ICSOC 2005:

Extending OWL for QoS-based Web Service Description and Discovery

Kyriakos Kritikos
PhD Candidate
kritikos@csd.uoc.gr

Computer Science Department, University of Crete
Heraklion, Crete, Greece

Overview

- Problem Area
- QoS definition and aspects
- QoS importance
- Prominent approaches in QoS-based WS Description and Discovery
- Our proposal

Problem Area (I)

- Facts:
  - Huge amount of Web Services (WSs) advertised in UDDI registries
  - UDDI is a de-facto standard
- Problems:
  - Discover WSs based on requester’s functional needs
  - UDDI uses syntax-based description and discovery approach leading to low precision and accuracy
- Solution:
  - Ontology-based (semantic) description and discovery approaches

Problem Area (II)

- However, many advertised WSs may provide the same functionality
- Focus now is in using non-functional characteristics of WSs like QoS to:
  - Filter the list of functionally equivalent WSs based on non-functional constraints (matchmaking/filtering)
  - Select the best WS by prioritizing non-functional characteristics (selection)
- Unfortunately, the current research efforts in QoS-based WS Description and Discovery fail because:
  - Either they are syntax-based
  - Or their QoS-based WS semantic description is inadequate
  - Or their QoS-based filtering and selection algorithms are ineffective or not accurate

What is QoS

- ISO 8402 def.: "*Totality of features of a product or service that bear on its ability to satisfy stated or implied needs*"
- 3 different views of QoS:
  - Quality as functionality
  - Quality as conformance
  - Quality as reputation
- We choose the second one as:
  - Functionality is captured by the current WS description standards
  - Reputation can be considered a QoS property derived over time for a WS
- So, we consider QoS of a WS as a set of non-functional characteristics/attributes that may impact the quality of the service offered by the WS

QoS Aspects

- Many aspects of QoS important to WSs organized into QoS categories, which are the following:
  - Runtime related containing QoS metrics like: scalability, capacity, performance (response time, latency, throughput, execution time, transaction time), reliability (MTBF, MTF, MTTT, availability), continuous availability, failure masking, operation semantics, server failure, data policy, robustness, exception handling, accuracy
  - Transaction Support related containing the two following QoS params: ACID_properties_supported, transaction_mechanisms_supported
  - Configuration management and Cost related containing QoS attrs like: regulatory, supported_standards, stability/change cycle, guaranteed messaging requirements, cost, completeness, reputation
  - Security related containing QoS attrs stating which security props are satisfied and which security mechanisms are supported
  - Network related

QoS Management Importance (I)

- We choose QoS, a subset of possible non-functional char/tics of WSs, because of the benefits of QoS Management for Web Processes (WPs) and WSs:
  - For organizations, being able to characterize WPs based on QoS has four distinct advantages:
    - Translate their vision into their business processes as WPs can be designed according to QoS metrics
    - Selection and execution of WPs based on their QoS
    - Monitoring of WPs based on QoS to assure compliance both with initial QoS requirements and targeted objectives and to trigger adaptation strategies
    - It allows for the evaluation of alternative strategies when adaptation becomes necessary
  - All the above advantages of QoS management of WPs also apply to WSs

QoS Management Importance (II)

- Further analysis on advantages for WSs:
  - In WS Discovery: functionally equivalent WS ads can be filtered by the values stated on QoS properties
  - In WS Selection: the results of QoS-based WS Discovery are ordered based on the stated values of some QoS properties and the importance of these QoS properties
  - Requester negotiates with the provider having the best advertised WS (derived from QoS-based WS Selection) in order to come into a commonly agreed SLA or contract (for WS Execution). If negotiation fails, the second best provider is contacted
  - In WS Composition: runtime selection of component services, during the execution of a composed WS, based on quality criteria and following a local or a global selection strategy.

Prominent Approaches (I)

- Zhou, Chia and Lee (2004) extend DAML-S by associating a *ServiceProfile* with many *QoSProfiles* (i.e. service offerings)
- They have developed an upper ontology that references external DAML ontologies for metrics and units
- They have developed a mid-level ontology containing basic QoS metrics that can be further extended to include custom-made metrics
- QoS-based WS matchmaking is based on the concept of QoS profile compatibility ($\neg(C_1 \cap C_2 \sqsubseteq \perp)$). It is performed by computing the subsumption relat/ship of a request's QoS profile with all available QoS ads. QoS-based WS Selection is not dealt!
- Disadvantages:
  1. QoS metrics model not rich enough.
  2. The range of metrics is the set $\mathbb{N}^+$
  3. Slow DL reasoners, not supporting complex math expressions

Prominent Approaches (II)

- Martin-Diaz, Ruiz-Cortes, Benavides, Duran and Toro (2003) use a syntax-based symmetric QoS model expressing math constraints for QoS metrics
- Before matchmaking, a QoS spec is transformed to a CSP, which is checked for consistency (any solution)
- Matchmaking is performed according to the concept of conformance (every solution of demand is a solution of offer)
- For WS Selection, a (QoS) score for a WS ad is expressed as a CSOP, where for every solution of the offer, we find the one that minimizes the weighted sum of the weight of each metric multiplied with its utility assessment value.
- Disadvantages:
  1. Syntax-based approach (similarity of QoS metrics based on names)
  2. CS(O)Ps have NP solutions if QoS constraints have non-linear expressions

Our Proposal

- Research and finalize a set of requirements for the QoS description of WSs.
- Based on requirements, we propose an ontology for QoS-based WS description
- We introduce the concept of semantic QoS metric matching
- We extend the most prominent QoS-based WS matchmaking and selection algorithms
- Implement and formally evaluate the above algorithms
- Further extend the QoS description ontology and discovery algorithms
- Develop tools for providers and requesters

Requirements

- Extensible and formal semantic QoS model
- Standards compliance
- Syntactical separation of QoS and functional parts of service spec
- Both requester and provider QoS specification
- Refinement of QoS specification (extensibility, reusability)
- Fine-grained QoS specification (for the whole WS and its parts)
- Extensible and formal QoS metric model which should specify:
  - The value-set of the attribute
  - The domain of discourse of the attribute
  - Its relationship with other attributes
  - Its association with a unit, measured property and measurement function, constructs that should also be specified
  - Functional description of how this QoS attr of a composed WS can be derived from the corresponding attrs of the individual WSs
- Classes of service (an ad should present many offers)

Ontology

- Based on the previous requirements, we have developed an ontology named OWL-Q
- This ontology is carefully separated into many facets, each capturing an aspect of QoS WS description and can be extended/modified independently of the other
- We choose OWL as the ontology formalism (W3C standard)
- Our ontology extends OWL-S (standard for the semantic description of WSs)
- It is an upper level ontology
- We plan to develop mid-level ontologies specifying the basic (domain independent) metrics
- We also plan to develop mid-level ontologies for the units, measured properties and measurement functions
- Domain experts should develop low-level ontologies for metrics

Semantic QoS Metric Matching

- So far, two QoS metrics are the same if they have the same name
- This leads to low accuracy and precision
- Semantic QoS metric matching is the key
- Based on OWL-Q, we have devised the following algorithm:
  - Two simple QoS metrics are the same, if they have the same domain, are of the same type and measure the same property
  - Two complex QoS metrics are the same, if they have the same previously defined factors plus they have the same measurement function that takes as input the same QoS metrics (recursive)
  - We cannot compare a simple and a complex QoS metric
- Mid-level ontologies should be developed for the "Function" and "Measured Property" concepts for better matching
- Ontology mapping techniques may be utilized in cases where there are pairs of QoS metrics that can not be easily characterized

Extend QoS-based WS matchmaking algorithms

- Based on the previous algorithm/concept, we extend the QoS-based WS matchmaking algorithm of Martin-Diaz et. al.:
  - OWL-Q advertisements and OWL-Q request are transformed to CSP problems following two directives:
    - Only metrics which are semantically equal should correspond to the same CSP variable
    - If two equal metrics do not use the same units, then we consider the request's metric unit as the default and a unit transformation procedure (from the provider's unit to the requester's) is performed
  - If a CSP of an advertisement does not contain all the variables of the CSP of the request, it is considered as *fail match*.
  - We solve the remaining advertisement CSPs and the CSP of the request
  - For every solution of the CSP of the request, we check if it is contained in the solution space of the CSP of an advertisement.

Extend QoS-based WS Selection algorithms

- Based on the previous algorithm/concept, we extend the QoS-based WS selection algorithm of Martin-Diaz et. al.:
  - We take the same first step of the matchmaking algorithm
  - Based on the CSP of an offer , we compute the minimum and maximum utility assessment of the offer
  - The overall utility assessment (QoS score) is given by calculating the weighted sum of the minimum and maximum utility assessments

Implement & Formally Evaluate the Algorithms

- The implementation is under development. It will use an OWL inference engine and an efficient CSP engine
- The formal evaluation is not yet performed. Its conditions and are investigated
- This evaluation is a necessity in order to prove that the previously defined algorithms are efficient, accurate and precise

Extend Ontology & Algorithms + Tools

- OWL-Q should be extended to incorporate other non-functional attributes (mainly contextual ones) and its design should be finalized
- The QoS-based WS matchmaking algorithm should be extended in order to distinguish between hard and soft constraints
- GUIS and other tools should be developed that will help the user in describing and discovering WSs

The End

- Please make your comments
- Do not hesitate to ask questions