**Tool Support for Managing Crosscutting Concerns in Existing Artifacts**

Elisa Baniassad, Martin Robillard, Albert Lai, Gail Murphy

Software Developers spend a lot of time handling crosscutting concerns. Tool support can help a developer find, understand and manage these aspects. We are working to address these needs for both source-level and design-level artifacts.

Documentation contains descriptions of many concepts, including cross-cutting concerns. Readers of documentation often want to read only portions of text relevant to those aspects.

**Design Rationale Graph**

The text of the documentation is analyzed and each sentence graphically depicted. Design concepts and entities become major nodes in the graph. The rest of the text is represented by minor nodes.

**Design Documentation**

Extracting Crosscutting Concerns

Finding Crosscutting Concerns by Navigating Structural Dependencies

Source Code

Modeling Crosscutting Concerns for Re-engineering through Conceptual Modules

Software Developers spend a lot of time handling crosscutting concerns. Tool support can help a developer find, understand and manage these aspects. We are working to address these needs for both source-level and design-level artifacts.

**Query to Extract Aspects**

Once the DRG is created, users can use regular expressions to extract portions of interest. This will help isolate relevant portions of the text that pertain to crosscutting concerns.

**Elisa Baniassad**

1. Code implementing concerns is often related through structural dependencies.
2. Discovering and capturing these dependencies directly in source code is difficult and unwieldy.
3. To find and describe concerns in source code, we are using a representation of a program that abstracts the implementation details of a concern by storing the key structures implementing the concern. This way, relationships between different parts of a program implementing a concern are made more explicit.

**Elisa Baniassad**

We hypothesize that a developer can perform a modification task more efficiently when the developer has access to an abstract, grounded, partial behavioral model of a concern (or concerns) relevant to the modification.

**Martin Robillard**

Modification tasks are cognitively taxing because developers need to remember details about the source code, existing and desired behavior, and mappings between them.

**Albert Lai**

To help developers reason about, analyze, and verify the modification, we have developed a tool that allows developers to perform queries about the relationships between model elements and source code.