

TECHNIQUES FOR COMPONENT REUSABLE APPROACH

Sukanay.M¹, Biruntha.S², Dr.Karthik.S³, Kalaikumaran.T⁴

¹ II year M.E SE, Department of Computer Science & Engineering (PG)
SNS College of Technology, Sathy Main Road, Coimbatore-641035

sukanmukesh@gmail.com

² II year M.E CSE, Department of Computer Science & Engineering (PG)
SNS College of Technology, Sathy Main Road, Coimbatore-641035

brindhavijay278872@gmail.com

³ Professor and Dean, Department of Computer Science Engineering
SNS College of Technology, Sathy Main Road, Coimbatore-641035

profskarthik@gmail.com

⁴ Professor and Head, Department of Computer Science Engineering
SNS College of Technology, Sathy Main Road, Coimbatore-641035

tkalaikumaran@gmail.com

ABSTRACT

To find out best reusable component is a difficult task for the user. The keyword based approach does not provide accurate result, which is largely used in descriptive methods for component retrieval. The keyboard based procedure is used for descriptive procedure broadly, it produces the result is too formless. For retrieval approach of component, to make them as reusable smart environment is used. In keyword search region, this will list out the components, the user must select appropriate component manually. The class diagram provides the information about structural description and contents of class effectively. That is class name, attributes, relationships, behavior operations and generalization etc. Repository is used for this information; this search result is better and produces high accuracy whenever it is compared with keyboard based search procedure.

KEYWORDS : MDL, UML models, search environment.

1. INTRODUCTION

A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and which is subject to component by third parties. The 3 parties are involved, one specifies the component, one implements the specification of a component and another one deploys or uses the component. Software reuse is a process of implementing or updating the software systems by using existing software assets. Software assets or components include all software products, from requirements and proposals, to specification and designs, to user manuals and test suite. Anything is produced from software development that can be potentially reused. The traditional approach for component retrieval is keyword based, so which produces the less relevant components only, the user can not select the exact appropriate component. Retrieval based approach produces the more number of components, which are related to the environment. The retrieval based component selection is based on the MDL file format. Where the content of file of diagrams are matched to

retrieval component. UML models are used to store the MDL file format. This contains the important and valuable rich information. The class diagram MDL file format have valuable information about the structural description and contents of a class, i.e. class name, attributes, behaviour, relationships, generalization, cardinality etc. These attributes can be used for specification matching with the contents of the repository. The Use case diagram file contains valuable information about the requirements specification of software. These include use cases and actors. If we search the repository on the basis of attributes of MDL file descriptions, the search result would be better and thus giving much higher precision, as compared to keyword based searching procedure. Hence the role of user to find the best suitable component from the search results would be much easier. In this paper we describe a tool named as a smart environment for retrieving the related components, which assists the software designers in the retrieval of the designs as well as source codes.

2. LITERATURE SURVEY

The UML models that are used for modelling are stored as MDL file format. These MDL file formats are practically very information rich and contains lot of meaningful information about the components. This available information can be structural as well as behavioural. To successfully combine two paradigms software reuse and UML, for Component Retrieval, an automated tool must be designed, named as Component Retrieval Search Engine. The Purpose of Component Retrieval Search Engine is to retrieve best-fit or most reusable Component from the Repository as intended by the (re) user. Moreover the search results to be displayed in descending order of percentage match with the input query. Hence the role of user to find the best-fit component from the search results would be much easier. Hybrid technique based on natural language description and formal specifications using K-nn technique has been discussed. It discusses the reuse and benefits of reuse using formal methods. It also demonstrates the benefits of using natural language along with formal methods, which is supported by the tool demonstrated. It discusses a Reuse WELL System which exploits the benefits of both formal methods and natural language in the retrieval of software components. This technique helps users in organization and storage of components and later can help in identifying most appropriate components. In the first stage while searching it makes use of keywords, their synonyms and their interrelationships. Then it makes use of ant colony optimization; initial pheromone of one is assigned to all domain representation terms of components.

2.1. Software Retrieval Method

Assets include source code, executable code, requirements specification, design description, test data, documentation, and proof. The library is to be used within a project, across a program, across a product line, across multiple product lines, worldwide. Queries can be submitted in various forms as functional specification, signature specification, keyword list, design pattern, behavioral sample. It dictates what form user queries should take. Assets can be represented as functional specification, signature specification, source code, executable code, requirements specification, documentation, and set of keywords. Storage structures can be flat structure, hypertext links, refinement ordering, ordering by generosity. Various navigation schemes include Exhaustive linear scan, navigation hypertexts links, and navigating refinement relations. Finding out assets that are correct with respect to a given query. Various retrieval goals include correctness, functional proximity, and structural proximity. Relevance Criterion includes correctness, signature matching, minimizing functional distance, minimizing structural distance. The condition that is chosen to check between the submitted query and a candidate library asset. Various factors for matching condition include correctness formula, signature identity, signature refinement, equality and subsumption of keywords, natural language analysis, and pattern recognition.

2.2. Existing Techniques

The major drawbacks of the traditional descriptive classification schemes for software component retrieval are:

Ambiguity problem in keyword based search procedure; when different words mean different things to different people.. Both Precision and Recall are not high. These techniques are based on a proscribed vocabulary that must be constructed manually for each and every application domain. Both classification and retrieval require important human effort because users must select appropriate terms for each facet in the classification scheme from usually list of terms in the controlled vocabulary. An approach combining main advantages of descriptive classification methods for component retrieval in order to improve retrieval effectiveness and provide a friendlier user interface through the use of queries in MDL format.

2.3. MDL File Reader

Diagrams are the pictorial representation of the system. Here we apply some principles to reveal the details from the diagrams. The diagrams can be easily modeled in some software like Rational Rose or Visualizer. After developing the designs of software in UML, store the file. By specifying and storing all the diagrams in repository, we can read and extract the details. Taking the example of class diagram, in a class diagram, there are different classes, operators, attributes, relationships and documentation or comments provided for explanation and behavior. There is relationship between the classes like inheritance, association or any other dependency. Then they are related with the Cardinality. These files must be read by applying some algorithm, to extract all the details. But one must keep trace of classes, operations, attributes and their relations. The attributes of one class should not mix with other class. To implement such algorithm, MDL file Reader is used to extract the necessary information. The MDL file can be read by treating it as a simple text file. Start reading the file and whenever a match is found for object class, object attributes and object operation store the string following next in the temporary storage. Likewise all other attributes can be stored by reading the file. The contents are stored in a temporary storage and used for match when the search is to be performed.

The Repository Design: The purpose of Repository is to store designs as well as source codes. Designs are stored in MDL format and source codes consist of programs in Java. The architecture of Repository is prepared for storing both designs as well as source codes.

Retrieval Mechanism: To extract diagrams or components from repository, to facilitate software reuse a search engine is required to fetch data as per requirements of the user. As there are many diagrams in UML the search for each diagram is different and the technique required is different. To search from the repository, different search techniques are utilized to extract relevant data.

MDL File – Class Diagram Search:

Search Technique – In this case, search is made on the MDL file Class Diagram. Now the search is not defined on the keywords or comments of a class. In this case the search is performed on the structure of the whole software i.e. class name, attributes name and operations name. The class diagram of new software is modeled in Rational Rose and the file is saved in MDL file format. Now for the search, this MDL file would be the input to the query. The software would read the class structure modeled and depending upon these structures will search for structure in repository and would result back with preexisting MDL files as well as source codes from the repository. Moreover some weights are assigned to different contents of a class as class name, class attributes and class operations. These weights are computed and stored along with the component path in a temporary storage. The search results are displayed in the decreasing order of their weights i.e. percentage match of the component. The component with highest weight is displayed first and

then next and so on. Thus the best-fit component is that which is having highest percentage match. Hence user can easily find the best-fit component from the search results.

MDL File – Use Case Diagram Search:

Search Technique – In this case, search is made on the MDL files Use Case Diagram. Now the search is not defined on the keywords or comments of a Use Case Diagram. In this case the search is performed on the functional requirements of the whole software i.e. use cases and actors. The use case diagram of new software is modeled in Rational Rose and the file is saved in MDL file format. Now for the search, this MDL file would be the input to the query. The software would read the requirements specification modeled and depending upon these requirements specification will search for requirement specifications in repository and would result back with pre-existing MDL files from the repository. Moreover some weights are assigned to different contents of a use case diagram as actors, which is a special kind of class and use case. These weights are computed and stored along with the component path in a temporary storage. The search results are displayed in the decreasing order of their weights. The component with highest weight is displayed first and then next and so on. Thus the best-fit component is which is having highest percentage match. Hence user can easily find the best-fit component from the search results.

2.4. Example Cases

As per the data in the database, required is a search of a data from the repository. The search technique and the search query depend upon the data to be extracted and the data present in the repository. The functionality of the retrieval mechanism is to fetch the data from the repository to the user in the desired output. Here, we are dealing with designs and source codes. Designs itself cannot be reused as it is, as there is no physical component that can be extracted, although source codes can be reused as it is. The output form would be the information about the components or the details provided by the repository manager at the time of insertion of the component in the repository. The form of output of the search query is the file itself. Next level of data extraction is to be provided by the developer as how to use the component. Also the output can provide the information about the components, that component could be found from the Repository and then reuse principles are applied. Here the main motive at the design and coding level is to know about the similar kinds of components as per the search query, present in the Repository.

3. CONCLUSIONS

UML diagram is the emerging trend in the software development process. The result says the whenever combining these techniques can help software development process by finding an existing components at the design time only. Whenever we combine the UML and software reuse that produces the required component that is best fit for software development modules. The search can be made on class diagrams to search according to structural description of the software. The search can be made on use case diagrams to search according to requirements specification of the software.

REFERENCES

- [1] Suresh Chand Gupta, Prof Ashok Kumar “Smart Environment for Component Reuse” International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 2, February 2012
- [2] Arun Sharma, Rajesh Kumar and P .S. Grover, “A Critical Survey of reusability aspects for component-based systems”, Proceedings of World Academy of Science, Engineering & Technology, Vol. 21, Jan 2007
- [3] Daniel Lucredio, Antonio Francisico do Prado, Eduardo Santana de Almeida, “A Survey on Software Components Search and Retrieval”, euromicro, pp.152-159, 30th EUROMICRO Conference (EUROMICRO’04), 2004
- [4] Rajesh K Bhatia, Mayank Dave, R.C Joshi, “A Hybrid Technique for Searching a Reusable Component from Software Libraries”, DESIDOC Bulletin of Information Technology, Vol.27, No.5, September 2007, pp. 27-34
- [5] Boehm, B “Managing software productivity and reuse”, IEEE Computer 16(9), 111- 113, 1999

AUTHORS

M.Sukanya was born on May 1990 at Dindigul, Tamilnadu, India, received B.E degree in Computer Science and Engineering from Angel College of Engineering and Technology, Tirupur in 2011. She is currently pursuing M.E degree in Software Engineering at SNS College of Technology, Coimbatore. She has presented three National and two International conferences. and published one paper in international journal. Her area of interests are Data Mining, Cloud Computing and Software Engineering.



S.Biruntha was born on August 1987 at Srivilliputhur, Tamilnadu, India, received B.Sc degree in Computer Science from Pioneer College of arts and science, Coimbatore in 2008 and also received MCA degree from Bharathiar University, Coimbatore. She is currently pursuing M.E degree in Computer Science and Engineering at SNS College of Technology, Coimbatore. She has presented one National and two International conferences and published one paper in international journal. and Her area of interests are Data Mining, Visual Cryptography, Biometrics privacy and Software Engineering.



T.Kalaikumar received MCA degree from the Madras University in 1996, and the M.E. degree in Computer Science and Engineering from the Anna University, Chennai, in 2006. He is a Postdoctoral fellow at the Anna University of Technology, Coimbatore. He is interested in the research areas of data mining, spatial data mining, machine learning, uncertain data classification and clustering, pattern recognition, database management system and informational retrieval system. He is a member of CSI and IEEE.



Dr.S.Karthik is presently Professor & Dean, Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University, Tamilnadu, India. He has received his Master Degree in Engineering from Anna University Chennai and Ph.D degree from Anna University Chennai in the area of information and Communication Engineering. His research interests include network security, web services and wireless systems. In particular, he is currently working in a research group developing new Internet security architectures and active defense systems against DDoS attacks. Dr.S.Karthik published papers in international journal and conference papers and has been involved many international conferences as Technical Chair and tutorial presenter. He is an active member of IEEE, ISTE and Indian Computer Society.

