

유비쿼터스 환경을 위한 워크플로우 기반의 시나리오 편집기의 설계 및 구현

전태연⁰ 한주현 최재영
승실대학교 컴퓨터학과

tyjeon⁰@ss.ssu.ac.kr, jhhan@ss.ssu.ac.kr, choi@ssu.ac.kr

Design and Implementation of Scenario Editor based on Workflow for Ubiquitous Environments

Taeyeon Jeon⁰, Joohyun Han, Jaeyoung Choi
School of computing, Soongsil University

요 약

본 논문에서는 유비쿼터스 환경의 사용자에게 적절한 서비스를 제공하기 위해 필요한 시나리오를 웹 서비스 기반의 워크플로우를 이용하여 편집할 수 있는 uFlow 시나리오 편집기를 소개한다. 시나리오 편집기는 유비쿼터스 환경에서 발생하는 컨텍스트와 이를 이용하는 서비스간의 연관 관계를 구조적 컨텍스트 모델 기반의 서술 방식으로 표현하는 uWDL(Ubiquitous Workflow Description Language)을 자동으로 생성하기 위한 도구로서, 해당 유비쿼터스 컴퓨팅 환경에서 사용 가능한 컨텍스트 및 서비스의 검색과 검색된 컨텍스트 리스트 정보에 따라 워크플로우의 흐름을 결정하기 위한 컨텍스트 표현 기술을 포함하며, 시나리오 개발자가 손쉽게 유비쿼터스 환경을 위한 시나리오를 uWDL 형태로 작성할 수 있는 기능을 제공한다.

1. 서 론

워크플로우는 주로 비즈니스 프로세스에서 사용되었던 개념으로 여기서 비즈니스 프로세스란 일반적으로 기능적인 역할과 관계를 정의하는 조직 구조의 맥락에서 업무 목표나 정책적인 목적을 총괄적으로 실행하는 일련의 연관된 업무 절차를 말한다. 유비쿼터스 환경에서의 워크플로우 기술은 사용자의 목적을 달성하기 위한 시나리오를 처리하기 위해 특정 상황에서 받게 되는 컨텍스트, 프로파일, 이벤트 등의 정보를 이용해 서비스들간의 업무 흐름을 정의하는 데 사용될 수 있다. 하지만 기존의 비즈니스 프로세스에 적용되었던 워크플로우는 그것이 사용되던 특정 비즈니스에 최적화되어 너무나 정적인 특성을 갖고 있다. 다양한 장치들을 이용하고 수많은 사용자 컨텍스트를 처리해야 하는 유비쿼터스 환경은 동적인 작업흐름을 필요로 하기 때문에 정적인 워크플로우의 특성을 동적인 웹 서비스 기반의 유비쿼터스 환경에 적용하기 위한 적용력 있는 워크플로우가 필요하다. uWDL 시나리오 편집기는 이러한 특성을 만족시키기 위해 컨텍스트 정보를 통해 워크플로우를 구성하여 해당 상황에서 보다 유연한 서비스가 가능하게 한다.

uWDL 시나리오 편집기가 사용하는 또 다른 주요 기술로서, 웹 서비스는 다양한 플랫폼상에서 수행되는 서로 다른 소프트웨어 어플리케이션간의 상호작용에 표준성을 제공한다[1]. 웹 서비스는 표준화된 방법을 제공한다는 점과 분산된 처리를 지원할 수 있다는 점에서 유비쿼터스 환경에 적합하다. uWDL 시나리오 편집기는 웹 서비스들이

이용하여 다양한 종류의 서비스를 제공 받을 수 있다.

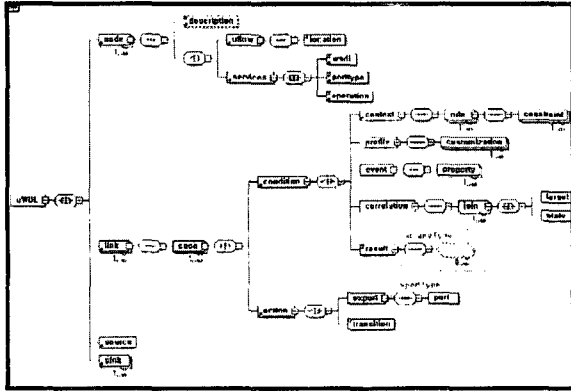
uFlow 시나리오 편집기는 기존의 컴퓨터 시스템이나 소프트웨어가 제공하던 정적인 기능과는 달리 다양한 사용자의 요구사항을 만족 시킬 수 있도록 설계 되었다. 웹 서비스로 구축된 다양한 서비스들과 사용자의 상황정보인 구조적 컨텍스트 모델을 워크플로우로 구성함으로써 여러 가지 서비스들의 조합을 만들어 다양한 시나리오를 표현할 수 있게 하므로써 동적인 유비쿼터스 환경에서 사용자의 요구사항을 만족 시킬 수 있다. uFlow 시나리오 편집기는 유비쿼터스 환경을 위한 웹 서비스 기반의 워크플로우 언어인 uWDL(Ubiquitous Workflow Description Language)을 기반으로 사용자의 시나리오를 변환한다. uWDL의 주요 4가지 요소(Node, Link, Sink, Source) 중 서비스 요소 인 Node, Sink, Source를 화면상에 배치하고 해당상황에서의 이벤트와 컨텍스트를 통하여 이들간의 작업 흐름을 제어할 수 있도록 Link 요소를 추가하였다

2. 관련 연구

2.1 uWDL (Ubiquitous Workflow Description Language)

기존의 웹 서비스 기반의 워크플로우 언어인 BPFL4-WS, WSFL, XLANG 들은 유비쿼터스 환경에서 발생하는 컨텍스트, 프로파일 및 이벤트 정보에 대한 기술을 포함하지 않아, 유비쿼터스 환경의 서비스를 표현하는데 한계가 있었다. 이를 해결하기 위하여 uWDL은 워크플로우

언어의 특징에 컨텍스트 정보를 표현하기 위한 요소(element)를 추가함으로써 보다 효과적으로 서비스의 흐름 관계를 표현할 수 있다[2].



[그림 1] uWDL 스키마

그림 1에서와 같이, uWDL은 크게 웹 서비스를 가리키기 위한 <node> 엘리먼트와 유비쿼터스 환경에서 발생할 수 있는 컨텍스트, 프로파일 및 이벤트 등을 명시하며, 이에 따라 서비스의 흐름이 변화되도록 정의하는 <link> 엘리먼트로 구분된다.

2.2 구조적 컨텍스트 모델링

컨텍스트 정보는 사용자와 응용 프로그램간의 상호작용을 필요로 하는 유비쿼터스 환경에서 매우 중요하게 사용된다. 여기서 말하는 컨텍스트란 엔티티의 상황을 설명하기 위한 정보를 말하고 엔티티는 사람, 장소 혹은 사용자와 응용 프로그램 또는 그들 자신의 상호작용을 위해 고려되는 모든 객체들을 말한다.[3]

구조적 컨텍스트 모델링은 uWDL의 요소중 <link>의 하위 요소인 <context> 및 <profile> 요소를 표현하기 위하여 고안되었다. 이것은 시간, 위치, 사용자, 장치, 네트워크 등의 저수준의 컨텍스트 정보를 엔티티로 구분한다. 하지만 이러한 단순한 정보들만으로는 복잡한 상황을 설명할 수 없다. 구조적 컨텍스트 모델링은 퍼스트 오더 로직(First Order Logic) [4] 서술방식을 사용하여 고수준의 컨텍스트 정보들을 주어, 동사, 목적어 형식으로 표현하고 표현된 하나의 문장이 상황을 설명할 수 있도록 한다. 또한 문장 하나만으로 설명될 수 없는 상황은 문장을 여러 개 작성하여 and, or, not 등의 연산자로 조합하여 보다 복잡한 컨텍스트 정보를 표현하는데 사용한다.

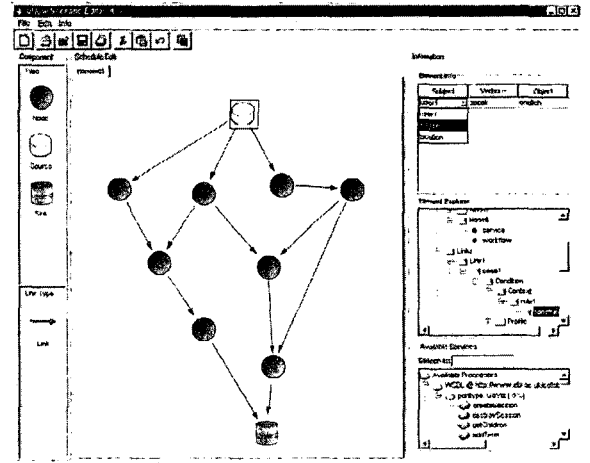
3. uFlow 시나리오 편집기

기존의 소프트웨어나 컴퓨터 시스템은 자신이 수행하는 기능이 정해져 있다는 점에서 정적인 특성을 가지고 있다. 하지만 유비쿼터스 환경에서의 소프트웨어나 컴퓨터 시스템은 수없이 다양한 사용자의 필요사항을 만족할 수 있어

야 한다.

그림 2에서 나타내는 uFlow 시나리오 편집기는 이렇듯 변화되는 목적을 만족시키기 위하여 제안되었다. 사용자의 다양한 요구사항들은 웹 서비스를 통하여 분산 네트워크 상에서 존재하는 여러 가지 서비스들을 상황정보, 즉 컨텍스트 미들웨어를 통해 받게 되는 컨텍스트에 맞게 선택하여 워크플로우로 구성하여 실행함으로써 만족시킬 수 있다.

uFlow 시나리오 편집기는 웹 서비스 기술을 통하여 다양한 서비스에 대한 요구사항을 만족시키고, 상황정보를 인지하기 위하여 구조적 컨텍스트 모델을 사용한다. 또한 하나의 시나리오를 달성하기 위한 과정들은 컨텍스트 정보를 통하여 선택된 서비스들을 워크플로우로 구성하므로써 이루어진다. GUI 환경상에서 구성된 정보들은 XML 기반의 uWDL로 변환되어 추후 개발될 uWDL 실행 엔진에서 실행될 수 있도록 한다.



[그림 2] uFlow 시나리오 편집기

3.1 구조적 컨텍스트 정보의 사용

특정상황에서 발생하는 컨텍스트 정보들은 구조적 컨텍스트 모델링의 entity 클래스로부터 속성을 상속받아 객체로 만들어져서 편집기에서 사용할 수 있도록 만들어진다. 컨텍스트 정보가 필요한 곳은 uWDL의 하위요소중 context, profile로서 객체화된 저수준의 컨텍스트를 주어 동사, 목적어로 이루어진 단순 컨텍스트 혹은 그것의 조합으로 이루어진 합성 컨텍스트로 만들어 이것이 나열된 서비스들간의 분기에 사용될 수 있는 고수준의 컨텍스트 정보로 만든다. 선택되는 컨텍스트 정보에 따라 타입체크(Type Checking)을 수행하여 연관된 컨텍스트 정보가 하나의 문장을 구성하도록 되어 있다. 시나리오를 편집하는 현재 상황에서 사용 가능한 컨텍스트에 대한 정보는 컨텍스트 미들웨어를 통해서 제공 받는다.

3.2 웹 서비스의 검색 및 사용

사용 가능한 웹 서비스 정보는 미리 지정된 URL에 위치한 WSDL을 참고하여 만들어 낸다. 현재 서비스가 가능한 목록을 오른쪽 하단의 목록에 나타내고 보여지는 단위는 WSDL의 operation을 기준으로 한다. 검색된 웹 서비스 정보는 uWDL의 요소중 Node에서 사용된다. 구체적으로 저장되는 WSDL의 정보는 WSDL 파일의 URL, portType, operation 등이다. 만약 서버가 현재 서비스가 가능하지 않다면 해당되는 WSDL 파일을 찾았더라도 사용 가능한 서비스 목록에는 포함시키지 않는다.

3.3 워크플로우가 적용된 웹 서비스의 구성

uWDL의 웹 서비스에 대한 정보를 갖고 있는 Node 요소들은 편집기상에 본기를 위한 컨텍스트 정보를 포함한 Link 요소로 연결되어 배치된다. 시나리오의 시작은 Source 요소로부터 시작되어 여러 Node들을 거쳐 Sink 요소에서 마치게 된다. Link 요소는 Link가 시작되는 Node와 끝나는 Node에 대한 정보를 가지고 있어 작업의 흐름이 연결되어 진행될 수 있게 한다.

3.4 사용자 인터페이스의 부분별 주요 기능

uFlow 시나리오 편집기는 uWDL 컴포넌트의 목록을 갖는 uWDL Component Panel과 시나리오를 달성하기 위한 서비스들의 흐름을 한눈에 파악할 수 있는 Main Editor Panel, 시나리오 편집에 포함된 여러 가지 컴포넌트들의 정보를 구조적으로 보여주기 위한 Element Explorer Panel, 선택된 컴포넌트의 세부정보를 편집하기 위한 Element Information Panel, 현재 사용 가능한 웹 서비스를 검색해 내고, 검색된 정보를 컴포넌트에게 제공하기 위한 Available Service Panel로 구성된다.

3.4.1 Element Information Panel

Main Editor Panel에서 선택된 컴포넌트나 Element Explorer Panel에서 선택된 컴포넌트의 하위 요소들의 세부 속성을 열람하거나 편집할 수 있게 한다. 일반적인 텍스트 데이터 값들은 텍스트를 직접 입력하도록 하고, 컨텍스트 정보는 그림 2에서와 같이 콤보 박스를 이용하여 만들도록 설계 되었다.

3.4.2 Element Explorer Panel

Main Editor Panel에 추가된 컴포넌트들은 Element Explorer Panel에서 표시된다. 현재 작성중인 시나리오가 갖는 모든 컴포넌트들을 구조적으로 이곳에서 열람이 가능하고 Main Editor Panel에서 표시되지 않는 하위 컴포넌트들은 Element Explorer를 통해서 접근하여 그것이 갖는 세부 정보 값들을 관리할 수 있다. 이러한 특성으로 인하여 Main Editor가 다른 기능에 대한 부담을 갖지 않고 시나리오에 대한 보다 큰 그림을 표시하는 기능에 충실할 수 있다.

3.4.3 Available Service Panel

사용 가능한 웹 서비스의 목록을 트리 형태로 나열한 곳

이다. 트리에 나열된 목록은 WSDL의 operation을 기준으로 하였으며, 이것을 원하는 Node에 끌어다 놓는 단순한 작업으로 해당되는 정보들이 Node에 매핑될 수 있도록 설계 되었다.

3.5 uWDL의 생성

그림 2와 같이 다양한 웹 서비스와 워크플로우로 구성된 시나리오는 그림 3과 같은 uWDL로 변환한다. 이것은 추후 개발될 uWDL 실행 엔진에서 실행된다.



[그림 3] uWDL 파일

4. 결론

uFlow 시나리오 편집기는 시나리오를 웹 서비스로 제공되는 여러 서비스들을 컨텍스트 미들웨어를 통해 전달 받게 되는 컨텍스트 정보를 통하여 워크플로우로 구성, 조합하여 다양한 시나리오를 표현 함으로서 유비쿼터스 환경의 동적인 특성을 만족시킨다. uWDL 시나리오 편집기를 통해 생성된 시나리오는 웹 서비스기반의 워크플로우 언어인 uWDL을 생성한다.

5. 참고 문헌

- [1] <http://www.w3.org/TR/2004/WD-wsdl20040803>.
- [2] 한주현, 김은희, 최재영, 유비쿼터스 환경을 위한 웹 서비스 기반의 워크플로우 언어의 설계, 정보과학회, 2004.
- [3] Anind K.Dey, "Towards a better understanding of context and context-awareness, Graphics, Visualization and Usability", Center and College of Computing, Georgia Institute of Technology.
- [4] Anand Ranganathan, An infrastructure for context-awareness based on first order logic, 9 November 2002, Springer-Verlag London-Limited 2003.