Quality Composition in Web-Service Design

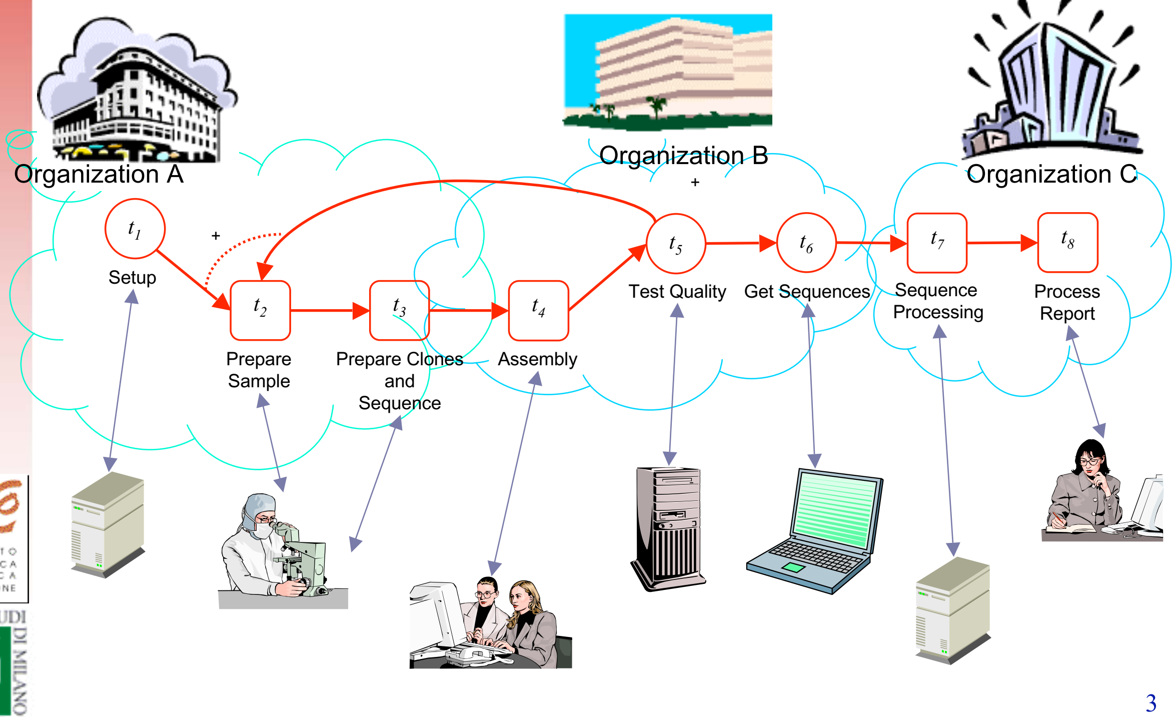
M. Comerio, *F. De Paoli*, S. Grega

SIUMI'06 – Jul 4, 2006

Contents

- Motivations & Scenarios
- Role of QoS in service (re)design
- Milestones of the methodology
- Quality evaluation by examples
- Conclusions & Current Work

Motivations & Scenarios



Motivations & Scenarios

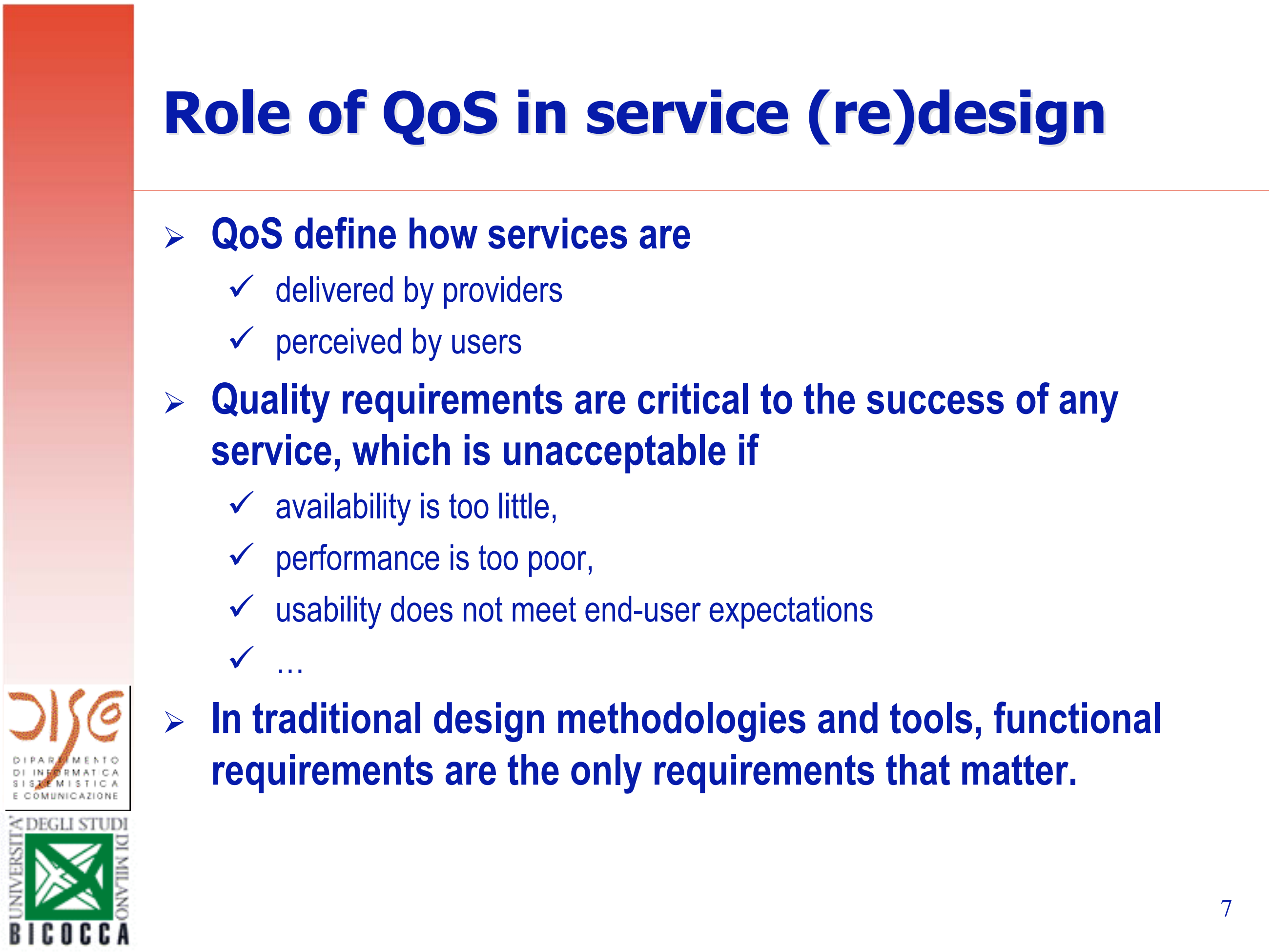
- **Business goals**
  - ✓ Fulfill business process requirements
    - User needs are first-class priorities
    - Supply functionalities with QoS
    - Support negotiation
- **Technological goals**
  - ✓ Adapt to ubiquitous and mobile requirements
    - Multi-channel adaptation
    - Multi-agreement support

Motivation & Scenarios

- QoS is a central issue
  - ✓ Technological constraints & opportunities
  - ✓ User requirements & preferences
  - ✓ Domain characteristics
- Mulit-facet problem
  - ✓ Infrastructures are required to support quality of service and monitoring
  - ✓ Component models should address quality issues
- Approach
  - ✓ Service Oriented Architecture (SOA)
  - ✓ Services are requested to publish their quality profile
  - ✓ Ontologies can help in the quality management

Contents

- Motivations & Scenarios
- Role of QoS in service (re)design
- Milestones of the methodology
- Quality evaluation by examples
- Conclusions & Current Work



Role of QoS in service (re)design

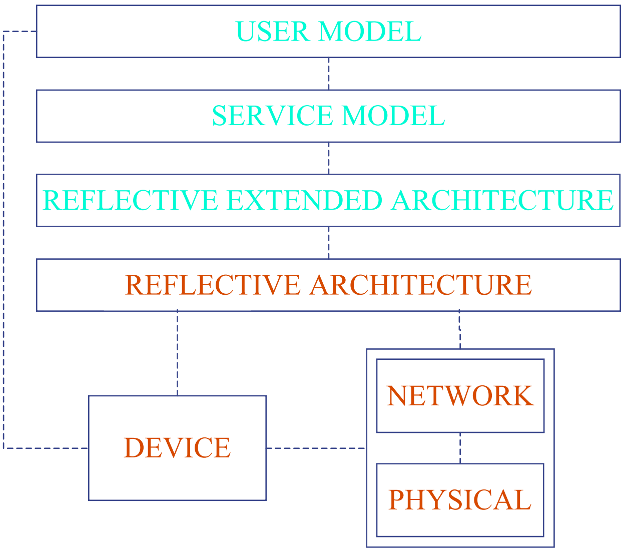
- QoS define how services are
  - ✓ delivered by providers
  - ✓ perceived by users
- Quality requirements are critical to the success of any service, which is unacceptable if
  - ✓ availability is too little,
  - ✓ performance is too poor,
  - ✓ usability does not meet end-user expectations
  - ✓ …
- In traditional design methodologies and tools, functional requirements are the only requirements that matter.

The proposal

- Define a quality model that addresses
  - ✓ quality dimensions (metrics and values)
  - ✓ relations among qualities (dependencies and aggregations)
  - ✓ composition rules (same and different qualities)
- Take qualities into account during service life-cycle
  - ✓ Identification of requirements
  - ✓ Service development
  - ✓ Service provisioning
  - ✓ Service monitoring
  - ✓ Service maintenance
- We concentrate on service design to deal with
  - ✓ Quality descriptions
  - ✓ Quality composition

Quality model

- **A quality is defined by**
  - ✓ definition,
  - ✓ metrics,
  - ✓ method of measure,
  - ✓ range of values
- **Quality model**
  - ✓ Qualities at each level have been classified
  - ✓ Intra-level and inter-level dependencies have been identified
- **High level QoS allows for platform independency**
- **Low level QoS are related to the actual architecture**



USER MODEL

SERVICE MODEL

REFLECTIVE EXTENDED ARCHITECTURE

REFLECTIVE ARCHITECTURE

DEVICE

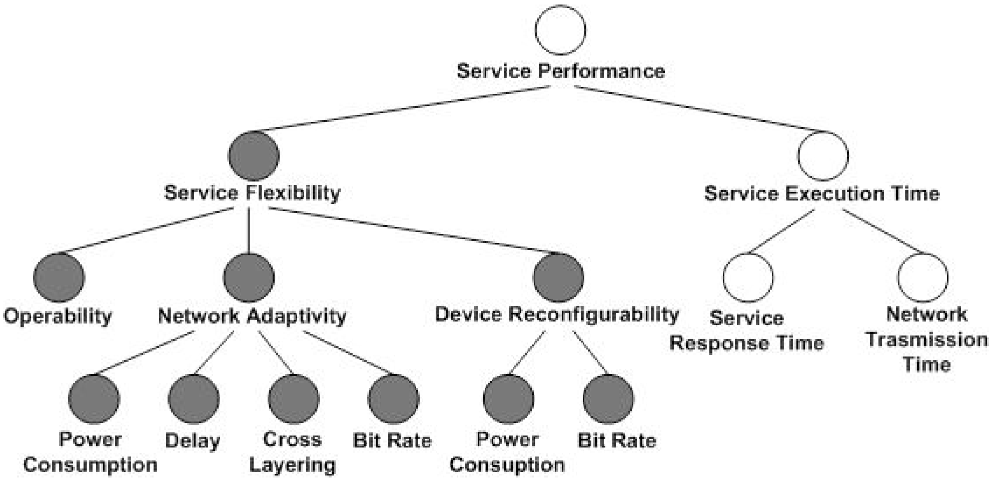
NETWORK

PHYSICAL

MAIS dependency model

An example

- **An example of QoS dependency tree**



321 QoS dimensions, 203 dependencies, explicit formulation in 8% of total cases, while 50% can be evaluated by running simulations

Contents

- Motivations & Scenarios
- Role of QoS in service (re)design
- Milestones of the methodology
- Quality evaluation by examples
- Conclusions & Current Work

Milestones of the methodology

- Input
  - ✓ Business goals
  - ✓ Technological opportunities
  - ✓ Quality model (ontologies)
- Abstract vs. concrete design
  - ✓ Capturing QoS with platform independent design
  - ✓ Design validation w.r.t. actual platforms
- Output
  - ✓ UML diagrams that includes QoS properties

The phases of WSMoD

Service Identification

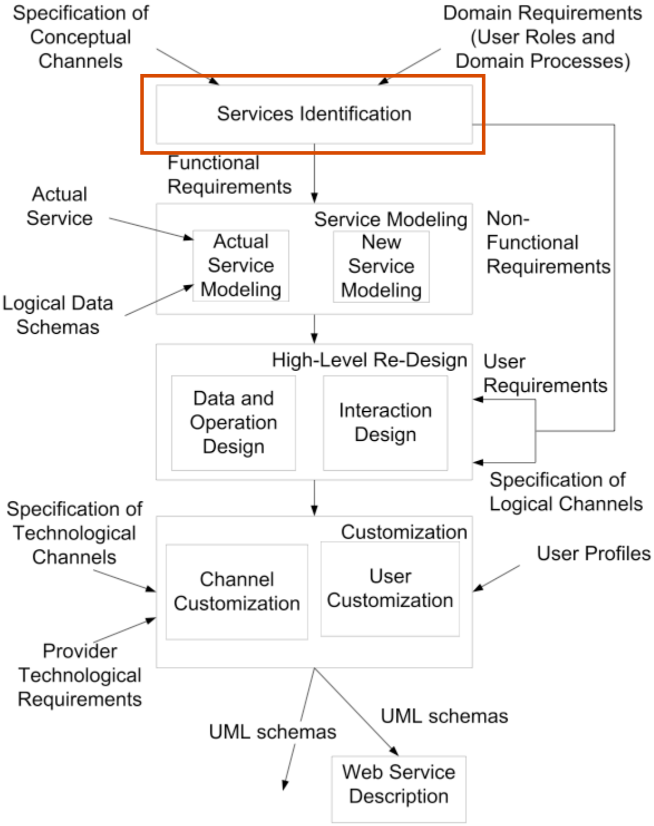
delivers the service requirements

Focus on

- ✓ business processes
- ✓ technological opportunities

Tools

- ✓ Requirement Engineering



Specification of Conceptual Channels

Domain Requirements (User Roles and Domain Processes)

Services Identification

Functional Requirements

Actual Service

Service Modeling

Actual Service Modeling

New Service Modeling

Non-Functional Requirements

Logical Data Schemas

High-Level Re-Design

Data and Operation Design

Interaction Design

User Requirements

Specification of Logical Channels

Specification of Technological Channels

Customization

Channel Customization

User Customization

User Profiles

Provider Technological Requirements

UML schemas

UML schemas

Web Service Description

The phases of WSMoD

Service Modeling

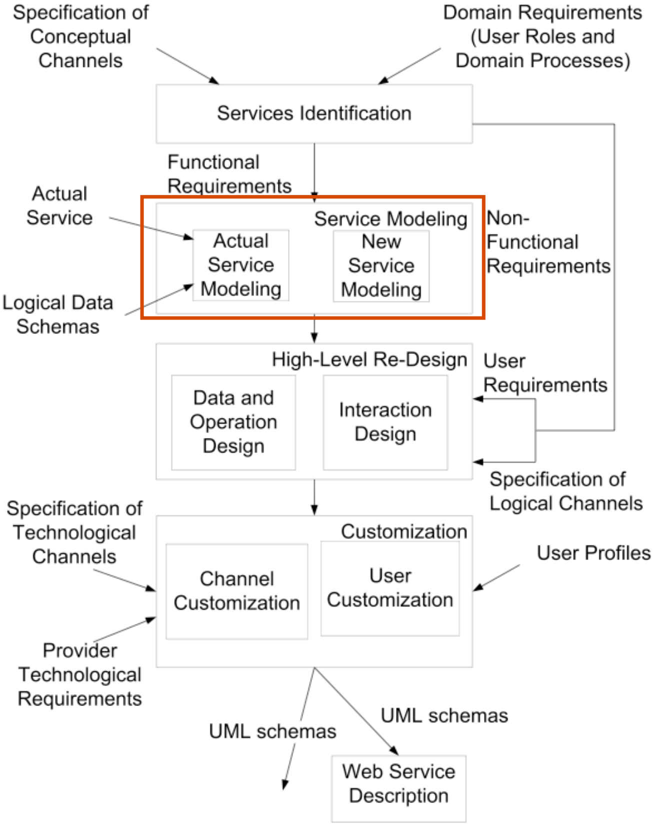
delivers the functional design of the service

Focus on

- ✓ service classification
- ✓ existing services

Tools

- ✓ Service ontologies
- ✓ Data modeling tools



Specification of Conceptual Channels

Domain Requirements (User Roles and Domain Processes)

Services Identification

Functional Requirements

Actual Service

Service Modeling

Actual Service Modeling

New Service Modeling

Non-Functional Requirements

Logical Data Schemas

High-Level Re-Design

Data and Operation Design

Interaction Design

User Requirements

Specification of Logical Channels

Specification of Technological Channels

Customization

Channel Customization

User Customization

User Profiles

Provider Technological Requirements

UML schemas

UML schemas

Web Service Description

The phases of WSMoD

High Level Re-Design

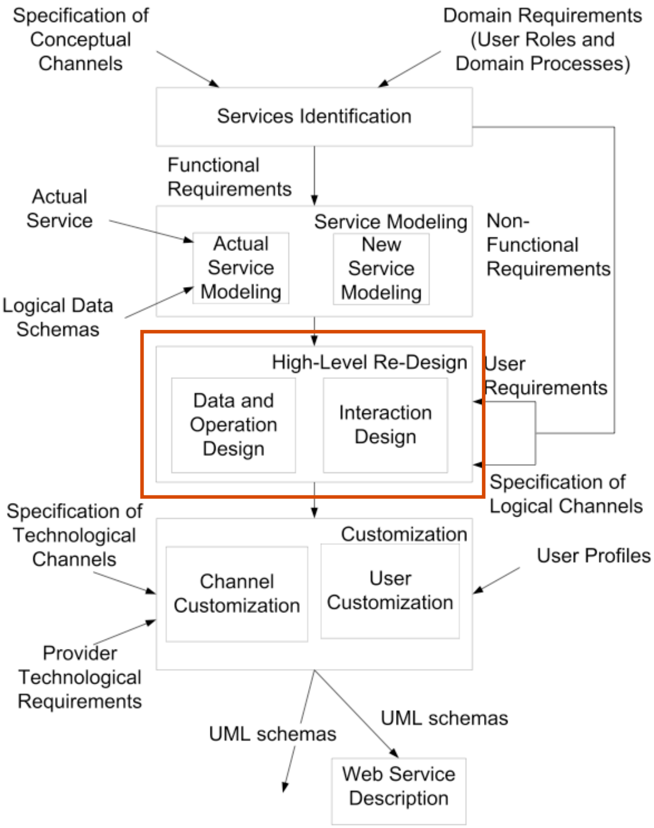
delivers the complete design of the service

Focus on

- ✓ non-functional requirements
- ✓ identification of qualities and their relations

Tools

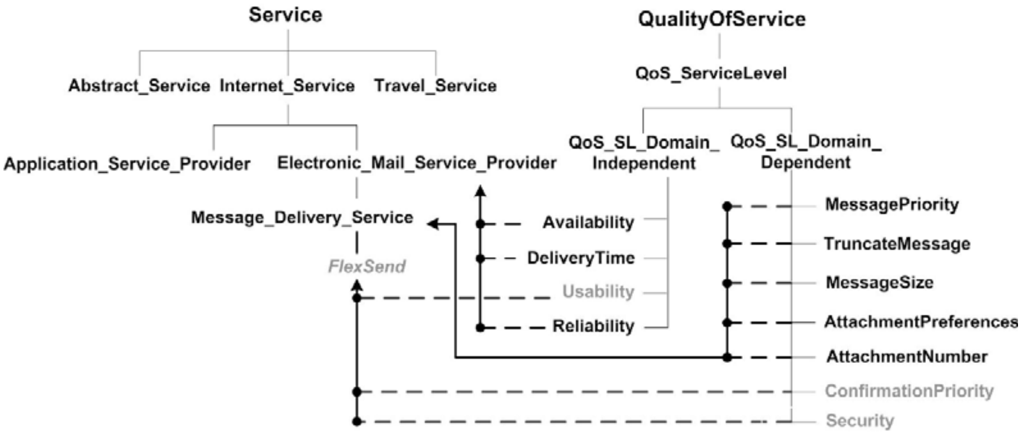
- ✓ Quality ontologies
- ✓ User ontologies
- ✓ Channel ontologies



An example

- ➢ **Abstract service:** *Message_Delivery_Service*
- ➢ **Concrete service:** *FlexSend*

… and realted qualities



The phases of WSMoD

Customization

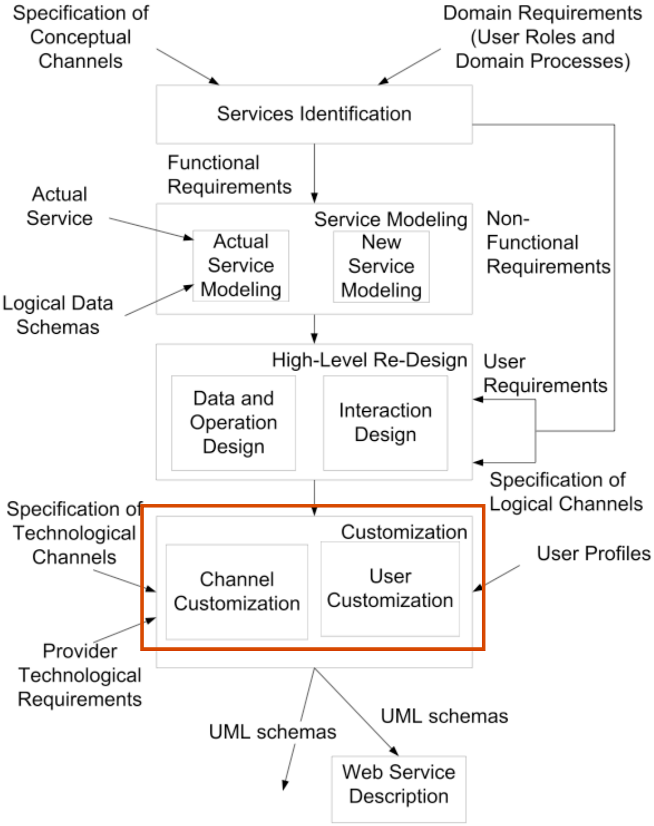
- ➢ **delivers the finalized design of the service**

Focus on

- ✓ actual profiles of stakeholders, users, services

Tools

- ✓ Quality composition rules
- ✓ Operational research tools


Specification of Conceptual Channels

Domain Requirements (User Roles and Domain Processes)

Services Identification

Functional Requirements

Actual Service

Service Modeling

Actual Service Modeling

New Service Modeling

Non-Functional Requirements

Logical Data Schemas

High-Level Re-Design

Data and Operation Design

Interaction Design

User Requirements

Specification of Logical Channels

Specification of Technological Channels

Customization

Channel Customization

User Customization

User Profiles

Provider Technological Requirements

UML schemas

UML schemas

Web Service Description

The phases of WSMoD

Web Service Description

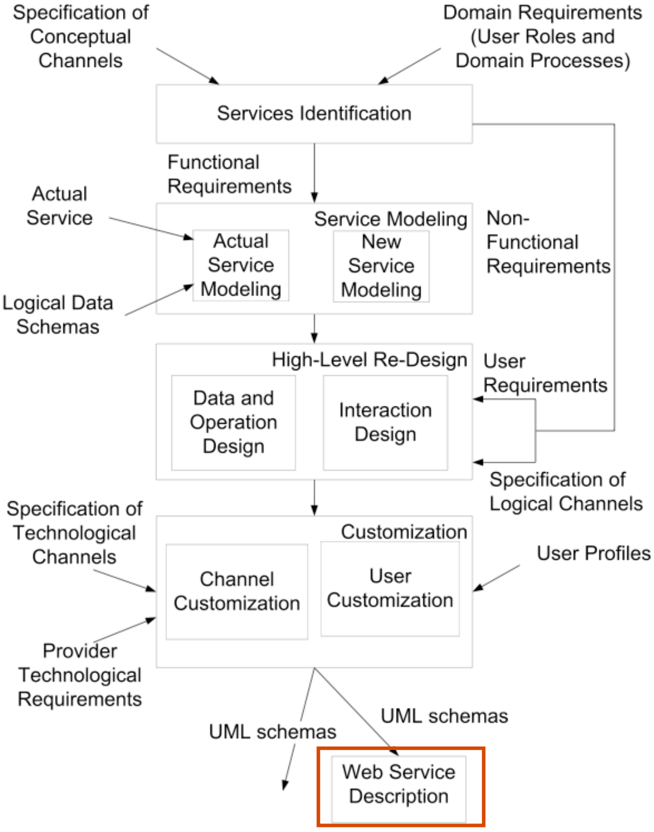
delivers the standard descrition of the service

Focus on

- ✓ Standards

Tools

- ✓ WSDL
- ✓ BPEL
- ✓ WSOL (?)


Specification of Conceptual Channels

Domain Requirements (User Roles and Domain Processes)

Services Identification

Functional Requirements

Actual Service

Service Modeling

Actual Service Modeling

New Service Modeling

Non-Functional Requirements

Logical Data Schemas

High-Level Re-Design

Data and Operation Design

Interaction Design

User Requirements

Specification of Logical Channels

Specification of Technological Channels

Customization

Channel Customization

User Customization

User Profiles

Provider Technological Requirements

UML schemas

UML schemas

Web Service Description

Contents

- ➢ Motivations & Scenarios
- ➢ Role of QoS in service (re)design
- ➢ Milestones of the methodology
- ➢ Quality evaluation by examples
- ➢ Conclusions & Current Work

The method by example

Case study:

the development of an Entertainment_Discovery (*EnDi*) service with the following features:

- ✓ searching for entertainment events (e.g., exhibitions, movies) by location;
- ✓ booking the event;
- ✓ searching for public transportation connections;

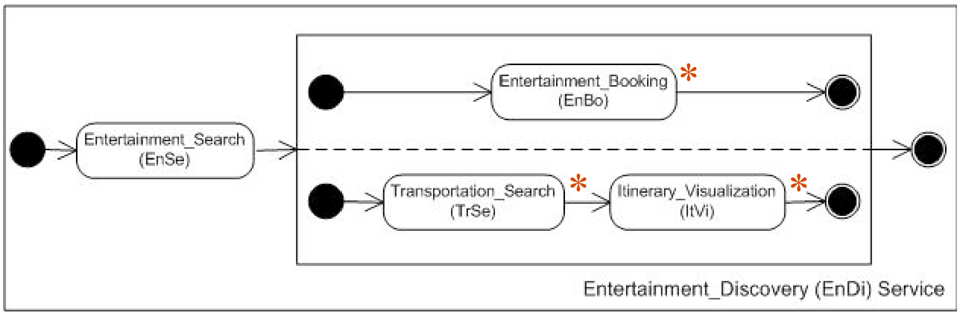
Expected qualities:

Service Flexibility (SF)	At least 0.7
Service Execution Time (SET)	At most 4 sec.
Service Operability (SO)	At least 0.6
Service Performance (SP)	At least 0.75

values are normalized between 0 and 1 or expressed in their metrics

The problem

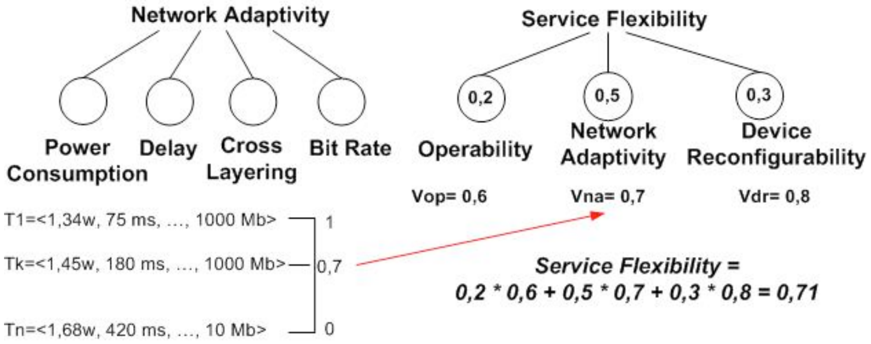
- The service modeling phase identify abstract services and a set of concrete services*


Entertainment_Search (EnSe)
Entertainment_Booking (EnBo)*
Transportation_Search (TrSe)*
Itinerary_Visualization (ItVi)*
Entertainment_Discovery (EnDi) Service

- How to manage the qualities of the involved services?
- Composition rules depends on
  - ✓ Type of the quality (e.g. time is additive, security is not)
  - ✓ Architectural pattern (sequence, parallel, iteration, branch)

Simple service design

- Customization phase
  - ✓ We need to evaluate the qualities of the simple service
- Hybrid method
  - ✓ Simple Additive Weight (SAW) techniques, to obtain a *score* from a set of dimensions having different units of measure
  - ✓ Expert evaluations, to obtain functions that give single values to sets of qualities



Composite service design

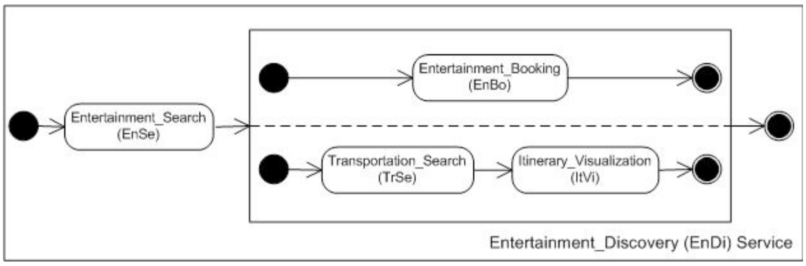
- Customization phase
  - ✓ We need to evaluate the qualities of the simple service
- The method depends on
  - ✓ Type of involved qualities
  - ✓ Architectural patterns
- Candidate functions

$$(Sum) \quad Q_{tot} = \sum_{i=i}^{n} v_i; \qquad (WAvg) \quad Q_{tot} = \sum_{i=1}^{n} w_i v_i;$$

$$(Min) \quad Q_{tot} = \min(v_1 ... v_n); \qquad (Max) \quad Q_{tot} = \max(v_1 ... v_n);$$

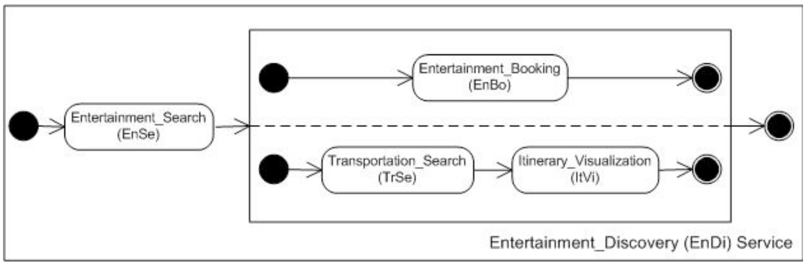
$$(Avg) \quad Q_{tot} = \frac{\sum_{i=1}^{n} v_i}{n};$$

The composition method


Entertainment_Search (EnSe)
Entertainment_Booking (EnBo)
Transportation_Search (TrSe)
Itinerary_Visualization (ItVi)
Entertainment_Discovery (EnDi) Service

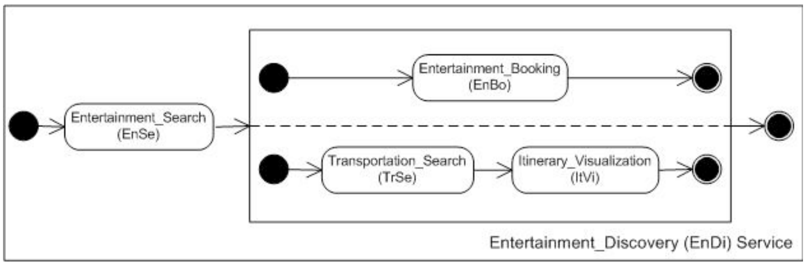
Architectural pattern Quality	**Parallel**	**Sequence**
Service Flexibility (SF)	**Avg**	**Avg**
Service Execution Time (SET)	**Max**	**Sum**
Service Operability (SO)	**WAvg**	**WAvg**
Service Performance (SP)	**Avg**	**Avg**

The problem


Entertainment_Search (EnSe)
Entertainment_Booking (EnBo)
Transportation_Search (TrSe)
Itinerary_Visualization (ItVi)
Entertainment_Discovery (EnDi) Service

Service Quality	EnSe	EnBo	TrSe	ItVi
Service Flexibility (SF)	0.71	0.8	0.8	?
Service Execution Time (SET)	1sec	3sec	1sec	?
Service Operability (SO)	0.6	-	0.7	-
Service Performance (SP)	0.7	0.6	0.8	?

The problem


Entertainment_Search (EnSe)
Entertainment_Booking (EnBo)
Transportation_Search (TrSe)
Itinerary_Visualization (ItVi)
Entertainment_Discovery (EnDi) Service

ItVi candidates Quality	WS 1	WS 2	WS 3
Service Flexibility (SF)	0.7	0.6	0.8
Service Execution Time (SET)	1,8 sec	2,5 sec	1,2 sec
Service Operability (SO)	-	-	-
Service Performance (SP)	0.68	0.5	0.9

Quality evaluation

- **The result with WS candidate 1 is**

SF_{TOT}	$= \mathrm{Avg}(SF_1,SF_2,SF_3,SF_4) = 0.75;$	***At least 0.7***
SET_{TOT}	$= \mathrm{Sum}(SET_1,\mathrm{Max}(SET_2,\mathrm{Sum}(SET_3,SET_4))) = 4;$	***At most 4 sec.***
SO_{TOT}	$= \mathit{WAvg}(0.8^*SO_1,0.2^*SO_3) = 0.62;$	***At least 0.6***
SP_{TOT}	$= \mathrm{Avg}(SP_1,SP_2,SP_3,SP_4) = 0.69.$	***At least 0.75***

- **Service Performance (SP) is not satisfying**
  - ✓ Search and select a different set of services (e.g., ItVi candidate 3)
  - ✓ Redesign the owned services (e.g., EnSe)
- **The service design/re-design can be modeled as an optimization problem:**
  - ✓ Identify the set of choices for the technical characteristics relevant for the end users and for the service requirements which minimizes design costs
- **This is a NP-hard linear integer programming problem**

The general formulation

$$
\begin{aligned}
&\min \sum_{i\in J}\sum_{j=1}^{n} c_{i,j}x_{i,j}\\
&\sum_{i\in J} x_{i,j} = 1; \forall i\\
&F(v_{1,j}x_{1,j}\ldots v_{n,j}x_{n,j}) \geq B\\
&x_{i,j} \in \{0,1\}
\end{aligned}
$$

I: set of technical choices
$v_{i,j}$: the quality value for alternative j
$c_{i,j}$: the cost associated with alternative j
$x_{i,j}$: the binary decision variable which is equal to 1 if the j alternative for the technical choice i is selected and 0 otherwise

- **The second formula guarantees that exactly one alternative for each simple service choice is selected;**
- **The third formula guarantees that the quality value provided by the solution is greater than the requirement B, hence the design assumptions is verified**

Contents

- Motivations & Scenarios
- Role of QoS in service (re)design
- Milestones of the methodology
- Quality evaluation by examples
- Conclusions & Current Work

Conclusions

- Enhancement of traditional methodologies by addressing QoS explicitly
- E.g., MDA focuses on functional requirements
  - ✓ Platform Independent Model (PIM)
  - ✓ Platform Specific Model (PSM)
- Proposed methodology
  - ✓ Enhance PIM by including QoS

Conclusions

- Toward open models to support business processes
  - ✓ QoS requirements address business process requirements
  - ✓ Accurate definition of QoS (and features) enables automatic integration in business processes
  - ✓ Negotiable QoS support different contracts

Conclusions

- Toward ubiquitous and pervasive computing
- Back-end services need to
  - ✓ adapt to different scenarios
  - ✓ support seamless integration
- The proposed methodology capture these aspects
  - ✓ Specific design to fulfill channel & user issues

Current Work

- Extension to Run-time negotiation support
  - ✓ Definition of user-side and provider-side terms of contract
  - ✓ Identification of candidate concrete processes (i.e. composite services) to be selected at run time
- Quantitative evaluation of qualities
  - ✓ Refinement of the validation/revision phase
  - ✓ Criteria and techniques to evaluate QoS (e.g., Multiple Criteria Decision Making)
- Development of a tool to support
  - ✓ Ontology management (including visual browsing and searching)
  - ✓ Integration with UML design tools

Thank you!
Questions?

SIUMI'06 – Jul 4, 2006