Concepts and Challenges in Service Composition

Dominik Kuropka

Business Process Technology
Hasso Plattner Institute of IT Systems Engineering
at the University of Potsdam

HPI at a glance

- Founded by Hasso Plattner in 1998 as public private partnership with University of Potsdam
- Bachelor and master university programs in IT Systems Engineering
- 356 students (275+81)




Teaching and Research at HPI

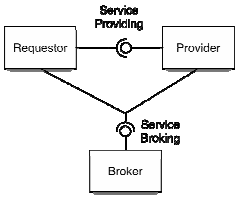
- Great emphasis on scientific excellence, practice orientation, and cooperation with industry
- National and international research projects, e.g.,
  - PESOA: BMBF funded; 6 partners, 4 M€, 3 years
  - ASG: EU FP6 funded; 21 partners, 11.5 M€, 2 years
- Teaching activities associated with research
  - SAP Labs Berlin, Native XML ext to MaxDB: NaXDB (2004)
  - SAP Research, BPEL ext to SAP BPM Suite: Canguru (2005)
  - Master thesis projects with SAP SI

Concepts and Challenges in Service Composition

1. Spirit of SOC
2. Service Composition: Adaptive Services Grid
3. Challenges and Conclusions

Service Oriented Computing

- Services
  - Loosely coupled computing tasks communicating over the net
- Service-oriented Architecture
  - Organizing principles that facilitate dynamic automated discovery and use of services
- Roles
  - Requestor, Broker, Provider



[Burbeck: The Tao of e-business services (2000)]

Service Definitions

There are many definitions existing, samples:

- Everything providing an WSDL-complaint interface is a web service. (W3C)
- A service is the non-material equivalent of a good. (Economics)
- E-business services are loosely-coupled computing tasks communicating over the Internet. (IBM)

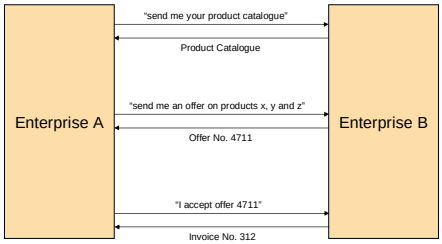
=> We need to agree on what a Service is.

OOP vs. SOC

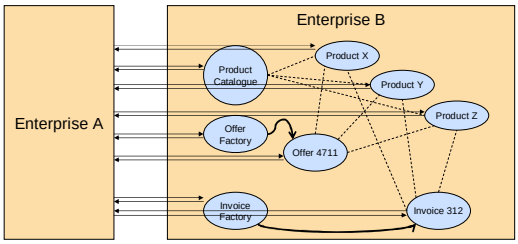
Object Oriented Programming
- Everything is an object
- Objects have attributes and methods
- Methods are invoked by messages
- **Stateful, fine granular communication**

Service Oriented Computing
- Every functionality is a service
- Services provide an interface and operations
- Services are requested by messages
- **Ideally stateless, document oriented**

Sample-Scenario: Without computing

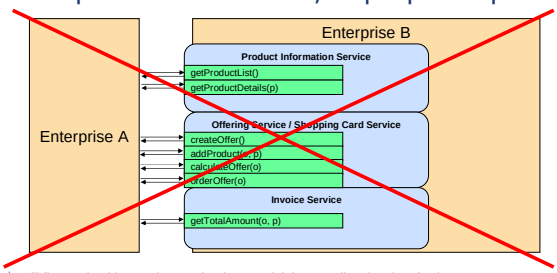


Sample-Scenario: OOP



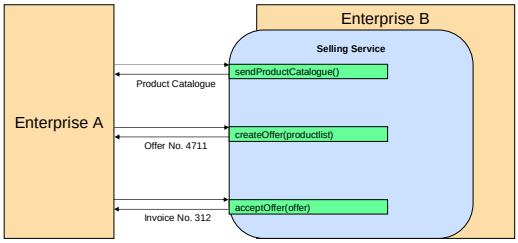
Fine-grained interaction mechanisms, tights coupling, complex process
=> Useful for low-level, local interfaces like GUI, Task Scheduling, Database Abstraction
=> Less useful for high-level business processes and services

Sample-Scenario: SOC, improper impl.



Still fine-grained interaction mechanisms and tights coupling despite of using SOC.
Services are tailored according to technological issues.
Complex process.

Sample-Scenario: SOC, proper impl.



Seldom high-level interactions ease loose coupling.
Services are tailored according to business needs.
Focus on document transfer and simple choreography/processes.

Intermediate Conclusions

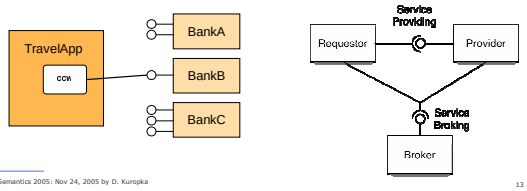
In this context:
- A service is a well-specified unit of work offered by a provider, which can be performed on demand.
- Services…
  - should be high level, document oriented
  - should be tailored by business needs
  - should represent elementary business capability
- Simply using SOA does not lead to "proper" service tailoring and SOC.
- Every functionality can be seen as a service. However it is not always useful to do so.

Concepts and Challenges in Service Composition

1. Spirit of SOC
2. Service Composition: Adaptive Services Grid
3. Challenges and Conclusions

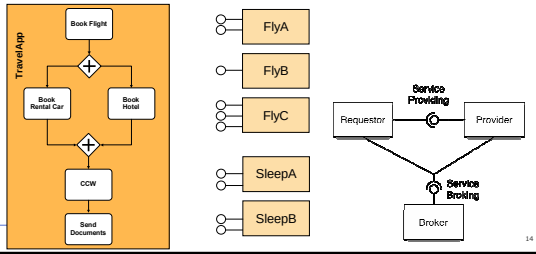
Static Binding

- Service impl. bound at design time
  - Providers register their services
  - Ambiguities in description resolved by programmer during application implementation
  - Appropriate for rather static service landscape



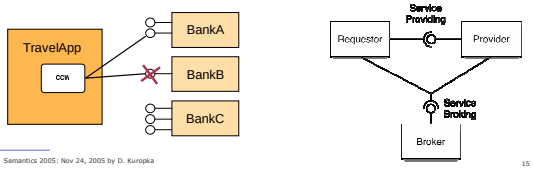
Service Composition

- Service composition realizes business process
- Is a laborious and costly tasks
  - => many steps: discovery, binding, data type mediation
  - => continuous adaptation needed

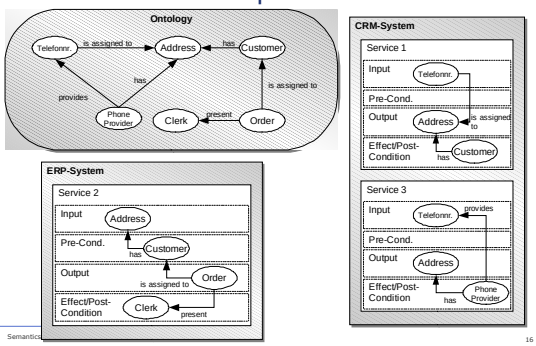


Dynamic Discovery and Binding

- Selection of proper service implementation at run time
  - => Optimization possibilities: e.g. take the cheapest service
  - => Failure recovery: choice of an alternative service impl.
  - => New services can be used without changing the application
- Requirement: Semantics
  - A formal specification of service semantics is required
  - Specifications need to be based on a common understanding, realized by domain ontology

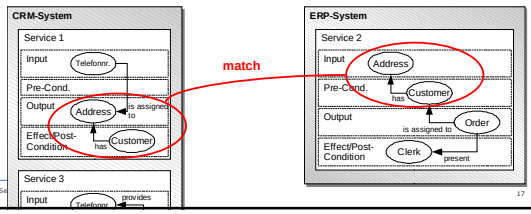


Semantic Service Specification



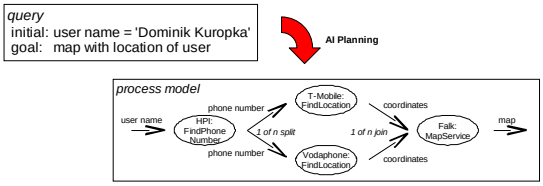
Additional value of semantic specifications: Semi-Automated Service Composition

- By using semantic service specification process modelling tool can help user in
  - Discovery of proper services
  - Data type mediation
  - Validity checking



Additional value of semantic specifications: Automated Service Composition

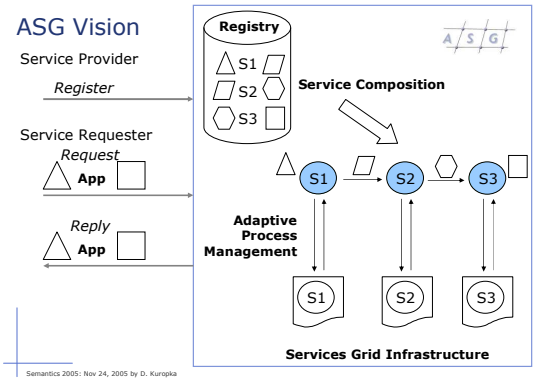
- Automated service composition by AI planning
  => Significant reduction of composition costs
  => Run-time adaptability and failure recovery



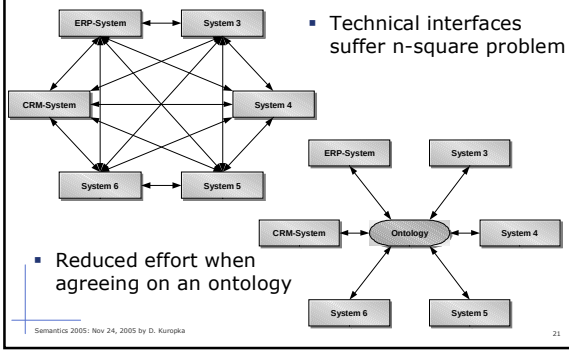
Adaptive Services Grid

- Prototype of open development platform for
  - Services specification, registration,
  - discovery, composition, negotiation, and
  - Enactment
- Services Grid Paradigm
- FP6 IST-2 IP

ASG Vision

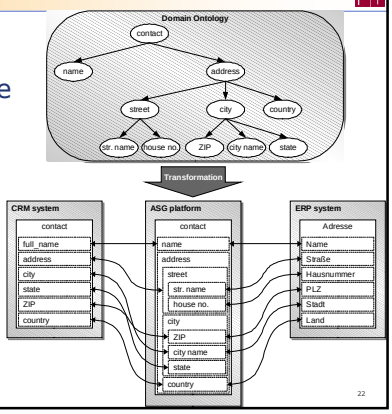


Challenge in connecting systems

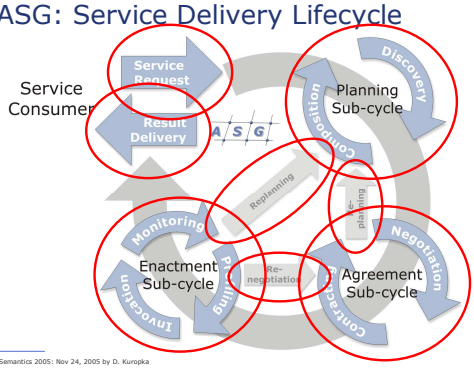


- Technical interfaces suffer n-square problem
- Reduced effort when agreeing on an ontology

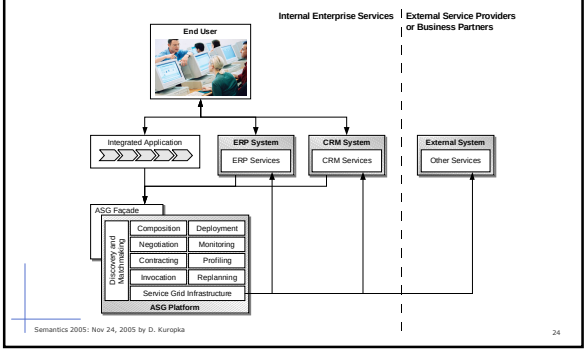
Grounding and data type mediation



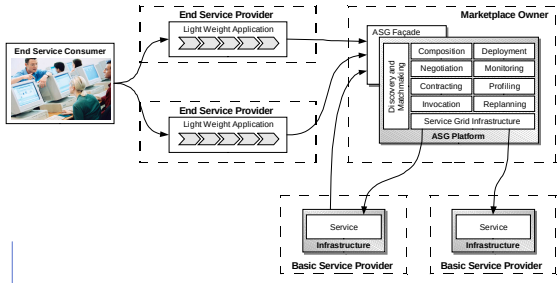
ASG: Service Delivery Lifecycle



ASG Integration Scenario



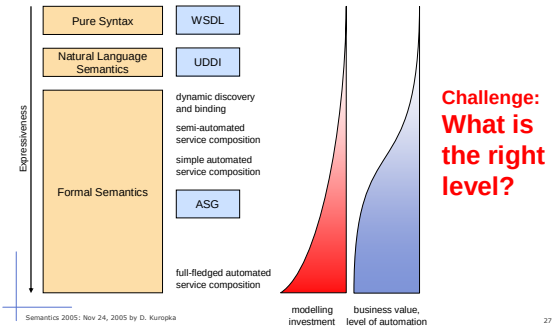
ASG Marketplace Scenario



Concepts and Challenges in Service Composition

1. Spirit of SOC
2. Service Composition: Adaptive Services Grid
3. Challenges and Conclusions

Expressiveness of Modelling Language



Challenge: What is the right level?

Challenges on Ontologies

- What is a proper level of granularity?
  - Fine granularity vs. coarse granularity
- How large should be the modelled domain?
  - Narrow domain model vs. broad domain model

=> need for Practical experiences

Conclusions

- Semantics enables exploration of SOC
  - by dynamic service discovery and binding
  - by automated service composition

=> reduces costs by rising efficiency of adaptation processes

- Need for tight cooperation between industry and science
  - to solve open challenges
  - to learn best practices for implementation