Software Fault Diagnosis for Grid Middleware with Bayesian Networks

March 29th, 2007

Jan Ploski and Wilhelm Hasselbring

OFFIS Institute for Information Technology
TrustSoft Graduate School on Trustworthy Software Systems
Software Engineering Group, University of Oldenburg

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

The problem of software diagnosis

- **Not every software fault is a software defect:**
  - Missing runtime resources
  - Misconfiguration
  - Misuse
  - Missing understanding
- **Consequence: software misbehaves, for example**
  - Error messages
  - Performance degradation
  - Unavailability
- **How can we find out *why* software misbehaves?**
- **Interpret**

  what we can see to

  find out what we can do about it?

Why can't software just tell us what is wrong?

- **The "cause/effect chasm" [Eisenstadt 1997]**
  - Separation of faults and their symptoms through *time*, *space* and *semantic distance*
- **While error detection can and should be implemented in individual software components...**
- **...fault diagnosis is a system-level issue:**
  - Openness of systems
  - Desire for reuse and flexibility
  - "Leaky" abstractions [Spolsky 2002]
  - Different use contexts

Fault diagnosis on the Grid

- **Heterogenous runtime environments**
- **Heterogenous user communities**
- **Relatively immature technology**
  - Insufficient documentation
  - Usability problems
  - Layers of leaky abstractions

Grid components that provide high-level abstractions when working do expose all gory details when broken.
[Medeiros 2003]

In a large, open software system...

Don't try to make your error messages make sense.

(You will fail.)

Instead,

Help people make sense of your error messages.

Deductive reasoning

- **If proposition A is true, then B is true.**

$$\frac{A \Rightarrow B \quad A}{B}$$

- **If proposition B is false, then A is false.**

$$\frac{A \Rightarrow B \quad \neg B}{\neg A}$$

- **Not applicable under uncertainty!**

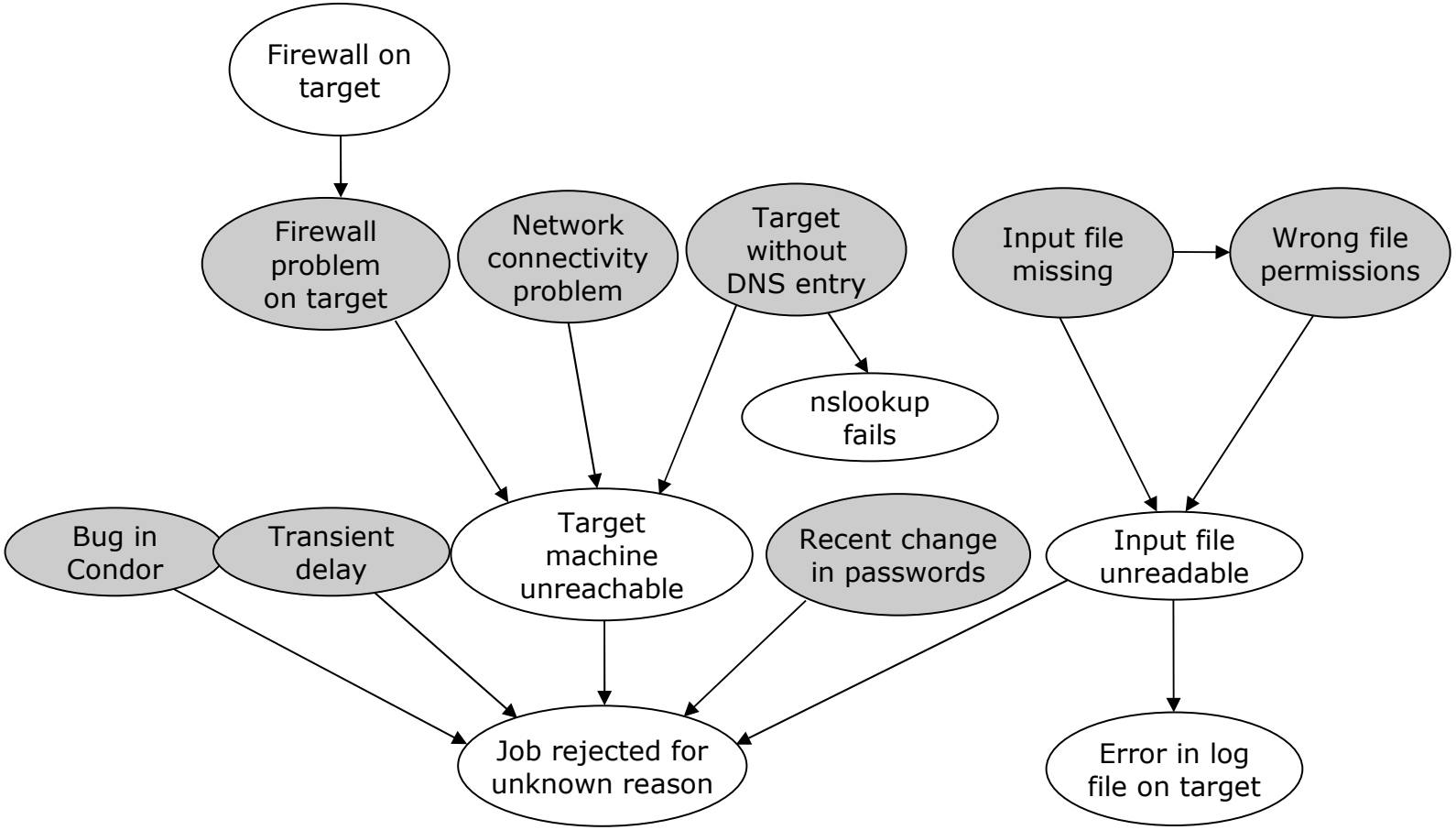
Inductive reasoning

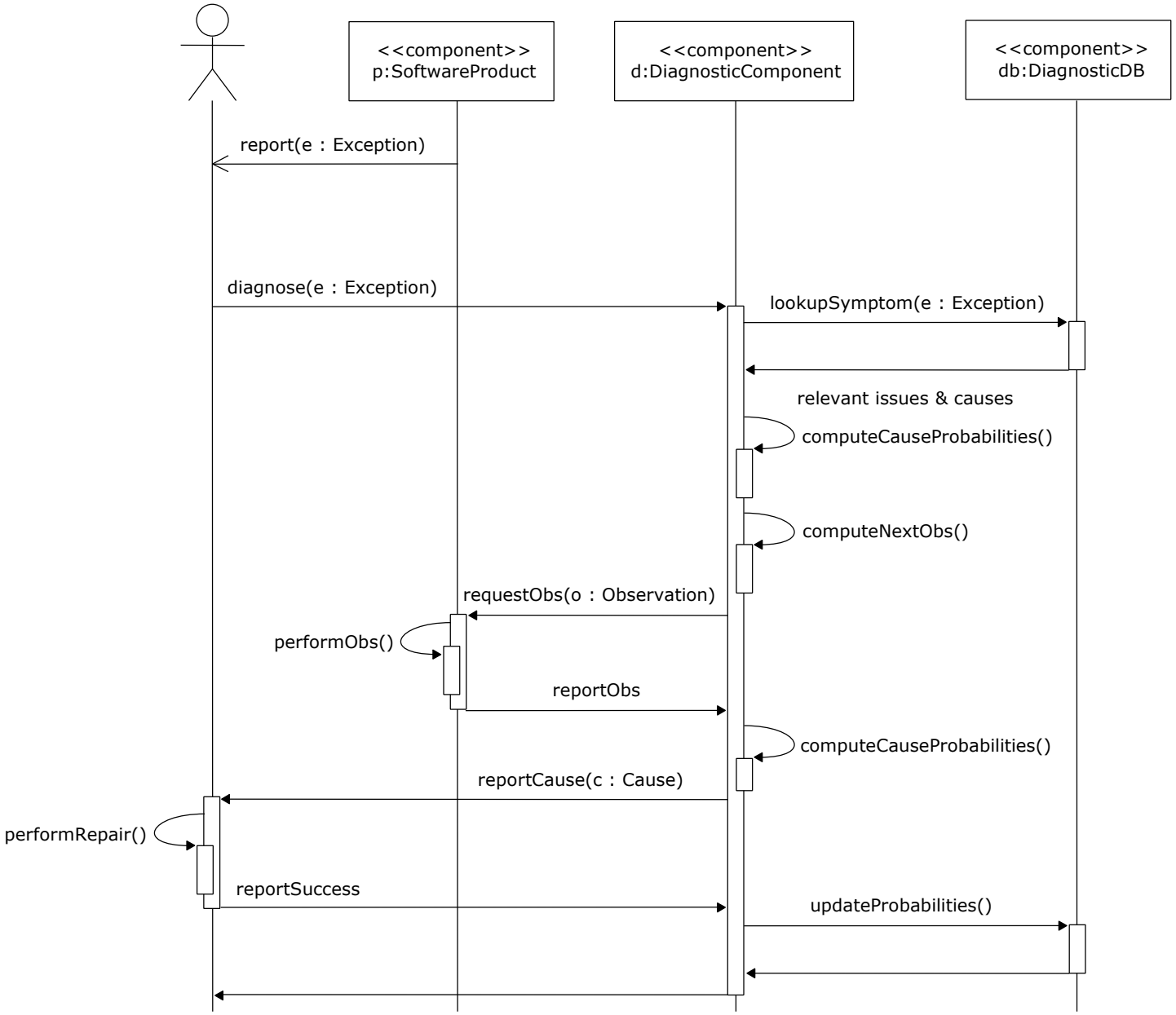
- **If A is true, B is more likely to be true.**
- **If A is more likely to be true, B is more likely to be true.**
- **Different interpretations of probability**
  - as a (limiting) frequency in a series of random experiments
  - as a measure of reasoner's uncertainty and perceived relationships among a set of propositions
- **Problem: inconsistent probability assignments, e.g.**

  $P(A) + P(\neg A) \neq 1$

- **Solution: Bayesian Networks**

Modelling of faults and symptoms with BN




<<component>>
p:SoftwareProduct
<<component>>
d:DiagnosticComponent
<<component>>
db:DiagnosticDB
report(e : Exception)
diagnose(e : Exception)
lookupSymptom(e : Exception)
relevant issues & causes
computeCauseProbabilities()
computeNextObs()
requestObs(o : Observation)
performObs()
reportObs
computeCauseProbabilities()
reportCause(c : Cause)
performRepair()
reportSuccess
updateProbabilities()

Conclusions

- **Software troubleshooting is more than debugging**
  - repeated fault activations are common
  - but for administrators/users, fault activations appear unique
- **Potential for reuse of diagnostic knowledge**
  - bridge the cause/effect chasm using explicit models
  - make models suitable for (semi-)automatic evaluation
  - support online collection of qualitative and quantitative data about faults and their symptoms
- **Challenges**
  - knowledge representation
  - user interface
  - consistency with modelled software