An overview of process algebras for fault tolerance

Vashti Galpin

vashti@cs.wits.ac.za

Programme for Highly Dependable Systems
Department of Computer Science
University of the Witwatersrand

http://www.cs.wits.ac.za/~vashti

Outline and introduction

- process algebras
  - syntax, operational semantics, equivalence semantics
  - example—CCS
- fault tolerance
  - definitions
  - concepts
- fault tolerance and process algebras
  - existing research
    - case studies
    - approaches
    - Janowski's process algebraic approach
- further research and conclusions

Process algebras

- concurrency + interaction
- components
  - syntax
  - operational semantics—define labelled transition system, proofs of transitions
  - equivalence semantics—equate processes with same behaviour, bisimulation
- examples
  - CCS
  - CSP
  - ACP

CCS

- syntax
  - $P ::= \text{nil} \mid \alpha.P \mid P + P \mid P|P \mid P\backslash L \mid P[f]$
  - $\alpha \in \{a, b, c, \ldots, \overline{a}, \overline{b}, \overline{c}, \ldots\} \cup \{\tau\}$
  - $L \subset \{a, b, c, \ldots, \overline{a}, \overline{b}, \overline{c}, \ldots\}$
- operational semantics

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

- equivalence semantics, bisimulation—$P \sim Q$ iff for all α
  1. whenever $P \xrightarrow{\alpha} P'$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \sim Q'$
  2. whenever $Q \xrightarrow{\alpha} Q'$, there exists P' such that $P \xrightarrow{\alpha} P'$ and $P' \sim Q'$

Fault tolerance

- Cristian—fault tolerant system either exhibits well-defined failure behaviour when components fail or masks component failures to users.
- Arlat *et al*—fault tolerant system fulfils its intended function despite the presence or occurrence of faults, fault tolerance is achieved through redundancy.

  Terminology:
  - *fault* is the cause of an error
  - *error* is a state that may lead to failure
  - *failure* occurs when service is not delivered

Dependability

- reliance can be justifiably placed on the service a system delivers.
- different aspects: availability, reliability, safety, confidentiality, integrity, security, maintainability
- to achieve dependability, use a number of different methods:
  - fault prevention
  - fault tolerance
  - fault removal
  - fault forecasting
- validation of fault tolerance—fault injection, use to evaluate effectiveness of fault tolerance mechanisms

Overview—case studies

- Jifeng and Hoare (Distr Comp 2, 1987)—uses CSP to describe and prove correct a distributed recovery algorithm
- Rowson (Tech Rep, 1991)—specifies and verifies ISO communication protocol using CCS, including error recovery methods
- Bruns (CAV '92)—models railway interlocking using CCS including failure behaviours and failure-handling mechanism, verifies safety properties
- Gilmore *et al* (Int J Prod Res 34, 1996)—uses a stochastic process algebra to model performance of robot control with and without failures
- Bernardeschi *et al* (FastAbstracts: FTCS 28, 1998)—uses process algebra to verify correctness properties of GUARDS project, represents faults as actions, uses standard concurrency tool kit

Overview—approaches

- Peleska (Distr Comp 5, 1991)—models fault tolerance achieved by dynamic redundancy in CSP, proposes a general approach for proving correctness properties
- Weber (FTRTFTS '93)—uses a notion similar to bisimulation to show fault-tolerance, distinguishes fault-tolerance from correctness
- Amadio and Prasad (FST-TCS '94)—presents extension to π-calculus with locations and failures, gives example of small fault-tolerant program
- Krishnan (TCS 128, 1994)—CCS-based, models majority voting, pre-orders to characterise relativised fault-tolerance, notion of fault injection
- Janowski (PhD thesis, 1995)—CCS-based approach to modelling fault-tolerance
- Riely and Hennessy (ICALP '97)—gives process algebra to describe a model of locations and failures, provides number of semantic equivalences

Janowski's research

- introduces faulty transitions to labelled transition systems

  $$\mapsto = \rightarrow \cup \dashrightarrow$$

- fault-tolerant bisimulation, may bisimulation, $P \Subset Q$ iff for all α
  1. whenever $P \xrightarrow{\alpha} P'$, there exists Q' and s such that $Q \xrightarrow{\alpha} Q'$, $\hat{s} = \hat{\alpha}$ and $P' \Subset Q'$
  2. whenever $Q \overset{\alpha}{\mapsto} Q'$, there exists P' and s such that $P \xrightarrow{\alpha} P'$, $\hat{s} = \hat{\alpha}$ and $P' \Subset Q'$
- fault monotonic theory—if correct for n faults, then correct for $< n$ faults
- conditional fault-tolerance—use finite deterministic automaton to say when faults can occur
- process description language—CCS with recursion
- fault description language—subset of CCS including recursion
- suitable for incremental refinement
- applications—two-phase commit, alternating bit protocol, mutual exclusion, distributed consensus

Further work

- application of approaches to PHDS virtual redirector project
- use of stochastic process algebra to evaluate efficiency of fault-tolerance mechanisms
- application of extensions of CCS to fault-tolerance

Conclusions

- overview of process algebras for fault-tolerance
  - definitions
  - case studies
  - approaches