

Dependability and QoS in Large-Scale MASs



Moderator:

- ◆ *Rogério de Lemos (University of Kent, UK)*

Panellists:

- ◆ *Paolo Bresciani (ITC-irst, Trento, Italy)*
- ◆ *Tom Maibaum (King's College London, UK)*
- ◆ *Alexander Romanovsky (University of Newcastle, UK)*
- ◆ *Arndt von Staa (PUC-Rio, Brazil)*

Computing systems can be characterized by five fundamental properties:

- ◆ functionality, usability, performance, cost, and dependability;

Quality of services (QoS) refer to the non-functional properties:

- ◆ usability, performance, and dependability;

Dependability of a computing system is the ability to deliver service that can justifiably be trusted.

- ◆ threats, attributes, and means;



“The interplay between the notions of *agents* and *objects* from a software engineering perspective.”

- ◆ *object* - a passive or reactive element that has *state* and behaviour and can be related to other elements;
- ◆ *agent* - an *autonomous, adaptive* and interactive element that has a *mental state*;

Taming Agents and Objects (*Silva et al 2003*)

Agents and Dependability



How an agent based approach may improve the dependability attributes of software systems?

- ◆ focus on fault tolerance:
 - ◆ provision of services despite the presence of faults;
- ◆ *autonomy & adaptability*:
 - ◆ what are the failure assumptions for autonomy?
 - ◆ how to evaluate the system attributes?
 - ◆ uncertainty leads to unpredictability;
 - ◆ what are the criteria for stability?
- ◆ *mental state*:
 - ◆ is it something can be introspected?
 - ◆ if state cannot be observed then it cannot be controlled;

Agents and Dependability



What is the role of agents in large scale systems?

- ◆ basic building blocks:
 - ◆ e.g., functions, objects, and components;
- ◆ additional agency layer for specific services:
 - ◆ SAFEGUARD IST Project: Security and Safety in Large Complex Critical Infrastructures;
 - ◆ collect information and control the allocation of resources and inform third parties;
- ◆ integration into legacy systems;

Agents and Dependability



What restrictions to imposed on agents to guarantee QoS?

- ◆ object-oriented/object-based:
 - ◆ real-time systems;
- ◆ Ada/SparkAda
 - ◆ safety-critical systems;

Agents and Dependability



MASs and the provision of dependable services, is it something like “*giants with clay feet*”?

- ◆ what are the failure assumptions to be associated with agents and their communications?
 - ◆ to tolerate certain classes of faults minimum redundancies are required;
 - ◆ malicious faults;
- ◆ how to guarantee agreement between different agents?
 - ◆ dynamic reconfiguration;

What makes dependability and QoS in MASs different from object-oriented systems (OOSs)?

- ◆ How do MAS features (e.g. autonomy, self-adaptation, intelligence, mobility, emergent behaviour...) make dependability/QoS easier or more difficult than in OOSs, from the software engineering viewpoint?
- ◆ To what extent conventional and OO dependability/QoS techniques can be used in the context of MAS engineering?
- ◆ Which are the challenges facing the promotion of dependability/QoS in MAS development?

Panellists



- ◆ *Paolo Bresciani (ITC-irst, Trento, Italy)*
 - ◆ security and criticality analysis in AOSE;

- ◆ *Tom Maibaum (King's College London, UK)*
 - ◆ agents and standard software engineering ideas and techniques;

- ◆ *Alexander Romanovsky (University of Newcastle, UK)*
 - ◆ fault tolerance in large MASs;

- ◆ *Arndt von Staa (PUC-Rio, Brazil)*
 - ◆ agents and dependability;