FACILITATING INFORMATION MANAGEMENT IN INTEGRATED DEVELOPMENT ENVIRONMENTS THROUGH VISUAL INTERFACE ENHANCEMENTS

Haipeng Cai

Department of Computer Science and Engineering

University of Notre Dame

Supported by ONR Award N000141410037
Developers deal with deluge of information

Motivation

Program Documentation

Documentation of a software system involves collecting, organizing, storing and maintaining a complete historical record of programs and other documents used or prepared during different phases of the system development.

Search public source code.

Pictures courtesy of Google Image Search
And switch among different information sources

Motivation

Program Documentation

Documentation of a software system involves collecting, organizing, storing, and maintaining a complete historical record of programs and other documents used or prepared during different phases of the system development.
Frequent switches reduce productivity

- Mental-model interruption
- Individual-task latency
- ......
Modern IDEs help yet still suffer

Visual aids

Text
Modern IDEs help yet still suffer
Modern IDEs help yet still suffer

API/Code examples
Facilitating information management in IDEs

- Interface enhancements
  - Context-driven API/code example views
  - Coworker views
  - In-situ interface

- Software visualization
  - Multiple code visualizations
  - Interactive linked visualization
Context-driven API/Code example views

- Automatic context-driven information foraging
  - API usage
  - Code examples

Proposed Approach
Co-worker views

- On-demand co-worker teaming up
  - Real-time coaching / demonstration
  - Online discussion

Proposed Approach
In-situ interface over code editing

- Integrate visual aids with source code editing
  - Automatic push/hiding of commonly used shortcuts
  - Object-sensitive recommendation

![Main code view](image)
Multiple visualizations of source code

- Different representations of code in separate views
  - Same data
  - Alternative visual depiction (textual and graphical)
Interactions over linked visualizations

- Linked operations across multiple views
  - Trigger an operation where it is most efficient to do
  - Map the operation to other representations

```java
class A {
    public int getValue() {
        Integer nCounter = B.MAX_N;
        nCounter += 2;
    }
}
```
Beyond the visual enhancements

- Incorporating program analysis in IDEs
  - Information extracted from programs is more often needed than external sources — outputs of program analysis
Summing up

- **Proposal**
  - **Motivation**
    - Reduce context switching in dealing with multiple information needs with modern IDEs
  - **Solution**
    - Interface enhancements
    - Interactive code visualization
  - **Approach**
    - Reduce switching within an IDE
      - Multiple-view interactive linked visualization
      - In-situ interface
    - Reduce switching over an IDE
      - Co-worker views
      - API/code example views

Conclusion
Future work

Implementation
- Via IDE plug-ins to start with

Evaluation
- User studies
  - Groups using the enhanced IDE versus a traditional IDE
  - Coding and comprehension tasks
  - Differences in developer performance
  - Quality and time of task completion
Acknowledgements

“Facilitating Information Management in Integrated Development Environments through Visual Interface Enhancements”

Haipeng Cai
http://cse.nd.edu/~hcai/
hcai@nd.edu
Take-away

Three *interface* features and two *visualization* enhancements are proposed to facilitate information management in modern IDEs.