



JACOBS
UNIVERSITY



Fault Resolution in Case-Based Reasoning

Ha Manh Tran and Jürgen Schönwälder
Computer Science, Jacobs University Bremen, Germany

10th Pacific Rim International Conference on Artificial Intelligence
Hanoi, Vietnam, 15-19 December 2008

Overview

- What?
 - Quickly providing useful information for resolving problems in communication systems.



- Why?
 - Fault data is semi-structured and frequently outdated.
 - Fault data exists in indexed discussion forums, bug tracking systems, etc.

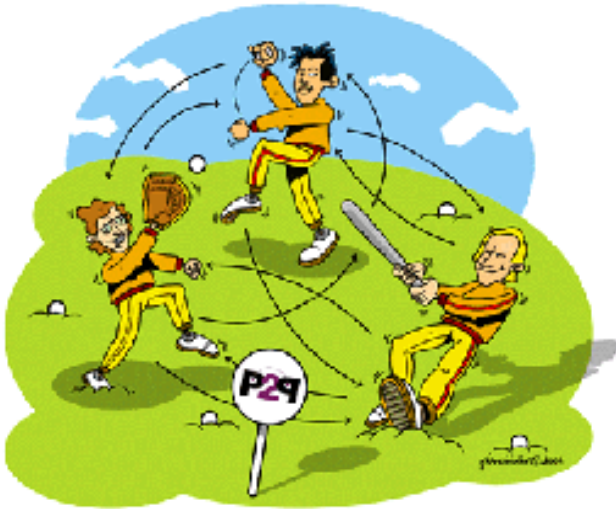
FoCUS

- Distributed case-based reasoning system (DCBR)
- Probabilistic approaches to case reuse in case-based reasoning (CBR)
- Two-process approach to reusing fault cases
- Few experiments and lessons learned



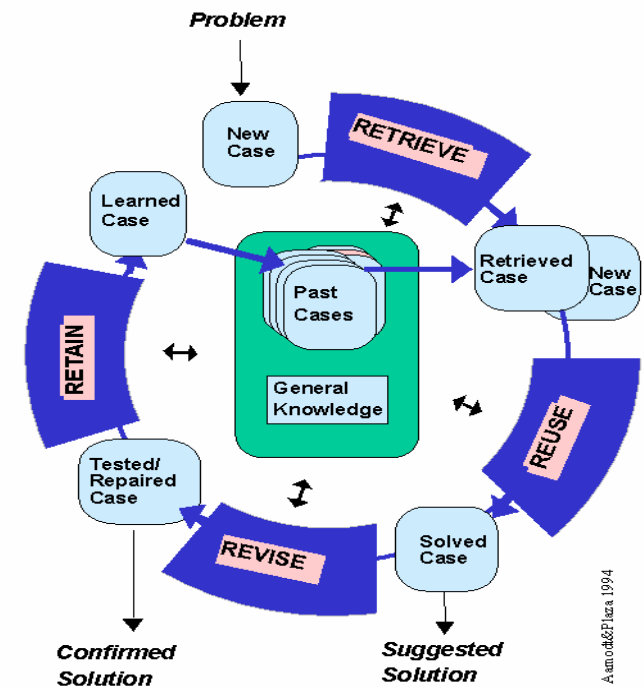
DCBR System

- Peer-to-Peer
 - self-organization
 - scalability in architecture
 - flexibility in search
- Case-Based Reasoning
 - Problem-solving method
 - Classification and resolution
 - Inference on experience



Inside CBR

- Case retrieval
 - Representing cases
 - Obtaining relevant cases
- Case reuse
 - Inferring the retrieved cases
 - Determining the best case
- Case revision and retention
 - Testing the adapted case
 - Updating the case database



Probabilistic Approaches

- Using entire case database
 - Huge number of cases
 - Small set of problem categories
 - Problem classification
 - Automatic learning algorithms

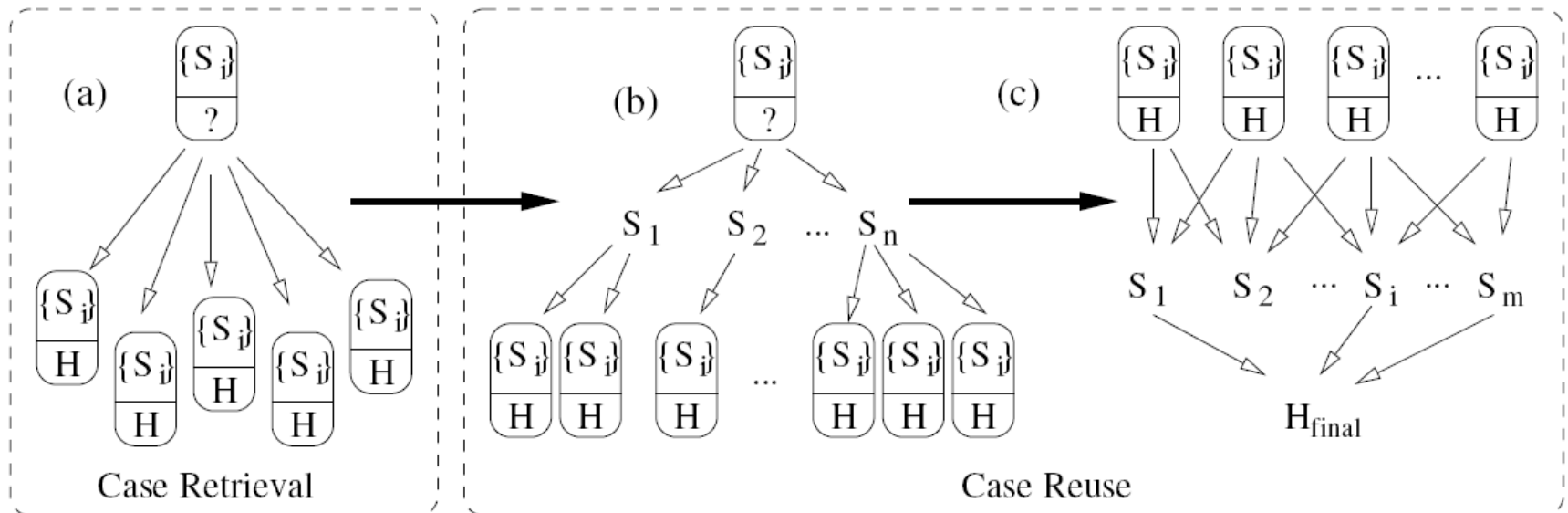
- Using partial case database
 - Small number of good cases
 - Various problem categories
 - Problem diagnosis and selection
 - Support from experts

Two-Process Approach

- Probabilistic approach using partial case database
- Ranking process
 - Restrict the scope of a problem by symptoms
 - Obtain small number of good cases
 - Use k-Nearest-Neighbor algorithm
- Selection process
 - Correlate the obtained cases by symptoms
 - Find the best case for the problem
 - Use Bayesian computation

Two-Process Approach*

- Ranking process (b)
 - Obtain good cases by computing distinct symptoms (S_j)
- Selection process (c)
 - Find the best case by computing correlative symptoms (S_j)



An Example

H_1 = Checking firewall software for blocking connections

- S_1 = Desktop keeps disconnecting from the Internet
- S_2 = Desktop and Laptop keeps connecting from the router
- S_3 = Connection usually goes really slow
- S_4 = Connection is fine before updating the firewall software
- S_5 = Router is WHR-HP-G54 and wireless adapter is Linksys WMP54G

H_2 = Reinstalling networking components (TCP/IP)

- S_1 = Desktop completely stops connecting to the Internet
- S_2 = Laptop can connect to desktop and the Internet
- S_3 = Desktop disconnects to laptop and D-Link router with a limited connectivity
- S_6 = Desktop uses an Etherlink 10/100 PCI card and laptop uses a wireless adapter
- S_7 = Registry was damaged on desktop few days ago

H_3 = Checking router configuration for the IP address range

- S_1 = Desktop cannot connect to a router and the Internet
- S_2 = Laptop connects to the router and the Internet
- S_4 = The firewall software is often updated on those machines
- S_8 = Desktop gets error message of address already used when renewing

$H = ?$

- S_1 = Desktop gets connection failure
- S_2 = Other machines still connect to routers and to the Internet
- S_4 = Desktop updated the firewall software two days ago.

Multiple Vector Representation

- Field-value vector
 - Pre-defined fields of status information
 - Fields of classification and diagnosis information
 - Meta-data search using keyword extraction and evaluation
- Semantic vector
 - Textual description of a problem
 - Full-text search using algebraic computation
- Good performance on the CISI and MED bibliographic data sets

Case Retrieval

- Fault data set
 - Bug reports from bug tracking systems: Bugzilla, Mantis, Trac, Debian, etc.
 - Problem reports from networking forums
- Implementation
 - Porter stemming algorithm
 - Single-vector Lanczos/iterative Jacobi algorithm
 - Testing on x86 64 GNU/Linux machine with two dual-core AMD Opteron(tm) processors running at 1.8 GHz with 16 GB RAM

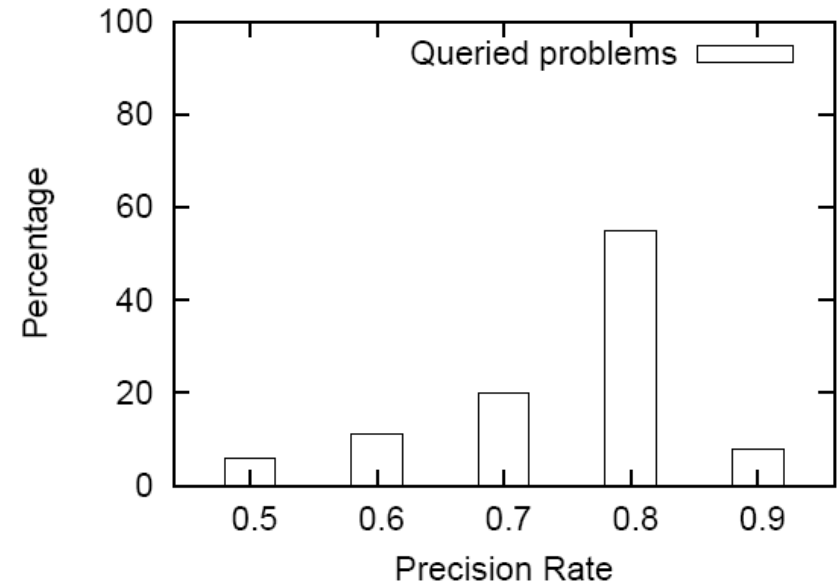
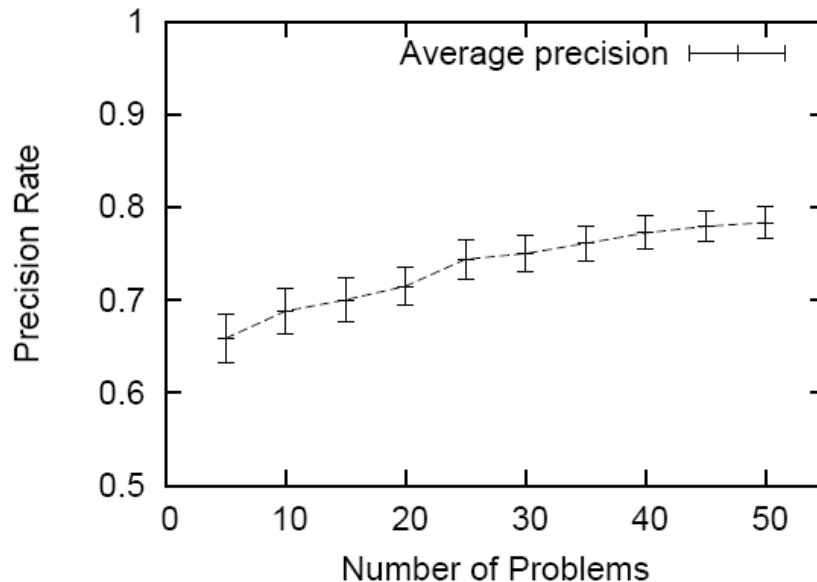
Experiment Setup

- Ranking process
 - Data set contains 71.450 bug reports
 - Problem set contains 50 problems
 - Threshold θ is used to obtain bugs (average of ranked values)

- Selection process
 - Data set contains 60 problem reports
 - Three problem sets S_1, S_2, S_3 with 1, 2, 3 shared symptoms
 - Values correct and incorrect are used to verify obtained solutions

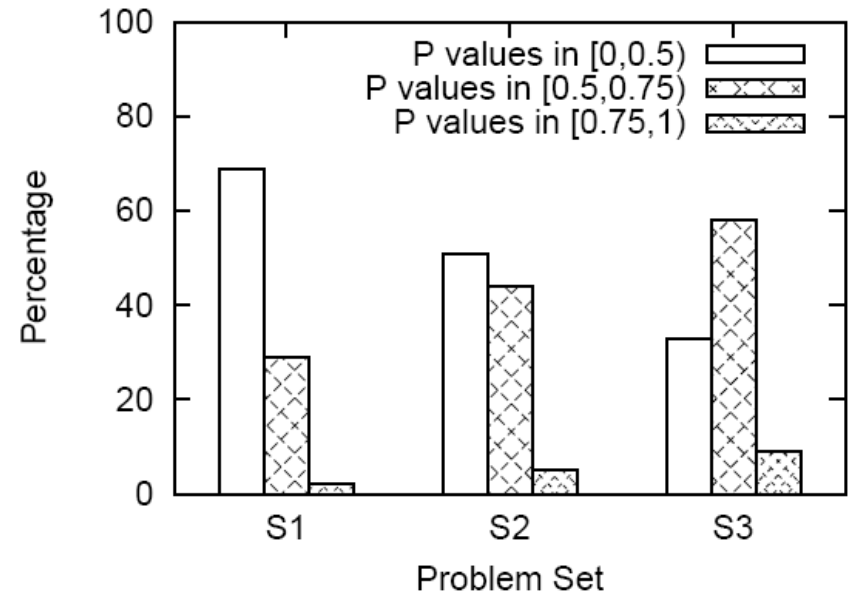
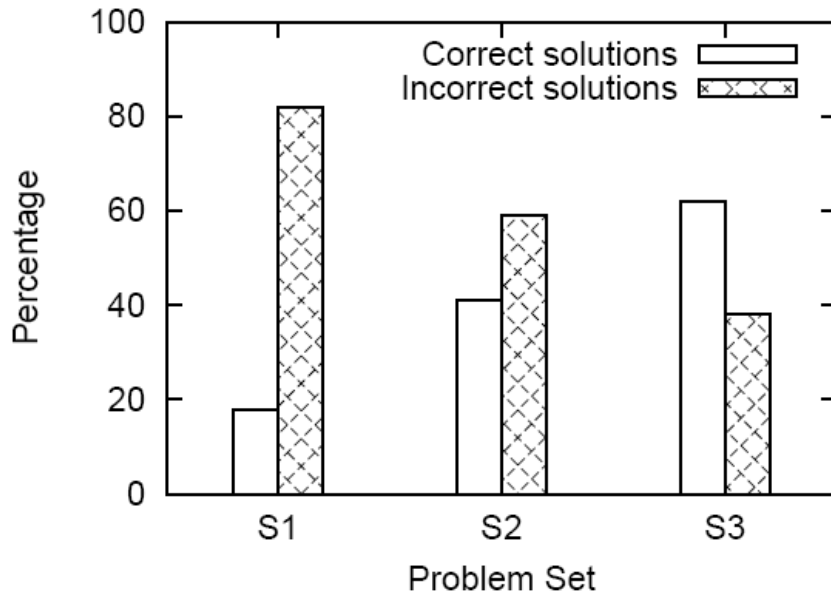
Ranking Performance

- Precision is high
 - Impact of retrieval process impact
 - Relaxation of threshold θ
- But, bug data set is wide and diverse in scope
 - Various problem categories



Selection Performance

- Problems with more shared symptoms
 - Reasonable performance
 - Average probability (due to noisy symptoms)
- Problems with fewer shared symptoms
 - Depending on distinct symptoms



Conclusions

- We studied probabilistic approaches for case reuse in a DCBR system
- We described a two-process approach to reusing fault cases
- We also presented the performance of the ranking and selection processes separately
- Future work focuses on experiments for a distributed setting and a complete system

Thank you for your attention.

Questions?