A Reverse Engineering Web Portal

Congrong Liao (congrong.liao@drexel.edu)
Spiros Mancoridis (spiros.mancoridis@drexel.edu)
William Mongan (william.marc.mongan@drexel.edu)
Department of Computer Science
Drexel University
Philadelphia, PA, USA

Abstract.

The software engineering research community has developed a suite of useful tools that run on a variety of platforms and have different user interfaces; as a result, they are sometimes difficult to configure, use and administer. Being research tools, they also undergo a great deal of change over time. Typically, the users of these tools are responsible for integrating and maintaining these tools as well as dealing with constant updates. In this paper, we present the idea of "software analysis tools as services." We create a portal web site called REportal that requires only an internet browser to deliver reverse engineering services to end users. REportal features a simple and friendly user interface that hides the complexities of the underlying tools. The goal of REportal is to integrate and deliver reverse engineering tools to facilitate the transfer of reverse engineering technology to professional software developers, educators and researchers. The portal can be accessed at http://reportal.cs.drexel.edu.

Keywords: reverse engineering tools, on-line application, portal web site

1. Introduction and Motivation

Large volumes of code are typically difficult to understand. This problem is made worse by the lack of good documentation for most software systems. Studies show that about half of the time spent making changes to software is spent on understanding the software itself [21]. Automated reverse engineering techniques help developers by reducing the amount of manual source code analysis needed to understand a system. Reverse engineering is the process of analyzing a subject system to identify its components and their relationships, creating a representation of the system at a higher level of abstraction than what exists in the source code. Specifically, reverse engineering provides an aid to the comprehension of complex systems [21].

The Software Engineering Research Group (SERG) [5] at Drexel University along with the AT&T Research Labs have developed several reverse engineering tools over the years, among which are the Bunch software clustering tool [4, 3, 32, 33], the Acacia source code analyzer for C/C++ [41], the Chava source code analyzer for Java [14], various

tools for profiling C/C++/Java programs, and the dotty and Grappa graph visualization tools [20, 6].

Even though many of these tools are developed in Java, which makes them portable across platforms, their installation and use can be cumbersome. For example, in order to install Bunch, one must first download `bunch.jar`. The installation requires the Java Runtime Environment (JRE). If the JRE is not present, it must be installed. One must set the classpath environment variable so that the Java Virtual Machine can find and load Bunch's class file, and install `GraphViz` from AT&T [2] to view the clustered graphs. This entails going to the AT&T Research Lab web site, finding the correct version of GraphViz for the desired platform, downloading it and installing it. Even after everything is set up properly, the user must periodically check for software updates of Bunch, GraphViz and the JRE in order to ensure that the latest versions are being used.

The difficulty of installing and updating these reverse engineering tools might prevent professional software engineers, educators, and other researchers from using them. Since these tools have been developed over several years by different people, the usability, reliability and programming interface of each tool may differ.

1.1. Our Research

In this work, we present the idea of "software analysis tools as services." We created a portal web site called REportal, which is a front-end to a repository of reverse engineering tools. Instead of downloading these tools to each client's machine and running them locally, we put these tools on a server and expose the features of these tools as web-based services. REportal enables authorized users to login to the site and upload their code to the server through a secure channel. REportal features a simple and consistent user interface that hides the complexities of using the underlying tools natively. Through a web browser, user requests are sent to the server. The server then executes the appropriate tools and sends the results to the user's browser for display. In this way, all users are able to analyze and browse source code, as well as to query and extract design information for C, C++ and Java programs.

1.2. Research Challenges

This work addresses the following research challenges:

- **Integration of complex interfaces**. REportal is the first comprehensive code browsing and analysis service available on the

WWW. As a result, there is no template that we can follow. Many challenges involve providing enough features to users while keeping the user interface simple. REportal consists of many integrated tools, which have different interfaces and operating system platform requirements. Some of these tools are written in Java and have comprehensive APIs, while others are Unix command-line utilities. Providing a simple and intuitive user interface by hiding the complexities of these tools is one of the challenges that this work addresses.

- **Limitation of web browsers**. The only software needed to use REportal is a Java-enabled web browser. User requests are sent to the server via the web browser; and the results of the requests are sent back to the browser to be displayed. The idiosyncracies of different web browsers limit the implementation of REportal in many ways. Some techniques, such as Dynamic HTML (DHTML) provide capabilities to enrich the user experience, but these implementation choices also limit the number of browsers that can be supported. There are many types of web browsers, among which are the popular Internet Explorer, Netscape and Mozilla. Some features are supported by one browser but not another. Other features (e.g., JavaScript) behave differently on different browsers. Our goal is that all features of REportal be supported by all popular web browsers such as IE6.0, Netscape6.0, and Netscape6.2, on all popular platforms such as Windows 95/98/00/XP and Linux. This goal limits our implementation choices, but enables REportal to be used by a large community of users.

- **Security**. Since REportal requires users to upload their source code to our server, it must provide a secure environment so that users are confident enough to use it. Security issues include: secure user authentication, secure transfer of code to the server and secure display of results. Security is provided by using standard SSL technology through a registered server-side certificate.

1.3. Paper Outline

The rest of this paper is organized as follows: Section 2 provides an overview of related work. Section 3 describes the functionality of REportal from a user's perspective. The architecture and developer's perspective of REportal is described in Section 4. The validation of RE-

portal is covered in Section 5. The paper concludes in Section 6 by summarizing our current work and outlining our plans for future work.

2. Related Work

There are several research areas that are relevant to this work: web-based software engineering, source code analysis, design extraction, and software visualization.

2.1. Web-based Software Engineering

Web-based software engineering tools fall into the following categories:

- A knowledge base of information about reverse engineering, program understanding, and software evolution.
- A repository of software packages that can be downloaded and run locally.
- A service provider in an area other than reverse engineering, such as graphing.

Reengineering Wiki [29] is a forum where all topics related to Reverse Engineering and Reengineering can be discussed. It also contains a collection of papers, tutorials, and surveys.

Software Bookshelf [22] catalogues software architectures and allows users to explore them. It has a primitive querying mechanism. Unlike REportal, the Software Bookshelf does not support source code browsing.

The graph drawing server at Brown University provides interactive graph drawing via the WWW [9]. It offers two kinds of services:

1 drawing graphs using a user-specified algorithm;

2 translating the description of a graph from one format to another [37].

Stephen North designed a graph service called *drawdag* [7]. The service accepts a dot file and layout format from users via email. The server constructs a drawing with dot [6] and sends the output to the users via e-mail.

Sourceforge.net [36] is one of the largest open source software development web sites. It supports source code browsing, discussion, bug/defect tracking, and a documentation repository.

The closest work to REportal is LintPlus Online [16] by Cleanscape. It provides online source code analysis for C and Fortran programs. It displays the compiling, call tree, and include-file trees in a text format. However, it does not support graph visualizations, interactive querying, or design extraction.

2.2. Code Analysis

Code analysis involves building repositories from a system's source code so that these repositories can be used for a variety of reverse engineering analyses. The repositories are useful because sophisticated reverse engineering tools can be built by analyzing the information stored in the repository without reparsing the system's source code. There are some commercial code analysis tools such as Red Hat's Source-Navigator [28] and Netcomputing's AnyJ [19].

Two popular approaches exist to construct software repositories. One is to store variants of abstract syntax trees in the repository; the other is to structure the repository as a relational database [41]. Once the repositories are constructed, queries can be made to the repositories in order to extract structural information about the source code.

Over the past several years, the AT&T Research Lab has developed a family of source code analysis tools such as Acacia and Chava. Acacia and Chava output two repositories: the entity repository and the relationship repository. The entity repository stores the entities such as file names, variables, functions, and classes along with their attributes such as scopes, line positions, and so on. Likewise, the relationship repository stores the relationships between entities, such as function calls, inheritance, and variable references.

A variety of Unix command line tools are available to query the repository, answering questions such as:

- Is variable a defined in file b?
- Is function a referenced by function b?
- What are the child classes of class a?

Advanced analyses such as dead code detection and reachability analysis can be applied to the repositories as well. The query results are displayed in text format. The work described in this paper uses Chava to create the repositories.

2.3. Design Extraction

Software systems often need to be modified to improve their performance, add new features, adapt them to new platforms or hardware, and so on. To modify a software system, developers have to understand the system. As the size and complexity of software systems increase, the time spent on understanding software systems increases as well. In most cases the relevant design documentation is missing or inconsistent. Therefore, tools that can provide a high-level system decomposition become very helpful to facilitate the comprehension of software systems.

The principle artifact that must be examined is the system's source code. Thus, a major task in reverse engineering is to build an abstract model of a software system from its source code.

Mancoridis and Mitchell developed a software tool called Bunch [4, 32], which automatically decomposes the structure of software systems into subsystems. Modules with high cohesion are grouped in the same subsystems (clusters), and independent modules are grouped into separate subsystems. The modules and dependencies of a system are mapped to a Module Dependency Graph (MDG) using source code analysis tools such as Acacia [41] for C and C++ and Chava [14] for Java. The goal of Bunch is to find a good partition of an MDG graph. It is the first system to employ meta-heuristic search algorithms to the software clustering problem [13].

Mancoridis and Mitchell introduced an objective function called Modularizaion Quality (MQ). The MQ rewards the creation of highly-cohesive clusters, and penalizes excessive coupling between clusters. Hence, Bunch treats the software clustering activity as an optimization problem where the goal is to maximize the value of MQ. The assumption behind MQ is that most software systems are designed in such a way that highly cohesive modules are organized into the same subsystem while loosely coupled modules are organized into separate subsystems.

2.4. Graph Drawing for Software Visualization

Visual presentations can ease the understanding of complex systems. Not surprisingly, extensive research has been conducted on how to store, layout, and display graphs that represent software systems.

Barghouti and Mocenigo developed an extensible graph drawing package written in Java, called Grappa [20]. It consists of a set of classes that implement graphs, in addition to representation and presentation services. It also provides an API so that it can be integrated into applications that require graph drawing, editing, and browsing. The second version of Grappa, in addition to supporting enhanced viewing features,

is able to handle large graphs, which the first version of Grappa could not handle. REportal integrates Grappa as an applet.

Grappa invokes *dot* [6], a graph layout tool. *Dot*, which runs fast enough for interactive use, is a command-line utility that takes a *dot* description file as input, and produces an output file where the nodes are assigned a position in a 2D space. By default, *dot* positions nodes to minimize edge length and edge crossings. Grappa renders a graph in a Java applet based on the layout information produced by *dot*. *dot* can also transform its output into a standard image file such as GIF, PS and PDF.

3. REportal: A User Perspective

The main services of REportal are presented along with usage scenarios that illustrate how REportal can be used to solve a variety of software engineering problems.

3.1. Introduction to REportal

As mentioned earlier, REportal is a web-based application that integrates many stand-alone software engineering tools. The output of these tools is presented in a web browser. The current version of REportal provides the following services:

- **Registration & Account Maintenance**. REportal enables any Internet user to create an account. All services provided by REportal are executed under a user context. Consequently, users can only analyze systems that they upload to REportal.

- **Code Analysis**. REportal provides analysis services for Java code. This service analyzes programs and generates repositories of information about the system. Using the repositories, code analysis involving querying or clustering is possible. Once created, the repository is associated with a particular project, enabling users to perform additional analysis without having to upload and analyze the source code again.

- **Code querying and browsing**. REportal allows users to perform entity or relationship queries that explore structural information about the software system. In addition, REportal can display fully cross-referenced source code in a web browser using HTML hyperlinks.

- **Design recovery and visualization.** REportal integrates the Bunch tool [4, 32, 33], which partitions source-level structures into subsystems.
- **Supporting services.** REportal provides several support services that make the above features easier to use. These services include user authentication, content-sensitive help, and user workspace management (e.g., creating, renaming, deleting, uploading and downloading files or folders).

3.1.1. *Installation*

Although REportal does not require users to install the reverse engineering tools, the client's workstation configuration must support JavaScript, Applets, Cascading Style Sheets (CCS), and HTML. Users must verify the following before using REportal:

- Java Runtime Environment (JRE) 1.3.1 or higher is installed locally.
- The web browser supports Java and JavaScript.
- The latest version of web browsers is highly recommended. We have tested REportal using Internet Explorer 6.0 or higher, and Netscape 6.0 or higher.
- The web browser supports Secure Sockets Layer (SSL).

3.1.2. *Login and Sign Up*

REportal uses SSL to encrypt the transmission of source code and user authentication information. When users go to the REportal website at http://reportal.cs.drexel.edu [24], they are notified that they are about to view pages over a secure connection.

To create a REportal account, users must provide their first name, last name, desired username, password, company name, and email address.

As soon as the request is submitted, a welcome email message is sent to the user's mail box, and an account is created automatically. Once users have an account, they are eligible to login to REportal by providing a valid username and password.

3.2. Workspace Management and Navigation

Authenticated users have their own workspace with some management privileges. Users are permitted to do the following:

- Create folders for projects.
- Upload source file packages (e.g., zip or jar) to their private workspace.
- Download files or folders to their local machines.
- Rename or delete projects.
- Upload graphs to the *graphs* subfolder.
- Browse the entire workspace.
- Select a project and analyze it.

Workspace Management

Users create folders in their workspaces; each folder holds one project. Once a folder is created, the *src* and *graphs* subfolders are created automatically. Although users are permitted to create additional folders for other projects, they cannot change the structure of the folders, meaning that they cannot delete, rename or create subfolders (Figure 1). All operations are performed by selecting an appropriate item from a context menu, which is activated by right-clicking on a folder or a file (2).

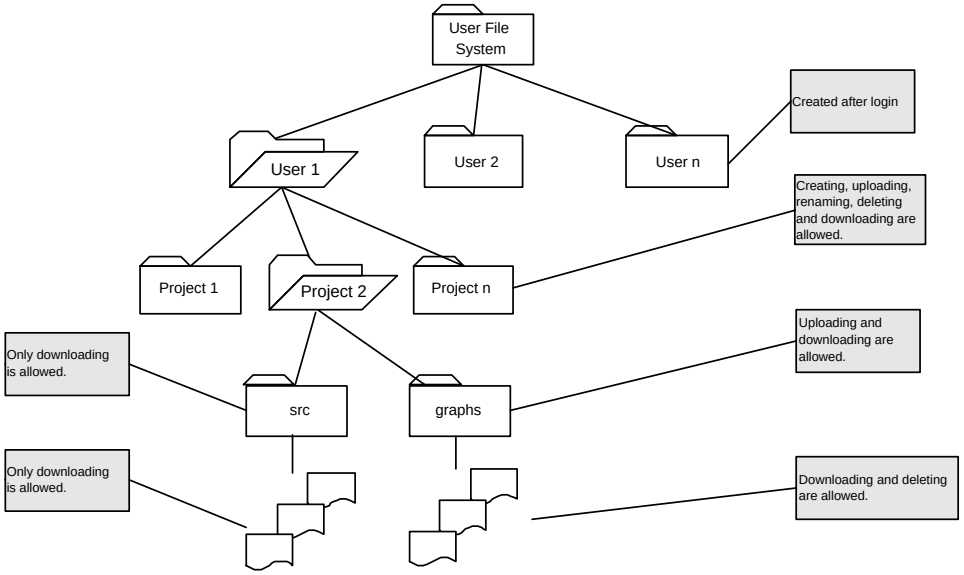

Figure 1. Structure of user's file system

Once a folder is created, users can upload zipped source code to the folder. All source code must be zipped into a single file. The user must

upload Java bytecode and, optionally, the accompanying source code. Java bytecode analysis can be performed by inspecting the bytecode. Java source code is required to take advantage of REportal's source code browsing feature; however, REportal will attempt to decompile the byte code if the source is not available.

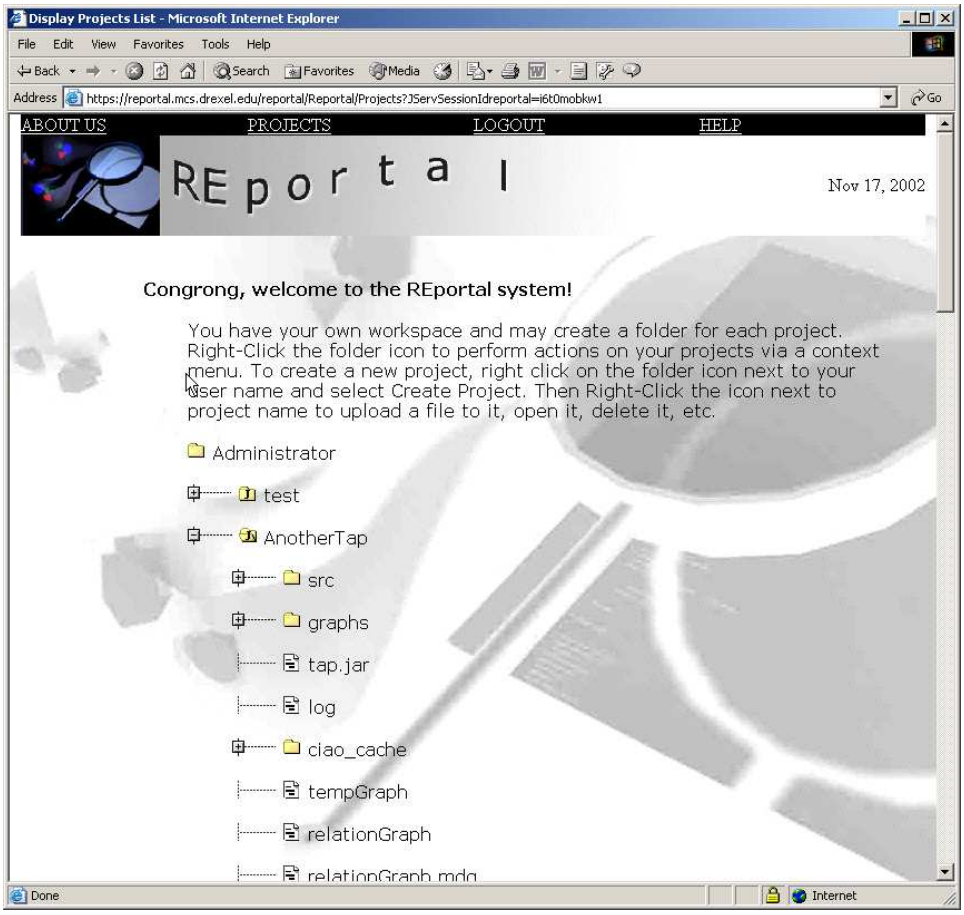

Figure 2. Workspace display and navigation

After the file is uploaded, it is upzipped into the *src* subfolder automatically. Then, REportal automatically generates the Chava repositories.

3.3. Code Querying and Browsing Service

Once users open a project, the query window is displayed and all tools are listed along the top of the window as tabs. The `Entity Search` tool is enabled, and all files in the selected project are listed in a table.

Hyperlinks to the source code browsing feature of REportal are associated with the files that have source code available. For example, clicking on the `TAP/TAPClient.java` link will direct REportal to display this file's source code. This source code is fully cross-referenced to facilitate browsing and querying.

3.3.1. *Code Querying*
The Code Querying service provided by REportal features: entity, relationship, text, and advanced searching.

Entity Search

For Java programs, entities represent files, classes, methods and fields. In an entity search, the user can select the entity type (file, class, method or field) and specify the name of the entity. Wild card characters may be used for entity names as described in Table I.

Table I. Wild Card Characters Accepted by Entity Search

Name	File
?	Matches any single character
*	Matches any sequence of zero or more characters.
[x...y]	Matches any single character specified by the set x...y.
	A minus sign may be used to indicate a range of Characters.

Query results are displayed in a table that has two fields: the name of the entity and the file in which the entity resides. Figure 3 illustrates a query of all methods in the Ticket Auction Program (TAP).

The entity search can answer questions such as:

- What classes are defined in the project?
- What are the methods whose names start with "b"?
- Where is the field "fieldName" defined?
- What files have source code available?

Relationship Search

A system-level relation is an association between two entities, such as an inheritance between two classes, an invocation between two methods, or a reference to a variable by a method. A relationship query takes two parameters: a source and a destination entity. The query returns a

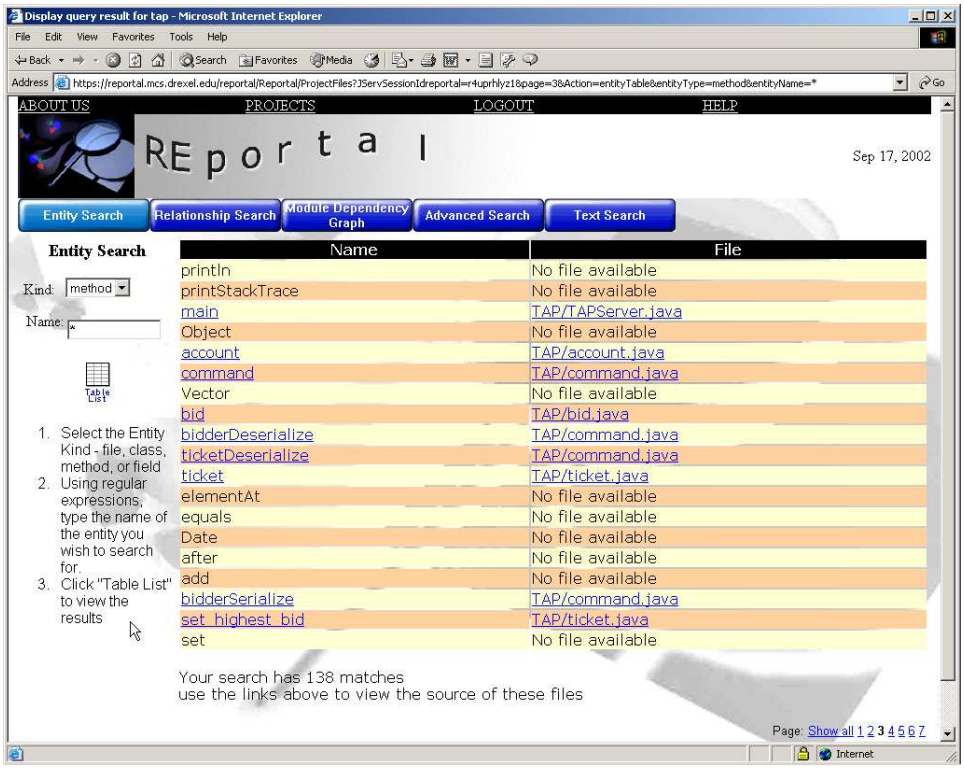

Figure 3. Entity Search

collection of directed relations from the source entity to the destination entity. For example, to list all entities that refer to a particular entity, users specify the destination entity and leave the source entity as a "*". Some examples of relationship queries are shown in Table II.

Query results of a relationship search can be listed in a table. The table has four columns for the names of source and destination entities, and the files in which they reside. If the source code of a file exists, hyperlinks are associated with the entity and the file name containing the entity. Figure 4 shows all the relationships between classes in the `TAP` system.

Query results can also be displayed as a graph (Figure 5). If users choose this option, a new window appears with the graph displayed using an applet in the center panel and a bird's eye view of the graph above it. In the graph, a node represents an entity, while an edge between two entities represents a relationship.

Several options are available from the graph display window. The Bird's Eye View (BEV) can be used to navigate through large graphs.

Table II. Examples of Relationship Search

Query	Target type	Target name	Destination type	Destination name
All entities that refer to method *bid*	All	*	method	*bid*
All entities that method *bid* refers to	method	*bid*	All	*
All classes that inherit from classes starting with a 'b'	class	*	class	b*
All fields ending with a 'c' being referenced by class 'c'	class	c	field	*c

Clicking within the BEV centers the graph around a point. Dragging a box within the BEV zooms the graph around that box.

The graph viewer can display the graph at different levels of abstraction by using the Bunch clustering system [4, 32]. This is suitable for graphs of many nodes and edges that are difficult to comprehend. Users may cluster the graph from within the graph window and take advantage of other graph viewing features using the following buttons:

- The **Standard Graph View** button provides the default, unclustered view of the graph as seen in Figure 5.
- The **Clustered Graph View** button displays the clustered version of the graph. Nodes are grouped into subsystems and displayed in the graph window.
- The **Interactive Cluster View** button also displays the clustered version of the graph. However, in the interactive view, clusters are represented by default as individual gold-colored nodes (Figure 6). An extra Bird's Eye View is shown to display the "top level graph," in which all the clusters are collapsed into individual clusters. Clicking on a cluster node expands that cluster to show its

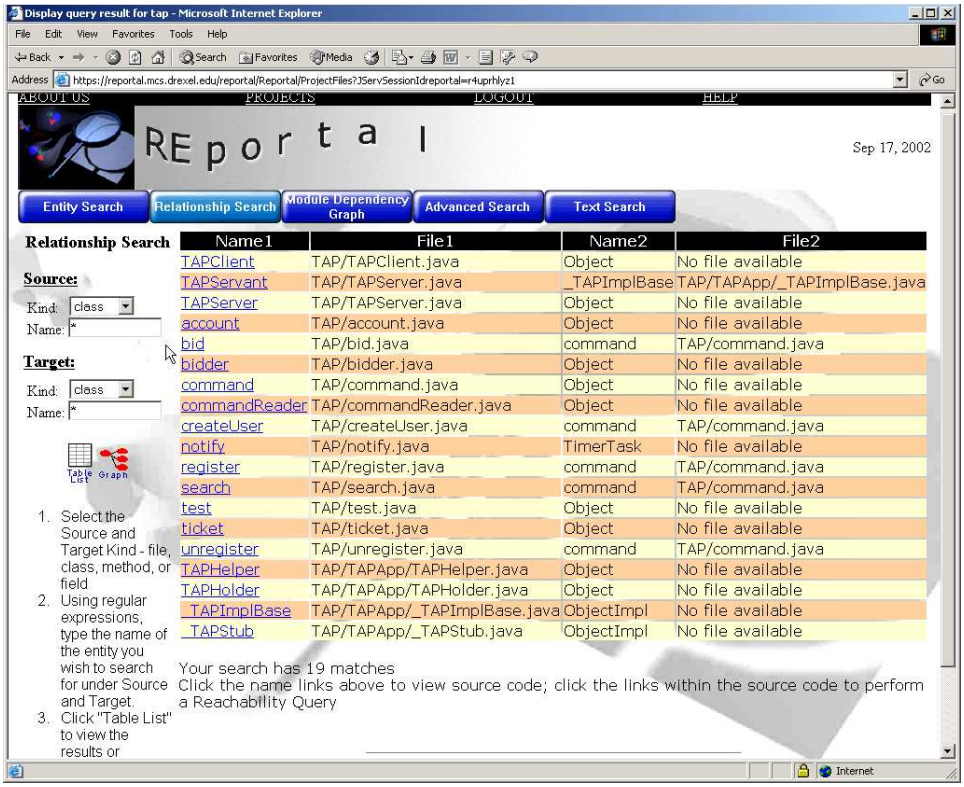

Figure 4. Relationship search

contents. Edges between gold cluster nodes indicate that at least one edge exists between nodes of those clusters.

– The **Advanced Clustering** button allows users to customize the clustering process by using the Bunch user interface directly. For example, they can exclude some modules from the clustering process and can upload a file that restricts the placement of particular modules into certain clusters.

– The **Reachability Query** button allows users to determine which entity can be reached by a particular entity, or which entity another entity can reach. A reachability query is performed by clicking on the corresponding node in the applet, and clicking the "Reachability Query" button. Users may query what can be reached by the entity by specifying the direction as *forward*; or they may query what can reach the entity by specifying the direction as *backward*. Users can also limit the type of entities that are reachable to/from a entity. For Java programs, entity types are packages, files, meth-

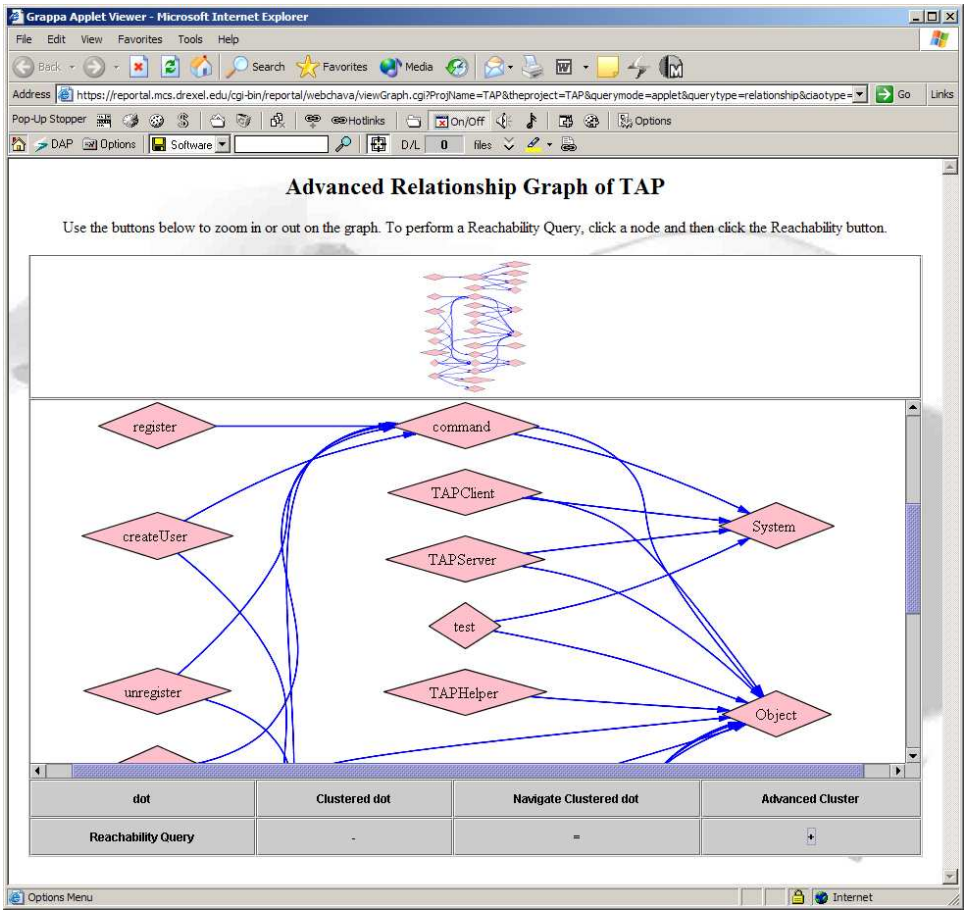

Figure 5. Display results as a graph

ods, classes, fields, strings, and interfaces. Users may also specify the depth and output format of a reachability query.

- The - button zooms out the graph.
- The = button resets the zoom scale to 1:1.
- The + button zooms in the graph.

Text Search

Text search allows users to search the entire source code of the system for a text pattern. It accepts wild cards as described in Table I. A user can, for example, search for all the source code lines that contain the text "main," or a text pattern that begins with "ma" and ends with "n".

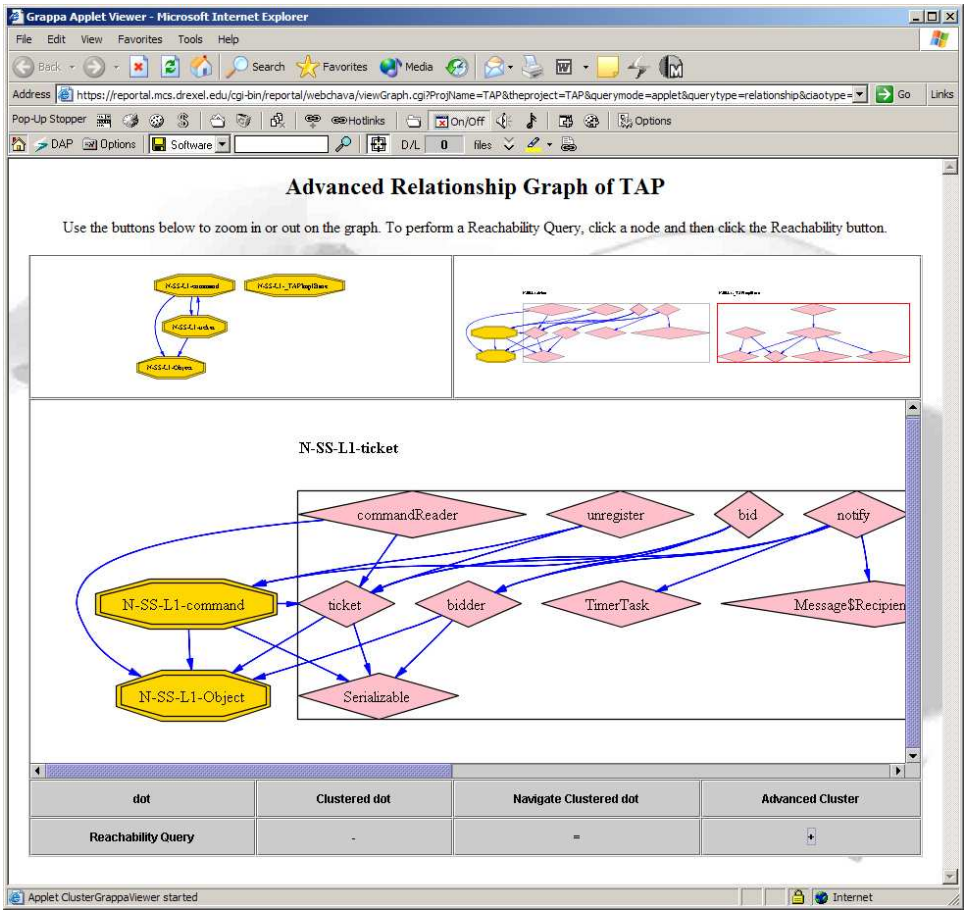

Figure 6. Interactive Bunch cluster view of the graph

The query results are displayed in a table. The table contains three columns: the file in which the text/pattern resides, the line number of the text/pattern in the file, and the contents of the line. As usual, if the source code of a particular file is available, hyperlinks are placed on the file name.

Advanced Search

The entity and relationship searches allow queries that accept entity types and names. More complicated queries can be performed by using the `Advanced Search` feature of REportal.

Not only can users specify entity names and types, but they can also specify the file in which the entity resides, the scope of the entity (*protected*, *public* or *private*), and the type of query (either an entity or a relationship search). Query results are displayed in tables with more

information about entity names, scopes, files in which they reside, and beginning and end line numbers of each entity.

With the additional capabilities of the advanced search feature, users can perform more complicated queries such as:

- Which private classes implement class *bidder*?
- Which classes in file *TAPServer* reference the class *bidder* in file *bidder*?
- List all of the private methods in file *search*?
- How many files does the package *bunch* contain?
- List all interfaces that are defined in files beginning with the letter "C"?
- Is the *run* method in the class *command* referenced by the *notify* method in class *mail*?

3.3.2. *Code Browsing*

If users provide source code when uploading their systems to REportal, they can browse the source code in a web browser. Like many commercial Integrated Development Environments, different types of information in the source code are displayed in different colors for easy reading. For example, line numbers in front of each line of code are displayed in a light gray indicating that the line numbers are not part of the source code; reserved words are displayed in green, comments in red, and the other source code in black.

When source code refers to a program entity, a link is placed in the browser window. This link allows users to perform a reachability query that involves the selected entity.

3.4. Design Recovery

REportal uses the Bunch clustering tool to infer high-level design information about a system. REportal mimics the stand-alone version of Bunch, so that the user's experience in Bunch can be carried to REportal, or vice versa.

Using Bunch

The Bunch software clustering service provides a web page that mimics the user interface of Bunch. On top of the "Option" window, there is a set of tabbed panels, which are "Basic," "Options," "Libraries," "Omnipresent," "User Directed" and "MQ Calculator."

When Bunch is launched, by default, the "Basic" option window is displayed, where users can specify input graph files in MDG format and select a clustering method. MDG stands for Module Dependency Graph, where nodes are classes or modules and edges are invocations or usage of a class or module. Clustering a graph requires users to select an MDG graph from the drop-down list of the "Input Graph File" tab, which lists all graphs in the "graphs" folder of the project. When a graph is selected, the "Run" button at the bottom becomes active indicating that clustering is allowed. MDG files can be created by two means:

1. Locally create a MDG file either manually or automatically, then instruct REportal to upload the file to the "graphs" folder.
2. Alternatively, users can create custom MDG files on REportal directly by clicking the "Create" button. Users can customize the MDGs by excluding or including method-to-variable relationships, package names, weights on relationships, method-to-method relationships, implementation relationships, and/or relation types.

Once the MDG file is created, it is selected as the input graph file automatically. Users can then run the clustering services. After the clustering finishes, four output files are generated for viewing – "dot", "pdf", "ps" and "gif". In addition, the "Download" and "View" buttons in the "Basic" window are enabled. Users may then choose one of the four formats and download an image of the clustered graph to their local machine. Alternatively, they can view graphs online using an applet.

Clustering method

"Bunch" supports two clustering methods: Hill-Climbing and Genetic Algorithm (GA). Users can choose either from the "Clustering Method" list. Hill-Climbing is the default clustering method. Hill-Climbing works best for most graphs, while the GA sometimes is more efficient than Hill Climbing for extremely large graphs.

Users also have opportunities to configure the two clustering algorithms by clicking the "Options" button. For more information about the configuration options, refer to Mitchell's Ph.D. dissertation [3].

Using libraries

This feature allows users to exclude from the clustering process those nodes that only have incoming edges (libraries). Library nodes tend to obfuscate the abstract view of the structure because many of the edges in the MDG are directed towards library modules. Thus, the library modules' placement into a cluster become somewhat arbitrary.

In REportal's *library* window, users can either select nodes manually from the list on the left (included nodes) and move them to the right (excluded nodes), or click the "FIND" button to move all libraries from left to right automatically. In the resultant clustered graphs, library nodes are placed in a special cluster.

Using omnipresent modules

Omnipresent modules are modules that either have many incoming edges or outgoing edges relative to the other nodes in the MDG. Modules that have many incoming edges are called omnipresent suppliers; while modules that have many outgoing edges are called omnipresent clients. In the "Omnipresent" window, which is activated only after a MDG file is selected, users can exclude omnipresent modules from the clustering process.

The list to the left shows all modules that may be selected as omnipresent modules. The list does not contain modules that have already been selected as libraries. Likewise, the list on the left in the library window does not contains modules that have already been selected as omnipresent. Users may select certain modules as omnipresent manually, or press the "FIND" button to have REportal suggest omnipresent modules automatically.

User-directed clustering

Users sometimes have prior knowledge about which modules should be placed within a subsystem. The "User Directed" feature allows users to upload a description file that specifies what modules should be placed together into clusters.

4. REportal: A Developer's Perspective

4.1. REportal Non-Functional Requirements

REportal was designed to satisfy the following non-functional requirements:

- **Extensibility.** As a long-term research project, REportal is designed to be a central repository of reverse engineering tools. Although it currently supports only static program analysis, more tools will be integrated to support security analysis and dynamic program dialysis.
- **Installation and Portability.** Currently, REportal runs on a server in the Software Engineering Research Group (SERG) laboratory at Drexel University. To use the services of REportal, users

must first upload their source code to the server. If security is a key concern for users, they may hesitate to do so. Hence, one feature of REportal is its portability to other servers, perhaps behind a corporate firewall. Therefore, it is important that REportal be easy to install, and adjustable to a variety of computing environments.

– **Usability.** As a portal web site that provides sophisticated software engineering services, usability is a top design concern. Providing robust, reliable services through an intuitive and friendly interface is important.

4.2. Development and Deployment Environment

REportal executes on a Unix machine that runs the Apache web server. The machine runs Red Hat Linux 7.2 on a 1 GHz Pentium III processor, with 256 MB RAM and an 18 GB SCSI hard drive. The machine has all of the necessary tools, shell scripts, and software to run REportal.

The web server that runs REportal and the techniques used to develop REportal are described in the following subsections.

4.2.1. *Apache*

The Apache server is used as the web server for REportal. Apache is powerful, flexible, and HTTP/1.1 compliant. It is available from the Apache Software Foundation at no charge and comes with an unrestrictive license [1]. Apache has been the most popular web server on the Internet since April of 1996 [1]. Apache runs on Windows NT/9x, Netware 5.x and above, OS/2, and most versions of Unix. It has been shown to be substantially faster, more stable, and more feature-rich than many other web servers.

4.2.2. *Java and Java Servlets*

Introduced by James Gosling at Sun Microsystems in 1995, Java has gained tremendous popularity over the years. As one the fastest growing programming technologies, Java has become an ideal language for server-side development of large applications such as REportal.

The cross platform nature of Java facilitates the portability of REportal. Java's object-oriented, memory-protected design reduces development cycles and increases reliability.

Java Servlets are Java classes that can be loaded dynamically to provide web developers with a simple, consistent mechanism for extending the functionality of a web server. Java Servlets have many unique features for creating dynamic web content. Many of these features overcome scalability and performance limitations that are associated with other server side technologies such as CGI and server-side JavaScript.

Portability

Java Servlets are supported on all major platforms, and work with all of the major web servers [11]. This feature enables REportal to be developed on a high-end Unix server running Apache, and to be deployed on other platforms running a different web server.

Efficiency

Unlike CGI, which uses a single process to handle each program and/or request, Servlets are all handled by separate threads that run within the web server process [11]. Once a servlet is loaded, it stays in the server's memory as a single object instance. The server invokes it to handle a request using a method invocation. This design makes Servlets efficient and scalable.

Extensibility

Servlets may access the entire family of Java APIs, including the JDBC (Java Database Connection) API, networking, multithreading, and object serialization. Thus, Servlets embody all of the benefits of Java environment, including portability, reusability, and protection [12].

Java Servlet technologies are used by much of REportal. The servlet engine used by the REportal project is Apache JServ [11].

4.3. REportal Architecture

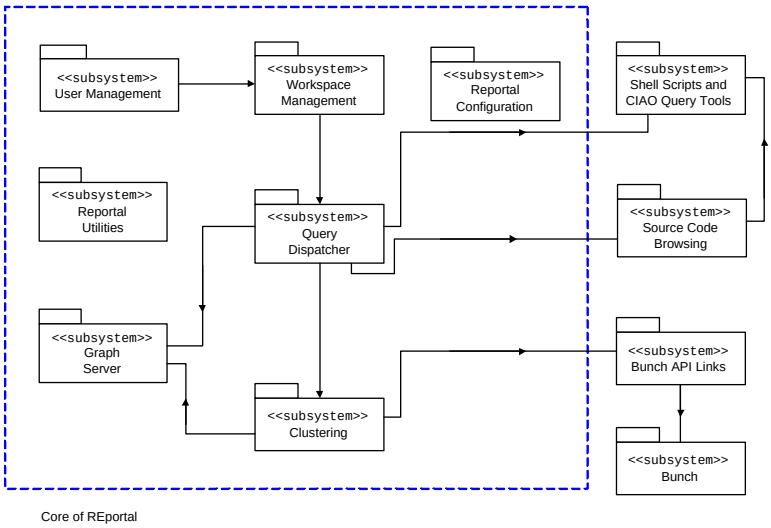

Figure 7. REportal architecture

Figure 7 illustrates the architecture of REportal, which consists of 11 subsystems. Some important aspects of the REportal architecture are described below:

- The `REportal Configuration` subsystem and the `REportalUtilities` subsystem provides system-wide services to almost all other core subsystems. For clarity, the two subsystems are not connected to the others.

- The `shell scripts and CIAO query tools, source code browsing, Bunch API Links` and `Bunch` subsystems are loosely-coupled to the other subsystems in the REportal architecture. These subsystems are integrated into REportal to support services such as code analysis, module dependency queries, and so on. We chose a loosely-coupled approach because these subsystems are likely to evolve in the future.

- Depending on the type of queries from users, the `query dispatcher` subsystem invokes the appropriate tools. In this way, future integration of other tools will require minimal effort.

The `User Management` subsystem deals with user authentication and registration. Once a user logs into REportal successfully, he enters the `Workspace Management` subsystem. All administrative services such as creating, renaming, deleting, uploading and downloading a file/folder are completed by the `REportal Utilities` subsystem. If a user selects a project and opens it, REportal advances to the `Query Dispatcher` subsystem, where the user's requests are dispatched to other subsystems. For example, entity searching requests are directed to the `CIAO Querying Tools` subsystem; clustering requests are directed to the `Clustering` subsystem.

4.4. REportal Subsystems

This section examines the pertinent design aspects of the major REportal subsystems.

4.4.1. *User Management & Configuration Subsystem*

When the REportal service is requested for the first time, a REportal servlet is created. When a client connects to the server and makes an HTTP request, the servlet produces one thread for each connection and directs the request to the `Reportal` object. The `Reportal` object serves three purposes:

- Creates an `Ruser` object, a `ReportalUser` object, and a session for the client. The `Ruser` object is associated with the session. During the creation, the configuration file of REportal is loaded in through the `REportal` subsystem. The `Ruser` objects contains information about the client such as user name, password, first name, last name and so on.

- Depending on the status of the session, it invokes methods in the `ReportalUser` object to display the login screen, to do user authentication, or to dispatch user requests to other subsystems.

- Invalidate the session when the connection is closed or the session expires. The lifetime of a session is set to 30 minutes.

If the client wants to create a REportal account, the request is directed to the `SignUp` class, which displays the sign up form. Once the client finishes and submits the form, the client's information is stored in a file on the server. The client's password is encrypted by a one-way hash function called `MD5` [25]. If the creation of the account is successful, a welcome email message is sent to the user via the Java Mail API.

It is worth pointing out that every HTTP request goes through the same path: the request is directed to the `Reportal` object, then forwarded to `ReportalUser`, which defines several states. Depending on the state of the request, it invokes objects in appropriate subsystems (e.g., `Workspace Management`, `Query Dispatcher`, or `Clustering`). Every HTTP request is processed in this manner.

4.4.2. *Workspace Management & Utility Subsystem*
The `REportal Utility` subsystem contains two primary classes: `FileUtilities` and `DisplayUtilities`. The `FileUtilities` class is a layer between the class `Projects` and user's file system. All manipulation of the file system, such as creating a directory, renaming a file, deleting a directory, and uploading a file must be done by the FileUtilities class. Currently, we assume that authenticated users have all the privileges described in Section 3.2 to manage their workspace. Once a user is authenticated, he can start managing his workspace without further validation. In the future we may require every manipulation to a user's workspace to be validated. The separation of this class allows these changes to be isolated from the other subsystems.

The `DisplayUtilities` class provides a layer on top of HTML generation. All of the HTML generated by REportal is handled by calls to this class. For example, the `openTable()` method writes a `<table>` tag and the `doHeader()` method produces a header that includes SERG's

logo and REportal's banner. This design makes the code more readable and easier to change.

The `Projects` class provides an interactive interface for users to manage their workspace with the support of the `FileUtilities` and `DisplayUtilities` classes. This class displays the user's workspace in a fashion that is similar to Windows Explorer. For example, all directories that are at the same level are aligned vertically; subdirectories are underneath the directories they belong to; directories that have subdirectories have a `+` sign in front of them; the directories are expanded if the `+` sign is clicked.

Every form in the workspace management page has a hidden variable called "action" associated with it. The value of this variable varies from `form` to `form`. When a request is sent to the `Projects` class, the request is forwarded to appropriate methods depending on the value of the variable. In this manner, expansions take minimum effort. These techniques are used in REportal's query page and clustering page.

4.4.3. *REportal Configuration*

The `REportal Configuration` subsystem contains a single class named `ReportalConfiguration`. As mentined previously, REportal uses many shell scripts and other tools to perform its tasks. This class defines the paths to these shell scripts and tools.

Whenever REportal refers to a shell script or a tool, it finds the location of it through this class. This class reads the paths from a configuration file. If the file is not found, the paths are set to default values. This design simplifies the installation of REportal on other servers where the shell scripts and tools may be located in different directories on the file system.

4.4.4. *The Query Dispatcher Subsystem*

The `ProjectFile` class provides an interactive interface to support code analysis, design extraction, and source code browsing services. This class, which is called when a project is opened, generates the "query" page. The interface of the "query" page is designed so that all services that support source code queries and browsing, and design extraction can be reached directly from this page.

The "query" page provides 7 services: a) entity search b) relationship search that lists results in a table c) relationship search that displays results as a graph d) create an MDG file e) cluster an MDG file f) advanced search, and g) text search. The `ProjectFile` class either completes the service request or redirects the service request to other subsystems. The "Query Dispatcher" sometimes calls upon subprocesses to handle these service requests. The subprocess' standard

input/output is redirected to the parent process through two streams (`Process.getOutputStream()` and `Process.getErrorStream()`).

4.4.5. *The Clustering Subsystem*
The `Clustering subsystem` integrates the Bunch clustering tool into REportal to support the design extraction service.

The `ClusterWizard` class implements the user interface and invokes the Bunch API to perform clustering tasks. The `ClusterWizard` is invoked by the "Query Dispatcher" when clustering services are needed. Users perform operations in a user interface similar to Bunch, except that calls are made directly to the Bunch API to respond to each operation. The output is retrieved from the API and displayed.

5. Validation

Computer software is designed to target a certain group of users; therefore, usability is essential to its success. Even well-designed software may be rendered useless simply because the end-user cannot use it easily. This is especially true for REportal since it is designed to provide a simple, consistent, and intuitive user interface that hides the complexities of the underlying reverse engineering tools.

Although the current user interface has evolved through many iterations, there may still exist flaws that are inconsistent with the expectations of software engineers for a portal web site. Likewise, although thorough tests have been done on REportal, it is very possible that there are some bugs that have gone undetected.

Currently, the first version of REportal is ready to be released. Before the release of REportal, we wanted to test the software to remove defects and investigate ways to improve the user interface.

Evaluating the user interface of software is not easy. Dr. Hewett from the Psychology department of Drexel University, who has rich experience in this area, gave us several ideas. According to Dr. Hewett, the easiest way to evaluate the user interface of a software system is to conduct usability studies in which a few typical users perform some tasks using the software. In the study, the users were observed to reveal usability design improvements.

5.1. Design of Tasks

The usability study was designed with the following requirements:

- The tasks should last approximately one hour, otherwise participants may lose patience.

- The tasks must be based on real-life scenarios so that the results are meaningful and convincing.
- To finish the tasks most, if not all, of the features provided by REportal should be covered.

Two software systems were chosen for this study: Apache Tomcat 4.0.4 and Bunch 3.3.5. Apache Tomcat is an implementation of the Java Servlet 2.3 and JavaServer Pages 1.2 Specifications. The open source code of Apache Tomcat 4.0.4 is available at FRESHMEAT [8] and Jakarta [10]. Bunch is integrated into REportal to provide the design extraction feature. Apache Tomcat 4.0.4 has about 40 classes while Bunch 3.3.5 has 220.

The participants were asked to answer six to seven questions for each system. These questions cover most of the features provided by REportal, such as entity searching, relationship searching, reachability querying, design extraction, source code browsing and administrative functions.

5.2. Conducting the Evaluation Study

The Software Engineering Research Group (SERG) at Drexel University consists of several undergraduate students, graduate students, and faculty. Some of them have full-time jobs in industry as software engineers. REportal contains a comprehensive set of reverse engineering tools to profile and mine the source code of software systems. The target users are researchers, students and software engineers, which can be represented by the people of SERG. Therefore, volunteers who represent typical users of REportal can be recruited from SERG.

Dr. Hewett offered a two-hour lecture for all of the participants involved in the evaluation study. The lecture covered the method by which information about users can be gained when evaluating the usability of a software system. To practice the method described in the lecture, participants were asked to complete the tasks using REportal during the information-gathering sessions following the lecture. Hence, their participation becomes a learning experience of a methodology for evaluating the usability of software,

An email message was sent to the SERG mailing list calling for volunteers to participate in the experiment. Four people volunteered to participate. Among them were one undergraduate student and three graduate students. One graduate student also has a full-time job as a software engineer.

After the lecture, each participant attended a one-hour information-gathering session, during which they finished the tasks using REportal.

Participants were encouraged to 'think aloud' and to give feedback about the user interface while completing the tasks. The authors observed and recorded problems that the participants encountered and the feedback that they gave. All information-gathering sessions were video-taped for further study.

5.3. Results of the Evaluation Study

Overall, the study went smoothly. All participants were able to answer most of the questions correctly within an hour. Although no major problems were found, some flaws and bugs were revealed. Many of these issues have since been addressed, but they are listed below:

Faults Found

- Sometimes the graph visualization windows crash. REportal uses Grappa [20] to draw graphs. According to the author of Grappa, it can only handle small graphs with sizes of up to 2 MB. REportal sometimes has to handle graphs as large as 3 to 5 MB. Grappa has since been improved to display larger graphs, so this problem has been rectified.
- The clustering task freezes occasionally.

User Interface Problems

- There are many types of relationships between two entities, such as inheritance, containment and method invocation. However, when relationship searching results are listed in a table, no information about the types of relationships is displayed.
- Clustering takes a long time for large systems. When clustering, the graph window is blank. It is hard to tell if the clustering is in progress. A progress bar would be helpful.
- When a project is opened, all important features are listed as tabs on the top of the window for easy access. However, "reachability searching," an important feature, is not listed there. As a result, all participants had difficulties finding and using this feature.
- `Advanced Search` needs general refinement and usability improvements.

Some other interesting observations were:

- Although REportal provides context-sensitive on-line help and the hyperlink to it is placed on the right top of every window, people only view the documentation as a last resort of help. When they get stuck, they prefer to go through many iterations of trial-and-error than to refer to the help documentation. Even when they go to the help documentation, they glance at screen shots in an attempt to find the answer as soon as they can, instead of reading the documentation.
- When a project is opened, a list of tabs is displayed on the top of the window. Each tab is associated with a feature and the first tab is enabled by default. The choice of the default tab is important. Since the first tab, `Entity Search`, is enabled as default, people attempt to answer all questions using this feature. They switch to other features only when this one fails. This implies that the most frequently used feature should be set as default.
- Although the user interface of REportal is designed to be intuitive, we found that participants still have trouble using it the first time. However, once the users know what features REportal provides and how to use them, they can use REportal very comfortably. This explains why participants spent less time studying the second system than the first one.

6. Future Work and Conclusions

This section summarizes our work and outlines some plans for future work.

6.1. Future Work

This section outlines our plans for future work.

- **C/C++ program analysis**. Currently REportal only supports Java programs. In the future, GAST-MP (a source code analysis tool developed at SERG for GNU C) will be integrated into REportal so that C programs can be supported.
- **Portability.** Professional software developers prefer to analyze and store their code on their own sites because of security and privacy concerns. One way this can be facilitated is to develop a portable version of REportal that can be installed locally (probably behind a corporate fire wall).

- **Personalization.** Users will be able to personalize REportal in the future by configuring personal preferences such as the layout of tools, background colors, and the number of query entries per page, and so on.

- **Data mining.** We expect that REportal will have many active users in the near future. As a result, this will create a large repository of source code available for data mining analysis to support future software engineering research.

6.2. Summary & Research Contributions

Practitioners and researchers can take advantage of the latest developments in reverse engineering if the services provided by the tools are made available on the WWW. In this paper we present REportal, a web-based application that integrates many stand-alone reverse engineering tools. These tools reside on a server and can be accessed by authorized users through web browsers. REportal provides a user interface that enables users to analyze and browse source code, as well as to query and extract design information for Java programs.

To work with REportal, users only need a Java-enabled web browser. This allows them to analyze and visualize source code without going through the troubles of obtaining, installing, and learning how to use a set of reverse engineering tools.

At the time of this writing, REportal has 120 users from institutions of higher learning and industry. REportal has been in beta and we are now ready to release our first production version.

References

1. The Apache Software Foundation. http://httpd.apache.org/.
2. The AT&T Labs Research Internet Page. http://www.research.att.com.
3. B. Mitchell. A Heuristic Search Approach to Solving the Software Clustering Problem. Drexel University Ph.D. Thesis, 2002.
4. The Bunch Project. Drexel University Software Engineering Research Group (SERG). http://serg.mcs.drexel.edu/bunch.
5. The Drexel University Software Engineering Research Group (SERG). http://serg.mcs.drexel.edu.
6. E.R. Gansner, E. Koutsofios, S.C. North, and K.P. Vo. A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3):214–230, March 1993.
7. The `drawdag` at AT&T. http://www.research.att.com/dist/drawdag/mail.
8. Freshmeat. http://freshmeat.net/.

9. Graph Drawing Server at Brown University http://loki.cs.brown.edu:8081/graphserver/gds/gds-home.shtml
10. The Jakarta Project. http://jakarta.apache.org.
11. Jason Hunter and William Crawford. *JAVA Servlet Programming.* Chapter 1, pages 1-7. O'Reilly, 1999.
12. JAVA Servlet Technology. http://java.sun.com/products/servlet/index.html.
13. J. Clarke, J. J. Dolado, M. Harman, R. Hierons, B. Jones, M. Lumkin, B. S. Mitchell, S. Mancoridis, K. Rees, M. Roper, M. Shepperd. Reformulating Software Engineering as a Search Problem. In the *Journal of IEE Proceedings - Software* 150(3):161-175, 2003.
14. J. Korn, Y. Chen, and E. Koutsofios. Chava: Reverse engineering and tracking of java applets. In *Proc. Working Conference on Reverse Engineering*, October 1999.
15. K. Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability.* McGraw-Hill, 1993.
16. LintPlus Online. http://www.cleanscape.net/products/lintonline.
17. Microsoft .Net. Microsoft Corporation. http://www.microsoft.com/net.
18. M. Storey, K. Wong, F. Fracchia, and H. Müller. On integrating visualization techniques for effective software exploration. In *Proc. of IEEE Symposium on Information Visualization*, October 1997.
19. The Netcomputing AnyJ Java IDE. http://www.netcomputing.de/html/main/html.
20. N. S. Barghouti, J. Mocenigo, and W. Lee. Grappa: A Graph Package in Java. in *Fifth International Symposium on Graph Drawing*, pages 336-343. SpringerVerlag, Sept. 1997.
21. P.A.V. Hall. *Software Reuse and Reverse Engineering in Practice.* Vhapman & Hall, 1992.
22. P. Finnigan, R. C. Holt, I. Kalas, S. Kerr, et al. The Software Bookshelf. *IBM Systems Journal*, 36(4):564-593, 1997.
23. R. C. Holt. *Concurrent Euclid, The UNIX System and Tunis.* Addison Wesley, Reading, Massachusetts, 1983.
24. The REportal Project. Drexel University Software Engineering Research Group (SERG). http://reportal.mcs.drexel.edu/.
25. R.L. Rivest. The MD5 message-digest algorithm. *Request for Comments (RFC) 1321*, Internet Activities Board, Internet Privacy Task Force, April 1992.
26. R. Koschke. *Evaluation of Automatic Re-Modularization Techniques and their Integration in a Semi-Automatic Method.* PhD thesis, University of Stuttgart, Stuttgart, Germany, 2000.
27. R. Koschke. Software visualization in software maintenance, reverse engineering, and reengineering - a research survey. *http://www.informatik.uni-stuttgart.de/ifi/ps/rainer/softviz/*, 2000.
28. The Red Hat Source-Navigator. http://sources.redhat.com/sourcenav/.
29. Reengineering Wiki. http://www.program-ransformation.org/re/.
30. Secure Socket Layer. http://wp.netscape.com/security/techbriefs/ssl.html.
31. S. Mancoridis. ISF: A Visual Formalism for Specifying Interconnection Styles for Software Design. *International Journal of Software Engineering and Knowledge Engineering*, 8(4):517–540, 1998.
32. S. Mancoridis, B. Mitchell, Y. Chen, and E. Gansner. Bunch: A clustering tool for the recovery and maintenance of software system structures. In *Proceedings of International Conference of Software Maintenance*, pages 50–59, August 1999.

33. S. Mancoridis, B.S. Mitchell, C. Rorres, Y. Chen, and E.R. Gansner. Using automatic clustering to produce high-level system organizations of source code. In *Proc. 6th Intl. Workshop on Program Comprehension*, June 1998.
34. S. Mancoridis, T. Souder, Y. Chen, E. R. Gansner, and J. L. Korn. REportal: A web-based portal site for reverse engineering. In *Proc. Working Conference on Reverse Engineering*, October 2001.
35. S. North and E. Koutsofios. Applications of graph visualization. In *Proc. Graphics Interface*, 1994.
36. Sourceforge.net. http://sourceforge.net/.
37. S. Bridgeman, A. Garg and R. Tamassia A graph drawing and translation service on the WWW. In *Lecture Notes Comput. Sci. Springer-Verlag*, 1997.
38. T.A. Wiggerts. Using clustering algorithms in legacy systems remodularization. In *Proc. Working Conference on Reverse Engineering*, October 1997.
39. Tom Sawyer. Graph Drawing and Visualization Tool. http://www.tomsawyer.com.
40. Y. Chen. Reverse engineering. In B. Krishnamurthy, editor, *Practical Reusable UNIX Software*, chapter 6, pages 177–208. John Wiley & Sons, New York, 1995.
41. Y. Chen, E. Gansner, and E. Koutsofios. A C++ Data Model Supporting Reachability Analysis and Dead Code Detection. In *Proc. 6th European Software Engineering Conference and 5th ACM SIGSOFT Symposium on the Foundations of Software Engineering*, September 1997.