A Fast Retrieval of Software Reusable Components
Using Bit Map Index

P. Niranjan, M. Venugopal Reddy, P. Shireesha
Kakatiya Institute of Technology and Science, Kakatiya University, Warangal, AP, India

Abstract
By the component libraries scaling up and the reuse practice deepening, the efficiency management of component repository is the precondition of component reuse. It is favorable for component reuse that components are organized in library according to felicitous classification. On the basis of introducing the faceted classification, this paper presents a component classification model based on facet and bit map indexing, and experiment prove it in effect. The input is the attributes of the component; in this paper we are using three Attributes: language, Operating System, File Type. Based on these attributes we perform bit map indexing and retrieve the required component from the component library.

Keywords
Software Component, Reuse Library, Faceted Classification, Bit Map Index

I. Introduction
Software reuse is an important area of software engineering research that promises significant improvements in software productivity and quality. Software reuse is the use of existing software or software knowledge to construct new software. It is of interest because people want to build systems that are bigger and more complex, more reliable, less expensive and that are delivered on time. They have found traditional software engineering methods inadequate, and feel that software reuse can provide a better way of doing software engineering. The ability to develop new applications (in particular Web-based applications) in a short time is crucial to the success of software companies that need to compete aggressively in today’s market. Considering the fact that software technologies emerge very fast, change on a daily basis, this becomes an even more complicated task. For this reason it is vital to share and reuse the knowledge and the programming experiences in an efficient and productive manner.

A. Component
Component is a well-defined unit of software that has a published interface and can be used in conjunction with other component to form large units. Reuse deals with the ability to combine separate independent software components to form a larger unit of software. A software component can be any product of the software development process -- a unit of code, a design specification, a test case, etc. Once the developer is satisfied with the component he has retrieved from library, then it is added to current project under development.

B. Software Reuse
“Reusability is a measure of the ease with which one can use those previous concepts or objects in the new situations.”

“Reuse is the use of previously acquired concepts or objects in a new situation, it involves encoding development information at different levels of abstraction, storing this representation for future reference, matching of new and old situations, duplication of already developed objects and actions, and their adaptation to suit new requirements.”

C. Reuse Environment
Reuse environment should include the following elements
User: A registered user who want to reuse the components.
Library or Repository: Is capable of storing software components and classification information to allow their retrieval.
Retrieval System: Enables client software to retrieve components and services from library server.
Matching Details: Is a mechanism carried out in an effective search for the components.
Software reuse is the use of engineering knowledge or artifacts from existing software components to build a new system. There are many work products that can be reused, for example source code, designs, specifications, architectures and documentation. The most common reuse product is source code. There are two basic technical approaches to reuse: Parts-based and Formal language-based. The parts-based approach assumes a human programmer integrating software parts into an application by hand. In the formal language-based approach, domain knowledge is encoded into an application generator or a programming language. The research reported here focuses on the parts-based approach. The aim of the good component retrieved system is to be locate either the execute component required, or the closest match, in the shortest amount of time using a suitable query. There have been many attempts to classify reusable components using various techniques. Normally, each of these methods has been implemented discretely. Each of the four main methods described (free text, attribute value, enumerated and faceted classification) has advantages and disadvantages associated with them. The proposed classification system takes advantage of the positive sides of each classification scheme, whilst hopefully rendering the negative sides redundant. This classification scheme uses the attribute value for different parts of a component. The attribute value scheme is initially used within the classification for specifying the vendor, platform, operating system and development language relating to the component. This allows the search space to be restricted to specific libraries according to the selected attribute values. Additionally, this method will allow the searches to be either as generic or domain specific as required. The functionality of the component is then classified using a faceted scheme.

Systematic software reuse is seen as a solution to address the need for short development time without compromising efficiency. Research is ongoing to develop more user-friendly and effective reuse systems. A considerable number of tools and mechanisms for supporting reuse activities in software development have been proposed.

II. Existing Software Component Classification Techniques
The following classification techniques had been employed to construct reuse repository, namely,
A. Free Text Classification
Free text retrieval performs searches using the text contained within documents. The retrieval system is typically based upon a keyword search. All of the document indexes are searched to try to find an appropriate entry for the required keyword. The major drawback with this method is the ambiguous nature of the keywords used. Another disadvantage is that a search may result in many irrelevant components.

B. Enumerated Classification
Enumerated classification uses a set of mutually exclusive classes, which are all within a hierarchy of a single dimension. The classification scheme will allow a user to find more than one item that is classified within the same section / subsection assuming that if more than one exists. For example, there may be more than one book concerning a given subject, each written by a different author. This type of classification schemes is one dimensional, and will not allow flexible classification of components into more than one place.

C. Attribute Value Classification
The attribute value classification scheme uses a set of attributes to classify a component. For example, a book has many attributes such as the author, the publisher, its ISBN number and its classification code in the Dewey Decimal system.

D. Faceted Classification
Faceted classification schemes are attracting the most attention within the software reuse community. Like the attribute classification method, various facets classify components however there are usually a lot fewer facets than there are potential attributes. Ruben Prieto-Diaz has proposed a faceted scheme that uses six facets. The functional facets are Function, Objects and Medium. The environmental facets are System type, Functional area, Settings.

III. Proposed System
A bitmap index is a special kind of database index that uses bitmaps. Bitmap indexes have traditionally been considered to work well for data such as gender, which has a small number of distinct values, for example male and female, but many occurrences of those values. This would happen if, for example, you had gender data for each resident in a city. Bitmap indexes have a significant space and performance advantage over other structures for such data. Some researchers argue that Bitmap indexes are also useful for unique valued data which is not updated frequently. Bitmap indexes use bit arrays (commonly called bitmaps) and answer queries by performing bitwise logical operations on these bitmaps. Bitmap indexes are also useful in data warehousing applications for joining a large fact table to smaller dimension tables such as those arranged in a star schema.

Consider a table that describes customers, the rating value is an integer in the range 1 to 5, and only two values are recorded for gender. Customers (custid: integer, name: string, gender: Boolean, rating: integer) Columns with few possible values are called sparse. For example, a gender value is either 10 or 01; a 1 in the first position denotes male, and 00001 denotes the rating value 5. Continuing the Customers example, a bitmap index may be logically viewed as follows:

<table>
<thead>
<tr>
<th>Custid</th>
<th>Name</th>
<th>Gender</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>Joe</td>
<td>M</td>
<td>3</td>
</tr>
<tr>
<td>115</td>
<td>Ram</td>
<td>M</td>
<td>5</td>
</tr>
<tr>
<td>119</td>
<td>Sue</td>
<td>F</td>
<td>5</td>
</tr>
<tr>
<td>112</td>
<td>Woo</td>
<td>M</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Bit Map for Gender

<table>
<thead>
<tr>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

On the left, identifier refers to the unique number assigned to each customer, gender is the data to be indexed, the content of the bitmap index is shown as two columns under the heading bitmaps. Each column in the above illustration is a bitmap in the bitmap index. In this case, there are two such bitmaps, one for gender Female and one for gender Male. It is easy to see that each bit in bitmap M shows whether a particular row refers to a male. This is the simplest form of bitmap index. Most columns will have more distinct values.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Consider the Query “How many male customers have a rating of 5?” Take the first bit vector for gender and do a bit-wise AND with the fifth bit vector for rating to obtain a bit vector that has 1 for every male customer with rating 5. We can then count the number of 1s in this bit vector to answer the query.

![Fig. 1: Proposed System Model](image-url)
The advantages of bit map indexes is that complex selection predicates can be computed very quickly, by performing bitwise AND, OR, and NOT operations on the bit map indices. Bit map indexes are also used to answer some aggregate queries directly. Compressed Bit Maps lead to speed increase.

IV. Experimentation and Results

Fig. 2: Add Component Form

Fig. 3: Inserting Component into Repository

Fig. 4: Bit Map Index Created

Fig. 5: Relevant Components Obtained

Fig. 6: Opening Downloaded Component

V. Conclusion and Future Scope of Work

There have been many attempts to classify reusable components using various techniques. The proposed classification system takes the advantage of the positive sides of each classification scheme, whilst hopefully rendering the negative sides redundant. This classification scheme uses the attribute value and faceted schemes for different parts of a component. The attribute value scheme is initially used within the classification for specifying the file type, Operating System, component name and development language relating to a component. This allows the search space to be restricted to specific libraries according to the selected attribute values. Additionally, this method will allow the searches to be either as generic or domain specific as required. The functionality of the component is then classified using a faceted scheme. This method gives better performance than current systems. Text and audio descriptions of components are also placed into the repository. Bit Map Index is created to all the attributes of the component. The component that matches with the attributes of the user is the relevant components.

Future work involved with this classification scheme will be to refine the scheme and formalize it for implementation. Though some classification scheme is developed it was not done with extensive study of all existing components. In order to be effective the scheme should be enhanced to meet the interests of people.
References

Niranjan Polala received the B.E Computer Science from Nagpur University in 1992 and M.Tech (Computer Science and Engineering) from NIT, Warangal in the year 2001. He worked as a Lecturer and Assistant Professor in the department of CSE of KITS, Warangal, Since 1996. He is doing a part-time research in Kakatiya University, Warangal since 2007. He authored two Text books, Theory of computation and Computer Graphics in the field of Computer Science. He published 10 papers in International Journals and 6 papers in International Conferences. Presently he is Professor and HOD of CSE in KITS, Warangal. He is the member of the ISTE and CSI.

Venugopal Reddy M. received B.E. (Mechanical Engineering) from Osmania University in 1990, M.B.A. from Kakatiya University and M.Tech. (Computer Science and Engineering) from NIT, Warangal in the year 2002. He is doing a part-time research in IIIT, Hyderabad since 2006. He published one paper in International Journal, one paper in International Conference and two papers in National Conferences. Presently he is Associate Professor in the Department of CSE in KITS, Warangal. He is the member of the ISTE.

Shireesha Pakala received the M.Sc. Computer Science from Kakatiya university University in 2001 and submitted her Ph.D. (Computer Science) from Kakatiya University, Warangal in October 2012. She is working as Assistant Professor in the Department of MCA in KITS, Warangal, since 2006. She published 6 papers in International Journals and 1 paper in International Conference. She is the member of the