Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp358-363)

An Approach to Manage and Search for Software Components*

Hao Chen[1], Zhong Ming[1], Shi Ying[2]
[1]College of Information Engineering, Shenzhen University, Shenzhen, 518060, P.R.China
[2]State Key Lab. of Software Engineering, Wuhan University, Wuhan, 430072, P.R.China

Abstract: - Currently component-based software engineering is increasingly being adopted for software development. This approach relies on using reusable components as the building blocks for constructing software systems. As the growth in the popularity of Internet, component providers should publish the software components easily on the Internet, and component reusers can find the appropriate software components conveniently with the aid of some tools. This makes it true for reusers to build software system integrated with the components provided by others. Therefore, the major two problems are how to manage the COTS components and how to find suitable components on the Internet. The solution to the first problem lies in constructing a software component repository based on the component classification, which can realize the sharing of software components resources. Applying search engine technology to search for matched components is a good idea to solve the second problem. This paper proposes a new approach to manage and search for software components, which support for component provider to publish the components and component reusers to search for software components.

Key-Words: - Systematic classification, faceted classification, software component repository

1 Introduction

Since the idea of software reuse was proposed, the technology and method of software reuse have been researched in depth. Coming in with the introduction of component-based reuse in later 90's, this approach has gained substantial interest not just in the research community but in numerous industry sectors [1]. Now this approach has been applied in software development broadly. Component-based reuse focuses on that software development cycle should be significantly shortened by reusing and assembly the existing software components. However, two key problems exist in this domain. One problem is how to publish and manage the existing software component resources. The solution to this problem lies in classifying the component and constructing the software component repository. The problem facing the component reusers is how to find suitable software component resources on the Internet. Our solution to this problem is to apply the technology of search engine to search for component resources. Additionally, the search engine for searching component resources should collaborate with the retrieval engine built-in the software component repository. The search engine provides one rough search manner, while the retrieval engine can be applied to find the components accurately. These two kinds of engines establish a two-level query mechanism together.

After comparing the related classification of software components, this paper proposes the hybrid approach to classifying the software components. This approach blends the systematic classification and faceted classification, which both come from library and information retrieval. The aim to adopt the systematic classification is to support the search engine to classifying the components. The motivation of applying the faceted classification lies in supporting the retrieval engine. We have developed the prototype system of software component repository management platform, which can be used for building and managing the component repository.

Search engine can facilitate component reuser to find the software component matching their requirements. However, general-purpose search engines are inappropriate to search software components. Some people developed the specific search engines for software components. But those engines have some limitations. This paper proposes a new specific search engine for software components(SE4SC), which provides convenient

* This research was supported by the the National Social Science Foundation of China under grant 05CTQ001; Natural Science Foundation of Guangdong under grant 04011304; Shenzhen Science & Technique Plan under Grant 200422;

support for component reusers to search software components on the Internet.

The rest of this paper is organized as follows: Section 2 reviews related works. Section 3 presents the approach to classify and managing the software components and shows the prototype implementation of the software component repository management platform. In Section 4, we introduces the SE4SC and illustrate the prototype system of SE4SC. Contributions of this work and areas for future work are finally presented in Section 5.

2 Related Work

2.1 Systematic Classification

Systematic classification is a kind of well-defined document classification, which is based on science knowledge and the division of concepts. This method is enough to provide a rough classification though its disadvantage for dynamic update. The main schedule of systematic classification is a schema of classes which consist of basic classes, basic large classes, summary tables and detail tables. Subdivisional table, which supply a schema of classes for subdividing the category of main schedule, is extracted from a set of subsections which classified by the same standard in the main schedule and compiled independent of main schedule.

Systematic classification has been adopted in the information retrieval domain widely, for instance, Yahoo! This method is also introduced to classify the components, for example ComponentSource, which is a professional website for components resources. But some shortcomings exist in the classification schema, such as only two level hiberarchy, short of subdivisional table.

2.2 Faceted Classification

The faceted method, as used in library science, relies on building up or synthesizing from the subject statements of particular documents, which construct a “Facet-Subfacet-classes” structure according to the contents and subjects of document. This method provides a expression for document’s subjects through the combination of some facets. Facet is a group of categories which are produced according to some classification standards and sometimes considered as perspective, viewpoint, or dimension of a particular domain.

Faceted classification has been applied to classification for software resources in early 1980. Dr. Rubén Prieto-Díaz proposed the software component classification, which is based on faceted classification in 1987 [2]. That faceted classification schema included the following six facets: Function facet, Objects facet, Medium facet, System type facet, Functional area facet, Setting facet. However, the original component classification schema has evident limitations, such as lack of reuse-oriented facets, short of available information supporting reuser to find and reuse components, and too simple to satisfy the needs of reusers.

2.3 General-purpose Search engines

A general-purpose search engine always consists of crawlers, indexer, indexes storage and query module. The crawlers visit the pages and documents on the Web and store them. The indexer extracts all the keywords from each page or document and builds the index for every keyword. The query module therefore is responsible for receiving search requests from users. This module relies heavily on the indexes [3]. Currently general-purpose search engines, such as Google, AltaVista, InfoSeek, WebCrawler, Nutch, enable users to search for web resources published on the Internet effectively, such as HTML web pages, documents with PDF, PostScript, or MS Word format [4]. However they are not proper for searching for component resources on the Internet.

General-purpose search engines are just used for locate the web pages and documents that have a particular extension name. For the purpose of providing search support for the users, search engines build indexes by analyzing the content of these web pages and documents. Software components are not similar to those general documents, which usually are binary code or byte code. They have no particular extension name for identifying and always adhere to some standard component model specifications, such as JavaBean, EJB (Enterprise JavaBean), COM, ActiveX, CCM (CORBA Component Model). Therefore, the process of extracting the information of interfaces from component entities must refer to the component model these entities follow. So it is unpractical for general-purpose search engines to locate component resources on the Internet and extract interface information from their entities.

2.4 Existing Specific Search engines

Software Engineering Institute(SEI) of CMU developed the specific engine: Agora for searching for software components. The object of Agora is to create an automatically generated, indexed, worldwide database of software products classified

by component type, and provide the service of searching for components for the reusers [5].

The workflow of Agora engine is similar to that of general-purpose search engines and contains two basic phases: the location and indexing of components and the search and retrieval of components.

Although Agora is the specific software search engine, it has two disadvantages:

(1) It still uses the AltaVista Web search engine to support the search for HTML documents.

Some deficiencies of this integration mechanism are as follows: First, in order to be located by the JavaBean Agent of Agora, the JavaBean components must be embedded into the web pages as applet. Second, not all the applet classes searched by the JavaBean Agent are JavaBean components in that the applet classes aren't need to follow the JavaBean component model. Finally, a lot of web pages that contain "applet" tag not really contain the applet code. So the results returned by the AltaVista Internet service may contain lots of useless information. Filtering this information will reduce the efficiency of Agora.

(2) It only lays emphasis on the characteristics of the component interfaces, but ignore the other characteristics that reflect the reusers' requirements.

In the Agora engine, the agent uses the particular introspection process depending upon the specific component model to extract the characteristics of interfaces and index these characteristics. To a certain extent, it solves the problem that the general-purpose search engine can't precisely match the component attributes, but it only emphasizes the interface characteristics. However, these characteristics extracted by the agent don't contain the information that the reusers concern more, such as, application domain, deploy environment, and performance. Agent should extract some high-level characteristics of components, rather than only low-level characteristics of components such as operation, attribute-level names, and so on.

In addition to Agora, the alphaBeans developed by IBM is another search engine used for searching for components. The description of the components searched by alphaBeans includes more information, such as simple introduction, installing information, requirement information, evaluation information, FAQ [6]. But it has evident limitations for using alphaBeans to search for software components. AlphaBeans search for JavaBean components exclusively, and can only locate the components resources on a small scale.

3 Component Management

Applying some method to classifying the software components and constructing a software component repository can manage the software components effectively.

3.1 Hybrid Component Classification

In order to make search engine and retrieval engine work effectively, we propose the hybrid approach to classifying the software components, which adopts two kinds of classification schema. One classification schema is based on the systematic classification. The other classification schema is based on the faceted classification. Both schemas are defined using the XML Schema. Systematic classification schema is search engine oriented. According this schema, component descriptor offered by the component provider supports for reusers to search for component roughly by using the search engine. Faceted classification schema is retrieval engine oriented. Component provider describes the software components based on this schema and offers the component descriptors that facilitate component reuser to retrieve components accurately with the aid of the retrieval engine [7]. All descriptors are described by XML, so the descriptors can be validated by the classification schema that is described by XML Schema. The process of acquiring the suitable components contains two phases: search and retrieval phases. Firstly, component reusers find the candidate components roughly with the aid of search engine. And then they retrieve components accurately among the candidates by using the retrieval engine. That is to say, these two kinds of engines establish a two-level query mechanism together.

When describing the component referring to hybrid classification, the component provider must generate the descriptors based on two classification schema respectively. As a result, there exist two descriptors based on different schema for each component. One descriptor based on the systematic classification contains the basic information used for rough search. Search engine need to generate the indexes of the classification information. To keep the indexes stable, systematic classification schema had better to be stable. Therefore, a loose systematic classification schema is adopted in terms of search requirement for software components. This systematic classification schema consists of main schedule and subdivisional tables. Main schedule includes basic large classes and basic schema of classes. Basic large classes include three classes: System Software, Supporting software for

development, and Application software. Laying out each large class, then a basic schema of classes is formed. For example, in regard to Application Software large class, the basic schema includes finance, telecom, business, taxation, government, health, and education class. There are three subdivisional tables with hierarchical structure: component specification table, development platform table and runtime platform table. For instance, the component specification table includes seven classes that are JavaBean, EJB, ActiveX, COM, COM+, DCOM, CCM class.

The other descriptor based on the faceted classification contains the information about basic information and domain information, which is helpful for accurate retrieval. Figure 1 shows the structure of faceted classification schema.

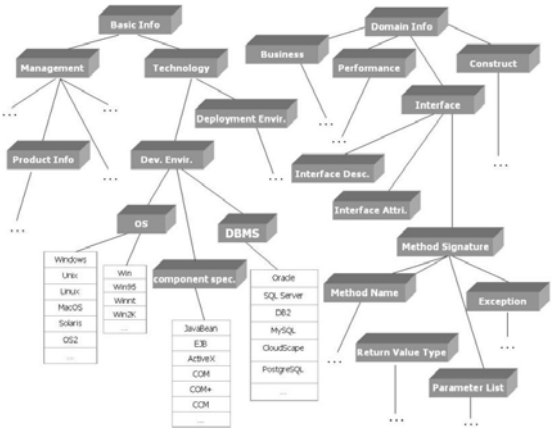

Fig.1 The structure of faceted classification schema

3.2 Software Component Repository

Building a large-scale software component repository based on some classification is an effective approach to manage the software components. A prototype system of the software component repository management platform has been developed [8]. This platform can give help to administrators for constructing and maintaining the component repository.

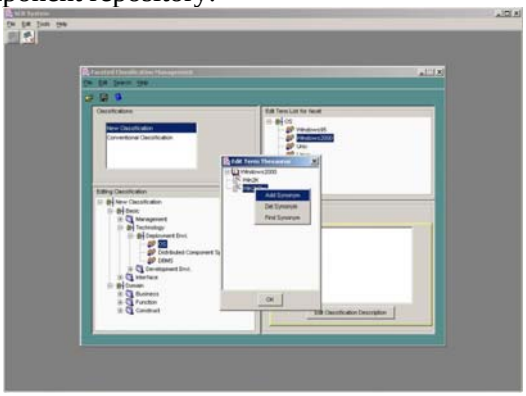

Fig.2 software component repository management platform interface

As the figure 2 shows, with the supports provided by the platform, the administrators can maintain the classification schemas, component providers can register the components into the repository, and component reusers can retrieve the components matching their requirement in the help of the retrieval engine.

4 SE4SC

Considering the deficiencies when existing search engine is adopted to search for components, we propose the SE4SC that is a new specific search engine for software components. The motivations of SE4SC are presented as follows:

(1) SE4SC mainly acquires the components from the software component repository.

(2) SE4SC can provide two means for reusers to search for components. One is the keyword-based search, and the other is the topic directory based on systematic classification schema.

(3) SE4SC should support the rough search for components, but also can forward the reusers' search requests to retrieval engine built in the software component repository for retrieving the components accurately.

4.1 Architecture of SE4SC

SE4SC consists of Repository Registry Center, crawlers, systematic classifier, search service interface, search service control, index database, Component Descriptor Repository and so on, as shown in figure 3. The details of each part will be presented as follows:

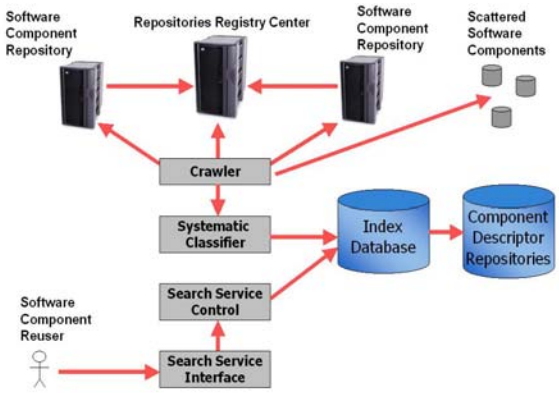

Fig.3 Architecture of SE4SC

(1) Repository Registry Center

Repository Registry Center provides registry service for component repositories and stores systematic classification schema.

(2) Crawlers

Crawlers locate the components and extract descriptors from component resources. At first,

crawlers obtain the URLs of the component repositories registered in Repository Registry Center, or find particular marked component repositories on the Internet. Then, crawlers call the retrieval component service provided by each software component repository and obtain all the component descriptors. Crawlers adopt the technology of multi-threads and cache to improve efficiency. Considering the problem of network load, crawlers extract component descriptors from repository in batches.

(3) Systematic Classifier

Systematic classifier indexes component descriptors based on the systematic classification schema and store the descriptors in the Component Descriptor Repository.

(4) Search Service Control

Search service control matches the candidate components and returns them to search service interface depending on the search means, search criterion and systematic classification schema. Search service control also takes on responsibility for forwarding the reusers' search requests to retrieval engine provided by the software component repository for further retrieval.

(5) Search Service Interface

Search service interface provides reusers two means of searching for components by calling search service control.

(6) Index Database

Index database is used for storing indexes that are generated by systematic classifier. The establishment of index database can improve the performance of SE4SC.

(7) Component Descriptor Repository

Component Descriptor Repository is used for storing component descriptors extracted by crawlers.

The workflow of SE4SC is similar to that of Agora and includes two phases:

(1) Locate the component resources, obtain the component descriptors and indexes the information in the component descriptors

The crawlers get the URLs of the software component repository, and start multiple threads to call the services supported by the repository, and then extract component descriptors in batches. Systematic classifier indexes the classification information in the component descriptors, classify and store the component descriptors.

The crawlers can also locate component resources adhering to SCDM in the distributed environment. Similarly, the systematic classifier indexes and stores the descriptors.

(2) Provide search service for the reusers

SE4SC provides two means of searching for components for the reusers. The advantage of keyword-based search is easy to use. However, the semantics of a keyword varies with different domains. So its imprecise maybe results in too many recalls. While topic directories based on systematic classification schema can assign the components to the right topic by using the important classification information that the component descriptors contain. So topic directories have higher precise than keyword-based search. Also, topic directories means facilitate the component reuser to narrow the search space, which is preparatory to accurately retrieve components by using the retrieval engine built in the software component repository.

The component reusers can select the keyword-based search, and enter any keywords as the query string, and then get the candidate components. Further, they can check the details of the component by clicking the link of the components. Alternatively, the reusers can use the search means based on systematic classification schema to search for components, and examine the details of the component description. Furthermore, the reusers' search request can be forwarded to the retrieval engine of software component repository for searching for components accurately among the candidates.

4.2 Prototype System Of SE4SC

The prototype system of SE4SC acquires component resources, and provides the convenient support for reusers to search for components. As follows, we illustrate that reusers search for components by using two means provided by SE4SC.

(1) Keyword-based search

Component reusers submit the search request by entering the keyword. As shown in the Figure 4, reusers enter the "EJB" as the search keyword in the input box, and then SE4SC returns the search results on the right web page.

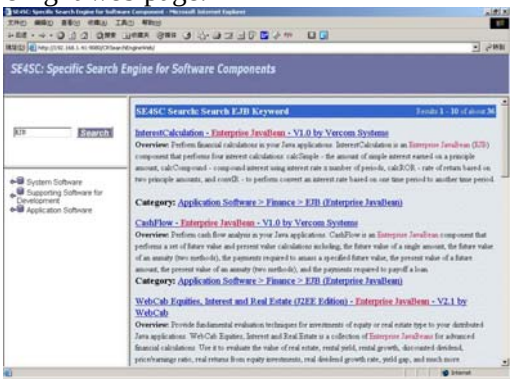

Fig.4 Component reusers search for components by using the keyword-based search.

(2) Topic directory search based on the systematic classification schema

Component reusers find components by using the topic hierarchy in the lower left frame of the web page. As shown in Figure 5, reusers specify the search request as searching all EJB components under the topic "Finance" which belongs to the topic "Application Software". Then SE4SC matches the components and returns the search results on the right web page. These results show that six candidate components are matched successfully by SE4SC. Component reusers can click the link of any candidate, and then view the component in details.

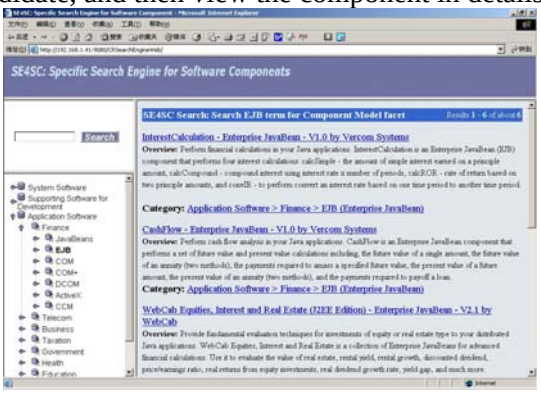

Fig.5 Component reusers search for components by using topic directories search.

After searching components by using the SE4SC, the reusers can use the retrieval engine of software component repository for accurate retrieval. The reusers select the faceted classification schema, specify the interesting facets, and assign the specific term for these facets, and then submit these retrieval requests for retrieving the components accurately among the candidates.

5 Conclusion

This paper has presented an approach to manage and search for software components. This method adopted a hybrid component classification for different purposes. The systematic classification can supports the SE4SC. The faceted classification is retrieval engine oriented, which focus on the reuse requirement of the component reusers. Furthermore, a software component repository management platform is implemented as a prototype system. This paper also proposes the architecture and prototype implement of SE4SC. SE4SC can collaborate with the retrieval engine built-in the software component repository. This search engine provides one rough search manner, while the retrieval engine can be applied to find the components accurately, which establish a two-level query mechanism.

In the future, the performance of SE4SC should be enhanced, and the hybrid component classification should be improved by defining a more stable systematic classification schema and introducing a specific application domain faceted classification schema. Furthermore, the SE4SC should seamlessly integrate with the retrieval engines and provide the convenient support for the reusers to search for the matched components by utilizing the two-level query mechanism.

References:

[1] C. Szyperski, *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley, New York, 1998.

[2] R. Prieto-Diaz and P. Freeman, *Classifying Software for Reusability*, IEEE Software4 (1), IEEE Computer. Society Press, Los Alamitos, CA, January 1987,pp.6-16.

[3] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan, *Searching the Web*, ACM Transaction on Internet Technology, Vol.1 No.1, August 2001.

[4] Shian-Hua Lin, Meng Chang Chen, Jan-Ming Ho, and Yueh-Ming Huang, *ACIRD: Intelligent Internet Document Organization and Retrieval*, IEEE Transaction on Knowledge and Data Engineering, Vol.14 No.3, pp.599-614, May/June 2002.

[5] R. C. Seacord, S. A. Hissam, and K. C. Wallnau, *Agora: A Search Engine for Software Component*, Technical Report, CMU/SEI-98-011, 1998.

[6] AlphaBeans Homepage, http://www.alphaworks.ibm.com/alphabeans.

[7] Padmal Vitharana, Fatemeh Mariam Zahedi, and Hemant Jain, *Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and an Empirical Analysis*, IEEE Transactions on Software Engineering, Vol.29, No.7, pp.649-664, July 2003.

[8] Zheng Ying, Chen Hao, Li Weizhai, Ying shi, *Construction and Implementation of Component Repository Based on Web Services Technology*, Journal of Wuhan University of Technology, Vol.26 No.2, pp.80-83, Feb.2004.