IRiSS – A Source Code Exploration Tool

Denys Poshivyanyk
Andrian Marcus
Yubo Dong
Andrey Sergeyev

Computer Science Department
Wayne State University
Detroit, MI, USA
Concept Location

- Locating the implementation of a concept or feature in the source code
- Static
  - Dependency based search [Chen’00]
  - String based search (i.e., grep)
  - IR methods [Marcus’04]
- Dynamic
  - Execution traces - Reconnaissance [Wilde’92]
- Combined (using concept analysis) [Eisenbarth’03]
- Used in incremental change, comprehension, debugging, etc.
Concept Location

- Locating the implementation of a concept or feature in the source code
- Static
  - Dependency based search [Chen’00]
  - String based search (i.e., grep)
  - IR methods [Marcus’04]
- Dynamic
  - Execution traces - Reconnaissance [Wilde’92]
- Combined (using concept analysis) [Eisenbarth’03]
- Used in incremental change, comprehension, debugging, etc.
“Find in Files” in .NET?
Results for “Find in Files”
Our Inspiration
How do We Want to Search?

- “Google-like” queries
- Ranked results
- Various granularity of results (i.e., classes, methods, etc.)
IRiSS

Information Retrieval based Software Searching
Information Retrieval

• An Information Retrieval System is capable of storage, retrieval, and maintenance of information (e.g., text, images, audio, video, and other multi-media objects) [Kowalski’97]

• IR methods (e.g., signature files, inversion, clustering, probabilistic classifiers, vector space models, etc.) are used often for text retrieval (e.g., libraries, search engine, etc.)
Searching with IRiSS

1. User sets query settings
2. Preprocessing
3. Corpus
4. Query
5. Finding similar elements
6. Mapping
7. Results

Source Code → Corpus
Corpus → Semantic Search Space
Semantic Search Space → Mapping Engine
Mapping Engine → User
User → Mapping Engine
Source Code ← Mapping Engine
Corpus Generation

- Parsing to extract semantic information (i.e., comments and identifiers)
- split_identifiers & SplitIdentifiers
- Define source code documents with user-defined granularity (e.g., class, methods, functions, declarations, interfaces, etc.)
- Works on C/C++
- It is easy to extend to other languages
Building the Semantic Search Space

- We use Latent Semantic Indexing (LSI)

- Each source code element is transformed into a vector, based on the words it contains

- A similarity measure between two documents is defined as the cosine between their corresponding vectors
Query Formulation

- User defined queries
  - Most common, based on user experience and domain knowledge, little known about querying patterns

- Only query terms presented in the corpus are considered
Conclusions

• Pros
  – Simple and flexible to use
  – Returns ranked results – advantage over grep

• Cons
  – Misses some data elements (granularity)
  – Depends on the quality of comments and identifiers (grep has the same problem)

• Using IR methods for concept location and source code browsing is promising
**Current Work**

- Address the query refinement issue
  - Identify misspelled words
  - Suggest additional words/elements based on first set of results – combine them with **and, or, xor, not**
  - Define the semantic signature of a source code element
  - Deal with scalability issues
- Combine LSI with dependency graph based search, dynamic methods, concept analysis
- Implement the tool as plug-in to Eclipse as well
Google Eclipse Search
Google Desktop Search + Eclipse
Future Work – Result Clustering

1. Carnegie Mellon - Software Engineering Institute Read about collaboration opportunities, products and services, publications, and management practices.
   www.sei.cmu.edu - M8M1, Vivisim1, AskJeeves 1, Open Directory 10, Looksmart 18

2. TCSE: Technical Council on Software Engineering IEEE CS WARNING: Your browser does not support Frames. Please upgrade your browser and try http://tcse.org ...
   www.tcse.org - Vivisim 3, AskJeeves 11, M8M12

3. Software-Engineer.org Engineers can discuss training and certification, absorb concepts such as Hierarchical Input Process Output Diagrams and real-time systems, and practice OOP principles.
   www.software-engineer.org - Looksmart 5, M8M 8, Lycos 8, AskJeeves 13

4. Center for Software Engineering Research at the University of Southern California includes software architecture and collaborative software engineering.
   sunsa.usc.edu - Looksmart 8, AskJeeves 14, Open Directory 18, M8M24

5. Software Engineering Laboratory - NASA Partnership developed between NASA, the University of Maryland and Computer Sciences Corporation. Find data, workshops and related sites.

5. RSP&A - Software Engineering Resources Pressman and Associates offers hundreds of resources related to products and development.
   www.rspa.com/spi - M8M 2, AskJeeves 3, Vivisim 23, Looksmart 24
DEMONSTRATION