An Architecture for Configurable Dependability of Application Services

Matthias Tichy
mtt@uni-paderborn.de
Software Engineering Group
University Of Paderborn
Contents

• Introduction
• Architectural Principles for Dependability
• Architecture
• Improving Availability
• Improving Reliability
• Conclusion & Future Work
Introduction

• Dependability
  – Availability
  – Reliability
  – Safety
  – Security
Architectural Principles

• Service Registry
  Provides support for dynamic online binding and spontaneous networking.

• Leasing
  The leasing principle extends the allocation of resources with time. The lease represents a period of time during which the resource is offered.

• Proxy
  A proxy is the placeholder for another object.

• Smart Proxy
  Smarter version of the proxy, may be a placeholder for more than one object.

• Redundancy
  Redundant services prevent a single-point-of-failure.

• Replication
  Replicating is the process of maintaining multiple copies of the same entity at different locations.
Architecture

- Architecture provides means to achieve high availability for application services
- Reliability is highly application specific
- Every infrastructure service is executed on every node
  - Redundancy of services
  - Replication of data
  - Service registry
Architecture – Monitor (Availability)

- For each application service instance, one monitor supervises its execution ⇒ configurable degree of availability
- Coordination by monitor responsibilities (registry + leasing)
Reliability

- Highly application specific
- 3 types:
  - Stateless session service
    - No problem, just use another service instance
  - Stateful session service
    - Relevant history must be replayed on another service instance
  - Entity service
    - Replicate data and use an appropriate consistency model
Architecture – Responsibility Storage (Reliability)

- Example for an entity service
- Smart Proxy communicates via Multicast messages
- Decentral majority voting
- Redundancy, replication, smart proxy

![Diagram of a smart proxy communicating via multicast messages with voting]

Messages:
1: getData("Chart")
2: "Mon2"
3: "Mon2"
4: "Mon5"

■ Timeout
Conclusion & Future Work

- Architectural principles for dependability
- Architecture based on these principles
- Provides means to achieve a configurable degree of availability
- Example for providing application specific reliability
- Implementation based on Jini

- Seamless UML support for service-based architectures
- Runtime measurements to adapt architecture parameters
- Complex embedded and real-time systems
Thank you for your attention!

Questions?
Service Description Storage (Reliability)

- Redundant and distributed storage of the service descriptions
- Strong consistency (sequentiel)
- Probability: \( P(\text{Read}) \gg P(\text{Write}) \)
- Algorithm „Weighted Voting“
- Implemented in a smart proxy