

유비쿼터스 환경에서의 동적 워크플로우 지원에 관한 연구

윤혜정*, 이민수*

*이화여자대학교 컴퓨터학과

e-mail:auroree@ewhain.net, mlee@ewha.ac.kr

Supporting Dynamic Workflows in a Ubiquitous Environment

Hyejung Yoon*, Minsoo Lee*

*Dept. of Computer Science and Engineering, Ewha Womans University

요 약

동적이고 분산된 환경인 유비쿼터스 환경에서의 워크플로우 디자인, 실행은 다양한 변화에도 적응할 수 있도록 융통성이 있어야 한다. 따라서, 유비쿼터스 웹 서비스 워크플로우 설계는 시맨틱 기반으로 동적으로 찾거나 호출되어야 한다. 이를 위해 워크플로우 디자인 언어인 BPEL4WS 에 시맨틱을 포함하여 이 문제를 해결한다.

1. 서론

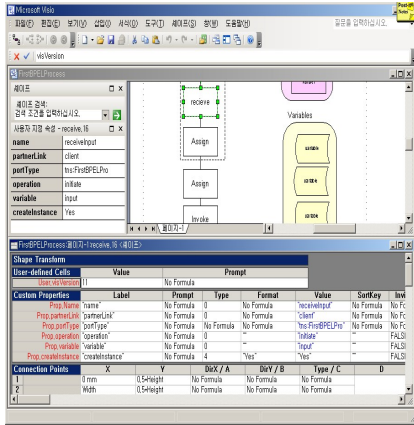
유비쿼터스 환경은 분산된 서비스를 찾고 초기화하기 위한 플랫폼이다. 유비쿼터스 환경에서 서비스는 웹 서비스 형태로 제공될 수 있고, 이러한 웹 서비스들은 워크플로우 기술을 사용하여 통합이 가능하다. 동적 환경 또는 분산된 환경에서의 워크플로우 디자인 및 실행은 동적으로 이루어져야 한다. 워크플로우 설계 시 웹서비스를 갑자기 사용할 수 없고, 백업 서비스를 찾을 수 없거나 몇몇 서비스 제공자를 식별할 수 없을 수 있다. 그러므로, 워크플로우 시맨틱 기반의 웹서비스를 동적으로 찾거나 호출하는 것이 제공되어야 한다. BPEL4WS 은 웹서비스의 워크플로우 디자인 언어 중 가장 대중적이고 장래성 있는 언어이지만, BPEL4WS 의 문제점 중 하나는 고정된 WSDL 파일을 참조하는 것으로, 이는 동적 환경 및 유비쿼터스 환경에서 융통성이 적은 워크플로우를 만드는 것이다. 또 다른 문제는 지정된 단 하나에만 리소스의 양이 제한된다는 것이다. 이는 유비쿼터스 네트워크에서 단순하거나 백업된 서비스를 찾기 위한 시맨틱을 제공하는 것을 어렵게 한다. 본 논문은 유비쿼터스 환경에 적합한 동적 워크플로우를 제공하기 위해 BPEL4WS 언어에 DAML-S 온톨로지의 시맨틱 요소를 추가하여 이를 해결하고자 한다. 이를 위해 BPEL4WS 엔진에서 온토폴로지를 사용함으로써 시맨

틱을 포함하는 BPEL4WS 를 확장하는데 중점을 두고 있다.

2. 기존 웹 서비스 및 워크플로우의 문제점

BPEL4WS 는 웹 서비스 합성 언어이자, 높은 기능성(functionality)를 지녀, 웹 서비스 프로세스 효율적으로 합성하기 위한 언어이다. 그러나 자동화되고 지능적인 시맨틱 웹 서비스의 합성을 위한 기술로는 많이 부족하다. 이를 위해 기존의 BPEL4WS 설계 도구에서 효율적인 방법으로 시맨틱을 지원하고자 한다. (그림 1)은 Microsoft Visio 를 사용하여 개발한 BPEL4WS 설계를 보여준다. BPEL4WS 설계에 있어 필요한 파일은 여러가지가 있지만, 이들 중 bpel 파일과 wsdl 파일이 가장 중요하다.

웹 서비스 조합(composition)은 비즈니스 흐름에 따라 몇 개의 웹 서비스를 통합하는 것이다. BPEL4WS 는 워크플로우 개념을 사용하여 웹 서비스 들을 조합한다. 그러나 BPEL4WS 는 웹 서비스의 기능(functional)적인 면을 표현해주기 때문에 웹 서비스의 개념적이거나 의미론적인 측면을 표현하기에 한계가 있다. BPEL4WS 는 비즈니스 흐름을 생성하기 위한 다양한 액티비티를 제공하기 때문에, 만일 워크플로우를 디자인 할 때 모든 가능한 흐름을 설계한다면 현 BPEL4WS 로도 문제가 없을 것이다. 그러나 유비쿼터



(그림 1) BPEL4WS 설계 도구

스환경은 워크플로우를 설계한 후 실행 시 웹 서비스 실행이 안될 수도 있고, 문제가 발생할 수도 있다. 따라서 다른 웹 서비스를 찾아주거나 교체해 줄 기술이 필요한데, 기존 BPEL4WS 은 시멘틱을 지원하지 않아 이를 해결할 수 없다. 따라서 기존의 BPEL4WS 에 DAML-S 의 온톨로지 개념을 추가하여 각각의 파일을 매핑시킴으로써 이 문제를 해결하고자 한다.

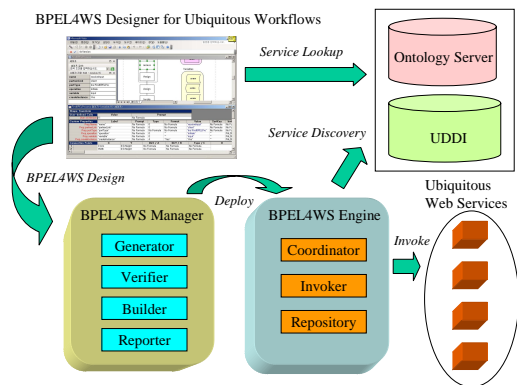
3. 동적 기반 워크플로우의 지원

DAML-S 는 웹 서비스를 기술하기 위한 DAML-OIL 온톨로지로, 웹 서비스들의 의미론적 정보를 제공해준다. DAML-S 는 서비스 프로파일, 서비스 모델, 그라운드링의 세 가지 부분으로 구성되어 있다. 서비스 프로파일은 서비스 수용능력(capabilities)의 의미론적인 설명을 담고 있으며, 서비스 모델은 어떻게 서비스들이 작용하는지를 설명하고, 그라운드링은 서비스의 실제화(실제 WSDL 과의 매핑)을 담당하며, Process Model 에서 표현된 웹 서비스와 WSDL 로 표현된 웹 서비스간의 관계를 기술한다

- A. 서비스 프로파일: 자동 검색에 필요한 서비스 속성들(입력(input), 출력(output), 선조건(precondition), 효과(effect):IOPE)을 기술한다. 입력과 출력은 BPEL 프로세스에서 제공되지만 선조건과 효과는 제공되지 않고 있으므로, 이를 추가하여 시멘틱 개념에 적용하고자 한다.
- B. 서비스 모델: 이 부분은 BPEL4WS 와 가장 유사한 부분으로 bpel 파일 자체를 서비스 모델로 간주하였다.
- C. 그라운드링: BPEL4WS 에서 사용하고 있는 WSDL 은 입출력 인자의 데이터 타입만 기술하게 되어 있어, 입출력에 대한 의미 정보를 유지할 수 없다. 그리고 기존의 WSDL 은 고정된 WSDL 파일을 참조하여, 그 흐름을 정적으로만 구성할 수 밖에 없고, 제한된 자원만 갖고 있기 때문에 의미를 부여할 수 없어 대체 서비스를 찾을 수 없다는 단점이 있다. 따라서 웹 서비스 흐름을 정의하는 BPEL4WS 가 사용하는 WSDL 파일에 시멘틱 정보를 추가하여, 현 WSDL 스키마를 유지하면서, DAML-S 의 기술 클래스인 DAML-OIL 로 온톨로지를 구성할 수 있게 한다.

4. 유비쿼터스 환경에서의 워크플로우 시스템 구축

의미 기반 BPEL4WS 시스템 구현은 다음 그림 8 과 같다. BPEL4WS 디자이너를 사용하여 설계자는 사용자가 UDDI 와 Ontology Server 를 검색하여 서비스를 찾을 수 있도록 하며, 또 이를 워크플로우에 추가할 수 있도록 한다. 워크플로우 설계가 완료되면 BPEL4WS Manager 는 설계한 워크플로우 파일을 받아 BPEL4WS XML 파일을 생성한다. Generator 는 BPEL4WS XML 파일을 생성하는 컴포넌트이며, Verifier 는 생성된 XML 파일을 받아 워크플로우 설계에 오류가 있는 지 검사한다. Builder 는 BPEL4WS 엔진에서 실행 시 필요한 추가적인 파일을 생성하며, Reporter 는 현 상태나, 오류 메시지 등을 사용자에게 제공해준다. BPEL4WS Manager 에서 생성된 결과는 BPEL4WS 엔진에 deploy 한다. BPEL4WS 엔진은 Coordinator, Invoker, Repository 로 구성되어 있다. Coordinator 는 유비쿼터스 환경에서의 모든 워크플로우의 흐름을 제어한다. 이는 조건을 검사하고 작업(task)들의 동기화 작업을 수행한 후 Invoker 를 호출하여 각 유비쿼터스 웹 서비스를 실행한다. Invoker 는 유비쿼터스 웹 서비스를 실행할 뿐 아니라, 온톨로지 서버와 UDDI 서버를 사용하여 동적인 워크플로우 작업을 수행한다. 만일 해당 파트너를 사용할 수 없는 경우, 온톨로지 서버와 UDDI 를 통해 사용 가능한 파트너를 검색하여 찾아준다. 또한 디자인 단계에서 워크플로우 스펙을 변경하지 않고도 런타임 시 관련된 웹 서비스를 찾아 바로 적용할 수 있어 동적인 워크플로우 설계가 가능하다. 이는 유비쿼터스 환경에서 알려지지 않은 공급자로부터 공급받는 시나리오일 경우 동적으로 파트너를 추가하는 데 매우 유용하게 사용될 수 있다.



* 이 연구는 BK 21 프로그램의 지원을 받아 수행되었습니다.

참고문헌

- [1] W3C, <http://www.w3.org/TR/soap/>
- [2] W3C, <http://www.w3.org/TR/wsdl>
- [3] Oracle BPEL Process Manager, <http://www.oracle.com/technology/products/ias/bpel/index.html>
- [4] Kunal Verma, Rama Akkiraju, et, al. On Accommodating Inter Dependencies in Web Process Flow Composition, American Association for Artificial Intelligence (www.aaai.org), 2004