Why Plugins?

- Applications not perfect
- Specialized features
- Allows users to customize
- Iterative Development
Plugin Architecture
Plugin Architecture
Realization of plugins
Realization of plugins

- Possible designs:
  - Protocols
  - (Abstract) base class
Realization of plugins

- Possible designs:
  - Protocols
  - (Abstract) base class
  - Code shipped in bundle
Realization of plugins

- Possible designs:
  - Protocols
  - (Abstract) base class
  - Code shipped in bundle
  - Bundles provide principal class as entry point
Realization of plugins

- Possible designs:
  - Protocols
  - (Abstract) base class
  - Code shipped in bundle
  - Bundles provide principal class as entry point
  - For Mail.app: Provide compatibility information (SupportedPluginCompatibilityUUIDs)
Realization of plugins

- Possible designs:
  - Protocols
  - (Abstract) base class
- Code shipped in bundle
- Bundles provide principal class as entry point
- For Mail.app: Provide compatibility information (SupportedPluginCompatibilityUUIDs)
- If plugins not supported by app: SIMBL
Reverse Engineering

- strings
- class-dump
- F-Script, F-Script Anywhere
- gdb
Unix Tools

$strings Mail.app/Contents/MacOS/Mail
    % P+
    [...] This method must be called off the main thread
    /SourceCache/Mail/Mail-1076/Assistant.subproj/AccountAutoconfigurator.m
    [...] Searching local account settings...
    LOCAL_BUNDLE_SEARCH
    DontAutoconfigureFromOnlineISPDatabase
    [...] setAutoconfigurationActivity:
    autoconfigurationActivity
    [...] Account
    [...]
class-dump

- Examines Objective-C runtime information
- Similar information to otool -ov
- But: formatted as Objective-C headers
- Usage: class-dump -H -o <output dir> <Mach-O file>
F-Script [Anywhere]

- Interactive introspection and manipulation of Cocoa objects
- Can be used stand-alone as scripting language
- Can be loaded into any Cocoa application at runtime
- Point and click interface to introspect any widget
## F-Script Syntax

<table>
<thead>
<tr>
<th>[object method]</th>
<th>object method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NSString</strong> <em>string = @&quot;hello&quot;</em>*</td>
<td><strong>string := 'hello'</strong></td>
</tr>
<tr>
<td>[[string substringFromIndex:2] length]</td>
<td>(string substringFromIndex:2) length</td>
</tr>
<tr>
<td>array = [NSMutableArray arrayWithObjects:@&quot;hello&quot;, @&quot;world&quot;, nil]</td>
<td>array := {'hello', 'world'}</td>
</tr>
<tr>
<td>array length =&gt; {5,5}</td>
<td>array length =&gt; {5,5}</td>
</tr>
</tbody>
</table>
Objective-C Runtime System
Objective-C Runtime System

- Takes care of dynamic behavior of Objective-C
Objective-C Runtime System

- Takes care of dynamic behavior of Objective-C
- 3 methods to interact with the runtime:
Objective-C Runtime System

- Takes care of dynamic behavior of Objective-C
- 3 methods to interact with the runtime:
  - Objective-C source code
  - NSObject methods
  - Runtime Library
Objective-C Runtime System

- Takes care of dynamic behavior of Objective-C
- 3 methods to interact with the runtime:
  - Objective-C source code
  - NSObject methods
  - Runtime Library
Objective-C Runtime System

- Use it for
  - Method Swizzeling
  - Subclassing
  - `poseAsClass`: gone in Snow Leopard
Method Swizzeling

- Exchange implementation of 2 methods
- Allows redefining a method and reusing the old one
- Works without subclassing
- Wherever method is used, the behavior is changed

```c
+(void)PIMswizzleMethod:(SEL)originalSelector withMethod:(SEL)newSelector fromClass:(Class)fromClass;
{
    Method originalMethod = class_getInstanceMethod([[self class], originalSelector);
    Method newMethod = class_getInstanceMethod(fromClass, newSelector);
    class_addMethod([[self class], newSelector, method_getImplementation(originalMethod),
                      method_getTypeEncoding(originalMethod)));
    method_setImplementation(originalMethod, method_getImplementation(newMethod));
}
```
Demo
Resources

- **F-Script**
  - http://www.fscript.org/

- **SIMBL**
  - http://www.culater.net/software/SIMBL/SIMBL.php

- **Tutorial by Mike Solomon**
  - http://www.culater.net/wiki/moin.cgi/CocoaReverseEngineering#head-ed78de1fd3f2ae13b6ed4435cad6602ddaee5a4f9

- **Runtime Programming Guide**

- **Mail.app Plugin Developers Google Group**
  - http://groups.google.com/group/apple-mail-dev
Thanks for your attention!

Questions?

Contact me:
jp@cs.rwth-aachen.de
http://hci.rwth-aachen.de/jpkraemer