Sharing is fun!
Thoughts on open access research

Mike Godfrey
University of Waterloo

Why share your research artifacts?

• So the community can validate your results
• So the community can build on your results
• It improves your visibility in the research community

Two experiences

• GXL
  – Ric Holt, Andy Schürr, Susan Sim, Andreas Winter, many others
  – c. 1999

• What's in a name?
  – Abram Hindle, Neil Ernst
  – c. 2011

Project 1: GXL

• Background: late 1990s, lots of research source code reverse engineering environments emerging:
  e.g., Rigi/Shrimp, SwagKit/PBS, Bauhaus, MOOSE, GuPro, Datrix, Dali, CIA/Acacia ...

Project 1: GXL

• Background: late 1990s, lots of research source code reverse engineering environments emerging:
  e.g., Rigi/Shrimp, SwagKit/PBS, Bauhaus, MOOSE, GuPro, Datrix, Dali, CIA/Acacia ...
Architectural Reconstruction

System Artifacts  Extractors  Repository  View Generation

Source Code  Scanning  Visualized Manipulation

Executing System  Parsing

Source Control  Profiling

Change Reporting  Extracted Facts

Transforming Between Schemas

Universal

High-Level

Procedural

Object-Oriented

PL/I

C

C++

Java

Dali C

PBS C

Rigi C

Let's share our tools!

• "I want to use your source code fact extractor with my analysis engine ... how hard can that be?"
  – "Just make your tools available for download!"
  – "Just make your APIs and output data format public!"
  – "Just make your source code available!"
  – "Just show me your main internal meta-model!"
  – "OK, maybe we need to talk ..."
GUPRO Multi-Language Model

Let's share our tools!

• Key events
  – CASCON 1999 / 2000 workshops on tool interoperability
  – ICSE 2000 Workshop of Std Exchange Formats (WoSEF)
  – Dagstuhl Seminar 01401, Interoperability of Reverse Engineering Tools, Jan 2001

• Months of discussion + arguing led to three levels:
  1. Software architecture
  2. "Middle model" (f calls g, h uses v)
  3. "Code level"
  PLUS a XML-based notation that can be used to represent all three: GXL

Some problems

• To pre-process or not?
• Templates/generics are a bear
• Are interfaces classes? It's important ...
• Naming, UIDs, mangling
• Lies my extractor told me

GXL

• After arguing and arguing, we realized that all we could really agree on as a community is that models of programs are graphs

• GXL: Graph eXchange Language
  – It's XML!
  – It's not XMI!
  – ... but, ummm, BYO schema!
Success! ...

• Some tool owners created GXL converters for their tools, but its use fizzles out

• It's a headache to maintain inter-tool compatibility when you're doing active research and keep changing your mind (and others do the same)
  – That's the nature of research!
  – Probably this works best with "stable" tools

Success! ...

• So this "sharing" turned out to be a lot harder than it looked, even with a lot of good will and energy

• Instead of building large, robust bridges, we built a raft factory
  – And that was good enough for its purpose

• Most importantly, we learned a lot about what to do "next time"

Project 2: What's in a name?

• Can we label/name topics automatically extracted from version control meta-data?

• For a given LDA topic, can we label it with non-functional requirements (NFRs) automatically + without training?

• ... semi-automatically?

What's in a name

- Abram Hindle is a big advocate of open access, wants to set a good example:

  http://softwareprocess.es/static/What’s_in_a_Name.html

  - Source code for original tools
  - Original data (MaxDB repo: 1GB)
  - Extracted data
  - Tool output (LDA topics)
  - VirtualBox VM (LDA, other tools + data preloaded: 3GB)
  - Wordnet-like list for NFRs (Please reference if you use it!)

Lessons learned and open sores

- It is our moral duty as scientists to be open

- Assume no one will care, but someone might

- Sharing is hard!
  - You can "design for sharing", but it takes effort
  - You will get better at it

Sharing is fun!
Thoughts on open access research

Mike Godfrey
University of Waterloo