Lifelong Verification of Dynamic Service Compositions
Domenico Bianculli
domenico.bianculli@lu.unisi.ch - Università della Svizzera italiana, Facoltà di scienze informatiche, Lugano, Switzerland

Motivations
• Service-based applications are an instantiation of open-world software
• Service compositions are made out of components that are and remain out of the control of the service aggregator
• The dynamic characteristics of the services and of the environment may invalidate verification results obtained at design time
• Additional verification activities are required at run time
• Existing approaches focus on analysis and verification techniques only for a specific service life cycle phase

Approach
• Model-driven and technology-neutral approach for continuous lifelong verification
• Goal: to provide a methodology and the accompanying tools to engineer dependable service compositions
• External services are known through their assumed specifications
• A service composition is verified against some guaranteed specifications
• Reuse of existing work on service compositions modeling languages
• Contributions:
  • a formal specification language
  • a design-time verification technique
  • a run-time verification technique

Specification language
• Support for:
  • assertions on service state
  • properties on events traces
  • QoS attributes (response time, throughput, reliability)
  • stateless and stateful interactions
  • Specifications can be associated with specific execution points
  • Technology-neutral

Design-time verification
• Based on model checking
• Challenges:
  • finding the proper abstraction level for modeling the environment and the exchanged data
  • support for compositional verification
  • handling time-related and probabilistic properties

Run-time verification
• Architecture based on software monitors and failure detectors
• Goal: minimal execution overhead
• Design decisions:
  • how to collect and analyze data
  • degree of invasiveness of the instrumentation
  • timeliness in discovering undesirable situations

Contributions:
compositions modeling languages

Examples:
- assumed specifications
- guaranteed specifications

Design of the service composition model
- service 1
- service n

Execution environment
- service 1
- service ...
- service n

Formal Verification Tool

Run-time verification
- service 1
- service n

Deployment

Validation
- service compositions
- violations

Service activation
- service compositions
- service aggregators

Verification
- design-time verification
- run-time verification
- service compositions
- assumed specifications
- guaranteed specifications