

Understanding and Debugging System Configuration

Earl Wagner

Ewagner@cs.northwestern.edu

Henry Lieberman

Lieber@media.mit.edu

What Happens When Things Go Wrong?

- You have a system and something has gone wrong
- User should be able to have a conversation with the system to understand what's wrong
- Why can't you now?
 - Systems don't keep track of relationships between procedures and data
 - Systems also don't keep track of relationships between procedures, data and people's goals

Software Agents Can Help When Things Go Wrong

- We track the data, procedures and relationships to operators' goals
- We enable user to visualize data and hypothesize sources of problems
- We'll present a solution in the domain of e-commerce, then discuss the application to autonomic systems

Overview

- Introduction to Woodstein
- Woodstein Demo
- Application to Interfaces for Autonomic Computing
- Summary

E-Commerce is great but...

- What happens when things go wrong?
 - "Your call is important to us... Please enter your 15-digit card number, your PIN...if you're calling from a touch-tone phone, please press 6 for customer service now, ...This is customer service, can I have your card number?"
- Customers' frustration: Mistakes waste time, Vendors deflect problems and blame others
- Vendors' frustration: Customer support is expensive
- Customer support problems in both e-commerce and managing systems

Central Point to Inspect User's Account Info and Processes

- Views with info from bank and credit card
- Step through procedures that bank and credit card go through
- System identifies problems for user
- Helps user diagnose problems, then resolve, in other words debug
- Software agent acts as assistant to user

Help through Software Agents

- Agent works on user's computer
 - Shares records only with user
- Tracks user actions and web site reactions
 - Implemented as web proxy (inspired by WBI)
- Matches with library of web task models
- Provides an integrated view of processes
 - Even when they span multiple sites
 - Who else could do that for you?
- Supports debugging process

Agent Recognizes User Actions

- Proxy that analyzes user's click-stream
 - Web page controls are instrumented
 - Watches page urls, clicks, text entry
- Matches user actions to task models
 - "Purchase at Amazon.com"
- Retrieves relevant data from web pages

Woodstein Tracking Demo

Features in Tracking Demo

- Interact with page data and processes directly through inspection
- Easily access the history of data and processes and pages in which they first appeared
- Integrated view across multiple sites
 - View where a process occurred in overall action
 - Auto-generated audit trail for individual data item
- Finally, what's the alternative?

Woodstein-style Interface for Understanding Systems

- Great for web-interfaces to remote systems
 - How to use for local systems?
- Two key features
 - Provide integrated view of state across multiple sub-systems
 - Explain reasons for actions, changes
- Both features come from having models for sub-systems
 - Already necessary for “self-awareness”

Integrated View of the System

- System’s state is the result of accumulated changes over time
- Work by Paul Maglio, others finds resolving problems involves
 - Talking with other operators
 - Looking at notes, records, logs, config files
 - Querying the system
- Operators have to search for information in many places to get the “big picture” of current state

Reasons for Changes

- Operators also look for reasons for previous changes
- Problem remains how to have a conversation with a computer about its state
- Eser Kandogan: Log files, configuration files, etc. are the system's face; like collaborators they should be transparent

Computer Self-Regulation Increases Complexity

- Promise of Autonomic Computing
 - “high-level policy-based control (supervision) will replace low-level parameter tuning (configuration setting)” (CHIACS page)
- Changes by both people *and* computers
 - Even more difficult to understand the current state of the system
 - How did the system get to this state?
 - Why were these changes made?

Having Models of Processes Creates New Opportunities

- “Self-awareness” comes from knowing reasons for actions, changes
- These reasons explain how high-level policies translate to low-level configuration
- For any data item, operator will be able to see why and how it has the value it does
 - Operator will communicate with computer through these models
 - When something goes wrong, quickly localize fault

Help in Maintaining Policies

- System will have policies governing operation
- Operator sees something wrong and will “try out” new policies
 - “What new policy would we have to add to prevent this from happening?”
 - System could offer recommendations
- System support for exploring hypothetical possibilities

Even Further Out: Sharing Problem-solving Expertise

- Operators share strategies for resolving problems
- How to save strategies and suggest when appropriate?
 - Work by Lau, Bergman on saving techniques
 - Work by Doug Riecken on multiple reasoning strategies

Summary

- We want to interact with the computer at a high-level
- Woodstein uses models to enable high-level interaction with web page info
- With models of system, when problems occur, operators can:
 - Inspect and see reasons for changes
 - Try out possible solutions
 - Save and retrieve strategies

Contact

- Henry Lieberman
 - Lieber@media.mit.edu
- Earl Wagner
 - Ewagner@cs.northwestern.edu
- Woodstein
 - <http://web.media.mit.edu/~ewagner/woodstein>
 - Or search for “Woodstein” on Google