Services and Objects:
Open issues

Vincenzo D’Andrea and Marco Aiello
DIT - University of Trento, Italy
an object is a service

a service is an object

OOP

SOC
what are the counterparts, in SOC, of:
  - class
  - instance

is it possible to apply techniques as containment or inheritance to services?

the goal: to exploit methodological tools available in OOP
"Service Oriented Computing Manifesto"

Services are autonomous platform-independent computational elements that can be described, published, discovered, orchestrated and programmed using XML artifacts for the purpose of developing massively distributed interoperable applications.

Do we need a definition of OOP?
OOP

invoke

shared context

synchronous method invocation

SOC

find-bind-use

multiple context

asynchronous method invocation
COMPOSITION:

- a key concept in SOC
  - in OOP: design activity
  - in SOC: dynamic discovery

- SOC: no precise definition of composition
  - "syntacticians" versus "semanticians"
INHERITANCE:

- OOP: derivation of a class from another
  - subclass or subtype (semantic coherence)
  - interface inheritance
  - a special kind of composition

- SOC: related concepts
  - service interface
  - service substitution
composition

subtype

subclass
POLYMORPHISM:

- OOP: related to inheritance
  - inclusion
  - overloading

- SOC:
  - requires definition of inheritance and types
  - same service different semantic
more …

- statefull object / stateless class
- encapsulation / information hiding
- separating interface and implementation
- frameworks
- design patterns and other SE tools
- …
summary & final remarks

- OOP - solid and well established
- SOC - new emerging field
  - goal: global interoperability between independent services
  - lot of open issues ...

- can we "borrow" concepts from the OO world?

- services exhibit a number of object-like behaviors

- inheritance and polymorphism for composition semantics
- need of state information for services

- space for more contamination between OOP and SOC