Multi-Dimensional Service Compositions

L. Baresi, E. Di Nitto, S. Guinea
Politecnico di Milano
{baresi | dinitto | guinea}@elet.polimi.it

S. Dustdar
Vienna University of Technology
dustdar@infosys.tuwien.ac.at

Pervasive Computing

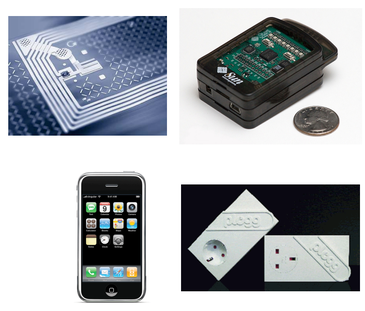
- **Internet as an Aggregator**
  - Allows for complex applications that mix data, logic, and presentation from different sources
- **Open-world Assumption**
  - Applications should be robust with respect to evolving scenarios and evolving requirements
  - Situational applications take advantage of the context of execution
- **Wine Transportation Example**
  - Temperature, light, and humidity sensors are needed
  - GPS used to track location

Software as a Service

- Simple and Lightweight model
- Accessible to anyone through Internet technology
- No need for installation and easy to maintain

Internet of Things

- Internet-enabled things are everywhere
- More than 10000 things per person in the next 10 years
- RFID, Sensors, GPS



Ad-hoc Development

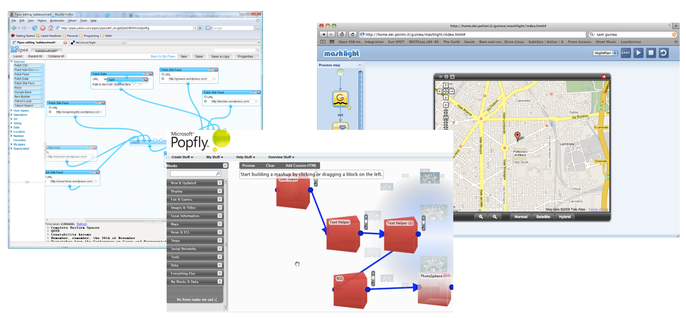
- Have professional designers develop the application by hard-coding the interactions amongst the parts
- Rely entirely on the developer's knowledge of the underlying technologies and APIs

Workflow-based Development

- Formally defined orchestrations
- Centralized execution environment
- For example, BPEL for Web Services

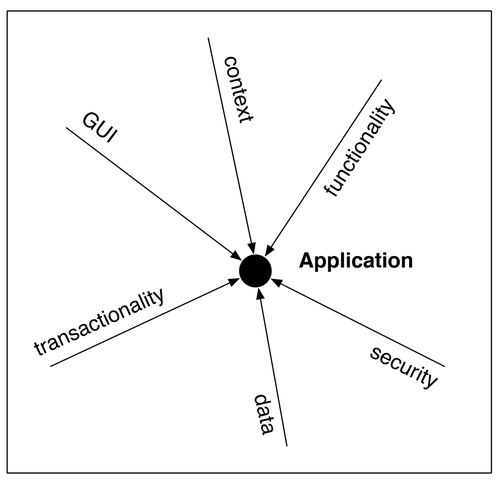
Mashup-based Development

- Do-it-yourself approach
- Web 2.0 development for end-users
- Yahoo pipes, MS Popfly, Mashlight



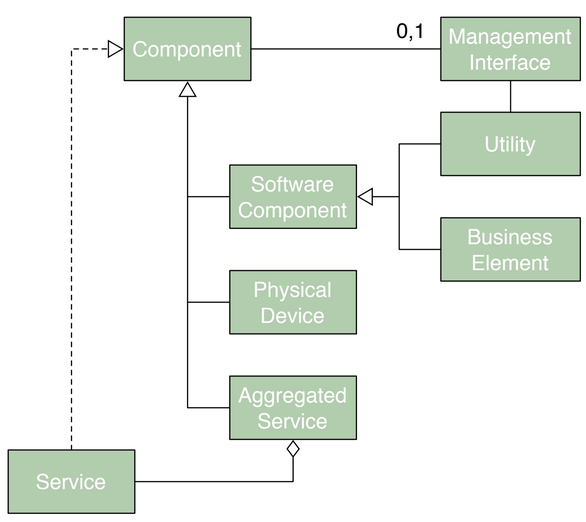
Our Vision

Acknowledge that the problem is multi-dimensional



- Identify key cross-cutting concerns

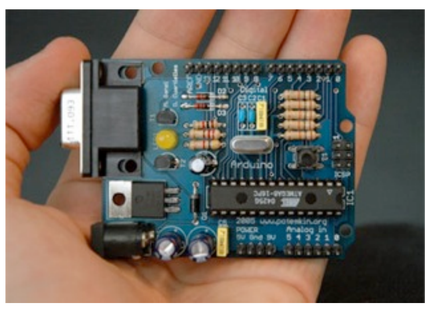
Adopt a model-based approach



- Treat things and cross-cutting concerns as key players in the model

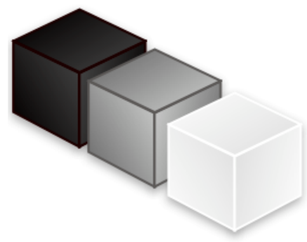
Provide appropriate abstractions

Service abstraction



- Two levels of abstraction:
  - Physical level
  - Service level

Services as manageable and configurable entities



- Black-, grey-, and white-box entities need management interfaces
- Use instrumentation/AOP as enabling technology

Future Work

- Refine conceptual model behind the idea of service
- Further research cross-cutting concerns as first-class composable entities
- Define computational model compatible with the defined abstractions
- Implement infrastructure that allows for run-time adaptation through management and customization