains.

VIII. REFERENCES

[1] Swarnalatha K S, G.N Srinivasan, Pooja S Bhandary "a constructive and dynamic framework for requirement engineering process model – bee hive model" International Journal of Computer Engineering and Technology (IJCET), ISSN 0976-6367(Print),ISSN 0976 - 6375(Online), Volume 5, Issue 7, July (2014), pp. 48-54 © IAEME.

[2] Swarnalatha K S, G N Srinivasan, Rakesh R, Vipin Dwivedi "software requirement collection enhancement using sampling technique and applying t-distribution" International Journal of Computer Engineering and Technology (IJCET), ISSN 0976-6367(Print), ISSN 0976 - 6375(Online), Volume 5, Issue 7, July (2014), pp. 67-72 © IAEME.

[3] Swarnalatha K S , G.N Srinivasan ,Harshvardhan V Rawoot, Rakesh R, Nishkarsh Singh" Different approaches for Dynamic Requirement Engineering Process Models: Cone and Hour Glass Model" Paper got accepted in IEEE conference 2014(ICAECC-14).

[4] AzlenaHaron and ShamsulSahibuddin, "TheStrength and Weakness of Requirement Engineering (RE) Process," presented at the ICCTD 2012: 2012 the 2nd International Conference on Computer Technology and Development, Cairo, Egypt, 2012.

[5] B. Mutschler, M. Reichert, and J. Bumiller,"Unleashing the Effectiveness of Process-Oriented Information Systems: ProblemAnalysis, Critical Success Factors, and Implications," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 3, pp. 280-291,2008.

[6] Liu, L. and L. Lin, 2012. ARED-CK: An automated requirements elicitation approach based on decision-making with complete knowledge. Proceeding of the 1st IEEE International Workshop on Managing Requirements Knowledge, Sch. of Software, Tsinghua Univ., Beijing, pp: 47-52.

[7]. Mishra, D., A. Mishra and A. Yazici, 2012. Successful requirement elicitation by combining requirement engineeringtechniques. Proceeding of the 1st IEEE International Conference on the Applications of Digital Information and Web Technologies,Dept. of Comput. Eng., Atilim Uni., Ankara, pp: 258-263.

[8]. Ramsin, R. and R.F. Paige, 2011. Iterative criteria-based approach to engineering the requirements of software developmentMethodologies. IET Software, 4(2): 91-104.

[9]. Sharp, H., C.W.A. Finkelstein and H. Galal, 1999. Stakeholder identification in the Requirements engineering process. Proceeding of the 10th IEEE International Workshop on Database and Expert Systems Applications, Centre for HCI Design, City Univ., London, UK, pp: 387-391.

[10]. Singh, Y., P.K. Bhatia and O. Sangwan, 2007. A review of studies on machine learning Techniques. Int. J. Comp. Sci. Secur., 1(1): 70- 84.

[11]. Wongthongtham, P., E. Chang, T.S. Dillon and I. Sommerville, 2009. Development of software engineering ontology for multisite software development. IEEE Trans.Knowl. Data Eng., 21(8): 1205-1217.