Building Wisdom from the Past

System Understanding and Change Patterns

A Chinese proverb...

It’s a detriment to yourself if you don’t listen to the elders’ sayings.

Does this apply to system understanding?
Outline

• wisdom mined from repository data for system understanding
  – papers from MSR 2004

• a more detailed example
  – our work on mining change patterns on source code
    (joint with Murphy, Ng, Chu-Carroll)

• challenges

Building maintenance strategies

• [Van Rysselberghe and Demeyer]
  – an approach to find FAC – “similar” changes that have frequently been changed
  – ideas on how to use FAC to build up maintenance strategies
Analyzing usage data

- [El-Ramly and Stroulia]
  - an approach interaction patterns by sequential data mining usage data
  - applied the approach to UI re-engineering and URL recommendation

Visualizing Repository data

[Godfrey, Dong, Kapser, and Zou]

Tracing code evolution

version y

[version x]
Identifying relevant code

- [Shirbad, Lethbridge and Matwin]
  - an approach of classification on attributes based on the text and the syntactic structure of code in CVS
  - result shows that classification on text-based combined with syntactic-based attributes give good results
- our work [Ying, Murphy, Ng, and Chu-Carroll]
Our approach

Our approach involves several steps:

1. Data preprocessing
2. Data mining
3. Query for recommendations

The data preprocessing step involves using data mining tools to analyze version information, patterns in file versions, and recommended files.

An “interesting” scenario

Bugzilla Bug 150339: huge font crashes X Windows

web page with font size > 10,000 pixels

Step 2. enter into address bar: http://www.adeliesolutions.com/Projects

Actual Results: X windows crashes

Created an attachment (id=87422) gtk/nsFontMetricsGTK.cpp

This patch limits the size of fonts to twice the display height.

The patch misses the Xlib gfx version should also change xlib/nsFontMetricsXlib.cpp

© 2004 IBM Corporation
Challenges: MSR on system understanding

Challenge 1: Wisdom mining

- Detailed vs. abstract info
- Out-dated data
- Information explosion
Challenge 2: "Interesting" wisdom presentation

better UI integration

"interesting"?

Challenge 3: Privacy and security

user information exploited by hackers?

well-informed about data collection?
Summary

• Many kinds of wisdom from MSR for system understanding, as seen in the MSR papers and previous work.
  • e.g., our work uses source code change patterns and we evaluated on “interesting” source code recommendations.
• Challenges
  – determining the applicability and the level of abstraction from voluminous data
  – presenting “interesting” wisdom
  – privacy and security