

3D Visualization for Concept Location in Source Code

Xinrong Xie
Denys Poshyvanyk
Andrian Marcus

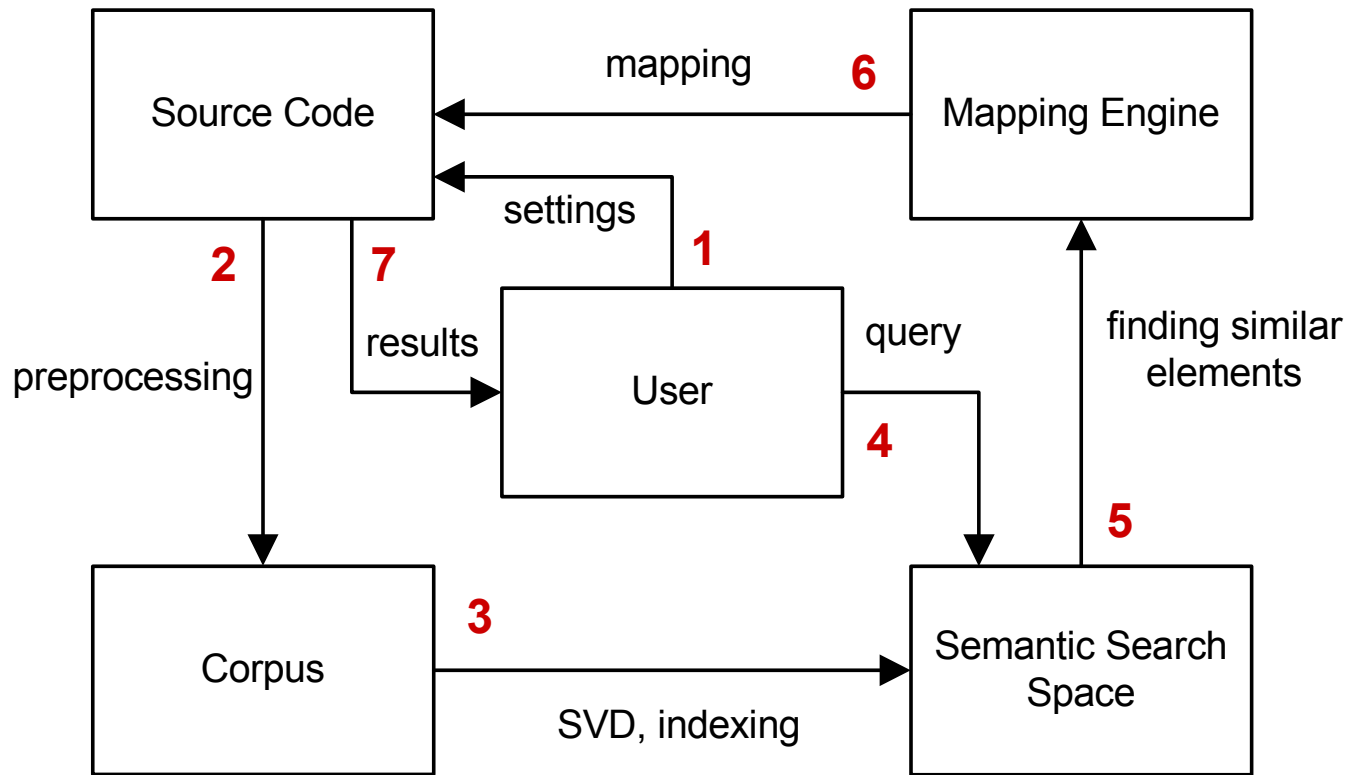
Computer Science Department
Wayne State University
Detroit, MI, USA



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.

Information Retrieval based Software Searching (IRiSS)



Corpus Generation

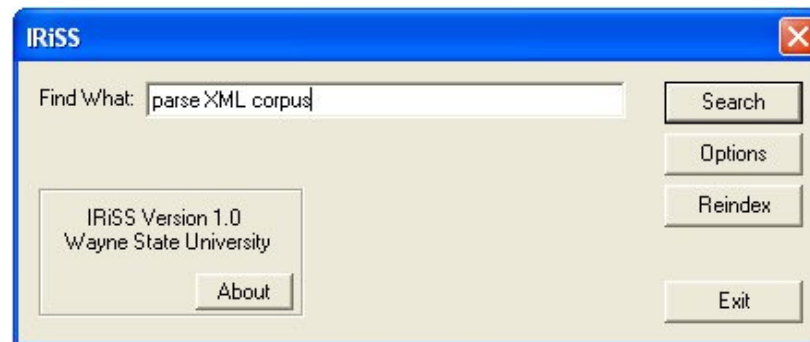
- Parsing to extract semantic information (i.e., comments and identifiers)
- `split_identifiers` & `SplitIdentifiers`
- Define source code documents with user-defined granularity (e.g., class, methods, functions, declarations, interfaces, etc.)
- Works on C/C++
- It is easy to extend to other languages

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

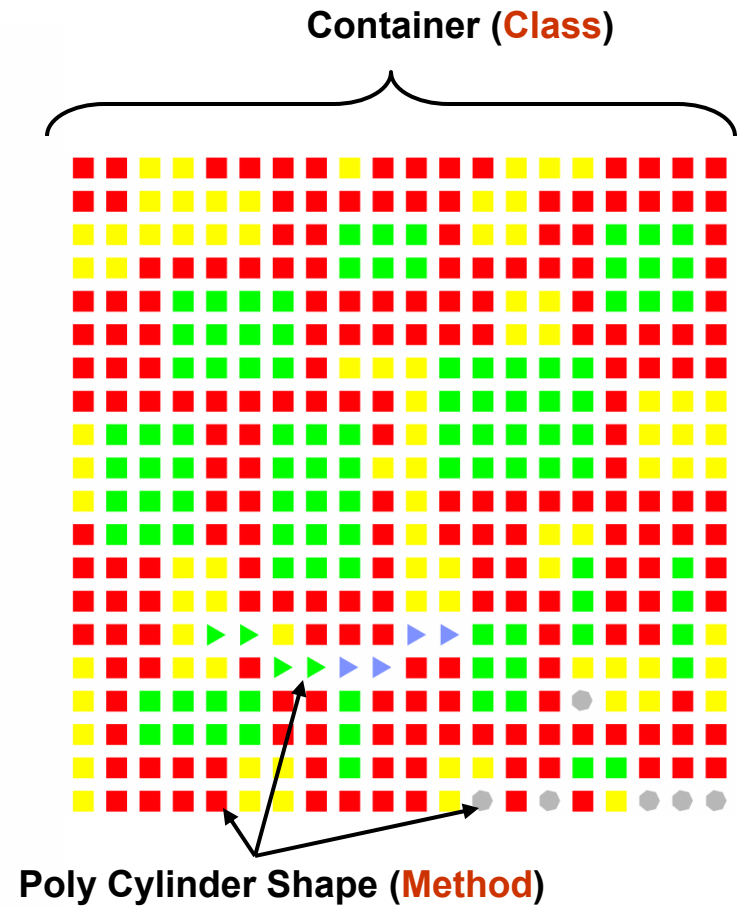
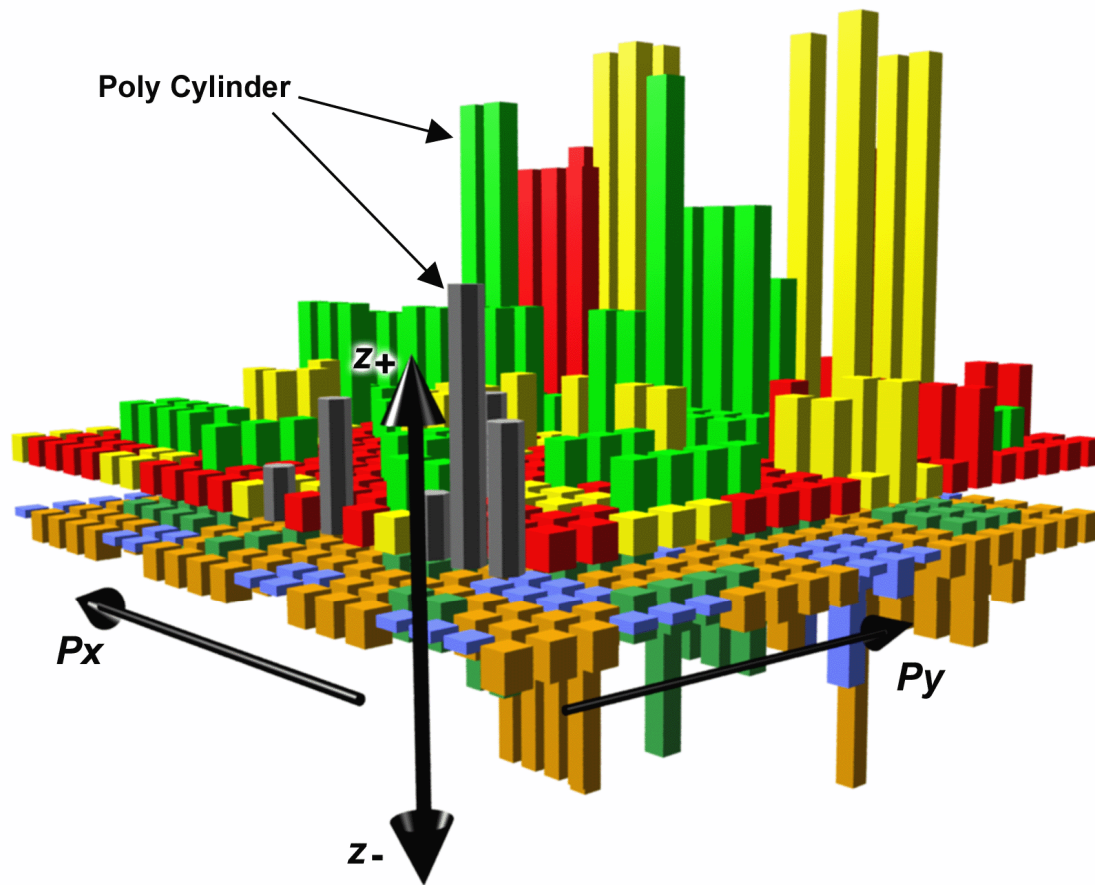
Query Formulation

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



- Only query terms presented in the corpus are considered

sv3D Visual Metaphor



Concept Location with IRiSS and sv3D

- Metaphors
 - Similarities between user queries and source code elements are mapped to **color**
 - User browsing history is mapped to **height**
- Benefits
 - Overview of the search results in the context of the whole system
 - Browsing history for multiple queries

Example

