Writing
Analyzing
Testing
Maintaining

Configurable Software
What is Configurable Software?

- Software that can be configured to “run differently.”
- Across different environments:
  - Operating systems (Windows vs. OS X)
  - Hardware (X86 vs. SPARC)
  - User interfaces (KDE vs. Gnome)
- Enabling additional functionality:
  - Via command-line flags or config files
Examples

... 
#else
  \(P_2\)
#endif
...
Examples

```c

...  
win = open_window();
...

Win *open_window() {
    Win *w = malloc(...);
    #ifdef KDE
        w.window = kOpenWindow();
    #elif GNOME
        w.window = g_open_window();
    #endif
    return w;
}

struct Win {
    ...
    #ifdef KDE
        KWindow *window;
    #elif GNOME
        GWin *window;
    #endif
    ...
}
```
Examples

```java
AST a = parse(lex(input));
...
if (opt_foo)
    a = foo(a);
if (opt_bar)
    a = bar(a);
...
```
Configuration Mappings
Configuration Mappings
Configuration Mappings

- They explain how a configuration relates to another.
- Fuzzy example:

<table>
<thead>
<tr>
<th>Gnome</th>
<th>KDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_open_window()</td>
<td>kOpenWindow()</td>
</tr>
<tr>
<td>g_close_window()</td>
<td>kCloseWindow()</td>
</tr>
<tr>
<td>GWin</td>
<td>KWindow</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Applications: Testing

- Write Gnome-specific test.
- Automatically generate corresponding KDE test.
Applications: Maintenance

- Modify Gnome code.
- Tool helps you identify where and how to make corresponding changes to the KDE code.
Applications: Maintenance

- Change both configurations.
- A tool helps you understand if the two configurations are in sync.
Applications: Bug Reports

- A bug database contains a Gnome-specific bug.
- A tool automatically inserts a “bug warning” into the database that corresponding code in the KDE configuration may contain an equivalent bug.
Project Ideas

- Automatically extract configuration-specific code from programs.
  - “Slice out” the configurations from a program.
- “Fairly easy” to do when configs are managed with the C preprocessor.
Project Ideas

- Study bug databases for configurable programs.
  - Bug was reported and fixed in one config.
  - Equivalent bug in the other configuration was reported and fixed at a later time.
- A bug was reported for one config.
- Determine if the equivalent bug exists in the corresponding configuration.
Project Ideas

- Empirical study of test harness management for configurable software.
- Does the harness structure match the program’s configuration structure?
- Is there unnecessary duplication?
- Are some configurations not tested at all?

- Suggestions for improvement.