SOFTWARE REUSE REPOSITORY

Mahmoud Diab
NEC America, Inc.
Advance Switching Laboratory
Irving, TX 75038

diab@asl.dl.nec.com
(214) 518 - 3463

Abstract

Software reusability is considered one of the most attractive ways to increase software productivity in an organization. To establish a successful reuse program, an organization must deal with two issues: 1) building an automatic reuse system which allows the users to browse the software entities, and order them. 2) The system must contain a valuable reusable data (reusable entities) which could be captured and reused.

In this paper I will discuss the software reuse repository system which was built by the software reuse group at NEC AMERICA. I will discuss in this paper the different parts of the system and their interactions.

Keywords: Software Engineering, Software Reuse, CASE, and Repository.

1 Software reuse repository

A software reuse repository is a system that stores the software reusable entities in a centralized location. The reusable entities could be software modules or documents (we could argue that documents contain software engineering information which could be reused, as an example a document that describes a software development methodology).

2 High level description of the software reuse repository

The software reuse system allows the users to retrieve records (entities) based on a search mechanism. The search mechanism is based on Keyword or attribute search.The users could select the retrieved entities, and ask the system to display detail information about them (the system supports two types of display, brief and detail display). The user could order the desired entities and ask the system to deliver them to his/her electronic address The system could be accessed by the members of the organization after they pass a level of security. The system will keep track of the users who accessed the system. A high level diagram is shown in Figure 1.

3 Functionality of software reuse repository

The software reuse system consists of the following parts: The logging and verification, the retrieval engine, and the ordering. I'll discuss briefly each part in the text below:

3.1 Logging and verification

Software reuse repository contains valuable software engineering knowledge and software code. The system must be secure enough. A user could invoke the system by typing the name of the system, then he/she could log into the system, the user must enter his/her password (currently, the system supports one level of system security, a multi level of security will be implemented in the future). The system will verify the user by checking his password with the user description table. The description table contains personal information like user name, ID number, organization, division, and electronic address. After the system verify the user, he/she will be given the permission to use the system (the system will display the main menu).

3.2 Retrieval engine

The software reuse repository support two types of search mechanism: the keyword, and the attribute search.

3.2.1 Keyword search

The keyword search is based on matching the input string with keyword list table. The user will enter the keyword(s), the system will exact match this keyword(s) with the previously entered list of keywords which are associated with each entity. As an example if the user enter "Switching" the system will display all records that are associated with "Switching".

We added extra features to the keyword search mechanism to improve upon this search. some of these features are listed below:

1. **Wild card search**

   The system will match the exact work and all keywords that vary in one or multiple character. As an example "air?line" would match "airline", where ? stand for a single arbitrary character.

2. **Thesaurus**

   The system will provide the ability to match the exact keyword and all keywords that have similar meaning. As an example, the system would relate entities such as "airplane", "Boeing 747", "aircraft", etc.

Figure 1: High level function diagram

3. **Phrase search**

   In this type of search, the user may enter a phrase (two or more words that are separated by space) instead of single keyword. As an example “expert system” would be an acceptable search criteria.

4. **Boolean combination**

   The user may provide a boolean search specification by entering queries joined by boolean operators.The default logical operator in the system is AND. If the user enters multi keywords separated by a space separator, the system will AND them together by default. Another logical operator is OR.

   The system examines selection criteria one at a time, from left to right, and it will do the logical operation that was called for. As an example, the system will process the following expression (switching OR software) AND hardware.

5. **Automatic stemming**

   The system will provide the ability to match the exact keywords and its keyword derivatives. As an example, the user might like to search all records that contain “switch” in their keyword list. The system will find all records that contain switch, switching, switcher, etc. Automatic stemming can be invoked by appending one or more characters to a search term.

3.2.2 Attribute search

Each entity has a set of attributes which define it. We identified common attributes between similar entities. The user could retrieve entities based on the value of these attributes. To enhance the attribute search we added the following features:

1. **Multi author search with any order** The user could enter multi authors in the different author fields, the system will AND them automatically. The user need not to be concerned with the order of authors. The system will take any order is desired by the user.

2. **Registration Date** The user could specify the exact date of the entity that were registered to the system. Additionally he could ask the system to give all entities that were registered from that date until now by specifying the following ‘ > “25-DEC-1990” ‘. The registration date search could support the following: equal (=), not equal (!=), greater (>), less (<), equal greater (=>), equal less (=<).

3. **Creation date** The same capability that are used in Registration date search could be used in Update date search.

The common attributes are demonstrated in the attribute screen shown in Figure 2.

Figure 2: Attribute screen.

Figure 3: Automatic entity submittal process.

3.3 Ordering

An order is defined by the process of sending a software entity from the source (in this case the software repository) to the destination (the user email address). This process must take place in an automatic way. The user will send the request of a specific entity, the system will accept the order, and will check the validity of the order, and send it to the requester electronic address. An entity order could take place any time and by multi users

4 Automatic entity submittal

The user who would like to submit an entity will select the submittal entity option, the system will display the entity submittal form. The user will enter the entity name and all related attributes, and keywords. The librarian will check on daily basis the maintenance/ attribute status. He/she will verify the information about this entity, then he/she will register it. The system will automatically generate the ID, and send it to the user.The automatic entity submittal process is shown in Figure 3.

5 Software reuse measurement process

To monitor software reuse we created a transaction file. In this file we capture different types of activities that were performed by the user during their interaction with the system.

This will allow the software reuse group to analyze these captured information, and to conclude about the software reusability of the stored entities. The following information will be captured in this transaction file:

- user name, and user ID
- In case of keyword search, what are the keyword that were typed.
- In case of attribute search what type of attribute were used.
- After successful search, what type of entities were looked at in detail (displaying detail screen).

These information will be useful in two ways:

First, it will help us to identify what are the mostly reused entities, and what is the scope of interest and the need in our organization.

Second, we could monitor our keywords and update our list of keywords by observing the user input, and what type of related keyword they had typed to retrieve specific entity.

Third, It will help us to build a list of the thesaurus table. We will relate what the user typed with the list of existing keywords.

6 Conclusion

The software reuse group built a prototype based on ORACLE database. This system will store all the organization software knowledge in one centralized system. This will allow users to retrieve this information and reuse it, or it allow users to enter their reusable software which in turn will facilitate information sharing.

References

[Bigg 87] . Biggerstaff and C Richer., Reusability Framework Assessment, and Directions. IEEE Software, july 1987.

[Bass 87] .G. Bassett., Frame-Based Software Engineering. IEEE Software, July 1987.

[Inco 90] . J. Incorvaia and A. Davis., Case Studies in Software Reuse. IEEE Software 1990.

[Frak 87] . B. Frakes, and B. A. Nejmeh., An Information System for Software Reuse. The Tenth Minnowbrook Workshop on Software Reuse, 1987.

[Arno 87] . Arnold and S Stepoway., The Reuse System: Cataloging and Retrieval of Reusable Software. COMPCON 1987.

[Prie 87] . Prieto-Diaz and G. Jones., Breathing New Life into Old Software. GTE Journal of Software, 1987.

7 biography

Mahmoud Diab is a senior member of technical staff in the software reuse group at NEC America, Inc. He received his BS, and MS in Electrical Engineering from University of Southern California. He worked in Saudi Arabia, and joined AT&T Laboratory as a member of technical staff in 1986.

His interest include Software Engineering, Software Reuse, CASE, and AI. He can be reached at NEC America., Advance Switching Laboratory, Irving, Texas.