

유비쿼터스 웹 서비스 환경을 위한 지능형 동적 워크플로우 관한 연구

윤혜정*, 이민수*

*이화여자대학교 컴퓨터학과

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

Intelligent Dynamic Workflow support for a Ubiquitous Web-Service based Environment

Hyjung Yoon, * Misoo Lee*

*Dept. of Computer Science, Ewha Womans University

요 약

유비쿼터스 환경에서 분산된 환경에 있는 서비스들을 찾고 통합하는 작업은 점차 중요해지고 있다. 이러한 유비쿼터스 환경에서 각각의 프로세스들은 웹 서비스와 같은 형태로 제공될 수도 있고, 이러한 프로세스는 워크플로우 형태로 통합될 수 있다. 워크플로우를 디자인하고 실행하는 데 있어서 매우 유연하게 시스템 간의 통합이 이루어져야 한다. 해당 웹 서비스의 실행 가능 여부, 백업 서비스의 발견, 설계 시 몇몇 서비스들이 실행되지 못할 수도 있다. 따라서 본 논문은 웹 서비스의 동적인 통합을 위해 워크플로우를 기반으로 한 동적인 웹 서비스 발견, 실행하는 시스템을 제안한다. 특별히 워크플로우 설계 시 의미를 부여하여 문제를 해결한다.

1. 서론

유비쿼터스 환경에서 분산된 서비스를 찾고 통합하는 것은 점점 중요한 일이 되어가고 있다. 유비쿼터스 환경에서의 웹 서비스는 어떠한 상황에서도 실행 가능해야 하고, 동적인 설계가 가능해야 한다. 또한 웹 서비스가 기업 IT 환경에 보편적으로 적용되기 위해서는 SOAP, WSDL, UDDI 등 기본적인 연결 기술들만으로는 부족하다. 동적으로 자동화된 업무흐름을 구성할 수 있어야 한다. 그러기 위해서는 프로세스 단위를 정의하고 통제하는데 적용되는 기술 표준이 필요하다. Workflow 기반은 주로 XPDL 이라는 표준기술을, EAI 기반은 BPEL 이라는 표준기술을 적용하여 개발되고 있다. 본 논문에서는 웹 서비스의 흐름을 정의하는 언어인 BPEL 에 의미를 부여하는 OWL 을 적용하여 이 문제를 해결하고자 한다.

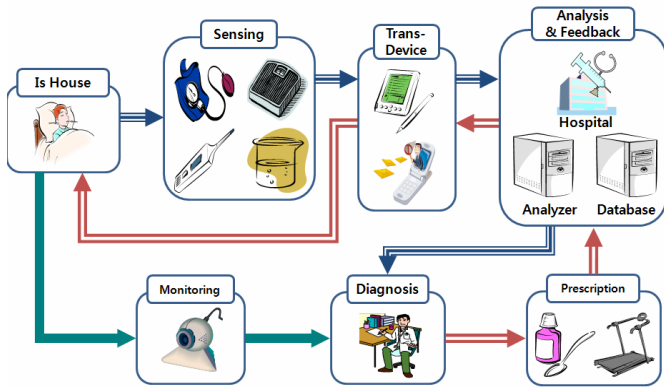
2. 관련연구

워크플로우(workflow)는 비즈니스 프로세스를 전체 또는 부분적으로 컴퓨터에 이롭게 하거나 자동화한 것으로서, 소프트웨어 실행을 위해 이를 정의하고 관리하는 시스템을 워크플로우 관리 시스템(WfMC)[1]라 한다. 과거의 워크플로우는 프로세스의 순서 및 실행에 중점을 두었다면, 최근의 워크플로우는 프로세스의 정형화, 규격화에 더 중점을 두었다고 볼 수 있다 [2]. 아울러, 기업간 혹은 고객과의 사이에서 일어날 수 있는 비즈니스 프로세스를 제대로 구축하고 관리하기 위한 여러 시스템을 구축하고 관리하는 기술을

BPM 이라 하고, 이러한 BPM 을 구축하는 대표적인 언어로는 BPEL4WS(Business Process Execution Language for Web Service: 이하 BPEL)[3]가 있다. BPEL 은 웹 서비스로 구성된 업무 및 시스템간을 연동하고 조정하는 언어이다. 이 BPEL 도구로는 Oracle BPEL Manager, IBM BPM Solution 등등이 있다. 그러나 이러한 BPEL 로 구성하는 워크플로우는 정형화되어 있어, 동적으로 빠르게 변하는 유비쿼터스 환경에는 적합하지 않다. 따라서 워크플로우 설계에 온톨로지 정의 언어인 OWL 의 의미를 부여하여 동적인 설계를 가능하게 한다.

3. 지능형 동적 워크플로우

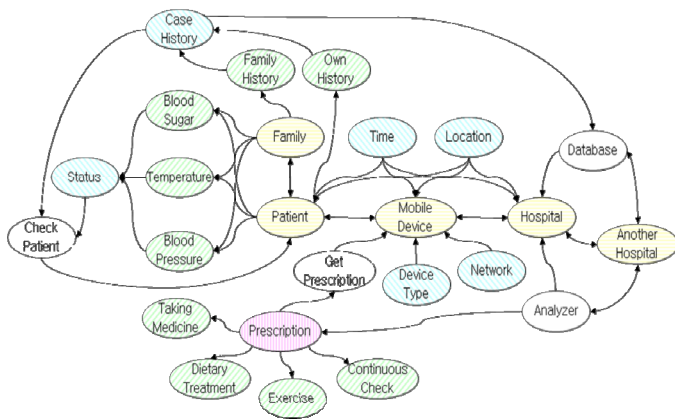
유비쿼터스 환경에서 의미가 어떻게 사용되는 지를 증명하기 위해서 U-Health 시스템 시나리오를 기준으로 구현하였다. 이 시스템은 병원이 아닌 집안에서 환자의 상태(혈당, 혈압, 몸무게 등)를 체크한 후 그 결과를 담당 병원과 의사에게 전달된다. 이 때 각 의료 장비는 센싱을 통해 측정되며 측정값들은 모바일 기기를 통해 전달된다. 병원으로 환자의 상태를 측정 한 값이 도달하면 이는 분석기를 통해 분석하여, 해당 의사나 담당자를 호출하여 이를 전달한다. 아울러, 측정값 및 이력은 모두 데이터베이스에 저장되어 추후 분석에 사용될 수 있다. 담당 의사는 환자의 상태를 전달받으면 처방을 내리고 이 처방에 따라 환자를 모니터링 하거나, 의사 및 간호사가 방문하는 등의 추후 조치를 취하게 된다. 그림 1 은 이러한 흐름도를 설명한 그림이다.



(그림 1) U-health 시스템 흐름도

이 시나리오에서 환자의 상태를 전달하고 처방을 받는 프로세스는 반드시 수행되어야 하는 프로세스 중에 하나이다. 즉 환자가 가진 장비들은 문제 없이 결과를 꼭 전송 하고 결과를 받아야 한다. 이러한 작업들은 워크플로우를 통해 수행되고, 선조건은 증상에 따라 측정된 측정값이 된다.

이러한 시나리오를 가지고 동적인 워크플로우를 구성하기 위해서는 다음과 같은 것이 단계를 거친다. 우선, BPEL 로 각각의 프로세스를 정의한다. 이는 워크플로우 설계 언어인 BPEL 을 이용하여 각각의 웹 서비스를 정의하고 통합하면 이는 하나의 커다란 웹 서비스로 실행된다. 또한 각 단계에 필요한 프로세스 온톨로지를 OWL 로 구성한다. 그림 2 는 위의 시나리오를 온톨로지를 구성한 그림이다. 이 단계까지 완료 되면 시스템 내부에서는 BPEL 에서 만들어진 WSDL 과 OWL 에서 만들어진 WSDL 파일을 비교 분석하여 필요한 부분을 자동 추출하여 실행될 WSDL 에 삽입 되게 되고, 수정된 WSDL 파일을 통하여 전체 프로세스가 실행되게 된다. 표 1 은 확장된 WSDL 의 일부이다.



(그림 2) AutoMedicalCare 프로세스 온톨로지

```
<wsdl:operation name="RequestTreatment"
parameterOrder="in0" owls_operator="U:CheckPatient">
<wsdl:input message="U:OperationRequest1" name="
PatientOfSymptoms"/>
```

```
<wsdl:input message="U:OperationRequest2"
name=" PatientOfHistory"/>
< U:Precondition name="StatusBloodSugar"
U:refer="U:BloodSugar"/>
< U:Precondition name="StatusTemperature"
U:refer="U:Temperature"/>
< U:Precondition name="StatusBloodPressure"
U:refer="U:BloodPressure"/>
< U:Precondition name="HistoryOwnHistory"
U:refer="U:OwnHistory"/>
< U:Precondition name="HistoryFamilyHistory"
U:refer="U:FamilyHistory"/>
<wsdl:operation name="ResponseTreatment" owls_operator="
U:GetPrescription" >
<wsdl:output message="U:OperationResponse"
name="PrescriptionOfPatient"/>
< U:Effect name="GetTakingMedicine"
U:refer="U:TakingMedicine"/>
< U:Effect name="GetDietaryTreatment"
U:refer="U:DietaryTreatment"/>
< U:Effect name="GetExercise" U:refer="U:Exercise"/>
< U:Effect name="GetContinuousCheck"
U:refer="U:ContinuousCheck"/>
</wsdl:operation>
```

(표 1) 확장된 WSDL

4. 결론

본 논문에서는 유비쿼터스 환경에서 동적인 워크플로우를 구성하기 위한 방법을 제안하였다. 특별히 U-Health 시스템에서의 시나리오를 이용하여 각 프로세스의 흐름에 어떠한 방식으로 의미를 부여하는 지에 대해서도 살펴보았다. 이러한 기술을 확장하면 또 다른 유비쿼터스 영역(U-Traffic, U-home 등등)으로 적용할 수 있게 되고, 향후 U-City 를 구성하는 데 도움이 될 수도 있을 것이다.

* 이 연구는 BK21 프로그램의 지원을 받아 수행하였음

참고문헌

[1] WfMC, <http://www.wfmc.org>
 [2] BPMi.org, <http://www.bpmi.org>
 [3] Business Process Execution Language for Web Services v.1.1, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbizspec/html/bpel1-1.asp>
 [4] Chris Peltz, Web Service Orchestration and Choreography, www.wsi2.com, 2003,7
 [5] Kunal Verma, Rama Akkiraju, et, al. On Accommodating Inter Dependencies in Web Process Flow Composition, American Association for Artificial Intelligence (www.aaai.org), 2004
 [6] Liangzhao zeng, Boualem Benatallah et.al, Flexible Composition of Enterprise Web Services, Routledge, 2003