

URC 를 위한 상황인지 워크플로우 처리기

정재훈*, 최종선*, 최재영*¹, 조용윤**

*송실대학교 컴퓨터학과

**국립순천대학교 정보통신공학부

e-mail : jhjeong@ss.ssu.ac.kr, jongsun.choi, choi@ssu.ac.kr, ycho@sunchon.ac.kr

A context-aware workflows handler for URC

Jae-Hoon Jeong*, Jong-Sun Choi*, Jae-Young Choi*, Yong-Yun Cho**

*School of computing, Soong-Sil University

**Division of Information & Computer , Sun-Chon National University

요 약

워크플로우는 세계적인 컴퓨터 회사들에 의해 지속적으로 발전되어 왔다. 이를 바탕으로 최근에는 유비쿼터스 네트워크 환경에서의 로봇 서비스에 적용하는데 노력을 기울이고 있다. 이와 같은 유비쿼터스 로봇(URC)은 현재의 상황을 이해하여 사용자에게 서비스를 지능적이고 능동적으로 제공할 수 있어야 한다. 이를 위해 본 논문에서는 로봇 서비스를 표현할 수 있는 상황인지 워크플로우 언어(CAWL)를 바탕으로 상황인지 URC 서비스를 제공하기 위한 워크플로우 처리기를 제안한다. 제안하는 워크플로우 처리기는 상황정보를 서비스 실행을 위한 전이 조건으로 표현할 수 있는 CAWL 시나리오 문서를 통해 URC 서비스를 제공한다.

1. 서론

워크플로우는 세계적인 컴퓨터 회사들에 의해 지속적으로 발전되어 왔다. 워크플로우는 비즈니스 오퍼레이션의 자동화 및 정보 처리와 밀접하게 관련된 개념으로 제조 및 생산, 서비스 등 다양한 분야에서 공통적으로 나타나는 업무절차 처리를 위한 태스크를 개념적 수준에서 기술 한 것이다 [1]. 워크플로우 기술은 다양한 서비스 도메인에서의 이질적인 웹 서비스의 통합을 용이하게 할 수 있는 수단을 제공한다.

이를 바탕으로 최근에는 유비쿼터스 네트워크 환경에서의 로봇 서비스에 적용하는데 노력을 기울이고 있다. 정부에서도 URC 라는 개념의 네트워크 기반의 지능형 로봇 산업을 육성하고 로봇 기술을 고도화시키려는 노력을 기울이고 있다. 뿐만 아니라 전 세계적으로도 같은 목표를 가지고 지능형 로봇의 연구에 집중하고 있다. [2, 3]

URC 로봇은 다양한 주변 상황을 이해하여 사람에게 최적의 서비스를 제공할 뿐만 아니라 필요한 시점에 적절한 정보와 서비스를 지능적이고 능동적으로 제공할 수 있어야 한다 [4]. 이와 같은 서비스를 제공하기 위해 워크플로우를 유비쿼터스 컴퓨팅 환경에 적용해야 하는 필요성이 있다. 이에 본 논문에서는 CAWL 을 사용하여 URC 를 제공하기 위한 워크플로우 처리기를 제안한다. 제안하는 워크플로우 처리기는 컨텍스트 정보를 로봇 서비스 제어에 필요한 정보

로 바꾸어 로봇 서비스를 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 워크플로우의 관련 연구를 살펴본다. 3 장에서는 상황인지 워크플로우 처리기에 대하여 설명한다. 4 장에서는 시나리오를 통한 실험에 대하여 기술하고 마지막 5 장에서는 결론에 대하여 언급한다.

2. 관련 연구

최근 워크플로우 기술을 유비쿼터스 컴퓨팅 환경에 적용하려는 연구가 활발하게 이루어지고 있다 [3]. 이러한 연구와 관련하여 유비쿼터스 컴퓨팅 환경을 위한 상황인지 워크플로우 기술에 대하여 살펴본다.

WS-BPEL, WSFL, XLANG 는 비즈니스 및 분산 컴퓨팅 환경에 적용하기 위한 웹 서비스 기반의 언어들이다. 이 언어들 은 서비스 전이 조건으로 이전 서비스의 결과 값과 서비스 및 시스템의 이벤트 정보를 사용한다. 그러나 이러한 언어들 은 유비쿼터스 컴퓨팅 환경에 적용하기에 부족하다. 유비쿼터스 환경에서 서비스를 제공하기 위한 전이 조건은 사용자가 처해 있는 환경에서 발생하는 다양한 상황 정보를 사용해야 한다. 결과 값이나 이벤트 정보로는 상황 정보를 표현하기가 어렵다.

uFlow 는 uWDL 을 바탕으로 구축한 상황인지 워크플로우 서비스 프레임워크이다. uWDL 은 기존 웹 서비스 기반 워크플로우 언어들 이 서비스 전이 조건으

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 육성지원 사업의 연구결과로 수행되었음 (NIPA-2011-C1090-1121-0010)

¹ 교신저자

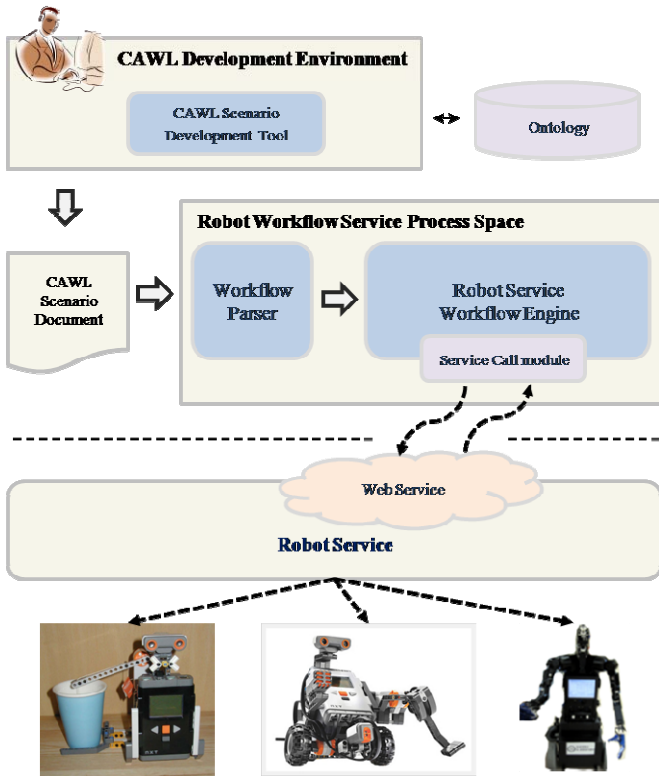
로 상황정보를 표현하지 못하는 점을 해결하기 위해 제안한 유비쿼터스 워크플로우 언어이다. uWDL은 워크플로우가 상황인지를 바탕으로 상태를 전이할 수 있도록 워크플로우의 상태 전이 조건에 컨텍스트 정보를 표현하고 있다. 그러나 uWDL은 시나리오 문서 내에 상황 정보를 정적으로 표현하고 있으며, 순차적인 하나의 워크플로우만을 표현할 수 있다. 따라서 uWDL은 상황 정보의 동적으로 표현하거나 다수의 워크플로우를 표현하는데 한계가 있다.

CAWL은 웹 서비스 형태로 설계된 이질적 환경의 서비스들을 이용하여 서비스의 흐름을 구성하고 상황 정보인 컨텍스트, 프로파일 및 이벤트 정보에 따라 사용자의 상황을 파악하여 유비쿼터스 컴퓨팅 환경에서 상황인지 기반의 서비스를 제공하기 위한 상황인지 워크플로우 언어이다. CAWL은 uWDL에 문제점을 해결하기 위해 동적으로 발생하는 컨텍스트 정보를 표현할 수 있도록 하였으며 서브-워크플로우 개념을 추가하여 복합 워크플로우 기능을 지원하였다.

본 논문에서는 앞서 언급한 CAWL의 개선된 기능을 바탕으로 로봇 서비스 시나리오 문서를 처리할 수 있는 워크플로우 처리기를 제안한다.

3. 상황인지 워크플로우 처리기

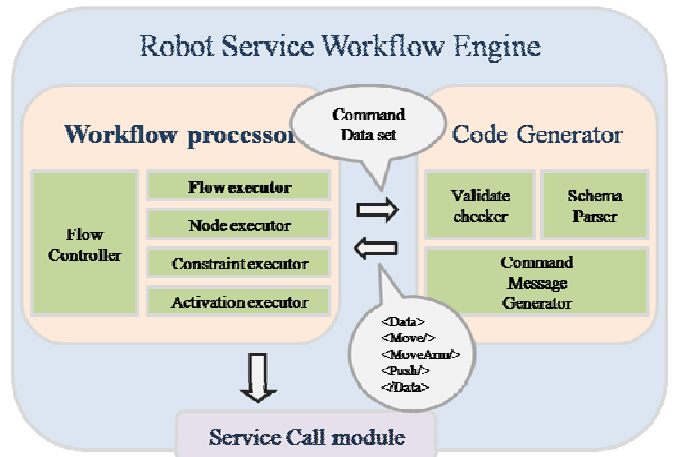
본 절에서는 유비쿼터스 네트워크 환경에서 상황인지 기반의 로봇 서비스를 지원하기 위한 상황인지 워크플로우 언어인 CAWL의 처리기에 대하여 살펴본다.



(그림 1) CAWL 기반 URC 제어 과정

그림 1은 CAWL을 기반으로 작성된 로봇 서비스 시나리오 문서를 처리하는 과정을 나타낸 것으로 일련의 과정은 다음과 같다. 사용자는 워크플로우 개발 환경에서 온톨로지 기반의 컨텍스트 어휘 정보와 워크플로우 시나리오 개발 도구를 사용하여 워크플로우 시나리오 문서를 작성한다. 작성한 문서는 로봇 워크플로우 서비스 처리 영역에 워크플로우 파서로 전달된다. 파서를 통해 DIAST(Document Instance Abstract Syntax Tree)가 생성되고, 생성된 DIAST는 로봇 서비스 워크플로우 엔진으로 전달된다. 이 때 워크플로우 엔진은 전달 받은 DIAST를 바탕으로 웹 서비스를 통해 로봇을 제어하기 위한 명령어를 생성한다. 이와 같은 제어 명령어를 통해 워크플로우 엔진은 웹 서비스 호출 모듈을 이용하여 호출한다. 이 때 웹 서비스가 호출되면 웹 서비스로 등록되어 있는 로봇 서비스가 실행되고, 로봇에게 생성된 제어 명령어가 전달되어 로봇을 움직이게 할 수 있다.

그림 2는 그림 1의 로봇 서비스 워크플로우 엔진을 좀 더 구체적으로 표현한 것이다. 워크플로우 처리기에서 로봇 명령어를 생성하기 위한 명령어 데이터를 xml 형태로 코드 생성기로 넘겨주게 된다. 넘겨 받은 xml 명령어 데이터에 유효성을 체크한다. 유효성 검사를 통과하게 되면 스키마 파서가 xml을 파싱하여 객체를 생성한다. 생성된 객체에 명령어 생성기를 통하여 값을 채워주게 된다. 채워진 값은 다시 xml 형태로 워크플로우 처리기로 넘겨진다. 워크플로우 처리기는 생성된 xml을 서비스 호출 모듈을 통하여 웹 서비스를 호출한다.

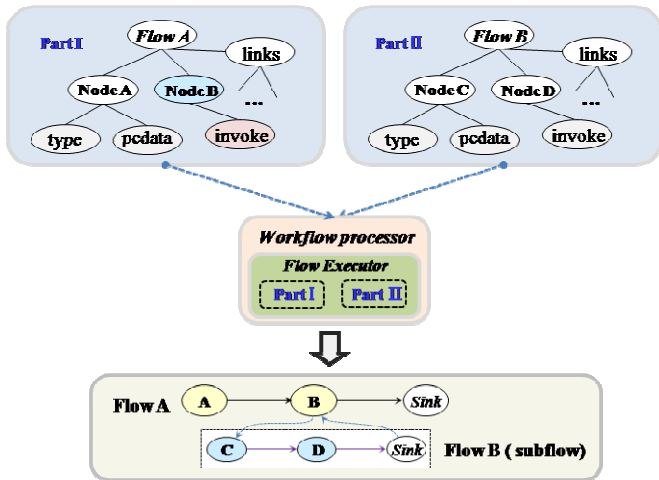


(그림 2) 로봇 서비스 워크플로우 엔진

그림 3은 그림 2에서 워크플로우 실행기(Flow executor)의 처리 과정을 상세히 그린 것이다. 워크플로우 처리기는 워크플로우의 실행 및 흐름을 제어하는 것이다. 워크플로우 파서를 통해 생성된 AST를 순회하며 각 노드의 링크 정보를 참조하여 시나리오에 기술된 워크플로우 흐름을 처리한다.

플로우 처리기에 입력 정보는 Part I, Part II로 구성된다. 좀 더 구체적으로 살펴보면 다음과 같다. Part I과 Part II는 Node A, Node B, links의 정보를 가지고

있다. Part I의 Node B는 Invoke를 사용하여 Flow B의 플로우를 참조한다. Flow B가 참조되는 하는 경우를 subflow로 지칭한다.



(그림 3) 서브플로우 처리과정

워크플로우의 처리 순서는 Flow A의 Node A, B가 수행된다. Flow B가 참조되어 Node C, Node D가 처리된다. Sink로 Flow B가 종료되어 Flow A로 돌아온다. Flow A에 Sink로 Flow A가 종료된다. CAWL은 다수의 워크플로우에 존재하는 개별적인 서비스 흐름을 하나의 워크플로우로 통합 표현할 수 있는 언어이다. CAWL을 사용하여 로봇 서비스를 기술할 때 subflow 형태로 기술한다.

4. 실험

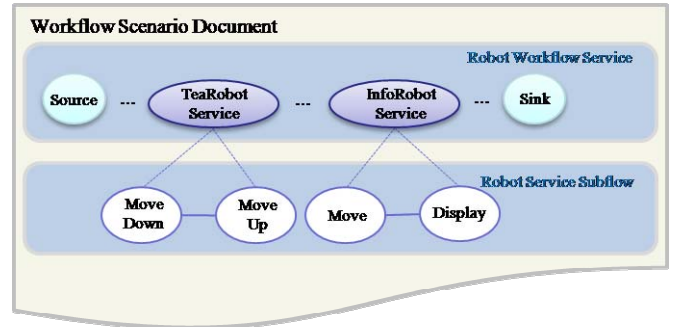
본 논문에서 제안하는 URC를 위한 상황인지 워크플로우 처리기는 CAWL을 사용하여 상황인지에 따른 로봇 서비스를 제공하는데 목적이 있다.

이를 위해 본 실험에서는 CAWL을 이용하여 상황인지를 통한 로봇 서비스를 제공하는 시나리오를 작성하였고 시나리오는 다음과 같다.

TeaRobot은 차를 준비하는 로봇이고 InfoRobot은 사람에게 정보를 알려주는 로봇이다. TeaRobot은 회의실에 있고 InfoRobot은 사무실에 있다. 사람 A는 사무실에서 발표를 준비하고 있다. 회의 시간 10분 전이 되면 TeaRobot은 차를 준비하기 시작한다. 차가 모두 준비되면 InfoRobot은 A에게 다가와 차가 준비되었고 회의실에 발표가 있음을 알려준다.

시나리오를 수행할 수 있는 워크플로우를 나타내면 그림 4와 같다. 로봇에 따른 명령어를 나누어 보면 TeaRobot은 차를 준비하기 위해 티백을 내리고 올리는 명령어가 필요하고 InfoRobot은 사람에게 이동하는 명령어와 정보를 전달하는 명령어가 필요하다. 실제 테스트를 위해 레고 로봇을 웹 서비스를 제공하는 서버와 블루투스로 연결하였고 웹 서비스를 사용하여 제어할 수 있도록 하였다.

그림 4는 CAWL을 기반으로 작성한 시나리오 문서를 워크플로우로 나타낸 것이다. 실행은 source, TeaRobot Service, InfoRobotService, Sink의 순서로 동작한다. TeaRobot 서비스에는 MoveDown과 MoveUp의 서브플로우가 기술되어 있고 InfoRobot 서비스에는 Move와 Display의 서브플로우가 기술되어 있다. 각 서브플로우는 왼쪽에서 오른쪽으로 실행된다.



(그림 4) 시나리오 문서의 워크플로우

표 1은 로봇 명령어를 사용하기 위해 기술된 일부 부분이다. 로봇 서비스를 수행하기 위해 serviceProvider에 "Subflow"라고 기술하여 서브플로우를 실행하도록 한다. 로봇 명령어를 위해 메시지를 선언하고 선언된 메시지에 어떤 로봇이 어떤 동작을 수행할지에 대한 정보를 입력한다. 값이 입력된 변수를 invoke태그를 사용해서 웹 서비스를 호출함으로써 로봇 서비스를 제공하게 된다.

<표 1> CAWL로 작성된 시나리오 문서의 일부분

```

<!-- 로봇 서비스 subflow 호출 -->
<invoke serviceProvider="Subflow"
  operation="InfoRobot" input=""/>
...
<!-- 로봇 명령어를 위한 메시지 선언 -->
<message name="requestRobotCMD">
  <part name="robotType" type="R-Type"/>
  <part name="robotSection" type="Section"/>
  <part name="robotCommand" type="Command"/>
  <part name="robotArgument" type="Argument"/>
</message>
...
<!-- 로봇 명령어에 변수 선언과 값을 입력 -->
<variable name="reqMove" type="requestRobotCMD">
  <initialize part="robotType">
    <from expression="InfoRobot"/>
  </initialize>
  <initialize part="robotSection">
    <from expression="Leg"/>
  </initialize>
  <initialize part="robotCommand">
    <from expression="Move"/>
  </initialize>
  <initialize part="robotArgument">
    <from expression="Tom"/>
  </initialize>
</variable>
...
<!-- 로봇 실행을 위한 웹 서비스 호출 -->
<invoke serviceProvider="ssCom"
  operation="legoService" input=""?reqMove"/>
    
```

표 1은 로봇을 제어하기 위해 온톨로지를 바탕으로

추상적으로 기술되어 있다. 로봇을 제어하기 위해 필요한 인자 값은 추상적이지 않다. 어떤 로봇에 모터를 얼마만큼 회전 시킬 것인지 정확하게 명시해 주어야 한다.

Technical Report TC00-1003, Workflow Management Coalition, 1994.

<표 2> 명령어 생성기를 통하여 생성된 xml

```
<ssCom:legoService xmlns:ssCom="http://Services:>
<ssCom:robotType>00:00:00:00:00:00</ssCom:robotType>
<ssCom:robotSection>MortorA,MortorB</ssCom:robotSection>
<ssCom:robotCommand>Move</ssCom:robotCommand>
<ssCom:robotArgument>50,80</ssCom:robotArgument>
</ssCom:legoService>
```

표 2는 명령어 생성기를 통하여 생성된 xml 형식으로 로봇을 식별하는 맥 어드레스, 동작시켜야 하는 부위, 어떻게 움직일 것인지에 대한 명령, 명령에 필요한 인자에 로봇에 필요한 값으로 변환된 것이다. 웹 서비스로 그 서비스에 필요한 인자를 정의해 둔다면 명령어 생성기를 통하여 xml을 생성할 수 있다.

5. 결론

본 논문에서는 URC 로봇을 위한 워크플로우 처리기를 제안하였다. 또한 제안하는 처리기의 실현을 입증하기 위해 레고 로봇을 웹 서비스로 제어할 수 있도록 만들어서 실험하였다. 제안하는 시스템은 웹 서비스 기술을 바탕으로 상황인지 기반의 URC 서비스를 제공하며, XML 형태의 명령어를 생성하여 로봇 서비스를 제공한다. 따라서 로봇 플랫폼에 독립적이며 호환성이 뛰어나다.

본 논문에서 제시한 시나리오는 로봇 명령어를 생성할 수 있다는 가능성을 보이기 위한 것으로, 레고를 가지고 웹 서비스를 생성하고 구현하였다. 따라서 실제 로봇에서 사용되기 위해서는 로봇 서비스를 정의하는 방식과 온톨로지에 대해서 더 연구가 필요하다.

참고문헌

- [1] D. Georgakopoulos and M. Hornick, "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure", Journal on Distributed and Parallel Database Systems, Vol3, No2, pp. 119-153, 1995
- [2] 김현, 조영조, 오상록, "URC(Ubiquitous Robotic Companion: 네트워크 기반 서비스 로봇", 정보과학회지 제 24 권 제 3 호, pp.5-11, 2006.
- [3] M Wieland, O. K., D Nicklas, F Leymann, "Towards context-aware workflows," CAiSE07 Proc. of the Workshops and Doctoral Consortium, 2007.
- [4] Joo Han, Eun Kim, Young Cho, Jaey Choi, "A Ubiquitous Workflow Service Framework", LNCS 3983, pp. 30-39, 2006.
- [5] Jun Li, Yingyi Bu, Shaxun Chen, Xianping Tao, Jian Lu, "FollowMe: On Research of Pluggable Infrastructure for Context-Awareness", 20th International Conference on Advanced Information Networking and Applications(AINA'06), Volume 1, pp. 199-204, 2006.
- [6] M. Wesier, "The Computer for the 21st Century", Sci. Amer., Sept., 1991
- [7] D. Hollingsworth, "The Workflow Reference Model",