Université de Genève

Designing Fault-Tolerant Mobile Systems

Giovanna Di Marzo Serugendo
University of Geneva, Switzerland

Alexander Romanovsky
University of Newcastle upon Tyne, UK

Outline

- Mobile Agents
- Fault-Tolerance and Mobile Agents
- Example
  - Electronic marketplace using mobile e-purses
- Techniques
  - Meta-Agent
  - Coordinated Atomic Actions
  - Asynchronous resolution
  - Self-Repair
  - Proof Carrying Code

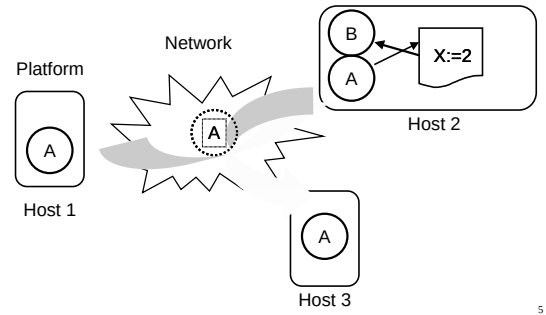
Mobile Agents = …

- Agents
  - Autonomous running entities, able to take decisions and initiatives
  - Work on behalf of some user (or other system)
  - Several agents can work co-operatively (to solve some task) or in **competition** (for acquiring some resource)
- Mobile agents
  - Moves from one platform to another
  - Weak mobility: code only moves, not internal state;
  - Strong mobility: both code and internal state moves, execution is resumed at remote location

Mobile Agents = …

- Mobile Systems characteristics:
  - Open: an agent interacts with other agents or systems
  - Large-scale size (agents roam the whole Internet)
  - Asynchronous communication (use of a blackboard)
  - Decentralised (no central control)
- Platforms
  - Need of full portability of mobile code
  - Usually Java-based platforms

Mobile Agents = …



Mobile Agents: Applications Domains

- Network Management
- Remote Device Control and Configuration
- Active Networks
- Wireless Applications
- E-commerce, M-commerce
- Distributed Information Retrieval
- Active Documents
- Workflow Management
- Grid Computing and Global Computing
- Emerging
  - IP telephony, service composition
- … and Fault-Tolerance!

Fault-Tolerance and Mobile Agents

- Abnormal Situations
  - Underlying components failures
    - Platform: memory, disk usage, unavailable resources
    - Network: communication delays, node crashes, disconnection
    - ☛ Tolerated by underlying support
  - Partner's errors
    - Impossibility of interaction, failures
    - ☛ Co-operative resolution
  - Agents own faults
    - ☛ Local resolution

7

Fault-Tolerance and Mobile Agents

- Mobile Agents Requirements
  - Recovery from errors at remote locations
  - Light recovery code
  - Impossible to maintain checkpoints (backward error recovery)
  - Mobile agents cannot stay in transactions
  - Unanticipated abnormal situations
  - Unknown locations of mobile agents

☛ Fault-tolerance at the application level

☛ Forward-error recovery

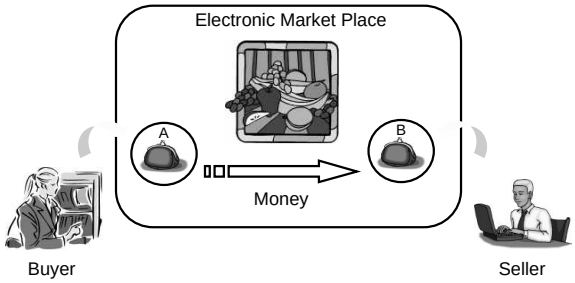
8

Fault-Tolerance and Mobile Agents

- Evaluation Criteria
  - Structuring Fault Tolerance (Built-in/Separated)
  - Error processing/Error Confinement
  - Recursive Structuring and Scalability
  - Overheads
  - Flexibility, Run-time reconfigurability

9

Example



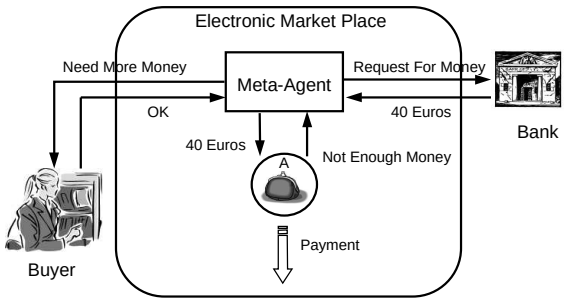
10

Fault-Tolerance Techniques

- Meta-Agent
- Coordinated Atomic Actions
- Asynchronous Resolution
- Self-Repair
- Proof Carrying Code

11

Meta-Agent



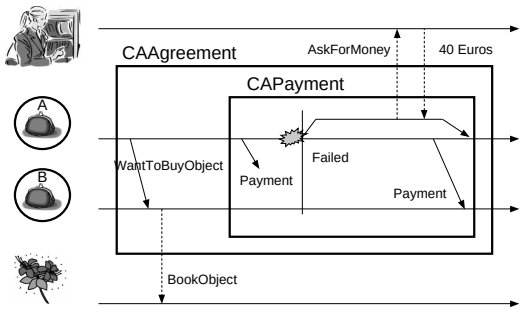
12

Meta-Agent

- Characteristics
  - Error handling code downloaded at run-time
  - Several exception handlers available simultaneously
  - No overhead in the agent code size
- Good for ...
  - Asynchronous recovery from local errors
  - No need for a co-operative resolution scheme
- Application domains
  - Wireless applications
  - Active networks

13

Coordinated Atomic Actions



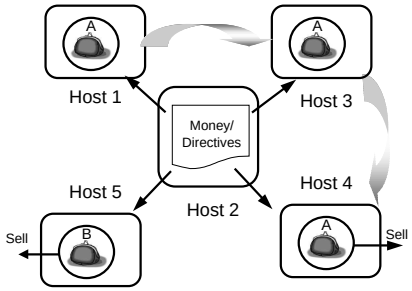
14

Coordinated Atomic Actions

- Characteristics
  - Define damage area (CA action)
  - Recursive System Structuring
- Good for ...
  - Complex applications involving co-operating agents
  - Need for co-operative handling of exceptions
- Application domains
  - Active documents
  - Workflow management applications

15

Asynchronous Resolution



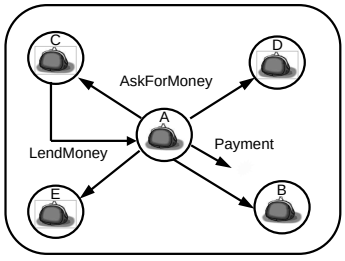
16

Asynchronous Resolution

- Characteristics
  - Scales to large-size systems
  - Combined with meta-agents
    - increased flexibility
    - overhead
- Good for …
  - Asynchronous recovery
  - Collaborative agents
    - can be individually recovered from same exception
    - in their particular location
- Application domains
  - Grid computing
  - E-commerce, M-commerce

17

Self-Repair



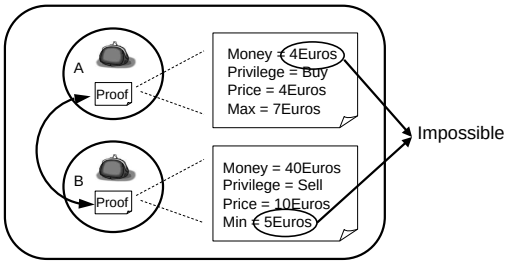
18

Self-Repair

- Characteristics
  - Scales to large-size systems
  - Error confined in the agent
  - Overhead in the code size
  - May employ meta-agents
- Good for …
  - Agent having full autonomy
  - Agent needs to recover even partially
  - Decentralised control for functionality and fault-tolerance
  - Situations requiring quick resolution
- Application domains
  - Wireless applications
  - Remote device control
  - Active Networks

19

Proof Carrying Code



20

Proof Carrying Code

- Characteristics
  - Proof is part of the agent, not customizable
  - Limited flexibility for recovering unanticipated errors
  - Overhead of code size and execution time to execute the proof
- Good for …
  - Agents interacting with unknown agents, or ...
  - … agents using different standards
- Application domains
  - Service discovery and composition at run-time

21

Conclusion

- Mobile Agent Systems
  - Specific characteristics: decentralised, dynamic, roaming entities
  - Specific abnormal situations: migration errors, resources and security problems, partner's failures

  ☞ Investigation of new techniques for fault-tolerance at: application level + forward error recovery
- Application to:
  - biologically inspired systems
  - portable devices applications (physical mobility)

22