

# QoS기준을 충족하는 웹 서비스의 구성 및 선정

이 우 기\*    조 대 래\*\*    강 석 호\*\*

## ◆ 목 차 ◆

- |                |                  |
|----------------|------------------|
| 1. 서 론         | 4. 웹 서비스 모델      |
| 2. 연구배경 및 관련연구 | 5. 웹 서비스를 위한 QoS |
| 3. 워크플로우 접근법   | 6. 결 론           |

## 1. 서 론

많은 기업들은 빠르게 변화하는 기술과 다양한 고객의 요구를 만족하기 위해 유연하고 효율적인 시스템을 구축하고자 노력하고 있다. 기업 내부적으로는 비즈니스 프로세스를 정의·실행하며, 발생한 인스턴스에 대해 모니터링하고 분석한 후 새로운 비즈니스 프로세스를 정의하여왔다. 하지만, 기업간 경쟁이 치열해지고 다양한 시장에 대응하기 위해서는 기업 내 프로세스뿐만 아니라 기업간 프로세스에 대해서도 이러한 노력이 필요하게 되었다. 기업간 어플리케이션이 가지고 있는 문제점은 그것을 구성하는 컴포넌트들이 자생적이고 이기종이며 분산된 환경에서 구축되었다는 점이다. 이러한 문제점을 해결해 줄 수 있는 대안으로 각광 받고 있는 것이 웹 서비스이다. 즉, 웹 서비스에 많은 노력을 들이 가장 큰 이유 중에 하나는 그것이 매우 효과적으로 통합의 복잡성과 비용을 완화시키는 잠재력을 가지고 있기 때문이다. 웹 서비스는 플랫폼에 독립적이고, 기업간 협업을 용이하게 해주며, 마지막으로 서비스를 조합하여 사용하기 편리하다는 장점이 있다. 웹 서비스 이전에도 많은 방법론이 존재하였으나, 웹 서비스는 인터넷을 기

반으로 한다는 점과 IBM, OASIS, ORACLE, Microsoft, SUN과 같은 IT시장에서 영향력이 큰 표준기구들과 업체들이 대다수 참여하는 가장 유망한 방법으로 평가 받고 있다 [7,9].

웹 서비스 어플리케이션과 이를 공급하는 서비스 공급자들의 수가 증대함에 따라 고객은 최적의 서비스 공급자를 선택하는 문제에 직면하게 된다. 이러한 문제는 기업에서도 동일하게 발생한다. 기업 간 협업을 위해 웹 서비스를 구축할 때, 비즈니스 프로세스의 각 태스크들을 수행해 줄 수 있는 적절한 웹 서비스 어플리케이션을 선택하여야 한다. 현재 많은 연구기관과 업체들은 품질을 기준으로 프로세스 기반 웹 서비스 구성 방법론에 대해 연구하고 있다. 하지만 이러한 기준들을 평가할 수 있는 적합한 방법에 관한 연구가 아직 부족하며, 서비스 공급자를 채택하는 방법 또한 만족스러운 수준은 아니다.

웹 서비스에는 다양한 정의가 있지만 W3C에서 다음과 같은 정의를 표준으로 정하고 있다 [W3C, 2003]. 웹 서비스는 URI로 인식되고(identified by) 전역 인터페이스(public interfaces)와 바인딩이 XML 기반 메시지로써 인터넷 프로토콜에서 정의되고 설명되는 소프트웨어 시스템이다. 또한 웹 서비스는 컴포지션을 하는지 여부에 따라 단일 서비스(Atomic Service)와 복합 서비스(Composite Service)로 구분할 수 있다. 단일 서비스는 서비스를 수행할 수

\* 성결대학교 컴퓨터학부  
\*\* 서울대학교 산업공학과

있는 최소 단위이고, 복합 서비스는 단일 서비스들의 컴포지션을 통해 새로운 서비스를 제공해 주는 것이다.

## 2. 연구배경 및 관련연구

### 2.1 웹 서비스

웹 서비스는 기존의 인터넷 비즈니스 솔루션의 단점을 극복할 수 있는 획기적인 대안으로서 평가 받고 있다. 여러 가지 패키지 솔루션의 통합을 위해 EAI(Enterprise Application Integration)와 같은 노력이 있었지만, 이러한 통합 솔루션들의 능력은 e비즈니스에 적합한 협업 메커니즘 구조의 증대되는 수요를 감당하지 못하였다 [1]. 또 다른 문제점은 비즈니스 프로세스간 통합의 어려움을 들 수 있다. 협업을 통한 비즈니스의 가치 창출은 기본적으로 파트너와의 공유된 프로세스의 통합이 전제되어야 한다. 이것은 형식적인 프로세스 통합이 아니라 기업과 파트너간의 내재화된 프로세스 통합이어야 한다 [2]. 예를 들면, 인터넷 서점의 경우 핵심적인 가치 사슬(Value Chain)은 주문과 지불 그리고 배송이다. 구체적으로, 주문에 대한 지불의 가치사슬은 은행과 협업관계가 수립되어야 하며 배송이라는 가치사슬은 택배업체와의 협업관계를 통하여 고객이 주문한 서적이 최종 목적지까지 최대한 빠른 시간에 도착할 수 있도록 유기적으로 가치사슬간 프로세스들이 통합되지 않으면 안 된다. 그러나 대부분의 기업들은 다른 기업과의 협력 관계에 대하여 상호간 불신 및 조직 이기주의 등으로 인해 데이터나 어플리케이션 통합의 단순한 수준에서 보다 고차원적인 제품의 라이프사이클 전체를 고려한 진정한 비즈니스 프로세스간 공유를 통한 통합 단계로의 발전이 이루어지지 않고 있는 실정이다 [10].

과거 인터넷 비즈니스 솔루션의 공통적인 문제점은 이기종 및 다른 플랫폼간의 통합 및 연동이 어렵다는 점이다. 이러한 문제점을 해결해 줄 수 있는 방안으로 각광 받고 있는 것이 웹 서비스이다. 웹 서비스는 서로 다른 소프트웨어 에이전트들

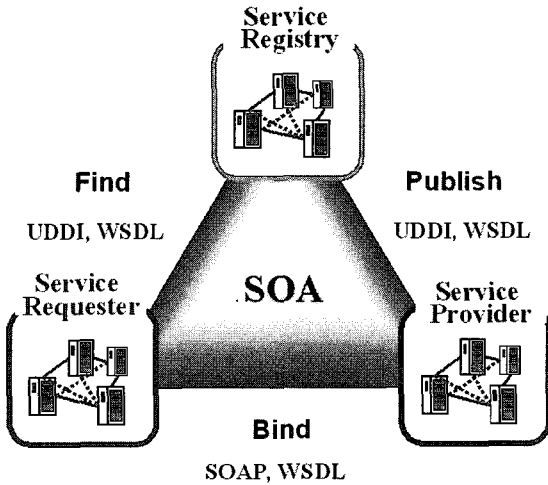
이 인터넷 기반의 프로토콜을 통해 XML기반의 메시지를 주고 받을 수 있도록 지원해주는 인터페이스이다[8]. 웹 서비스는 SOAP(Simple Object Access Protocol), XML(Extensible Markup Language), WSDL(Web Services Description Language), UDDI(Universal Description Discovery Integration) 등과 같은 인터넷 공개표준 기술의 인프라를 제공하여 기업 간 연동 및 프로세스간 통합을 가능하게 해준다.

### 2.2 웹 서비스의 구성도(Architecture)

SOA(Service Oriented Architecture)는 어플리케이션이나 솔루션을 개발하기 위해 기본이 되는 요소를 서비스로 간주하고 개발하는 컴퓨팅 패러다임이다. SOA에서 말하는 서비스란 완결된 비즈니스 로직을 담고 있는 컴포넌트 혹은 기능을 말하며, 비즈니스 요구 사항에 맞는 기능을 제공하는 것이 가장 중요한 목적이라고 할 수 있다. 그러나 어떤 형태로 설계된 것이 서비스인지에 대한 명확한 정의는 나오지 않고 있다. 한편에서는 표준에 맞추어 기술된 인터페이스를 통해 접근이 가능하고 독립적인 동작이 가능하면 어떤 형태이든 서비스로 보고 있다.

웹 서비스는 표준을 기반으로 인터페이스로서, 여러 기능들에 접근할 수 있도록 하여 서비스를 제공해 준다. 그러므로 웹 서비스는 SOA(Service Oriented Architecture)에 따라 어플리케이션이나 솔루션을 개발하는 방법론에 해당된다. 이 절에서는 웹 서비스로 구현될 SOA 기본적인 모델을 살펴본 후 복합 서비스를 제공해 줄 수 있는 확장된 모델을 고려하기로 하자.

그림 1은 웹 서비스를 통해 SOA를 구현한 그림이다. SOA는 서비스 공급자와 서비스 고객, 그리고 서비스를 등록하는 저장창고가 필요하다. 웹 서비스는 이들이 서비스를 게재하고, 찾고, 이용할 수 있도록 하는 XML기반의 인터페이스라고 할 수 있다. 위 그림에서의 SOAP, WSDL등이 웹 서비스 기술이라고 할 수 있다.



(그림 1) The Basic Service Oriented Architecture

• 서비스 공급자 (Service Provider)

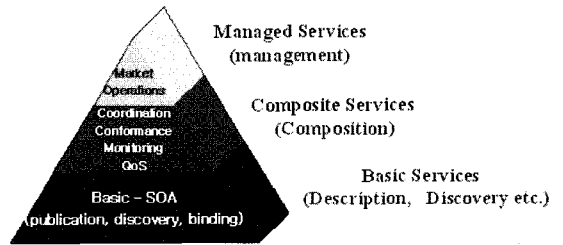
서비스 공급자는 웹 서비스를 제작하는 이를 말한다. 서비스 공급자는 웹 서비스 형태로 사용자에게 비즈니스 기능을 제공한다. 예를 들어 온라인 서점은 서비스를 웹 서비스로 공개할 수 있다. 이때 사용자가 웹 서비스를 이용하도록 하려면 다음과 같은 일을 해야 한다. 먼저 웹 서비스를 사용하는 단체에게 널리 알려진 표준 양식으로 서비스의 기능을 기술하고, 그 다음 공개적으로 접근할 수 있는 중앙 레지스트리에 웹 서비스의 세부사항을 공개해야 한다. 본 연구에서는 서비스 공급자가 서비스의 품질-시간, 비용, 신뢰도 등을 그 세부사항으로 공개한다고 전제하였다.

• 서비스 고객 (Service Customer)

서비스의 고객은 웹 서비스를 사용하는 대상자를 의미한다. 서비스 고객은 서비스 공급자가 기술한 웹 서비스 상세 내역을 통해 서비스의 기능을 알아낼 수 있다. 먼저 서비스가 공개된 레지스트리를 검색하여 서비스의 세부 사항을 알아낸다. 그 다음 원하는 서비스에 바인딩하여 실제로 해당 서비스의 기능을 이용하게 되는 것이다.

• 서비스 레지스트리 (Service Registry)

서비스 레지스트리는 서비스 공급자가 자사의



(그림 2) Extended Service Oriented Architecture

웹 서비스 상세 내역을 올려두고 서비스 수요자가 원하는 서비스를 발견할 수 있도록 하는 중앙 저장소를 의미한다. 수요자는 원하는 서비스를 검색하고 바인딩하여 서비스를 이용할 수 있다. 일반적으로 웹 서비스를 제공하는 회사의 상세정보, 서비스 자체정보 등이 레지스트리에 저장된다.

그림 2는 확장된 Service Oriented Architecture이다 [10]. 맨 밑단의 Basic Service Oriented Architecture는 개개의 서비스들의 기본적인 운용 (publication, discovery, binding)이 이루어진다. 서비스 컴포지션 레이어는 여러 개의 서비스들을 묶어서 하나의 복합 서비스 (Composite Service) 형태로 통합하기 위해 필요한 기능과 역할을 포함하는 층이다. 예를 들면 Coordination, Monitoring, Conformance, QoS Composition 등이 서비스 컴포지션 레이어에 해당한다. 마지막으로 서비스 매니지먼트 레이어는 서비스의 전반적인 기능과 정확성을 관리하고 고객에게 보증하는 것에 관련된 층이다. 본 연구는 QoS Composition을 하기 위해 어떤 서비스 공급자를 선택할 것인지의 문제를 다루고 있기 때문에 서비스 컴포지션 레이어와 관련 있다.

2.3 프로세스 구조

프로세스 구조란 프로세스를 이루는 단위업무와 이들간의 선후 관계를 말하는 것으로, 본 논문에서는 복합 웹 서비스의 QoS를 평가하기 위해 기존의 워크플로우 프로세스 구조에 대한 연구를 이용하였다. 워크플로우 프로세스의 구조는 분기와 병합의 형태에 따라 몇 가지 유형으로 구분된다. 이 구분을 위해서 본 연구에서는 블록(block) 개념[4]을 도

입한다. 블록은 크게 직렬(serial) 블록과 병렬(parallel) 블록으로 나누어진다. 직렬 블록은 분기와 병합이 발생하지 않는 하나의 경로를 가지는 구조이다. 병렬 블록은 분기 단위업무( $a_s$ )로부터 병합 단위업무( $a_M$ )까지 두 단위업무 사이에 다수의 경로가 존재하는 구조이다

병렬 블록은 다시 AND 블록과 OR 블록으로 나눌 수 있다. AND 블록은 모든 경로를 동시에 진행하는 구조인 반면, OR 블록에서는 분기된 여러 경로 중 하나 이상만 수행하면 된다. 즉, AND 블록은 모든 경로를 완료해야 전체 구조를 완료했다고 하고, 하나의 경로라도 완료하지 못하면 전체 구조를 완료하지 못했다고 한다. 반면에 OR 블록은 하나의 경로라도 완료하면 전체구조를 완료했다고 하며, 모든 경로를 다 완료할 수 없을 때 전체 구조를 완료할 수 없다고 말한다. OR 블록은 분기의 의미에 따라 다시 몇 가지 종류로 나눌 수 있다. OR블록의 유형은 일반 OR (NOR: Normal OR), 우선 순위 OR (POR: Priority OR), 조건 OR (COR: Conditional OR), 배타적 OR (XOR: eXclusive OR) 등으로 분류될 수 있다.

### 3. 워크플로우 접근법

#### 3.1 워크플로우 기반 접근법

WFMS(Workflow Management System)는 조직의 구조를 리엔지니어링하고 비즈니스 프로세스를 능률적이고 자동화하는 기반이 된다 [3]. 많은 연구자들은 workflow가 웹 기반에서 정보를 교환하고

프로세스를 연동하는 표준이 되는 컴퓨팅 모델이라고 규정하고 있다 [5]. 워크플로우 기반 접근법에서 웹 서비스를 컴포지션할 때, QoS를 고려하는 방법은 다음과 같은 순서로 진행된다. 먼저 프로세스의 quality를 규정하는 평가기준을 선정한 후, 각 태스크들의 수행 시간, 수행 비용, 신뢰도와 같은 quality를 추정한다. 그리고 추정된 태스크들의 quality를 바탕으로 프로세스의 quality를 추정하여 웹 서비스 프로세스의 quality를 판단한다. 태스크들의 quality로부터 프로세스의 quality를 추정하는 방법은 프로세스의 구조를 이용하여 이루어진다. 프로세스 내에 존재하는 직렬 및 병렬블록들을 하나의 태스크로 압축(Reduction)하는 작업을 반복 수행하여 프로세스의 quality를 추정한다. SWR(Stochastic Workflow Reduction) 알고리즘은 프로세스의 quality를 추정하기 위해 미리 정의한 프로세스 구조들을 이들의 하나의 태스크로 압축(Reduction)하도록 하는 접근법으로서 본 연구에서 채택하였다.

SWR(Stochastic Workflow Reduction) 알고리즘을 적용하기 전에, 프로세스 내에 있는 태스크들의 quality dimension을 추정하는 방법이 필요하다. J. Cardoso가 태스크들의 quality를 추정하는 방법을 제시하였는데 이는 표 1과 같이 정리해 볼 수 있다.

$QoS_{Dim}(t)$ 는 기존의 데이터가 없는 경우 태스크의 QoS를 추정하는 값이고,  $QoS'_{Dim}(t)$ 는 동일한 workflow가 아닌 다른 workflow에서 태스크가 수행되었을 때 QoS를 추정하는 경우이다.  $QoS_{Dim}(t, w)$ 는 동일한 workflow의 동일한 태스크가 수행된 경우 사용되는 추정값이고,  $QoS_{Dim}(t, w, i)$ 는 loop를 포함하는 동일한 workflow에서의 추정할 수 있

(표 1) 각 태스크에 대한 QoS 추정요소

Task	QoS 추정요소
$QoS_{Dim}(t)$	Designer average $_{Dim}(t)$
$QoS'_{Dim}(t)$	$((w_1 \times \text{Designer average}_{Dim}(t)) + (w_2 \times \text{multi workflow average}_{Dim}(t)))$
$QoS_{Dim}(t, w)$	$((w_1 \times \text{Designer average}_{Dim}(t)) + (w_2 \times \text{multi workflow average}_{Dim}(t)) + (w_3 \times \text{workflow average}_{Dim}(t,w)))$
$QoS_{Dim}(t, w, i)$	$((w_1 \times \text{Designer average}_{Dim}(t)) + (w_2 \times \text{multi workflow average}_{Dim}(t)) + (w_3 \times \text{workflow average}_{Dim}(t,w)) + (w_4 \times \text{instance workflow average}_{Dim}(t,w,i)))$

(표 2) 5가지 SWR 적용 룰

Process Structure	Time	Cost	Reliability
Sequential Structure	$T(t_{ij}) = T(t_i) + T(t_j)$	$C(t_{ij}) = C(t_i) + C(t_j)$	$R(t_{ij}) = R(t_i) \times R(t_j)$
AND Structure	$T(t_{in}) = \text{Max}_{i \in \{1, \dots, n\}} \{T(t_i)\}$	$C(t_{in}) = \sum_{1 \leq i \leq n} C(t_i)$	$R(t_{in}) = \prod_{1 \leq i \leq n} R(t_i)$
COR Structure	$T(t_{in}) = \sum_{1 \leq i \leq n} P_{a_i} \times T(t_i)$	$C(t_{in}) = \sum_{1 \leq i \leq n} P_{a_i} \times C(t_i)$	$R(t_{in}) = \sum_{1 \leq i \leq n} P_{a_i} \times R(t_i)$
Simple Loop Structure	$T(t_{ii}) = \frac{T(t_i)}{1 - p_i}$	$C(t_{ii}) = \frac{C(t_i)}{1 - p_i}$	$R(t_{ii}) = \frac{(1 - p_i) \times R(t_i)}{1 - p_i R(t_i)}$
Dual Loop Structure	$\frac{T(t_i) + T(t_j) - (1 - p_j)T(t_j)}{1 - p_j}$	$\frac{C(t_i) + C(t_j) - (1 - p_j)C(t_j)}{1 - p_j}$	$\frac{(1 - p_j)R(t_j)}{1 - p_j R(t_i)R(t_j)}$

는 값이다.

태스크들의 quality를 추정하는 작업이 끝나면, SWR (Stochastic Workflow Reduction) 알고리즘을 이용하여 웹 서비스 프로세스의 QoS를 산출해야 한다. 대표적인 5가지의 프로세스 구조에 대해서 SWR 알고리즘을 적용한 룰은 표 2와 같다.

행 비용(Cost)에 대해서는 태스크들의 수행 시간과 수행 비용의 합으로 압축(Reduction)되고, 신뢰도(Reliability)에 대해서는 그들 신뢰도의 곱으로 표현된다.

- AND Structure: 태스크들의 수행 시간 중에 최대 수행 시간이 AND Structure의 수행 시간이 된다. 수행 비용(Cost)은 태스크들의 수행 시간과 수행 비용의 합으로 간주되고, 신뢰도(Reliability)는 그들 신뢰도의 곱으로 표현된다.
- COR Structure: 수행 시간의 경우 태스크들이 수행될 확률과 태스크들의 수행 시간을 곱한 후 더한 값이 COR Structure의 수행 시간이 된다. COR의 수행 비용(Cost), 신뢰도(Reliability)를 계산하는 방법은 수행 시간을 계산하는 방법과 동일하다.
- Dual Loop Structure: Dual Loop Structure의 경우 dual loop가 이루어지는 두 태스크들의

수행 시간의 합에서 loop가 이루어지지 않았을 때의 수행 시간을 뺀 후, loop가 이루어지지 않을 확률로 나눈 값이 Reduction된 수행 시간이 된다. 수행 비용은 수행 시간과 동일하게 계산되고, 신뢰도는 loop가 발생하지 않을 확률과 태스크의 신뢰도를 곱한 후, loop가 발생하였을 때 시스템이 다운될 확률로 나눈 것이 Reduction된 신뢰도가 된다.

이상에서 본 접근법은 프로세스 구조에 따라 quality를 계산할 수 있기 때문에, 구조에 상관없이 적용이 가능하다는 장점이 있다. 하지만, 태스크를 수행해 줄 수 있는 서비스 공급자의 수가 증가할 때 적용하기 어렵고 시뮬레이션을 통해 최적의 서비스 공급자를 찾아 내기 때문에 가장 좋은 서비스 공급자를 선정할 수 없다는 단점이 있으므로 제외할 수 있다. Cardoso는 이상의 5가지 구조 이외에도 Fault-Tolerance System, Network System등에도 적용하였다 [3].

### 3.2 수리적 접근법

프로세스 기반의 웹 서비스 컴포지션은 이기종의 분산된 어플리케이션의 통합에 효과적인 접근

방법이다 [11]. 그러나 UDDI 레지스트리에는 동일한 기능을 제공하는 수많은 웹 서비스 공급자들이 존재하고 있으며 그들의 등록은 실시간으로 변화하고 있다. 그러므로 복합 웹 서비스를 실행하는 동안 QoS를 고려하여 서비스 공급자를 실시간으로 선택하는 문제는 여전히 어려운 문제로 남아있다 [6]. 웹 서비스 공급자를 선택하는 문제는 서비스 공급자가 선택되거나 선택되지 않는 문제이므로, AND structure와 XOR structure의 프로세스 구조만 고려하면 된다. 수리적 접근법은 웹 서비스 공급자를 선택하는 문제를 이 두 프로세스 구조에 대해서 어떻게 포물레이션 하는지를 보여주고자 하였다. Zeng[11]의 연구에 따르면 우선 XOR 분기는 분기를 분리하여 가능한 경로로서 정의될 수 있고, 분리된 경로의 선택 여부는 XOR 분기 이전의 태스크에 의해서 결정된다고 가정하였다. 그림 3는 웹 서비스 프로세스 내에 존재하는 XOR structure를 두 개의 execution paths로 분리한 예를 DAG (Directed Acyclic Graph)를 이용하여 나타낸 그림이다.

Zeng은 'execution path'내에 존재하는 태스크들을 수행해 줄 수 있는 서비스 공급자들의 조합을 'execution plan'으로 정의하였다. 따라서 Zeng의 연구는 웹 서비스를 컴포지션할 때 최적의 서비스 공급자를 선택하는 문제를 최적의 execution plan을 찾는 문제로 바꾼 것이 된다. 그는 각 execution plan의 QoS는 선형 관계식으로 표현하였고, 표현된 식을 scaling 작업을 거친 후, 선형 계획법(Liner Programming)을 이용하여 최적의 execution path를 찾아낼 수 있다.

Zeng의 접근방법은 global optimization을 보장하고, 선형 형태로 포물레이션이 가능한 QoS 평가 기준에 대해서는 일반적인 접근법이라는 장점이 있다. 하지만, XOR structure를 다수의 execution paths로 분리할 때 분기 이전에 수행되는 태스크에 의해서 결정된다는 가정은 다른 프로세스 구조에 대해서는 일반적이지 못하다.

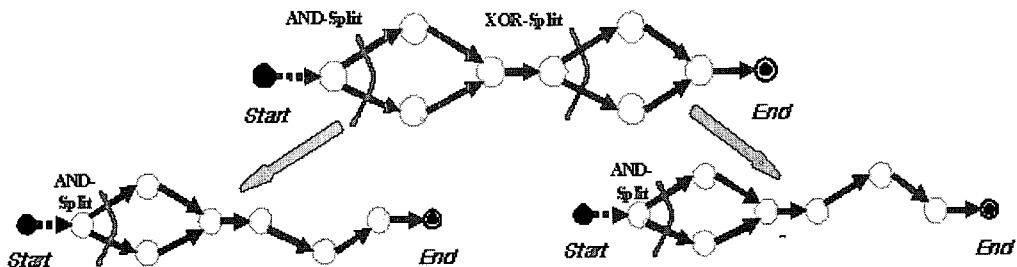
#### 4. 웹 서비스 모델

구조정보에 기반하여 다음과 같이 웹 서비스를 정의한다.

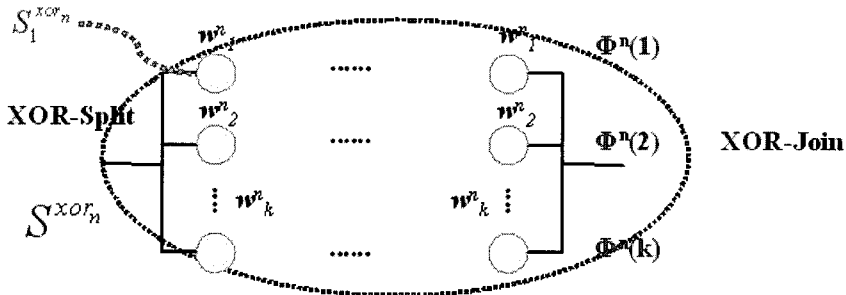
**Definition 1.** n번째 XOR 집합( $S^{xor_n}$ )에서 k번째 경로 내에 있는 모든 웹 서비스 태스크들을  $\Phi_n(k)$ 로 정의한다.

그림 4를 보면 n번째 XOR 집합인 ( $S^{xor_n}$ )의 경우에는 k개의 경로가 존재한다. 그리고 각 경로들이 선택될 지의 여부는  $w_k^i$ 로 정의한다. 즉, MBED Equation.DSMT4  $w_k^i$ 는 n번째 XOR 집합에서 k번째 경로가 선택될 지의 여부이고, 이 값은 0 (선택 안됨) 또는 1(선택됨)로 표현될 수 있다.

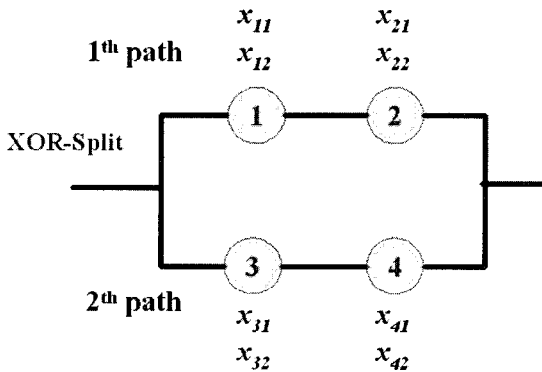
그림 5는 간단한 simple XOR structure의 예를 보여주고 있다. 그림에서는 2개의 경로가 존재하며, 각 경로에는 두 개의 태스크가 있다. 그리고 각각의 태스크에는 이를 수행해 줄 수 있는 서비스 공급자 ( $x_{i1}, x_{i2}, \dots, x_{i42}$ )가 두 개씩 존재한다는 것이다.



(그림 3) DAG representation of the execution paths



(그림 4) XOR split 및 XOR-join



(그림 5) Simple XOR structure

구축하였을 때, 서비스 공급자를 선택하는 문제를 도식화 하였다. 예를 들면, 대금회수를 하는 태스크가 있다면, UDDI내에 존재하는 수 많은 단일 웹 서비스 공급자들(A, B, C, D)중에서 어떤 서비스 공급자를 선택해야 하는지를 보여주고 있다.

## 5.2 웹 서비스 QoS 요구사항

본 논문에서는 웹 서비스의 파트너를 선택하는 평가기준으로 고객이 요구하는 서비스의 품질(QoS)를 이용하였다. 2002년도에 IBM에서 제시한 ‘웹 서비스 QoS 요구사항’[6]에는 웹 서비스 프로세스의 시간, 비용, 신뢰도 등과 같은 비 기능적 특성까지도 고려하고 있다. 본 연구에서는 ‘웹 서비스 QoS 요구사항’ 중에서 정량인 평가가 가능한 5가지를 고려하였다.

## 5. 웹 서비스를 위한 QoS

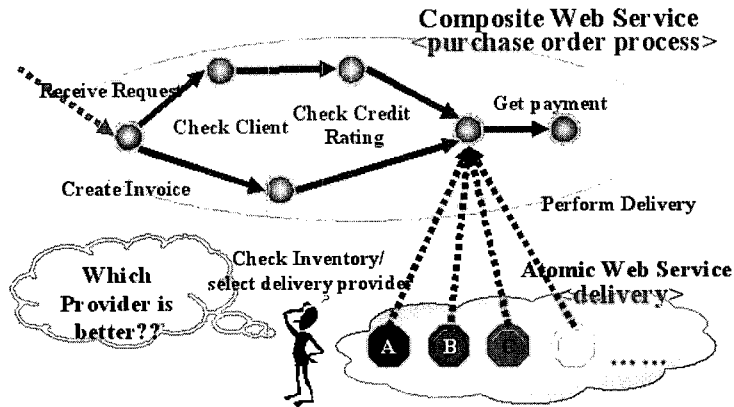
### 5.1 웹 서비스 문제의 정의

복합 웹 서비스를 컴포지션 한다면, 웹 서비스 프로세스 내의 태스크를 수행할 수 있는 다수의 서비스 공급자를 UDDI에서 찾을 수 있다. 본 연구에서는 동일한 기능을 제공해 주는 다수의 서비스 공급자 중에서 누구를 선택할 지의 기준으로 비 기능적인 요소인 서비스의 품질을 고려하였다.

만약, 웹 서비스 프로세스 내에 N개의 태스크가 존재하고, 각 태스크들에는 M개의 서비스 공급자가 있다고 가정한다면, 웹 서비스 공급자를 선택하는 모든 경우의 수는  $M^N$ 개가 된다. 모든 가능한 경로를 고려하여 태스크 별 최적의 서비스 공급자를 선택한다면  $O(M^N)$ 이 된다.

그림 6은 구매프로세스를 웹 서비스 기반으로

- 수행시간(Execution Duration) - 웹 서비스 고객이 웹 서비스 공급자에게 서비스를 요청한 후, 응답을 받을 때까지 걸리는 시간이다. 그러므로 프로세스 수행시간은 요청하는데 걸리는 시간과 서비스를 수행하는 시간, 그리고 서비스의 결과를 보내는 데 걸리는 시간의 합으로 생각할 수 있다.
- 수행 비용(Execution Cost) - 수행비용은 웹 서비스를 수행하는데 드는 금전적 비용을 수행 비용으로 정의한다.
- 신뢰도(Reliability) - 신뢰도란 웹 서비스를 요청하였을 때, 임의의 예상 시간 내에 요청의



(그림 6) 웹 서비스 기반의 구매 프로세스에서의 서비스 공급자 선택

결과를 받을 수 있는 확률이라고 정의한다. 따라서 웹 서비스의 신뢰도는 서비스 고객과 공급자들간의 메시지 전송을 보장해주는 정도로 표현될 수 있다.

- 가용성(Availability) - 웹 서비스의 가용성은 웹 서비스를 현재 또는 즉시 사용할 수 있는지를 평가하는 기준이다. 식(1)에서 보는 것처럼 가용성(Availability)은 특정 관측 시점 동안 서비스를 수행할 수 있는 시간의 비율로 표현된다.

$$Availability = \frac{\langle Up\ Time \rangle}{\langle Total\ Time \rangle} = \frac{\langle Up\ Time \rangle}{(\langle Up\ Time \rangle + \langle Down\ Time \rangle)} \quad (1)$$

- 평판도(Reputation) - 웹 서비스의 평판도는 서비스의 최종 사용자의 경험치에 의거하여 서비스의 신뢰도를 평가하는 요소이다. 본 연구에서는 웹 서비스의 평판도를 최종 사용자의 평가들의 평균으로 정의하였다 식(2).

$$Reputation = \frac{1}{n} \sum_i^n \langle user's\ Rating \rangle \quad (2)$$

### 5.3 AND 구조

AND structure에서도 sequential structure와 동

일하게 모든 태스크들이 수행되어야 한다. 그러므로 AND structure에서의 QoS Computation은 sequential structure와 동일하다. 단, AND structure에서는 수행시간이 critical path에 의해 좌우 되므로, PERT/CPM을 사용한다.

### 5.4 XOR 구조

XOR structure는 XOR structure내에 또 다른 병렬 구조가 중첩되어 있는지의 여부에 따라 simple XOR structure와 nested XOR structure로 구분하였다. XOR structure에서의 QoS계산은 Sequential Structure와 동일하게 계산될 수 있다. 그러나, 웹 서비스를 구성하는 사람이나 고객이 XOR분기의 가능한 경로 중 하나를 능동적으로 선택한다고 가정하였기 때문에 부가적인 제약 조건이 필요하다.

### 6. 결론

웹 서비스 어플리케이션과 이를 공급하는 서비스 공급자들의 수가 증대함에 따라 가장 좋은 서비스 공급자를 선택하는 데에서 발생하는 문제점과, 또한 단일 웹 서비스를 컴포지션하여 복합 웹 서비스를 구축하는 경우에도 웹 서비스 프로세스의 각 태스크들을 수행해 줄 수 있는 적절한 웹 서비스



어플리케이션을 선택하는데 어려움이 있다. 이를 해결하기 위한 웹 서비스 프로세스의 QoS를 고려하는 방법이 요구되며, 본 연구에서는 다양한 웹 서비스 컴포지션을 고려한 복합 웹 서비스를 구축을 위해서는 품질수준을 고려한 접근법과 모델링이 매우 유용하다는 것을 보였다. 향후에는 좀 더 정교한 수리 모델에 입각한 정량적 분석연구를 수행할 계획이다.

### 참고 문헌

- [1] Aalst, W., Hofstede, "Business Alignment: Using Process Mining as a Tool for Delta Analysis," CAiSE Workshops (2) 2004: 138-145
- [2] Basu A., Blanning, R., "Synthesis and Decomposition of Processes in Organizations," *Information Systems Research*, Vol. 14, No. 4, December 2003.
- [3] Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K., "Quality of Service for Workflows and Web Service Process," *Journal of Web Semantics*, Vol 1, Issue 3, pp. 281-308, 2004.
- [4] Dogac, A., Laleci, G., Kabak, Y., Cingil, I., "Exploiting Web Service Semantics: Taxonomies vs. Ontologies," *IEEE Data Engineering Bulletin*, Vol. 25, No.4, pp. 10-16, 2002.
- [5] Fensel, D., Bussler, C., "The Web Service Modeling Framework WSMF," *Electronic Commerce Research and Applications*, Vol. 1, No. 2, pp. 113-137, 2002.
- [6] Jaeger, M., Rojec-Goldmann, G., Mühl, G., "QoS Aggregation for Web Service Composition using Workflow Patterns," *EDOC*, pp.149-159, 2004.
- [7] Miller, G., "The Web Services Debate: .NET vs. J2EE," *Communications of the ACM*, Vol. 46, No. 6, pp. 64-67, 2003.
- [8] W3C Web Services Architecture Working Group (<http://www.w3.org/2002/ws/arch/>)
- [9] Williams, J., "The Web services debate: J2EE vs. .NET," *Communications of the ACM*, Vol. 46, No. 6, pp. 58-63, 2003.
- [10] Yang, J., Papazoglou, M., "Service Components for Managing the Life-Cycle of Service Compositions," *Information Systems*, Vol. 29, No. 2, pp. 97-125, 2004.
- [11] Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H., "QoS-Aware Middleware for Web Services Composition," *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 311-327, 2004.

## ◎ 저 자 소 개 ◎

### 이 우 기

1987년 서울대학교 산업공학과 (공학사)  
1993년 서울대학교 대학원 산업공학과 (공학석사)  
1996년 서울대학교 대학원 산업공학과 (공학박사)  
2000년 Carnegie Mellon University MSE 과정수료  
2002년~2003년 University of British Columbia 교환교수  
2004년 한국경영정보학회 이사  
1996년~현재 성결대학교 컴퓨터학부 부교수  
1998년~현재 노동부 전자전문위원  
2004년 10월 한국경영과학회에서 최우수논문상 수상  
관심분야 : 웹 정보검색, 데이터웨어하우스, E-business  
Email : wook@sungkyul.edu

### 조 대 래

2003년 부산대학교 (학사)  
2005년 서울대학교 (석사)  
2005년~현재 현대자동차 재직  
관심분야 : 웹 정보검색, E-business  
Email : drcho@snu.ac.kr

### 강 석 호

1970년 서울대학교 (학사)  
1972년 University of Washington (석사)  
1976년 Texas A&M University (박사)  
1976년~현재 서울대학교 공과대학 교수  
1991년~1997 한국 O.R. 학회 부회장/감사/차기회장/회장  
1999년~현재 아시아 태평양 O.R. 학회 부회장  
1993년~현재 행정고시, 외무고시, 공장관리기술사 출제위원  
1996년~현재 산업자원부 자동화 부문 심사위원  
관심분야 : MIS, BPM, E-business  
Email : shkang@snu.ac.kr