International Journal of Information Technology and Knowledge Management
December 2012, Volume 6, No. 1, pp. 89-92

DISCOVERING SOFTWARE REUSES PATTERNS AND SCALING REQUIREMENT USING DATA MINING

Manisha Shinde*

Abstract: A data Mining and Data Warehousing implementation strategy for software reuse is an integrated, time variant collection of data about software components, requirements. It aims at supporting project decisions and logical continuation of data warehousing for mining knowledge about software requirements and reuse components decisions and for software objective settings for system management.

By applying proper filtering, clustering and analysis under technical supervision of complex, unstructured demands, the researcher can predict set of precise, correct and optimized set of requirements in data marts to get logical classified patterns as essential, desirable and deferred. This will help to design generic, template-based, set of software goals and to prioritize requirements for better software project management.

Keywords: Data Warehouse, Data Mining, Data Mart, Software Reuse, Patterns, Requirement Prioritization. Software project management

1. INTRODUCTION

Software projects requirements are highly complex to anticipate and involve numbers of different stakeholders specifying requirements. Sometimes unstructured and semi structured data becomes complex and vague; to restrict it for transaction processing and for further references, it is important to have proper structure. The key to successful software development has always been Risk Reduction and Accurate requirement framing from stakeholders communication and with Customer Minds. By segmenting clusters of the collected requirements for classification of similar kind of needs of stakeholders, more precise collections to get homogeneous requirements can be done. From this set software analyst can derive software reuse patterns. Furthermore by having priority group clusters, objectives and scope of software projects may be defined more precisely, specifically and very rightly.

By considering above points, Software Component data mart can be established and by applying data mining algorithmic methods to this mart one will discover software reuse pattern to use in applications: Requirements decision setting for software reuse, Requirement Classification, Categorization, Resource Allocation and Scheduling, Requirement Analysis, Software Configuration Management, Testing Inferences.

2. METHODOLOGY

The researcher would like to explore, if by applying data mining method to software components mart as shown in **Table No. 1** to discover software reuse patterns, more refined approach can be applied to identify, develop and to get generic components. Patterns once established can be easily grasped and implemented often to get more appropriate and bug free solutions.

Data warehousing is used:

1. To provide powerful and prompt decision making tool.
2. To reduce amount of resources, time and manpower.

Table No 1

Data Mart	*Applications*
System Management.	1. Requirements decision setting for software reuse. 2. Requirement Classification, Categorization, 3. Resource Allocation and Scheduling. 4. Requirement Analysis. 5. Software Configuration Management. 6. Testing Inferences.

Category Variables:

- Source- Employees/Operators/Managers/CEOs.
- Zone/Area- Head Office/ Branch/ Department.
- Performance of separate reports, documents, codes, designs.
- Hardware and software requirements.

* Faculty of Management (Information Technology), Vidyapeeth University, Pune, IMRDA, Sangli, *E-mail: mjs.imrda@gmail.com, alishallsept@gmail.com*

Table No. 2

Sr. No.	*Behavior for Clustering and Supervising*	*Predicted Patterns*
1	More Similar kind of needs those are generally required in most of the systems	Generic Requirement
2	As need of performance depending on time and functioning parameters	Parameterized Requirements
3	Need of specialization of functioning and exclusive high performance	Specific Requirements

Requirement Management involves elicitation, verification, validation, modeling, classification of requirements; all these things are done on analytic base. Technically it is very difficult to clearly specify and analyze set of requirements; it needs expertise knowledge and preciseness to take decisions about requirements. For the same reason and to get similarities that is to get generic requirements as standard set with all basic needs expert system can help to analyst.

A data warehouse constructed from integrated data source systems does not require ETL, staging databases, or operational data store databases. The integration from of elicitation process involves existing systems, stakeholders demands, goals, current tools, business processes, day to day transactions are integrated as data source system and then through its systematic collaboration and clusters are used to track life, origin and scope of requirements during discovering patterns of requirements.

Employing cross-sectional data to guide in selection and decision of requirement for often situations of time, leads in similar kind of set of behavior/functionality of requirement. It helps to carefully identify and select authorized, collaborative, knowledgeable, committed sub groups for cross-sectional analysis. Collaborative representatives will be simply categorized in generic roles by having multiple metrics for analysis.

As shown in figure 1 one can design data ware house extracting data and by applying analysis aspects to different data clusters. All possible sources can be integrated together in forums or nearly related clusters. Requirements generalization can be confirmed studying similar behavior or kind of nature of requirements. If other metric dimensions are applied one can get time or condition-based variant parameterized requirements. Or with more specialization one can have system and domain specific set for special purpose modules.

Generic requirements patterns will not only help to reuse software in application developments but also mainly it can be applied to get group of similar kind of requirements to decide and design software packages, namespaces while designing software development tools and technologies. For example mostly software needs authentication and security for software transactions, one can design generic authentication modules, functions and objects and it can be made available in form of factories included in namespaces, classes or library files.

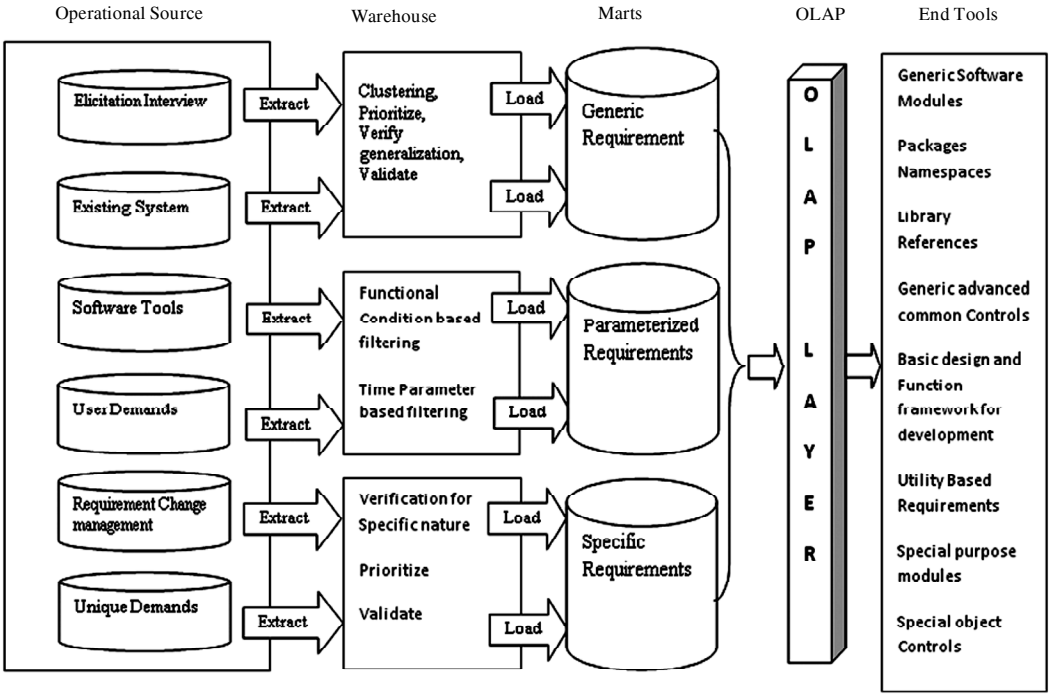

Figure 1: Data Marts to Get Pattern

Similarities and generic patterns can be very useful to develop pre-written Interfaces with abstract methods and template designs.

In next staging metric based filtering and analysis of data can predict another mart area of parameter dependent requirements can be classified as Parameterized Requirements. These requirements can help to develop utility based requirements. It can help to design data structures and related events controlling to those structures. Basic parameterized framework or utility based requirements can form factory or building blocks of different functional needs.

By Verifying specific nature, researcher can prioritize and validate requirements to get specific requirements. Specific requirements can help to design and develop special purpose modules or special object controls. All specific transactions can be refined separately and can be concentrated for sound transaction processing performance.

While taking decision about priority of requirement, requirements can be clustered and its behavior supervised as referred in Table No. 2. Based on these cluster behavior prediction of Essential Requirements will became more easy, because of which no important requirement excluded from set of software goals. As well as this Essential set will be concentrated than others.

Table No. 3

Sr. No.	*Behavior for Clustering and Supervising*	*Predicted Priority*
1	Mostly required by Different Forums/Groups	Essential
2	Average Requirements	Desirable
3	Least required	Extra/ Deferred/Optional

This classification will help to differentiate desirable requirements to plan those properly and to differentiate deferred requirements to reduce unnecessary efforts and cost on unwanted functions. This will provide basis for resource allocation in software project management.

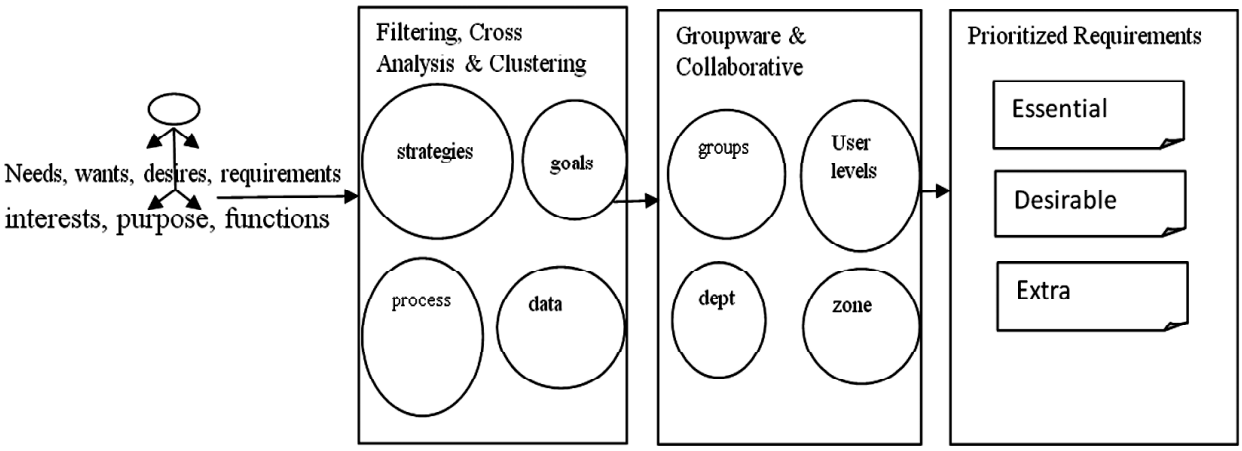

Figure 2: Prioritization based on Source Group/Forum Wise Classification

Either a requirement is to be essential, desirable or deferred can be dependent on sources from where the requirement arose. So as shown in Figure 2 clusters can be collaborative or groupware filtered and analyzed from collected needs, wants, desires, requirements, interests, purpose and functions to understand strategies, goals, process and data.

As modeling techniques are used to analyze data, its entities and relationship among entities or to get interaction between activities, to get sequence and use case and its actors. But all the data modeling technique can be applied exclusively for specific software systems. To get more precise and expected, refined and valid requirements and to reuse those based on existing past behavior of the same frequent occurrences past data from dataware housing and analyzing in different situation-based, role-based, constraint-based, preferences-based metrics are applied which leads in cross-sectional (multi dimensional) analysis.

Requirements need to be processed during elicitation, modernization of existing system, software tools, on user demands, in change management or time-policy changes. So from all these sources requirement generated is filtered, clustered, prioritized, verified, validated, classified to get differentiated patterns to reuse in software development.

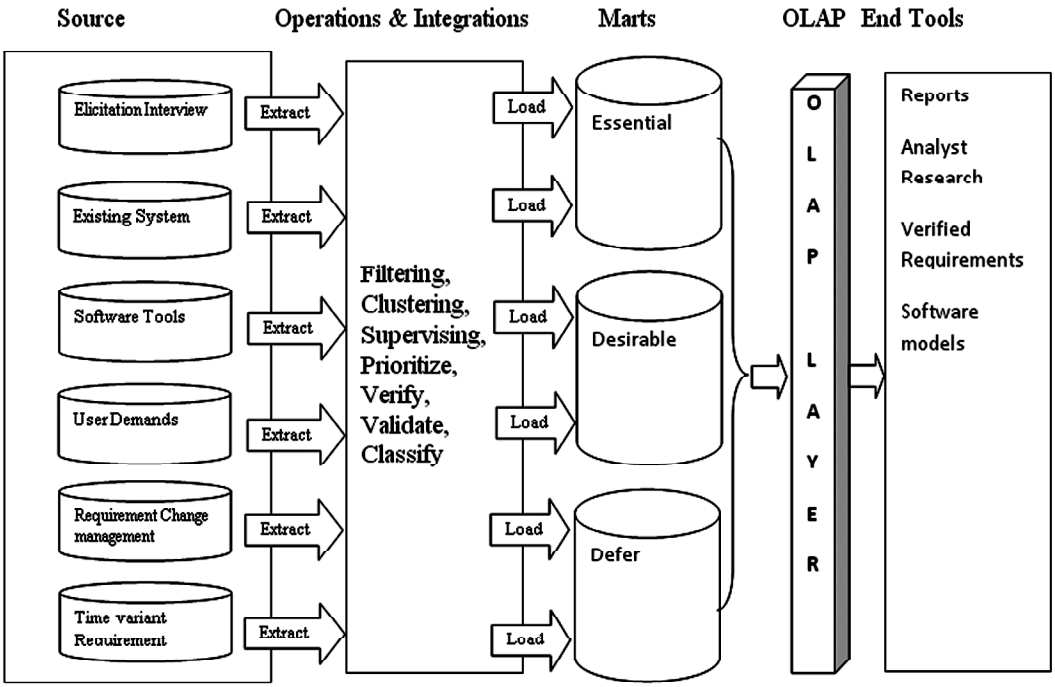

Figure 3: Data Marts to Prioritize Requirements

3. CONCLUSION

Different predictive modeling and data link analysis will be combined together to discover more refined solution to any requirement problem.

By applying (using) data mining method to software components to discover software reuse patterns, more refined approach can be applied to identify, develop and to get generic components. Patterns once established can be easily grasped and implemented often to get more appropriate and bug free solutions. Timing constraints and incrementing requirements impacts can be handled more carefully.

Precise optimized requirements will reduce cost, efforts of software and generic and specific relationship can be used in logical organizations such developments tool, libraries, packages, objects and advanced controls designs.

REFERENCES

[1] http://en.wikipedia.org/wiki/Data_warehouse

[2] http://en.wikipedia.org/wiki/Data_mining

[3] http://en.wikipedia.org/wiki/Requirements_analysis

[4] "Software Engineering", A Practitioner's Approach, Roger S. Pressman, Seventh Edition, McGraw-Hill International Edition.

[5] "Data Mining", A Tutorial Based Primer, Richard J. Roiger and Michael W. Geatz, Second Impression, 2008.Pearson Education.