Panel Session: Information Retrieval and Software Reuse

Panel Chair: W.B. Frakes, AT&T Bell Laboratories; **Panelists**: N. Belkin, Rutgers, R. Prieto-Diaz, Contel Technology Center, S. Wartik, Software Productivity Consortium

ABSTRACT: *Software reuse is widely believed to be the most promising technology for improving software quality and productivity. There are many technical and non-technical problems to be solved, however, before widespread reuse of software lifecycle objects becomes a reality. One class of problem concerns the classification, storage, and retrieval of reusable components. Panel members will discuss these problems and some approaches to solving them.*

Classification, Storage, and Retrieval of Reusable Components

W.B. Frakes, P.B. Gandel

1. Introduction

A fundamental problem in software reuse is the lack of tools for representing, indexing, storing, and retrieving reusable components. Previous research has identified three promising approaches to this problem:

- Information storage and retrieval (IR),
- Database Management Systems (DBMS),
- Artificial Intelligence (AI).

Since each of these approaches offers quite different advantages and disadvantages, it is likely that future reuse systems will combine aspects of all three. The reuse database for such systems is envisioned as being split into many smaller databases. This will allow the database to be modularized and possibly distributed across a large user organization. Modularization will also allow the databases to be more easily customized for specific environments, and will allow different kinds of reusable objects to be placed in separate databases.

The major issues to be faced in designing a storage and retrieval system for reusable components are,

- How to classify and index the components,
- How to store and search for the components,
- How to evaluate the effectiveness of the system.

We will now examine each of these issues in more detail.

2. Classification and Indexing

Classification is the process of assigning an object to a category. Classification is one of the commonest activities that human beings perform, and many classification schemes have been developed that help with the classification of objects. In the natural sciences taxonomic keys provide classification advice for plants, animals, and minerals. Research on expert systems for classification has renewed interest in the process and technology of classification.

When one indexes an item, one assigns it to one or more classes. Indexing is the process of assigning a surrogate record to represent an item. For example, in document retrieval systems one often uses a set of descriptive keywords to represent a document. As such, the keywords denote the classes to which the document belongs.

Indexing methods are often classified along a scale whose endpoints are controlled vocabulary and uncontrolled vocabulary. This continuum refers to the degree of freedom that an indexer (human or machine) has in assigning index terms to an item. For example, in uncontrolled vocabulary indexing, the indexer is not restricted in the assignment of indexing terms to an item. This continuum also refers to the degree of predetermined relationships among the indexing terms. For example, the Dewey decimal system for classifying books uses a predefined hierarchical organization of the classes where each class is mutually exclusive. A faceted system is synthesized at the time index terms are

assigned to an item.

Controlled Vocabulary

| Hierarchical Classification

| Faceted Classification

| Subject Headings

| Descriptors

| KWIC (keyword in context)

| KWOC (keyword out of context)

| Text Derived Terms

Uncontrolled Vocabulary

Uncontrolled vocabulary indexing is often referred to as free text indexing since the index terms are usually derived form the documents themselves. This is the approach taken by (Frakes, 1988) for software reuse. (Prieto-Diaz, 1987), on the other hand has taken the opposite approach by developing a controlled vocabulary using a faceted classification scheme.

While uncontrolled vocabularies will tend to be less expensive to implement, the performance tradeoffs between controlled and uncontrolled vocabularies are unknown for collections of reusable components. Experiments with document retrieval systems, beginning in the early sixties, have shown that uncontrolled vocabularies produce retrieval results that are comparable to those produced with uncontrolled vocabularies. A good overview of these experiments can be found in (Sparck-Jones, 1981) and (Salton and McGill, 1983). Other experiments (Katzer, 1983) have demonstrated that while different indexing methods will perform roughly equally on such measures as recall and precision, they will cause different documents to be retrieved. Thus these methods may be viewed as complementary.

Another way of classifying reusable components is to use the various knowledge representation techniques from artificial intelligence. Frames, semantic nets, and production rules are the most popular formalisms for knowledge representation (Winston, 1984). Since a central problem of IR has been how to represent the meaning of text or other records in a way comprehensible to a computer, the knowledge representation techniques used in AI systems seem promising as a growing literature attests (Smith, 1987). Production rules, for example, have been used to create an intelligent thesaurus (McCune, 1985), and natural language systems have been used to extract and formalize the information in medical documents (Sager, 1981).

Rosales and Mehrotra (Rosales, 1988) have used a rule based system to help users select and modify code components. Devanbu et. al (Devanbu, 1989) have used a semantic net approach to classifying software components, and have experimented with natural language techniques as well. The effectiveness of these methods for reuse remains an area for research.

Classification Methods	
Issues	Tools
Controlled Vocabulary - Prieto-Diaz	Context Clarification Tool Semantic Closeness Tool Thesaurus Construction Tool - Expert Boolean IR System
Uncontrolled Vocabulary - Frakes, Nejmeh	Vector Space IR System Word Processing Tools - Stemming - Soundex
Knowledge Representation - Devanbu - Semantic Net - Entity Relation Model - Frames - Rosales - Rules	Semantic Net Shell Frame Shell Rule Based Shell

3. Storage and Retrieval

Many different types of systems for handling information are currently in use. These systems have different underlying models and capabilities. Perhaps the best known type of system is the database management system (DBMS) (Date). DBMS are widely used for storing, managing, and retrieving highly structured information such as parts lists, personnel files, etc. While DBMS are powerful, they are usually limited in their ability to handle data that is not highly structured, such as text or source code. Current systems for handling this

kind of data are information retrieval (IR) systems (Lancaster, 1973; Salton, 1983). Originally developed to manage the literature of the natural sciences, IR systems incorporate many techniques for storing and retrieving unstructured data, such as boolean queries and partial string matching.

Because IR systems are capable of handling unstructured data, they can be used to store and retrieve products produced throughout the software lifecycle such as feasibility documents, requirements documents, design documents, code, test cases, test documents, methodology documents, maintenance documents, quality information, etc. This is not to say that the problems of designing databases to hold these documents, and assuring their quality are not difficult. IR systems do, however, offer a powerful and flexible means of coping with these problems.

It appears probable that given the vast ammount of software to be reused, future IR systems for software reuse will need capabilities for massive storage in the gigabyte range, and specialized hardware for text searching, and set combination. Such systems will need better semantic representation of records, and will need to provide intelligent interfaces that will guide users in system use.

Searching Methods and Tools	
Alternatives	Tools
Natural Language	Catalog (Bell Labs.)
Structured Query	Automated Library Systems (GTE)
Browsing	Hypertext
Hierarchical	IMSI Smalltalk
Semantic Search	Alicia (RADC)
Citation Search/Call Ref.	Semantix (Winkler)

4. Evaluation

Once systems for classifying, storing, and retrieving reusable components have been developed, it will be necessary to evaluate them to determine their effectiveness. The following table provides a summary of these methods.

Evaluation Methods and Tools	
Methods	Tools
Relevance Based - recall, precision	Smart Environment
Usage	Standard Test Sets
Set Analysis - Katzer, et al.	Attitude Measurement Tools
Attitude Survey	
Reuse level	

One way to evaluate such systems is simply to observe whether and how often people use these systems to locate and retrieve reusable components. While unsatisfying from a scientific point of view, this method is the defacto standard for evaluating most software systems. An adjunct to this method would be to carry out attitude surveys among the users of a system to determine needs not addressed by the system.

The traditional way of evaluating an IR system in a laboratory setting is by means of recall and precision measures. Recall is the ratio of relevant documents retrieved over the number of relevant documents in the database. Except for tiny collections, this denominator is generally unknown and must be estimated. Precision is the ratio of number of relevant documents retrieved over the total number of documents retrieved. Numerous experiments using recall and precision have been done. Some of the better known were carried out using the SMART system and its variants developed by Salton and others at Cornell University (Salton 1971).

One source of difficulty with studies based on recall and precision measures is that both require judgements about a document's relevance. Such *relevance judgements*, which must be made by human judges, are unreliable. These problems have led some to the view that evaluation experiments based on relevance judgements should be abandoned. A detailed discussion of this issue can be found in (Salton 1983) (Sparck-Jones 1981).

Overlap measures quantify the uniqueness of sets of documents retrieved by different methods (Katzer, 1983). As stated above, experiments done using overlap measures have shown that different indexing methods result in the retrieval of different documents. Further analyses that attempt to divide the retrieved sets into relevant and non-relevant, however, are also based on relevance judgements with all of the attendant problems.

References

Date, C. J. *An Introduction to Database Systems*, 3rd Ed. Reading, Mass., Addison Wesley, 1981.

Devanbu, P., Selfridge, P., Ballard, B., Brachman, R., "A Prototype of an Intelligent Software Reuse Librarian", Personal Communication.

Fox, Christopher, "Future Generation Information Systems," *Journal of the American Society for Information Science* 37(4), July 1986, pp. 215-219.

Frakes, W. B. and Nejmeh, B. A., "An Information System for Software Reuse", in Tracz, W. (Ed.), *IEEE Tutorial: Software Reuse: Emerging Technology*, IEEE Computer Society, 1988.

Katzer, Jeffrey, McGill, Michael, Tessier, Judith, Frakes, William B., and Das Gupta, Padmini, "A Study of the Overlaps among Document Representations", *Information Technology : Research and Development*, January, 1983.

Lancaster, F.W., *Vocabulary Control for Information Retrieval*, Washington, Information Resources Press, 1972.

Lancaster F. W. and Fayen, E. G. *Information Retrieval On-Line*, Los Angeles, Melville Publishing Co., 1973.

Maron, M. E., "On Indexing, Retrieval and the Meaning of About", *Journal of the American Society of Information Science*, January 1977, pp. 38-43.

McCune, B. et. al. "RUBRIC: A System for Rule Based Information Retrieval", *IEEE Transactions on Software Engineering*, 1985.

Prieto-Diaz, R., and Freeman, P., "Classifying Software for Reusability", *IEEE Software*, v.4, no.1, January, 1987.

Rosales,R., Mehrotra, P. "MES: An Expert System for Reusing Models of Transmission Equipment", *Proceedings of the Fourth International Conference on Artificial Intelligence Applications*, SanDiego, 1988.

Sager, Naomi, "Information Structures in Texts of a Sublanguage", *Proceedings of 44th ASIS Annual Meeting*, Washington D.C., October 1981.

Salton G. *The SMART Retrieval System: Experiments in Automatic Document Processing*, Englewood Cliffs, N.J., Prentice-Hall, 1971.

Salton G. and McGill M. *Introduction to Modern Information Retrieval*, New York, McGraw-Hill, 1983.

Smith, L., "Artificial Intelligence and Information Retrieval", in Williams, M. (Ed.), *Annual Review of information Science and Technology*, Elsevier, 1987.

Sparck-Jones, K., Ed., *Information Retrieval Experiment*, London, Butterworths, 1981.

Winston, Patrick Henry, *Artificial Intelligence* 2nd Ed., Reading Mass., 1984.

Classification and Sofware Reusability: Research Issues

Ruben Prieto-Diaz
The Contel Technology Center

One of the problems in reusing software from an existing collection is the dificulty in finding similar components to partially match the target component. Typically, it is very unlikely that one will find a reusable component that exactly matches all of the requirements. Once a list of similar components has been retrieved, a potential reuser faces another problem: to select the candidate component that offers the least reuse effort, that is, the one that is the easiest to integrate into the new system.

A partial solution to the similarity problem is using faceted classification. It has been demonstrated that a faceted classification scheme has features that make it very attractive for classifying reusable software. The arrangement, for example, of facet terms by some conceptual relationships offers an indirect measure of similarity during retrieval. Other features include: the ease of extensibility necessary in continuously expanding collections, its precision in classification to create specific descriptors during retrieval, and its tabular format for ease of implementation.

A research issue, however, is the creation of faceted schemes. It has been observed that several specialized classification schemes are more effective for reusable software than a single universal scheme. A systematic procedure from library science, used for deriving faceted classification schemes for specialized collections, has been adopted. This manual process consists, briefly, of grouping related terms from a sample

of selected titles, defining facet names from such groups, and ordering the terms within each facet. Term ordering is user-defined and is based on conceptual relationships among terms. Multiple executions of this process, as required for a reuse library, may be a very demanding manual task. A recurrent activity in this process is the grouping and regrouping of terms during facet definition. It is, basically, a conceptual clustering analysis that requires several iterations. Conceptual clustering is a new area of research where significant, yet modest, progress has been made. A short term alternative to building multiple faceted schemes is the development of highly interactive tools to support the clustering process of facet definition and term ordering.

Another research issue is the question of selecting the best reuse candidate. Although we have tried some empirical approaches, the issue is that of representation. What is the best representation that offers ease of understanding? Providing the reuser with a representation that makes selection easy is a key factor. Related to this issue is the reuse of higher forms of abstraction of software workproducts like designs and specifications. We can reduce the selection problem by representing designs and specifications as standard models. The selection problem changes from selecting the most similar component to selecting the one that best fits in the standard model or architecture. There is a need, therefore, for research in identifying, capturing, and organizing software development information to support the creation or discovery of standard models and architectures. This process is called domain analysis.

Impact Analysis and Software Reuse

Steve Wartik
Software Productivity Consortium

A software system is the result of many decisions. Each step of analysis, design and coding involves identifying a problem, posing one or more solutions to that problem, and choosing the "best" solution. The criteria for making the choice depend on the goals of the system, the development organization, and the software development process. If, for instance, one's objective is to product reusable software, then one can identify a specific set of questions that are helpful in making the decisions. These questions are concerned with abstraction, information hiding, and other factors currently seen as important in reuse.

Obtaining answers to such questions is difficult. The design must be carefully crafted if the questions are to be properly phrased. Most developers have difficulty anticipating the directions a system (or its design) will take, and the implications of those directions on a given decision. Often, the answer only becomes clear through hindsight, and the costs of applying that hindsight are unacceptably high. Software designers need tools that help them understand the impact of each problem and a solution to that problem.

Impact analysis technology can help in obtaining the answers. Impact analysis attempts to help people understand the impact of a proposed change, *before* the change is made. The resulting knowledge aids in deciding the effect of a change in light of a specific factor. For example, suppose designers are trying to decide between two algorithms. They might, through impact analysis technology, discover that one results in more intermodule dependencies than the other, and consequently would yield code that is less reusable.

This talk will discuss the relation of impact analysis technology to reuse. Topics will include the environmental needs of impact analysis (focusing on information models), and the ways in which impact analysis helps in producing reusable system components. The role of impact analysis technology at the Software Productivity Consortium will be covered, including a discussion of automated impact analysis tools.

User Modeling for Software Reuse

Nicholas J. Belkin

Significant problems for information retrieval systems in general are the issues of how best to represent the problems which people bring to the system, and the information within the system, and how to compare these representations for searching and retrieval from the data base. I propose here that an appropriate way to approach all of these problems is by gaining an understanding of the tasks for which the information is going to be used, and of how the user of the system thinks both about those tasks, and about the structure of knowledge or

information in the domain. In particular, in the software reuse case, this will entail understanding how programmers think about the task of programming, how they themselves break down programs into units, and how they talk about those units. It will also entail an understanding of how they think about the organization of of units within programs, and of what it is that they construe as software reuse. Attaining understanding of these, and related issues is, in effect, constructing a model of the software reuser. This presentation will demonstrate how such models can be used to attack the problems of information retrieval for software reuse.

BIOGRAPHIES

Nick Belkin has been a Professor in the School of Communication, Information & Library Studies at Rutgers University since 1985. Previously, he taught at the City University, London, and was a visiting professor and researcher at the University of Western Ontario and at the Free University Berlin. He has been conducting a research program in Intelligent Information Retrieval in all these places since 1978, with special emphasis on dynamic and adaptive user modeling. He is currently working on a project to apply some of the knowledge elicitation and modeling techniques developed in his general research to the software reuse environment.

Bill Frakes is supervisor of the Intelligent Systems Research Group at AT&T Bell Laboratories, Holmdel New Jersey and an adjunct member of the Computer Science Faculty at Columbia. He has an M.S. from the University of Illinois at Champaign-Urbana, and an M.S. and Ph.D. from Syracuse University. Bill's research interests are in the areas of intelligent systems and software engineering. Bill is co-editor of *ACM SIGIR Forum.*

Ruben Prieto-Diaz recently joined The Contel Technology Center, Fairfax VA, as Principal Scientist. He was previously at GTE Laboratories where he was Senior Member of Technical Staff in the Software Reuse Project. Dr. Prieto-Diaz's research interests are in software reusability with emphasis in library systems, classification, and retrieval. He holds a BS in aerospace engineering from St. Louis University, a MS in engineering design and economic evaluation, and a MS in electrical engineering both from University of Colorado, Boulder, and a PhD in computer science from University of California, Irvine.

Steven Wartik received his B.S. in Computer Science from the Pennsylvania State University in 1977, and his M.S. and Ph.D. in Computer Science from the University of California at Santa Barbara in 1979 and 1984, respectively. From 1981 to 1984 he was employed by TRW; from 1984 until 1988 he was an Assistant Professor at the University of Virginia. He is currently a member of the Software Productivity Consortium in Herndon, Virginia. His research areas are in software environments, software configuration management, and software specifications.