EU-USA meeting, Venice, April 20, 21

Dependability and Open Source Software

(aka. Robust Open Source)

John Rushby
http://www.csl.sri.com/fm.html

Computer Science Laboratory
SRI International
Menlo Park, California, USA

Open Source: What Is It?

- A crusade, a software development model
  - Formerly, free software
- Main points of definition at `www.opensource.org`
  - Source code must be provided
  - No restrictions on redistribution (free or for profit)
  - No restriction on derivative works (i.e., modifications)
    But can require them to be distinguished from original

  Heretical variations exist (e.g, allowing restricted derivatives)
- Essential idea is that these conditions allow and foster cooperative software development by a distributed community of interested parties
- Examples: Linux, GNU, Apache, sendmail, Perl, Mozilla
- Availability of source code changes the business model

Open Source: Why The Buzz?

- The internet shifted focus from the desktop to servers
  - Everyone needs a web server; majority choose Apache (more than twice nearest rival, all four top sites)
  - And CGI-bin scripts; most are written in Perl
  - And a system to run it on; most big sites choose Unix, may of those use Linux (E.g., Yahoo!, Microsoft Hotmail)
- Major corporations started supporting Open Source products
  - IBM's WebSphere product line is built on Apache
  - Oracle, Informix etc. ported their DBMSs to Linux
  - Dell sells systems with Linux preinstalled
- And investing in companies associated with Open Source
  - IBM in RedHat (Linux distributor)
  - Intel in VA Research (Linux-based servers)

Open Source: Why The Buzz?

- "The Cathedral and the Bazaar" (by Eric Raymond of fetchmail) articulated the case for Open Source
  - "Given enough eyes, all bugs are shallow"
  - "If your source is open, you get peer review, you get reliability. If your source is not open, you don't get peer review and you don't get reliability, end of story"
- Contributed to decision of Netscape to go open source
- The "Halloween Documents" show Microsoft is concerned
  - "Open Source Software . . . promotes rapid creation and deployment of incremental features and bug fixes in an existing code / knowledge base. In recent years, . . . OSS projects have acquired the depth and complexity traditionally associated with commercial projects such as Operating Systems and mission critical servers"

Open Source: What Contributes To Success?

- Start with something that works
  - The community fixes bugs, provides enhancements
  - Cannot build from scratch this way (cf. Mozilla)
- It has to be something many people want to use, modify
  - S/W development tools, utilities, OS, internet tools
  - I.e., infrastructure, not applications
- Exploit the community of developers and users
  - Release early, release often
  - Use your own stuff (Bootstrap)
- Keep control
  - Strong modularity
  - There must be a coordinator or coordination process

Open Source: Contribution To Dependability?

- Context: assume component-based systems (buy, build, blend) using extensive off-the-shelf (OTS) technology
  - Formerly, OTS meant COTS
  - Suppose some of it is Open Source
- Causes of Undependability
  - When first deployed
    - ⋆ Component doesn't do what you thought it did
    - ⋆ Component is undependable
  - After something changes
    - ⋆ Something changes but component doesn't
    - ⋆ Component changes and you'd prefer it didn't

Component Doesn't Do What You Thought

- This is a major issue with COTS
  - Supplied as black boxes
  - Usually without accurate specifications
  - Specifications have to be reverse engineered
    - ⋆ Often incorrect and incomplete
- Less of a problem, and solvable, with Open Source
  - Have access to the code so reverse engineering of specifications is more reliable
  - Modularity tends to encourage clean/simple specifications
  - Since the process is open: could propose a specification to be adopted by all participants

Component Is Undependable

What's the data?

- Unix: number of utilities/functions with serious failures
- Fuzz (1990), 1995: Random testing of Unix utilities

HP	AIX	Solaris	Irix	Ultrix	Linux	GNU
(33%) 18%	(24%) 20%	(29%) 23%	15%	21%	9%	6%

Many bugs reported in 1990 were still present in 1995

- Ballista 1998: Random testing of Posix system calls

Failures	HP	AIX	Solaris	Irix	Ultrix	Linux	FreeBSD
Functions	51%	42%	45%	42%	53%	45%	56%
Rate	13%	10%	15%	13%	15%	13%	20%

Component Is Undependable

What's the data?

- Unix vs. NT
- 1997 DataPro survey

  Availability: Linux/Ultrix . . . the rest . . . NT

  Functionality: Ultrix, Linux . . . the rest . . . NT

  Performance: Ultrix, Linux . . . the rest . . . NT

  Satisfaction: Linux . . . Solaris . . . HP . . . the rest (incl. NT)
- Anecdotally (e.g., `http://www.unix-vs-nt.org`), Windows NT is far less dependable than any Unix
- Ballista data expected summer 1999
- The USS Yorktown: "A Navy official said that COTS is more challenging than expected—although the Navy has the license. . . they don't have access to its source code"

Component Is Undependable

- Available remedies
  - COTS
    - ⋆ Complain to the supplier
  - Open Source
    - ⋆ Complain to the community
    - ⋆ Fix it yourself
    - ⋆ Also new proactive opportunities: glass box testing
- Factors for success in Open Source also help dependability and maintainability
  - Start with something that works (i.e., spiral model)
  - Strong modularity
  - Avoid bloated code or features

Something Changes But Component Doesn't

- Change in requirements or environment
- COTS
  - Complain/request change from the supplier
    - ⋆ Supplier may not be interested
    - ⋆ Supplier may have moved on (e.g., Rational Ada)
    - ⋆ Supplier may have gone out of business
  - Work around it
- Open Source
  - Complain/request change from the community
  - Fix it yourself
  - Work around it

Component Changes and You'd Prefer It Didn't

- New release has different behavior or properties
- COTS
  - Business model requires gratuitous changes
  - Can stay with obsolete version, but lose other benefits
- Open Source
  - More concern for backward compatibility
  - Can do diffs to see what's really changed
  - Because source is available, can build customized version

Other Concerns and Vulnerabilities

- Deliberate Attacks
  - COTS
    - ⋆ Lack of "genetic diversity" leads to massive propagation, catastrophic collapse
  - Open Source
    - ⋆ Hackers have access as well, can find vulnerabilities
- Who do I call?
  - COTS
    - ⋆ Dependence on a single supplier
    - ⋆ May disappear, may not care
  - Open Source
    - ⋆ Emerging competitive market for professional support
    - ⋆ Ultimate remedy: do it yourself

Open Source: Possible Cooperative Initiatives

- More and better Open Source Software would contribute to more dependable systems
- Rework mandates for COTS to embrace Open Source
  - You may get better systems that way
  - Would improve competition
- Prime the pump for enhanced Open Source Software
  - Encourage researchers to develop/apply their work on Open Source platforms
  - Port needed components to Open Source
  - Develop new Open Source components
    E.g., RT/Embedded/FT "Linux"

Open Source: Possible Cooperative Initiatives (ctd.)

Exploit availability of source

- Use analysis tools to improve Open Source components
  - Glass box testing and analysis tools
  - Formal specification/mechanized analysis
  - Code visualization/management
  - Fault injection/dependability assessment

  Persuade development community to adopt these
- Research on new development/distribution methods
  - Trustworthy customization/composition/updates
  - Optimization of compositions (cf. Horus stack)
  - Model based (re)configuration
  - Proof carrying code (trust without the sandbox)