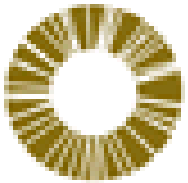
Dealing with Quality Tradeoffs during Service Selection



Caroline Herssens – University of Louvain

Ivan Jureta - University of Namur

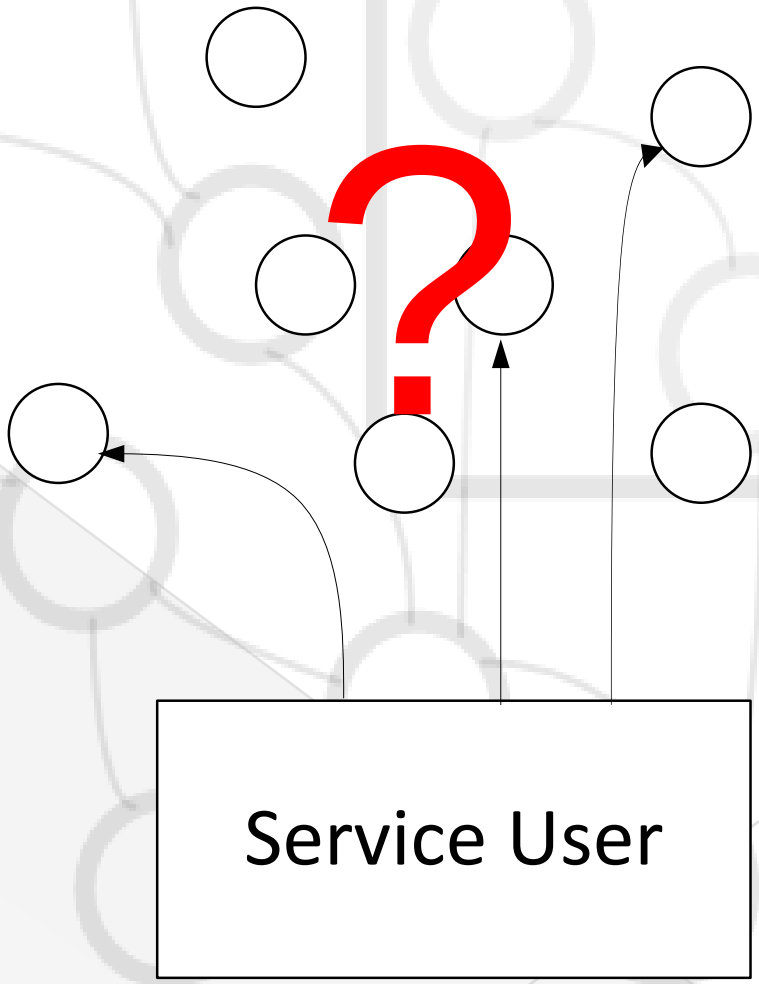
Stéphane Faulkner – University of Namur

Presentation Outline

- Motivations
- Service Oriented Computing (SOC)
- Quality of Service (QoS)
- QoS Modeling
- QoS Based Selection of services
- Conclusions

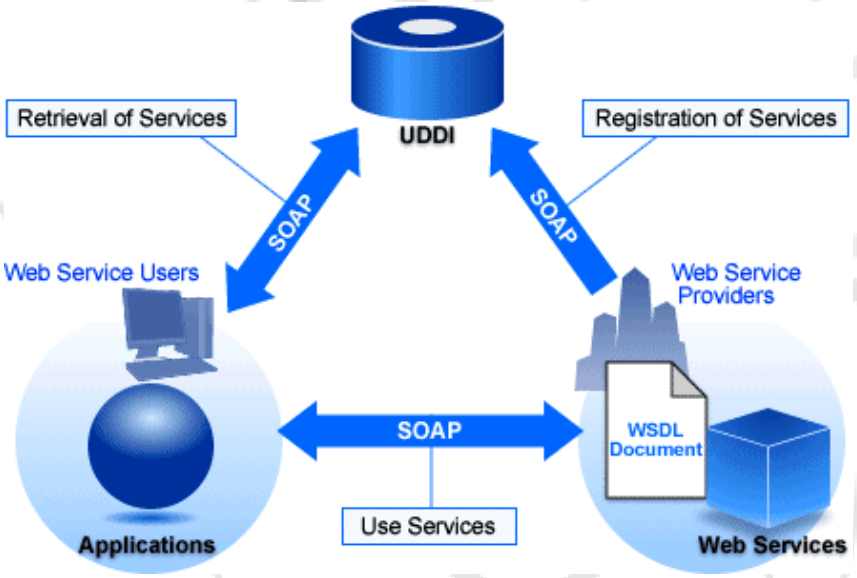
Motivations

- Multiple services available to perform a same task
- Automatic selection of services
  - Based on service user expectations
- Dealing with conflicting expectations of service users
  - Users want to maximize different criteria
- Enable autonomic computing based on services
  - Self optimization



SOC

- SOC = Service Oriented Computing
  - A *service* is a self-describing and self-contained modular application designed to execute a well-delimited task, and that can be described, published, located, and invoked over a network [Papazoglou, Georgakopoulos]
- Web service = service made available on the World Wide Web
- Utilization standards (UDDI, SOAP, WSDL)
- Examples : Information retrieval, Flight reservation,etc.

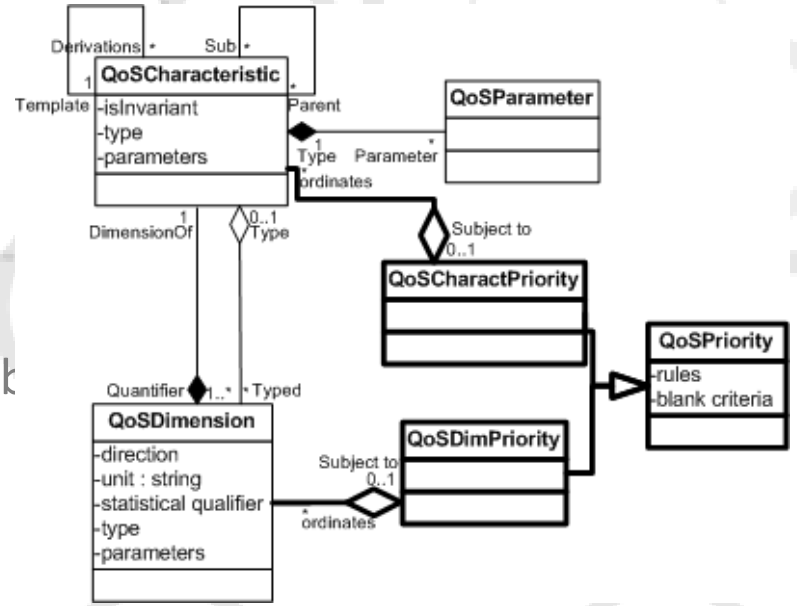


Quality of Service

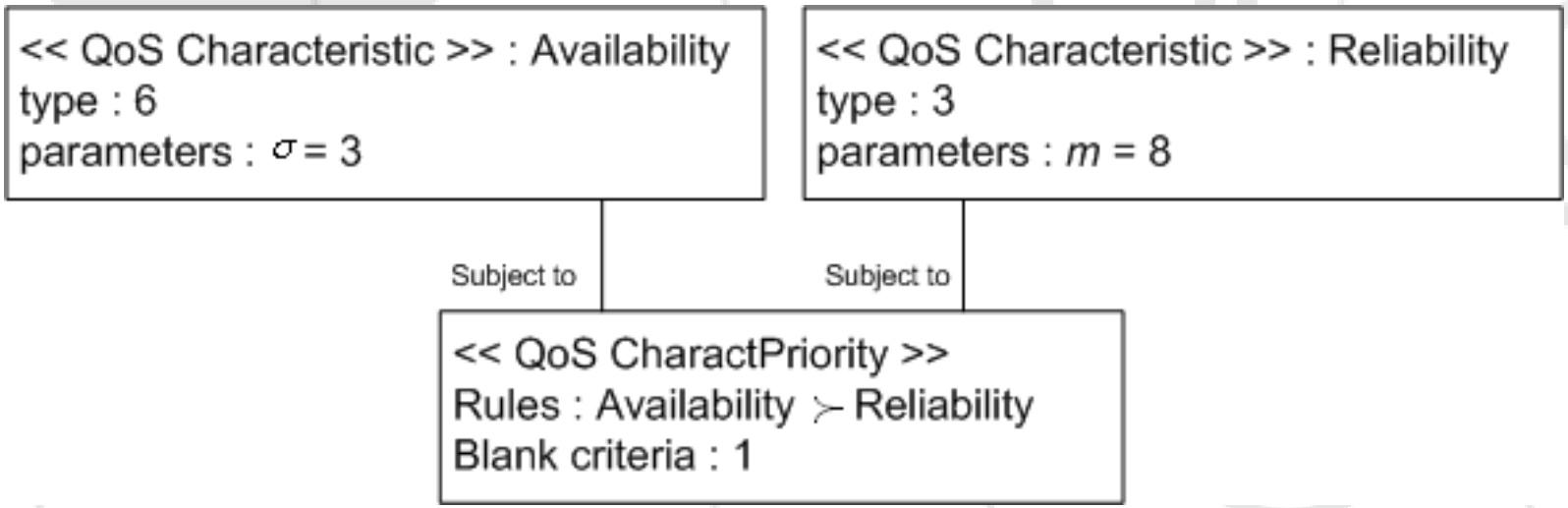
- QoS = Quality of Service
- Non Functional characteristics of a service
  - Availability, reliability, latency, security, cost, reputation, etc.
  - Cannot be directly implemented
    - Vs functional requirements
  - Step in configuration and implementation choices
  - Specified by
    - Provider - advertisement
    - User - requirement

QoS Modeling

- Allow a clear specification of QoS
  - QoS specifications can be used at various stages
    - Selection,composition, etc.
  - Q-WSDL, WSLA, DAML-QoS, etc.
- UML QoS Profile [OMG06]
  - Metamodel
  - QoS Characteristics are quantified b QoS Dimensions
- Extensions
  - Priorities
    - Availability favored to Reliability



QoS Modeling: Example



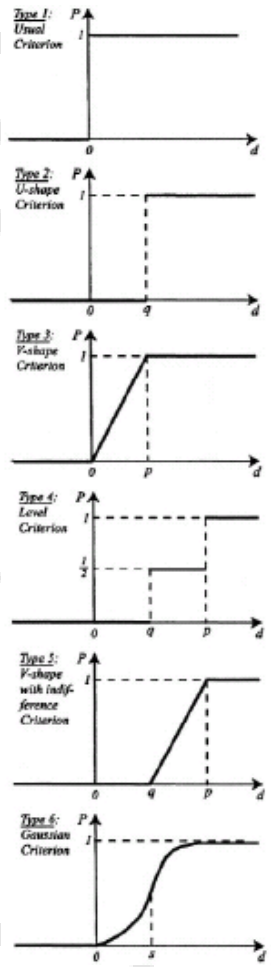
QoS Based Selection of Services

- Multiple QoS dimensions to optimize simulteanously
- Multi Criteria Decision Making (MCDM) Selection
  - Aggregation of multiple criteria
    - Criteria = QoS specified with QoS model
- Selection steps
  - User preferences definition → preference type
  - User priorities specification → weights of QoS Dimensions
  - MCDM ranking: outranking method

User preferences (1/2)

- User Preference definition states the type of preference of each QoS property
  - Preferences are defined on values of QoS Dimensions or Characteristics
  - Preferences type and optional parameters are direct input of outranking methods
    - Electre Methods
    - Promethee methods
      - Support more preference types

User preferences (2/2)



Immediate preference

service A with reliability of 95 % is preferred to service B with reliability of 94%

Preference threshold – 2%

service A with reliability of 95% is favored to service B with reliability < 93 %

Continuous increasing until indifference threshold

Indifference and preference thresholds

Continuous increasing between indifference and preference thresholds

Gaussian law with a fixed standard deviation

User preferences: example

Table 3. Within criteria information

Criteria	Unit	Type	Direction	Parameter	Parameter's value
Availability	%	Criterion with Linear preference	increasing	m	8
Latency	ms	Quasi-Criterion	decreasing	l	45
Reliability	%	Gaussian Criterion	increasing	σ	3
Reputation	%	Criterion with Linear Preference	increasing	m	4
Security	Level (1 to 10)	Usual Criterion	increasing	-	-

User Priorites

- Priorities used to fix weights
- Priorities can be defined over distinct characteristics or dimensions
- $d_i \succ d_j$
  Improving the value of d_i is more important than improving the value of d_j
- Weights are used to express priorities on outranking methods
  - Non compensatory approach
    - Vs compensatory approach: weights as substitution rates
  - Represent the intrinsic relative importance of the attributes
  - Multiple existing methods
    - Simos, Salinen, Mousseau

Fixing weights

- Revised Simos procedure [Figueira02]
  - Rank criteria
    - Criteria with the relative same importance put on the same rank
  - Add eventual white cards
  - Importance between the last and the first criterion in the ranking (z parameter)
  - Figures to take into account after the decimal point (w parameter)
  - Normalization to allow for a uniform measurement of service qualities independently of unit specifics
  - Rounding-off optimized
    - Dysfunctions of relative error of rounding up (d_i) and down ($đ_i$) minimized
  - Weighting used in outranking methods

Fixing weights: example

$$\begin{cases} reputation \succ_c security \\ security \succ_c latency \\ latency =_c availability \\ availability \succ_c reliability \end{cases}$$

Table 1. Non-normalized weights for $z = 6$

Rank r	Criteria in the rank r	Number of white cards according to rank r, e'_r	e_r	Non-normalized weights $k(r)$	Total
1	reliability	0	1	1.00	1.00 * 1 = 1.00
2	security	0	1	2.25	2.25 * 1 = 2.25
3	availability, latency	1	2	3.50	3.50 * 2 = 7.00
4	reputation	...	...	6.00	6.00 * 1 = 6.00
sum	5	1	4	...	16.25

Table 2. Determining the normalized weights of each criterion for $w = 1$ and $z = 6$

Rank	Criteria	Normalized weights k_i^*	Normalized weights k_i''	Ratio d_i	Ratio $\bar{d}_i$	Normalized weights k_i
1	reliability	6.1538	6.1	0.00750755	0.00874256	6.2
2	security	13.8461	13.8	0.00389279	0.00332945	13.9
3	availability	21.5384	21.5	0.00241205	0.00150362	21.5
3	latency	21.5384	21.5	0.00241205	0.00150362	21.5
4	reputation	36.9230	36.9	0.00208542	0.00062291	36.9
sum	5	100	99.8	...	...	100

Promethee Process (1/2)

$$P_j(a,b) = F_j[d_j(a,b)] \forall a,b \in A$$
$$P_j(a,b) = F_j[-d_j(a,b)] \forall a,b \in A$$
$$d_j(a,b) = g_j(a) - g_j(b)$$
$$0 \le P_j(a,b) \le 1$$

Performs pairwise comparisons of service alternatives on all QoS considered by the user

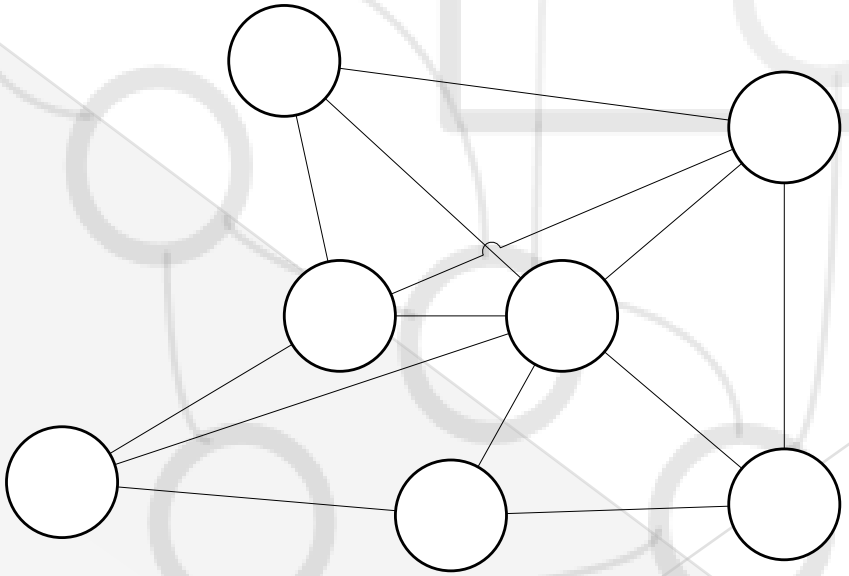
$$\pi(a,b) = \sum_{j=1}^{k} P_j(a,b) w_j$$

$$\pi(b,a) = \sum_{j=1}^{k} P_j(b,a) w_j$$

$$\Phi^+(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a,x)$$

$$\Phi^-(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x,a)$$

$$\Phi(a) = \Phi^+(a) - \Phi^-(a)$$



Promethee Process (2/2)

$$P_j(a,b) = F_j[d_j(a,b)] \forall a,b \in A$$
$$P_j(a,b) = F_j[-d_j(a,b)] \forall a,b \in A$$
$$d_j(a,b) = g_j(a) - g_j(b)$$
$$0 \le P_j(a,b) \le 1$$

Preference type used to make the pairwise comparison

$$\pi(a,b) = \sum_{j=1}^{k} P_j(a,b) w_j$$
$$\pi(b,a) = \sum_{j=1}^{k} P_j(b,a) w_j$$

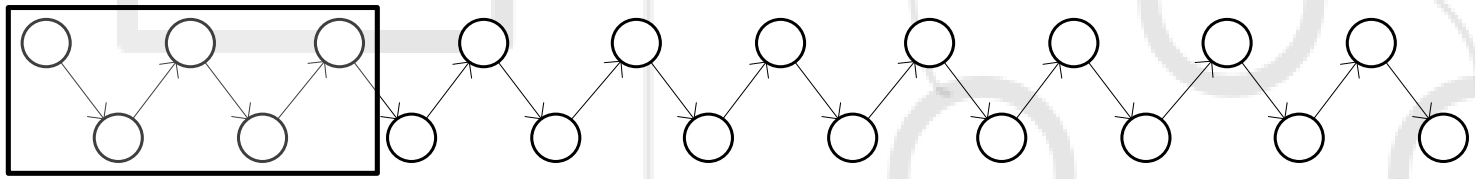
Weights determined with the Priorities Specification make appear the relative importance of each QoS Dimension or Characteristics

$$\Phi^+(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a,x)$$
$$\Phi^-(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x,a)$$
$$\Phi(a) = \Phi^+(a) - \Phi^-(a)$$

Selection process driven by user specifications

MCDM ranking

- Promethee methods outputs a ranking over alternatives
- Numeric ranking provides relative importance between successive alternatives
  - Determine the best available service
  - Restrict the set of service candidates
    - Selection optimized with trust
    - Integration in service composition problems



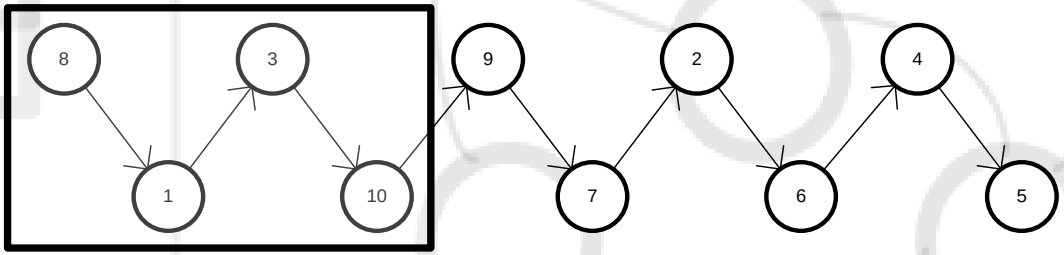
MCDM ranking: example

Table 4. Characteristics of available services

Provider	Availability	Latency	Reliability	Reputation	Security
1	78	410	83	92	8
2	71	380	78	91	7
3	87	455	74	95	6
4	57	240	86	76	9
5	82	380	67	86	7
6	92	520	87	90	6
7	74	450	85	91	7
8	86	400	76	92	6
9	76	380	82	89	9
10	78	390	87	90	8

Table 5. Outranking flows

Provider	ϕ^+	ϕ^-	ϕ
1	53,09	32,46	20,63
2	38,12	47,92	-9,80
3	49,90	37,61	12,29
4	37,30	58,47	-21,17
5	35,40	65,05	-29,65
6	33,08	44,11	-11,03
7	28,95	37,10	-8,15
8	60,61	26,71	33,90
9	45,48	42,63	2,85
10	46,17	36,03	10,14



Conclusions

- Applicable to various quality dimensions
  - Not limited to a predefined set of QoS Dimensions
- Weights and preference types of QoS Dimensions are specified by the user
- Generic approach
  - Can be integrated in larger composition approaches