



# An Object-Centric Programming Framework for Ambient-Aware, Service-Oriented Sensor Networks



VANDERBILT UNIVERSITY

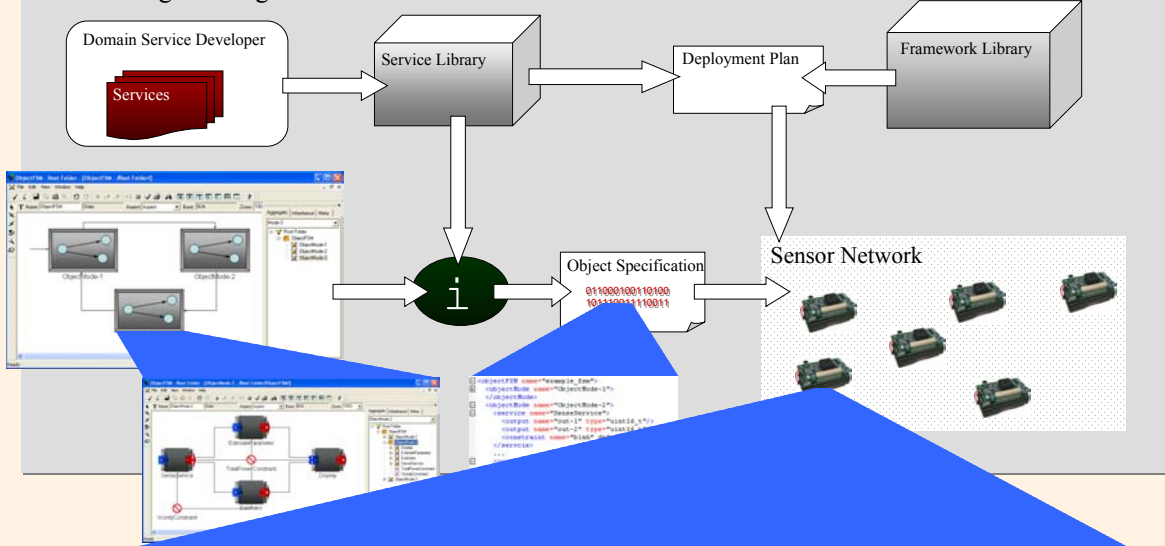
Institute for Software Integrated Systems  
Vanderbilt University

Princeton University

Manish Kushwaha, Isaac Amundson, Xenofon Koutsoukos, Sandeep Neema, Janos Sztipanovits

Chang Hong Lin, Wayne Wolf

## OASiS: Programming Framework



## OASiS: Overview

- Object-Centric programming abstraction for sensor networks
- Combines the concepts from object-centric programming, ambient-awareness and service-oriented architecture to provide reliability
- Prototype implementation in galsC

## Object-Centric Programming Framework

- Objects represent external physical entities
- Objects are first-class programming elements
- Provides application developer a higher level of abstraction

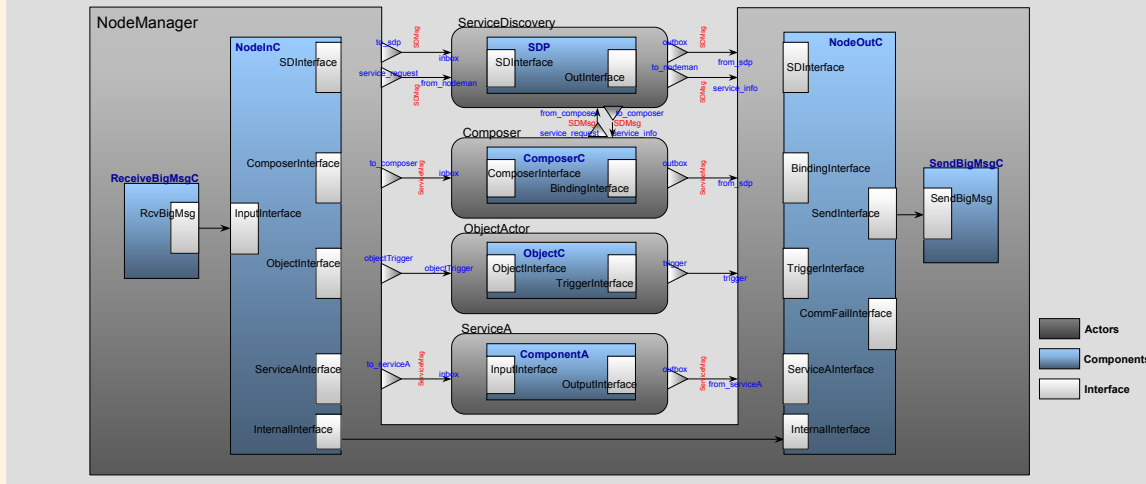
## Ambient-Aware Sensor Network

- Dynamic service discovery by the object
- Key points of ambient-aware networks are,
  - Reflective objects allow reconfiguration and error recovery,
  - organize the distribution of collaborative tasks,
  - ensure that the application cannot be blocked, for example, due to a link failure

## Service-Oriented Architecture

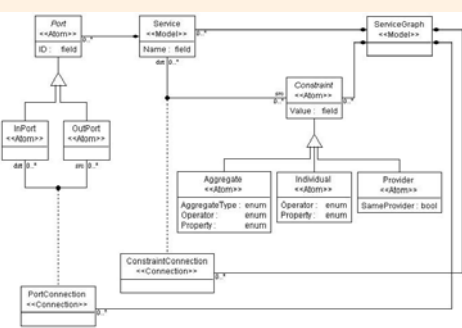
- SOA comprises of loosely-coupled services that are modular, autonomous, and have well-defined interfaces that allow them to be described, published, discovered and invoked over the network

## Node Architecture



## Service Graph Metamodel and Constraints

### Service Graph Metamodel

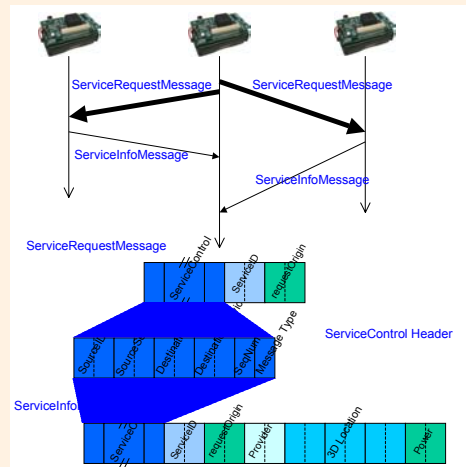


- Contains one or more services connected by service connections
- Contains service constraints
- Service constraints are associated with one or more services
- Service constraints are defined as composite operations on properties of services

### Service Graph Constraints

- Provider Constraint**
  - Specifies that services must either run on the same node, or must all run on separate nodes
- Individual Constraint**
  - Specifies a constraint that applies only to a specific service
  - Specified by:
    - Property (power level, position, etc.),
    - Operator (=, ≠, >, ≥, <, ≤, etc.),
    - Comparison Value (provided by user)
- Aggregate Constraint**
  - Specifies a constraint that must hold across several services
  - Specified by property, operator, and comparison value
  - Additional attribute:
    - Aggregate Type (sum, average, etc.)

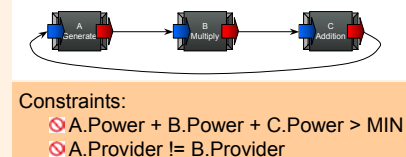
### Service Discovery



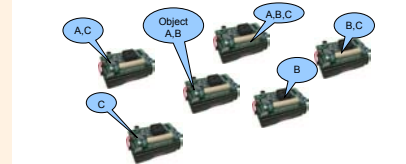
- On-Demand service discovery
- Broadcast service request message
- Service provider reply with relevant service information message

### Example Application

#### Computational Operation Chain

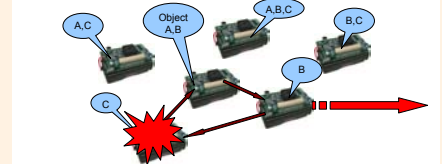


#### Scenario #1



Select among multiple service providers that satisfy the constraints

#### Scenario #2



Communication failure – nodes die or move out of communication range