



Open Proofs: Maturing Formal Methods (FM) with Open Source Software (OSS) Approaches

David A. Wheeler

2009-03-11

This presentation contains the views of the author and does not indicate endorsement by IDA, the U.S. government, or the U.S. Department of Defense.



Key Terms

- **Open Source Software (OSS):**
 - Licenses give users the freedom to:
 - run the program for any purpose
 - study and modify the program, and
 - redistribute copies of either the original or modified program (without royalties, etc.)
 - OSS *is* commercial software
 - Antonym: Proprietary
- **Formal methods (FM): (For our purposes) Application of mathematics to (model of) software/hardware to prove properties (“it will always/never do X”)**
 - Vs. testing: can’t completely test trivial programs



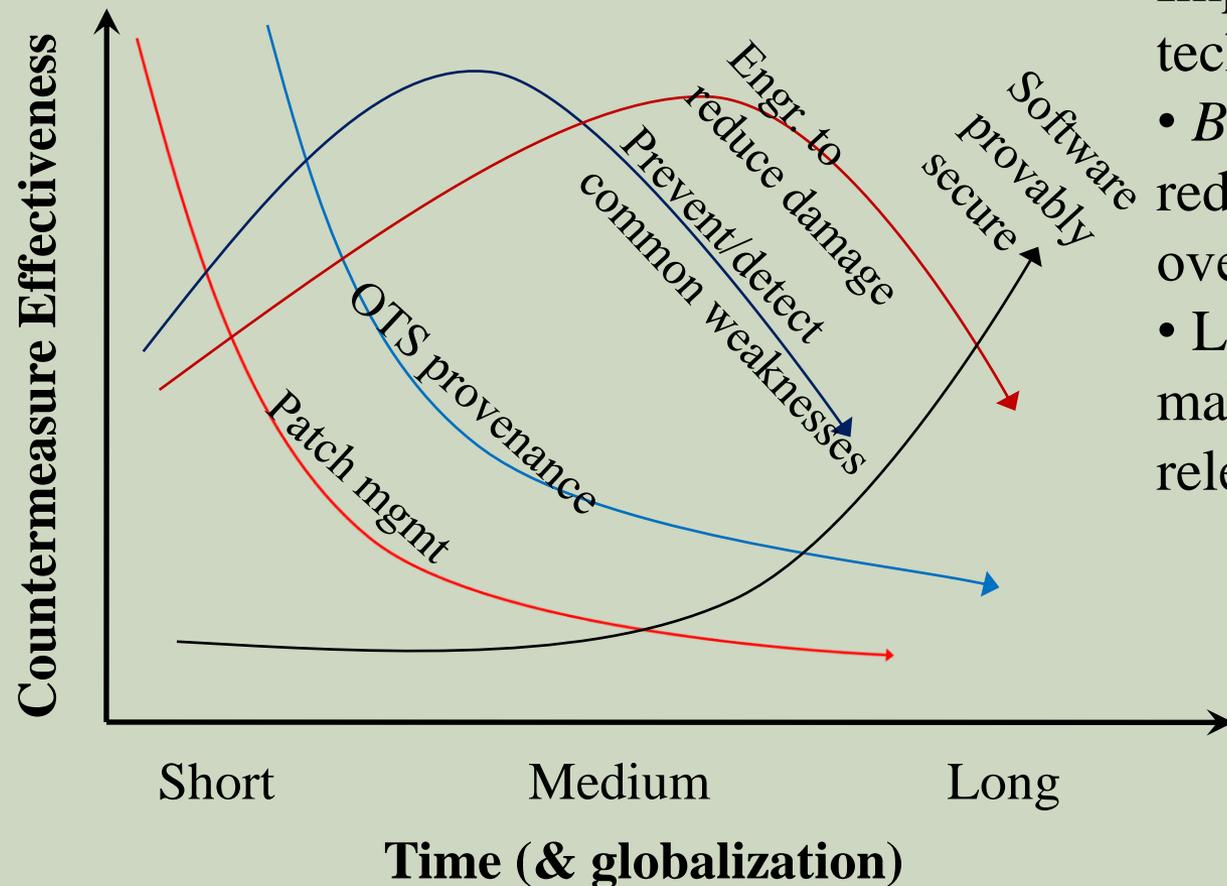
Outline

- **Long-term need: Provably secure software**
- **What's slowing FM maturation?**
 - **“Culture of secrecy”**
 - **Need a “working ecosystem”**
- **New idea: “Open Proofs”**
 - **OSS: Implementation & Proofs & Tools**
- **Making “open proofs” a reality (status)**
- **Observations**

<http://www.openproofs.org>



Long term: If we want secure software, we'll need to prove it



- Short/Medium term: Implement variety of techniques
- *But* adversaries adapt, reducing effectiveness over time. Persist, but...
- Long term: Need to make systems secure at release, and *know* it...
 - DoD SA CONOPS requires high assurance toolkit of open standards (IA) components
 - Leads to FM 4



What's slowing FM maturation?

- Much research & some use, but FM tools are often:
 - Hard to install, hard to learn to use
 - Hard to use, time-consuming, & don't scale
 - Poorly integrate with other tools/existing environments
- Need to mature FM if they're to be broadly used
 - Hard problem, relatively few research \$... but decades?
- FM maturation hindered by “*cult of secrecy*”
 - Details of FM use often unpublished, classified
 - Details of FM tools (& the tools!) often unshared/lost
- Result (broadly stated):
 - Researchers/toolmakers receive inadequate feedback
 - From developers & other researchers/toolmakers
 - Researchers/toolmakers waste time/\$ rebuilding tools
 - Educators difficulty explaining (esp. without examples)
 - Developers don't understand, uncertain value
 - Evaluators/end-users don't know what to look for





Researchers/toolmakers need more than papers: LINMAT to NANOSAT

Researchers/toolmakers suffer from lack of information

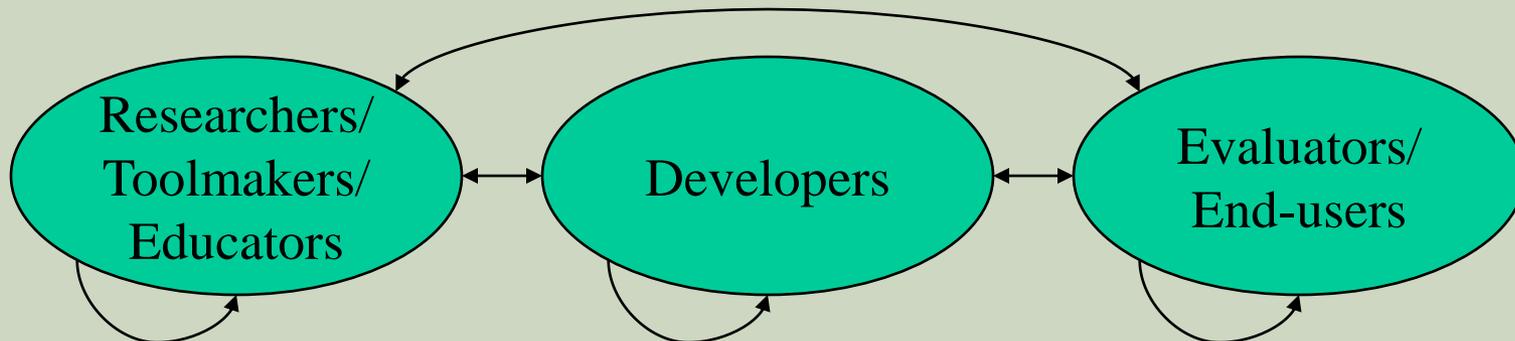
LINMAT/NANOSAT developers: "From the publications alone, without access to the source code, various details were still unclear... what we did not realize, and which hardly could be deduced from the literature, was [an optimization] employed in GRASP and CHAFF [was critically important]... Only [when CHAFF's source code became available did] our unfortunate design decision become clear... The lesson learned is, that important details are often omitted in publications and can only be extracted from source code. It can be argued, that making source code of SAT solvers available is as important to the advancement of the field as publications"

- [Biere, "The Evolution from LINMAT to NANOSAT", Apr 2004]



Need: working ecosystem

- **Researchers/Toolmakers/Educators**
 - Learn *details* from others (papers often inadequate)
 - Build on/experiment with existing tools (vs. rebuilding)
- **Developers of implementations to be proved**
 - Learn from other developers
 - Build on/experiment with proven systems/components
 - Share issues with toolmakers (so tools can improve)
- **Evaluators/End-users**
 - Evaluate evidence (determine adequacy, give feedback)
 - Evaluate other systems based on this experience





“Open proof” idea

- **“Open proof” (new term):**
 - Source code, proofs, and required tools: OSS
- **Anyone can examine/critique, improve upon, collaborate with others for improvements**
 - Not just software, but what’s proved & tools
 - Example for training, or as useful component
- **Extends OSS idea for high assurance**
 - Enables legal collaboration
 - Similar to mathematics field
 - Method for speeding up tech transition
- **Goal: Make supplier identity irrelevant**
- **Don’t need *everything* to be an open proof**
 - Examples & building blocks (inc. standards’ API)



Making “open proofs” a reality

- **Establish collaboration mechanism**
 - ✓ Identified OSS FM tools, select initial “most promising”
 - ✓ Set up website (controlled Wiki & mailing list)

<http://www.openproofs.org>

- **Publicize broadly, encourage collaboration**
- **Encourage/aid community to:**
 - Package “most promising” tools for easy installation
 - Encourage integration/improvement of tools
 - Encourage/aid development of open proofs
 - Examples & components to build on
 - Foster Community of Interest (COI) to ID & deliver “top 10” components as proven OSS



Example: “The Why Stack” (ProVal)

Annotated C Annotated C Annotated Java

Frama-C +
Jessie

Caduceus

Krakatoa

Why

Automated
provers

Interactive
provers

Alt-Ergo

Zenon

CVC3

miniSAT

...

Coq

...

PVS

OSS can usually
be combined (if
common licenses)



Packaging

- **Package: Make promising tools “one click to install” using standard OS tools**
 - Necessary first step
 - Started with Fedora & 23 promising tools

Status	%	Tools
Packaged via Open-Proofs	~40% (9/23)	Prover9, Coq, Why/ Caduceus/ Krakatoa, Alt-Ergo, Zenon, MiniSAT2, tex-zfuzz, STP, E
Packaging in progress	~30% (7/23)	PVS, ACL2, BLAST, DiVinE [†] , Proof General, Frama-C/Jessie, KodKod
Packaging to do	~30% (7/23)	KeY, Alloy, HOL 4, Isabelle/HOL, HOL Lite, Gandalf, mCRL2

Fedora Packaging Status, 2009-03-10

CVC3 & NuSMV omitted due to licensing issues



Consistent with USG Direction

- **Vivek Kundra appointed as USG CIO by President**
 - “[USG must] ensure the public has access to information, and [must] rethink the way the public interacts with the government in an information economy.... [the USG will implement this by] embracing off-the-shelf applications, cloud computing, open-source technology, and concepts that encourage citizens to self-organize on the Web.”
 - “We need to make sure that's all that data that's not private or restricted for national security reasons be made public”
 - “Using off-the-shelf, as well as open-source technologies ... could result in significant savings for the federal government”
- “Obama's CIO wants more citizen activity on Web”, Cnet, Stephanie Condon, 2009-03-05, http://news.cnet.com/8301-13578_3-10190069-38.html



Observations

- If releasing software as OSS, don't write a new OSS license
 - OSS for collaboration; incompatible licenses defeat that
 - OSS is standardized on a few licenses; use them!
 - MIT, BSD-new, LGPL, GPL; ~Apache 2.0
 - Legal to include, legal to combine, widely used
- Much research software not commercialized → disappears
 - Not OSS nor supported proprietary (Z/Eves, ESC/Java)
 - Companies: If won't sell as proprietary, release as OSS!
 - “We the people” should get unclass code we've paid for
 - Government-funded research sw: “OSS by default”
- Many FM tools (focus on different issues)
- FM needs more research \$
- Please help/join us!

<http://www.openproofs.org>



Backup Slides



The Good News...

- **OSS development tends to involve collaborative review**
 - **Consistent with mathematical approaches, also tend to involve collaborative review**
- **Many OSS community leaders have extensive mathematical training**
- **Many HA/FM tools are OSS**
 - **Many OSS developers avoid depending on proprietary tools (“Java Trap”)**
 - **Easier to distribute OSS tools (e.g., place in Linux distribution repositories)**



Many OSS tools support high assurance

- **Configuration Management**: CVS, Subversion (SVN), GNU Arch, git/Cogito, Bazaar, Bazaar-NG, Monotone, Mercurial, Darcs, svk, Aegis, CVSNT, FastCST, OpenCM, Vesta, Supersversion, Arx, Codeville...
- **Testing**: opensourcetesting.org lists 275 tools Apr 2006, inc. Bugzilla (tracking), DejaGnu (framework), gcov (coverage), ...
- **Formal methods**: Community Z tools (CZT), ZETA, ProofPower, Overture, ACL2, Coq, E, Otter/MACE, PTPP, Isabelle, HOL4, HOL Light, Gandalf, Maude Sufficient Completeness Checker, KeY, RODIN, Hybrid Logics Model Checker, Spin, NuSMV 2, BLAST, Java PathFinder, SATABS, DiVinE, Splint (as LCLint), ...
- **Analysis implementation**: Common LISP (CMUCL, Steel Bank CL), Scheme, Prolog (GNU Prolog, SWI-Prolog, Ciao Prolog, YAP), Maude, Standard ML, Haskell (GHC), ...
- **Code implementation**: C/C++/Ada (gcc/GNAT), Java, ...



Why should this work (now)?

- **FM has had some success, but only niches**
 - **Bowen & Hinchey, “The Use of Industrial-Strength Formal Methods” (survey)**
 - **Larsen’s “A Survey of Industrial Applications of Formal Methods”, FM 08 (id’s problems)**
- **Rise of COTS & globalization: “Software by only cleared U.S.” impractical, & testing hopeless**
 - **Need different software development approach**
- **Reasons for hope**
 - **Much faster machines & improved analysis approaches (esp. SAT, FP)**
 - **OSS approaches proven for large systems**
 - **OSS tools address cost, vendor lock-in, flexibility**
 - **OSS FM tools exist & can be combined**



OSS is commercial, extant OSS is COTS

- **U.S. Law (41 USC 403), FAR, & DFARS: OSS is commercial!**
 - Commercial item is “(1) Any item, other than real property, that is of a type customarily used by the general public or by non-governmental entities for purposes [not government-unique], and (i) Has been sold, leased, or licensed to the general public; or (ii) Has been offered for sale, lease, or license to the general public... (3) [Above with] (i) Modifications of a type customarily available in the commercial marketplace; or (ii) Minor modifications... made to meet Federal Government requirements...”
 - Intentionally broad; "enables the Government to take greater advantage of the commercial marketplace" [DoD AT&L]
- **Dept. of the Navy “OSS Guidance” (June 5, 2007) confirms**
- **17 USC 101: OSS projects’ improvements = financial gain**
 - 17 USC 101: “financial gain” inc. “receipt, or expectation of receipt, of anything of value, including the receipt of other copyrighted works.”
- **OMB Memo M-03-14 (Commercial software): OSS support**
- **Important: U.S. Law (41 USC 403), FAR, DFARS require U.S. gov’t contracts prefer commercial items (inc. COTS) & NDI:**
 - Agencies must “(a) Conduct market research to determine [if] commercial items or nondevelopmental items are available ... (b) Acquire [them] when... available ... (c) Require prime contractors and subcontractors at all tiers to incorporate, to the maximum extent practicable, [them] as components...”



OSS is clearly commercial by other measures too

- Many OSS projects supported by commercial companies
 - IBM, Sun, Red Hat (solely OSS, market cap \$4.3B), Novell, Microsoft (WiX, IronPython, SFU, Codeplex site)
- Big money in OSS companies
 - Citrix bought XenSource (\$500 million), Sun bought MySQL (\$1 billion), Red Hat bought JBoss (\$350 million; *OSS buys OSS*)
 - IBM reports invested \$1B in 2001, made it back in 2002
 - Venture capital invested \$1.44B in OSS 2001-2006 [InfoWorld]
- Paid developers
 - Linux: 37K/38K changes in 2004; Apache, X Window system
- OSS licenses/projects approve of commercial support
- Sell service/hw, commoditize complements, avoid costs
- Users use COTS/NDI because they share costs – OSS does!
- Even GPL'ed software is commercial