Visualization of Software and Activity

Paul Dourish
Institute for Software Research
UC Irvine
jpdp@ics.uci.edu

NASA ARC, August 2002
what we already know

• from studies of software development
  – complexity of task and interaction
  – the impact of distance
  – complexity of interdependence
• from studies of collaboration
  – formal and informal
  – the role of awareness
    • qualitative understandings of the actions of others
    • provides a context for your own actions
approach

• visual approaches
  – cognitive -> perceptual
  – focus on artifacts rather than processes

• theoretical background
  – the experience of computation
    • making computation “present”
  – embodied interaction
    • interaction as an embodied practice
    • directness and engagement

• two projects: seesoft and vavoom
seesoft

- CSCW research focus on awareness
  - passive understanding of the activity of others
- Trying to understand “awareness in the large”
  - large-scale, distributed software development
  - in large projects, the codebase is the artifact
  - awareness of changes in the artifact
  - awareness of the actions of others
- seesoft
  - adopted from work of Eick, Wills, et al.
  - view the activity in CVS repositories
seesoft
seesoft
seesoft
• interface abstractions hide mechanism
  – “folders” are folders whether local or networked
• but mechanism is how we understand the world
  – understandings of cause and effect
  – temporal dynamic coupling
  – the temporal organization of social action
• manifesting computation
  – how to give people a picture of what’s happening?
vavoom

- initial exploration: vavoom
  - focus on novice programmers
    - we understand the models in terms of which to explain
    - vested interest in finding out what’s going on
    - there’s plenty of them lying around
  - vavoom is the visual virtual machine
    - visualize java program execution
    - dynamic, real-time visualizations
    - unmodified class files
vavoom
vavoom
vavoom
vavoom
• vavoom is an early technical exploration
  – the focus on programmers is a hack
    • but convenient... as long as they have the right programs
  – a more interesting strategy is to:
    • focus on end users
    • focus on more abstract visualisations
    • focus on more specialised tasks
      – currently exploring network security
conclusions

• system support for informal interaction
  – non process-centric collaboration
  – qualitative understandings of
    • software behavior
    • collaborative activity

• exploiting software as artifact
  – software as primary coordinative resource
  – directness and malleability

• open questions
  – integration with practice
  – balance between translucence and distraction
  – demonstration of theoretical approach