웹 서비스 지원을 위한 예외처리 기반 동적 서비스 조정 프레임워크

정종하° 장우혁 이성독 한동수
한국 정보통신 대학원
{jongha°, dshan, torajim}@icu.ac.kr

Exception based Dynamic Service Coordination Framework for Web Services

Jongha Jung° Dongsu Han woohyuk Jang
Information Communication University

요 약

웹 서비스는 서비스 가용성(availability)과 성능(performance)면에서 신뢰성을 항상 보장해 주지 못한다. 동적 서비스 조정(Dynamic Service Coordination)은 웹 서비스를 호출하는 시스템이나 응용 프로그램 내에서의 비 신뢰적인 문제상황을 처리하는 기술이다. 이 환경 내에서의 웹 서비스는 제한된 시간 내에 응답 하지 못하는 등의 문제가 발생 할 경우, 신뢰적인 웹 서비스의 호출을 위해 다른 웹 서비스로의 대체작업이 런타임(run-time)에 이뤄진다. 본 논문에서는 웹 서비스를 위한 동적 서비스 조정 프레임워크를 제안한다. 프레임워크 내에서 동적 서비스 조정의 지원 및 웹 서비스 호출을 담당하는 클래스와 워크플로우가 생성되고, 생성된 클래스 메소드를 호출함으로써 신뢰적인 웹 서비스 호출이 가능하다. 호출작업이 간접적으로 이뤄짐으로 인해 어느 정도의 성능적 손실이 발생하나, 이 방식을 통해 얻는 시스템 유연성과 신뢰성의 측면을 고려한다면, 충분히 감 수 될 수 있다.

1. 서 론

웹 서비스는 각각의 서비스가 느슨히 연결된 표준 기반의 프로세스 중심 컴포넌트 서비스이다[1]. 이 같은 특성은 웹 서비스 기반 응용프로그램의 개발 과정에서 보다 다양한 웹 서비스 소스를 선택할 수 있도록 하며, 그것들 서로간의 연결과 조정을 용이하게 한다.

또한 웹 서비스의 SOAP(Simple Object Access Protocol)[2], WSDL(Web Services Description Language)[3], UDDI(Universal Description Discovery and Integration)[4], HTTP(Hypertext Transfer Protocol)[5]등의 웹 표준 언어 및 프로토콜을 통해 광범위한 서비스의 지원이 가능하다.

하지만 웹 서비스는 웹 서비스 자체로의 접근이 불가능한 경우나 일정 시간 안에서의 응답을 보장하지 못하는 경우, 또는 잘못된 결과값을 반환하는 경우와 같은 서비스 가용성(availability)과 성능(performance)상의 측면에서는 신뢰성을 항상 보장하지 못한다.

웹 서비스 환경에서는 현재 자신이 제공하는 서비스와 비슷하거나 동일한 서비스를 지원하는 사이트가 존재하는 것을 가정한다. 이 가정에 기반하여, 문제점을 지닌 기존 서비스의 대체작업을 통해 신뢰적인 서비스의 제공이 가능하다. 동적 서비스 조정(Dynamic Service Coordination)은 문제가 발생하는 기존 웹 서비스를 그것과 동일한 서비스를 제공하는 웹 서비스로 대체하는 작업을 런타임(run-time)에 진행하는 기술이다.

본 논문은 웹 서비스 상에서 동적 서비스 조정을 지원하는 프레임워크를 제안한다. 프레임워크내의 모든 정보는 속성의 집합(attribute set)으로 명시되며, 이 정보를 바탕으로 동적 서비스 조정 및 웹 서비스 호출을 담당하는 클래스와 워크플로우가 자동 생성된다. 이것으로 클라이언트 프로그램의 개발자는 생성된 클래스내의 메소드를 이용하여 신뢰성 있는 웹 서비스 호출이 가능하다.

본 논문에서 제안한 방식은 웹 서비스의 호출이 간접적으로 이뤄짐으로 인해 웹 서비스 호출과정에서 필연적으로 약간의 성능상 손실을 유발한다. 워크플로우를 이용하여 웹 서비스를 호출하는 경우의 성능 손실을 측정한 결과, 평균 0.5초정도가 지연되는 것을 확인하였다. 이러한 성능 손실은 제안한 방식을 통해 얻는 시스템 유연성(flexibility)과 신뢰성(reliable)의 측면을 고려한다면, 많은 응용에 있어서 충분히 감수될 수 있을 것으로 예상되는 만큼 향후 제안된 방식의 폭 넓은 활용이 기대된다.

본 논문의 구성은 다음과 같다. 2장에서 동적 서비스 조정의 일반적인 개념을 설명하고 3장에서는 동적 서비스 조정 프레임워크를 설명하고, 4장에서 예외상황 처리 메커니즘, 5장에서 성능 평가를 다루며 6장에서 결론을 맺는다

2. 웹 서비스를 위한 동적 서비스 조정

동적 서비스 조정은 관련 속성(attribute)값이 런타임(run-time)에 변경이 가능하도록 지원하는 워크플로우의 동적 재설정(Dynamic reconfiguration)에 기반한다. 동적 서비스 조정 역시 프로그램 내에서 미리 정의된 웹 서비스가 적절하게 동작하지 못하는 경우, 다른 웹 서비스로의 대체가 런타임(run-time)에 일어난다.

웹 서비스 환경이 여러 타입으로 나뉘고 또한 계속적인 변화하고 있지만, 동적 서비스 조정 기술을 응용프로그램 상에 적용시키는 방법으로는 크게 두 가지가 있다.

UDDI는 후보 웹 서비스를 찾고 런타임(run-time)에 동적 서비스 조정을 위한 접근포인트(access point)를 변경한다.

보 서비스의 위치를 쉽게 제공한다는 매력이 있다. 하지만 실제로 현재 UDDI버전이 완벽하지 못하며, 동적 서비스 조정을 지원하기 위한 UDDI 표준 또한 미흡한 상태다.

두 번째로는 워크플로우에서 사용되는 예외상황 핸들링 메커니즘을 이용하는 것이다. 웹 서비스가 적절히 응답하지 못할 때를 예외상황(exception)으로 규정한다.

대체 웹 서비스를 검색하고 검색된 것을 호출함으로 해당 예외상황을 처리하는 과정을 거친다

이 방법은 몇 가지 이점을 지닌다. 그 중 하나로 웹 서비스 호출작업이 워크플로우를 통해 이루어지기 때문에, 워크플로우 시스템상의 많은 장치가 웹 서비스상의 트랜젝션 문제, 예외상황 처리 메커니즘 등에 이용가능 하며, 웹 서비스 상에서 신뢰성 있는 호출을 지원하기 위해 사용될 수 있다 이 같은 이유로 본 논문은 이 방법을 택한다

3. 동적 서비스 조정 프레임워크

3.1 동적 서비스 조정을 지원하기 위한 접근 방법

UDDI를 사용하지 않는 응용프로그램상에서 동적 서비스 조정을 지원하기 위해 상태 체크루틴, 웹 서비스 호출 및 선택 루틴등 몇 가지 추가 루틴의 첨가가 요구된다. 유동적으로 변화하는 웹 서비스 환경에서 이 같은 루틴을 응용 프로그램으로 즉각 반영 및 관리되기 위해 웹 서비스 호출 응용프로그램과 동적 서비스 조정모듈을 분리하여 관리하는 구조를 택한다. 그럼으로써 웹 서비스가 호출되기 이전에 미리 신뢰성 있는 웹 서비스 호출 디자인 작업이 가능하다 그림1 은 이 두 부분을 분리한 구조를 나타낸다

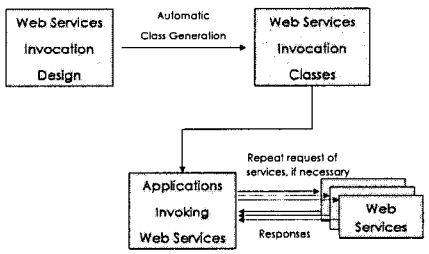

그림 1. 웹 서비스 호출 클래스와 동적 서비스 조정

하지만, 그림 1의 경우, 새로운 웹 서비스를 반영하고 서비스 후보리스트에서 해당 웹 서비스를 제거하는 작업을 위해 계속적인 클래스 재 컴파일 과정과 클래스 생성과정이 필요하다.

웹 서비스 호출 시 발생하는 약간의 성능적 손실을 어느 정도 수용할 수 있다면, 더 유연한(flexible)서비스 지원이 가능하다. 그림 2는 본 논문의 프레임워크 동작 시나리오를 나타낸다. 이 과정을 통해 생성된 워크플로우는 웹 서비스 호출을 담당했던 응용 프로그램을 대신해 웹 서비스를 호출한다. 즉, 워크플로우 호출 클래스를 생성하고 생성된 클래스의 인스턴스를 호출함으로 워크플로우 인스턴스가 동작하게 됨을 의미한다. 위에서 언급한 바와 같이 워크플로우 때문에 이상적인 경우, UDDI는 정해진 시간 내에 최선의 후 시스템 장치들을 이용함으로써 웹 서비스 환경내의 변화되는 작업이 워크플로우 시스템의 예외상황 핸들링 메커니즘을 통해 쉽게 명세화되고 제어된다. 즉, 웹 서비스 환경의 변화가 응용프로그램에 영향을 미치지 않은 체 즉각 반영되며, 응용프로그램의 재 컴파일과정과 재 생성 과정 또한 불필요한 이점을 지닌다

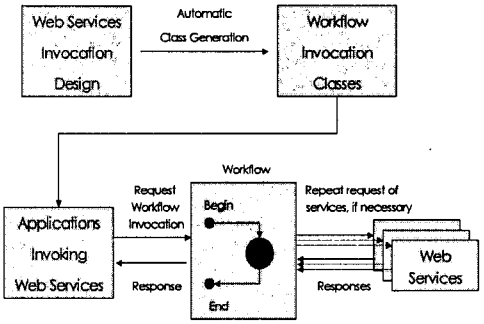

그림 2. 클래스 호출 워크플로우와 동적 서비스 조정

4. 워크플로우 시스템의 예외 상황 핸들링 메커니즘

그림 3는 응용 프로그램상의 워크플로우 호출 클래스를 이용한 웹 서비스를 호출 과정 중, 타임아웃 예외상황(time-out exception)의 진행 단계적 흐름을 나타낸다.

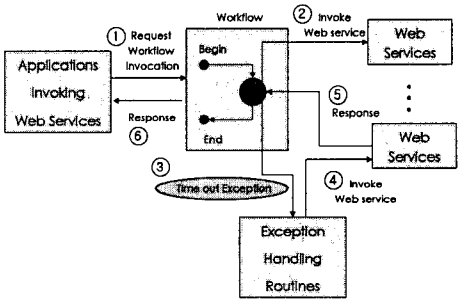

그림 3. 예외상황 메커니즘을 이용한 동적 서비스 조정

타임아웃 예외상황이 발생한 후, 워크플로우 시스템은 핸들링 루틴을 호출하고, 호출된 핸들링 루틴은 대체 웹 서비스를 호출한다. 대체 웹 서비스 호출작업의 성공 시 결과값을 워크플로우를 통해 응용 프로그램으로 전달하고, 실패 시, 또 다른 서비스의 호출을 시도한다.

4.1 예외상황 핸들링 메커니즘을 동적 서비스 조정으로 적용

워크플로우 환경에서 동적 서비스 조정을 표현하기 위해 필

필요한 주요 요소는 다음과 같다

- *Activity(엑티비티)* - 동적 서비스 조정을 활성화 하는 예외상황을 의미한다
- *Transitions(전이)* - *ExceptionalTransition*요소(element)는 예외상황이 발생한 경우, 동적 서비스 조정 단계로 이르는 길을 명시한다
- *Exceptions(예외상황)* - 동적 서비스 조정을 동기화하는 예외상황의 집합이 정의되며 각각의 예외상황은 Exception요소(element)로 표현된다.
- *GuardedBlocks(감시구역)* - 예외상황이 발견되고, 동적 서비스 조정 단계가 활성화되는 영역을 명시한다. From, To 이 두 속성으로 활동범위를 정한다

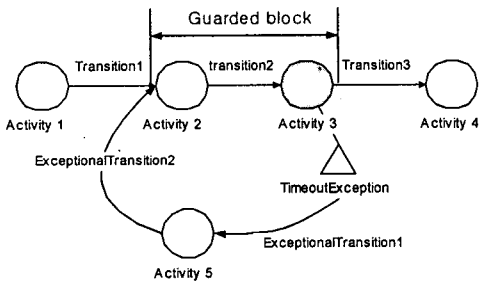

그림 4. 감시구역내의 예외상황진행 시나리오

그림 4에서 동적 서비스 조정을 지원하기 위한 예외상황 핸들링 메커니즘을 설명한다
TimeoutException을 감시구역(guarded block)에 첨가한다. TimeoutException은 엑티비티3(Activity3)이 실행되는 도중 일어나며, ExceptionTransition1으로 지정된 경로를 따라 제어가 이동하여 예외상황 핸들러(동적 서비스 조정자)Activity5를 호출한다 (그림4에서 원:엑티비티, 삼각형: 예외상황을 뜻한다).

5. 성능 평가

동적 서비스 조정을 통해 웹 서비스는 간접 호출되기 때문에 웹 서비스를 직접 호출하는 것과 비교 시 성능상의 손실이 있을 수 있다. 본 논문에서는 성능적 손실의 정도를 보다 정확히 이해하기 위해 웹 서비스 호출의 경과시간을 각기 다른 방법을 이용해 측정하였다 - 웹 서비스와 응용프로그램이 동일컴퓨터에 있는 경우와 LAN환경 내에 있는 경우, 그리고 인터넷 환경에서 웹 서비스를 원격 호출하는 경우다. 측정된 결과값은 표1에 나타내었고, 요청(request)에 대한 응답(respond)시간만을 고려한 결과값이다. 괄호안의 값은 표준편차이다.

표 1. 웹 서비스 호출을 위한 평균 대기시간 (단위: 초)

웹 서비스의 위치	직접호출	간접 호출
동일 PC	0.010(0.008)	0.55(0.091)
LAN환경	0.081(0.020)	0.53(0.086)
WAN(원격호출)환경	0.100(0.045)	0.57(0.087)

표1에서 주목할 사항은 원격환경에서 간접 호출한 결과값(0.57)이 동일 컴퓨터 환경에서 간접 호출(0.55)한 결과값에 비해 더 작다는 것이다. 웹 서비스와 워크플로우 시스템이 동일 장소에 상주하는 것이 그 원인인데, 많은 리소스(resource)가 워크플로우 시스템의 가동에 할당되기 때문이다
표1의 결과에서 워크플로우를 통해 웹 서비스를 호출하는 작업의 경우, 평균 약 0.5초의 지연시간이 필요함을 알 수 있다.

6. 결론 및 향후 과제

웹 서비스기술의 보급으로 비즈니스 유저는 웹 서비스들로부터 얻어지는 기업 내. 외부에 적합하고 다양한 응용 프로그램을 더 빠르게 조합할 수 있을 것으로 기대한다.
이와 같이 웹 서비스 기술의 미래는 밝다. 하지만 본 논문에서 언급한 바와 같이 인터넷상에서 웹 서비스는 서비스 가용성(availability)과 성능(performance)적인 면에서 신뢰성을 항상 보장하지는 못한다.
이 같은 문제를 해결하고자 본 논문에서는 웹 서비스를 위한 예외상황에 기반한 동적 서비스 조정 방법을 제안하였다. 웹 서비스 호출 클래스와 워크플로우 호출 클래스를 신뢰성 있는 웹 서비스의 호출을 위해 사용하였고, 시스템 구현을 위해 워크플로우 시스템의 예외상황 핸들링 메커니즘을 이용하였다.
이러한 웹 서비스의 간접적 호출방법이 시스템 성능상 어느 정도의 손실을 가져옴을 확인하였다. 하지만 오랜 시간 실행되는(long-term)업무 프로세스가 웹 서비스상의 주요 서비스라는 웹 서비스의 특성을 고려할 때, 이 정도의 성능적 손실은 충분히 수용될 수 있다.
UDDI는 동적 접근 포인트(dynamic access point)관리에 유용하다고 잘 알려져 있다. 때문에 본 논문의 동적 서비스 조정이 UDDI의 이 기능과 결합한다면 더욱 개선된 동적 서비스 조정이 가능할 것이다. 진보된 UDDI의 주요특징과 본 논문에서 제안한 동적 서비스 조정 프레임워크를 통합하는 것을 향후 과제로 한다.

참 고 문 헌

[1] " A Critical Foundation for Service - Oriented Development and Web Services " , http://www.wakesoft.com/product/WhitePapers.html,The Stencil Group and Wakesoft, August 2003
[2]Gudgin, M, World Wide Web Consortium Recommendation, " http://www.w3.org/TR/soap/"
[3] Christensen, E., Curbera, F., Meredith, G., Weerawarana, World Wide Web Consortium note, " http://www.w3.org/TR/2001/NOTE-wsdl-20010315, 2001"
[4]Ehnebuske, D., McKee, B., Rogers, " http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm" , 2002
[5] Han, D., Goo, J-Y., Song, S-D., Lee, S-D., Seo, B-S, Design of a Web Services Based eAI Framework,6th International Conference on Advanced Communication Tecnnology (ICACT 2004), 2004