Ranking of Database Query Results Based on Concept Hierarchies

Kaligotla Ravi Kumar

Lecturer, Department of Computer Science Engineering
Bule Hora University, Ethiopia, Africa.

Abstract:

Search queries on biomedical databases, such as PubMed, often return a large number of results, only a small subset of which is relevant to the user. Ranking and categorization, which can also be combined, have been proposed to alleviate this information overload problem. Results categorization for biomedical databases is the focus of this work. A natural way to organize biomedical citations is according to their MeSH annotations. MeSH is a comprehensive concept hierarchy used by PubMed. In this paper, we present the BioNav system, a novel search interface that enables the user to navigate large number of query results by organizing them using the MeSH concept hierarchy. First, the query results are organized into a navigation tree. At each node expansion step, BioNav reveals only a small subset of the concept nodes, selected such that the expected user navigation cost is minimized. In contrast, previous works expand the hierarchy in a predefined static manner, without navigation cost modeling. We show that the problem of selecting the best concepts to reveal at each node expansion is NP-complete and propose an efficient heuristic as well as a feasible optimal algorithm for relatively small trees. We show experimentally that BioNav outperforms state-of-the-art categorization systems with respect to the user navigation cost. We have implemented BioNav for the MEDLINE database at http://db.cse.buffalo.edu/bionav.

1. Introduction

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

Many solutions have been proposed to address this problem—commonly referred to as information overload [1], [2], [3], [9], [16]. These approaches can be broadly classified into two classes: ranking and categorization—which can also be combined. Ranking presents the user with a list of results ordered by some metric of relevance [9] or by content similarity to a result or a set of results [16]. In categorization [1], [2], [3], query results are grouped based on hierarchies, keywords, tags, or attribute values. User studies have demonstrated the usefulness of categorization in finding relevant results of exploratory queries [12]. While ranked results are useful when the ranking function is aligned with user preferences or the result list is small in size, categorization is generally employed by users when ranking fails or the query is too "broad" [12]. BioNav belongs primarily to the categorization class, which is especially suitable for this domain given the rich concept hierarchies (e.g., MeSH [19]) available for biomedical data.

An intuitive way to categorize the results of a query on PubMed is by using the MeSH static concept hierarchy [19], thus, utilizing the initiative of the US National Library of Medicine (NLM) to build and maintain such a comprehensive structure. Each citation in MEDLINE is associated with several MeSH concepts in two ways: 1) by being explicitly annotated with them, and 2) by mentioning those in their text (see Section 7 for details). Since these associations are provided by PubMed, a relatively straightforward interface to navigate the query result would first attach the citations to the corresponding MeSH concept nodes and then let the user navigate the navigation tree. Fig. 1 displays a snapshot of such an interface where shown next to each node label is the count of distinct citations in the sub-tree rooted at that node.

A typical navigation starts by revealing the children of the root ranked by their citation count, and is continued by the user expanding on or more of them, revealing their ranked children and so on, until she clicks on a concept and inspects the citations attached to it. A similar interface and navigation method is used by e-commerce sites, such as Amazon

and eBay. For this example interaction, we assume that some of the citations the user is interested in are available on the three indicated concepts corresponding to three independent lines of research related to prothymosin, and therefore the user is interested in navigating to these concepts. These include, "Histones," which play a role in gene regulation and are essential for virus replication and tumor growth, "Cell Growth Processes" and "Transcription, Genetic," a key process for synthesis and replication of RNA and thus plays an important role in the duplication of cancer cells.

1.1 Overview of Computer Attacks

A computer attack is any malicious activity directed at a computer system or the services provided by the system. The attacks could come in the form of viruses, Denial of Service (DoS), exploitation of bugs, use of services by those unauthorized or even a physical attack against the computer hardware. An understanding of some techniques that have been used by crackers or even script kiddies will go a long way in helping provide a system that will protect institutions against the intrusions. Hackers have employed a varied number of techniques to break into systems and these include and are not limited to:

- **Social Engineering**: This is the type of attack where the attacker fools an authorized network user into divulging information that leads to access to network resources. The details may include: passwords, security policies, network or host addresses. The attacker could also impersonate genuine users, service providers or the support staff and sometimes uses this identity to deliver harmful software such as Trojan.
- **Abuse of Feature**: In abuse of feature, the authorized users perform action that are considered illegitimate. He may open illegitimate sites, send request or open up telnet sessions with into prohibited servers or hosts that could for example fill up the users assigned quarters or storage space.
- **Configuration errors** - The attacker may gain access to a network with configuration errors. These weaknesses could be allowing access without prompting for any form of authentication such as assigning users guest accounts which are left without password requirements. Users may erroneously be assigned administrative privileges. A user may also obtain privileges of other users by obtaining their passwords or bypassing control that restrict access(Root attack).
- **Masquerading** - Having monitored the flow of traffic, the attacker can modify a TCP packet or originate a packet but send it with a forged source address to give the impression that it is from a trusted source. The attacker can therein send Trojan, perform a denial of service or make request for sensitive data.
- **Worms** - These are destructive self-replicating programs that spread across the network. They can be sent as e-mail attachments to hamper network performance.
- **Viruses** - Are programs that replicate when a user performs some actions, such as running a program. The viruses may have the ability to destroy sensitive documents.
- **Implementation Bug** - Most trusted applications and programs have bugs. These bugs are exploited by hackers to gain access to the computer systems or networks. Examples of these include: buffer over flows, race conditions or even mishandled files. This is also referred to as client attack, where the attacker may exploit bugs on either the client, server, network software or protocol.

  Generally these attacks can be classified into four major classes, interception, interruptions, modification and fabrications.
- Interception: This means that some unauthorized party gains access to an asset. This could be a program, person or a computing system. An example of this could be wire tapping, or illicit copying of program or data files. The damage could be worsened when the attacker leaves no traces on the network.
- Interruption: In this situation an asset of the system is made unavailable or unusable. An example is a malicious destruction of a hardware device, erasure of a program or data file or malfunctioning of an operating system so that it does not get say a particular disk.
- Modification: The attacker both accesses and tampers with the asset. This may include altering the program so that it can perform an additional computation or modify data being transmitted. They may range from simple changes to more subtle changes that may not even be detected.
- Fabrication: This involves creating counterfeit objects on the computing system. When skillfully done may also go

undetected and thus a very serious threat to the network security.

2. Theoretical Analysis

In this paper, we present the BioNav system, a novel search interface that enables the user to navigate large number of query results by organizing them using the MeSH concept hierarchy. First, the query results are organized into a navigation tree. At each node expansion step, BioNav reveals only a small subset of the concept nodes, selected such that the expected user navigation cost is minimized. In contrast, previous works expand the hierarchy in a predefined static manner, without navigation cost modeling. We show that the problem of selecting the best concepts to reveal at each node expansion is NP-complete and propose an efficient heuristic as well as a feasible optimal algorithm for relatively small trees. We show experimentally that BioNav outperforms state-of-the-art categorization systems with respect to the user navigation cost. We have implemented BioNav for the MEDLINE database at http://db.cse.buffalo.edu/bionav.

2.1 Existing System

Existing search operation Information overload is a major problem when searching Biomedical databases such as PubMed, where typically a large number of citations are returned, of which only a small subset is relevant to the user.

2.2 Proposed System

The proposals dynamically categorize SQL query results by inferring a hierarchy based on the characteristics of the result tuples. Their domain is the tuple attributes and their problem is how to organize them hierarchically in order to minimize the navigation cost. They also decide the value ranges for each attribute, for both categorical and numerical ones, and how to rank them. One of the systems takes into consideration the user's preferences during the inference for a more personalized experience. Once the hierarchy is inferred, they follow a static navigation method. BioNav is distinct since it offers dynamic navigation on a predefined hierarchy, as is the MeSH concept hierarchy. Hence, BioNav is complementary to these systems, since it can be used to optimize the navigation, after these systems construct the navigation tree.

2.3 System Study

2.3.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

2.3.2 Economical feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.3.3 Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.3.4 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3. Experimental Investigation

3.1 System Design

3.1.1 Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

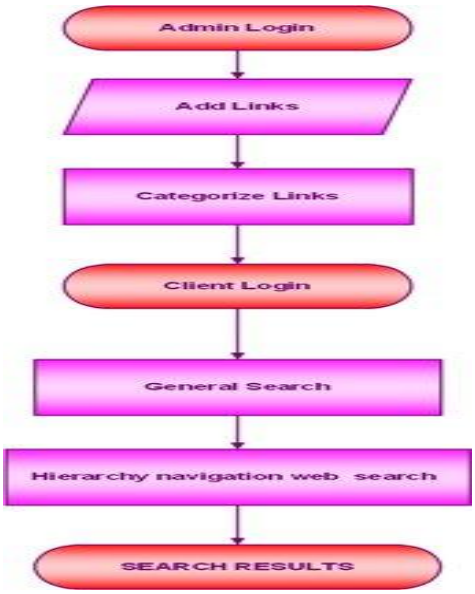
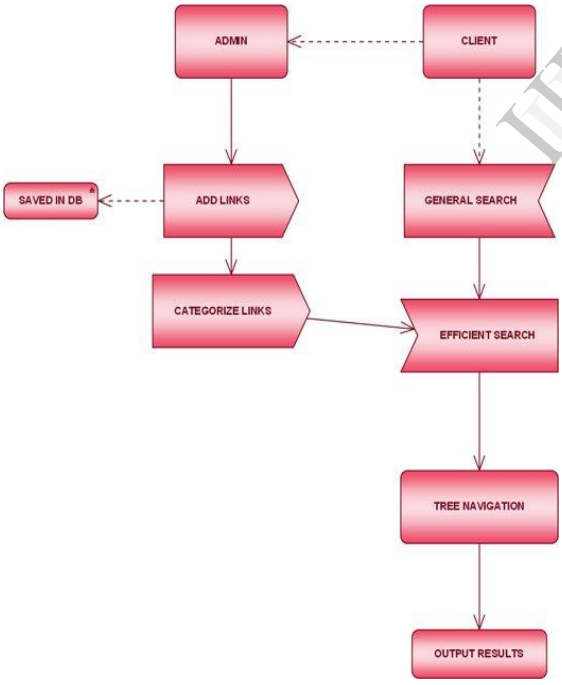


Fig. Data Flow Diagram

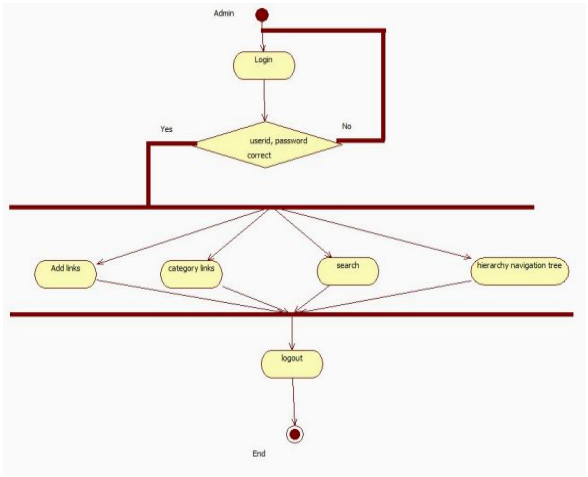
3.1.2 System flow Diagram



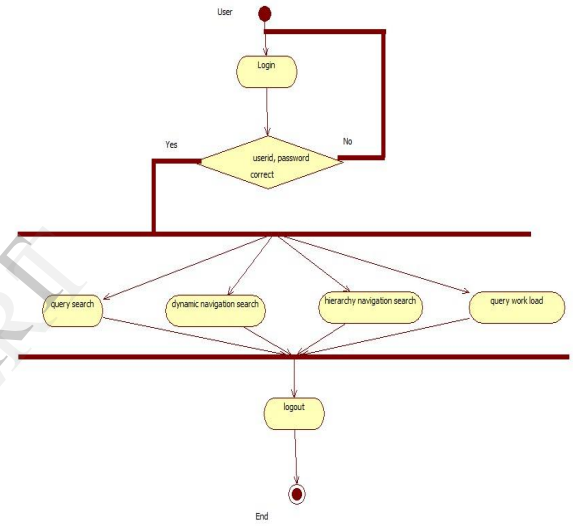
3.1.3 Activity Diagrams

3.1.3.1 Admin Activity Diagram

An activity diagram describes a system in terms of activities. Activities are states that represent the execution of a set of operations. Activity diagrams are similar to flowchart diagram and data flow.



3.1.3.2 User Activity Diagram



4. Implementation

4.1 Introduction

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

4.2 Modules

4.2.1 Query Search process module (or) Biomedical Search Systems module

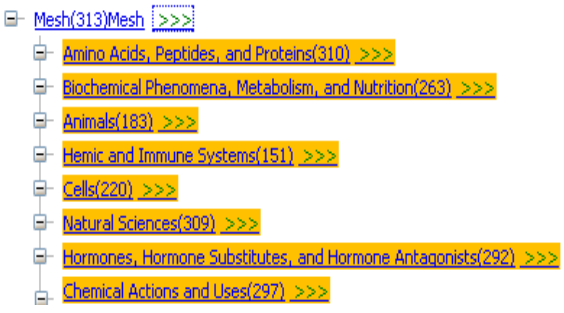
PubMed– using a keyword search interface. Currently, in an exploratory scenario where the user tries to find citations relevant to her line of research and hence not known a priori, she submits an initially

broad keyword- based query that typically returns a large number of results. Subsequently, the user iteratively refines the query, if she has an idea of how to, by adding more keywords, and re-submits it, until a relatively small number of results are returned. This refinement process is problematic because after a number of iterations the user is not aware if she has over-specified the query, in which case relevant citations might be excluded from the final query result.

Query on PubMed is using the MeSH static concept hierarchy , thus utilizing the initiative of the US National Library of Medicine (NLM) to build and maintain such a comprehensive structure. Each citation in MEDLINE is associated with several MeSH concepts in two ways: (i) by being explicitly annotated with them, and (ii) by mentioning those in their text . Since these associations are provided by PubMed, a relatively straightforward interface to navigate the query result would first attach the citations to the corresponding MeSH concept nodes and then let the user navigate the navigation tree

4.2.2 Dynamic navigation tree module

Navigation tree. Fig displays a snapshot of such an interface where shown next to each node label is the count of distinct citations in the subtree rooted at that node. A typical navigation starts by revealing the children of the root ranked by their citation count, and is continued by the user expanding on or more of them, revealing their ranked children and so on, until she clicks on a concept and inspects the citations attached to it. A similar interface and navigation method is used by e-commerce sites, such as Amazon and eBay. For this example, we assume that the user will navigate to the three indicated concepts corresponding to three independent lines of research related to prothymosin



BioNav introduces a dynamic navigation method that depends on the particular query result at hand and is demonstrated in Fig The query results are attached to the corresponding MeSH concept nodes as in Fig. but then the navigation proceeds differently. The key action on the interface is the expansion of a node that selectively reveals a ranked list of descendant (not necessarily children) concepts, instead of simply showing all its children.

4.2.3 Hierarchy navigation web(interface) search module

BioNav belongs primarily to the categorization class, which is ideal for this domain given the rich concept hierarchies (e.g., MeSH) available for biomedical data. We augment our categorization techniques with simple ranking techniques. BioNav organizes the query results into a dynamic hierarchy, the navigation tree. Each concept (node) of the hierarchy has a descriptive label. The user then navigates this tree structure, in a top-down fashion, exploring the concepts of interest while ignoring the rest.

4.2.4 Query Workload online operation module

On-Line Operation. Upon receiving a keyword query from the user, BioNav executes the same query against the MEDLINE database and

retrieves only the IDs (Pub Med Identifiers) of the citations in the query result. This is done using the ESearch utility of the Entrez Programming Utilities (eUtils) . eUtils are a collection of web interfaces to PubMed for issuing a query and downloading the results with various levels of detail and in a variety of formats. Next, the navigation tree is constructed by retrieving the MeSH concepts associated with each citation in the query result from the BioNav database. This is possible since MeSH concepts have tree identifiers encoding their location in the MeSH hierarchy, which are also retrieved from the BioNav database. This process is done once for each user query.

4.3 Sample Code

File Name : JSONWriter.Java

```
package org.json;
import java.io.IOException;
import java.io.Writer;
public class JSONWriter {
private static final int maxdepth = 20;
private boolean comma;
protected char mode;
private JSONObject stack[];
private int top;
protected Writer writer;
public JSONWriter(Writer w) {
this.comma = false;
 this.mode = 'i';
this.stack = new JSONObject[maxdepth];
this.top = 0;
this.writer = w;
}
 public JSONWriter array() throws JSONException {
if (this.mode == 'i' || this.mode == 'o' || this.mode ==
'a') {
this.push(null);
this.append("[");
this.comma = false;
 return this;
}
throw new JSONException("Misplaced array.")
}
private JSONWriter end(char m, char c) throws
JSONException {
  if (this.mode != m) {
 throw new JSONException(m == 'a' ? "Misplaced
endArray." :
"Misplaced endObject.");
}
this.pop(m);
try {
 this.writer.write(c);
} catch (IOException e) {
 throw new JSONException(e);
 }
this.comma = true;
 return this
}
public JSONWriter endArray() throws
JSONException {
 return this.end('a', ']');
}
 public JSONWriter endObject() throws
JSONException {
 return this.end('k', '}');
}
 public JSONWriter key(String s) throws
JSONException {
  if (s == null) {
 throw new JSONException("Null key.");
 }
if (this.mode == 'k') {
 try {
 stack[top - 1].putOnce(s, Boolean.TRUE);
  if (this.comma) {
 this.writer.write(',');
}
 this.writer.write(JSONObject.quote(s));
this.writer.write(':');
  this.comma = false;
  this.mode = 'o';
 return this;
 } catch (IOException e) {
  throw new JSONException(e);
                    }
 throw new JSONException("Misplaced key.");
               }
 public JSONWriter object() throws JSONException
{
 if (this.mode == 'i') {
this.mode = 'o';
 }
 if (this.mode == 'o' || this.mode == 'a') {
 this.append("{");
this.push(new JSONObject());
this.comma = false;
  return this;
 }
 throw new JSONException("Misplaced object.");
 }
 private void pop(char c) throws JSONException {
  if (this.top <= 0) {
   throw new JSONException("Nesting error.");
  }
char m = this.stack[this.top - 1] == null ? 'a' : 'k';
  if (m != c) {
 throw new JSONException("Nesting error.");
 }
this.top -= 1;
this.mode = this.top == 0 ? 'd' : this.stack[this.top - 1]
== null ? 'a' : 'k';
```

```
 }
  private void push(JSONObject jo) throws
JSONException {
 if (this.top >= maxdepth) {
 throw new JSONException("Nesting too deep.");
 }
this.stack[this.top] = jo;
this.mode = jo == null ? 'a' : 'k';
 this.top += 1;
  }
 public JSONWriter value(boolean b) throws
JSONException {
 return this.append(b ? "true" : "false");
 }
public JSONWriter value(double d) throws
JSONException {
 return this.value(new Double(d));
}
public JSONWriter value(long l) throws
JSONException {
return this.append(Long.toString(l));
 }
 public JSONWriter value(Object o) throws
JSONException {
return this.append(JSONObject.valueToString(o));
} }
```

4.4 Specific Requirements

4.4.1 Hardware Requirements:

Processor	- Dual Core
RAM	- 512 MB
Speed	- 1.65 GHZ
Hard Disk	- 160 GB

4.4.2 Software Requirements:

Operating System	: Windows 2000/XP
Application Server	: Tomcat 7.0.11
Front End	: J2EE-(HTML, Java, Jsp, Servlets)
Scripts	: JavaScript.
Development tool	: Net beans IDE 7.0
Build tool	: Ant
Server side Script	: Java Server Pages.
Database	: MsAccess
Database Connectivity	: JDBC

5. Experimental Results

5.1 Testing Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.2 Test Strategy and approach

5.2.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application, it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.2.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.2.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.2.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

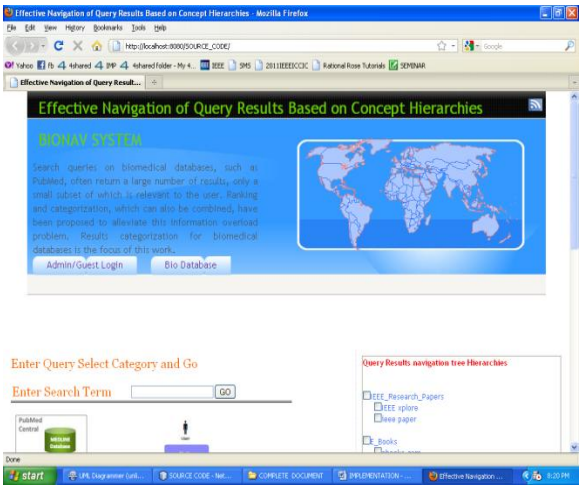
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

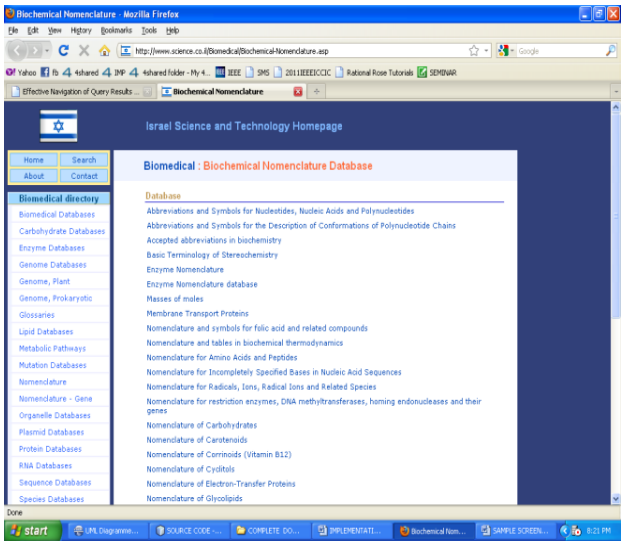
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

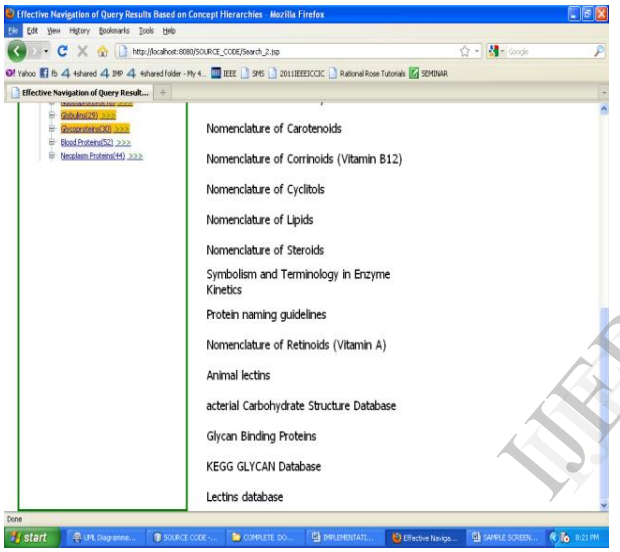
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

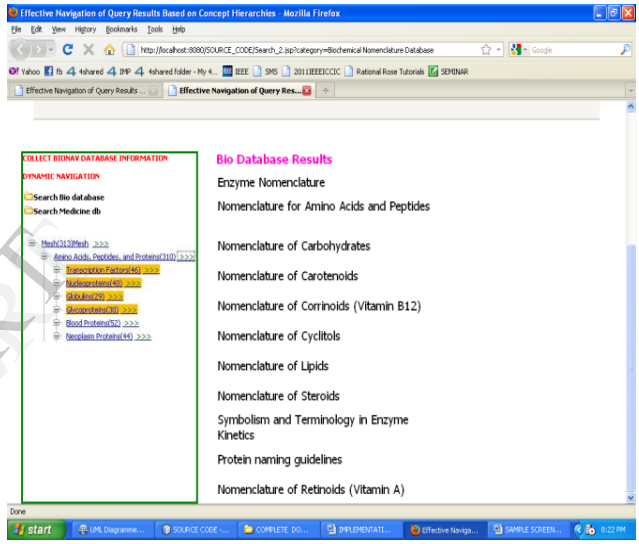
5.3 Screen Shots

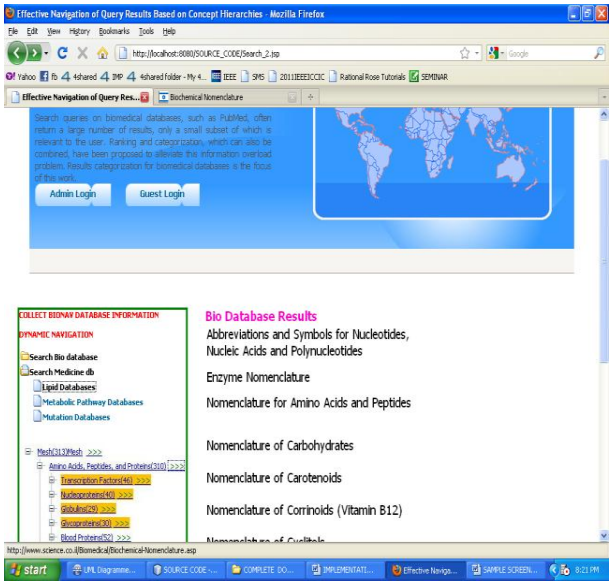


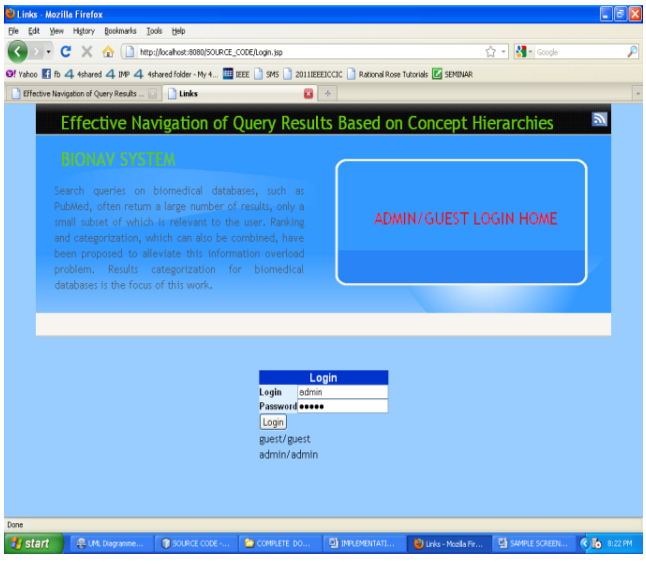

Effective Navigation of Query Results Based on Concept Hierarchies
BIONAV SYSTEM
Admin Login
Guest Login
Bio Database Results
Abbreviations and Symbols for Nucleotides, Nucleic Acids and Polynucleotides
Enzyme Nomenclature
Nomenclature for Amino Acids and Peptides
Nomenclature of Carbohydrates

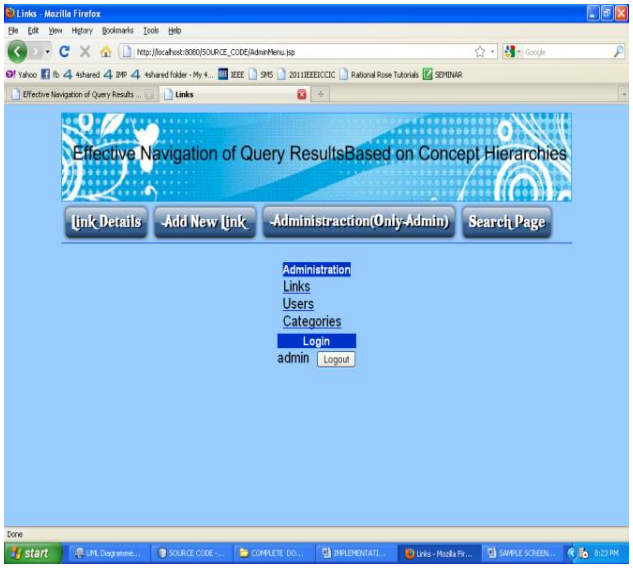

Israel Science and Technology Homepage
Biomedical : Biochemical Nomenclature Database
Biomedical directory
Database
Abbreviations and Symbols for Nucleotides, Nucleic Acids and Polynucleotides
Abbreviations and Symbols for the Description of Conformations of Polynucleotide Chains
Accepted abbreviations in biochemistry
Basic Terminology of Stereochemistry
Enzyme Nomenclature
Enzyme Nomenclature database
Masses of moles
Membrane Transport Proteins
Nomenclature and symbols for folic acid and related compounds
Nomenclature and tables in biochemical thermodynamics
Nomenclature for Amino Acids and Peptides
Nomenclature for Incompletely Specified Bases in Nucleic Acid Sequences
Nomenclature for Radicals, Ions, Radical Ions and Related Species
Nomenclature for restriction enzymes, DNA methyltransferases, homing endonucleases and their genes
Nomenclature of Carbohydrates
Nomenclature of Carotenoids
Nomenclature of Corrinoids (Vitamin B12)
Nomenclature of Cyclitols
Nomenclature of Electron-Transfer Proteins
Nomenclature of Glycolipids

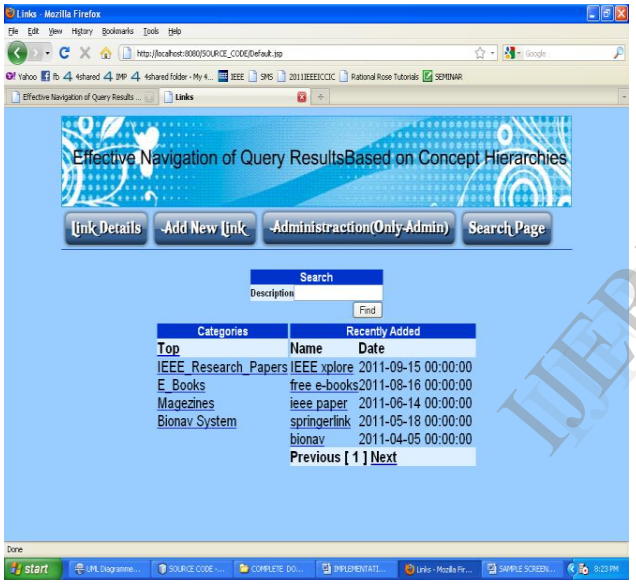

Nomenclature of Carotenoids
Nomenclature of Corrinoids (Vitamin B12)
Nomenclature of Cyclitols
Nomenclature of Lipids
Nomenclature of Steroids
Symbolism and Terminology in Enzyme Kinetics
Protein naming guidelines
Nomenclature of Retinoids (Vitamin A)
Animal lectins
acterial Carbohydrate Structure Database
Glycan Binding Proteins
KEGG GLYCAN Database
Lectins database

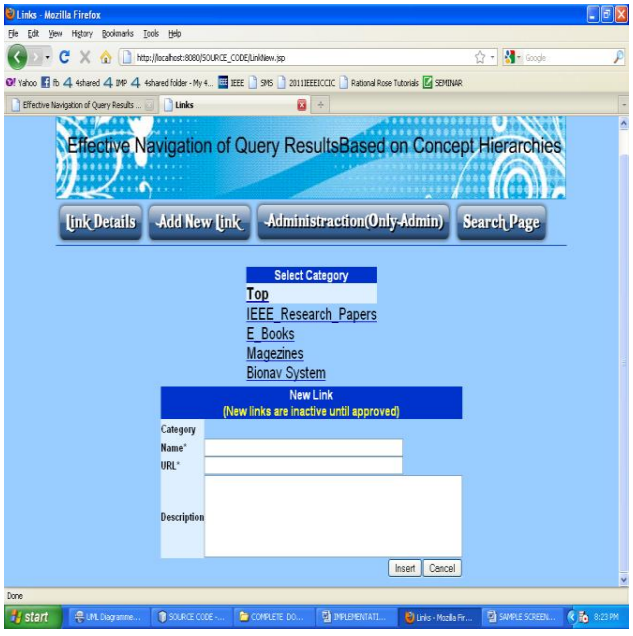

Bio Database Results
Enzyme Nomenclature
Nomenclature for Amino Acids and Peptides
Nomenclature of Carbohydrates
Nomenclature of Carotenoids
Nomenclature of Corrinoids (Vitamin B12)
Nomenclature of Cyclitols
Nomenclature of Lipids
Nomenclature of Steroids
Symbolism and Terminology in Enzyme Kinetics
Protein naming guidelines
Nomenclature of Retinoids (Vitamin A)


Admin Login
Guest Login
Bio Database Results
Abbreviations and Symbols for Nucleotides, Nucleic Acids and Polynucleotides
Enzyme Nomenclature
Nomenclature for Amino Acids and Peptides
Nomenclature of Carbohydrates
Nomenclature of Carotenoids
Nomenclature of Corrinoids (Vitamin B12)


Effective Navigation of Query Results Based on Concept Hierarchies
BIONAV SYSTEM
ADMIN/GUEST LOGIN HOME
Login
Login admin
Password
Login
guest/guest
admin/admin







6. Discussion of Results

MEDLINE is a medical literature analysis an retrieval system and bibliographic database of life science and biomedical information.

MEDLINE is free available on the internet and searchable via PubMed and USNLM (United States National Library of Medicine) Center for biotechnology information.

Search queries on biomedical databases, such as PubMed, often return a large number of results, only a small subset of which is relevant to the user. Ranking and categorization, which can also be combined, have been proposed to alleviate this information overload problem. Results categorization for biomedical databases is the focus of this work. A natural way to organize biomedical citations is according to their MeSH annotations. MeSH is a comprehensive concept hierarchy used by PubMed. In this paper, we present the BioNav system, a novel search interface that enables the user to navigate large number of query results by organizing them using the MeSH concept hierarchy. First, the query results are organized into a navigation tree. At each node expansion step, BioNav reveals only a small subset of the concept nodes, selected such that the expected user navigation cost is minimized. We have implemented BioNav for the MEDLINE database.

7. Conclusion

Information overload is a major problem when searching biomedical databases such as PubMed, where typically a large number of citations are returned, of which only a small subset is relevant to the user. In this paper, we presented the BioNav system to address this problem. Our solution is to organize the query results according to their associations to concepts of the MeSH concept hierarchy, and provide a dynamic navigation method that minimizes the information overload observed by the user. When the user expands a MeSH concept on our web interface, BioNav reveals only a selective list of descendant concepts, instead of simply showing all its children, ranked based on their estimated relevance to the user's query. We formally stated the underlying framework and the navigation and cost models used for the evaluation of our approach. Our complexity result proved that the problem of expanding the navigation tree in a way that minimizes the user's navigation cost is NP-complete. A feasible (for small trees) optimal algorithm and an efficient heuristic were developed. Experimental results validated the effectiveness of the proposed heuristic for diverse sets of queries and navigation trees, when compared to categorization systems using a static navigation method. The

architecture of the BioNav system was implemented and is available at http://db.cse.buffalo.edu/bionav.

References:

1. J S. Agrawal, S. Chaudhuri, G. Das and A. Gionis: Automated Ranking of Database Query Results. In Proceedings of First Biennial Conference on Innovative Data Systems Research (CIDR), 2003.
2. K. Chakrabarti, S. Chaudhuri and S.W. Hwang: Automatic Categorization of Query Results. SIGMOD Conference 2004: 755-766.
3. Z. Chen and T. Li: Addressing Diverse User Preferences in SQLQuery- Result Navigation. SIGMOD Conference 2007: 641-652
4. L. Comtet: Advanced Combinatorics: The Art of Finite and Infinite Expansions, rev. enl. ed. Dordrecht, Netherlands: Reidel, pp. 176- 177, 1974.
5. R. Delfs, A. Doms, A. Kozlenkov and M. Schroeder: GoPubMed: Ontology-Based Literature Search Applied to Gene Ontology and PubMed. German Conference on Bioinformatics 2004: 169-178.
6. D. Demner-Fushman and Jimmy Lin: Answer Extraction, Semantic Clustering, and Extractive Summarization for Clinical Question Answering. International Conference on Computational Linguistics and the Annual Meeting of the Association For Computational Linguistics, 2006: 841-848
7. (2008) Entrez Programming Utilities. . Available:http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
8. U. Feige, D. Peleg and G. Kortsarz: The Dense k-Subgraph Problem. Algorithmica 29 (2001) 410-421
9. V. Hristidis and Y. Papakonstantinou: DISCOVER: Keyword Search in Relational Databases. In Proc. of VLDB Conference, 2002.
10. R. Hoffman and A. Valencia: A gene network for navigating the literature. Nature Genetics, 36(7):664, 2004.
11. (2008) Humboldt-Universität zu Berlin – Ali Baba: PubMed as a graph. [Online]. Available: http://alibaba.informatik.huberlin.
12. (2008) iHOP - Information Hyperlinked over Proteins. Available:http://www.ihop-net.org/UniPub/iHOP/
13. A. Kashyap, V. Hristidis, M. Petropoulos, and S. Tavoulari: BioNav: Effective Navigation on Query Results of Biomedical Databases. (Short Paper), ICDE 2009, to appear. Available at http://www.cs.fiu.edu/~vagelis/publications/BioNavICDE09.pdf
14. S. Kundu and J. Misra: A Linear Tree Partitioning Algorithm. SIAM J. Comput. 6(1): 151-154 (1977)
15. W. Lee, L. Raschid, H. Sayyadi and P. Srinivasan: Exploiting Ontology Structure and Patterns of Annotation to Mine Significant Associations between Pairs of Controlled Vocabulary Terms. DILS 2008: 44-60.
16. J. Lin and W.J. Wilbur, "Pubmed Related Articles: A Probabilistic Topic Based Model for Content Similarity," BMC Bioinformatics, vol. 8, article no. 423, 2007.
17 D. Lindberg, B. Humphreys, and A. McCray, "The Unified Medical Language System," Methods of Information in Medicine, vol. 32, no. 4, pp. 281-291, 1993.
18. D. Maglott, J. Ostell, K.D. Pruitt, and T. Tatusova, "Entrez Gene: Gene-Centered Information at NCBI," Nucleic Acids Research, vol. 33, pp. D54-D58, Jan. 2005.
19. Medical Subject Headings (MeSH), http: //www.nlm.nih.gov/ mesh/, 2010.
20. J.A. Mitchell, A.R. Aronson, and J.G. Mork, "Gene Indexing: Characterization and Analysis of NLM's GeneRIFs," Proc. AMIA Ann. Symp., pp. 460-464, Nov.