BPEL4WS 명세의 효율적인 실행 방안

정종윤[0] 류기열 이정태
아주대학교 정보통신전문대학원
{jongyun, kryu, jungtae}@ajou.ac.kr

Efficient Execution Method of BPEL4WS Specification

Jong-Yun Jung[0] Ki-Yeol Ryu Jung-Tae Lee
Graduate School of Information and Communication, Ajou University

요 약

현재 웹 서비스의 결합을 기술하기 위한 다양한 방법들이 제시되고 있지만 XLANG과 WSFL의 장점을 수용한 BPEL4WS로 통합되고 있는 추세이다. BPEL4WS는 비즈니스 프로세스에 참여하는 웹서비스들간의 상호동작과 프로세스의 상태 정보를 기술할 수 있는 효과적인 방법을 제공한다. 그러나, BPEL4WS로 기술된 명세를 웹서비스화 하기 위한 실행 방안에 대한 연구는 미흡한 실정이다. 이에 본 논문은 비즈니스 프로세스에 내재된 워크플로우 그래프로부터 간략화 된 실행그래프의 생성 방법을 제안한다. 또한 실행 그래프를 이용하여 BPEL4WS 명세의 병렬 및 분산 실행을 지원하는 구현 방안에 대해 논의한다.

1. 서 론

기존 서비스의 결합을 통해 새로운 서비스를 작성하는 웹 서비스 결합에 대한 관심과 필요성이 계속 증가하고 있다. 현재 웹 서비스 결합을 기술하기 위한 다양한 방법들이 제시되었지만 마이크로소프트의 XLANG[3]과 IBM의 WSFL(Web Service Flow Language)[4]의 장점을 수용하고 있는 BPEL4WS(Business Process Execution Language for Web Service)[5]로 통합되고 있는 추세이다. BPEL4WS는 비즈니스 프로세스를 기술하는 고수준의 언어로 비즈니스 프로세스에 참여는 웹서비스들간의 상호작용을 워크플로우 형태로 기술할 수 있다. BPEL4WS 명세에 기술된 비즈니스 프로세스는 BPEL4WS의 Activity들로 구성된 워크플로우를 포함한다. 즉, 결합에 참여하는 기존 웹 서비스간의 상호동작은 워크플로우 패턴들의 조합으로 표현이 가능하다.

BPEL4WS 명세에 기술된 비즈니스 프로세스는 웹 서비스-결합 웹 서비스-화되어 다시 다른 비즈니스 프로세스에 포함되어 질 수 있다. BPEL4WS 명세를 실행하기 위해서는 비즈니스 프로세스에 참여하는 웹서비스들간의 상호동작을 위해 각 서비스에 필요한 메시지를 전송하고 상태 정보를 관리하는 것은 필수적인 요소이다. 결국 BPEL4WS 명세를 실행하기 위한 필수요소는 결합에 참여하는 서비스간의 상호동작에 대응되는 워크플로우를 실행하는 것이다. 그러나, BPEL4WS로 기술된 비즈니스 프로세스에 참여하는 기존 서비스들간의 상호동작을 지원하는 실행 방안에 대한 연구는 미흡한 실정이다.

이에 본 논문은 BPEL4WS로 기술된 비즈니스 프로세스에 내재된 워크플로우 그래프로부터 간략화 된 실행 그래프를 생성하는 방법을 제안한다. 제안된 실행 그래프에 기반하여 BPEL4WS 명세를 효율적으로 실행할 수 있는 구현 방안과 병렬 실행 및 분산 실행의 가능성에 대해 논의한다. 2장에서는 관련연구에 대해서 언급하고 3장에서는 워크플로우 그래프에서 실행 그래프를 생성하는 방법에 대해서 기술한다. 4장에서는 실행그래프를 이용한 결합 프레임웍의 구현 방안에 대해서 기술하며 5장의 결론 및 향후 연구로 마친다.

2. 관련 연구

2.1 BPEL4WS

BPEL4WS는 WS-I(Web Services Interoperability Organization)가 표준화를 추진하고 있으며 현재 웹 서비스 결합 언어는 점차 BPEL4WS로 통합되는 추세이다. BPEL4WS는 비즈니스 관점에서 웹서비스의 결합을 기술하는 고수준의 언어로 기업 간의 비즈니스 프로토콜을 표현할 수 있다. 비즈니스 프로세스는 하나의 트랜잭션을 완료하기 위해 긴 시간을 필요로 하는 경우가 많으며 이를 위해 BPEL4WS는 비즈니스 프로세스에 대응하는 결합 웹서비스의 상태 정보를 기술할 수 있다. 그림 1은 BPEL4WS로 기술된 명세들의 일반적인 구조를 나타내며 결합에 참여하는 웹 서비스간의 상호작용은 Activity 부분에 기술된다. BPEL4WS에 정의된 Activity는 모두 기본 Activity와 구조화 Activity로 분류되며 실행 그래프의 각 노드들을 추출하는 기준이 되는 Activity들은 워크플로우의 분기, 동기화, 선택 등을 나타내는 <flow>, <sequence>, <switch> 이다. 나머지 Activity들은 실행 그래프의 노드에 포함되어서 실행되어야 할 요소로 간주된다.

2.2 워크플로우 기반 웹 서비스 결합

현재 대부분의 워크플로우 기술하는 방법들은 워크플로우를 그래프 문제로 다루고 있으며 정보와 제어의 흐름을 선으로 표

```
<process>
      <partnerLinks> </partnerLink>
      <partners> </partner>
      <variables> </variables>
      <falutHandlers> </falutHandlers>
      <eventHandlers> </evnetHandler>

      activity
</process>
```

그림 1 BPEL4WS의 명세 구조

현하고 있다. WSFL이나 BPEL4WS를 통해 기술된 비즈니스 프로세스에는 기존의 웹 서비스간의 정보의 흐름과 제어의 흐름이 포함되며 기존 웹 서비스에 대한 서비스 호출은 하나의 Activity로 간주될 수 있다. 워크플로우에 기반 하여 웹 서비

스를 결합하려는 연구는 전통적인 WFMS(Work Flow Managment System) 연구에 기반하고 있다.

3. BPEL4WS 명세의 실행 그래프

BPEL4WS에 의해 기술된 비즈니스 프로세스는 기존의 서비스들의 결합에 의한 결합 웹서비스의 형태로 실행되어 진다. 이를 위해 BPEL4WS로 기술된 결합 규칙을 분석하는 것이 선행되어야 한다. BPEL4WS에 의해 표현될 수 있는 웹 서비스간의 상호작용은 정형화된 워크플로우 패턴들의 그래프로 표현될 수 있고 축약된 실행 그래프로 변환될 수 있다. 다음은 BPEL4WS가 표현할 수 있는 워크플로우 패턴 중에 기본 패턴과 분기 및 동기화 패턴들에 대해서 논의한다.

3.1 워크플로우 패턴에 기반 한 실행 그래프

복잡한 워크플로우는 다양한 워크플로우 패턴들의 조합으로 분석 될 수 있다[1]. BPEL4WS로 기술된 비즈니스 프로세스는 하나의 Activity가 노드가 되는 워크플로우 그래프로 표현된다. 다시 이 그래프는 동일한 의미를 가지는 실행 그래프로 축약될 수 있다. 실행 그래프는 워크플로우 그래프에 비해 노드의 수와 제어 및 정보의 흐름이 최소화된 그래프이다. 실행 그래프를 통해 런타임 시 전체 프로세스의 실행 성능을 향상시킬 수 있으며 실행 그래프의 각 노드들은 분산 실행 가능한 실행 단위가 된다.

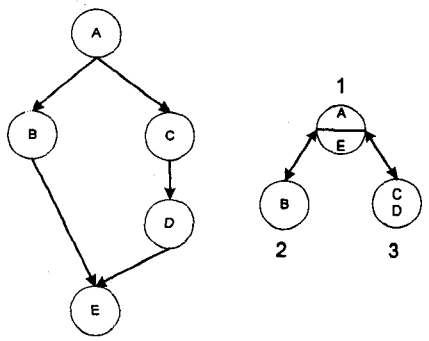

그림 2 워크플로우 그래프와 실행 그래프

실행 그래프를 구성하는 노드는 실행 노드라 명명하며 적어도 BPEL4WS의 Activity를 하나 이상 포함할 수 있으며 노드간의 에지는 실행 순서 및 동기화와 병렬 수행 등을 나타낸다. 실행 노드가 여러 개의 Activity를 포함할 경우에는 Activity들이 실행되어야 할 순서에 대한 정보를 포함한다. 그림 2에서 A, B등은 Activity를 의미하며 왼쪽의 워크플로우 그래프는 A가 실행되어진 후에 B와 C가 동시에 실행될 수 있으며 B와 D의 실행이 완료되면 E가 실행되어짐을 의미한다. 1노드의 원을 가로지르는 선에서 분기된 화살표들은 A의 실행이 완료된 이후에 동시에 실행되는 노드들이며 E는 2와 3노드가 완료되면 실행되어지는 것을 의미한다. BPEL4WS에 의해 표현될 수 있는 워크플로우 패턴에 대한 실행 그래프는 다음과 같다.

가)순차 실행

BPEL4WS에서 순차적으로 실행되는 Activity들은 하나의 실행 노드로 표현되며 마지막 Activity가 완료되면 순차실행이 종료된다.

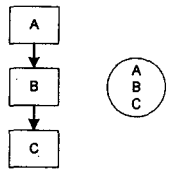

그림 3 순차실행

나)병렬 분기

병렬 분기는 하나의 Activity가 종료한 후에 둘 이상의 Activity가 분기되어 동시에 실행되는 것을 의미한다. 그림 4에서 왼쪽의 병렬 분기는 오른쪽의 실행 그래프로 표현될 수 있다. 1번 노드의 ● 는 특정 Activity가 아니고 Empty Activity를 의미한다. 1노드에서 2와 3노드로 연결된 화살표는 A의 종료 후에 2와 3노드가 동시에 실행되는 것을 의미하며 1노드의 중심을 가로지르는 실선은 2와 3의 완료까지 블록되는 것을 의미한다.

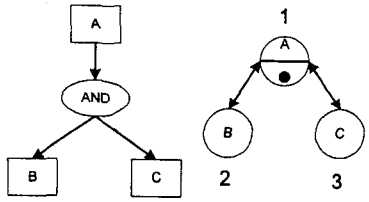

그림 4 병렬 분기

다)동기화와 단순 병합

동기화는 둘 이상의 Activity가 완료되는 조건을 만족할 경우에 다음 Activity가 실행되는 것을 의미한다. 동기화는 병렬 분기와 동일한 실행 모델로 표현될 수 있다. 그림 5의 단순 병합은 둘 이상의 Activity중에 하나라도 완료되면 다음 Activity실행 노드가 실행될 수 있음을 의미하며 동기화와 유사한 모습을 가진다.

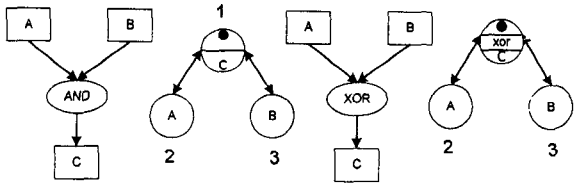

그림 5 동기화와 단순 병합

라)병렬 분기와 동기화의 조합

워크플로우에서 병렬 분기와 동기화가 혼합된 형태가 흔히 나타난다. 그림 6은 병렬 분기와 동기화가 혼합된 두 가지 패턴을 보여준다. 두 번째 예에서 B' Activity의 실행은 A의 완료가 만족할 경우에 이루어지며 실행 그래프에서 2노드에서 3노드로의 점선으로 된 화살표로 표현된다. 점선으로 표현된 화살표는 A의 종료를 알리는 비동기 메시지로 구현될 수 있다.

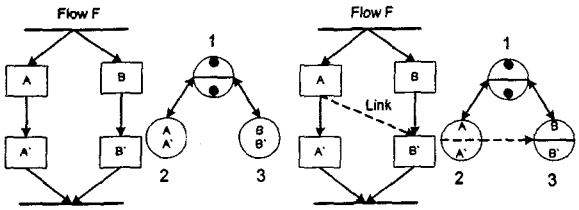

그림 6 병렬 분기와 동기화의 조합

마) 배타적 선택과 다중 선택

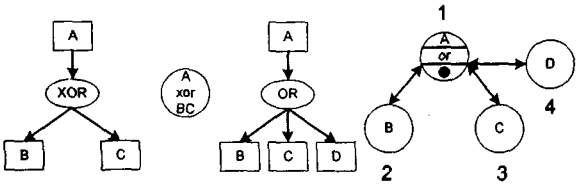

그림 7 배타적 선택과 다중 선택

배타적 선택은 A의 완료 후에 B 혹은 C가 실행될 수 있음을 의미한다. 배타적 선택은 하나의 실행 노드로 표현되며 내부에 조건을 검사하여 선택하는 작업이 필요하다. 그림 7의 오른쪽은 다중 선택을 나타내며 동시에 다수의 노드가 실행 가능함을 나타낸다

바) 배타적 선택과 단순 병합의 조합

그림 8에서 배타적 선택과 단순병합의 조합은 단순 병합과 동일한 형태를 보인다.

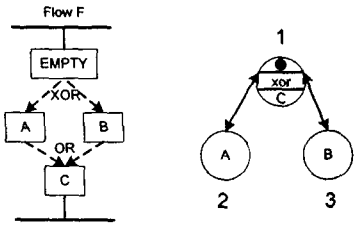

그림 8 배타적선택과 단순병합의 조합

각 패턴별 실행 그래프들은 다른 실행 그래프의 구성하는 하나의 실행 노드로 표현되어 진다.

3.2 주문 처리 예제

그림 9는 일반적인 물품 구매의 워크플로우 그래프와 그로부터 생성된 실행 그래프를 보여준다. 실제 재고정리 Activity는 내부에 하나 이상의 순차적인 Activity들이 내재되어 있다. 주문처리 실행 그래프는 1노드의 주문접수가 완료된 이후에 2,3,4 노드의 병렬 및 분산 실행이 가능하며 2 노드는 1노드와 함께 하나의 실행 노드로 볼 수 있다.

4. 구현 방안

BPEL4WS 명세를 실행하기 위해서는 웹 서비스간의 상호동작과 관련된 상태 정보를 유지하고 결합에 참여하는 서비스에 대한 적절한 서비스 호출과 그 결과를 처리하는 코디네이터가 필요하며 실행 그래프의 각 노드들은 분산되어 실행될 수 있다. 이를 위해 OASIS의 WS-CAF(Web Service Composite Application Framework)[6]을 구성하는 컨텍스트, 코디네이터,그리고 트랜잭션 프레임웍 상에서 BPEL4WS로 기술된 비즈니스 프로세스를 지원하는 프레임웍을 구현할 수 있다. WS-CAF는 웹 서비스의 결합을 통해 응용을 생성하기 위한 기본 기능을 정의하고 있다. BPEL4WS의 워크플로우 패턴에 대한 각 실행 모델들의 공통 구현 요소의 추상화를 통해 WS-CAF 상에 구현이 용이하다.

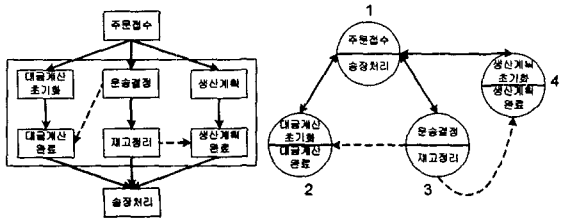

그림 9 주문처리를 위한 실행 그래프

5. 결론 및 향후 연구

본 논문에서는 BPEL4WS로 기술된 비즈니스 프로세스에 내재된 워크플로우 그래프에서 실행 그래프를 생성하는 방법을 제안하였다. 실행 그래프는 원래의 워크플로우 그래프에 비해 적은 노드로 축약되어 병렬 실행을 통한 성능 향상과 분산실행을 위한 효과적인 구현을 가능하게 한다.

향후 연구로는 수학적 모델링을 통해 실행 그래프의 생성 알고리즘을 개발하고 다양하고 복잡한 워크플로우에 대한 연구가 필요하다. 이를 기반으로 BPEL4WS 명세를 작성하고 실행하기 위한 결합 프레임웍의 설계 및 구현이 필요하다.

참고 문헌

[1] W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and P. Wohed, Pattern-Based Analysis of BPEL4WS, QUT Technical re-port,FIT-TR-2002-05, Queensland University of Technology, Brisbane, 2002

[2] W. M. P van der Aalst, Dont go with the Flow: Web Services Composition standards exposed Trends & Controversies, Jan/Feb 2003 issue of IEEE Intelligent Systems

[3] S. Thatte, XLANG:Web Services for Business Process Design, Microsoft, Redmond, Wash., 2001, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm

[4] F. Leymann, Web Services Flow Language(WSFL 1.0), IMB, May 2001, www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[5] F. Curbera et al., Business Process Execution Language for Web Services(Version 1.0), IBM, July 2002, www-106.ibm.com/developerworks/webservices/library/ws-bpe

[6] OASIS, Web Service Composite Application Framework (WS-CAF), http://www.arjuna.com/standards/ws-caf/