DAIMLERCHRYSLER

SOA: Choreography and Orchestration

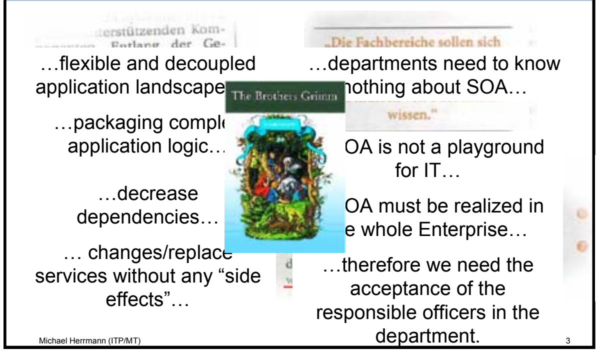
Why SOA lacks reuse

Michael Herrmann, Richard Golden

Agenda

- Service Oriented Architecture (SOA)
- Semantics
- Agility and Reuse
- Cross Platform Scenario (planned)
- CO-Layer (Choreography & Orchestration)
  - Ontology Web Language (OWL)
  - Ontology Web Language for Services (OWL-S)
  - Reasoning Example

Service Oriented Architecture (SOA)?!



…flexible and decoupled application landscape

…packaging compl application logic…

…decrease dependencies…

… changes/replace services without any “side effects”…

…departments need to know nothing about SOA…

OA is not a playground for IT…

OA must be realized in e whole Enterprise…

…therefore we need the acceptance of the responsible officers in the department.

Some definitions…

„A SOA is a set of components which can be invoked,
and whose
interface descriptions can be published and discovered.”

[W3C]

„SOA is a software architecture that builds
a topology of interfaces,
interface implementations and interface calls.
SOA is a relationship of services and service consumers,
both software modules large enough
to represent a complete business function.
So, SOA is about reuse, encapsulation, interfaces,
and ultimately, agility.”

[Gartner Group]

Definition at DaimlerChrysler (EMEA)

Service Oriented Architecture (SOA):

“Technology neutral architectural concept based on generally (re-)usable services”

[Fritz, Quast, Rothländer (ITI/T),
Dr. Flehmig, Dr. Melzer, (REI/ID), Groß, Herrmann (ITP/MT)]

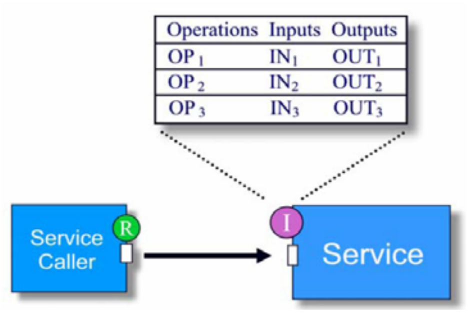
Our comprehension

“SOA is a technology neutral architectural concept
based on generally (re-)usable services.”

„SOA is a concept of software architecture
that represents one or more *business functions* as a *service*.
The *interface* of a service is *platform independent*.
The implementations of the services are *reusable*,
encapsulated and *loosely coupled*.
The service *interactions* are realized by
a *standardized/uniformed* infrastructure.”

ESA = SOA + ES

Services inside SOA



Operations	Inputs	Outputs
OP_1	IN_1	OUT_1
OP_2	IN_2	OUT_2
OP_3	IN_3	OUT_3

Where is the benefit?

	Tight Coupling	Loose Coupling
Physical	point-to-point	intermediate
Comm. style	synchronous	asynchronous
Type system	strong	weak
Control of process logic	central control	distributed control
Binding	statically	dynamically
Transactionality	2PC (2-phase-commit)	compensation

[Krafzig]

Loose sounds simpler,
but is harder to implement and increases complexity

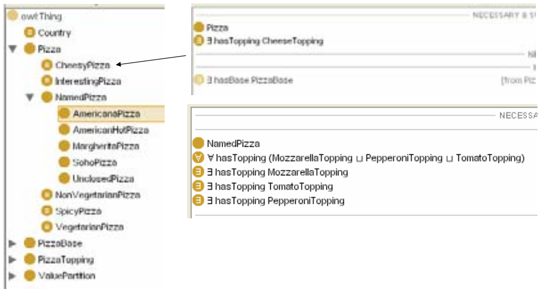
Reaching SOA

- State „SOA reached“ vs. „Goals of SOA reached“
- Create an adequate Organization (SOO)
- Support SOA by governance and guidance
- Constraints derived from definition
  - encapsulated (services from each other)
  - Loose coupling (see previous slides)
  - Independent of technical implementation
  - Interaction via standardized/unified infrastructure (Infrastructure Layer + ESB)
  - Represent process steps as services
  - reuse (technically vs. semantically)
- Issue: distribution in your Enterprise
- Annotation: SOA should also have a methodology (Roles, software model, techniques in modeling, etc.) supported / attended by business process management

What are semantics?

- Semantics are the study of meaning
- Knowledge can be described by one or more ontologies
- An Ontology is a definition of concepts and relations
- Description Logic (DL) is a cornerstone of the design of ontologies
  - Non-finiteness
  - open world assumption
  - Inheritance
  - Automatic classification

Ontology (example)



- owl:Thing
  - Country
  - Pizza
    - CheesyPizza
    - InterestingPizza
    - NamedPizza
      - AmericanaPizza
      - AmericanHotPizza
      - MargheritaPizza
      - SohoPizza
      - UnclosedPizza
    - NonVegetarianPizza
    - SpicyPizza
    - VegetarianPizza
  - PizzaBase
  - PizzaTopping
  - ValuePartition

NECESSARY & S

Pizza

∃ hasTopping CheeseTopping

∃ hasBase PizzaBase [from Piz

NECESSA

NamedPizza

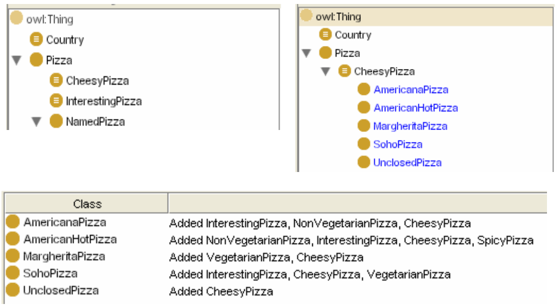
∀ hasTopping (MozzarellaTopping ⊔ PepperoniTopping ⊔ TomatoTopping)

∃ hasTopping MozzarellaTopping

∃ hasTopping TomatoTopping

∃ hasTopping PepperoniTopping

Reasoning (example)

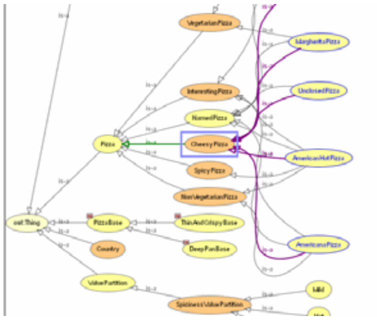


- owl:Thing
  - Country
  - Pizza
    - CheesyPizza
    - InterestingPizza
    - NamedPizza

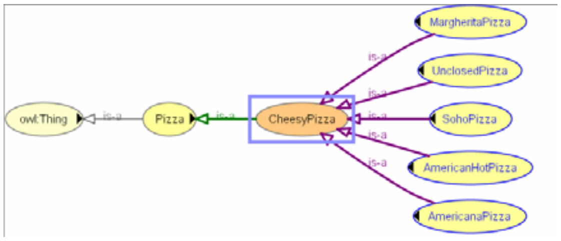
- owl:Thing
  - Country
  - Pizza
    - CheesyPizza
      - AmericanaPizza
      - AmericanHotPizza
      - MargheritaPizza
      - SohoPizza
      - UnclosedPizza

Class	
AmericanaPizza	Added InterestingPizza, NonVegetarianPizza, CheesyPizza
AmericanHotPizza	Added NonVegetarianPizza, InterestingPizza, CheesyPizza, SpicyPizza
MargheritaPizza	Added VegetarianPizza, CheesyPizza
SohoPizza	Added InterestingPizza, CheesyPizza, VegetarianPizza
UnclosedPizza	Added CheesyPizza

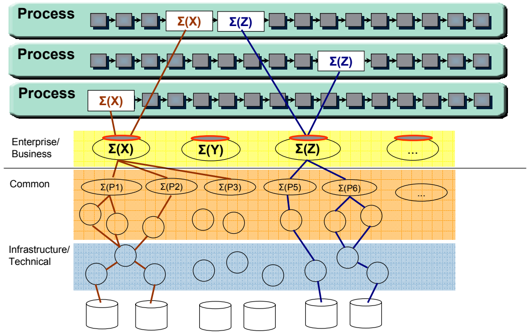
Reasoning (example)



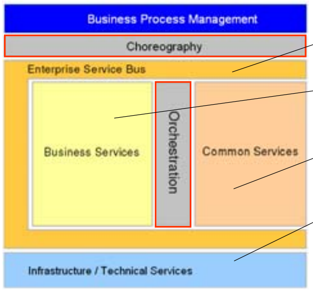
Reasoning (example)



Agility and Reuse



Architecture (example)



The Enterprise Service Bus (ESB) is a part of the infrastructure and regulates e.g. communication

An “Enterprise Service” maps professional functions and reflects a part of a business process directly

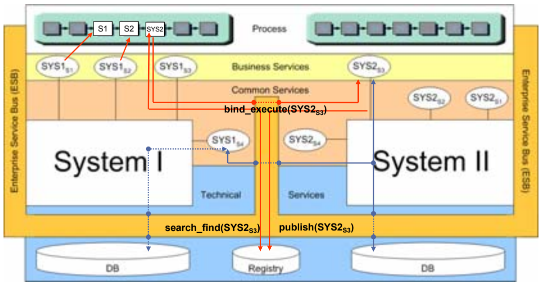
This infrastructure consists of several levels of different granularity and is implemented by services itself

Supports Hardware, OS, Network, DB, etc.

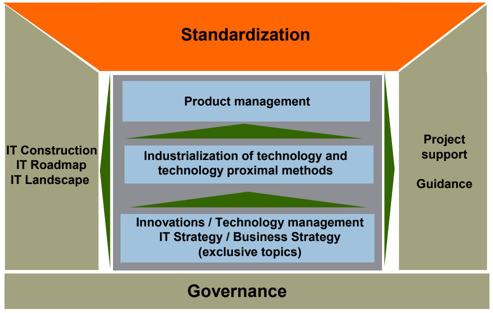
Enterprise Service Bus (ESB)

- Transport and content-based routing (Brokerage: Publish-Subscribe)
- Multi-protocol communication
- Validation and transformation (messages)
- Supports
  - Event-handling
  - Security
  - Monitoring
  - Adapter-connectivity
  - Dynamic addressing and service binding (Registry)

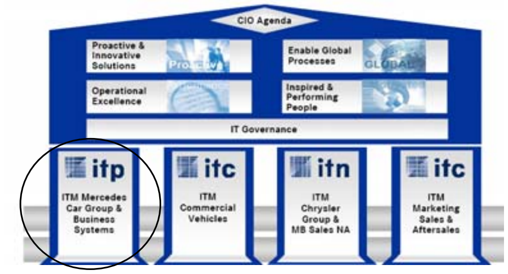
Scenario consumer / provider



“Introducing” my working area: IT:Factory

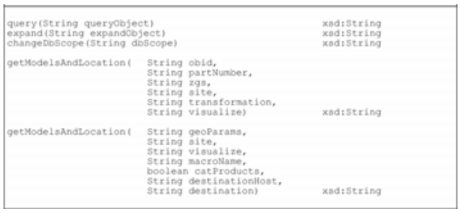

Standardization
Product management
Industrialization of technology and technology proximal methods
Innovations / Technology management IT Strategy / Business Strategy (exclusive topics)
IT Construction
IT Roadmap
IT Landscape
Project support
Guidance
Governance

“Introducing” my working area


CIO Agenda
Proactive & Innovative Solutions
Proactive
Enable Global Processes
GLOBAL
Operational Excellence
Inspired & Performing People
IT Governance
itp
ITM Mercedes Car Group & Business Systems
itc
ITM Commercial Vehicles
itn
ITM Chrysler Group & MB Sales NA
itc
ITM Marketing Sales & Aftersales

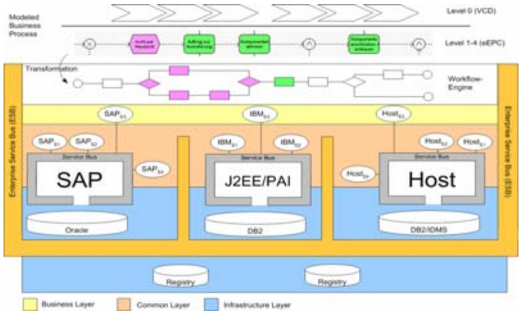
ITP/MT (Information Technology Passenger Cars / Methodology & Technology)

Real-Life example

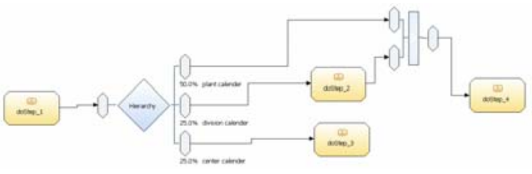


```
query(String queryObject)                              xsd:String
expand(String expandObject)                            xsd:String
changeDbScope(String dbScope)                          xsd:String

getModelsAndLocation(   String obid,
                        String partNumber,
                        String zgs,
                        String site,
                        String transformation,
                        String visualize)              xsd:String

getModelsAndLocation(   String geoParams,
                        String site,
                        String visualize,
                        String macroName,
                        boolean catProducts,
                        String destinationHost,
                        String destination)            xsd:String
```

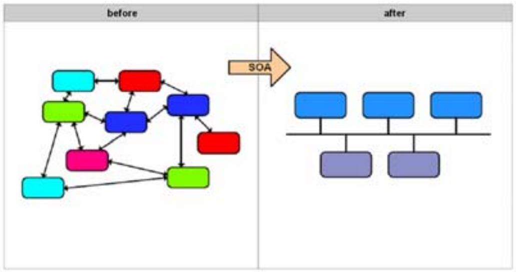
Scenario Cross Platform (planned)



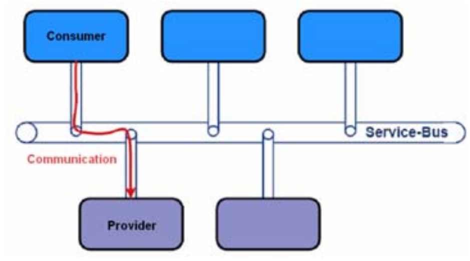
The modeled Process



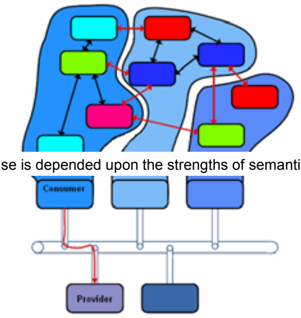
SOA & ESB



SOA & ESB



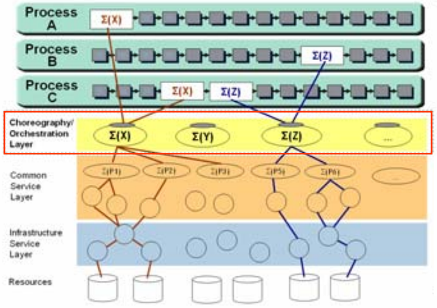
SOA & ESB & Semantics?!


…reuse is depended upon the strengths of semantics…

SOA Stages (outlined)

- Static binding to static Service (Point-to-Point Integration)
  - Consumer bind Provider
  - Wrap Legacy Systems
  - Security (identify and access management) / Quality of Service (QoS)
- Dynamic binding to static Service (Increase Visibility & Control)
  - Orchestration / Choreography
  - Service intermediary (Reg.) / Consumer (find, bind) / Provider (publish)
  - Governance & Policy / Service Contract
  - Management (Life-Cycle, System, Business, Meta data)
- Dynamic binding to dynamic Service (Improve Agility & Reuse)
  - Meta model (Business Model, Service Model (in-)dependent of tech.)
  - Semantic Integration (Taxonomy, Ontology, Business Semantic)
  - B2B (Federation, Dynamic Contract, …)

CO-Layer



Semantics in the CO-Layer (Approaches)

- AI Planning
- Semantic Discovery Service (SDS)
- WSDL-S
- OWL
- OWL-S
- etc.

OWL

- Ontology Web Language
  - W3C Standard
  - Semantically Markup Language to describe and split Ontologies
- Goals
  - Build relationships between concepts
  - Create machine readable descriptions and correlations
- OWL Lite (subset of the OWL language constructs)
- OWL Full and OWL DL support the same set of OWL language constructs.
  - OWL DL (has some constraints)
  - OWL Full (not guaranteed terminated)

DaimlerChrysler

OWL

- Subset of First Order Logics used to describe objects in a domain
- Allows three types of objects
  - **Concepts**: describe general concepts of things in the domain
    - Car: the generic description of cars
      (or is common to all the things that are cars)
  - **Individuals**: an object in the domain
    - SLK the car (“sporty, light, compact” – this is no official acronym)
  - **Properties**: relations between concepts
    - Bird `subclassOf` Animal
    - Car `has` Wheel

DaimlerChrysler

The power of OWL

- Different types of properties
  - Transitivity, Symmetry, Function, Inverse, etc.
    - A `siblingOf` B and B `siblingOf` C then A `siblingOf` C (Transitive)
    - A `siblingOf` B then B `siblingOf` A (Inverse)
- Cardinality restrictions
  - Specify how many elements are in relation with each other
  - at-most, at-least, exactly, optionally (0 or more)
    - Car `has` (exactly) 4 Wheel
- Allows concepts that have been defined in different ontologies to be equated (ont1:Cars `equvalentClass` ont2:PassengerCars)
- etc.

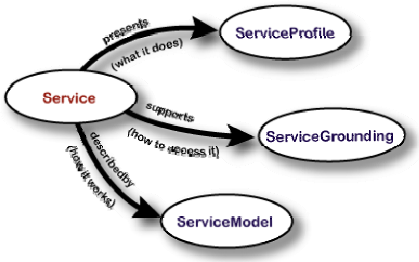
DaimlerChrysler

OWL-S

- Ontology Web Language for Services
- Machine readable description of Web Services
- An Ontology consists of basic classes, properties and declarations in order to describe a service
- Motivation
  - Agents are able to find and invoke Web Services
  - You can advertise for properties and abilities / skills of a Web Service
  - An automatically orchestration of Web Services probably come true
  - etc.

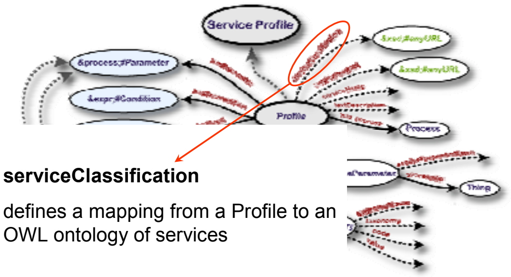
DaimlerChrysler

OWL-S



DaimlerChrysler

OWL-S Profile

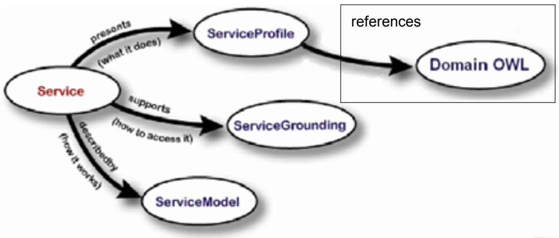


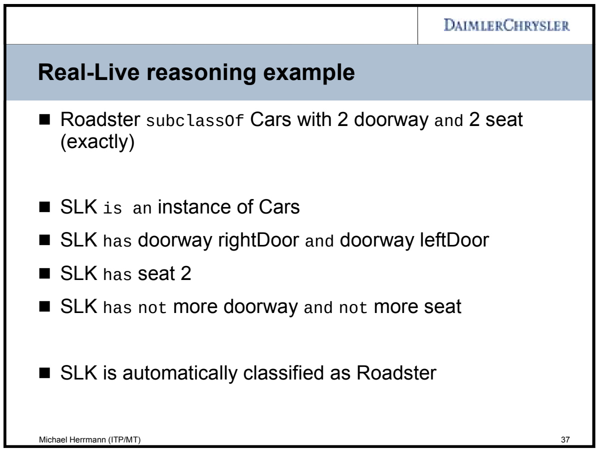
serviceClassification

defines a mapping from a Profile to an OWL ontology of services

DaimlerChrysler

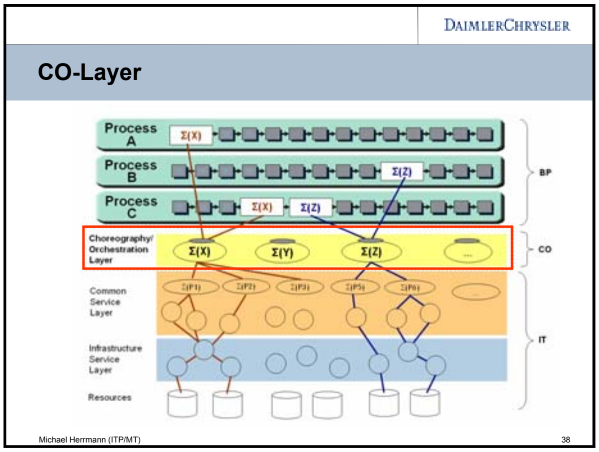
OWL-S



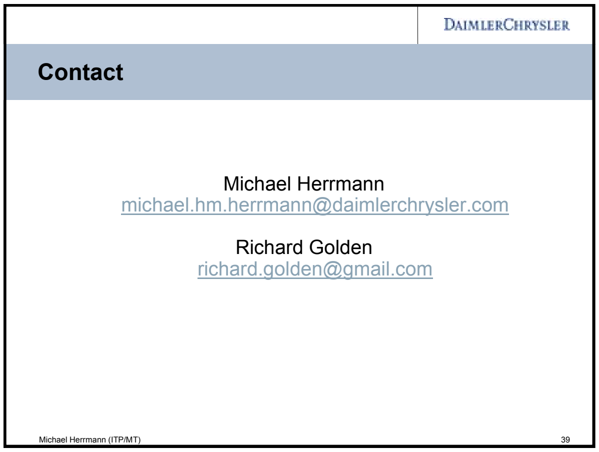


Real-Live reasoning example

- Roadster subclassOf Cars with 2 doorway and 2 seat (exactly)
- SLK is an instance of Cars
- SLK has doorway rightDoor and doorway leftDoor
- SLK has seat 2
- SLK has not more doorway and not more seat
- SLK is automatically classified as Roadster



CO-Layer



Contact

Michael Herrmann
michael.hm.herrmann@daimlerchrysler.com

Richard Golden
richard.golden@gmail.com