Integrating COTS Search Engines into Eclipse: Google Desktop Case Study

Denys Poshivanyk, Maksym Petrenko, Andrian Marcus

SEVERE Group @

WAYNE STATE UNIVERSITY
Issues in concept location

- Simple and efficient methods -> poor precision and recall
- Complex methods -> higher precision and recall at the cost of efficiency
IR based concept location

- Uses Latent Semantic Indexing to index and search the source code
- Problems with the prototype
  - slow
  - inefficient re-indexing
  - no resources to support and adoptable version
Solutions for adoption

• Industry grants
  - not always fits our research agenda

• Component based approach with industrial strength COTS
Advantages

• Support of natural language queries
• Scalability, efficiency, and reliability
• Sorting results by relevance (date)
• Search in intra- and extra-net
Integrating GES into Eclipse

- GDS Java API
- org.eclipse.search extension points:
  - searchPages
  - searchResultViewPages
Google Eclipse Search (GES) Plug-in
Google Eclipse Search (GES) Plug-in
Advantages of COTS GDS

• GES improves as GDS does
• E.g., GDS v4 removes deleted files from the search results
• Modifiers to restrict scope of the search
  - previously had to tweak undocumented Win registry keys
Full Price of GDS

• Indexes everything (pdf, mail, app data)
• Re-indexing is done while PC is idle
• GDS v3 feature ‘share across computers’
  - indexed files are copied to the server
  - security and privacy issues
• “Lock Search” feature
Availability

- Open-source

- Researchers extended and compared GES with other tools:
  - Shepherd et al [AOSD’07]
Future Work

• Searching based on developer profiles
  - Visited/edited/deleted files
• Hybrid search engines (COTS-based)
  - LSI + GDS
• Googling public source code from IDE