A Plague of Plug-ins

Thomas Ball
Microsoft Research
May 2011
Plug-ins

- The why and wherefore of plug-ins
- Problems of composition
  - Isolation
  - Connectedness
  - Dependences
  - Protocols
  - Feature Interaction
- Other thoughts
  - Smart phone
  - Cloud computing
  - Human-based computation
Car Plug-ins

Shop for after market car stereo

- 2006 GMC Canyon Car Stereo: $315.00
- 2006 Toyota Camry Car Stereo: $265.00
- 2008 Acura TL Car Stereo: $495.00
- 2004 Nissan Quest Car Stereo: $395.00

Guide to Aftermarket Car Stereo Wiring | eHow.com
A common form of customization for car owners is aftermarket additions to their car stereo system. There are many options for creating a custom system from speakers...
www.ehow.com/way_5700933_guide-aftermarket-car-stereo-wiring.html
Plug-ins are aftermarket extensions of a product that

– Give consumers more choice

– From a market of plug-in producers
My Start With A Very Pluggable Computer, 1979

Apple II serial interface card that required cutting and soldering to reconfigure. The user would cut the wire trace between the >> cones at X1 and X3 and solder the <-> cones together at X2 and X4.
The first personal computer to make use of the 80386 was designed and manufactured by Compaq, marking the first time a fundamental component in the IBM PC de facto standard was updated by a company other than IBM.
The Pluggable PC Platform

### Conventional PCI

**PCI Local Bus**

- **Three 5 V 32-bit PCI expansion slots on a motherboard (PC bracket to left)**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Year created</strong></td>
<td>July 1993</td>
</tr>
<tr>
<td><strong>Created by</strong></td>
<td>Intel</td>
</tr>
<tr>
<td><strong>Supersedes</strong></td>
<td>ISA, EISA, MCA, VLB</td>
</tr>
<tr>
<td><strong>Superseded by</strong></td>
<td>PCI Express (2004)</td>
</tr>
<tr>
<td><strong>Width in bits</strong></td>
<td>32 or 64</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>133 MB/s (32-bit at 33 MHz)</td>
</tr>
<tr>
<td></td>
<td>266 MB/s (32-bit at 66 MHz or 64-bit at 33 MHz)</td>
</tr>
<tr>
<td></td>
<td>533 MB/s (64-bit at 66 MHz)</td>
</tr>
<tr>
<td><strong>Style</strong></td>
<td>Parallel</td>
</tr>
<tr>
<td><strong>Hotplugging interface</strong></td>
<td>Optional</td>
</tr>
</tbody>
</table>

PCI Express slots (from top to bottom: ×4, ×16, ×1 and ×16), compared to a traditional 32-bit PCI slot (bottom), as seen on DFI’s LanParty nF4 SLI-DR.
USB
The ultimate plug-in enabler!

Cup is not included
Pluggable Platforms: Compare/Contrast
Plug-ins
Plug-ins
Plug-ins
Isolation
1980s: BSD Unix to Microkernels

Monolithic Kernel based Operating System

Application

System Call

VFS

IPC, File System

Scheduler, Virtual Memory

Device Drivers, Dispatcher, ...

Hardware
1980s: BSD Unix to Microkernels
Software Isolated Programs

• Shared memory
  – Java, C#: safe user-level prog. language
  – Cyclone, Sing#: safe systems prog. language
  – Instrumentation (BGI, next slide)

• Message passing (distributed systems)
  – Bell Labs’ 5ESS
  – Modern web services/platforms
Byte-Granularity Isolation for Kernel Drivers (MSR Cambridge)
Tab Isolation

The browser as operating system

“In IE8 runs the browser frame and tabs in separate processes, which prevents glitches and hangs from bringing down the entire browser and leads to higher performance and scalability.”

“In IE8-9, tabs run without permissions to install software, modify settings, or change files of any user.”
Connectedness
Reiss' Field IDE

Figure 1. The Field environment.
Connectedness via COM, 1993

“The Component Object Model (COM) grew out of Microsoft’s efforts to make the various parts of its Office productivity suite work together.”

http://www.polberger.se/components/read/com.html
# My band's song list

## The Middle Third

### Play List 4.10

<table>
<thead>
<tr>
<th>Current Set List</th>
<th>Lead vocalist</th>
<th>Backup vocalist</th>
<th>Inst.</th>
<th>Keys</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. All I Wanna Do</td>
<td>E</td>
<td>Mo</td>
<td></td>
<td></td>
<td>Sheryl Crow</td>
</tr>
<tr>
<td>2. Back in the USSR</td>
<td>B</td>
<td>DT</td>
<td></td>
<td></td>
<td>Beatles</td>
</tr>
<tr>
<td>3. Bad Boy</td>
<td>B</td>
<td>DT</td>
<td></td>
<td></td>
<td>Beatles</td>
</tr>
<tr>
<td>4. Before He Cheats</td>
<td>B</td>
<td>DT</td>
<td></td>
<td></td>
<td>Carole Underwood</td>
</tr>
<tr>
<td>5. Call Me the Breeze</td>
<td>A</td>
<td>DT</td>
<td></td>
<td></td>
<td>Lynyrd Skynyrd</td>
</tr>
<tr>
<td>6. Chameleon</td>
<td>BB</td>
<td>BB</td>
<td></td>
<td></td>
<td>Herbie Hancock</td>
</tr>
<tr>
<td>7. Cold Shot</td>
<td>E</td>
<td>DT</td>
<td></td>
<td></td>
<td>Eric Clapton</td>
</tr>
<tr>
<td>8. Crossroads</td>
<td>E</td>
<td>DT</td>
<td></td>
<td></td>
<td>Chilli Peppers</td>
</tr>
<tr>
<td>9. Dani California</td>
<td>Amin</td>
<td>DM</td>
<td></td>
<td></td>
<td>Chilli Peppers</td>
</tr>
<tr>
<td>10. Eight Days A Week</td>
<td>C</td>
<td>Dm</td>
<td></td>
<td></td>
<td>Beatles</td>
</tr>
<tr>
<td>11. Groove is in the Heart</td>
<td>A</td>
<td>DT</td>
<td>Keys/Sax</td>
<td>D-Lites</td>
<td></td>
</tr>
<tr>
<td>12. Hum de Bump</td>
<td>Dmin</td>
<td>Dm</td>
<td></td>
<td></td>
<td>Chilli Peppers</td>
</tr>
<tr>
<td>13. I'm So Thankful</td>
<td>A</td>
<td>DM</td>
<td>Acoustic</td>
<td></td>
<td>Gangnam's Call</td>
</tr>
<tr>
<td>14. I'm the One</td>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td>AWB</td>
</tr>
<tr>
<td>15. Lajita Pacuita</td>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td>Tom Ball</td>
</tr>
<tr>
<td>16. Jump Live in Wall</td>
<td>B</td>
<td>DT</td>
<td>DM</td>
<td>Brian Setzer</td>
<td></td>
</tr>
<tr>
<td>17. Just What I Needed</td>
<td>E</td>
<td>DM</td>
<td></td>
<td></td>
<td>Brian Setzer</td>
</tr>
<tr>
<td>18. Love Shack</td>
<td>C</td>
<td>A</td>
<td>BT</td>
<td></td>
<td>B2's</td>
</tr>
<tr>
<td>19. Miss You</td>
<td>Amin</td>
<td>DM</td>
<td></td>
<td></td>
<td>Rolling Stones</td>
</tr>
<tr>
<td>20. My Best Friends Girl</td>
<td>Dm</td>
<td>-</td>
<td></td>
<td></td>
<td>The Cars</td>
</tr>
<tr>
<td>21. Pick Up the Pieces</td>
<td>F</td>
<td>-</td>
<td>Sax</td>
<td></td>
<td>AWB</td>
</tr>
<tr>
<td>22. Play That Funky Music</td>
<td>Emin</td>
<td>Mo</td>
<td>*</td>
<td>Wild Cherry</td>
<td></td>
</tr>
<tr>
<td>23. Price and Joy</td>
<td>E</td>
<td>BT</td>
<td></td>
<td></td>
<td>Steve Ray Vaughn</td>
</tr>
<tr>
<td>24. Rapture</td>
<td>Emin</td>
<td>Mo</td>
<td></td>
<td></td>
<td>Blondie</td>
</tr>
<tr>
<td>25. Romeo</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>26. San Juan Strut</td>
<td>G</td>
<td>-</td>
<td>Sax</td>
<td></td>
<td>Tom Ball</td>
</tr>
<tr>
<td>27. Sharp Dressed Man</td>
<td>A</td>
<td>BT</td>
<td>ZZ Top</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28. Shattered</td>
<td>E</td>
<td>DM</td>
<td></td>
<td></td>
<td>Rolling Stones</td>
</tr>
<tr>
<td>29. Spooky</td>
<td>E</td>
<td>DM</td>
<td></td>
<td></td>
<td>Classics IV</td>
</tr>
<tr>
<td>30. Spooky</td>
<td>E</td>
<td>DM</td>
<td></td>
<td></td>
<td>Classics IV</td>
</tr>
<tr>
<td>31. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>32. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>33. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>34. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>35. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>36. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>37. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>38. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>39. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>40. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>41. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>42. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>43. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>44. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>45. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>46. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>47. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>48. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>49. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>50. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>51. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>52. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>53. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>54. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>55. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>56. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>57. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>58. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>59. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>60. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>61. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>62. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>63. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>64. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>65. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>66. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>67. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>68. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>69. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>70. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>71. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>72. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>73. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>74. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>75. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>76. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>77. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>78. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>79. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>80. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>81. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>82. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>83. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>84. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>85. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>86. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>87. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>88. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>89. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>90. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>91. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>92. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>93. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>94. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>95. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>96. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>97. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>98. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>99. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
<tr>
<td>100. Sweet Sixteen</td>
<td>Emin</td>
<td>Mo</td>
<td>BT,DM</td>
<td>B2's</td>
<td></td>
</tr>
</tbody>
</table>
Key Ideas of COM

• **Binary standard** for components, classes, and interfaces
• **Memory layout** follows pure virtual C++ classes
• **Interface definition language**
• **Type libraries** (metadata)
• **Automation**: the ability for one program, typically a script, to access and control another
1990s: Connectedness (Web 1.0)

- 1993: Mosaic
- 1995: Java applets
- 1996: JavaScript, “DOM Level 0”
- 1996: Microsoft’s ActiveX
- 1998: “DOM Level 1” recommended by W3C
2000s: Buffer Overflow and Security Exploits

- Connectedness of code with many buffer overflows (lack of isolation) leads to a security crisis at Microsoft
- Security Definition Lifecycle
- Static tools for buffer overflows (SAL, PREfast)
- SAGE white box file fuzzing
- IE 8-9 security improvements
2000s: Connectedness (Web 2.0)

• Search and advertising (Google)
  – Connect consumers to producers via search and auction of ad terms
  – Plug-ins for site-targeted advertising

• Facebook
  – “Open graph” connects individuals, pages, groups, pictures, etc. in a huge graph
  – Plug-ins (apps) require access to your information to run
2011: Humans Plugged Into the Web

Welcome | ICSE 2011
The International Conference on Software Engineering (ICSE) is the premier software engineering conference, providing a forum for researchers, practitioners and ...
Clickjacking

• Fooling users into “liking” a page

• **Connect:** “Clickjacks succeed because people tend to trust information given to them on social networking sites, especially if it appears to have won the approval of several friends.”

• **Isolate:** “Preventing clickjacking attacks requires users trust no one.”
The Pendulum Keeps Swinging

Isolate

Connect

Security, Reliability, Availability, ...
Dependences
Unix Package Management

• Package Manager
  – The Unix system administrators best friend
  – Leading edge in SunOS -> Solaris (early 1990s)
  – Also in A/IX, HP/UX, even Windows from the early days

  – [Ian Murdock](http://en.wikipedia.org/wiki/Ian_Murdock) has commented that package management is "the single biggest advancement [Linux](http://en.wikipedia.org/wiki/Linux) has brought to the industry", that it blurs the boundaries between operating system and applications, and that it makes it "easier to push new innovations into the marketplace and evolve the OS".
Managing Dependences with SAT

Feature A requires
  (Feature B1 and ... and Feature Bn)
or ... or
  (Feature Z1 and ... and Feature Zm)

**Feature X** modeled by boolean variable **F_X**

**Requirements**(A) =
  F_A =>
    ( (F_B1 \ ... \ \ F_Bn)
    \ ... \ \n    (F_Z1 \ ... \ \ F_Zm) )
Feature Conflicts

Features A and B are incompatible

FeatureConflict(A,B) :

!F_A ∨ !F_B
Can We Combine Features A, B, and C?

Is the following formula satisfiable?

\[
F_A \land F_B \land F_C \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
\land \\
Managed Extensibility Framework (.NET 4)
Managed Extensibility Framework

• MEF presents a solution for the runtime extensibility problem.
• MEF provides a standard way for a host to expose and consume extensions
  – Extensions can be reused amongst different applications.
  – Extensions themselves can depend on one another
  – MEF will make sure extensions are wired together in the correct order
Comparison

MEF is to Visual Studio as Equinox is to Eclipse

Sort of ...
Protocols (Rules)
What are Rules? A Fact of Life!

• Proper temporal sequencing of calls to APIs
  – “Never call IoCompleteRequest while holding a spin lock. Attempting to complete an IRP while holding a spin lock can cause deadlocks.”

• Typestate: A Programming Language Concept for Enhancing Software Reliability, Strom, Yemini, 1986

• Message Sequence Charts
SpinLock Rule

I/O System

Entry Point

Driver

Device Driver Interface

KeAcquireSpinLock

KeReleaseSpinLock

Comments

This call is a reciprocal to **KeAcquireSpinLock**.

Parameters

**SpinLock**
Pointer to a spin lock for which the caller provides the storage.

**NewIrql**
Specifies the IRQL value saved from the preceding call to **KeAcquireSpinLock**.

Return Value

None

Headers

Declared in `wdm.h` and `ntddk.h`. Include `wdm.h` or `ntddk.h`.

Comments

This call is a reciprocal to **KeAcquireSpinLock**. The input **NewIrql** value must be the **OldIrql** returned by **KeAcquireSpinLock**.

For more information about spin locks, see [Spin Locks](#).

Callers of this routine are running at **IRQL = DISPATCH_LEVEL**. On return from **KeReleaseSpinLock**, **IRQL** is restored to the **NewIrql** value.
SpinLock Rule as State Machine

```c
enum {unlocked, locked} s = unlocked;
```

**I/O System**

**Device Driver Interface**

**Driver**

- KeAcquire SpinLock
- KeRelease SpinLock

**State Machine**

- **Unlocked**
  - Driver called
  - Driver returns

- **Locked**
  - Acquire
  - Release

- **Abort**
  - Release
  - Driver returns
SpinLock Rule (in SLIC language)

```c
state {
    enum {unlocked, locked} s = unlocked;
}

RunDispatchFunction.exit {
    if (s != unlocked) abort;
}

KeAcquireSpinLock.entry {
    if (s != unlocked) abort;
    else s = locked;
}

KeReleaseSpinLock.entry {
    if (s != locked) abort;
    else s = unlocked;
}
```
Source Code

API Usage Rules

Verification Engine

100% path coverage

Defects

Read for understanding

Automate testing

Static Analysis

Rules

Development

Testing
Facebook Protocols

User's Browser → Your App → Facebook

GET Your app's frontpage
Redirect

User's Browser ← Your App → Facebook

GET OAuth Dialog
302 Redirect including code parameter

User's Browser ← Your App → Facebook

GET Your app's callback URL

User's Browser ← Your App → Facebook

GET /oauth/authorize
Access Token

User's Browser ← Your App → Facebook

GET /me?access_token=...
API Response

User's Browser ← Your App → Facebook

Render user data in page
How to Shop for Free Online
Security Analysis of Cashier-as-a-Service Based Web Stores

Rui Wang¹, Shuo Chen², Xiaofeng Wang¹, Shaz Qadeer²
¹ Indiana University Bloomington ² Microsoft Research
Feature Interaction
Feature Interaction (Zave)

• In a software system, a feature is an increment of functionality, usually with a coherent purpose.

• If a system description is organized by features, then it probably takes the form $B + F_1 + F_2 + F_3 \ldots$
  – $B$ is a base description,
  – each $F_i$ is a feature module, and
  – “$+$” denotes some feature-composition operation.

• Consider $B=$Integrated Development Environment.
Eclipse JDT Extension Points

- org.eclipse.jdt.core.manipulation.changeMethodSignatureParticipants
- org.eclipse.jdt.debug.breakpointListeners
- org.eclipse.jdt.debug.javaLogicalStructures
- org.eclipse.jdt.junit.testRunListeners
- org.eclipse.jdt.ui.cleanUps
- org.eclipse.jdt.ui.foldingStructureProviders
- org.eclipse.jdt.ui.javaCompletionProposalComputer
- org.eclipse.jdt.ui.javaCompletionProposalSorters
- org.eclipse.jdt.ui.javaEditorTextHovers
- ...
Feature Interaction:
org.eclipse.jdt.debug.breakpointListeners

Description:

Allow clients to contribute listeners for Java breakpoint notifications. For example, listeners are called when a breakpoint is hit and about to suspend execution. The listener can vote to resume or suspend the debug session. ...
Some Visual Studio Extensions
From MSR

Editor
MSIL Rewriting

CLR Profiler

VS Debugger

Extension Points

Debugger Canvas
Smart Phones
Apps

• Installation requires users to permit app access to phone sensors

• Isolate, isolate, isolate

• Connectedness via copy/paste
TouchStudio: Script the Phone, On The Phone
Cloud Computing and Plug-ins
Cloud Computing = Complete Control
Programming the Salesforce Platform

• Developers can also use the Web services API to issue data manipulation commands ... Because the controlling logic for these client-side programs is not located on Force.com platform servers, they are restricted by:
  – The performance costs of making multiple round-trips to the salesforce.com site to accomplish common business transactions
  – The cost and complexity of hosting server code, such as Java or .NET, in a secure and robust environment

• To address these issues, and to revolutionize the way that developers create on-demand applications, salesforce.com introduces Force.com **Apex code, the first multitenant, on-demand programming language** for developers interested in building the next generation of business applications.
Human-based Computation
Example from MIT
Soylent: A Word Processor with a Crowd Inside

An IDE with a Crowd Inside?