Modeling and Simulating Free/Open Source Software Development Processes

Walt Scacchi
Institute for Software Research
School of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
Wscacchi@ics.uci.edu

Overview

- Background
- Results from recent studies of F/OSS
- F/OSS processes
- Implications for Software Process Modeling and Simulation
- Conclusions

What is free/open source software development?

- Free (as in “freedom”) vs. open source
  - Freedom to access, browse/view, study, modify and redistribute the source code
  - Free is always open, but open is not always free
- F/OSSD is not “software engineering”
  - *Different*: F/OSSD can be faster, better, and cheaper than SE
- F/OSSD involves *more* software development tools, Web resources, and personal computing resources

Who is investing in OSSD?

- Large corporations: (IT and Financial)
  - IBM-*Eclipse*, Sun-*NetBeans* and *OpenOffice*, HP-*Gelato*, Apple-*Darwin*, Microsoft Research-*Rotor*, SAP-*DB*, etc.
  - Barclays Global Investors, DKW, Merrill Lynch, UBS
- Mid-size corporations:
  - RedHat, Novell, Borland, BEA Systems
- Small (start-up) companies:
  - ActiveState (now part of Sophos), Collab.Net, Jabber, Ximian (now part of Novell), JBoss, Compiere, etc.
- Government agencies:
  - U.S.: DoD, Energy, NSF

Findings from F/OSS Studies

- *CIO* 2002-2003:
  - OSS primarily for new system deployments
  - OSS benefits
    - enable lower TCO
    - lower capital investment
    - greater reliability
  - OSS weaknesses:
    - lack of in-house skills or skills in labor market,
    - lack of vendor support or vendor viability
    - switching costs

Findings from F/OSSD Studies

- Hars and Ou 2002:
  - >60% of F/OSS developers work on 2-10 F/OSS projects
- Madey, et al. 2003:
  - <5% of OSS projects on SourceForge.net sustained; >90% have only one contributor (i.e., Power Law)
- Nichols and Twidale 2003:
  - Usability of F/OSS systems generally neglected
- Scacchi 2002-2004:
  - Largest F/OSSD projects sustain exponential growth; most F/OSSD projects fail to grow to any sustainable effort

Sample processes for F/OSSD

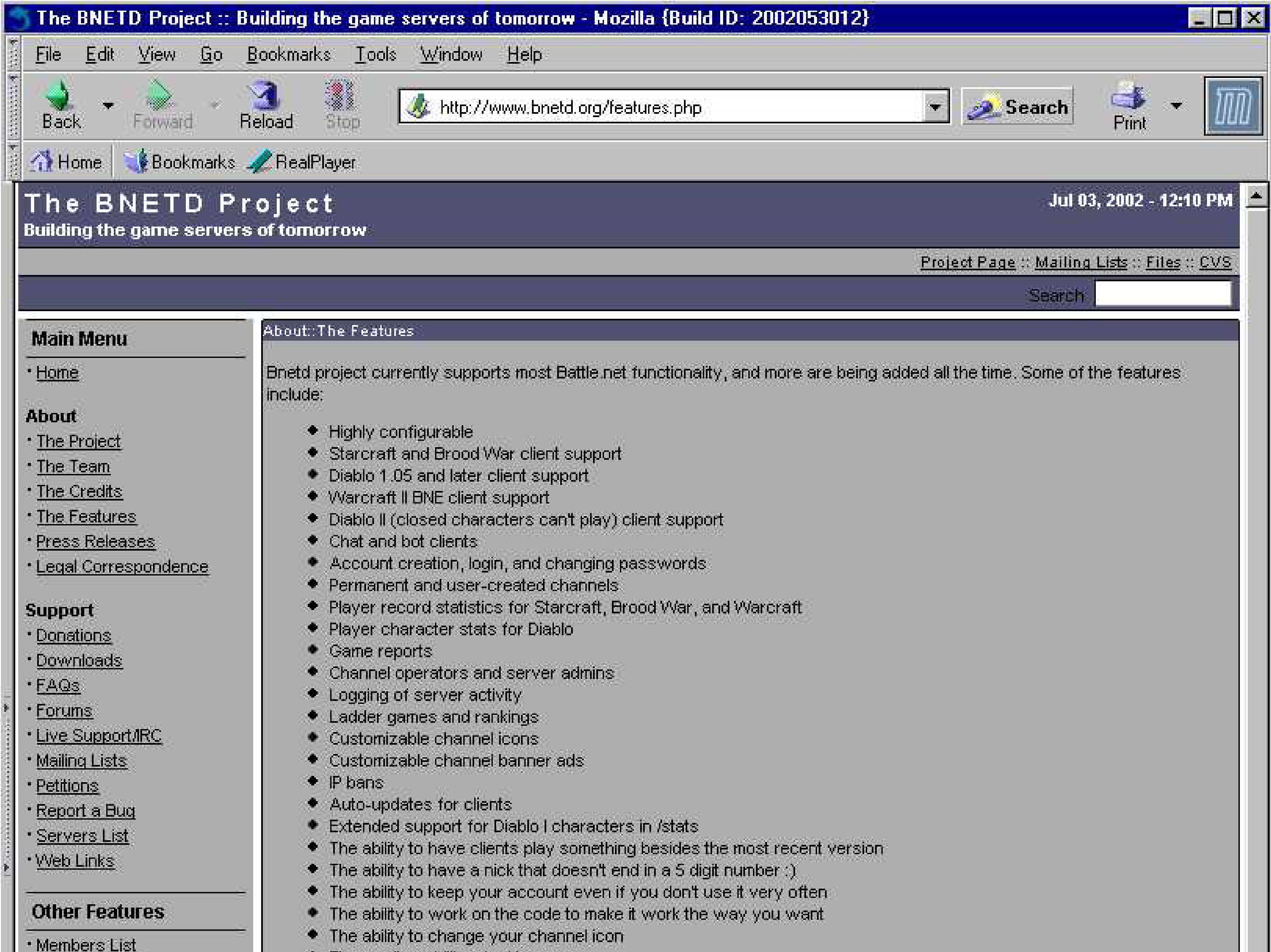
- Requirements and design
- Configuration management and work coordination
- Maintenance/Evolution
- Project management/career development
- Software technology transfer and licensing

F/OSS Processes for Requirements or Design

- F/OSS Requirements/Designs
  - Not explicit, not formal
  - Downstream output, not upstream input
- F/OSS Requirements/Designs are embedded within “informalisms”
  - Example OSS informalisms to follow (as screenshot displays)
- F/OSS Requirements/Design processes are different from their SE counterparts.

SE vs. F/OSS processes for Requirements

• Elicitation	• *Post-hoc* assertion
• Analysis	• Reading, sense-making, accountability
• Specification and modeling	• Continually emerging webs of discourse
• Validation	• Condensing and hardening discourse
• Communicating and managing	• Global access to discourse


The BNETD Project :: Building the game servers of tomorrow - Mozilla {Build ID: 2002053012}
File
Edit
View
Go
Bookmarks
Tools
Window
Help
Back
Forward
Reload
Stop
http://www.bnetd.org/features.php
Search
Print
Home
Bookmarks
RealPlayer

The BNETD Project

Building the game servers of tomorrow

Jul 03, 2002 - 12:10 PM

Project Page :: Mailing Lists :: Files :: CVS

Search

Main Menu

- Home

About

- The Project
- The Team
- The Credits
- The Features
- Press Releases
- Legal Correspondence

Support

- Donations
- Downloads
- FAQs
- Forums
- Live Support/IRC
- Mailing Lists
- Petitions
- Report a Bug
- Servers List
- Web Links

Other Features

- Members List

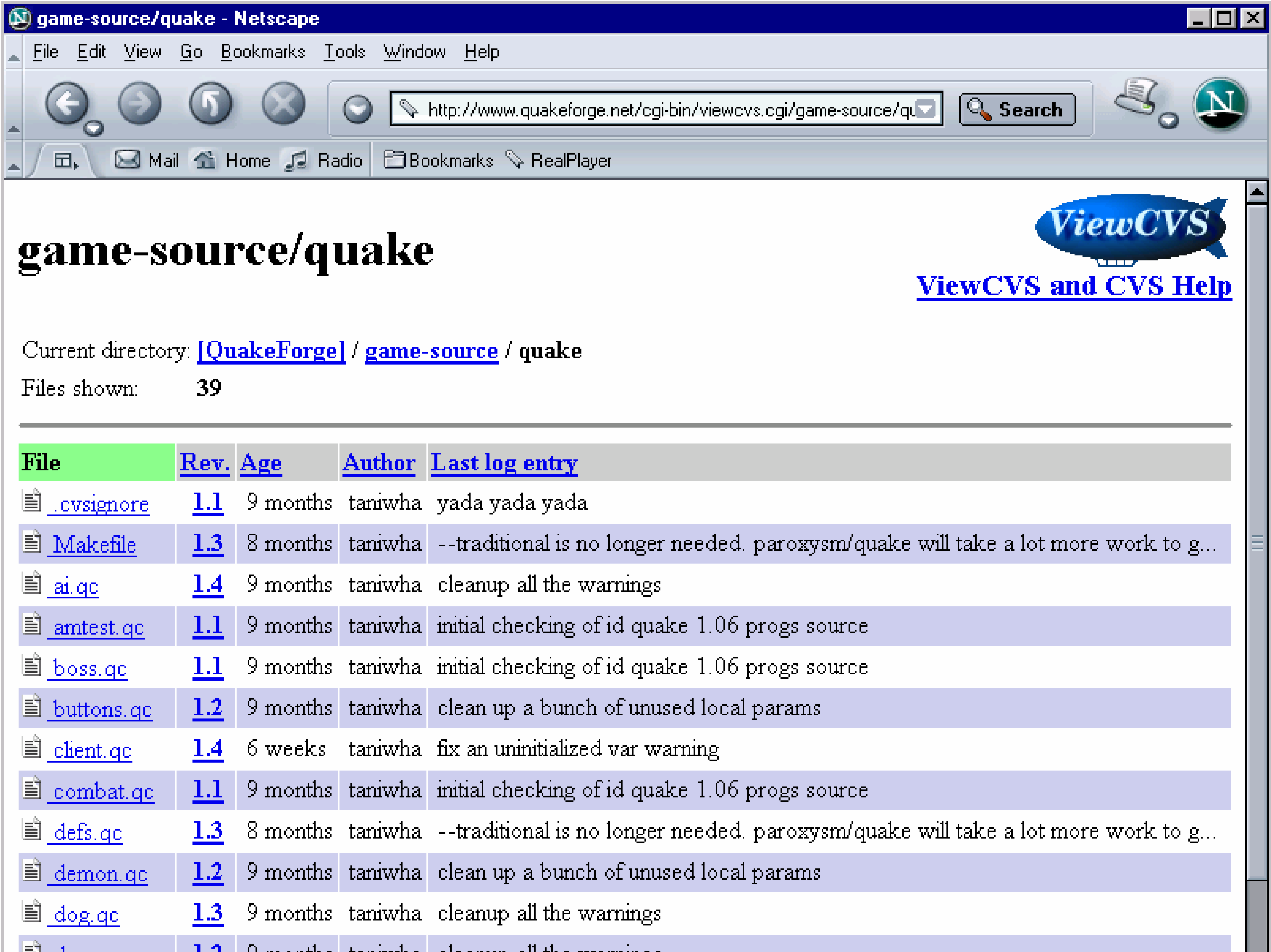
About::The Features

Bnetd project currently supports most Battle.net functionality, and more are being added all the time. Some of the features include:

- Highly configurable
- Starcraft and Brood War client support
- Diablo 1.05 and later client support
- Warcraft II BNE client support
- Diablo II (closed characters can't play) client support
- Chat and bot clients
- Account creation, login, and changing passwords
- Permanent and user-created channels
- Player record statistics for Starcraft, Brood War, and Warcraft
- Player character stats for Diablo
- Game reports
- Channel operators and server admins
- Logging of server activity
- Ladder games and rankings
- Customizable channel icons
- Customizable channel banner ads
- IP bans
- Auto-updates for clients
- Extended support for Diablo I characters in /stats
- The ability to have clients play something besides the most recent version
- The ability to have a nick that doesn't end in a 5 digit number :)
- The ability to keep your account even if you don't use it very often
- The ability to work on the code to make it work the way you want
- The ability to change your channel icon

Configuration management and work coordination

- Use CM to coordinate and control who gets to update what part of the system
  - Many F/OSSD projects use **CVS** (single centralized code repository with update locks) and frequent releases (*daily releases* on active projects)
  - Linux Kernel: **BitKeeper** (multiple parallel builds and release repositories)
  - Collab.Net and Tigris.org: **Subversion** (CVS++)
  - Apache: Single major release, with frequent "*patch*" releases (e.g., "A patchy server")
  - GNU **arch**: Bitkeeper + Subversion + (more)



game-source/quake - Netscape
File Edit View Go Bookmarks Tools Window Help
http://www.quakeforge.net/cgi-bin/viewcvs.cgi/game-source/qu
Search
Mail
Home
Radio
Bookmarks
RealPlayer
ViewCVS

game-source/quake

ViewCVS and CVS Help

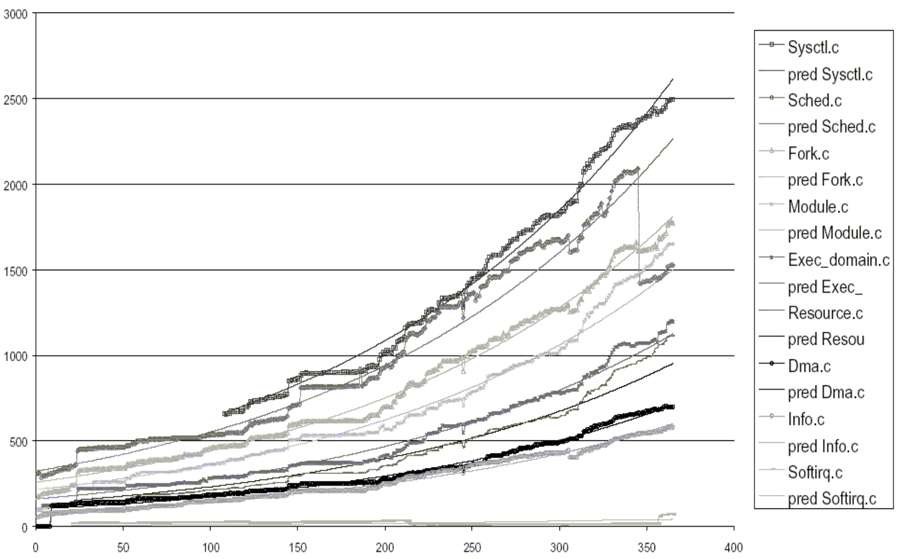
Current directory: [QuakeForge] / game-source / quake

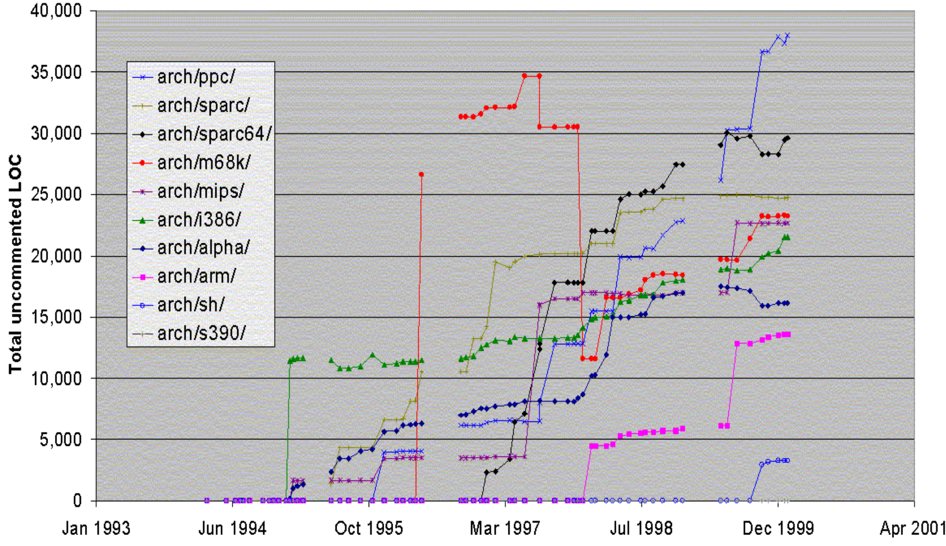
Files shown: 39

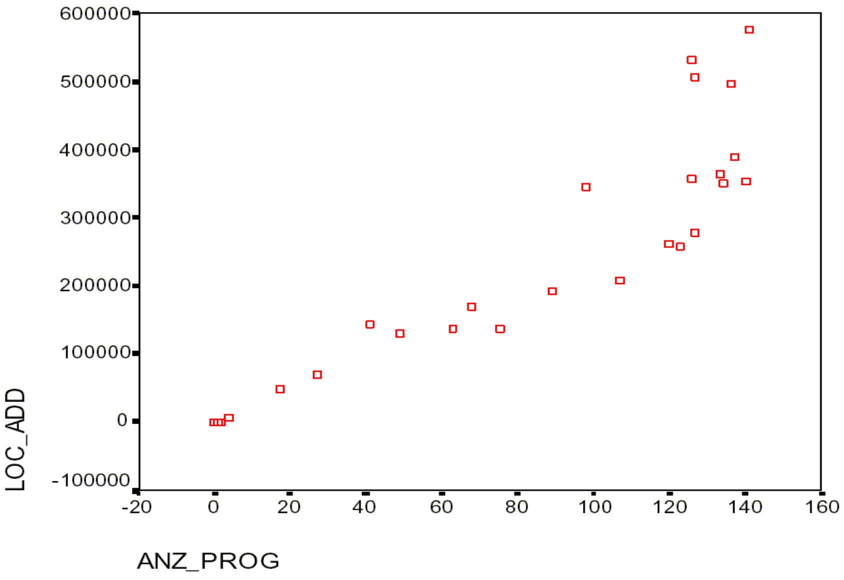
File	Rev.	Age	Author	Last log entry
.cvsignore	1.1	9 months	taniwha	yada yada yada
Makefile	1.3	8 months	taniwha	--traditional is no longer needed. paroxysm/quake will take a lot more work to g...
ai.qc	1.4	9 months	taniwha	cleanup all the warnings
amtest.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
boss.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
buttons.qc	1.2	9 months	taniwha	clean up a bunch of unused local params
client.qc	1.4	6 weeks	taniwha	fix an uninitialized var warning
combat.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
defs.qc	1.3	8 months	taniwha	--traditional is no longer needed. paroxysm/quake will take a lot more work to g...
demon.qc	1.2	9 months	taniwha	clean up a bunch of unused local params
dog.qc	1.3	9 months	taniwha	cleanup all the warnings

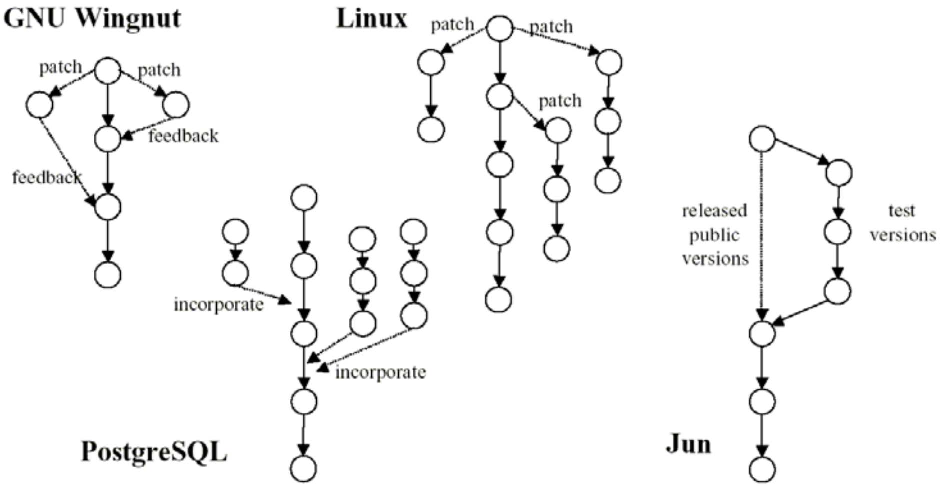
Evolution: Reinvention, redistribution, and revitalization

- Overall evolutionary dynamic of F/OSSD is *reinvention*
  - Reinvention enables continuous improvement
- F/OSS evolve through minor mutations
  - Expressed, recombined, redistributed via incremental releases
- F/OSS systems *co-evolve* with their development community
  - Success of one depends on the success of the other
- Closed legacy systems may be *revitalized* via opening and redistribution of their source
  - When enthusiastic user-developers want their cultural experience with such systems to be maintained.


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400
Sysctl.c
pred Sysctl.c
Sched.c
pred Sched.c
Fork.c
pred Fork.c
Module.c
pred Module.c
Exec_domain.c
pred Exec_
Resource.c
pred Resou
Dma.c
pred Dma.c
Info.c
pred Info.c
Softirq.c
pred Softirq.c

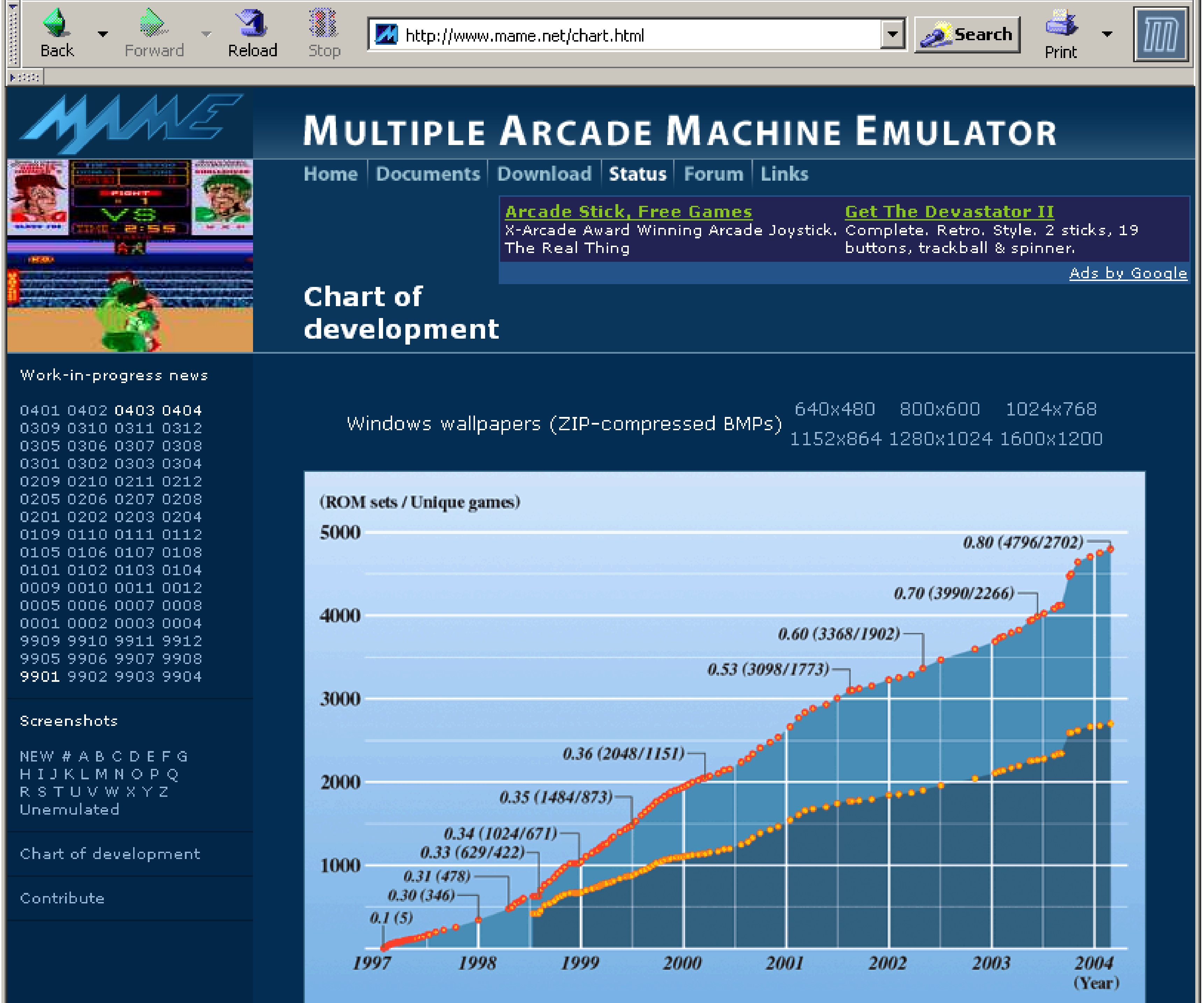

40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
Total uncommented LOC
arch/ppc/
arch/sparc/
arch/sparc64/
arch/m68k/
arch/mips/
arch/i386/
arch/alpha/
arch/arm/
arch/sh/
arch/s390/
Jan 1993
Jun 1994
Oct 1995
Mar 1997
Jul 1998
Dec 1999
Apr 2001


600000
500000
400000
300000
200000
100000
0
-100000
LOC_ADD
-20
0
20
40
60
80
100
120
140
160
ANZ_PROG


GNU Wingnut
patch
patch
feedback
feedback
Linux
patch
patch
patch
incorporate
incorporate
PostgreSQL
released public versions
test versions
Jun

Ins

Univer



http://www.mame.net/chart.html

Multiple Arcade Machine Emulator

Home | Documents | Download | Status | Forum | Links

Chart of development

Work-in-progress news

0401 0402 0403 0404
0309 0310 0311 0312
0305 0306 0307 0308
0301 0302 0303 0304
0209 0210 0211 0212
0205 0206 0207 0208
0201 0202 0203 0204
0109 0110 0111 0112
0105 0106 0107 0108
0101 0102 0103 0104
0009 0010 0011 0012
0005 0006 0007 0008
0001 0002 0003 0004
9909 9910 9911 9912
9905 9906 9907 9908
9901 9902 9903 9904

Screenshots

NEW # A B C D E F G
H I J K L M N O P Q
R S T U V W X Y Z
Unemulated

Chart of development

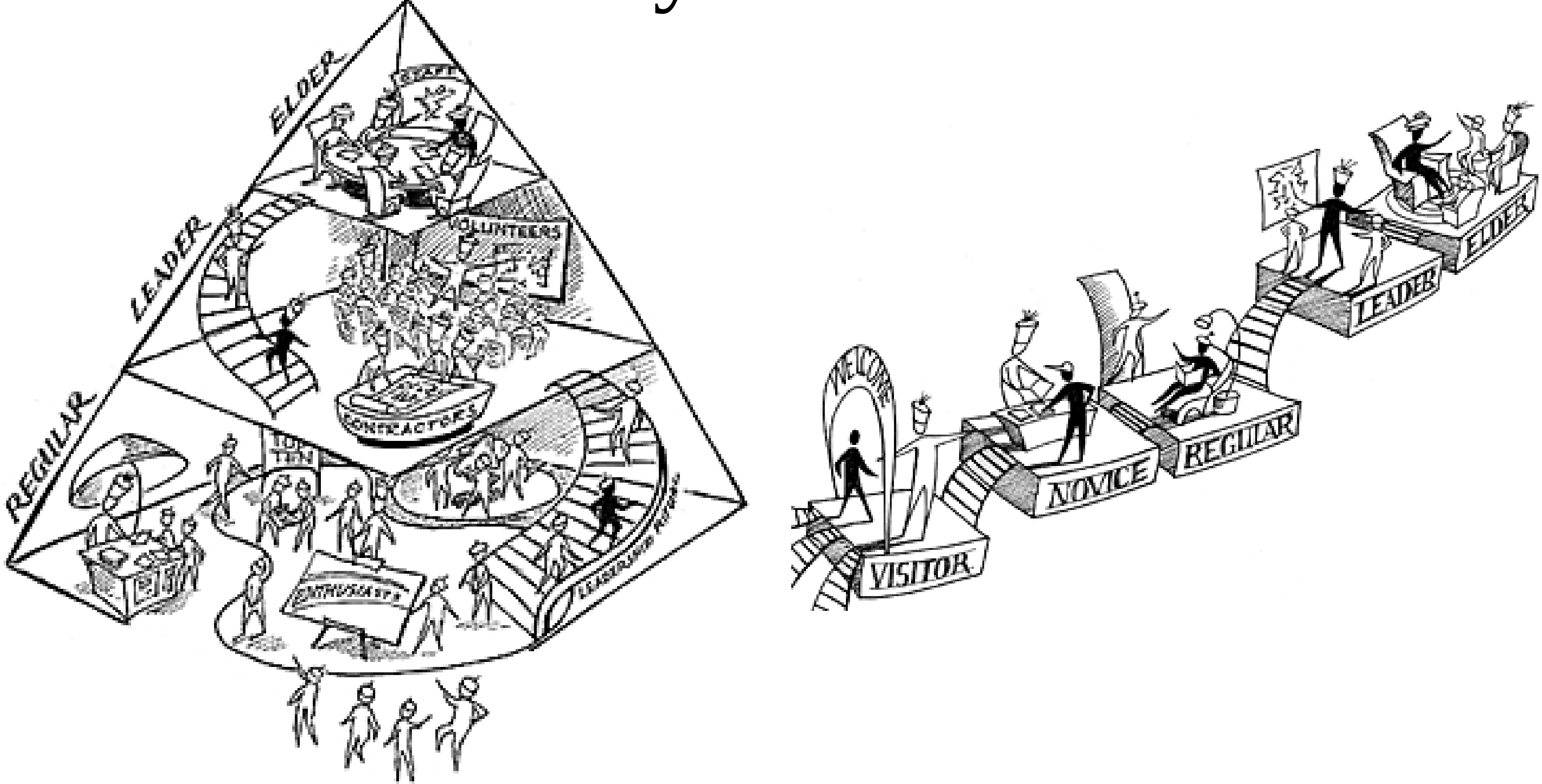
Contribute

Windows wallpapers (ZIP-compressed BMPs) 640x480 800x600 1024x768 1152x864 1280x1024 1600x1200

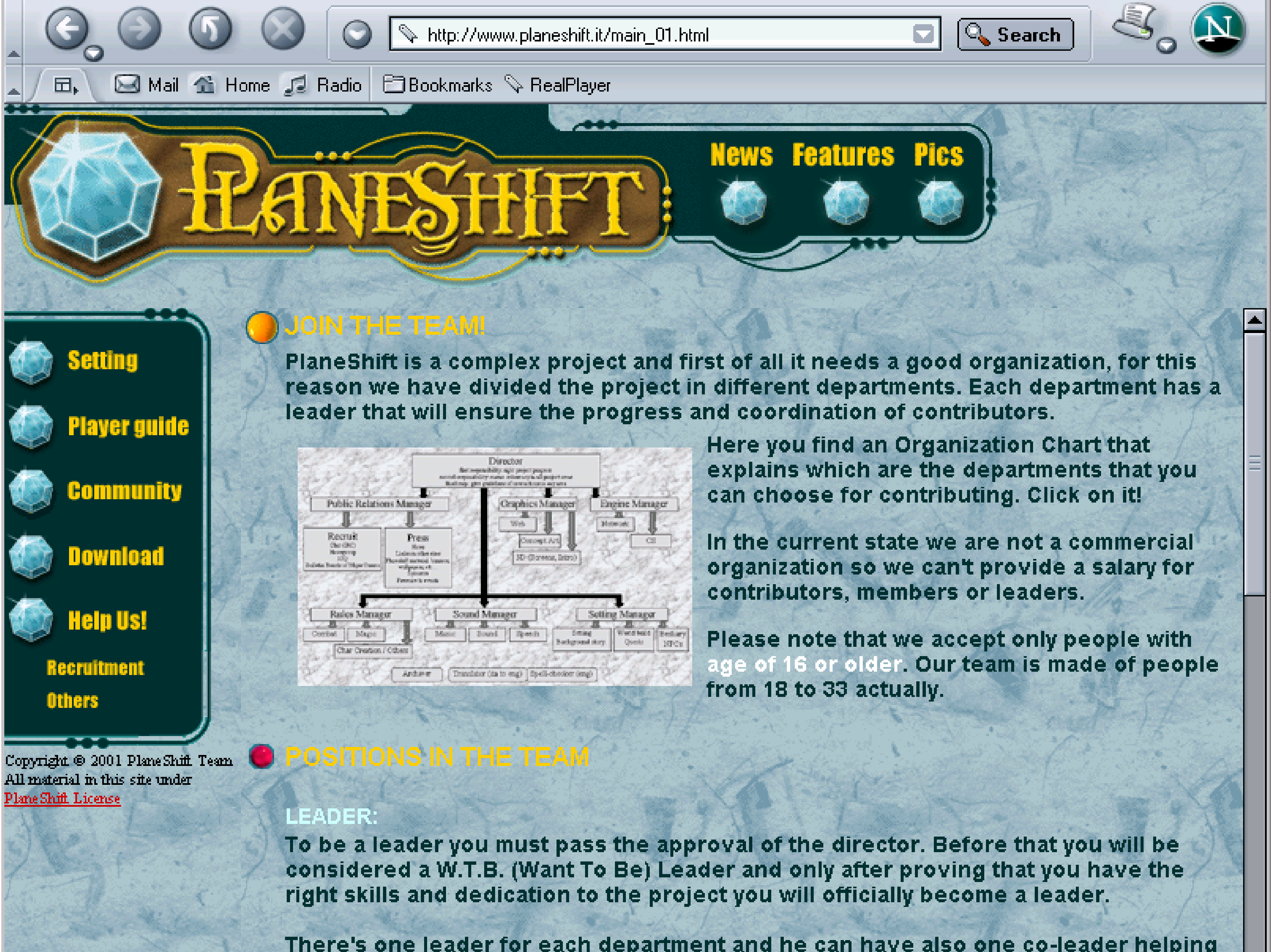
Project management and career development

- F/OSSD projects self-organize as a *layered meritocracy* via *virtual project management*
  - Meritocracies embrace incremental mutations over radical innovations
  - VPM requires people to act in leadership roles based on skill, availability, and belief in project community
- F/OSS developers want to have fun, exercise their technical skill, try out new kinds of systems to develop, and/or interconnect multiple F/OSSD projects (*freedom of choice and expression*).

A layered meritocracy and role hierarchy for F/OSSD



(images from A.J. Kim, *Community Building on the Web*, 2000)



JOIN THE TEAM!

PlaneShift is a complex project and first of all it needs a good organization, for this reason we have divided the project in different departments. Each department has a leader that will ensure the progress and coordination of contributors.

Here you find an Organization Chart that explains which are the departments that you can choose for contributing. Click on it!

In the current state we are not a commercial organization so we can't provide a salary for contributors, members or leaders.

Please note that we accept only people with age of 16 or older. Our team is made of people from 18 to 33 actually.

POSITIONS IN THE TEAM

LEADER:

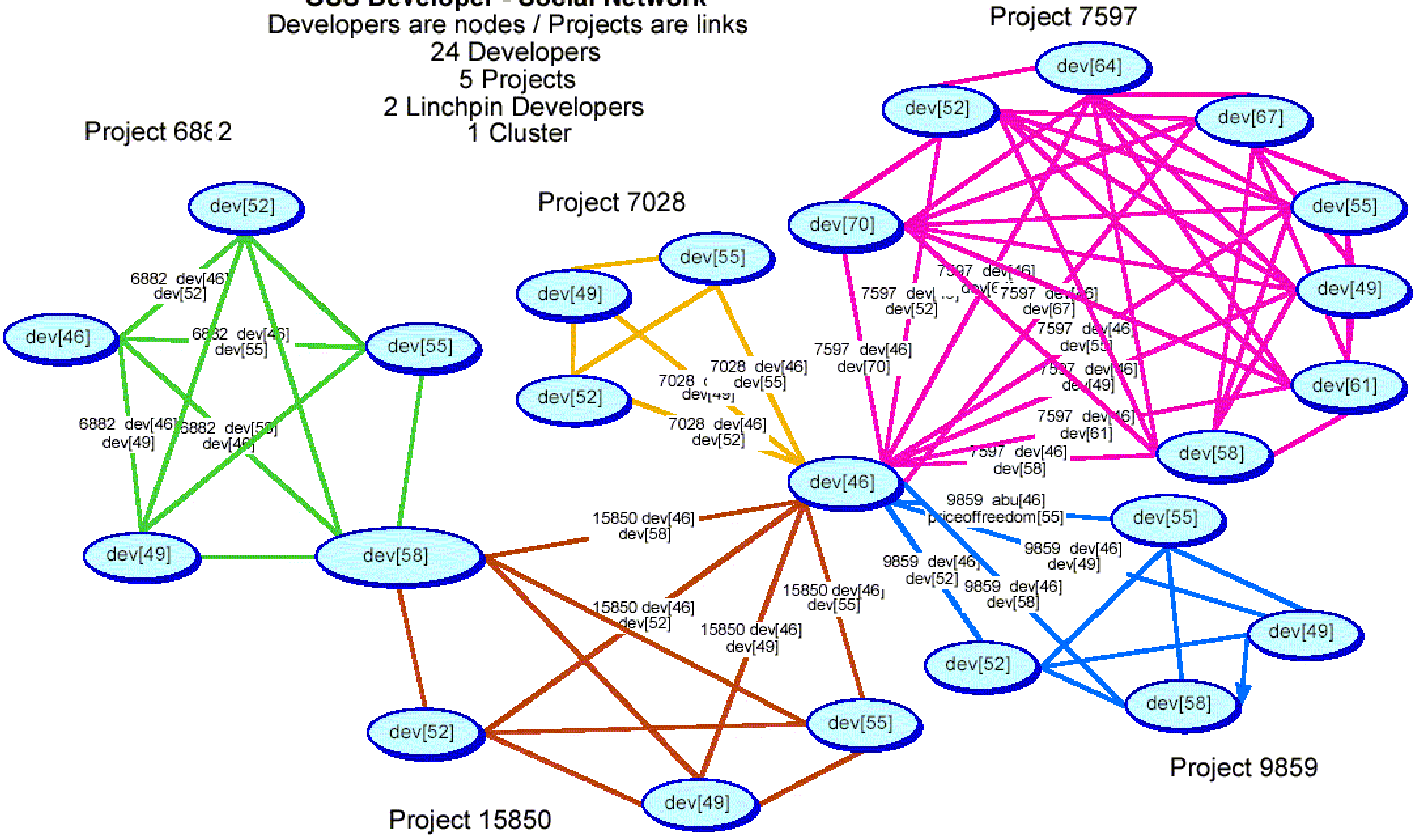
To be a leader you must pass the approval of the director. Before that you will be considered a W.T.B. (Want To Be) Leader and only after proving that you have the right skills and dedication to the project you will officially become a leader.

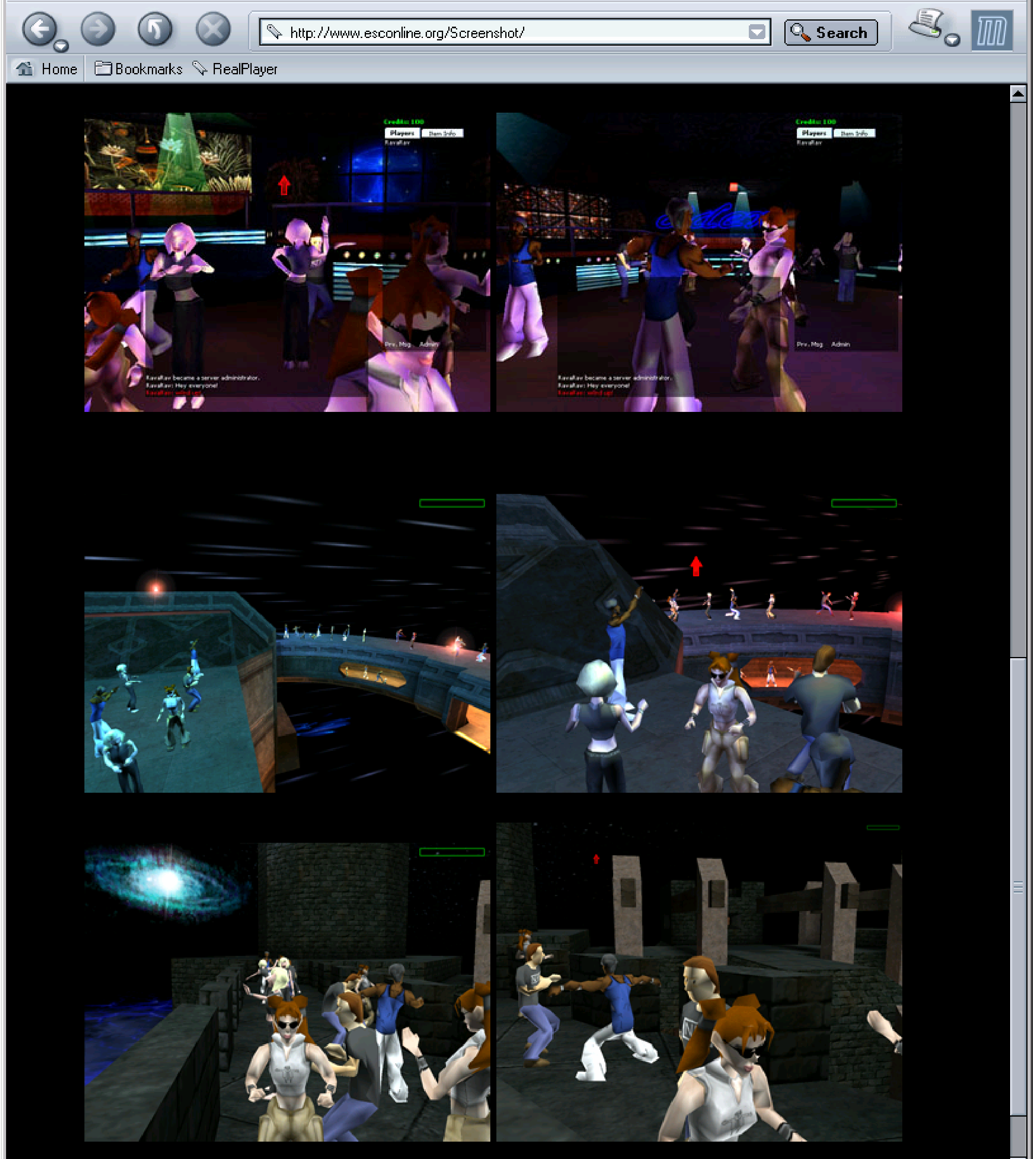
There's one leader for each department and he can have also one co-leader helping

Software technology transfer and licensing

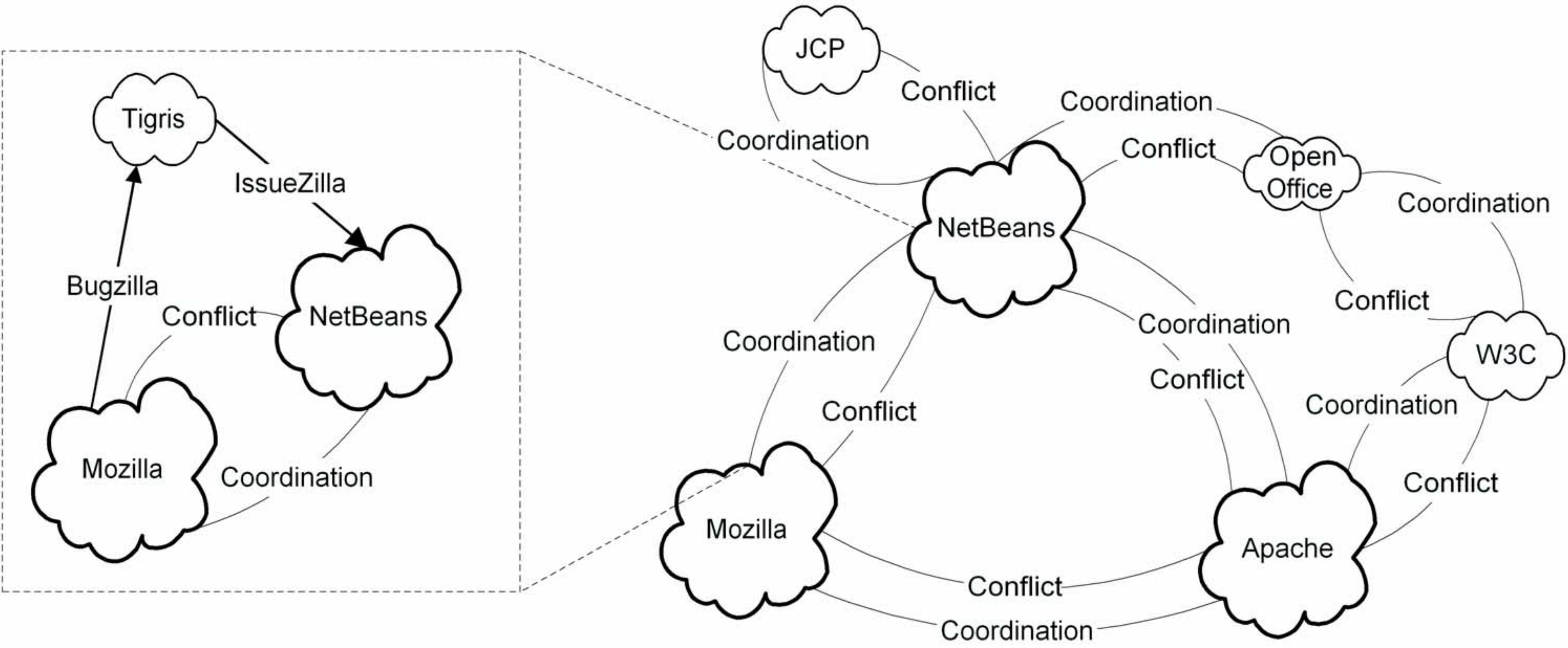
- F/OSS technology transfer from existing Web sites is a *community and team building process*
  - Not (yet) an engineering process
  - Enables unanticipated applications and uses
  - Enables F/OSSD to persist *without* centrally planned and managed corporate software development centers

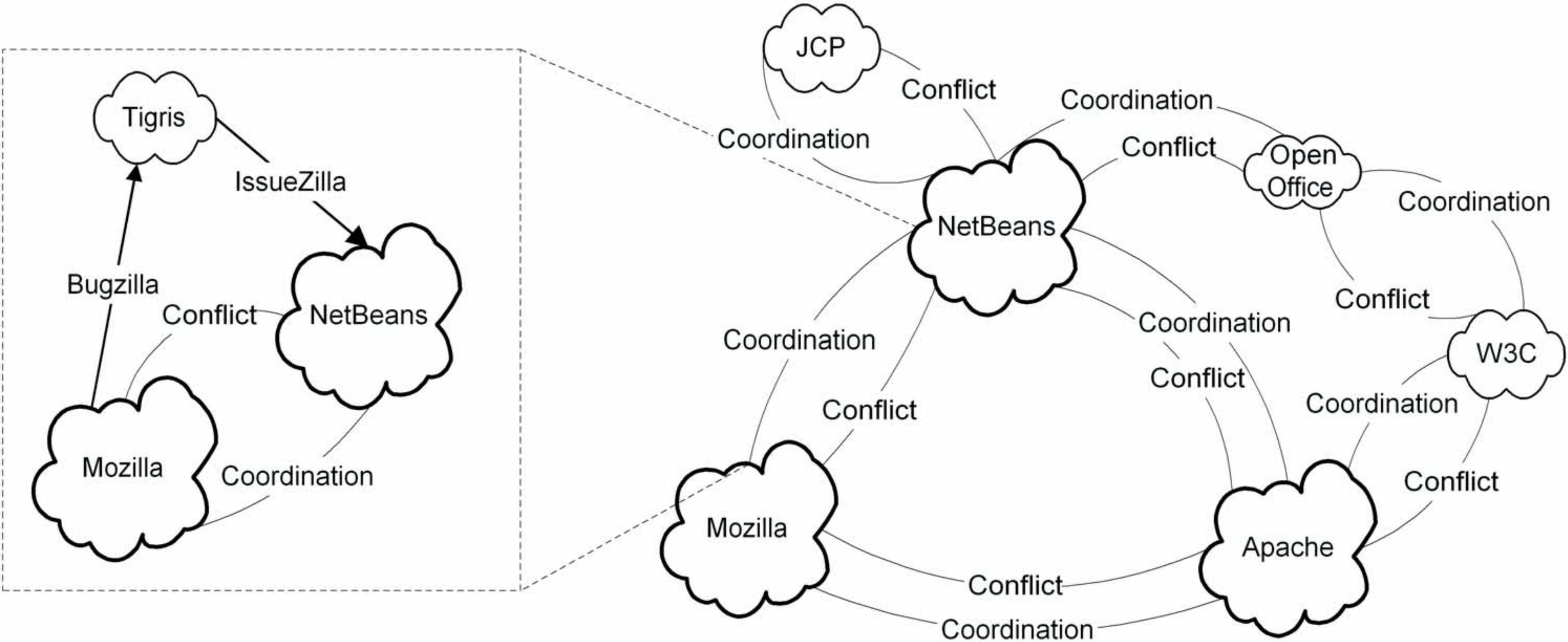
OSS Developer - Social Network
Developers are nodes / Projects are links
24 Developers
5 Projects
2 Linchpin Developers
1 Cluster





Kavallav became a server administrator.
Kavallav: Hey everyone!
Kavallav: wind up!

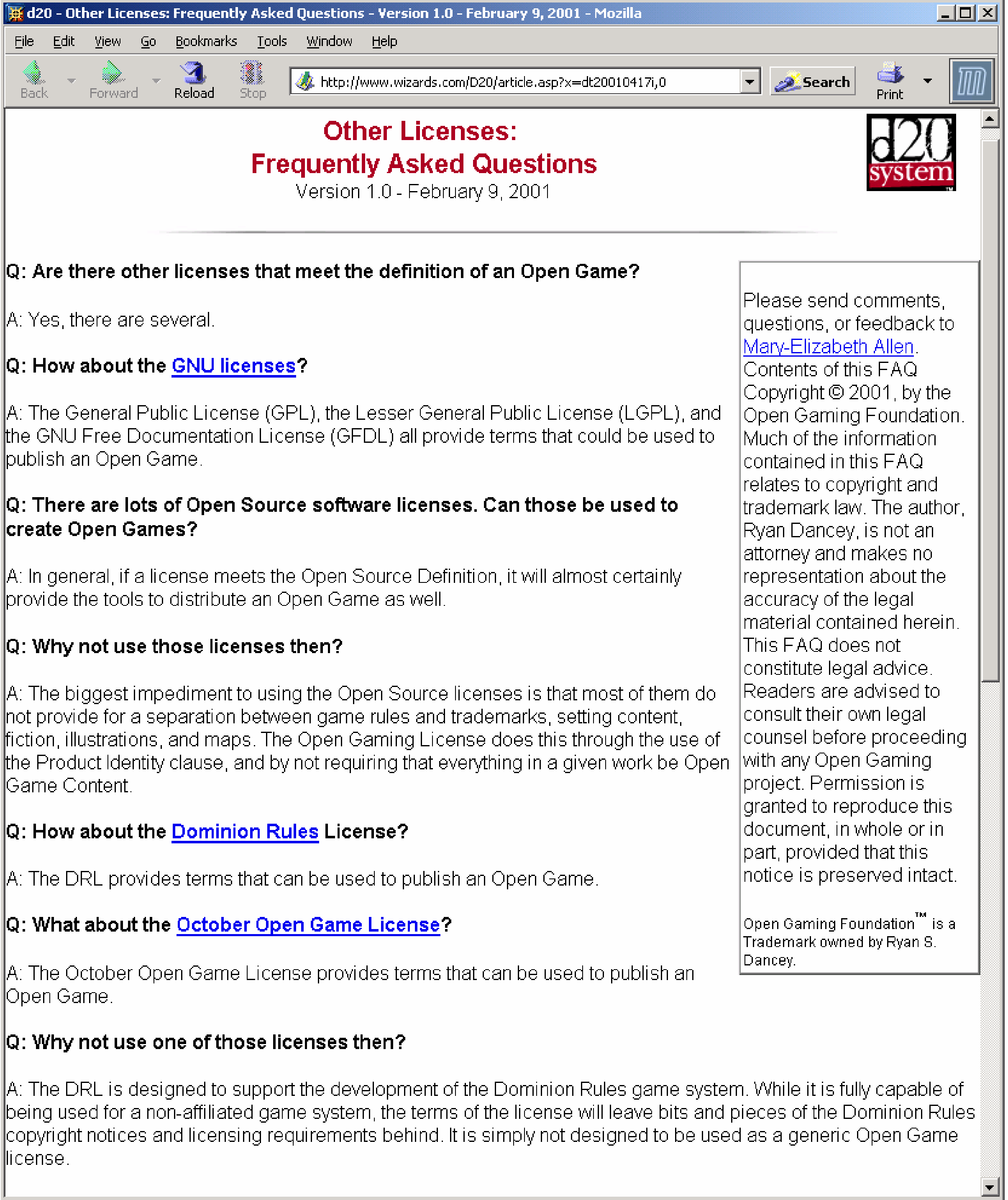

Tigris
IssueZilla
Bugzilla
Conflict
NetBeans
Mozilla
Coordination
JCP
Conflict
Coordination
Coordination
Conflict
Open Office
NetBeans
Coordination
Coordination
Coordination
Conflict
Conflict
Conflict
W3C
Coordination
Conflict
Mozilla
Apache
Conflict
Coordination



An *OSS Ecosystem*: a socio-technical community and infrastructure for tech transfer and inter-project coordination

Free/OSS licenses

- Reiterate and institutionalize F/OSS *culture* (values, norms, and beliefs)
  - – GNU Public License (GPL) for *free* software
  - – More than 35 other open source licenses (http://opensource.org)
  - – "Creative Commons" Project at Stanford Law School developing public license framework
- F/OSS culture affects technical choices for tool selection, integration, and interoperability options



d20 - Other Licenses: Frequently Asked Questions - Version 1.0 - February 9, 2001 - Mozilla

http://www.wizards.com/D20/article.asp?x=dt20010417i,0

Other Licenses: Frequently Asked Questions

Version 1.0 - February 9, 2001

Q: Are there other licenses that meet the definition of an Open Game?

A: Yes, there are several.

Q: How about the GNU licenses?

A: The General Public License (GPL), the Lesser General Public License (LGPL), and the GNU Free Documentation License (GFDL) all provide terms that could be used to publish an Open Game.

Q: There are lots of Open Source software licenses. Can those be used to create Open Games?

A: In general, if a license meets the Open Source Definition, it will almost certainly provide the tools to distribute an Open Game as well.

Q: Why not use those licenses then?

A: The biggest impediment to using the Open Source licenses is that most of them do not provide for a separation between game rules and trademarks, setting content, fiction, illustrations, and maps. The Open Gaming License does this through the use of the Product Identity clause, and by not requiring that everything in a given work be Open Game Content.

Q: How about the Dominion Rules License?

A: The DRL provides terms that can be used to publish an Open Game.

Q: What about the October Open Game License?

A: The October Open Game License provides terms that can be used to publish an Open Game.

Q: Why not use one of those licenses then?

A: The DRL is designed to support the development of the Dominion Rules game system. While it is fully capable of being used for a non-affiliated game system, the terms of the license will leave bits and pieces of the Dominion Rules copyright notices and licensing requirements behind. It is simply not designed to be used as a generic Open Game license.

Please send comments, questions, or feedback to Mary-Elizabeth Allen. Contents of this FAQ Copyright © 2001, by the Open Gaming Foundation. Much of the information contained in this FAQ relates to copyright and trademark law. The author, Ryan Dancey, is not an attorney and makes no representation about the accuracy of the legal material contained herein. This FAQ does not constitute legal advice. Readers are advised to consult their own legal counsel before proceeding with any Open Gaming project. Permission is granted to reproduce this document, in whole or in part, provided that this notice is preserved intact.

Open Gaming Foundation™ is a Trademark owned by Ryan S. Dancey.

Implications

- F/OSSD is a *community building process*
  - not just a technical development process
  - F/OSS peer review creates a *community of peers*
- F/OSSD processes often iterate *daily* versus infrequent singular (milestone) Software Life Cycle Engineering events
  - F/OSSD: frequent, rapid cycle time (easier to improve) *vs.*
  - SLC: infrequent, slow cycle time (harder to improve)

Process Modeling and Simulation Implications

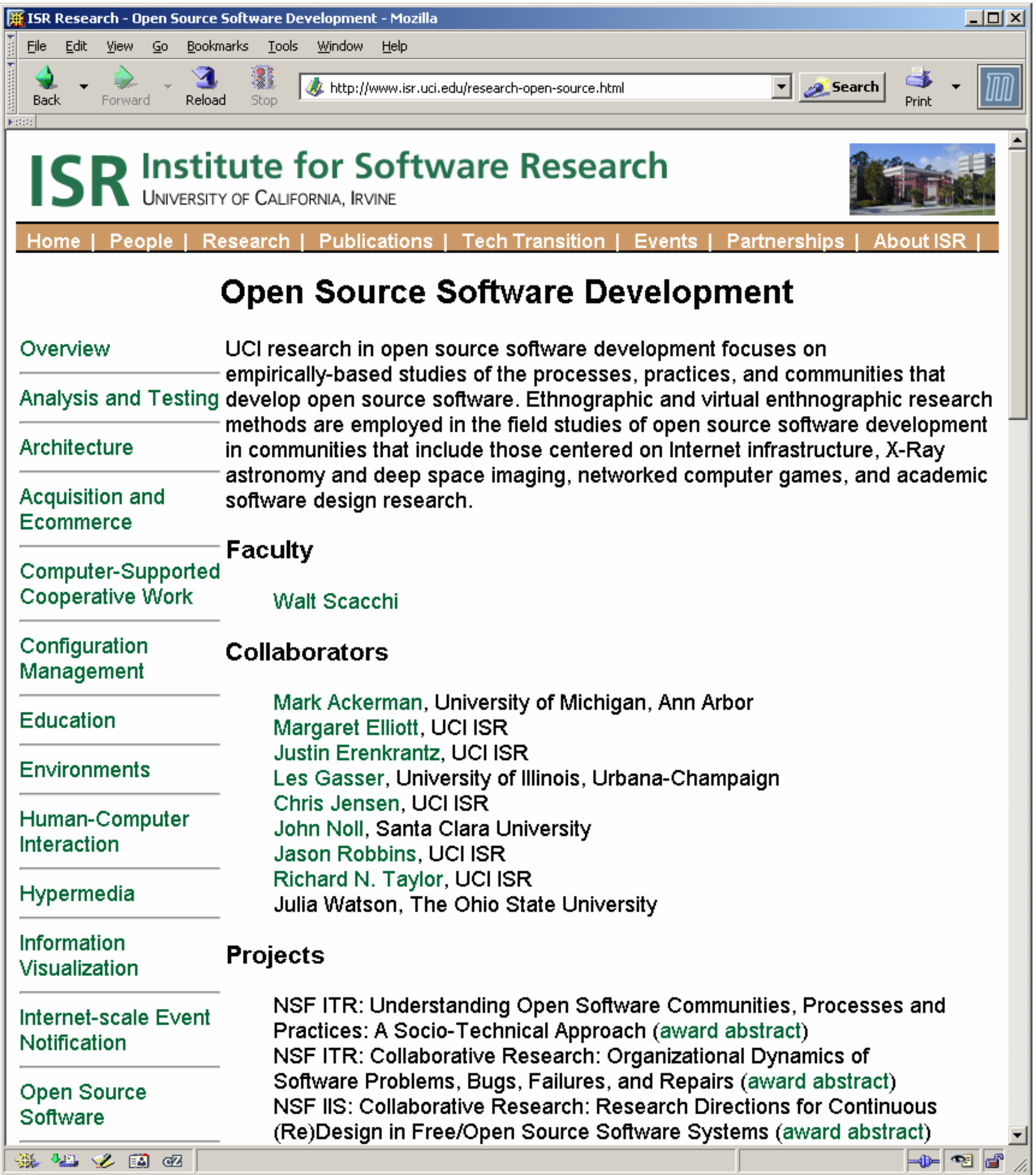
- F/OSS provides new *types* and new *kinds* of processes to model and simulate
- F/OSS process modeling and simulation requires empirical approaches
- New challenges in modeling and simulating what affects the productivity, quality, and cost of F/OSS development
- F/OSS process models and simulation should also be F/OSS

Process Modeling and Simulation Implications

- Understanding and modeling software processes in large F/OSS projects
  - may require automated process discovery techniques
  - spanning software ecosystem
- F/OSS processes (still) need to be modeled as narrative, hypermedia, and computational models.
- Modeling large, aggregated F/OSS projects likely to require advances in software process modeling tools and techniques.

Conclusions

- F/OSS processes are *different* than traditional software engineering processes
  - – not better, not worse, but different and new
  - – more social, more accessible, more convivial
- F/OSS systems need, but may not seek, the benefits from classic software engineering.



ISR Research - Open Source Software Development - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://www.isr.uci.edu/research-open-source.html
Search
Print
ISR Institute for Software Research
University of California, Irvine
Home | People | Research | Publications | Tech Transition | Events | Partnerships | About ISR |

Open Source Software Development

Overview
Analysis and Testing
Architecture
Acquisition and Ecommerce
Computer-Supported Cooperative Work
Configuration Management
Education
Environments
Human-Computer Interaction
Hypermedia
Information Visualization
Internet-scale Event Notification
Open Source Software

UCI research in open source software development focuses on empirically-based studies of the processes, practices, and communities that develop open source software. Ethnographic and virtual enthnographic research methods are employed in the field studies of open source software development in communities that include those centered on Internet infrastructure, X-Ray astronomy and deep space imaging, networked computer games, and academic software design research.

Faculty

Walt Scacchi

Collaborators

Mark Ackerman, University of Michigan, Ann Arbor
Margaret Elliott, UCI ISR
Justin Erenkrantz, UCI ISR
Les Gasser, University of Illinois, Urbana-Champaign
Chris Jensen, UCI ISR
John Noll, Santa Clara University
Jason Robbins, UCI ISR
Richard N. Taylor, UCI ISR
Julia Watson, The Ohio State University

Projects

NSF ITR: Understanding Open Software Communities, Processes and Practices: A Socio-Technical Approach (award abstract)
NSF ITR: Collaborative Research: Organizational Dynamics of Software Problems, Bugs, Failures, and Repairs (award abstract)
NSF IIS: Collaborative Research: Research Directions for Continuous (Re)Design in Free/Open Source Software Systems (award abstract)

Acknowledgements

- *Project collaborators*:
  - – Mark Ackerman, UMichigan, Ann Arbor
  - – Les Gasser, UIllinois, Urbana-Champaign
  - – John Noll, Santa Clara University
  - – Margaret Ellliot, Chris Jensen, UCI-ISR
  - – Julia Watson, The Ohio State University
- *Funding support*:
  - – National Science Foundation, ITR#0083075, ITR#0205679, ITR#0205724, and ITR#0350754.
  - – No endorsement implied.

References

- Elliott, M. and Scacchi, W., Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture, in S. Koch (ed.), *Free/Open Source Software Development*, Idea Publishing, to appear, 2004.
- A. Hars and S. Ou, Working for free? Motivations for participating in open source projects, *International Journal of Electronic Commerce*, 6(3), Spring 2002.
- G. Madey, V. Freeh, and R. Tynan, Modeling the F/OSS Community: A Quantitative Investigation, in *Free/Open Source Software Development*, in Stephan Koch, (ed.) Idea Publishing, forthcoming.
- D.M. Nichols & M.B. Twidale, The Usability of Open Source Software, *First Monday*, 8(1), January 2003.

References

see http://www.isr.uci.edu/research-open-source.html

- C. Jensen and W. Scacchi, Discovering, Modeling, and Reenacting Open Source Software Development Processes, Institute for Software Research, March 2004.
- C. Jensen and W. Scacchi, Process Modeling the Web Information Infrastructure, *Proc. 5th. Software Process Simulation and Modeling Workshop*, Edinburgh, Scotland, May 2004.
- W. Scacchi, Understanding the Requirements for Developing Open Source Software, *IEE Proceedings--Software*, 149(1), 24-39, 2002.
- W. Scacchi, When is Free/Open Source Software Development *Faster, Better*, and *Cheaper* than Software Engineering? Working Paper, Institute for Software Research, UC Irvine, April 2003.
- W. Scacchi, Free/Open Source Software Development Practices in the Computer Game Community, *IEEE Software*, Special Issue on Open Source Software, 21(1), 59-67, January-February 2004.

- This presentation will be found at: http://www.ics.uci.edu/~wscacchi/Presentations/ProSim04/