

# A Methodological Approach for Mining the User Requirements Using Apriori Algorithm

Anuja Soni, Deen Dayal Upadhyaya College, University of Delhi, New Delhi, India

Anand Saxena, Deen Dayal Upadhyaya College, University of Delhi, New Delhi, India

Parul Bajaj, D.A.V. Public School, Gurugram, India

## ABSTRACT

Users of enterprise software are multiple, and their requirements are diverse. Often their specifications are masked by mundane details and at times are vague too. Acknowledging these complexities in requirements engineering, the paper proposes a multistage methodological approach based on Apriori algorithm, a data mining technique. It extracts useful information from the given data on the criteria of mutual association and sufficient frequency. The user requirements captured through interviews and brainstorming are pre-processed for eliminating unnecessary stop words and developing a uniform structure of small stories. Mutual association and occurrence of the requirements are represented through association rules and rule metrics, for example, 'Lift', 'Support', and 'Confidence'. The requirements having strong and moderate association are placed in 'Top Priority List'; those with nominal, weak, or nil association are placed in 'Low Priority List'. Gap analysis is employed to validate the defined requirements with respect to stakeholders' expectations. The complete and correct lists of requirements significantly influence the client satisfaction, software development process, and its eventual success.

## KEYWORDS

Apriori Algorithm, Association Rules, Confidence, Gap Analysis, Lift, Requirements Engineering, Support, User Requirements

## 1. INTRODUCTION

Ascertaining user requirements is the foremost exercise in any product, more so Business-to-Business (B2B) product development. The paper pertains to mining the user-requirements for software development.

The paper acknowledges that the user requirements, especially in multi-use, multi-domain, multi-location environments that most large businesses of today are characterised by, are multiple, diverse and variously expressed. This necessitates development of a methodology that permits effective sourcing, coalescing and collating of the user-requirements.

The paper proposes Apriori algorithm as the preferred technique for mining the user requirements. It addresses and automates two prime issues of user requirements in an integrated manner. One, it focuses on knowledge-driven elicitation of user requirements, where knowledge is extracted in the form of mutual association and frequent occurrence of user-requirements. Two, it stratifies the elicited

DOI: 10.4018/JCIT.2020100101

This article, originally published under IGI Global's copyright on September 23, 2020 will proceed with publication as an Open Access article starting on January 18, 2021 in the gold Open Access journal, Journal of Cases on Information Technology (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

user-requirements into two categories-“Top Priority Apriori List” and “Low Priority Apriori List”- that reflect the comparative strength of the concerns of all categories of users and prime stakeholders in the development of the software. Thus, Apriori algorithm filters out the mundane details in the user descriptions of their requirements (Mussabacher, 2016; Wong, Mauricio, & Rodriguez, 2017; AlMousa, Al-Khalifa, & AlSobayel, 2017). Further, the requirements so elicited are validated with the help of a fuzzy linguistic survey to generate “High Priority Survey List” and “Low Priority Survey List.” The paper uses two-way gap analysis to ensure the correctness and completeness of the user requirements.

### 1.1 Prevalent Requirements Elicitation Techniques

The prevalent requirements elicitation techniques include User Surveys, Scenario centred practices, Investigations, Use cases, User expressions using natural language, Goal-directed structural model, Social Networks, Ontology-operated and Analytic Hierarchy Process (AHP) etc. (Krishna & Lu, 2008; Farfeleder et al., 2011; Goh & Kok, 2010; Macasaet, Chung, & Garrido, 2011; Lim, & Finkelstein, 2012; Jiang, Ruan, Zhang, & Li, Lew, 2014; Jin, Donzé, Deshmukh, & Seshia, 2015; Karras, Kiesling, & Schneider, 2016; Mishra, Aydin, & Ostrovska, 2017).

Krishna et al. (2008) use domain acquaintance and Business Model to simulate the users’ requirements.

Farfeleder et al. (2011) offer an experiment based semantic regulation technique that supports the requirements analyst to elicit the requirements from users using the principles of domain ontology. Maintaining the consistency in the requirements, however, is a challenging task within domain ontology framework in view of the growing number of ontology axioms.

Macasaet et al. (2011) take into account the case study of micro-business and apply various methods viz. business-activity models, pictographic blueprints and quality requisites to delineate user-requirements, while (Przybylek, 2014; Valvas & Milani, 2015) employ industrial process engineering to obtain the requirements intended to fulfill system needs.

Lim et al. (2012) make use of a software prototype viz. StakeNet to collect sanctions and priorities from the participants, whereas Pecchia (2013) utilizes a logical and understandable structure for decision-making to gather the requirements from the users.

Jiang (2014) offers online criticism based requirements elicitation for the future releases of the product, however it is accompanied with the challenge to deal with the requirements for a new product from the scratch.

Jin (2015) offers Simulink based approach to capture the requirements of a system which is restricted to a time-specific control-systems.

Karras (2016) integrates video analysis with textual notes to capture user requirements. It appears to be a simpler technique for small scale projects, however with the increasing size of the project, it may result in complications and ambiguous requirements.

Mishra (2017) proposes an effective online tool SRMS to capture the experiences of other developers to meet adapting requirements of the system. However, this approach needs some improvements viz. replacing discrete input values of the users by their subjective views, specification of user-acquaintance with the respective domain etc.

While the techniques reviewed in the foregoing paragraphs have their distinctive merits vis-à-vis each other, yet these are constrained in the following ways:

- These techniques employ the graphical notations, computational logical based specifications, large sized graphical business models, scenarios, responders’ judgment, questionnaires and domain axioms etc. These methods however are fraught with the challenges of scalability, long period of maintenance and enhanced notations.

- Some researchers (Manzoor, Shaheen, & Khalid, 2018) make use of survey forms and interviews to obtain user requirements. However, such endeavours seem suitable more during the formative phases of generating a comprehensive set of system requirements.
- Most techniques for requirements elicitation viz. (Przybylek, 2014; Valvas & Milani, 2015; Kushihiro, Shimizu, & Ehira 2016; Lian, Rahimi, Huang, & Li Zhang, 2016) are manual, instruction based and pictographic and hence prone to errors.
- These techniques miss out on their mutual association and multiple occurrences. Such as omission deviates the requirements analyst from users' actual needs leading to the faulty design and coding in the development of the software system.

## 1.2 Merits of Apriori Algorithm for Mining the User Requirements

Apriori algorithm applied for mining the user requirements, has the following benefits over existing requirements elicitation approaches:

- It has the potential to deal with huge amount of user-requirements with trouble-free maintenance.
- It supports an automated approach for excavating knowledge from user requirements in the form of association rules and hence is less prone to misperception and flaws.
- It is capable of capturing clear patterns in user requirements instigated by their mutual association and frequent occurrence and hence characterizes the true needs of the users.
- It yields complete and correct prioritized lists of requirements.

## 1.3 The Methodological Approach of the Paper

The paper follows a multi-stage approach in which, at first, the proposed software system is divided in different sub-parts/modules and several types of users working in different domains are identified. At this stage, traditional techniques like interviews and brainstorming are applied to capture the requirements from the users. Pre-processing is applied to eliminate unnecessary stop words from the requirements while giving them a uniform shape of small user stories. Apriori algorithm is applied on these user stories, which enables the requirements analyst to find out the associated user requirements by predicting their mutual association and frequent occurrence.

Secondly, mutual association and occurrence of requirements are depicted in the form of association rules. The metrics 'Lift', 'Confidence' and 'Support' act as the benchmarks for measuring the mutual association and frequent occurrence of requirements. 'Support' is a metric that computes the frequency of a user requirement with respect to total number of data-capturing sessions held with the users. Higher value of 'Support' of a requirement will indicate that requirement is highly desired by the users. 'Confidence' and 'Lift' are the parameters to measure the mutual association between two or more requirements. 'Confidence' is a metric that quantifies the mutual association between requirements (R1, R2) by enumerating their co-occurrences with respect to total occurrence of R1 whereas 'Lift' is a metric that is represented as a ratio of 'Confidence' and 'Expected Confidence' of (R1, R2). 'Expected confidence' is the probability of appearing R2 in the absence of R1. High value of 'Confidence' and 'Lift' will lead to a high level of mutual association among requirements. On the basis of the extent of mutual association and frequency of their occurrence, the requirements are segregated into two parts: 'Top Priority Apriori List' and 'Low Priority Apriori List', where 'Top Priority Apriori List' and 'Low Priority Apriori List' are lists of high priority and low priority requirements obtained after applying aforementioned Apriori based approach.

Thirdly, for requirements validation, a survey technique is adopted and some highly experienced stakeholders are short-listed to express the importance of the requirements on a fuzzy linguistic scale i.e. Highly Important (HI), Moderately Important (MI), Nominally Important (NI), and Weakly Important (WI). 'Top Priority Survey List' and 'Low Priority Survey List' are obtained after applying the survey technique.

Finally, Gap-analysis is applied to map the requirements in both the directions forward and backward i.e from 'Top Priority Apriori List' and 'Top Priority Apriori List' to 'Top Priority Survey List' and 'Low Priority Survey List' respectively and vice versa. Ultimate mapping will endorse the completeness and correctness of the requirements.

The requirements thus, prioritized will facilitate the requirements analyst and users to incline to the main goals of final system leading to success of final system.

The complete layout of the paper is as follows: Section 2 acquaints the reader with a background of data mining concepts viz. Apriori algorithm, association rules and rule-metrics. Section 3 presents a methodological approach for mining the user requirements, while section 4 presents an experimental study exercised on inventory and procurement requirements. Section 5 provides a comparative study of various requirements prioritization techniques. Section 6 concludes the paper and finally section 7 introduces challenges and forthcoming work.

## 2. BACKGROUND

The Apriori algorithm, a popular data mining process, discovered by Agrawal and Srikant is capable of discovering frequent patterns inherent in database containing lots of transactions (Wang, Shi, Bai, & Zhao 2009). This algorithm employs association rules and rule metrics to extract items frequently occurring together. The following sub-sections concisely explains Apriori algorithm, association rules and rule metrics- 'Support', 'Confidence' and 'Lift'.

### 2.1 The Apriori Algorithm

Apriori methodology is a data mining approach that has been applied successfully in the sphere of multiple domains viz. market basket analysis, healthcare domain and e-commerce recommender system (Abullah, 2008; Wang et al., 2009). This algorithm helps in identifying frequently occurring items in a large data set by generating the association rules among them. This algorithm employs several rule-metrics viz. 'Support', 'Confidence' and 'Lift' to filter out intense association rules with a strong relationship among them. The brief description of association rules and rules measurements metrics is given in the following sub-sections.

### 2.2 Brief Introduction to Association Rules

Association rules, in data mining are used for discovering certain patterns in bulk of data. A specific group of data may be strongly related or may lack any kind of relationship. Association rules are used to determine the relationship as strong or weak in a certain data. It is a relation between two or more entities which is formulated as (Abullah, 2008).

$$P \Rightarrow Q$$

where:

- P, Q are disjoint items  $\in$  DtSet: a data set of multiple items.
- ' $\Rightarrow$ ' is the association symbol which implies a relation from 'P' to 'Q'

An association rule has the following structure:

Left – Hand – Side  $\Rightarrow$  Right – Hand – Side

Left-Hand-Side (LHS) is termed as ‘antecedent’ while Right-Hand-Side (RHS) is named as ‘consequent’.

The intensity of an association rule consisting of antecedent and consequent can be fragile, strong or null depending upon the values of rule-measurement metrics. ‘Support’, ‘Confidence’ and ‘Lift’ are called as rule-measurement metrics to evaluate the intensity of an association rule.

A short explanation of association rule-metrics is provided in the succeeding section.

## 2.3 Association Rule-Metrics

‘Support’, ‘Confidence’ and ‘Lift’ are the units for measuring the strength of association rules.

### 2.3.1. Support

‘Support’ signifies the ratio of number of occurrences of an item ‘P’ with respect to total number of transactions in a specific data set. ‘Support’ of P is computed by Equation 1 (Wang et al., 2009):

$$\text{Support}(P) = \frac{\text{Frequency}(P)}{N} \quad (1)$$

where:

N: Total number of transactions in a certain data set.

Frequency(P): A numeric value in between [0,N] that refers to number of times ‘P’ occurs in the data set.

Value of ‘Support(P)’ is always in between 0 and 1. Value of ‘Support(P)’ as 0 means that there is no transaction in the data set that contains ‘P’, whereas value as 1 signifies that ‘P’ appears ‘N’ number of times in the data set.

In the same manner, ‘Support’ of an association rule ‘ $P \Rightarrow Q$ ’ is calculated by Equation 2:

$$\text{Support}(P \Rightarrow Q) = \frac{\text{Frequency}(P, Q)}{N} \quad (2)$$

Frequency(P, Q) is the count of co-occurrence of ‘P’ and ‘Q’, whereas ‘N’ denotes a numerical value referring to overall transactions in the given data set.

### 2.3.2. Confidence

‘Confidence’ is a measure of assurance of co-occurrence of two or more items in a specific data set.

In a data set DtSet, the confidence of the rule ‘ $P \Rightarrow Q$ ’ is measured by Equation 3:

$$\text{Confidence}(P \Rightarrow Q) = \frac{\text{Frequency}(P, Q)}{\text{Frequency}(P)} \quad (3)$$

Confidence( $P \Rightarrow Q$ ) is the ratio of frequency of co-occurrence of ‘P’ and ‘Q’ with respect to the frequency of occurrence of ‘P’ in the given dataset. The value of ‘Confidence’ is always in between 0 and 1.

Value of ‘Confidence’ as 0 signifies that there is no such transaction in the database, which consists of both the items P and Q while value of ‘Confidence’ as 1 means Q happens to be there in each and every transaction where P appears.

### 2.3.3. Lift

Lift( $P \Rightarrow Q$ ) is defined as the ratio of actual value of ‘Confidence’ observed and ‘Expected confidence’ as shown in Equation 4 (Wang et al., 2009):

$$\text{Lift}(P \Rightarrow Q) = \frac{\text{Confidence}(P \Rightarrow Q)}{\text{Expected confidence}(Q)} \quad (4)$$

Confidence( $P \Rightarrow Q$ ) is the probability of appearing ‘Q’ in the presence of ‘P’, whereas Expected confidence( $Q$ ) is the probability of appearing ‘Q’ in the absence of P.

Value of Confidence( $P \Rightarrow Q$ ) is computed by the Equation (3) whereas expected confidence is computed by Equation 5:

$$\text{Expected Confidence}(Q) = \text{Support}(Q) = \frac{\text{Frequency}(Q)}{N} \quad (5)$$

Value of lift is always greater than zero and can assume any rational value up to infinity (Abullah, 2008).

The value of lift  $> 1$  pertaining to the rule  $P \Rightarrow Q$ , indicates that the variable P is strongly related to the variable Q, on the other hand the value of lift  $< 1$  will mean that variable P is negatively co-related to Q tending to a fragile relation between P and Q.

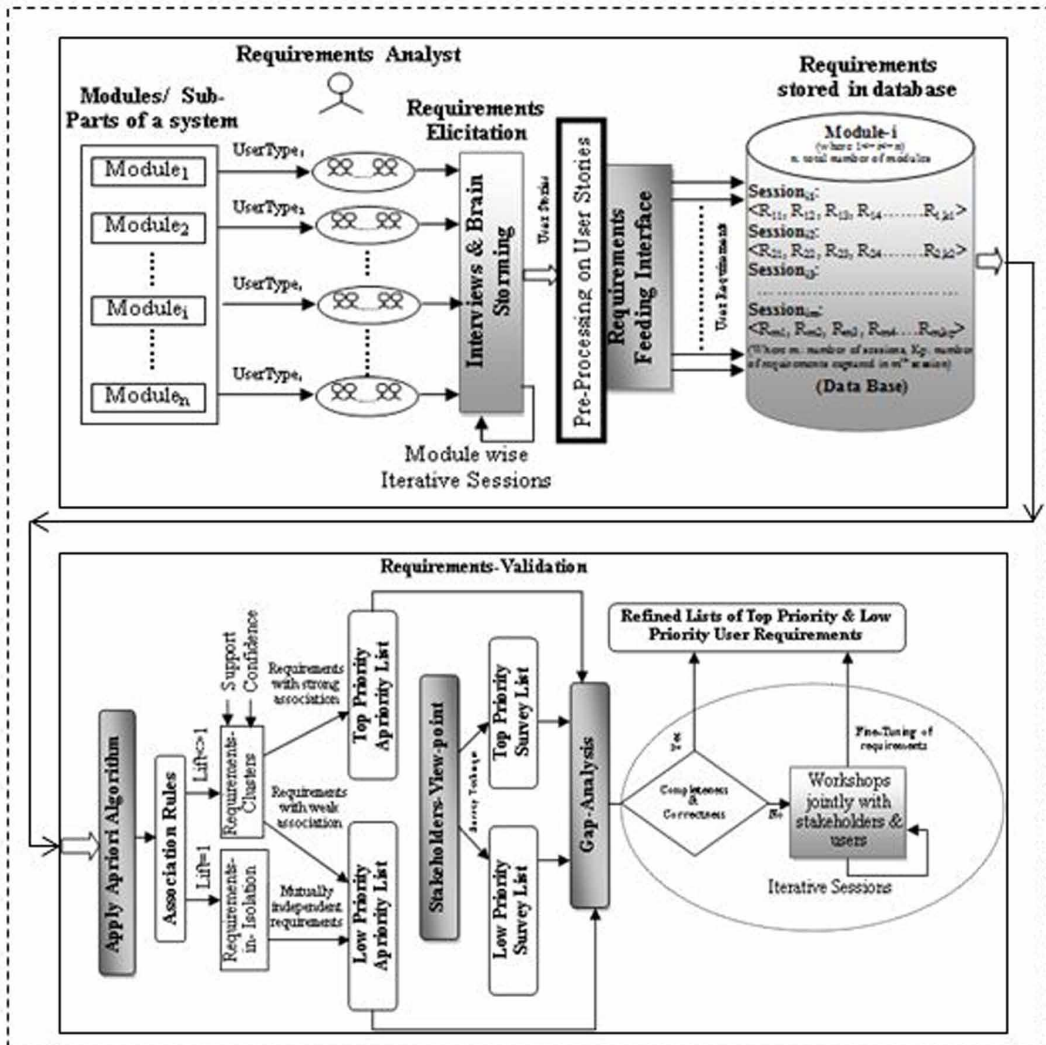
If lift = 1, then it means the variable P (antecedent) and variable Q (consequent) are mutually independent and are not appearing together in any association rule. In case of their mutual independence, the value of Confidence( $P \Rightarrow Q$ ) becomes equal to Expected Confidence( $Q$ ) tending to the value of lift as 1.

## 3. THE METHODOLOGICAL APPROACH USED IN THE PAPER

The methodological approach for mining the user requirements using Apriori algorithm is shown in Figure 1. As may be seen from Figure 1, at first, the software system is divided in different sub-parts/modules and several types of users working in different work domains are identified. The conventional techniques “Interviews” and “Brain-Storming” are used by requirements analysts to elicit the requirements from the users. Multiple interactive and iterative sessions are held between users and requirements analysts. The user requirements are recorded in the form of small user-stories. User-story is a small sequence of user requirements which represents the crux of the requirement briefly in a few words (Gaur, Soni & Bedi, 2010). User stories can be documented manually using paper-pen or directly keyed in the computer through some requirements feeding interface.

All the user-stories may not be uniform. Two or more user-stories may consist of distinct words but with same meaning. Moreover, the user-stories may contain some stop words (the insignificant words having no semantic value) e.g. ‘of’, ‘from’, ‘in’ etc. In order to remove such kind of anomalies and bring uniformity to the user-stories, a pre-processing is applied in the following manner:

Figure 1. The methodological approach for mining the user requirements using apriori algorithm



- Stop words are identified in the user-stories and removed.
- Distinct words but with the same meaning are replaced by synonymous words.

Refined user-stories representing the user requirements are stored in a database. Apriori algorithm is applied on the requirements of each and every module. The metrics 'Lift', 'Confidence' and 'Support' are used to measure the level of mutual association among frequently recurring requirements.

This approach enables the requirements analyst to divide the requirements on the basis of their mutual association in two groups: 'Requirements-Clusters' and 'Requirements-in-isolation'. 'Requirements-Clusters' represent the group of those requirements which are frequently occurring together whereas 'Requirements-in-isolation' represents the group of those requirements which are appearing in isolation and hence do not have any mutual association among them.

Further, the requirements are placed in two types of lists: 'Top Priority Apriori List' and 'Low Priority Apriori List'. The requirements having strong or moderate association are placed in 'Top

Priority Apriori’ list whereas the requirements with marginal, weak or nil association are placed in ‘Low Priority Apriori list’. A survey technique is applied and skilled and highly experienced stakeholders are involved to provide their view-points in online questionnaires. The gap analysis is utilized to bridge up the gap between two sets of prioritized requirements: one obtained after applying Apriori algorithm whereas another obtained after applying Survey technique. The process of requirements-validation using gap-analysis is repeated until all requirements are refined upto the satisfaction of users and all stakeholders which ultimately will lower the risk of collapse of final system.

The proposed approach is mentioned in detail in the following steps:

Step 1: Divide the entire software system in ‘n’ number of modules and identify various types of following ‘UserTypes’ pertaining to these modules:

$< \text{UserType}_1$   
 $\text{UserType}_2$   
 $\text{UserType}_3$   
 $\text{UserType}_n >$

where  $\text{UserType}_{i(1 \leq i \leq n)}$  refers to a group of users working in the specific domain of  $i^{\text{th}}$  module of the system.

Step 2: Any of following kind of techniques existing in literature can be applied iteratively to obtain the requirements from a  $\text{UserType}_{i(1 \leq i \leq n)}$  :

- Brain storming
- Interviews etc.

Step 3: Let ‘m’ be number of sessions for obtaining the requirements from  $\text{UserType}_{i(1 \leq i \leq n)}$  under  $i^{\text{th}}$  module.

Value of ‘m’ can be fixed to a discrete value by the management committee to capture the true requirements of the users.

Let, Dt. be a database that comprises of frequently recurring requirements obtained from users during various sessions pertaining to  $i^{\text{th}}$  module ( $1 \leq i \leq n$ ):

Database ‘Dt’

**Session<sub>1</sub>:**  $\langle Rq_1, Rq_3, Rq_5, \dots Rq_{j1} \rangle$   
**Session<sub>2</sub>:**  $\langle Rq_1, Rq_2, Rq_5, Rq_6, Rq_7, \dots Rq_{j2} \rangle$   
**Session<sub>3</sub>:**  $\langle Rq_1, Rq_3, Rq_6, \dots Rq_{j3} \rangle$   
**Session<sub>m</sub>:**  $\langle Rq_5, Rq_6, Rq_1, Rq_2, \dots Rq_{jm} \rangle$



where requirements are repeated over multiple sessions and number of requirements happens to be a dynamic number which varies from session to session.

$Rq_{jm}$  : refers to  $j^{th}$  requirement during  $m^{th}$  session

Step 4: The requirements are stored in a database using any data base management system viz. Ms-Access, Oracle, MySQL, Ms. Excel or any other.

Step 5: Apriori algorithm is applied on database of requirements 'Dt' in the following manner to obtain frequent requirement-patterns and henceforth commonly befalling association rules:

- Scan the requirements-database 'Dt'
- Generate a new database that consists of all candidate requirements along with their occurrence-count. The 'occurrence-count' of a requirement refers to its frequency in a given database.
- Scan the database mentioned in step-b and extract those requirements which have their occurrence-count  $\geq 2$  (where '2' is assumed to be a minimum desired level of occurrence-count). The requirements with their occurrence-count  $\geq 2$  indicate a single order database of frequent-requirements-patterns and is designated as 'Dt-1' as shown in Table 1.

**Table 1. Single order database 'Dt-1' of frequent-requirements-patterns with occurrence-count  $\geq 2$**

Requirements	Occurrence - Count ( $\geq 2$ )
$Rq_1$	(Count) <sub><math>Rq_1</math></sub>
$Rq_2$	(Count) <sub><math>Rq_2</math></sub>
$Rq_3$	(Count) <sub><math>Rq_3</math></sub>
$Rq_4$	(Count) <sub><math>Rq_4</math></sub>
.....	.....
$Rq_{j1}$	(Count) <sub><math>Rq_{j1}</math></sub>
.....	.....
$Rq_{j2}$	(Count) <sub><math>Rq_{j2}</math></sub>
.....	.....
$Rq_{jm}$	(Count) <sub><math>Rq_{jm}</math></sub>

Here ' $Rq_{ji}$ ' is a requirement pattern of single order and (Count) <sub>$Rq_{ji}$</sub>  refers to its occurrence-count:

- Apply the self-join operation 'Dt-1 JOIN Dt-1'. The self-join operation will generate a 2-order database of frequent-requirements-patterns by combining each and every candidate requirement with remaining all candidate requirements.
- The aforementioned database is scanned and the requirements with their occurrence-count  $\geq 2$  are chosen and saved in database 'Dt-2' as shown in Table 2. During the process the requirements patterns e.g.  $\{Rq_1, Rq_{j1}\}$  and  $\{Rq_2, Rq_{j2}\}$  assumed to have their occurrence-count  $\leq 2$  are eliminated and not included in database 'Dt-2'.

Here ' $\{Rq_i, Rq_{j1}\}$ ' is a requirement pattern of 2<sup>nd</sup> order and (Count) <sub>$\{Rq_1, Rq_{j1}\}$</sub>  refers to its occurrence-count.

**Table 2. Second order database ‘Dt-2’ of frequent-requirements-patterns with occurrence-count  $\geq 2$**

Requirements	Occurrence-Count ( $\geq 2$ )
$\{Rq_1, Rq_2\}$	(Count) <sub><math>\{Rq_1, Rq_2\}</math></sub>
$\{Rq_1, Rq_3\}$	(Count) <sub><math>\{Rq_1, Rq_3\}</math></sub>
$\{Rq_1, Rq_4\}$	(Count) <sub><math>\{Rq_1, Rq_4\}</math></sub>
.....	.....
$\{Rq_1, Rq_{j1}\}$	(Count) <sub><math>\{Rq_1, Rq_{j1}\}</math></sub>
.....	.....
$\{Rq_1, Rq_{j2}\}$	(Count) <sub><math>\{Rq_1, Rq_{j2}\}</math></sub>
.....	.....
$\{Rq_1, Rq_{jm}\}$	(Count) <sub><math>\{Rq_1, Rq_{jm}\}</math></sub>
$\{Rq_2, Rq_3\}$	(Count) <sub><math>\{Rq_2, Rq_3\}</math></sub>
$\{Rq_2, Rq_4\}$	(Count) <sub><math>\{Rq_2, Rq_4\}</math></sub>
.....	.....
$\{Rq_2, Rq_{jm}\}$	(Count) <sub><math>\{Rq_2, Rq_{jm}\}</math></sub>
.....	.....

The self-join operation ‘Dt-(k-1) JOIN Dt-(k-1)’ with ‘k-1’ order is iteratively applied till  $Dt-k = \phi$ , where Dt-k is a database of k-order.

During the self-join operation, the Apriori property is applied to avoid large number of computations. The Apriori property (Wang et al., 2009) states that if a set is not frequent in the database then all its supersets definitely will not be frequent and hence can directly be eliminated.

The above sequence of steps facilitates to obtain frequent requirements patterns and induce the corresponding association rules. For instance, the frequent requirement pattern  $\{R_1, R_2\}$  obtained through aforesaid process will indicate an association rule of type: ‘ $R_1 \Rightarrow R_2$ ’.

Step 6: The metrics ‘Lift’, ‘Confidence’ and ‘Support’ are used to measure the intensity of so obtained association rules.

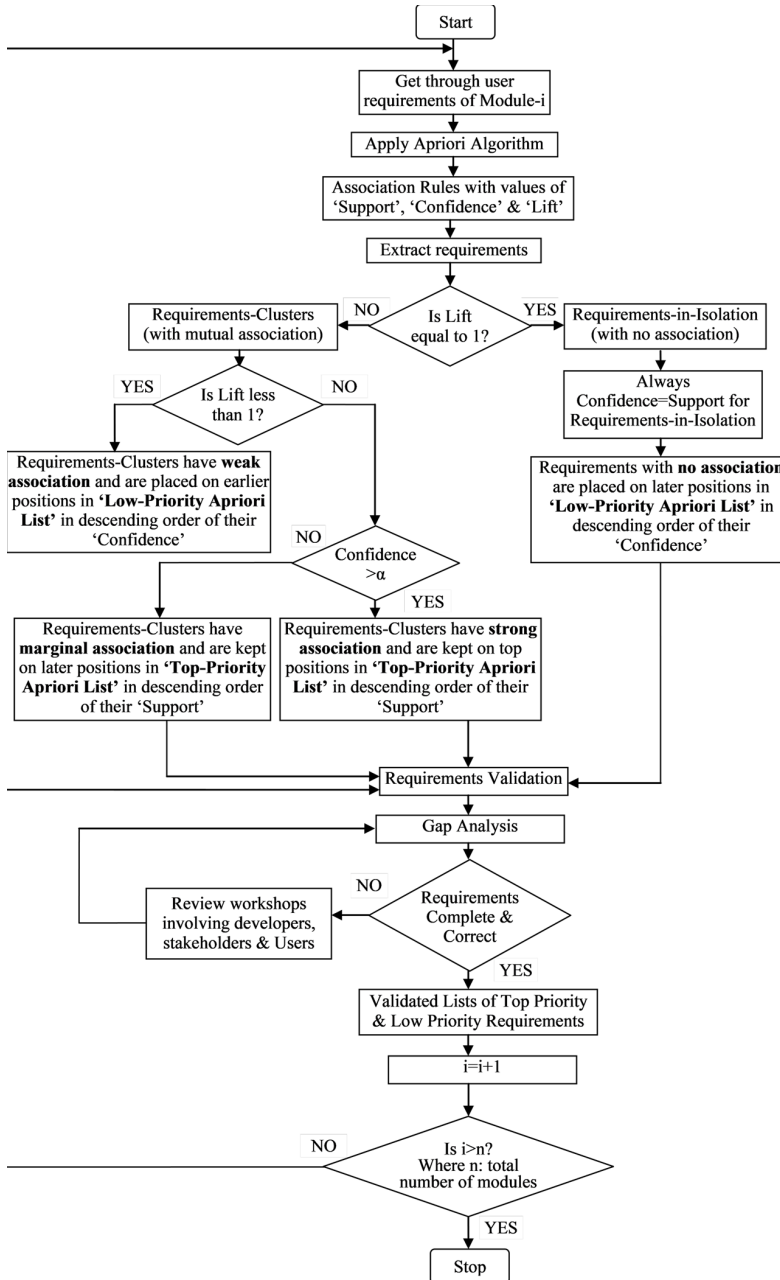
Step 7: The requirements inclined to association rules are segregated in two parts: ‘Top priority list of requirements’ and ‘Low priority list of requirements’. The entire process for extracting desired sets of requirements is explained in detail through the flow chart shown in Figure 2.

Step 8: Gap analysis as explained in section 3.1, is employed to validate the requirements of top priority list and same process is repeated for the low priority list.

Few of the terms used in the flow chart are defined as below:

- **Requirements-Cluster:** A set of frequently occurring requirements together in ‘m’ number of sessions pertaining to  $i^{th}$  module of the system (where  $1 \leq i \leq n$ , while ‘n’ denotes total number of modules).
- **Requirements-in-Isolation:** A group of requirements occurring in isolation in  $i^{th}$  session (where  $1 \leq i \leq m$ ).
- **Top Priority Apriori List:** A list of highly desired user requirements obtained after applying Apriori-based approach.
- **Low Priority Apriori List:** A list of less desired user requirements after applying Apriori-based approach.

Figure 2. The flow chart depicting the methodological approach of mining user requirements



Values of 'Lift', 'Confidence' and 'Support' enable the requirements analyst to identify mutual association among requirements and henceforth place these requirements in a specific priority list in the following manner.

Value of '**Lift=1**' signifies that requirements are mutually independent and occurring in isolation. These requirements have no mutual association; therefore these will have the least priority to be added in the descending positions of the 'Low Priority Apriori' list of requirements.

The value of '**Lift**<1' will indicate that requirements have some mutual association and happen to occur together in the form of clusters:

- Value of '**Lift**<1' will indicate that requirements belonging to a particular domain are negatively associated to each other and occasionally happen to occur together leading to a weak association among them. Hence these are placed in later positions of 'Low Priority Apriori' list of requirements in descending order of their values of 'Confidence'.
- Value of '**Lift**>1', will imply substantial level of association among the requirements. In order to decide precise level of association, values of 'Confidence' and 'Support' are observed.

Fix up the desired level of 'Support' and 'Confidence' as  $\alpha$  and  $\beta$  respectively. The threshold values of  $\alpha$  and  $\beta$  can mutually be decided by the requirements-collecting team.

- Value of '**Confidence**>= $\alpha$ ' will indicate that requirements in a cluster are occurring together quite frequently leading to a high-level of domain-binding:
  - For such requirements, value of 'Support' is further observed. Value of '**Support**>= $\beta$ ' will imply that frequency of occurrence of requirements in a cluster is very high. Simultaneous values of '**Confidence**>= $\alpha$ ' and '**Support**>= $\beta$ ' will imply that requirements in a cluster have strong association with high frequency of occurrence. Hence these requirements will represent the true needs of the users to be placed on higher positions of 'Top Priority Apriori' list'.
  - On the other hand, simultaneous values of '**Confidence**>= $\alpha$ ' and '**Support**< $\beta$ ' will imply that though level of confidence of the users in the requirements is high but frequency of such requirements is comparatively less. Hence these requirements will have moderate association among them to be kept in later positions of 'Top Priority Apriori' list of requirements'.
- Value of '**Confidence**< $\alpha$ ' will indicate that occurrence of mutually associated requirements is less leading to the nominal association. Such requirements, though, placed in 'Low Priority Apriori' list of requirements' but at the top positions.

The completeness and correctness of the Apriori lists is validated by gap analysis.

### 3.1 Gap Analysis to Validate User-Requirements

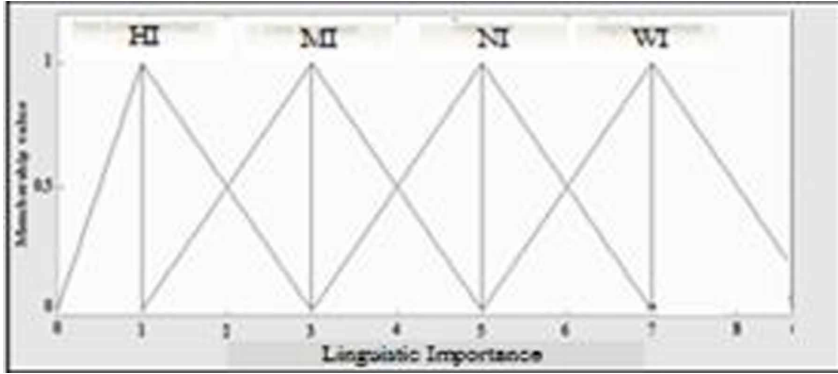
Gap Analysis is portrayed as the contrast between what is required and what is accessible. It could be a comparison handle of two frameworks, and is embraced as a method of bridging the crevice between them (Amaral and Faria, 2010). The objective of the gap analysis is to distinguish the inconsistencies within the characterized necessities and users' needs.

The gap analysis in this work bridges up the gap between two lists of requirements: one 'Top Priority Apriori List' and another 'Top Priority Survey List'.

'Top Priority Apriori List' is a sorted list of high priority requirements obtained after applying aforementioned Apriori based approach, whereas 'Top Priority Survey List' is a sorted list of high priority requirements obtained after applying the survey technique. A survey technique is a technique in which online questionnaires are designed and skilled and highly experienced stakeholders are involved to provide their view-points using the linguistic terms viz. Highly Important (HI), Moderately Important (MI), Nominally Important (NI), Weakly Important (WI). The linguistic terms are mapped to Triangular Fuzzy Numbers (TFNs) using the scale shown in the Figure 3.

The linguistic importance of each requirement is mapped to TFN (l, m, u) with l as smallest, m as most promising and u as the largest value, which further is converted to a crisp interval  $^{\alpha}CrispInterval = [B_{al}, B_{au}]$  with  $B_{al}$  =lower bound and  $B_{au}$  =upper bound using the Equation 6 (Gaur & Soni, 2010):

Figure 3. Linguistic terms mapped to TFNs



$${}^{\alpha}CrspIntrvl = \left[ (m-l)\alpha + l, -(u-m)\alpha + u \right] = \left[ B_{\alpha l}, B_{\alpha u} \right] \quad (6)$$

where  $\alpha \in [0,1]$  signifies the confidence level of the stakeholders and is largely taken as 0.5 to signify the moderate level of confidence.

The lower and upper bounds of crisp interval are mapped to the crisp value  $C_{\alpha}^{\mu}$  using the Equation 7:

$$C_{\alpha}^{\mu} = \mu B_{\alpha u} + (1 - \mu) B_{\alpha l} \quad (7)$$

where  $\mu \in [0,1]$  denotes the optimistic level of the stakeholders and is usually taken as 0.5 to show their moderate level of optimism.

The fuzzy linguistic validations of the user-requirements rendered by the experts are mapped to their crisp values. The final priority of each requirement is computed by taking the arithmetic mean of all its crisp equivalents. The requirements with their priority values  $\geq 0.5$  are placed in descending order in the “Top-Priority Survey List” whereas remaining requirements with low priorities are placed in the “Low-Priority Survey List”.

To ensure the correctness and completeness of the requirements, ‘Top Priority Apriori List’ and ‘Top Priority Apriori List’ are mapped to the ‘Top Priority Survey List’ and ‘Low Priority Survey List’ respectively in forward and backward directions.

### 3.1.1. Completeness

The completeness of requirements is a capacity that alludes to the joining of users’ needs totally within the last list of necessities (Gaur & Soni, 2013). Equation 8 is used to ensure the coverage of all relevant user requirements in the ‘Top Priority Apriori List’:

$$COM_{Top\ Priority\ List\ Apriori} = \frac{Map\ Requirements_{Top\ Priority\ List\ Apriori}}{TOT\ Requirements_{Top\ Priority\ Survey\ List}} \quad (8)$$

where ‘MapRequirements<sub>Top Priority Apriori List</sub>’ refers to number of those requirements which are mapped from ‘Top Priority Apriori List’ to the requirements of ‘Top Priority Survey List’.

The value  $COM_{Top Priority Apriori List} = 1$  would endorse the completeness of all requirements in ‘Top Priority Apriori List’.

If  $COM_{Top Priority Apriori List} \neq 1$ , then it indicates a further scope of refinement of requirements. Developers are required to conduct workshops with the users to validate and refine unmapped requirements.

### 3.1.2. Correctness

Correctness (Gaur & Soni, 2013) signifies whether the system reflects the requirements of all the stakeholders precisely. Correctness can be termed as a reverse mapping between the requirements of “Top-Priority Survey list” to the requirements of ‘Top Priority Apriori List’.

It deals with finding out the several unmapped requirements of ‘Top-Priority Survey list’ in ‘Top Priority Apriori List’. Once endorsed would ensure the correctness of the requirements.

Equation 9 is used to ensure the correctness of requiremes:

$$CORR_{Top Priority Survey List} = \frac{Map Requirements_{Top Priority Survey List}}{TOT Requirements_{Top Priority Apriori List}} \quad (9)$$

The value  $CORR_{Top Priority Survey List} = 0$  would indicate that none of the stakeholders’ concerns have been mapped to the ‘Top Priority Apriori List’, hence ‘Top Priority Apriori List’ is considered to be erroneous to meet users’ requirements. The value  $CORR_{Top Priority Survey List} = 1$  would endorse the users’ expectations appropriately. If  $CORR_{Top Priority Survey List} \neq 1$ , then developer cannot proceed further and is required to conduct a workshop with users to validate unmapped requirements.

The process of iteratively refining the requirements is repeated till both the lists ‘Top Priority Apriori List’ and ‘Top-Priority Survey list’ are completely mapped leading to the values  $COM_{Top Priority List Apriori}$  and  $CORR_{Top Priority Survey List}$  as ‘1’.

The prioritised list of requirements, thus, obtained will not only ensure the users’ concerns based on mutual association and occurrence, but will also be close to the subjective concerns of all the stakeholders and thus will act as a baseline for the goals of final system inclining to its complete success.

The section mentioned below outlines the proposed methodologies through an exploratory ponder.

## 4. EXPERIMENTAL STUDY

An experiment was carried out to demonstrate the functionality of the recommended approach on the requirements elicitation from employees of a commercial ceramic manufacturing company. Orient Bells Pvt. Ltd. Industry was chosen as a candidate client for the experiment. The company manufactures ceramic and vitrified tiles and has its head office in Delhi and three factories spread across at Baroda (Gujarat), Sikandrabad (Uttar Pradesh) and Bengaluru (Karnatka). Around 1500 users are working in its various departments across multiple locations.

A software system envisioned to be replaced by the manual system of ‘Supply Chain Management’ will consist of different modules representing the operations of various departments viz. Inventory Management and Procurement Operations, IT Operations, Sales and Marketing Operations and Production & Quality Operations etc.

It was divided in five modules with successive functionalities to be achieved:

**Module1: Inventory Management and Procurement-** The software should be capable of managing, storing and analysing the widespread inventory and procurement of materials at all branches of the company.

**Module2: Production and Quality Operations-** The software should be capable of handling production functions in a centralized manner.

**Module3: Sales and Marketing Operations-** The software should be capable of handling Sales and Marketing functions for different branches of the company.

**Module4: IT Operations-** The software should be capable of handling IT functions like creating different users in different locations, providing them access permissions and setting passwords etc.

**Module5: Accounts and Financial Operations-** The software should be capable of handling accounting and financial functions.

For each module, various user representatives viz. Inventory Officials, Purchase Executives, Sales and Marketing executives, Production Officials, IT administrators, accounts and financial officials engaged in the company in multiple domains were identified.

This experimental study focuses only on the requirements mining of the Module1 'Inventory Management and Procurement'. Similarly, same experiment was repeated for all modules of the system.

A team of requirements analysts was made for the experiment of the proposed approach. We contacted and requested few requirements professionals from some of software development industries viz. HCL, Infosys and TCS to provide their contribution in requirements elicitation process of 'Inventory and procurement' requirements. Multiple physical and video conference sessions of interviews and brainstorming were held in different branches of ceramic industry to capture the user requirements.

The inventory officials were divided in the group of four users in all four branches of the company. A total of 80 sessions were held to obtain user-requirements. Some of the sessions were repeated iteratively for same set of users to capture left-out requirements. A large number of requirements including all trivial and non-trivial were obtained from the users working in all branches of the company.

An illustration of a session for capturing user requirements of module1 'Inventory Management and Procurement' would be in order as shown in Figure 4.

A user story interface as shown in Figure 5, was created to eliminate unnecessary stop words from the requirements and convert the requirements in brief user stories of meaningful words.

A combined experimental environment involving JSP and MySQL was used to generate the user interface. The specification of the Intel-make CPU for the experiments is as follows: 4.00 GB of random access memory with speed 2.4 GHz. and 'JBoss' web-server.

Few of the user-stories after elimination of insignificant words are listed in Table 3.

Apriori algorithm was applied on bulk of above types of user-stories. R-programming-3.4.3. was used for the experiment. The mining of user requirements was expressed in the form of association rules. Rules-metrics 'Support', 'Confidence' and 'List' were evaluated to estimate the strength of the rules. Total 206 association rules were generated for mining user requirements, the screenshot of which is shown in Figure 6. The group matrix shows all user requirements on LHS and RHS along with number of rules showing their association.

The 15 association rules with 2<sup>nd</sup> order as shown in Figure 7, were extracted having the value of 'Support>0.5'.

Following is the interpretation of the rules regarding the requirements in LHS (Left-Hand-Side) and RHS (Right-Hand-Side):

**Rule 1 & 2:** If the user requirement "Analysis of demand and supply" was captured, "Inventory analysis" was also bagged and vice versa.

**Rule 3 & 4:** "ABC analysis" also indicates the requisition of "Inventory planning" & vice-versa.

Figure 4. Inventory Session1

<b>Interview Session1:</b> <b>Module1:</b> <b>Type of Module: Inventory Management and Procurement</b>
<b>User Type:</b> Inventory Officials <b>Location:</b> Baroda
<b>Requirements analyst:</b> What do you do in managing inventory? <b>User1:</b> I make the inventory report in the company <b>User2:</b> I do inventory planning <b>User3:</b> I do inventory analysis to avoid inventory wastage <b>User4:</b> I receive and update inventory and so on..

Figure 5. User story interface

The screenshot shows a web browser window with the address bar displaying 'localhost:8889/myproject/new/main/userarea.html'. The page title is 'User Oriented View of Requirements (Supply-Chain Management)'. The main heading is 'USER STORY'. Below the heading are four buttons: 'View user story', 'View All', 'Update', and 'Delete'. On the left, there is a section 'Choose the module' with a dropdown menu showing 'Inventory & Procurement' selected. Below this are four options: 'Production & Quality Operations', 'Sales & Marketing Operations', 'IT Operations', and 'Accounts & Financial Operations'. To the right of the dropdown, there are three input fields: 'User Name' (containing 'Mr. John'), 'User Type' (containing 'Inventory Officials'), and 'Requirement' (containing 'I make the inventory report in the company'). Below these fields, there is a section 'Stop & unnecessary words to be eliminated' with a text input field containing 'I, make, the, in, the, company'. Below this, there is a section 'User Story (brief Requirement) after elimination of stop words' with a text input field containing 'Inventory report'. At the bottom, there are two buttons: 'Next Requirement' and 'Apply Apriori Algorithm'.

Table 3. Few User-stories after elimination of stop words

User Type: Inventory Officials	User Requirements Expressed in the Form of User-Stories
User1	"inventory report"
User2	"inventory planning"
User3	"inventory analysis"
User4	"receive and update inventory"



Figure 6. Group matrix to display break-up of 206 association rules for user-requirements

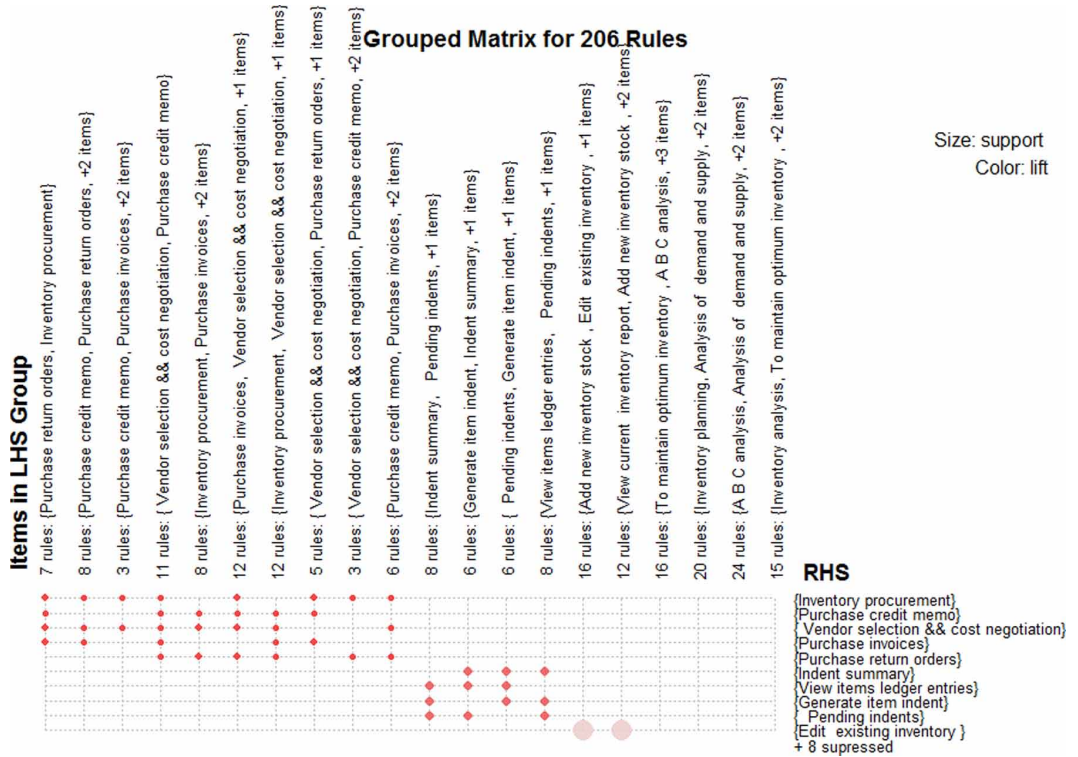


Figure 7. Top fifteen 2<sup>nd</sup> order association rules with value of Support>0.5

Console

Terminal

C:/Program Files/RSudio/datasets/

> inspect(subrules1)

	lhs	rhs	support	confidence	lift	count
[1]	{Analysis of demand and supply}	=> {Inventory analysis}	0.5132743	1.0000000	1.948276	58
[2]	{Inventory analysis}	=> {Analysis of demand and supply}	0.5132743	1.0000000	1.948276	58
[3]	{A B C analysis}	=> {Inventory planning}	0.5044248	1.0000000	1.982456	57
[4]	{Inventory planning}	=> {A B C analysis}	0.5044248	1.0000000	1.982456	57
[5]	{A B C analysis}	=> {To maintain optimum inventory }	0.5044248	1.0000000	1.982456	57
[6]	{To maintain optimum inventory }	=> {A B C analysis}	0.5044248	1.0000000	1.982456	57
[7]	{A B C analysis}	=> {Analysis of demand and supply}	0.5044248	1.0000000	1.948276	57
[8]	{Analysis of demand and supply}	=> {A B C analysis}	0.5044248	0.9827586	1.948276	57
[9]	{A B C analysis}	=> {Inventory analysis}	0.5044248	1.0000000	1.948276	57
[10]	{Inventory analysis}	=> {A B C analysis}	0.5044248	0.9827586	1.948276	57
[11]	{Inventory planning}	=> {To maintain optimum inventory }	0.5044248	1.0000000	1.982456	57
[12]	{To maintain optimum inventory }	=> {Inventory planning}	0.5044248	1.0000000	1.982456	57
[13]	{Inventory planning}	=> {Analysis of demand and supply}	0.5044248	1.0000000	1.948276	57
[14]	{Analysis of demand and supply}	=> {Inventory planning}	0.5044248	0.9827586	1.948276	57
[15]	{Inventory planning}	=> {Inventory analysis}	0.5044248	1.0000000	1.948276	57

**Rule 5 & 6:** The presence of “ABC analysis” also indicates the presence of “To maintain optimum inventory” and vice versa.

**Rule 7 & 8:** The presence of “ABC analysis” also indicates the presence of “Analysis of demand & supply” and vice versa and so on.

#### 4.1. Results and Strength Analysis of Association Rules

The Apriori algorithm applied on user-requirements yielded two priority lists. ‘Top-Priority Apriori List’ of requirements implying most desirable attributes in the software is to be implemented the

foremost, ‘Low-Priority Apriori List’ of requirements’ in contrast, is to be implemented, if at all, later. Association rules were analysed on the basis of their strength. Strong and Moderate rules led to obtain ‘Top-Priority Apriori List’ of requirements, whereas association rules with nominal, weak or NIL strength were included in the ‘Low-Priority Apriori List’. Some of the association rules that were causing redundancies were left out.

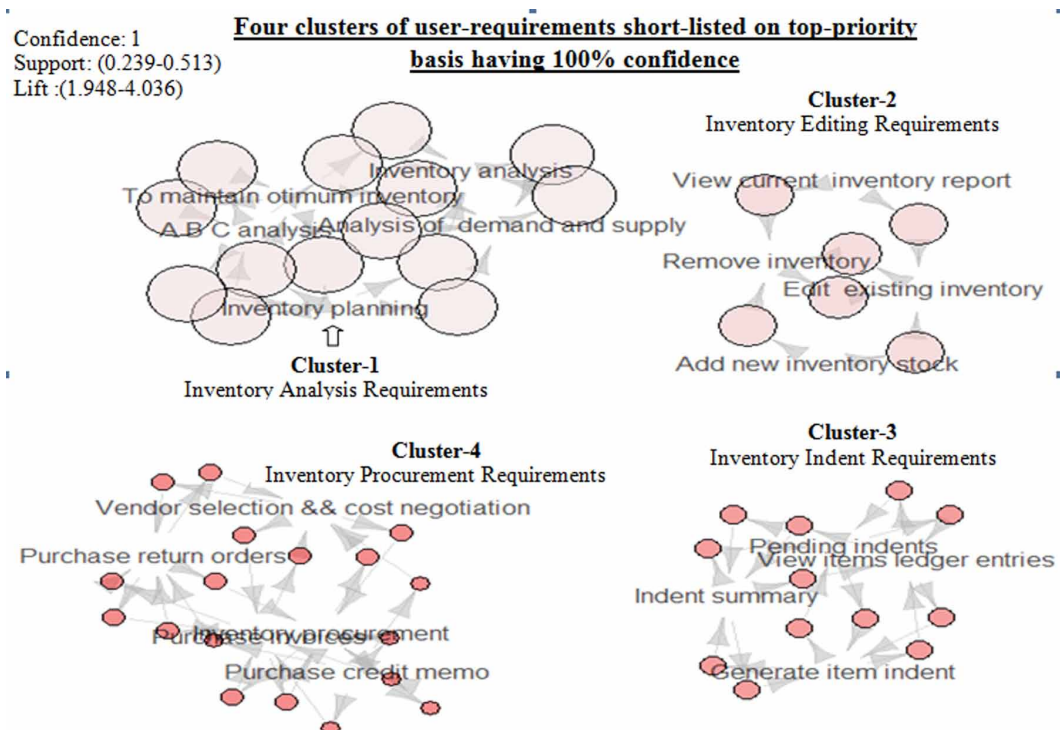
#### 4.1.1. Obtaining Top-Priority Apriori List of User-Requirements With Strong and Moderate Association

##### 4.1.1.1. Case 1 Lift >1 and Confidence=100%

In order to focus on the true and most desirable requirements of the users which are frequently occurring together, threshold limit of confidence was chosen as 100% by the result analysts-team.

The association rules with values ‘Lift>1’ and ‘Confidence=100%’ were filtered which represent those requirements of the users which are closely related to each other under the same domain and contribute in extracting clusters of user requirements occurring together. Figure 8 shows a network graph of four clusters of user-requirements obtained from frequently occurring rules with 100% confidence and higher values of Support and Lift. These clusters exhibit highly needed user requirements to be given utmost priority. The four clusters of user-requirements are ‘Inventory analysis’, ‘Inventory Editing’, ‘Inventory Indent’ and ‘Inventory Procurement’. The larger size of circles indicates the higher value of support whereas small sized circles indicate lower value of support and the arrows represent the association among different nodes. In the above graph, there are four clusters of user-requirements all having 100% confidence which indicate frequent joint occurrence of the requirements and hence the close association among the requirements of a cluster.

Figure 8. Clusters of user requirements obtained from frequently occurring rules



The cluster1 belonging to the ‘Inventory analysis’ group of requirements consists of largest sized circles which indicate the highest value of support greater than or equal to 50%. This means ‘Inventory analysis’ user-requirements belonging to the cluster1 have not only strong mutual association due to 100% confidence but also their frequency of occurrence is highest most. Therefore, these requirements will have the top most priority to be kept in the uppermost position of the ‘Top-Priority Apriori List’ of requirements.

Further, it is observed that in the graph, the size of circles is decreasing in remaining three clusters- cluster2, cluster3 and cluster4 respectively. The decreasing size of the circles in a particular cluster indicates that the size of the support is also decreasing which indicates that though requirements of these clusters are closely related to each other due to having 100% confidence but the frequency of occurrence of these clusters is comparatively low.

Such kinds of requirements with 100% confidence and value of support below 50%, will have ‘moderate association’ and will be placed in the ‘Top-Priority Apriori List’ but at the lower positions in decreasing order of their value of support.

Table 4 enumerates the requirements of ‘Top-Priority Apriori List’ along with the values of support and type of association.

**Table 4. Top priority apriori list of requirements (with 100% confidence)**

Clusters of Requirements in Descending Order of Their Priority	User Domain	User Requirements	Value of Support	Type of Association
Cluster1	Inventory Analysis	<ul style="list-style-type: none"> <li>• Inventory Analysis</li> <li>• To maintain optimum inventory</li> <li>• ABC inventory analysis</li> <li>• Inventory Planning</li> <li>• Analysis of demand and supply</li> </ul>	Greater than or equal to 50%	Strong
Cluster2	Inventory Editing	<ul style="list-style-type: none"> <li>• Add new inventory stock</li> <li>• View current inventory report</li> <li>• Edit existing inventory</li> <li>• Remove inventory</li> </ul>	In between (40%-50%)	Moderate
Cluster3	Inventory Indent	<ul style="list-style-type: none"> <li>• Generate item indent</li> <li>• Pending indents</li> <li>• Indent summary</li> <li>• View items ledger entries</li> </ul>	Less than 40%	Moderate
Cluster4	Inventory Procurement	<ul style="list-style-type: none"> <li>• Inventory procurement</li> <li>• Vendor selection &amp; cost negotiation</li> <li>• Purchase invoices</li> <li>• Purchase return orders</li> <li>• Purchase credit memo</li> </ul>	Less than 40%	Moderate

Remaining requirements with Lift>1 and Confidence<100% were placed in the top positions of ‘Low Priority Apriori List’ of requirements.

#### **4.1.2. Obtaining Low-Priority Apriori List of User-Requirements With Nominal, Weak and NIL Association**

Following is the description of obtaining user requirements with ‘nominal’, ‘weak’ and ‘NIL’ association.

#### 4.1.2.1. Case 2. Value of Lift>1 but Confidence<100%

The association rules with 2nd order having ‘Lift>1’ and ‘Confidence’ <100%’ were generated as shown in Figure 9. The requirements obtained from LHS and RHS of such rules were said to have nominal relation and were placed on top positions of ‘Low-Priority Apriori List’.

Figure 9. Low-priority apriori list with value of lift greater than 1 and confidence less than 100%

	lhs	rhs	support	confidence	Lift	Count
[1]	Vendor negotiation for secondary goods	=> in-house process for buying secondary goods	0.3001335	0.6710239	1.5134618	3
[2]	Vendor negotiation for acquiring raw materials	=> List out monthly vendors report	0.5331233	0.7711203	1.4122367	3
[3]	Procure packaging items	=> Prepare purchase order for packaging items	0.2001230	0.8711231	1.5134618	3
[4]	Display vendor settlement for packaging items	=> in-house process for buying packaging items	0.4034210	0.8711233	1.2412317	2
[5]	Appraising vendors performance	=> List out vendors with lowest cost	0.5331231	0.7711203	1.4122367	3
[6]	Procure consumable items	=> Prepare purchase order for consumable items	0.3001232	0.7711203	1.5134618	3
[7]	Display vendor settlement for consumable items	=> in-house process for consumable items	0.2536213	0.6711233	1.2412317	2
[8]	Develop vendors for supplying consumable items	=> Monthly report of consumable items	0.4331231	0.7711233	1.4122367	3
[9]	Procure fuel	=> Prepare purchase order for fuels	0.2403331	0.6711233	1.5134618	3
[10]	Display vendor settlement for fuels	=> in-house process for procuring fuels	0.2334216	0.7711203	1.2412317	2
[11]	Develop vendors for supplying fuels	=> Monthly report of consumable fuels	0.4331236	0.8711231	1.4122367	3
[12]	Procure raw materials	=> Prepare purchase order for raw materials	0.2201237	0.8711231	1.5134618	3
[13]	List out vendor settlement for materials	=> in-house process for buying materials	0.6034211	0.6711233	1.2412317	2

#### 4.1.2.2. Case 3. Value of Lift=1

The user-requirements which are occurring in isolation will have ‘Lift=1’ and low value of support as well as ‘Confidence’ and hence will have the least priority to be given by the developer.

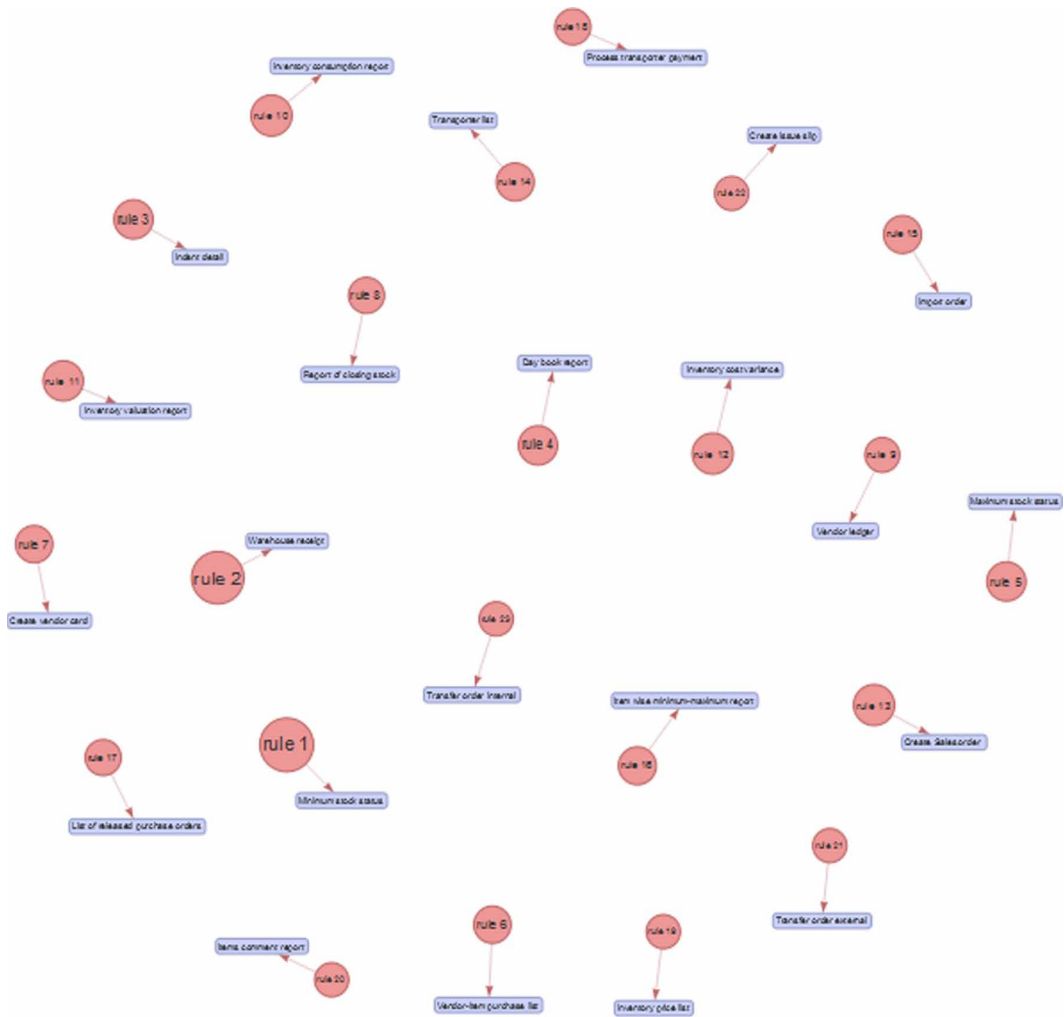
The value of ‘Support’ and ‘Confidence’ for such requirements will always be equal. Such requirements will be taken care by the developer in last in descending order of the value of ‘Support’. These requirements are having no parameter in left hand side (lhs). These requirements are occurring in right hand side(rhs) in the association rules. Figure10 shows all such requirements appearing in isolation. All these low-priority requirements have their value of support and confidence in the range (0-0.1). Figure 11 shows a scatter plot of all these least important mutually independent user-requirements.

Figure 10. Low priority mutually-independent user-requirements

	lhs	rhs	support	confidence	lift	count
[1]	{}	=> {}	0.001574803	0.001574803	1	1
[2]	{}	=> {Transfer order internal}	0.011023622	0.011023622	1	7
[3]	{}	=> {Transfer order external}	0.014173228	0.014173228	1	9
[4]	{}	=> {Inventory price list}	0.017322835	0.017322835	1	11
[5]	{}	=> {Process transporter payment}	0.020472441	0.020472441	1	13
[6]	{}	=> {List of released purchase orders}	0.022047244	0.022047244	1	14
[7]	{}	=> {Import order}	0.026771654	0.026771654	1	17
[8]	{}	=> {Inventory valuation report}	0.028346457	0.028346457	1	18
[9]	{}	=> {Inventory consumption report}	0.029921260	0.029921260	1	19
[10]	{}	=> { vendor ledger}	0.029921260	0.029921260	1	19
[11]	{}	=> {Report of closing stock}	0.033070866	0.033070866	1	21
[12]	{}	=> {Create vendor card}	0.037795276	0.037795276	1	24
[13]	{}	=> {transporter list}	0.037795276	0.037795276	1	24
[14]	{}	=> {Vendor-item purchase list}	0.039370079	0.039370079	1	25
[15]	{}	=> {Maximum stock status}	0.045669291	0.045669291	1	29
[16]	{}	=> {Indent detail}	0.047244094	0.047244094	1	30
[17]	{}	=> {Create issue slip}	0.050393701	0.050393701	1	32
[18]	{}	=> {Create Sales order }	0.051968504	0.051968504	1	33
[19]	{}	=> {Items comment report}	0.056692913	0.056692913	1	36
[20]	{}	=> {Item wise minimum-maximum report}	0.062992126	0.062992126	1	40
[21]	{}	=> { Day book report}	0.077165354	0.077165354	1	49
[22]	{}	=> {warehouse receipt}	0.080314961	0.080314961	1	51
[23]	{}	=> {Minimum stock status}	0.086614173	0.086614173	1	55
[24]	{}	=> { Inventory cost variance}	0.091338583	0.091338583	1	58



Figure 11. Scatter plot of low priority requirements having no mutual association



#### 4.1.2.3. Case 4. Value of Lift<1

Requirements with negative co-relation will be placed on middle positions of 'Low-Priority Apriori List' in descending order of their 'Support'. These requirements with weak relation as listed in Figure 12, occur together occasionally.

The comprehensive record of low-priority requirements resulted after applying the proposed methodology are cited in Table 5.

So obtained segregation of user-requirements in 'Top Priority Apriori List' and 'Low Priority Apriori List' will incorporate the prime concerns of all users.

## 4.2. Endorsing Completeness and Correctness of User Requirements

To validate the correctness and completeness of so obtained 'Top Priority Apriori List' and 'Low Priority Apriori List' with respect to skilled and highly experienced stakeholders' expectations, gap analysis was applied.

Figure 12. Association rules with value of lift less than 1 (requirements with weak relation)

	lhs	rhs	support	confidence	Lift	Count
[1]	Import and export of good	=> Appoint inventory staff	0.2001231	0.1331221	0.3334111	2
[2]	Appoint inventory staff	=> Create sales order	0.1001231	0.1611161	0.5134618	3
[3]	Expand vendor list	=> Create sales order	0.2034210	0.1879132	0.2412317	2
[4]	Negotiate cost	=> Sales auditing	0.4331230	0.2349103	0.4122367	3
[5]	Appoint inventory staff	=> List Marketing strategies	0.0176354	0.3457113	0.2222353	3

A survey was done. Four to six stakeholders from the top hierarchy of Inventory Department of all three branches of Orient Bells Pvt. Ltd. were chosen who were having their work experience of more than 15 years. Online questionnaire as shown in Figure 13, was prepared and the stakeholders were asked to feed their view-points for the priorities of the requirements using linguistic terms viz. “Highly Important”, “Moderately Important”, “Nominally Important”, “Weakly Important”.

Equations (4) and (5) were applied to map the fuzzy view-points of all stakeholders to their crisp-values and henceforth the final priorities of requirements were obtained by calculating the arithmetic mean of their respective crisp-priorities. This resulted in two requirements lists: ‘Top-Priority Survey List’ and ‘Low-Priority Survey List’.

The gap analysis was applied to bridge the gap between ‘Top-Priority Apriori List’ and ‘Top-Priority Survey List’. The analysis is shown in Figure 14. Same process was applied for low priority list of requirements.

Forward and backward mapping was applied on the requirements of both lists ‘Top-Priority Apriori’ and ‘Top-Priority Survey’. The performance metrics ‘Completeness’ and ‘Correctness’, as mentioned in the proposed approach were applied.

The requirements of ‘Top-Priority Apriori List’ were mapped in forward direction to the requirements of ‘Top-Priority Survey List’. Number of links mapped was found to be 15 leading to the value of completeness as 60%, whereas the remaining requirements viz. ‘Remove Inventory’, ‘Pending Indents’, ‘View items ledger entries’ and ‘Purchase return orders’ were found to be missing in ‘Top-Priority Survey List’.

During the backward forwarding, 13 requirements of ‘Top-Priority Survey List’ were mapped to the requirements of ‘Top-Priority Apriori List’ leading to the value of correctness as 72.22%.

It was observed the some of the additional requirements viz. ‘Inspecting sales patterns’, ‘Computerised inventory tracing’ and ‘Automation of Vendor-analysis’ were found in ‘Top-Priority Survey List’ whereas missing in ‘Top-Priority Apriori List’.

To fulfil the gap in the correctness and completeness of requirements and bring the requirements close to the ‘Top-Priority Survey List’, workshops were required to be held with an integrated team of users and stakeholders. The process was iteratively executed till all requirements were refined and all users and stakeholders were satisfied leading to the values of completeness and correctness as 1.

The same process was extended for rest of all modules of “Supply Chain Management”.

### 4.3. Performance Analysis

The proposed technique was executed for different sets of user requirements and execution time was recorded for various requirements-sets by subtracting the beginning and culmination time of the program code. In order to observe the performance of the proposed approach, two most popular methods named AHP (Goh & Kok, 2010) and Integrated Prioritization Approach (Dabbagh & Lee, 2014) were chosen and executed for the same sets of requirements. The execution time recorded for these two techniques including the proposed approach is shown in Figure 15. It was observed that for

**Table 5. Low-priority Apriori list of requirements**

<b>Lift&gt;1 With Confidence&lt;100%</b>			
<b>User Requirements</b>	<b>Support</b>	<b>Confidence</b>	<b>Type of Association</b>
List out vendor settlement for materials in-house process for buying materials	0.6034211	0.6711233	Nominal
Vendor negotiation for acquiring raw materials List out monthly vendors report	0.5331233	0.7711203	Nominal
Appraising vendors performance List out vendors with lowest cost	0.5331231	0.7711203	Nominal
Develop vendors for supplying fuels Monthly report of consumable fuels	0.4331236	0.8711231	Nominal
Develop vendors for supplying consumable items Monthly report of consumable items	0.4331231	0.7711233	Nominal
Display vendor settlement for packaging items in-house process for buying packaging items	0.4034210	0.8711233	Nominal
Vendor negotiation for secondary goods in-house process for buying secondary goods	0.3001335	0.6710239	Nominal
Procure consumable items Prepare purchase order for consumable items	0.3001232	0.7711203	Nominal
Display vendor settlement for consumable items in-house sourcing for consumable items	0.2536213	0.6711233	Nominal
Procure fuel Prepare purchase order for fuels	0.2403331	0.6711233	Nominal
Display vendor settlement for fuels in-house sourcing for procuring fuels	0.2334216	0.7711203	Nominal
Procure raw materials Prepare purchase order for raw materials	0.2201237	0.8711231	Nominal
Procure packaging items Prepare purchase order for packaging items	0.2001230	0.8711231	Nominal
<b>Lift&lt;1</b>			
Appoint inventory staff, List marketing strategies	0.3457113	0.0176354	Weak
Negotiate cost, Sales auditing	0.2349103	0.4331230	Weak
Expand vendor list, create sales order	0.1879132	0.2034210	Weak
Import and export of goods	0.1331221	0.2001231	Weak
<b>Lift=1</b>			
<b>Confidence=Support for Requirements-in-isolation</b>			
Inventory cost variance Minimum stock status Warehouse receipt	In between (0.08-0.1]		NIL
Day book report	In between (0.07-0.08]		NIL
Item wise minimum-maximum report	In between (0.06-0.07]		NIL
Items comment report Create sales order Create issue slip	In between (0.05-0.06]		NIL
Indent detail Maximum stock status	In between (0.04-0.05]		NIL
Vendor-item purchase list Transporter list Create vendor card Report of closing stock	In between (0.03-0.04]		NIL
Vendor ledger Inventory consumption report Inventory valuation report Import order List of realeased purchase orders Process transporter payment	In between (0.02-0.03]		NIL
Inventory price list Transfer order external Transfer order internal	In between (0.01-0.02]		NIL

Figure 13. Questionnaire (Stakeholder-View Point)

Office Memo by Free CSS Tem

### Stakeholder-View Point Requirements-Validation Requirement-Priority Feedback (Supply-Chain Management)

**Choose module-category from Drop-Down Menu**

Inventory & Procurement

- Production & Quality Operations
- Sales & Marketing Operations
- IT Operations
- Accounts & Financial Operations

**Choose requirement-category**

Top-Priority List of Requirements

- Top-Priority List of Requirements
- Low-Priority List of Requirements

**Stakeholder Name** Mr. Sonnet **Work Experience** 20 years

**Designation** General Manager

**Choose the requirement from Drop-Down menu**

User-Requirement
Inventory Analysis

- Inventory Analysis
- To maintain optimum inventory
- ABC inventory analysis
- Inventory Planning

**Choose the Requirement-Importance from Drop-down menu**

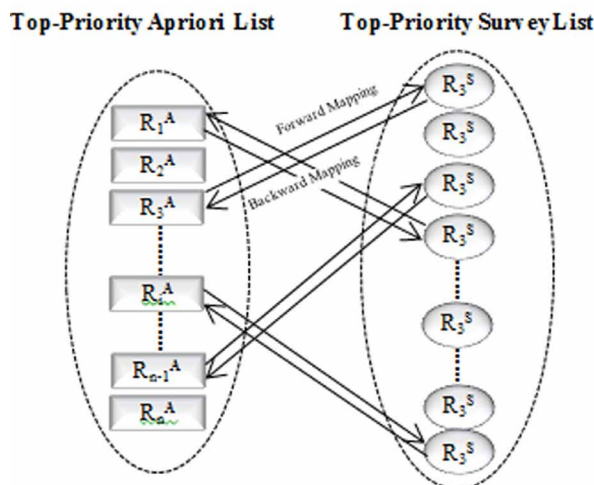
Enter Requirement-Priority in Linguistic term
Highly Important

- Highly Important
- Moderately Important
- Nominally Important
- Weakly Important

View modules View Requirements ADD Update Delete View Requirement-Priorities Submit Data

Figure 14. Gap analysis to validate Top-Priority Apriori List and Top-Priority survey list

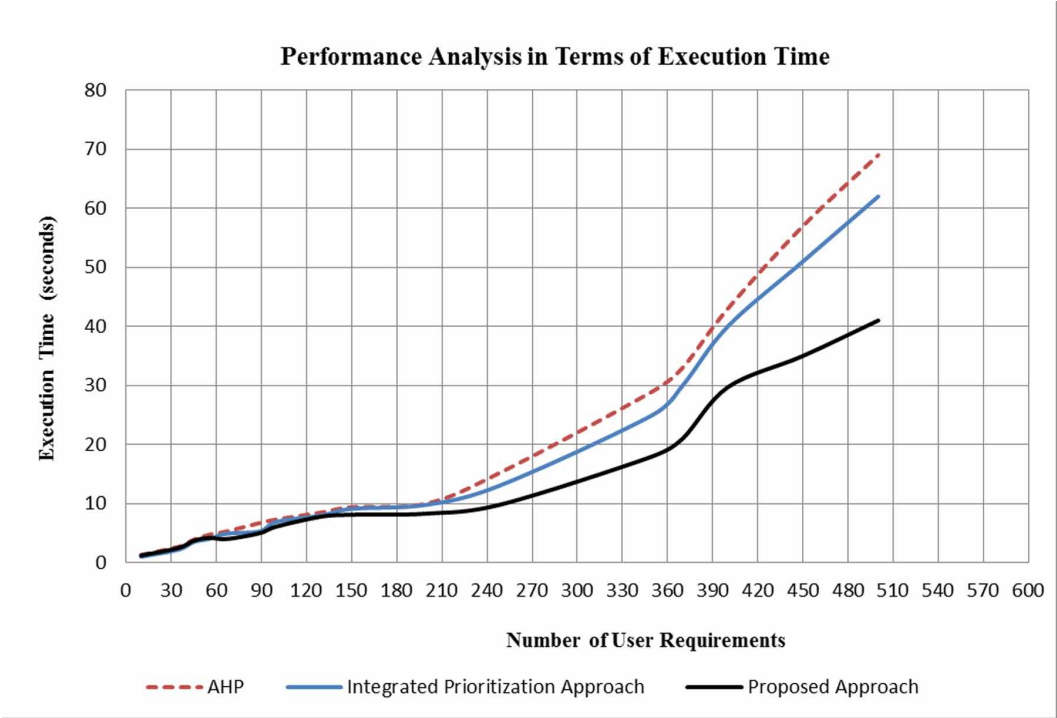
### Gap-Analysis



Number of mapped links in forward direction: 15  
 Number of mapped links in backward direction: 13  
 Completeness: 60%  
 Gap in obtaining completeness: 40%  
 Correctness: 72.22%  
 Gap in obtaining correctness: 27.78%



Figure 15. Performance analysis



small sets of requirements, the execution time of all three approaches is almost same. But as the number of requirements increase, the increase in the execution time of the proposed approach is significantly lesser than the increase in the execution time of AHP and Integrated Prioritization Approach.

## 5. A COMPARATIVE ANALYSIS OF REQUIREMENTS PRIORITIZATION TECHNIQUES

Requirements prioritization is an indispensable and vital but stimulating pursuit in software development. Various requirements prioritization practices testified in literature (Goh & Kok, 2010; Dabbagh & Lee, 2014; Santos & Albuquerque, 2016; Minas, Asif, & Borstler, 2019) own their key characteristics. “Analytical Hierarchy Process” (AHP) (Goh & Kok, 2010) is a choice-making technique that acts on the concept of matching every two requirements pair wise with respect to various parameters. It works efficiently in decision making with respect to multiple criteria and objectives, however, it is restricted to small projects due to large number of pair-wise comparisons involved. “Towards the Applied Hybrid Model” (Santos & Albuquerque, 2016) envisages market-driven features for requirements-prioritization whereas “Mining and Prioritization of Association Rules” (Ait-Mlouk, Agouti, & Gharnati, 2017) is capable of dealing with big data at the quick response time, however, it does not cope with real time and fuzzy data. “RFP based Requirement Prioritization” (Krishnan, 2018) is a simple-solution model that guarantees the choice of reasonable and best prioritization strategy based on the needs of a client, however, it is at its primitive junctures for conferring ample points of interest. “Verbal Decision Analysis” (Barbosa et al., 2019) facilitates prioritization of requirements with respect to goals, constraints and available budget, however, it is quite challenging due to requirements-elicitation based on verbal factors. “Stakeholders’, Weight,

Vote and Priority” (SWVP) (Minhas, 2019) is a basic voting based approach which strengthens the issues of communication and coordination among stakeholders during requirements-prioritization.

An evaluation of certain requirements prioritization methods prevalent in the literature including the proposed approach was done. The weaknesses and strengths of these methodologies along with several key parameters viz. author and publisher details, key-concepts, performance analysis, fuzzy concerns and automation are presented in Table 6.

It was observed that all requirements prioritization techniques have their own prominent features accompanied with some weaknesses. However, the proposed approach provides a number of following benefits over the cited methods:

- It is able to capture clear patterns of user requirements in the form of their mutual association and frequent occurrence and hence represents the true needs of the users. It is an automated approach that excavates the knowledge from user requirements in the form of association rules. Due to automation, it is less prone to misperception and flaws.
- It is capable of producing prioritized lists of huge number of complete and correct user-requirements in significantly lesser time.

## **6. CONCLUSION**

Correct and complete mining of user requirements is an important precedent to software development. The paper demonstrates that a multistage methodological approach based on Apriori algorithm can yield duly validated prioritized list of desired attributes from the vaguely and variously expressed user-requirements. The methodological approach displayed here may be a clear change over the extant approaches.

## **7. CHALLENGE AND FUTURE WORK**

This approach captures the clear patterns in the user-requirements. The requirements overlapping in multiple groups and with vague patterns need to be treated cautiously. Such requirements-forecasting is simultaneously a science and an art evolving from the rules of thumb to the rule metrics. An aspect that needs to be considered here is the possibility of incorporating dynamic capabilities in the software that adapt to modification in the user stories on real-time basis. Bordering on artificial intelligence, Apriori algorithm applied on user requirements needs to be extended to real time requirement algorithms.

## REFERENCES

- Krishna, A., & Lu, H. (2008). Requirement elicitation using goal-based organizational model. In *ASWEC '08 Proceedings of the 19th Australian Conference on Software Engineering*. IEEE.
- Gaur, V., Soni, A., & Bedi, P. (2010). An Agent-Oriented Approach to Requirements Engineering. In *Proceedings 2010, IEEE 2nd International Advance Computing Conference*. IEEE Press.
- Gaur, V., & Soni, A. (2010). An Integrated Approach to Prioritize Requirements using Fuzzy Decision Making. *International Journal of Engineering and Technology*, 2(4).
- Amaral, L. M. G., & Faria, J. P. (2010). A Gap Analysis Methodology for the Team Software Process. *Proceedings of the Seventh International Conference on the Quality of Information and Communications Technology*, 424-429. doi:10.1109/QUATIC.2010.78
- Wang, P., Shi, L., Bai, J., & Zhao, Y. (2009). Mining Association Rules Based on Apriori Algorithm and Application. In *International Forum on Computer Science-Technology and Applications*. IEEE.
- Goh, H. H., & Kok, B. C. (2010). Application of Analytic Hierarchy Process (AHP) in load shedding scheme for electrical power system. In *9th International Conference on Environment and Electrical Engineering (EEEIC)*. IEEE. doi:10.1109/EEEIC.2010.5489942
- Farfeleder, S., Moser, T., Krall, A., Stalhane, T., Omoronyia, I., & Zojer, H. (2011). Ontology-Driven Guidance for Requirements Elicitation. In *8th Extended Semantic Web Conference, LNCS 6644*, 212–226.
- Macasaet, R., Chung, L., & Garrido, J. L. (2011). An agile requirements elicitation approach based on NFRs and business process models for micro-businesses. In *The 12th International Conference on Product Focused Software Development and Process Improvement*. ACM.
- Lim, S., L., & Finkelstein, A. (2012). Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation. *IEEE Transactions on Software Engineering*, 38(3), 707-735.
- Gaur, V., & Soni, A. (2013). A Fuzzy Traceability Vector Model for Requirements Validation. *International Journal of Computer Applications in Technology*, 47(2/3).
- Dabbagh, M., & Lee, S. P. (2014). An Approach for Integrating the Prioritization of Functional and Non-Functional Requirements. *The Scientific World Journal*.
- Przybylek, A. (2014). A business-oriented approach to requirements elicitation. In *9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*. IEEE.
- Jiang, W., Ruan, H., Zhang, L., Lew, P., & Jiang, J. (2014). For User-Driven Software Evolution Requirements Elicitation Derived from Mining Online Reviews. *Lecture Notes in Computer Science*, 8444, 584–595.
- Valvas, S., & Milani, F. (2015). Requirement Elicitation Using Business Process Models. *Lecture Notes in Business Information Processing*, (229), 67-81.
- Jin, X., Donzé, A., Deshmukh, J. V., & Seshia, S. A. (2015). Mining Requirements from Closed-Loop Control Models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(11), 1704–1717. doi:10.1109/TCAD.2015.2421907
- Santos, R., & Albuquerque, A. (2016). Towards the Applied Hybrid Model in Requirements Prioritization. *Information Technology and Quantitative Management, Procedia Computer Science*, 909-918.
- Karras, O., Kiesling, S., & Schneider, K. (2016). Supporting Requirements Elicitation by Tool-Supported Video Analysis. *IEEE 24th International Requirements Engineering Conference*, 1-10.
- Kushiro, N., Shimizu, T., & Ehira, T. (2016). *Requirements Elicitation with Extended Goal Graph*. *Procedia Computer Science*, 96, 1691–1700.
- Lian, X., Rahimi, M., Huang, J. C., & Zhang, L. (2016). Michael Smith, Mining Requirements Knowledge from Collections of Domain Documents. *IEEE 24th International Requirements Engineering Conference*, 1-10.
- Mishra, D., Aydin, S., Mishra, A., & Ostrovska, S. (2017). Knowledge management in requirement elicitation. *Computer Standards & Interfaces*, 49-61.

Wong, L. R., Mauricio, D., & Rodriguez, G. D. (2017). A Systematic Literature Review about Software Requirements Elicitation. *Journal of Engineering Science and Technology*, 12(2), 296–317.

AlMousa, M., Al-Khalifa, H., S., & AlSobayel, H. (2017). Requirements Elicitation and Prototyping of a Fully Immersive Virtual Reality Gaming System for Upper Limb Stroke Rehabilitation in Saudi Arabia. *Mobile Information Systems*.

Ait-Mlouk, A., Agouti, T., & Gharnati, F. (2017). Mining and Prioritization of Association Rules for Big Data: Multi-Criteria Decision Analysis Approach. *Journal of Big Data*, 4(1), 42. doi:10.1186/s40537-017-0105-4

Manzoor, M. I., Shaheen, M., & Khalid, H. (2018). Requirement Elicitation Methods for Cloud Providers in IT Industry. *I.J. Modern Education and Computer Science*, 10, 40–47.

Krishnan, M. S. (2018). RFE based Requirements Prioritization –A One Step Solution. *Materials Today: Proceedings*, 5(1), 642–649.

Barbosa, P. A. M., Pinheiro, P. R., & Silveira, F. R. V. (2019). *Selection and Prioritization of Software Applying Verbal Decision Analysis*. Hindawi Complexit.

Minas, N. M., Asif, M., & Borstler, J. (2019). SWVP-A Requirements Prioritization Technique for Global Software Development. In *45th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE. doi:10.1109/SEAA.2019.00010

## APPENDIX

Table 6. A comparative analysis of various requirements prioritization methods

Requirements Prioritization methods	Year	Publisher	Author	Concepts	Support for		Automation Tool	Weaknesses	Strengths
					Performance Analysis	Fuzzy Concerns			
AHP	2010	IEEE	Goh, H. H.	Pair-Wise Comparison, Electrical Power System	Missing	Missing	Missing	Restricted to small projects due to overheads of pair-wise comparisons	Works efficiently in decision making with respect to multiple criteria and objectives
Integrated Prioritization Approach	2014	Hindawi	Dabbagh	Functional & Non-Functional Requirements, Decision Matrix	Execution time	Linguistic Terms	Microsoft Visual Studio 2008, Net Framework 3.5, C# Programming, Microsoft SQL Server	Limited to small ventures with large number of pair-wise comparisons	Supports prioritization of both functional and non-functional requirements
Towards the Applied Hybrid Model	2016	Elsevier	Santos, R.	Hierarchical Cumulative Voting, HCV Points, Macbeth	Benefit, competition and volatility analysis	Missing	Partially in the form of Web-based questionnaire	It seems to be in its early phases for imparting adequate operative details	Deals with the market-driven features in requirements-prioritization
Mining & Prioritization of Association Rules	2017	Springer	Ait-Mlouk, A.	PFP Growth, Big data, Apache Spark, Multi-Criteria Decision Analysis	Quality measurement of association rules	Missing	Apache Spark	It should be strengthened to process real time and fuzzy data	Capable of dealing with big data at the quick response time
RFP based Requirement Prioritization	2018	Elsevier	Krishnan, M., S.	Transitional Process-Flow Model	Missing	Missing	Illustrative case study but automation of the approach is missing	It is at its primitive junctures for conferring ample points of interest	Simple-solution model that guarantees the choice of reasonable and best prioritization strategy based on the needs of a client
Verbal Decision Analysis	2019	Hindawi	Barbosa, P., A., M.	Search based S/W Engineering, ORClass, VDA	Cost & Budget	Missing	Instance generator, ORclass Web, Aranau Tool	This approach is quite challenging due to requirements-elicitation based on verbal factors	This work facilitates to prioritize the requirements with respect to goals, constraints and available budget.
SWVP	2019	IEEE	Minhas, N., M.	SWVP,	Weight-vote-priority	Linguistic Terms	PHP, MySQL	A basic voting based approach confined to little measure ventures	This technique emphasizes and strengthens the issues of communication and coordination among stakeholders
Proposed Approach		Communicated		User-Story, Apriori Technique, Gap analysis, Survey Analysis	Strength Analysis: Association-Rules Metrics, Gap analysis: 'Completeness' and 'Correctness', Performance Analysis: Execution Time	Triangular Fuzzy Numbers	Automation support using JSP, MySQL, JBoss Web-Server & R-Programming	Relatively challenging to capture dubious and overlapping patterns in user-requirements	It settles two prime concerns of Requirements Engineering while using single approach. One, it supports knowledge-driven elicitation of user requirements; Second, it enables mutual association and occurrence-driven prioritization of requirements.

*Anuja Soni (PhD) is an Assistant Professor (Sr. Scale) in Deen Dayal Upadhyaya College, University of Delhi and has about 20 years of teaching experience. She has done her Ph.D. from University of Delhi, Delhi, India. She has published several papers in reputed journals and international conferences. She has organized multiple seminars and workshops in the field of computer science at college and national level. Her research interests include Requirements Engineering, Software Engineering, Neural Networks, Artificial Intelligence, Fuzzy Logic, Multi-Agent System, Data Mining, Soft Computing, and Machine Learning.*

*Anand Saxena (PhD) is an alumnus of University of Delhi. Currently he is engaged in teaching undergraduate courses at Deen Dayal Upadhyaya College (where he is a resident faculty as an Associate Professor) and a course in Management & Strategy at Department of Financial Studies, University of Delhi South Campus (where he serves as a Visiting Faculty). His current teaching and research interests include MSME Entrepreneurship and Innovation, Business Ethics, Meaningful Organizations and Corporate Governance. He has developed content for Entrepreneurship, Ethics and Management Education. He is a proponent of the ideas of Conscientious Commerce, All Walks of Life (AWL) Entrepreneurship and Organizations as Enactments of Harmony. He is a NIESBUD-accredited entrepreneurship trainer and as such serves as a resource person for Entrepreneurship Awareness and Entrepreneurship Development Programmes. Besides he is associated with faculty development programmes for university and school teachers.*

*Parul Bajaj is an Economics teacher. She obtained her MA in economics from MDU in 2014. She is currently working as economics teacher at DAV PUBLIC SCHOOL SEC-14 GURUGRAM. She has a teaching experience of about 7 years. She has a deep knowledge and passion for the subject matter. She has ability to build community and strong relationships with students. Her research interests centre around the field of agriculture economics specifically in the daily routine of farmers and their shifting role in relation to industrialisation, urbanisation, etc. Also, in determining the impact of reducing interest of the farmers in their ongoing profession due to above factors on the future of agriculture growth and development in context of Indian economy.*