Modeling and Simulating Free/Open Source Software Processes

Walt Scacchi
Institute for Software Research
Donald Bren School of Information and Computer Sciences
Computer Game Culture and Technology Laboratory
Center for Research in Information Technology and Organization
University of California, Irvine
Irvine, CA 92697-3425 USA
http://www.ics.uci.edu/~wscacchi

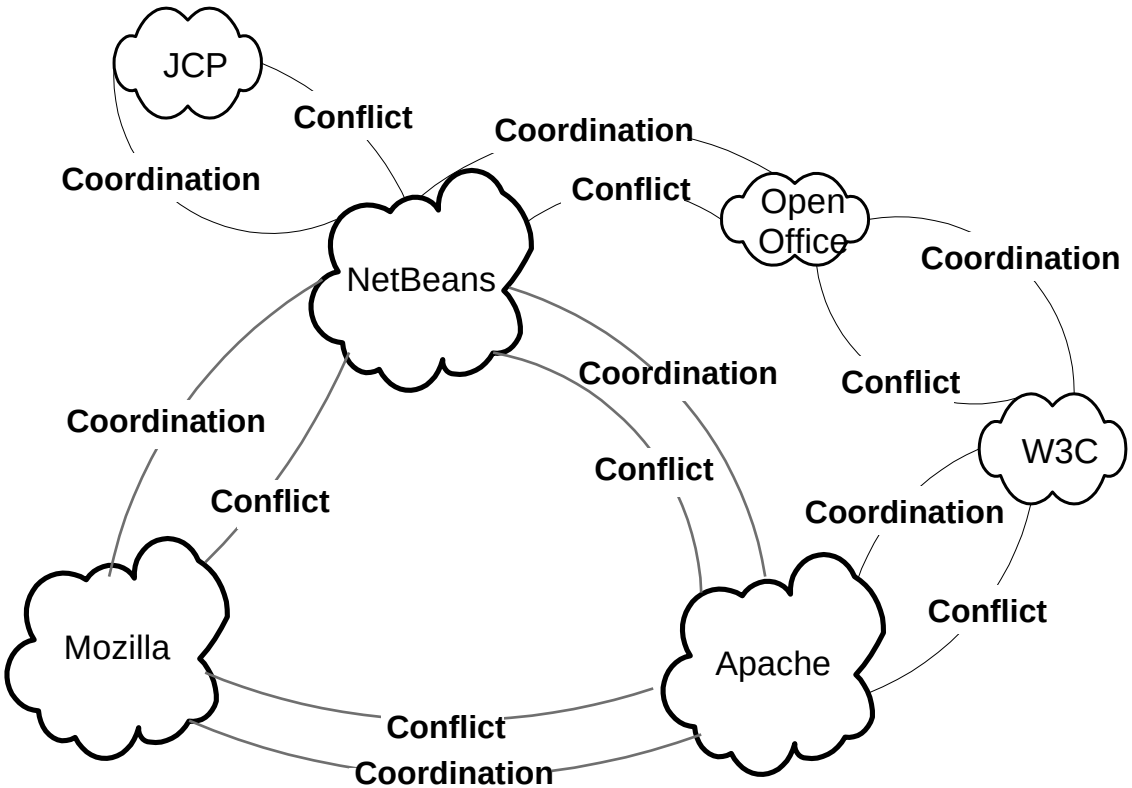
Institute for

UNIVERSITY OF CALIFORNIA, IRVI

Overview

- Monolithic vs. (dis)aggregated OSS projects
- Software process engineering techniques
- Process modeling and discovery in F/OSS
- Issues, results, experiences in F/OSS process modeling and simulation

NetBeans.org Software Project Ecosystem



(Boundary) Objects of Interaction

- Development artifacts (“software informalisms”)
- Protocols
  - HTTP, RPCs
- Shared data formats
  - HTML, XML, CGI
- Community infrastructure tools
  - Defect repositories (e.g. Bugzilla), Collaborative development tools (e.g. WIKI, CVS, mail list managers, PostNuke, SourceCast)
- Product infrastructure
  - Plug-ins, Modules
- OSS development processes

Software/Organization Process Engineering Techniques

Meta-modeling	Visualization	Instantiation and enactment
Modeling	Prototyping and walkthrough	Monitoring and measurement
Analysis	Change mgmt.	History capture and replay
Simulation	Integration	Breakdown Repair and Improvement
Redesign	Environment generation	Evolution and asset mgmt.

F/OSS Process Modeling

- Current solutions
  - *Narrative* process descriptions
    - rich context, but imprecise and limited generalization
  - *Hypermedia* process descriptions/representations
    - multi-media, navigation-based (re)enactment
  - *Computational* process representations
    - formal, parseable, scalable, amenable to process engineering

F/OSS Process Modeling

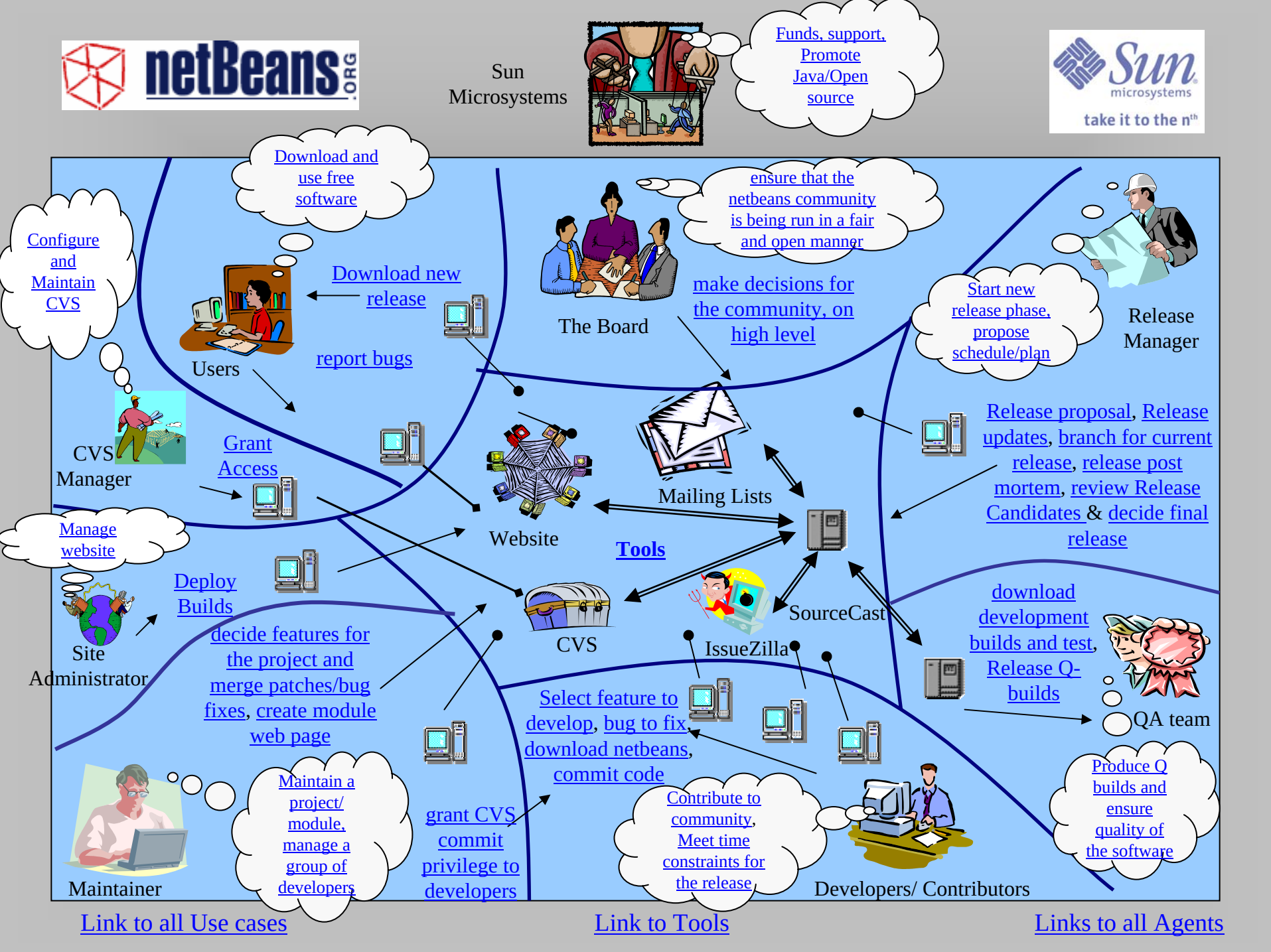
- Process knowledge discovery via *virtual ethnography* methods
  - Participant (local/remote) observation
    - Elicitation of situated accounts and sense-making
    - Gathering and jointly creating artifacts
    - Coding and iterative participant validation
  - Automated remote observation
    - Web/text data mining
      - Software informalisms on Web site
      - F/OSS project repositories (e.g., Bugzilla database, CVS)

F/OSS Process Modeling

- Representation and multi-mode modeling
- Analysis
  - inspection, measurement, re-enactment (navigational walkthrough, simulation), validation)
- Re-representation
  - visualization, briefing, publication, etc.

Case Study: *NetBeans.org*

- Developing an Integrated Development Environment for Java-based Web/enterprise applications
  - >10K contributing software developers
  - >30K contributors
  - Subsidized by SUN (vs. IBM. Microsoft)
- Software product requirements and release management process


netBeans.ORG
Sun Microsystems
Funds, support, Promote Java/Open source
Sun microsystems
take it to the nth
Download and use free software
Configure and Maintain CVS
Download new release
report bugs
Users
ensure that the netbeans community is being run in a fair and open manner
The Board
make decisions for the community, on high level
Start new release phase, propose schedule/plan
Release Manager
CVS Manager
Grant Access
Website
Mailing Lists
Release proposal, Release updates, branch for current release, release post mortem, review Release Candidates & decide final release
Manage website
Tools
Deploy Builds
Site Administrator
decide features for the project and merge patches/bug fixes, create module web page
CVS
IssueZilla
SourceCast
download development builds and test, Release Q-builds
QA team
Select feature to develop, bug to fix, download netbeans, commit code
Maintain a project/ module, manage a group of developers
grant CVS commit privilege to developers
Contribute to community, Meet time constraints for the release
Developers/ Contributors
Produce Q builds and ensure quality of the software
Maintainer
Link to all Use cases
Link to Tools
Links to all Agents

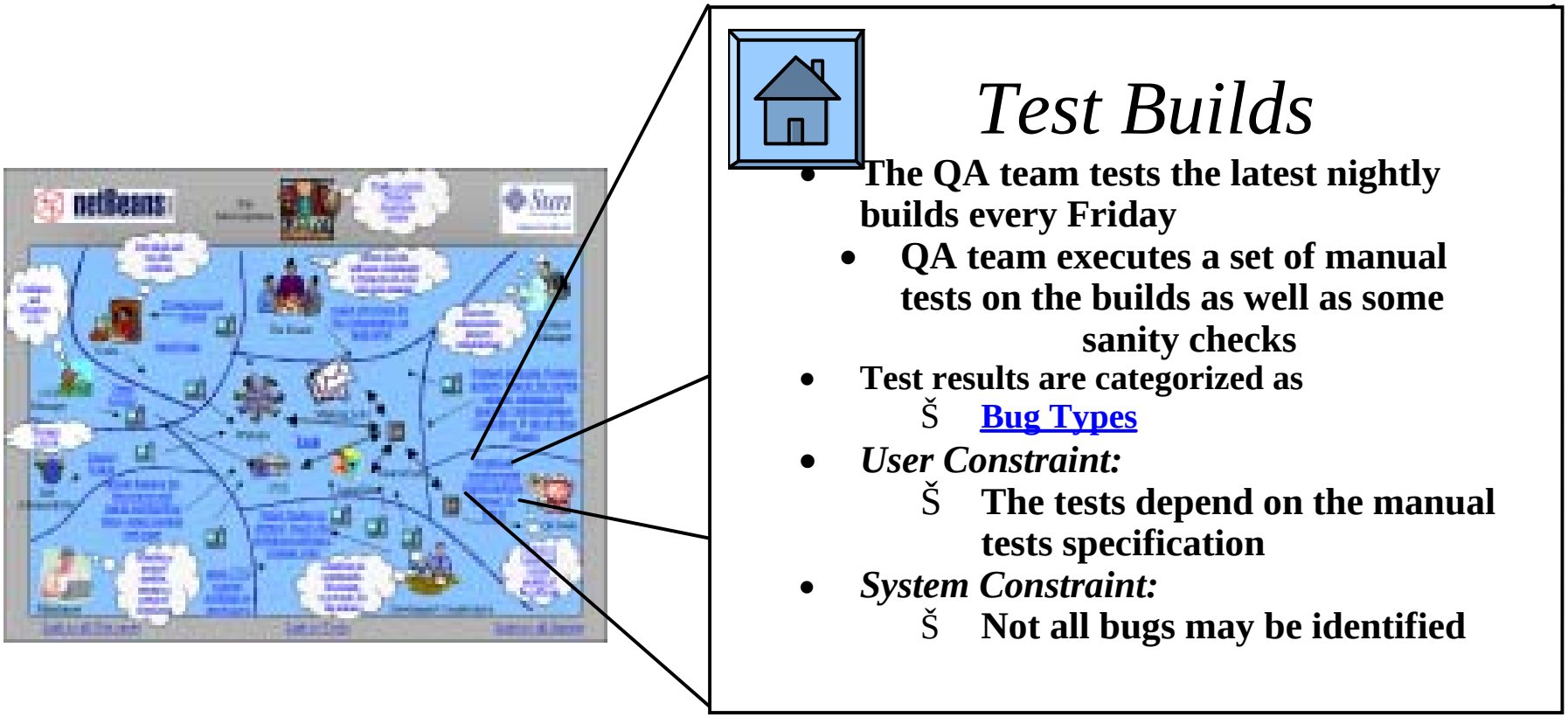
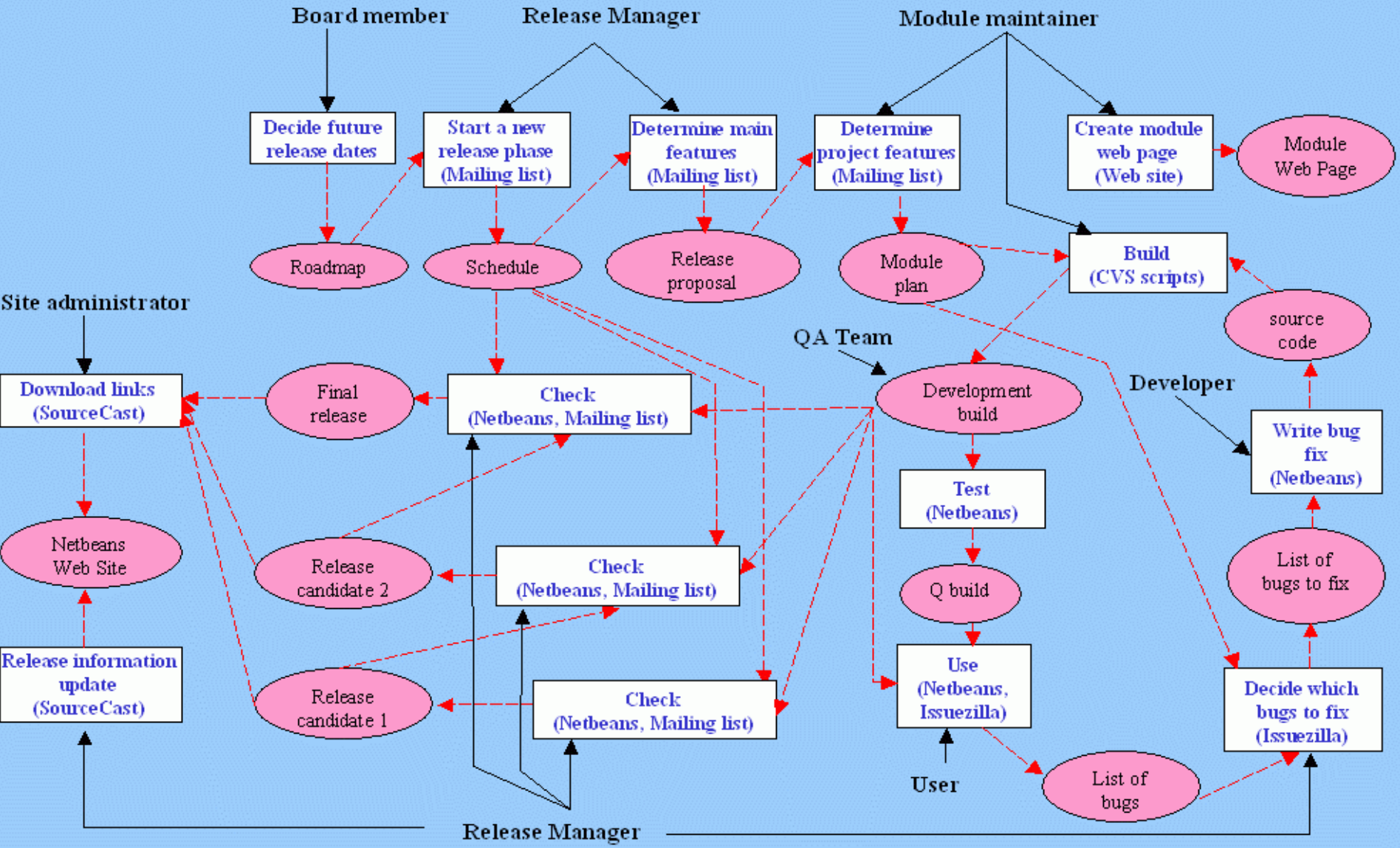
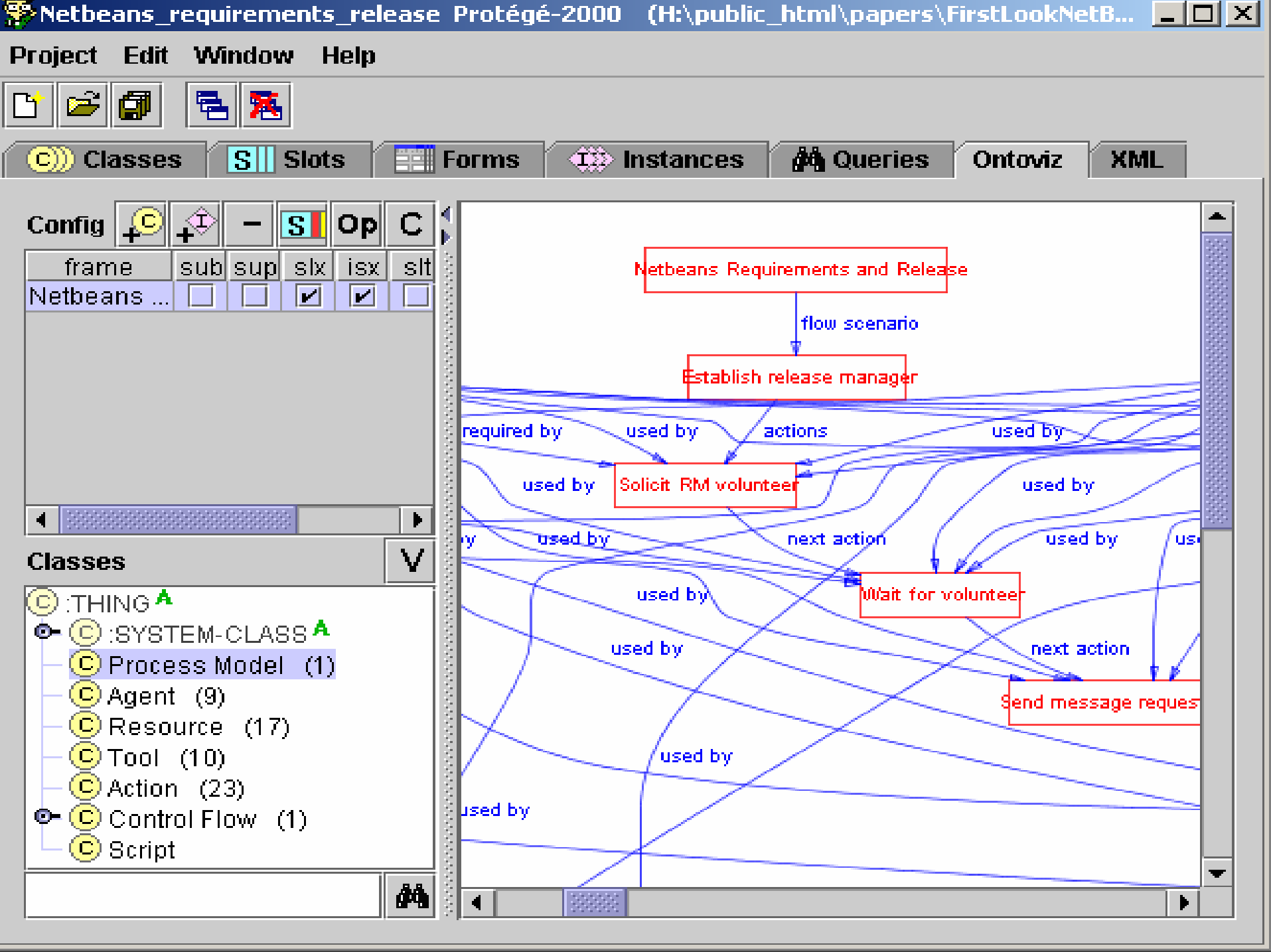


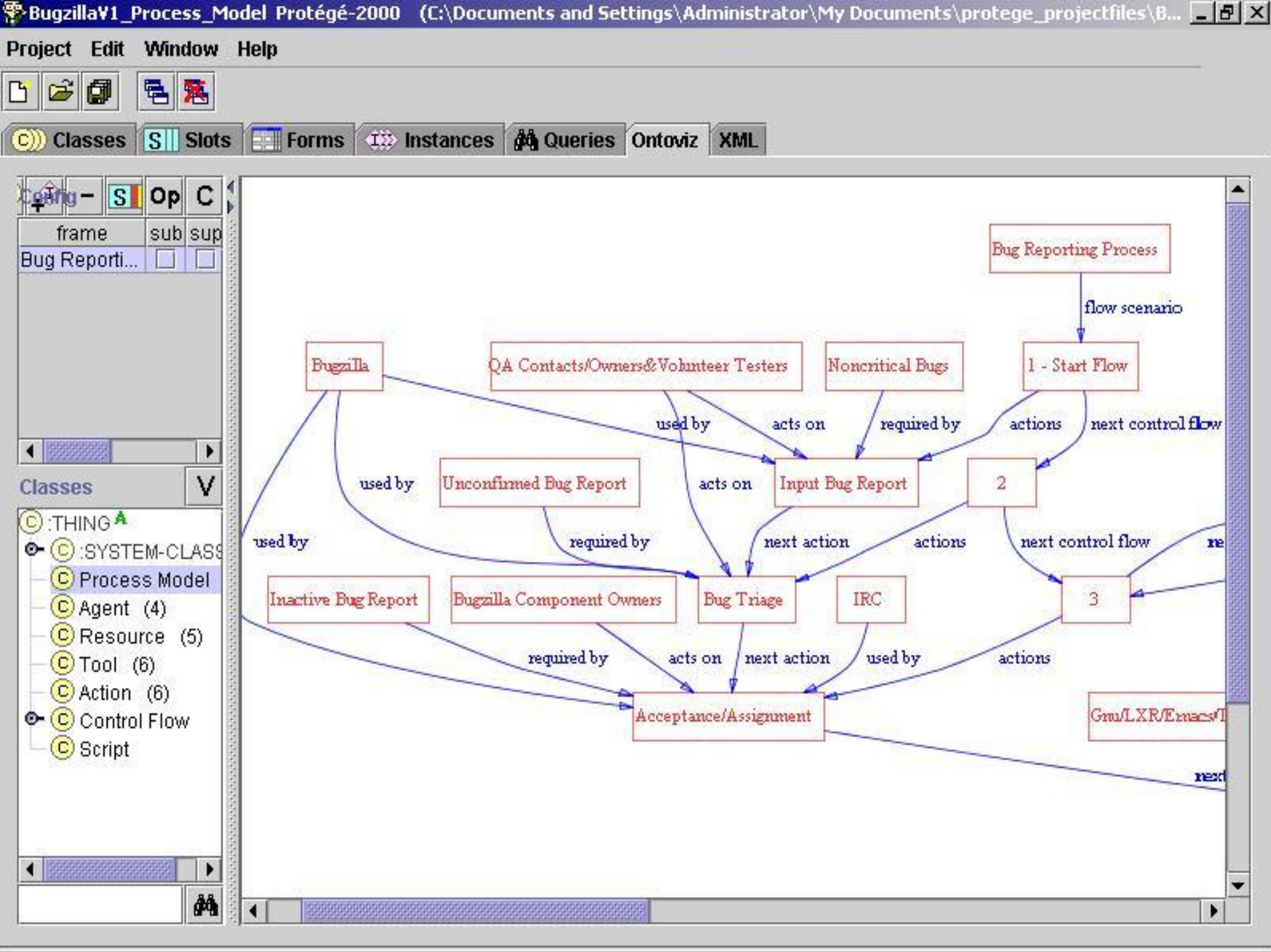
Figure 2. A hyperlink selection within a rich hypermedia presentation that reveals a corresponding use case.

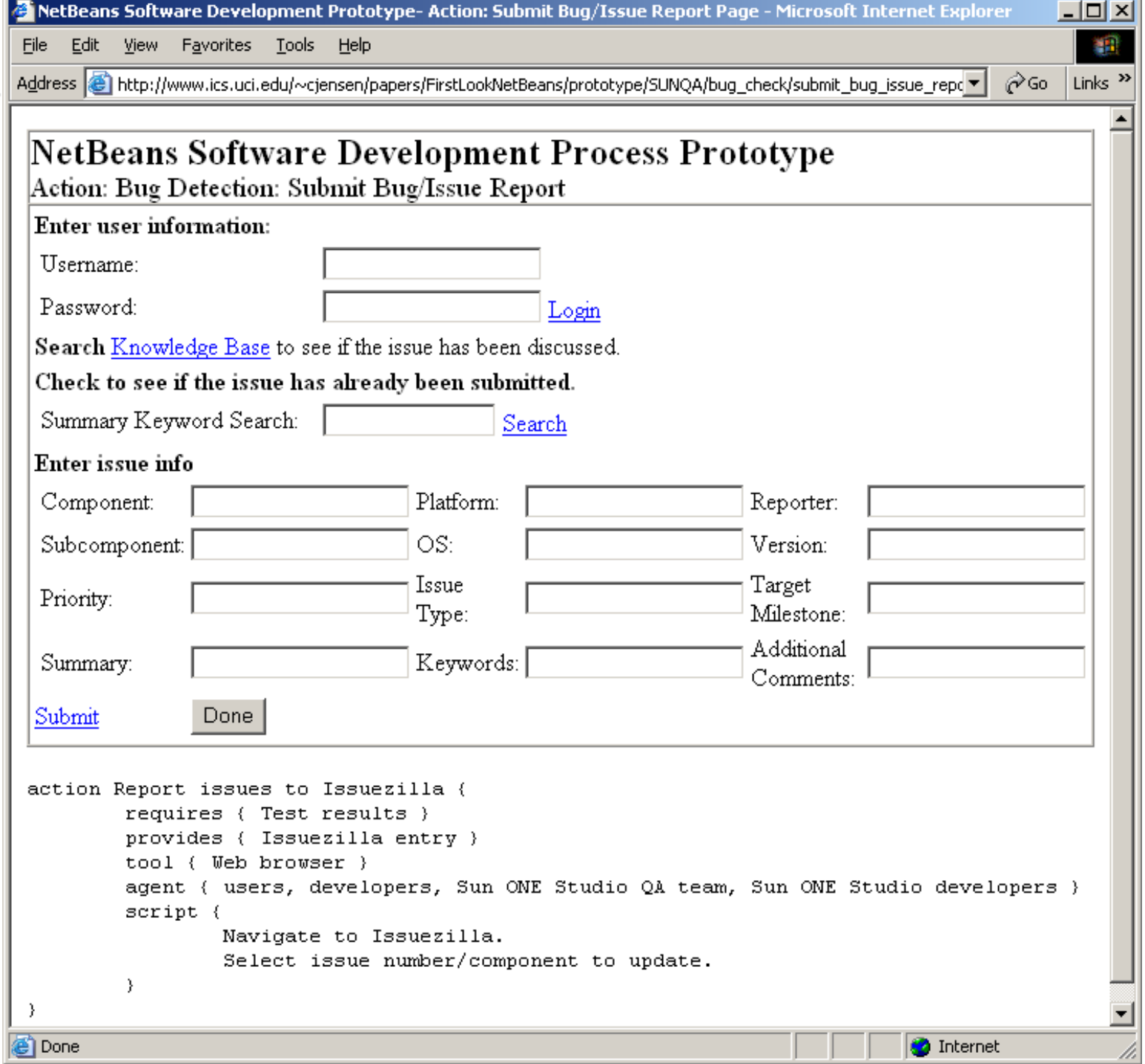

Board member
Release Manager
Module maintainer
Decide future release dates
Start a new release phase (Mailing list)
Determine main features (Mailing list)
Determine project features (Mailing list)
Create module web page (Web site)
Module Web Page
Roadmap
Schedule
Release proposal
Module plan
Build (CVS scripts)
Site administrator
QA Team
source code
Download links (SourceCast)
Final release
Check (Netbeans, Mailing list)
Development build
Developer
Write bug fix (Netbeans)
Test (Netbeans)
Netbeans Web Site
Release candidate 2
Check (Netbeans, Mailing list)
Q build
List of bugs to fix
Release information update (SourceCast)
Release candidate 1
Check (Netbeans, Mailing list)
Use (Netbeans, Issuezilla)
Decide which bugs to fix (Issuezilla)
User
List of bugs
Release Manager

Formal Process Model (excerpt)

```
sequence Test {
   action Execute automatic test scripts {
         requires { Test scripts, release binaries }
         provides { Test results }
         tool { Automated test suite (  xtest, others) }
         agent { Sun ONE Studio QA team }
         script { }
   }
   action Execute manual test scripts {
         requires { Release binaries }
         provides { Test results }
         tool {  NetBeans IDE }
         agent { users, developers, Sun ONE Studio QA team, Sun
ONE Studio                              developers }
         script { }
   }
   iteration Update  Issuezilla {
         action Report issues to  Issuezilla {
         requires { Test results }
         provides {  Issuezilla entry }
         tool { Web browser }
         agent { users, developers, Sun ONE Studio QA team, Sun
ONE Studio
         developers }
```


Netbeans_requirements_release Protégé-2000 (H:\public_html\papers\FirstLookNetB...
Project Edit Window Help
Classes Slots Forms Instances Queries Ontoviz XML
Config
frame sub sup slx isx slt
Netbeans ...
Classes
:THING
:SYSTEM-CLASS
Process Model (1)
Agent (9)
Resource (17)
Tool (10)
Action (23)
Control Flow (1)
Script
Netbeans Requirements and Release
flow scenario
Establish release manager
required by
used by
actions
used by
Solicit RM volunteer
used by
used by
next action
used by
Wait for volunteer
used by
used by
next action
Send message reques
used by
used by


BugzillaV1_Process_Model Protégé-2000 (C:\Documents and Settings\Administrator\My Documents\protege_projectfiles\B...
Project Edit Window Help
Classes Slots Forms Instances Queries Ontoviz XML
frame sub sup
Bug Reporti...
Classes
:THING
:SYSTEM-CLASS
Process Model
Agent (4)
Resource (5)
Tool (6)
Action (6)
Control Flow
Script
Bug Reporting Process
flow scenario
Bugzilla
QA Contacts/Owners&Volunteer Testers
Noncritical Bugs
1 - Start Flow
used by
acts on
required by
actions
next control flow
Unconfirmed Bug Report
Input Bug Report
2
next action
Inactive Bug Report
Bugzilla Component Owners
Bug Triage
IRC
3
Acceptance/Assignment


NetBeans Software Development Prototype- Action: Submit Bug/Issue Report Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ics.uci.edu/~cjensen/papers/FirstLookNetBeans/prototype/SUNQA/bug_check/submit_bug_issue_repo
Go Links
NetBeans Software Development Process Prototype
Action: Bug Detection: Submit Bug/Issue Report
Enter user information:
Username:
Password:
Login
Search Knowledge Base to see if the issue has been discussed.
Check to see if the issue has already been submitted.
Summary Keyword Search:
Search
Enter issue info
Component:
Platform:
Reporter:
Subcomponent:
OS:
Version:
Priority:
Issue Type:
Target Milestone:
Summary:
Keywords:
Additional Comments:
Submit
Done
action Report issues to Issuezilla {
requires { Test results }
provides { Issuezilla entry }
tool { Web browser }
agent { users, developers, Sun ONE Studio QA team, Sun ONE Studio developers }
script {
Navigate to Issuezilla.
Select issue number/component to update.
}
}
Done
Internet


NOTE PAD T201

Experiences and Lessons

- Understanding and modeling software processes in large F/OSS projects benefits from virtual ethnography and semi-automated process discovery techniques.
- 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 and interactive visual simulation tools and techniques.
  - 2D (*Sims*-like) and 3D (Visual MOO-like) computer game engines *will* be useful here.

References

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

- M. Elliott and W. Scacchi, Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture, to appear in S. Koch (ed.), *Free/Open Source Software Development*, Idea Publishing, 2004.
- C. Jensen and W. Scacchi, Discovering, Modeling, and Reenacting Open Source Software Development Processes, submitted, June 2004.
- W. Scacchi, Understanding the Requirements for Developing Open Source Software, *IEE Proceedings--Software*, 149(1), 24-39, 2002.
- W. Scacchi, Free/Open Source Software Development Practices in the Computer Game Community, *IEEE Software*, 21(1), 59-67, 2004.
- W. Scacchi, Understanding Free/Open Source Software Evolution, revised version to appear in N.H. Madhavji, M.M. Lehman, J.F. Ramil and D. Perry (eds.), *Software Evolution*, John Wiley and Sons Inc, New York, 2004.
- W. Scacchi, Socio-Technical Interaction Networks in Free/Open Source Software Development Processes, to appear in S.T. Acuña and N. Juristo (eds.), *Peopleware and the Software Process*, World Scientific Press, 2004.
- W. Scacchi and P. Mi, • Process Life Cycle Engineering: A Knowledge-Based Approach and Environment, *Intelligent Systems in Accounting, Finance, and Management*, 6(1), 83-107, 1997.

Acknowledgements

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