Multi-Modal Modeling, Analysis, and Validation of Open Source Software Development Processes

Walt Scacchi, Chris Jensen, John Noll, and Margaret Elliott

Institute for Software Research
University of California, Irvine
Irvine, CA, 92697-3425 USA

www.isr.uci.edu/research-open-source.html
Motivation

• Goal: Discover hidden processes within large-scale, global, loosely-coordinated open source software development (OSSD) projects.
  – Thousands of project participants
  – Developing, managing, and evolving over one million knowledge artifacts
  – Weakly coordinated by centralized authorities
  – All data are open source
Motivation

- Discover, model, re-enact, and repair OSSD processes
- Recognize process context, participant roles, tools, resources, interdependencies within and across projects over the Web
- Why?
  - Software development organizations and OSSD projects don’t know their processes
  - Companies and new OSSD projects want to adopt “OSSD best practices”
  - Process improvement, optimization, redesign, or transformation requires knowledge of processes
Overview

– Process discovery
– Process modeling
– Process re-enactment
– Discussion
– Conclusions
Process discovery

• Participant observation (online, Web-based ethnography)

• Collection and annotation of participant created/modified artifacts
  – Objects of interaction
  – How objects are situated in facilitating collaboration, conflict, or conflict mitigation

• Tracking artifacts added or modified in response to intra-community or inter-community dynamics

• Automated process data mining, categorization, and composition
Annotated chat transcript

• <CB> Hello (Outsider Critique-1)
• <CB> Several images on the website seem to be made with non_free Adobe software, I hope I'm wrong: it is quite shocking. Does anybody know more on the subject?
• <CB> We should avoid using non_free software at all cost, am I wrong? (Extreme belief in free software (BIFS)-1)
• <CB> Anyone awake in here? Outsider Critique-1)
Modeling OSSD Processes

- “Rich Pictures” -- overall scenarios and stakeholders
- Use cases -- hyperlinked from Rich Pictures
- Attributed flow graphs -- process control flow, data flow, role and tool bindings
- Process meta-model -- provides formal reference model and ontology
- Computational process models -- formal representations that can be executed or re-enacted
- Ethnographic hypermedia -- Web-based documents that include above representations, links to source data, and analytical narrative.
- Sun Microsystems
- Download and use free software
- Share knowledge and ensure all community issues are addressed
- Ensure that the netbeans community is being run in a fair and open manner
- Start new release phase, propose schedule/plan
- Funds, support
  Promote Java/Open source
- Make decisions for the community, on high level
- Release proposal, release updates, branch for current release, release post mortem, review release candidates (2) & decide final release
- Download development builds and test, release Q-builds
- Produce Q-builds and ensure quality of the software
- Link to all Use Cases
- Link to all Agents
- Site Administrator
- Configure and maintain CVS
- Manage website
- CVS Manager
- Users
- Community Manager
- download new release
- respond to tech issues, unanswered questions
- The Board
- make decisions for the community, on high level
- Mailing Lists
- Tools
- Website
- IssueZilla
- SourceCast
- CVS
- QA Team
- Developers/Contributors
- Maintain a project/module, manage a group of developers
- grant CVS privilege to developers
- grant access
- report bugs
- decide features for the project and merge patches/bug fixes, create module web page
- Contribute to community, meet time constraints for the release
- Significant milestones
- Link to Tools
- Site Administrator
- Site Administrator
- Link to all Agents
**Test Builds**

- The QA team tests the latest nightly builds every Friday
  - QA team executes a set of manual tests on the builds as well as some sanity checks
  - Test results are categorized as
    - **Bug Types**
    - **User Constraint:**
      - The tests depend on the manual tests specification
    - **System Constraint:**
      - Not all bugs may be identified

**Figure 2.** A hyperlink selection within a rich hypermedia presentation that reveals a corresponding use case.
Process re-enactment

• Generating executable or re-enactable process specifications derived from ontology

• “Low-fidelity” process re-enactment support
  – We don’t try to model everything
  – Focus on resource flow patterns
  – Accommodate gaps and detect inconsistencies in process enactment models

• Re-enactments are interactive, navigational, and grounded in artifacts, tools, roles, and resource dependencies resulting from discovery and modeling
Formal model of an OSSD process coded in PML (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 { /* Executed off-site */ } }
  - 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 { /* Executed off-site */ } }
- 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 }
    - script {
      <br><a href="http://www.netbeans.org/issues/">Navigate to Issuezilla</a>
      <br><a href="http://www.netbeans.org/issues/query.cgi">Query Issuezilla</a>
      <br><a href="http://www.netbeans.org/issues/enter_bug.cgi">Enter issue</a>
    }
  - ...
PML validation analysis

Summary of analysis for netbeans_req_release.pml

Model size (source lines): 307
Actions: 36
Resources: 72
Actions neither requiring nor providing resources: 1
Resources required but not provided (potential inputs): 0
Resources provided but not required (potential outputs): 0
Miracles: 2
Black holes: 6
Transformations: 30
ReportIssuesToIssuezilla

State: NONE

Required Resources: TestResults

Provided Resources: IssuezillaEntry

Script: "Navigate to Issuezilla Query Issuezilla Enter issue"
Project Issue Tracking: netbeans.org

Issue type: Component: Subcomponent:
DEFEKT "UNCATEGORIZED" "ALL"
ENHANCEMENT 3rd-party NEW
FEATURE ally
TASK accelerators
PATCH ant

Status: Resolution: Priority:
UNCONFIRMED FIXED P1
NEW INVALID P2
STARTED WONTFIX P3
REOPENED LATER P4
Discussion

• Socio-technical and cultural evolution processes
• Validation strategies and tactics
• Implications for discovering, modeling and re-enacting OSSD processes
Socio-technical and cultural evolution processes

• New processes under study
  – Joining and contributing to a project in progress
  – Role-task migration: from project periphery to center
  – Alliance formation and community development

• Independent and autonomous project communities can interlink via social networks that manipulate objects of interaction
  – Enables possible exponential growth of interacting and interdependent community as socio-technical interaction network
Validation strategies and tactics

• Multi-mode modeling
  – Collection and annotation of artifacts
  – Rich pictures with hyperlinked Use Case scenarios
  – Directed and attributed resource flow graph
  – Process domain ontology construction

• Simulated process re-enactment
  – Process model language generated from ontology
  – PML compiled into re-enactment environment
  – Automated PML source validation
  – Simulated walkthrough of process

• Integration via ethnographic hypermedia
• Open to independent validation and interactive traceability
Implications for discovering, modeling and re-enacting OSSD processes

• Discovering, modeling, and understanding “hidden” software processes in large OSSD projects
  – requires semi-automated process discovery techniques
  – must span multi-project ecosystem

• Discovered processes (still) need to be modeled as narrative, hypermedia, and formal computational models.

• Understanding large, aggregated Internet-based projects requires process discovery, modeling tools, re-enactment and validation techniques.
Conclusions

• We examine open source software development processes within and across multiple projects spanning multiple loosely-coupled communities.

• OSSD process patterns are continuously emerging, but can be detected, modeled, analyzed, simulated and re-enacted.

• Multi-modal modeling techniques are needed to study complex socio-technical processes found in OSSD.

• Discovering, modeling, validating, and re-enacting hidden processes within and across multiple inter-dependent projects is challenging and important.
Acknowledgements

• Project collaborators:
  – Darren Atkinson, Santa Clara University
  – Mark Ackerman, University of Michigan, Ann Arbor
  – Les Gasser, University Illinois, Urbana-Champaign
  – and others at UCI-ISR

• Funding support (no endorsement implied):
  – National Science Foundation #0083075, #0205679, #0205724, and #0350754.