

## 시맨틱 웹 서비스를 위한 QoS 매칭 메커니즘

유소연<sup>0</sup> 유정연 이규철  
 충남대학교 컴퓨터공학과  
 {syyou<sup>0</sup>, jyyou, kclee}@ce.cnu.ac.kr

### QoS Matching Mechanism for Semantic Web Services

So-Yeon You<sup>0</sup> Jeong-Youn Yu Kyu-Chul Lee  
 Dept. of Computer Engineering, Chungnam National University

#### 요 약

웹 서비스의 등록, 검색, 조합, 실행을 자동적으로 수행할 수 있도록 웹 서비스에 시맨틱 웹 기술을 적용시킨 것이 시맨틱 웹 서비스(Semantic Web Services)이다. 시맨틱 웹 서비스는 원하는 서비스를 찾으려는 사용자의 노력을 줄이기 위해 기계가 이해할 수 있는 정보를 서비스에 추가시킴으로써 정확하게 원하는 서비스를 찾을 수 있게 해준다. 수많은 서비스들 중 같은 역할을 하는 서비스들이 있을 수가 있다. 그 중 하나의 서비스만을 선택해야 할 때 서비스의 기능이 아닌 질적인 측면에서 QoS(Quality of Service)를 고려하면 최선의 서비스를 선택할 하는데 도움을 줄 수 있다. 따라서 시맨틱 웹 서비스의 검색과 조합의 측면에서 매치메이킹(matchmaking)에 대한 연구의 하나로써 QoS의 매칭에 대한 연구를 수행하게 되었다.

기존의 QoS 매칭과 관련된 연구에서 QoS의 유사도만 계산하는 방법은 QoS 요소의 값의 특성을 반영하지 않는다는 것과 QoS 유사도의 순위가 높더라도 좋은 서비스라는 것을 의미하지 않는다는 두 가지 문제점을 발견하였다.

따라서 본 논문에서는 기존 연구의 문제점을 해결할 수 있는 방법을 제시하고자 한다. 먼저 기존의 웹 서비스 관련 연구에서 논의된 여러 QoS 모델을 수렴하여 QoS 모델의 요소들을 결정하였다. 그리고 기존 연구의 두 가지 문제점을 해결하기 위해 각 QoS 요소의 표준편차를 이용한 표준값을 구하여 QoS 요소의 값의 특성을 반영하였다. 또한 매칭 결과 순위가 높은 것이 사용자에게 더 선호되는 좋은 서비스라는 것을 보장하는 메커니즘을 제안하였다.

#### 1. 서 론

시맨틱 웹 서비스를 이루기 위해 가장 기본이 되는 작업은 사용자가 원하는 서비스를 정확하게 찾아내는 것이다. 찾아진 서비스는 그것들 중 최선의 것을 선택하여 실행을 시키거나 다른 서비스와의 조합을 할 때 이용할 수 있다.

이렇게 사용자의 요구에 맞는 서비스를 발견하였을 때 서비스의 QoS를 고려하여 같은 조건의 서비스가 여럿일 경우 그 중 최선의 선택을 할 수 있게 도움 수 있다. 예를 들어 책을 판매하는 같은 기능을 가진 두 서비스가 있을 때 하나는 신뢰도가 90%이고, 또 다른 하나는 신뢰도가 99%이라고 하자. QoS를 고려하지 않는다면 두 서비스는 같은 서비스이므로 어떠한 것이 더 좋은지 판단을 할 수가 없다. 그러나 신뢰도라는 QoS를 고려하였을 때에는 신뢰도가 더 높은 서비스가 더 질이 좋은 서비스라는 것을 판단할 수 있을 것이다.

본 논문은 시맨틱 웹 서비스를 위한 매치메이킹 시스템을 설계하는 중간 단계로써 시맨틱 웹 서비스의 QoS에 관한 매칭 메커니즘에 대해 자세히 다루고자한다.

본 논문은 다음과 같이 구성된다. 2장에서 시맨틱 웹 서비스의 QoS 매칭과 관련된 연구에 대해 살펴보았다. 그리고 3장에서는 본 논문에서 제안한 매칭 메커니즘에서 사용하는 QoS 모델의 요소에 대해 설명하였고, 4장에서는 3장에서 설명된 QoS 요소를 가지고 QoS 매칭을 어떠한 방법으로 수행하는 지에 대한 메커니즘을 제안하였다. 마지막으로 5장에서 결론과 향후 연구에 관해 기술하였다.

#### 2. 관련 연구

현재 시맨틱 웹 서비스에 대한 연구는 활발히 수행되고 있다. 하지만 시맨틱 웹 서비스의 QoS에 대한 연구는 아직 초기 단계이다. OWL-S[1]와 UDDI[2]에서는 서비스를 표현할 때 QoS 요소를 추가해야 한다는 정도만을 표현하고 있다. 구

체적으로 어떻게 QoS를 매칭에서 이용할지에 대한 연구는 METEOR-S 프로젝트[3]에서만 이루어지고 있다. 따라서 본 논문에서는 METEOR-S 프로젝트에 대해서만 관련 연구로 다루고자 한다.

METEOR-S 프로젝트는 Georgia 대학의 LSDIS(Large Scale Distributed Information Systems) 연구실에서 시맨틱 웹 서비스에 대한 연구의 일환으로 수행하고 있는 프로젝트이다. 시맨틱 웹 서비스를 이루기 위해 웹 서비스의 QoS, 검색, 조합, 실행에 시맨틱을 적용하는 연구를 수행하고 있다. 본 논문에서는 METEOR-S 프로젝트 중 QoS 매칭과 관련된 부분에 대해서만 다루겠다.

METEOR-S 프로젝트는 QoS 요소를 웹 서비스의 응답 시간을 나타내는 시간(Time), 웹 서비스를 실행하는데 드는 비용을 나타내는 가격(Cost), 사용자가 서비스를 요구했을 때 서비스를 수행할 수 있는 지의 확률을 나타내는 신뢰성(Reliability) 이렇게 세 가지로 정의 하였다. 위의 QoS 요소들은 각각이 최소 값, 평균 값, 최대 값을 가진다. 이들 값을 가지고 QoS의 유사도를 구하는데 이용한다.

METEOR-S 프로젝트에서는 사용자의 질의에 해당되는 것을 ST(Service Template)라고 하고, 실제 웹 서비스의 정보를 담고 있는 것을 SO(Service Object)라고 하였다.

ST와 SO의 유사도는 OpSimilarity(ST,SO)의 함수에 의해 계산된다. OpSimilarity(ST,SO)의 값은 0과 1사이의 값으로 계산되는데 1에 가까울수록 SO가 ST에 더 유사하다는 것을 보여준다.

$$OpSimilarity(ST,SO) = \frac{1}{\sqrt[3]{QoSDimD(ST,SO,time) * QoSDimD(ST,SO,cost) * QoSDimD(ST,SO,reliability)}}$$

[그림 1] ST와 SO의 유사도 계산식

QoSdimD(ST,SO,dim)의 값은 ST와 SO의 QoS 요소의 거리 차이(dcd)를 계산하여 그것의 제곱근으로 구한다.

$$QoSdimD(ST,SO,dim) = \sqrt[3]{dcd_{min}(ST,SO,dim) * dcd_{avg}(ST,SO,dim) * dcd_{max}(ST,SO,dim)}$$

[그림 2] 각 요소별 유사도 계산식

$$dcd_{min}(ST,SO,dim) = 1 - \frac{|\min(SO,qos(dim)) - \min(ST,qos(dim))|}{\min(ST,qos(dim))}$$

[그림 3] 각 요소별 ST와 SO의 거리 차이

그러나 QoS의 유사도를 구하는 위의 방법에는 두 가지 문제가 있다.

첫째, QoS의 유사도는 ST와 SO사이의 거리 차이가 제일 작은 것을 최상위에 랭크함으로써 QoS의 순위가 높더라도 좋은 서비스라는 것을 의미하지는 못한다.

[표 1] QoS 유사도 예제

Reliability	min	avg	max
ST	90	94	99
SO1	90	93	99
SO2	90	98	99
ST와 SO1의 차이	0	1	0
ST와 SO2의 차이	0	4	0
SO1의 QoS 유사도	0.995		
SO2의 QoS 유사도	0.978		

예를 들어 위의 [표 1]에서 보면 최소 90%, 평균 94%, 최대 99%의 신뢰성을 갖는 서비스를 찾아달라는 질의가 있다. 그때에 SO1과 SO2에 대해 METEOR-S의 QoS 유사도 메커니즘으로 유사도를 구하면, 신뢰성이 93%인 서비스(SO1)가 98%인 서비스(SO2)보다 질이 낮음에도 불구하고 질의와의 거리 차이가 작다는 이유로 상위에 랭크하게 된다. 이것은 QoS의 순위가 서비스의 질이 좋다는 것을 의미하지는 않는다는 것을 보여준다.

둘째, 각 요소의 유사도를 통합하여 전체 유사도를 구할 때 특정 한 요소가 전체 유사도를 결정지을 수 있다는 것이다.

[표 2] Cost에 대한 유사도 예제

Cost	min	avg	max
ST	200	250	300
SO1	200	230	300
SO2	220	280	300
ST와 SO1의 차이	0	20	0
ST와 SO2의 차이	20	30	0

[표 3] Reliability 유사도 예제

Reliability	min	avg	max
ST	90	94	99
SO1	87	96	98
SO2	90	94	99
ST와 SO1의 차이	3	2	1
ST와 SO2의 차이	0	0	0

[표 2]와 [표 3]의 유사도를 바탕으로 전체 유사도를 계산하면 [표 4]와 같다.

[표 4] 전체 유사도

SO1의 전체 QoS 유사도	0.963
SO2의 전체 QoS 유사도	0.943

가격에 대한 값의 특성과 신뢰도에 대한 값의 특성이 다르다. 그러나 ST와 SO 사이의 값의 차이를 단지 숫자로만 비교한다면 신뢰성에 비해 가격의 차이가 크므로 신뢰성의 차이는 가격의 차이에 묻혀 전체 유사도에 영향을 미치지 못하게 된다. 위의 두 가지 문제점을 해결하는 방법을 본 논문에서 제시하고자 한다.

### 3. QoS 요소

본 논문의 QoS 모델의 요소들은 IBM에서 제안한 일반적으로 시맨틱 웹 서비스에서 QoS를 나타내는 요소들[4]에 METEOR-S, UDDI, OWL-S의 QoS 요소들을 고려하여 그것을 모두 수렴하고자 하였다.

- **가격(Cost)** : 웹 서비스를 실행하는데 드는 비용을 나타낸다. 재정 계획을 세울 때 도움을 줄 수 있는 요소이다.
- **시간(Time)** : 웹 서비스의 응답 시간(response time)을 나타낸다. 응답 시간은 지연 시간(delay time)과 처리 시간(process time)의 합으로 계산된다.
- **신뢰성(Reliability)** : 사용자가 서비스를 요구했을 때 서비스를 수행할 수 있는 지의 확률을 나타낸다. 1- 웹 서비스 실행 시 발생하는 실패(failure) 확률로 계산된다. 이것은 실패 없이 얼마나 버틸 수 있는지를 의미한다.
- **정확성(Accuracy)** : 웹 서비스의 실행 시 얼마나 오류(error)없이 실행될 수 있는 지의 확률을 나타낸다. 1- 웹 서비스 실행 시 발생하는 오류(error)율로 계산된다.
- **가용성(Availability)** : 어떤 특정한 순간에 서비스가 이용 가능할 확률을 나타낸다.
- **마지막 갱신 시간(Last Update Time)** : 서비스가 마지막으로 갱신(update)된 시간을 나타낸다. 얼마나 서비스가 최신의 상태로 유지되는지를 나타낸다.

가격, 시간, 신뢰성의 경우는 METEOR-S의 요소를 참조하였고, 특히 시간의 경우는 METEOR-S와 OWL-S에서 QoS를 나타내는 요소로 표현되어있다.

UDDI 프로젝트의 요청의 수(Request Count), 응답의 수(Reply Count)는 가용성으로, 실패의 수(Fault Count)는 신뢰성으로, 마지막 갱신 시간(last Update Time)은 그대로 본 논문의 QoS 모델의 요소로 추가하였다.

그리고 IBM에서 제안한 일반적으로 시맨틱 웹 서비스에서 필요로 하는 QoS요소인 정확성을 본 논문의 QoS 모델에 추가하였다.

### 4. QoS 매칭 매커니즘

사용자가 서비스를 선택할 때 QoS의 측면에서 선호하는 기준을 본 논문에서는 다음 두 가지로 나누어 정의하였다.

- QoS 요소의 값이 낮을수록 선호도가 높은 것  
가격, 시간
- QoS 요소의 값이 높을수록 선호도가 높은 것  
신뢰성, 정확성, 가용성, 마지막 갱신 시간

본 논문에서 제안한 QoS 매칭 매커니즘은 QoS 요소의 값은 최소 값, 최대 값, 평균 값 중 평균 값만을 선택하여 계산한다. 그리고 다수의 서비스들 중 질의의 범위에 해당되는 서비스들이 선택되었다고 가정하고 다음의 순서대로 QoS 매칭 매커니즘을 수행한다.

- QoS 요소의 값 조정
- 표준값 계산
- 전체 QoS 점수 계산
- 전체 QoS 점수 정렬 후 순위 결정

다음은 본 논문에서 제안한 매칭 메커니즘을 사용하여 실제 서비스에 대한 QoS 순위를 구한 예이다.

[표 5] QoS 매칭 메커니즘 적용 예제

	SO1	SO2	SO3	SO4	SO5	SO6
cost	250	250	300	350	550	570
reliability	95	97	90	90	85	99
cost.표준값	40.35	40.35	44.11	47.87	62.91	64.41
reliability.표준값	39.74	39.51	40.33	40.33	40.91	39.27
전체 QoS 점수	80.09	79.86	84.44	88.20	103.82	103.68
QoS 순위	2	1	3	4	6	5

4.1. QoS 요소의 값 조정

본 논문에서 제안하는 QoS 매칭 메커니즘은 전체 QoS 점수가 낮을수록 선호도가 높고 좋은 서비스라는 것을 나타낸다. 따라서 QoS 요소의 값이 높을수록 선호도가 높은 요소 즉, 신뢰성, 정확성, 가용성의 경우는 최대 값인 100%에서 QoS 요소의 원래 값을 빼서 값을 조정한다. 마지막 갱신 시간의 경우는 현재 시간에서 마지막 갱신 시간의 차를 계산하여 값을 조정한다. 그러나 QoS 요소의 값이 작을수록 선호되는 요소 즉, 가격과 시간에 대해서는 값의 조정이 필요 없이 그대로 사용된다.

예를 들어 가격이 \$250이고 신뢰성이 95%인 서비스가 있다면, 가격의 경우는 그대로 250의 값을 사용하고 신뢰성의 경우는 값이 낮을수록 좋은 서비스라는 것을 나타내기 위해 100에서 95를 뺀 5로 그 값을 조정해 준다.

4.2. 표준값 계산

표준값[5]은 한 서비스의 QoS 요소의 원래 값이 전체 서비스들의 QoS 요소 집단에서 어느 정도 위치에 있는지를 보여준다. 일반적으로 표준값은 Z점수, T점수 등이 있는데, Z점수는 평균을 0, 표준편차를 1로 하는 점수이고, T점수는 음수나 소수로 표현되는 Z점수의 불편함을 해결하기 위해 평균을 50, 표준편차를 10으로 하는 점수이다. 본 논문에서는 T점수를 사용하여 표준값을 계산하였다.

서비스 SO1의 가격의 표준값은 T(SO1, cost) 함수로 계산된다.

$$T(SO1, cost) = \left( \frac{SO1.cost - m}{s} \right) \times 10 + 50$$

$$s = \frac{\sqrt{(SO1.cost - m)^2 + (SO2.cost - m)^2 + \dots + (SO_n.cost - m)^2}}{n}$$

m = 평균, n = 서비스의 개수  
[그림 4] 각 요소별 표준값 계산식

4.3. 전체 QoS 점수 계산

위에서 구해진 QoS 요소들의 표준값의 합으로 전체 QoS 점수를 계산한다.

$$TotalScore(SO1) = T(SO1, cost) + T(SO1, time) + T(SO1, reliability) + T(SO1, accuracy) + T(SO1, availability) + T(SO1, lastUpdateTime)$$

[그림 5] 전체 QoS점수 계산식

4.4. 전체 QoS 점수 정렬 후 순위 결정

[그림 5]의 방법으로 구해진 전체 QoS의 점수는 [그림6]의 정렬 알고리즘을 통해 가장 작은 값이 최상위에 링크되도록 한다.

각 서비스마다의 전체 QoS의 점수는 TSortList에 저장되어 있다고 가정할 때 TotalScoreSort함수를 통해 순위가 결정된다.

```
TotalScoreSort(TSortList) {
    for (i = 0; i < TSortList.Size - 1; i++) {
        for (j = i + 1; j < TSortList; j++) {
            if (TSortList[i] > TSortList[j]){
                Temp = TSortList[i];
                TSortList[i] = TSortList[j];
                TSortList[j] = Temp;}}}}
[그림 6] 전체 QoS점수 정렬 알고리즘
```

SO1과 SO2에서 보면 가격이 같으나 신뢰성이 더 높은 SO2가 SO1보다 QoS 순위가 더 높은 것을 볼 수 있다. 그리고 SO3와 SO4에서는 신뢰성이 같을 경우 가격이 더 낮은 SO3가 SO4보다 순위가 더 높은 것을 볼 수 있다. SO5와 SO6에서는 단순히 숫자로서 비교했을 때에는 가격의 차이가 신뢰성의 차이보다 크지만, 표준값을 적용시킴으로써 보다 정확한 QoS 순위를 구하는 예를 보여주고 있다.

위의 방법은 QoS의 각 요소의 중요도가 모두 같다는 가정 하에서 계산된 것이고, 사용자의 판단에 의해 각 요소에 가중치를 적용하여 보다 사용자의 요구에 맞는 QoS 매칭을 이룰 수 있도록 매칭 메커니즘을 확장할 수 있다. 예를 들어 다른 QoS의 요소들보다도 가격이 저렴한 서비스를 원할 때에는 가격의 표준값에 가중치를 곱하여 전체 QoS 순위를 계산하는데 반영할 수도 있다.

5. 결론 및 향후 연구

본 논문은 시맨틱 웹 서비스를 위한 매치메이킹 시스템을 구현하기 위한 중간 단계로써 QoS 매칭모듈을 만들면서 연구한 내용이다. METEOR-S 프로젝트의 QoS 매칭 방법의 문제점을 발견하였고, 그것을 해결하는 방법을 제안하였다. 따라서 본 논문에서 제안한 방법으로 QoS 매칭 모듈을 매치메이킹 시스템에 추가한다면 같은 기능을 수행하더라도 선호도가 높고 질이 더 좋은 서비스를 찾아낼 수 있는 매치메이킹 시스템을 구현할 수 있을 것이다.

본 논문에서 제안한 방법으로 실제 모듈을 구현하는 것이 향후 연구과제로 남아있다.

6. 참고문헌

[1] W3C, "OWL-S: Semantic Markup for Web Services," <http://www.daml.org/services/owl-s/1.1B/owl-s.pdf>, 2004  
 [2] OASIS, "Technical Note, Representing Web Services Quality of Service Information in UDDI," 2004  
 [3] Cardoso, J. and A. Sheth, "Semantic e-Workflow Composition," Journal of Intelligent Information Systems (JIIS), 2002  
 [4] Anbazhagan Mani and Arun Nagarajan, "Understanding quality of service for Web services," <http://www-106.ibm.com/developerworks/library/ws-quality.html?n=ws-1172>, 2002  
 [5] 정을수, "표준점수의 이해", [http://busanedu.net/portal/edu/up\\_files/e2/표준점수의 이해 2.hwp](http://busanedu.net/portal/edu/up_files/e2/표준점수의 이해 2.hwp), 2004