

URC 서비스를 위한 상황인지 기반의 워크플로우 시스템

(A Context-aware Workflow System for URC Services)

최 종 선 [†] 곽 동 규 ^{**} 최 재 영 ^{***} 조 용 윤 ^{****}
 (Jongsun Choi) (Dong-gyu Kwak) (Jaeyoung Choi) (Yongyun Cho)

요 약 URC 로봇은 인간을 중심으로 발생하는 주변 상황을 이해하고, 그것에 따라 최적의 서비스를 제공하는데 목적이 있다. 따라서 URC 로봇을 위한 미들웨어에는 인간 친화적인 서비스 제공을 위한 상황 정보 기반의 프로세스 제어와 서비스 자동화 기술이 요구된다. 본 논문에서는 상황 정보에 따라 사용자에게 표준 웹 서비스 기반의 상황인지 URC 서비스를 제공하기 위한 워크플로우 시스템을 제안한다. 제안하는 시스템은 이기종의 URC를 제어하기 위해 플랫폼에 독립적인 XML 기반의 명령어 객체 모델을 제공하고, 웹 서비스 형태의 상황인지 URC 서비스를 지원한다. 따라서 제안하는 시스템은 다양한 로봇과 플랫폼이 존재하는 유비쿼터스 네트워크 환경에서의 URC 서비스에 대한 신뢰성을 향상시키고, URC 시스템의 기능 및 구조 변화에 대한 높은 유연성과 적응성을 제공할 수 있다.

키워드 : 유비쿼터스 네트워크 환경, URC, 상황인지, 워크플로우 시스템, 웹 서비스, 명령어 객체 모델

Abstract An URC (Ubiquitous Robot Companion) is aimed for providing the best service according to situational information that it recognizes. In order to offer human-friendly and intelligent services, a robot middleware requires the technique to automate and control URC service processes, which are based on context-awareness. In this paper, we propose a context-aware workflow system to provide web services based URC services according to situational information. The proposed system offers a platform-independent command object model to control heterogeneous URCs, and supports web services based context-aware URC services. Therefore, the proposed system can increase the reliability of URC services in ubiquitous network environment, on which the diverse URC robots and platforms exist. And it can enhance the flexibility and adaptability of the functional and structural changes of URC systems.

Key words : ubiquitous network environments, URC, context-aware, workflow system, web services, command object model

· 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음(NIPA-2009-(C1090-0902-0007))

[†] 정 회 원 : 숭실대학교 컴퓨터학부
jschoi@ss.ssu.ac.kr

^{**} 학생회원 : 숭실대학교 컴퓨터학부
coolman@ss.ssu.ac.kr

^{***} 종신회원 : 숭실대학교 컴퓨터학부 교수
choi@ssu.ac.kr
(Corresponding author인)

^{****} 정 회 원 : 국립 순천대학교 정보통신공학부 교수
yycho@sunchon.ac.kr

논문접수 : 2010년 2월 16일
심사완료 : 2010년 7월 7일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제37권 제9호(2010.9)

1. 서 론

유비쿼터스 컴퓨팅과 같은 새로운 패러다임의 등장으로 로봇틱스 분야에도 변화가 일고 있다. 3세대 로봇으로 불리는 유비쿼터스 로봇이 바로 그것이며, 이러한 패러다임의 변화는 로봇 산업의 초점을 산업용 로봇에서 자율방식으로 일상 공간에서 사람에게 유용한 서비스를 제공하는 지능형 서비스 로봇 분야로 확장시키고 있다. 이 같은 변화에 따라 최근 정부에서도 URC(Ubiquitous Robotic Companion)라는 개념의 네트워크 기반 지능형 로봇을 바탕으로 로봇 산업을 육성하고, 세계 최고 수준의 통신 인프라를 이용하여 로봇 기술을 고도화시키려는 노력을 기울이고 있다[1]. 아울러 전 세계적으로는 일본을 필두로 미국, 유럽 또한 같은 목표를 가지고 지

능형 로봇의 연구에 집중하고 있다.

URC 로봇은 다양한 주변 상황을 올바르게 인식하여 인간에게 최적의 서비스를 제공하는데 그 목적이 있다. URC 로봇을 위한 미들웨어는 인간 친화적인 서비스 제공을 위해 상황 정보를 기반으로 하는 프로세스 제어와 로봇 서비스에 대한 자동화 기술이 필요하다. 그러나 로봇이 주변 환경 및 상황을 인지하여 자동화 서비스를 독립적으로 해결하기에는 각각의 복잡성과 다양성이 크기 때문에 현재의 로봇 기술 수준으로는 해결하기 어려운 것이 사실이다[2]. 따라서 로봇 미들웨어는 로봇 스스로 해결하기 어려운 기능을 네트워크 인프라를 통해 해결해야 하며, 이를 기반으로 원격으로 제어 가능한 로봇을 통해 사용자에게 상황 정보 기반의 자동화 서비스를 제공해야 한다. 그리고 이와 같은 서비스를 실현하기 위해서는 BEA, IBM, Microsoft, Oracle 등과 같은 세계적인 기업들에 의해 업계에 성공적으로 자리잡은 워크플로우 기술을 유비쿼터스 네트워크 환경에 적용해야 할 필요성이 있다. 워크플로우는 다양한 서비스 도메인에서의 이질적인 웹 서비스의 통합을 용이하게 할 수 있는 수단을 제공한다. 또한 기계 학습, 추론 등과 같은 고도의 인공지능 기술이 필요한 에이전트 시스템에 비해 구현이 용이하여 URC 자동화 서비스를 신속히 개발할 수 있는 장점이 있다.

이를 위해 본 논문에서는 URC 로봇이 주변의 상황 정보에 따라 사용자에게 표준 웹 서비스 기반의 상황인지 서비스를 제공하는 워크플로우 시스템을 제안한다. 제안하는 시스템은 URC 로봇이 가져야할 필수 기능¹⁾ 가운데 프로세싱 기능에 해당하며, 이는 상황인지 기반의 워크플로우 기술을 바탕으로 한다. 제안하는 시스템은 크게 두 가지 기능을 갖는다. 우선 XML 기반의 명령어 객체 모델을 통해 생성한 명령어에 따라 로봇을 제어하는 것이고, 다른 하나는 웹 서비스 기술을 바탕으로 상황인지 기반의 URC 서비스를 제공하는 것이다.

본 논문에서 웹 서비스는 URC 응용 서비스를 지칭하는 것으로 제안하는 시스템이 각각의 상황에 적합한 조건에 따라 호출하는 대상이 되며, 웹 서비스를 활용하는 시스템은 SOAP과 같은 표준 인터페이스만으로 용이한 접근성을 가질 수 있다. 제안하는 시스템은 이와 같은 웹 서비스를 기반으로 하며, 상황 정보를 서비스의 전이 조건으로 활용할 수 있는 상황인지 워크플로우 언어인 CAWL[3]²⁾을 기반 기술로써 활용한다. CAWL은 서비스의 흐름을 표현하는 기존의 워크플로우 기능에 유비

쿼터스 컴퓨팅 환경을 위한 상황 정보의 표현기능을 갖고, 이를 바탕으로 복합적인 워크플로우 서비스를 지원한다. 따라서 제안하는 시스템은 앞서 언급한 특성을 통해 URC 서비스를 제공하는데 필요한 자원의 상호 연동성과 사용자의 요구에 따른 서비스 제공의 유연성을 향상시킬 수 있으며, 다양한 로봇과 플랫폼이 존재하는 유비쿼터스 네트워크 환경에서의 URC 서비스에 대한 신뢰성을 향상시키고, URC 시스템의 기능 및 구조 변화에 높은 적응성을 제공한다.

제안하는 시스템을 통해 사용자에게 자동화 서비스를 제공하기 위해서는 다음과 같은 일련의 과정이 필요하다. 우선 로봇 서비스의 흐름을 제어하기 위해 워크플로우 기반의 서비스 시나리오가 있어야 하며, 서비스 개발자는 상황인지 기반의 워크플로우 언어인 CAWL을 이용하여 서비스 시나리오를 작성해야 한다. 제안하는 시스템은 이 문서를 바탕으로 실제 로봇 클라이언트를 작동시키기 위한 제어 명령어를 생성하고, 로봇 클라이언트의 실행을 위해 이를 관리하고 실행시키는 고성능의 URC 서버에 명령어를 전달한다.

본 논문의 구성은 다음과 같다. 2장에서는 상황인지 기반 워크플로우 및 로봇 미들웨어의 연구 동향을 살펴보고, 3장에서는 제안하는 시스템의 설계 및 구조에 대한 설명을 주요 기능을 중심으로 기술한다. 4장에서는 제안하는 시스템의 실행 가능성을 입증하기 위해 시나리오를 바탕으로 로봇 서비스를 제공하는 과정을 보여주며, 마지막으로 결론으로 끝맺는다.

2. 관련연구

최근 컴퓨팅 패러다임의 변화에 따라 WSBPEL[4]과 같은 비즈니스 프로세스 통합을 위한 자동화 모델인 워크플로우 기술을 유비쿼터스 컴퓨팅 환경에 적용하려는 연구가 활발하게 이루어지고 있다[5]. 또한 이와 같은 환경에 적용하기 위한 연구는 상황인지 워크플로우[6-9], 또는 스마트 워크플로우[10] 등과 같은 개념으로 시도되고 있다. 이들 연구와 관련하여 본 장에서는 유비쿼터스 컴퓨팅 환경을 위한 상황인지 워크플로우 기술과 이를 기반으로 하는 상황인지 미들웨어, 그리고 지능형 로봇을 위한 URC 미들웨어에 대하여 살펴본다.

2.1 상황인지 기반의 워크플로우

Context4BPPEL[7]은 WS-BPEL를 확장하여 이를 실행하기 위한 플랫폼의 구조를 제안하고 있으며, 이를 바탕으로 확장한 연구[10]에서는 스마트 워크플로우 개념을 소개하였다. Context4BPPEL은 WSBPEL을 확장하였으므로 이를 처리하기 위한 엔진의 추가적인 연구가 필요하다. FollowMe[8]는 유비쿼터스 컴퓨팅 환경을 위한 OSGi 프레임워크로써, 유비쿼터스 컴퓨팅 환경에서 존

1) 센싱 기능(외부 센서 이용), 프로세싱 기능(원격의 고성능 서버 이용), 행동 기능(로봇의 구동).

2) Context-aware Workflow Language의 약어로 로봇 서비스를 표현하기 위한 상황인지 워크플로우 언어임.

제하는 다양한 도메인에 적용할 수 있는 상황인지 워크플로우 기반구조를 제공한다. FollowME의 초점은 디바이스의 제어와 사용자 어플리케이션에 국한되어 있으며, 또한 유비쿼터스 컴퓨팅 환경에서 네트워크에 연결된 디바이스의 제어를 위해 상황인지 기반의 워크플로우 엔진을 제안한 연구[11]는 리즈닝 메커니즘에 초점을 맞추고 있다.

uFlow[9]는 기존 웹 서비스 기반 워크플로우 언어들이 서비스 전이 조건으로 상황 정보를 기술하지 못하는 점을 해결하기 위하여 새로운 유비쿼터스 워크플로우 언어인 uWDL[12]을 기반으로 하는 상황인지 워크플로우 서비스 프레임워크이다. uWDL은 워크플로우가 주변 상황에 대한 상황을 인지하고, 상태를 전이할 수 있도록 워크플로우의 상태 전이 제약 조건에 상황 정보를 명시하기 위한 구조적 컨텍스트³⁾ 모델[9]을 기반으로 하고 있으며, 이러한 uWDL을 통해 시나리오에 기술한 상황 정보에 따라 사용자의 서비스 요구를 처리하고 실행한다. 그러나 uWDL은 시나리오 문서 내에 상황 정보 정보를 정적으로 기술하고, 하나의 워크플로우만을 기술할 수 있으므로 다수의 워크플로우 서비스를 표현하는데 한계가 있다.

상황인지 워크플로우에 관한 또 다른 연구[5]에서는 상황인지 워크플로우의 적용에 대하여 상황인지 워크플로우와 관련된 연구 사례를 통해 다양한 접근방법에 대하여 조사하였다. 이 연구에서는 ContextBPTEL을 응용 서비스 수준에서 사용자의 주변 상황을 추상적으로 기술하는 추상적 워크플로우로 분류하고 있으며, 본 논문에서 제안하는 시스템을 포함하여 FollowMe, uFlow 등은 ContextBPTEL과 같은 추상적인 워크플로우의 기능을 포함하면서 이를 처리할 수 있는 워크플로우 엔진도 포함하는 구체적 워크플로우로 분류하고 있다. CAWL은 uWDL에서의 문제점을 해결하기 위해 동적으로 발생하는 컨텍스트 정보를 바탕으로 상황 정보를 표현할 수 있도록 하였으며, 서브워크플로우⁴⁾와 같은 복합 워크플로우를 지원하여 여러 워크플로우를 통합 표현할 수 있는 기능을 갖게 되었다. 본 논문에서는 서브워크플로우를 이용하여 로봇 서비스의 흐름을 표현한다.

2.2 URC를 위한 미들웨어

현재 OMG에서 로봇 미들웨어의 표준을 이끌고 있는 일본을 필두로 한 세계 각 나라들은 네트워크 기반의

지능형 로봇을 활용하여 로봇 산업을 육성하기 위한 목표를 가지고 있으며, 특히 우리나라 정부는 세계 최고 수준의 통신 인프라를 이용하여 로봇 기술을 고도화하려는 노력을 기울이고 있다[2]. 이 같은 노력의 일환으로 국내의 경우 ETRI에서는 2005년부터 USN 기반 Ubiquitous Robotic Space 과제를 통하여 RFID, UWB, ZigBee 등의 네트워크 환경을 적극 이용하는 지능형 서비스 로봇을 개발하고 있다[13]. ETRI에서 개발한 상황인지 기반 미들웨어인 CAMUS는 CORBA를 기반으로 개발하고 있는 URC 시스템을 위한 상황인식 미들웨어로, URC 환경 내에서 획득된 상황 정보를 기반으로 환경 내에 있는 사용자에게 적절한 서비스를 제공할 수 있도록 상황 기반 응용의 개발과 실행을 지원한다[14-16]. 국내의 또 다른 대표적인 연구[17]는 KAIST에서 제안한 다중 계층 구조의 미들웨어가 있다. 이 연구에서는 유비쿼터스 환경에서 서로 다른 HW와 SW 플랫폼들 간 상호운용성의 문제에 대한 해결책을 네트워크 기반의 U-Space⁵⁾에서 Ubibot이라는 지능형 로봇을 통해 제시하고 있다.

국외의 경우는 일본의 산업기술종합연구소(AIST)에서 진행 중인 OpenRTM-aist[18]가 있으며, 2006년 표준으로 채택되었다[19]. RT-미들웨어는 분산 객체 기술에 바탕을 둔 공용 플랫폼이며, RT-컴포넌트⁶⁾를 통합함으로써 네트워크로 연결된 다양한 로봇 시스템의 구축을 지원한다. 로봇 시스템을 지원하는 소프트웨어 개발은 각각의 로봇 시스템이 가지고 있는 특수성 때문에 재사용성이 떨어지는 문제점이 있다. 이를 위해 RT-미들웨어는 네트워크 기반의 다양한 로봇 디바이스들을 RT-컴포넌트와 같은 개념을 적용하였으며, 이를 활용하여 소프트웨어의 모듈화를 바탕으로 한 컴포넌트의 재사용성을 향상시킨다[20].

이와 같이 컴포넌트를 기반으로 한 로봇 미들웨어들[16,20]은 현재 지속적으로 연구가 진행 중에 있으나 실제 URC 로봇에 미들웨어를 적용하여 구현한 사례가 아직까지 없어 신속히 개발하는데 어려움이 있음을 알 수 있다. 따라서 본 논문에서는 궁극적으로 추구하는 방향은 자동화된 URC 서비스로써 동일하다. 그러나 본 논문에서는 실제 개발에 있어서 검증된 기술인 워크플로우 기술을 채택하였으며, 이에 상황 정보와 URC 로봇 서비스의 흐름을 표현할 수 있는 CAWL을 활용하였다.

3) 센서로부터 전달받은 정보를 저수준의 컨텍스트로 저장하며, 이를 (주어, 동사, 목적어)와 같은 RDF 기반의 표현 방법을 사용한 형태를 고수준의 컨텍스트 정보로 분류한다. 본 논문에서는 저수준의 컨텍스트를 "컨텍스트"로, 고수준의 컨텍스트 정보를 "상황 정보"로 정의한다.

4) 서브 워크플로우(subworkflow)는 미리 정의된 다른 하나의 워크플로우에 속한 서비스 노드를 표현하기 위한 개념으로, 구조적 프로그래밍 방법론에서 한 루틴에서 다른 루틴(subroutine)을 호출하는 메커니즘과 동일함.

5) 연구 [17]에서 정의한 가상의 유비쿼터스 공간임.

6) RTC(Robot Technology Component): 로봇 소프트웨어 컴포넌트 모델 미들웨어로서, 로봇 소프트웨어 컴포넌트들 사이에 인터페이스가 되도록 하기 위하여 OMG에 의하여 채택된 미들웨어 모델 규격의 표준이며, OMG에서는 향후 만들어지는 소프트웨어 컴포넌트는 RTC 규격을 따르도록 권장한다(6).

3. 제안하는 URC 워크플로우 시스템

본 장에서는 유비쿼터스 네트워크 환경에서 URC 로봇 서비스를 제어하기 위한 상황인지 기반의 워크플로우 시스템에 대하여 살펴본다. 제안하는 시스템의 목적은 상황인지 워크플로우 언어인 CAWL을 이용하여 작성한 로봇 서비스 시나리오를 바탕으로, 이를 처리하여 사용자가 처한 상황에 적합한 URC 로봇 서비스를 제공하는 것이다.

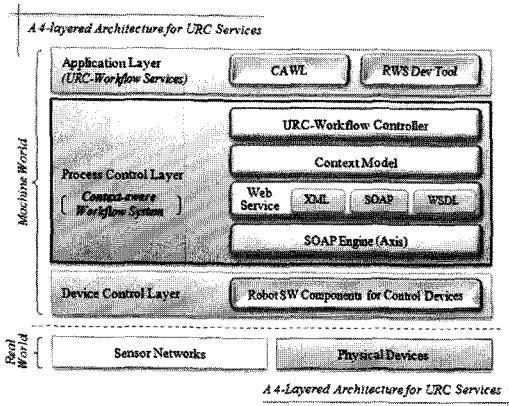


그림 1 제안하는 상황인지 워크플로우 시스템 개념도

그림 1은 워크플로우를 기반으로 하는 미들웨어의 개념적 구조를 나타낸다. 이것은 사용자의 실제 환경을 나타내는 실세계(Real World)와 컴퓨팅 시스템 환경을 나타내는 기계 영역(Machine World)으로 나눌 수 있다. 특히 본 논문에서는 제안하는 시스템은 기계 영역에서 프로세스 제어 계층(Process Control Layer, 이하 PCL)으로 표현하였다.

3.1 URC 서비스를 위한 미들웨어

본 절에서는 제안하는 시스템을 중심으로 이와 관련된 주변 계층에 대하여 살펴보고, 이와 더불어 센서 네트워크 및 로봇의 물리적 디바이스에 대하여 간략히 언급하도록 한다. 제안하는 상황인지 기반의 워크플로우 시스템은 그림 1에서 기계 영역의 프로세스 제어 계층(PCL)에 해당한다. 기계 영역은 크게 3계층(응용 계층, 프로세스 제어 계층, 디바이스 제어 계층)으로 나뉜다. 응용 계층은 사용자에게 제공되는 최상위 수준의 URC 서비스를 정의한다. 그리고 프로세스 제어 계층은 AL에서 정의한 서비스를 제어한다. 마지막으로 디바이스 제어 계층은 로봇의 물리적 디바이스를 관리 및 제어한다.

■ 응용 계층(Application Layer, 이하 AL)

AL은 PCL의 수행을 위한 입력 정보를 제공해주는 것으로, 워크플로우 개발 도구(그림 1에서 RWS Dev

Tool)를 이용하여 로봇 서비스를 표현하는 역할을 수행한다. AL에서 기술하는 로봇 서비스는 상황인지 기반 워크플로우 언어인 CAWL을 이용하여 기술한다. 세부적인 내용은 3.2절에서 설명한다.

■ 프로세스 제어 계층(Process Control Layer, 이하 PCL)

PCL은 제안하는 URC 서비스를 제어하기 위한 상황인지 기반의 워크플로우 시스템에 해당하는 부분이다. PCL의 역할은 AL으로부터의 입력 정보(로봇 서비스 시나리오 문서)를 전달받아, 이를 처리하고 제어하는 것이다. 그림 1에서와 같이 PCL은 프로세스 제어를 위한 워크플로우 제어기, 상황 정보를 표현하기 위한 컨텍스트 모델 그리고 이들의 기반 기술인 웹 서비스, 그리고 URC 미들웨어 계층과의 통신을 위한 SOAP 엔진으로 나눌 수 있다.

• 워크플로우 제어기(URC-Workflow Controller)

워크플로우 제어기의 역할은 로봇 서비스의 흐름을 워크플로우를 이용하여 실행 및 제어하고, 로봇에게 전달할 명령어를 생성하는 것이다. 따라서 제어 모듈에는 워크플로우 실행 및 제어를 위한 파서 및 실행 엔진이, 로봇 서비스 제공을 위한 명령어를 생성하는 로봇 명령어 생성기(Robot-CMD Generator)가 포함되어 있다. 보다 구체적인 내용은 3.2절에 언급하기로 한다.

• 컨텍스트 모델(Context Model)

PCL은 상황인지를 위해 상황비교 모듈을 포함한다. 상황비교 모듈의 역할은 시나리오 문서에 RDF 형식으로 기술한 상황 정보와 센서로부터 전달받은 저수준의 컨텍스트 정보를 가공한 상황 정보가 일치하는지를 검증하는 것이다. 이때 센서로부터 전달 받은 가공되지 않은 컨텍스트 정보는 구조적 컨텍스트 모델(structural context model)[6]을 통해 RDF 형식의 상황 정보로 가공된다. 이와 같이 가공된 상황 정보와 시나리오 문서에 작성한 상황 정보는 RDF 형식의 (주어, 동사, 목적어)와 같은 형태를 가진 엔티티(entity)들에 대한 집합으로 표현된다. 워크플로우 제어기는 이와 같은 과정을 거쳐 가공된 상황 정보를 시나리오 기반의 워크플로우 서비스를 실행하기 위한 전이조건으로 활용한다.

• 웹 서비스 및 통신 모듈

상황인지 기반의 워크플로우 언어(CAWL)는 XML 기반의 웹 서비스를 기반 기술로 활용한다. 따라서 CAWL을 이용하는 AL 및 PCL은 웹 서비스가 포함되어 있는 요소 기술을 근간으로 한다. 그리고 PCL에서 생성된 추상적인 로봇 명령을 URC 미들웨어로 전달하기 위해, 통신 수단으로 Axis와 같은 SOAP 엔진을 이용한다.

■ 디바이스 제어 계층((Device Control Layer, 이하 DCL)

DCL은 상황인지 기반의 워크플로우 시스템을 기반으

로 로봇 디바이스를 제어하고 관리하는 미들웨어이다. 그림 1에서와 같이 DCL 로봇 디바이스를 제어하기 위한 SW 컴포넌트들로 구성된다. DCL은 PCL과의 통신 수단으로 Axis를 동일하게 이용하며, 로봇 디바이스를 제어하기 위한 SW 및 이를 관리하기 위한 방법을 제공한다. 그리고 URC 서버들 간의 호환성을 통해 로봇 클라이언트 개발자들이 URC 서버에 관계없이 로봇 플랫폼을 개발할 수 있는 환경을 제공한다. 또한 URC 로봇 서비스를 제공하기 위한 DCL의 주된 역할은 상위 계층(AL, PCL)으로부터의 결과 값(로봇 서비스 명령어)을 전달받아, 실제로 실행할 로봇에게 전달하는 것이다.

■ 센서 네트워크 및 물리적 디바이스

센서 네트워크는 PCL의 상황인지 워크플로우 시스템 및 클라이언트 로봇에 가공되지 않은 센서 정보, 즉 컨텍스트 정보(사용자 프로파일, 위치, 시간, 온도, 습도 등)를 제공한다. 워크플로우 시스템은 사용자의 상황을 판단하기 위해 그리고 클라이언트 로봇의 각종 센서들은 로봇 자신의 주변 상황을 판단하기 위해 컨텍스트 정보를 바탕으로 가공한 상황 정보를 이용한다.

3.2 URC 로봇 서비스 제어

본 절에서는 제안하는 상황인지 기반의 워크플로우 시스템(프로세스 제어 계층)을 이용하여 URC 로봇 서비스의 흐름을 제어하는 과정에 대하여 설명한다. 그림 3은 그림 1의 PCL에서 URC-Workflow Controller(이하 워크플로우 제어기) 모듈을 좀 더 구체적으로 표현한 것이다. 워크플로우 제어기의 역할은 크게 두 가지로 나눌 수 있다. 하나는 로봇 디바이스의 구동을 위해 명령어 객체 모델을 이용하여 로봇 명령어를 생성하는 것이고, 다른 하나는 URC 로봇 서비스를 실행 및 제어하는 것이다. 전자는 그림 2의 명령어 생성기(CMD Set Generator)가, 후자는 워크플로우 처리기(Workflow Handler)가 각각의 역할을 담당한다.

■ URC 로봇 서비스 제공을 위한 처리 과정

그림 2는 URC 로봇 서비스의 처리과정을 워크플로우 제어기를 중심으로 나타낸 것으로, 일련의 과정은 다음과 같다. 워크플로우 제어기는 AL에서 시나리오 개발자가 기술한 XML 기반의 로봇 서비스 시나리오 문서를 입력정보로 받으며, 파서는 입력정보를 바탕으로 DITree(Document Instance Tree)를 생성한다. 이때 DITree는 시나리오 문서 전체에 대한 AST(Abstract Syntax Tree)를 나타내는 것으로, 워크플로우 처리 엔진(URC-Workflow Processing Engine)은 워크플로우를 실행시키기 위해 DITree를 입력정보로 사용한다. 워크플로우 핸들러(Workflow Handler)는 AST를 로봇 서비스 풀(URC Service Pool, 이하 서비스 풀)에 저장하고, 생성된 로봇 명령어를 로봇 제어 계층(DCL)에 전달한다.

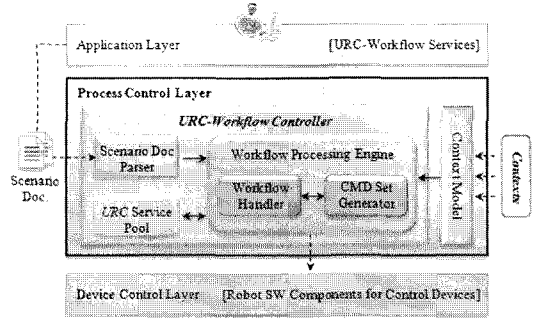


그림 2 프로세스 제어 계층의 로봇 서비스 실행 제어 과정

사용자에게 받았던 서비스를 다시 받아야 하는 경우가 종종 발생할 수 있다. 이러한 경우를 대비하여 워크플로우 제어기는 파서를 이용하여 시나리오 문서를 파싱하여 AST 객체 형태로 서비스 풀에 저장해 둔다. 만약 이전에 제공하였던 서비스를 다시 제공해야 할 경우, 워크플로우 제어기는 시나리오 문서를 파싱하지 않고 저장해 두었던 AST 객체를 재사용하여 워크플로우 서비스를 실행시킨다.

■ URC 로봇 서비스의 흐름 제어

PCL의 워크플로우 제어기(URC-Workflow Controller)는 실제 주변상황으로부터의 센서 네트워크를 통해 얻은 상황 정보(사용자 프로파일, 디바이스 상태, 위치, 습도 정보 등)와 로봇 서비스 개발 도구(RWS Dev Tool)를 이용하여 작성한 로봇 서비스 시나리오 문서를 입력정보로 활용한다. PCL은 두 입력정보를 비교함으로써, 주변의 상황을 판단하여 사용자에게 적합한 로봇 서비스를 제공한다. 구체적으로 살펴보면 워크플로우 파서는 PCL에 입력된 로봇 서비스 시나리오 문서를 파싱한다. 상황 비교 모듈은 RDF 형식으로 작성된 시나리오 문서의 상황 정보와 실제 각종 센서로부터 얻은 정보를 가공한 상황 정보를 비교하여 사용자가 처한 환경이 시나리오에 작성된 상황 정보와 같은지를 검증한다. 워크플로우 실행 엔진은 시나리오에 기술된 로봇 서비스의 실행 조건에 따라 서비스를 실행시킨다.

3.3 워크플로우 기반의 URC 로봇 서비스

본 절에서는 상황인지 기반 워크플로우 언어인 CAWL을 이용하여 URC 로봇 서비스를 어떻게 표현하는지에 대해 언급하고, URC 로봇 서비스를 제공하기 위해 PCL에서 로봇 명령어를 어떻게 생성하는지에 대하여 살펴본다.

■ URC 로봇 서비스의 표현 방법

예를 들어 사용자가 제공받아야 할 서비스 중 로봇이 문을 열어주는 서비스가 있다고 가정하자. 이때 서비스 개발자는 지원 가능한 로봇의 물리적 특성을 고려하여

- ① 로봇은 문으로 이동한다.
- ② 로봇의 팔을 들어올린다.
- ③ 로봇은 문고리를 잡는다.
- ④ 로봇은 문을 연다.

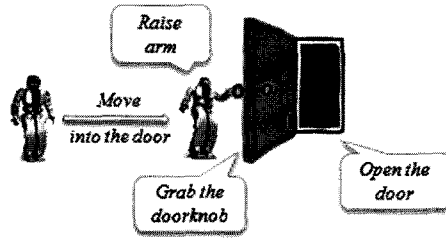


그림 3 로봇 서비스의 표현

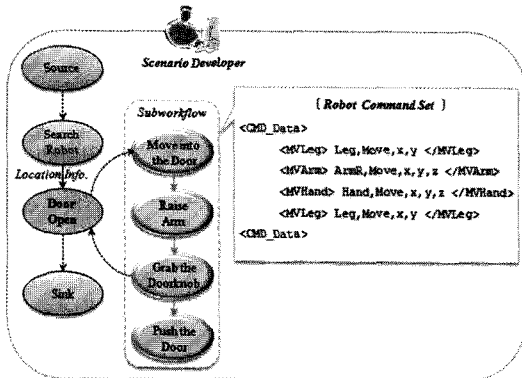


그림 4 로봇 서비스 제공을 위한 명령어 생성 과정

문을 열어주는 서비스를 그림 3과 같이 세분화하여 기술할 수 있다. 본 논문에서는 이와 같은 로봇 서비스를 그림 4에서와 같이 서브워크플로우로 표현한다.

■ URC 로봇 명령어의 생성

그림 2에 표현한 로봇 명령어 생성기(CMD Set Generator)는 입력받은 시나리오 문서로부터 서브워크플로우로 표현된 로봇 서비스를 제공하기 위해 표현된 부분만을 추출하여 실제 로봇 디바이스를 제어하기 위한 로봇 명령어를 생성한다. 생성된 명령어는 XML 형태의 로봇 제어 정보를 담고 있으며, 내용은 그림 4와 같다. 이와 같은 명령어 생성을 위한 모델은 그림 5에 표현하였다.

앞서 언급했던 바와 같이 생성된 명령어 정보는 DCL을 통해 서비스를 제공해야 할 로봇에게 전달된다. 그림 4는 XML 기반의 명령어 집합의 표현 방법을 나타낸다. 그림 4에서 “Door Open” 노드는 사용자에게 제공할 서비스를 표현하는 서비스 노드 중 로봇 서비스에 해당하는 부분이다. 이때 로봇 서비스에 해당하는 부분은 서브워크플로우로 기술하며, 로봇 명령어 생성기는 이 부분

에 대하여 로봇 명령어 집합(Robot Command Set)을 생성한다.

그림 4에서 표현한 로봇 명령어 집합은 그림 5에 정의한 명령어 객체 모델을 기반으로 작성된 것으로 워크플로우 시스템이 해당 노드의 웹 서비스를 호출할 때 매개변수 형태로 전달되는 내용을 포함한다. 일반적으로 시나리오 개발자는 로봇을 제어하기 위한 정보를 표현하기 위한 방법으로 그림 5에서와 같은 명령어 객체 모델을 이용하여 로봇의 제어 정보를 표현할 수 있다.

예를 들어 로봇 명령어 집합의 4번째 줄에 표현한 <MVleg>Leg, Move, x, y<MVleg>에서 <MVleg>는 다리(또는 이동 바퀴)를 제어하기 위해 제안하는 시스템 내에서 정의한 명령어 태그이고, 명령어의 첫 번째 인자(Leg)는 명령모드(Mode)를, 두 번째 인자(Move)는 실제 액추에이터를 제어할 수 있는 명령어의 이름을, 그 다음 인자들은 좌표 정보와 같은 실제 이동 또는 동작을 위한 제어 정보를 포함한다. 예를 들어 공간을 이동하는 팔(Arm)의 경우에는 로봇이 이해할 수 있는 수준의 3차원 좌표(x, y, z)를 매개변수로 사용한다.

그림 5에서와 같은 모델을 활용하여 작성 가능한 로봇 명령어는 제한적인 환경을 가정하였을 때 표 1과 같이 나타낼 수 있다. 표 1은 오피스 환경에서 회의 준비 서비스를 제공함에 있어 로봇 및 스마트 디바이스를 제어하여 제공받을 수 있는 서비스와 그에 해당하는 명령어의 리스트를 표현한 것이다. 각각의 명령어 리스트는 일반화하여 표현하였으며, 실제 로봇을 제어하기 위해서는 명령어의 인자 값에 실제 로봇이 이해할 수 있는 수준의 제어 정보를 포함해야 한다. 사무실에서 어떤 사용자의 일정 중에 회의 준비 서비스가 제공되어야 한다면, 회의 시간을 앞두고 표 1에서 정보 출력을 제외한 서비스들이 사전에 제공될 수 있다. 그리고 정보 출력 서비스 같은 경우는 회의 준비 서비스가 종료되고 나면 사

```
<CMDTag> MODE, COMMAND, ADDITIONAL_ARGUMENTS </CMDTag>
```

그림 5 명령어 객체 모델

표 1 로봇 및 스마트 디바이스에 대한 제어 명령어 정보

서비스 분류		로봇 및 스마트 디바이스 제어 명령어 정보	장치
회의 준비	조명	<Light> Light, setLED, On </Light> <Light> Light, setLED, Off </Light>	스마트 디바이스
	음악 재생	<Music> Mp3, PlayOn, Filename </Music> <Music> Mp3, PlayOff, Filename </Music>	
	회의 자료	<Download> PC, Download, Filename </Download> <Execute> PC, executePPT, Filename </Execute>	
	차 준비	<MVArm> Arm, MoveUp, angle </MVArm> <MVArm> Finger, Grab, angle </MVArm> <MVLeg> Leg, Move, Location </MVLeg> <MoveArm> Arm, MoveDown, angle </MoveArm>	로봇 디바이스
	정보 출력	<Display> LCD, Display, OutputInfo </Display> <Display> Voice, PlaySound, OutputInfo </Display>	

용자에게 로봇이 다가와 서비스 준비가 완료되었음을 알려줄 수 있을 것이다. 또한 음성 서비스가 가능한 로봇이라면 음성으로 서비스의 준비 상황을 알려줄 수 있을 것이다.

이와 같은 로봇 명령어 집합을 구성하는 각각의 명령어는 실제 서비스를 제공하는 로봇의 제어 명령어를 나타내는 스키마를 기반으로 작성된다. 그리고 이와 같은 스키마 정보는 웹 서비스로 배포하여 공유할 수 있으며, 따라서 PCL과 DCL에서 모두 명령어에 관한 스키마 정보를 공유할 수 있다. 이를 바탕으로 PCL은 실제 로봇 디바이스가 이해할 수 있는 로봇 명령어 정보를 전달하기 위해, 공유하고 있는 스키마를 바탕으로 로봇 명령어를 생성하여 DCL로 전달한다. 따라서 이와 같이 생성된 로봇 명령어 정보는 DCL에서 로봇을 구동하기 위한 명령어로 활용된다.

4. 실험

본 논문에서 제안하는 워크플로우 시스템의 설계 목적은 원격으로 제어 가능한 URC 로봇을 통하여 사용자에게 필요한 로봇 서비스를 제공하는 것이다. 이를 위해 본 실험에서는 상황인지 기반 워크플로우 언어인 CAWL을 이용하여 로봇 서비스 시나리오를 작성하였으며, 이를 이용하여 사용자에게 적합한 서비스를 제공하는지 시뮬레이션 실험으로 확인한다.

본 실험을 위한 시스템의 배치는 그림 6과 같다. 그림에서는 URC 서버와 워크플로우 시스템을 분산 네트워크 환경으로 표현하고 있지만, 리치 클라이언트 로봇의 경우에는 워크플로우 시스템과 URC 서버가 함께 리치 클라이언트 로봇에 탑재될 수도 있다. 이와 같은 배치는 어떤 경우가 되었든지 명령의 전달체제는 동일하며, 사용자의 요구 사항에 대한 인식을 “로봇 자체가 하느냐?” 아니면 “분산 환경에 있는 시스템이 하느냐?”의

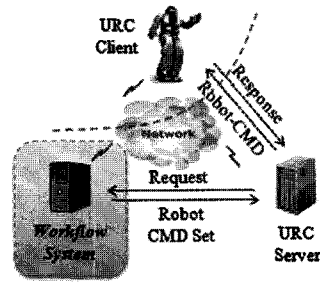


그림 6 로봇 서비스를 위한 시스템 배치

차이만 있을 뿐이다. 그림에서 워크플로우 시스템은 로봇을 제어하기 위해 명령어를 URC 서버에 전달하고 URC 서버는 디바이스를 구동할 클라이언트 로봇에게 명령을 지시한다. 이때 워크플로우 시스템과 URC 서버는 URC 클라이언트가 이해할 수 있는 로봇 명령어에 대한 XML 기반의 스키마를 공유한다.

실험 내용은 먼저 로봇 서비스 시나리오 문서를 CAWL을 기반으로 작성하고, 이를 바탕으로 로봇 디바이스를 제어하기 위한 명령어를 생성하는 과정과 생성된 명령어를 디바이스 제어 계층(DCL)에 전달하는 내용을 보이는 것이다. 실험을 위한 로봇 서비스의 구체적인 내용은 그림 3에 표현한 것과 같이 “로봇이 문으로 이동하여 팔을 들어 올린 후, 문고리를 잡아 문을 열어 준다”는 것이다. 그리고 이와 같은 로봇 서비스의 흐름은 그림 4에서 워크플로우 모델을 이용하여 표현하였다.

실험을 위해 사용자(마이클)가 상황 정보에 따라 분기를 결정하는 워크플로우 시나리오를 사용하며, 시나리오의 내용은 그림 7에 기술하였다. 시나리오의 내용은 일반적인 사용자가 겪을 수 있는 상황을 가정하여 작성하였다.

그림 8은 서비스 시나리오를 바탕으로 작성한 로봇 서비스 시나리오 문서의 일부분이다. 그림에서 “Door

몸이 불편하여 휠체어를 타고 있는 마이클은 아직 수동인 여닫이문이 있는 방으로부터 나가려고 한다. 이때 "문을 열어줘"라고 말하면, 음성 인식 센서가 장착된 고성능 서버는 이를 인식하고, 마이클과 가장 근접한 위치에 있는 로봇을 검색하여 이 로봇에 대한 정보를 워크플로우 시스템에 전달한다. 이 때 워크플로우 시스템은 해당 로봇이 이해할 수 있는 제어 명령어를 생성하여 고성능 서버에 전달하고, 이 명령어는 다시 해당 로봇에게 전달되어 문을 열어주는 서비스를 제공하게 된다.

그림 7 로봇 서비스가 요구되는 서비스 시나리오

```

<CAWL name="HomeRobotService" version="2.0.1">
... (중략) ...
  <flow name="Michael_ServiceFlow">
... (중략) ...
    <node name="DoorOpen">                                <!-- 그림 4의 3번째 노드 -->
      <condition expression="c1">
        <context>
          <rule contribute="positive">
            <constraint name="c1">
              <subject type="Location"?PersonLoc</subject>
              <verb>is</verb>
              <object type="Location"?InVar/RobotLoc</object>
            </constraint>
          </rule>
        </context>
      </condition>
      <invoke serviceProvider="AnyCompany" portType="DoorOpenPT"
        operation="DoorOpenFlow" subflow="true"/>
    </node>
... (중략) ...
    <link to="SearchRobot" from="Source"/>
    <link to="DoorOpen" from="SearchRobot"/>
    <link to="Sink" from="DoorOpen"/>
  </flow>

  <flow name="DoorOpenFlow">                            <!-- 그림 4의 Subworkflow -->
... (중략) ...
    <node name="MoveIntoTheDoor">
      <condition expression="c1 and c2">
        <context>
          <rule contribute="positive">
            <constraint name="c1">
              <subject type="Location"?currentRobotLoc</subject>
              <verb>is</verb>
              <object type="Location"?InVar/RobotLoc</object>
            </constraint>
          </rule>
        </context>
      </condition>
      <invoke serviceProvider="Company" portType="MoveIntoTheDoorPT"
        operation="RobotService" input="RobotCommand"/>
... (중략) ...
    <link to="PushTheDoor" from="GrabTheDoorknob"/>
    <link to="Sink" from="PushTheDoor"/>
  </flow>
</CAWL>

```

그림 8 개발자가 작성한 로봇 서비스 시나리오 문서의 일부분

Open” 서비스는 그림 4에서 좌측에 있는 워크플로우의 세 번째 노드를 표현한 것이며, 그림에서 문서 중간에 위치한 “DoorOpenFlow”는 그림 4의 서브워크플로우를 나타낸 것이다. 서브워크플로우는 “DoorOpen” 서비스를 구체적으로 표현한 것이며, 또 다른 하나의 독립적인 워크플로우 단위의 서비스를 나타낸다.

본 실험에서는 실제 사용자 주변 환경으로부터 발생되는 컨텍스트 정보를 얻기 위해 난수 발생기를 활용하였다. 난수 발생기로부터 발생하는 정보의 단위는 constraint를 나타내는 {subject, verb, object}의 엔티티(entity) 집합이며, 집합을 이루는 각각의 엔티티는 실제 센서로부터 들어오는 값들이며, 이들은 구조적 컨텍스트 모델을 거쳐 constraint 단위로 구성된 상황정보로 변환된다. 그리고 이렇게 가공된 상황 정보를 저장해 두었다가 난수 발생기를 통해 {주어, 동사, 목적어}의 집합으로 구성된 임의의 상황 정보를 지속적으로 발생시켜, 실제 상황에서 센서를 통해 발생하는 상황과 동일하도록 구현하였다. 다음 그림 9는 상황 정보에 따라 로봇 서비스를 제공하는 과정을 보여준다.

그림 9에서 첫 번째(1st) 화면은 마이클의 플로우(그림에서 Michael_ServiceFlow)를 실행하기 위한 전이

조건을 판단하기 위해 상황 정보를 비교하는 과정을 보이는 것이고, 붉은 색 점선 박스로 되어 있는 부분은 DoorOpen에 해당하는 서브워크플로우를 호출하고 있음을 나타낸다. 그리고 첫 번째 화면의 붉은 색 박스로 되어 있는 부분에서의 link order는 서브워크플로우에서 여러 서비스 노드들의 실행 순서를 나타낸다. 두 번째(2nd) 화면은 호출된 서브워크플로우의 첫 번째 노드(MoveIntoTheDoor)의 상황 정보를 판단하는 과정이 완료되고 있음을 나타낸다. 이 과정이 완료되고 나면, 웹 서비스로 구현되어 있는 로봇 제어 명령어를 생성하여 URC 서버에 전달한다. 다음 그림은 이와 같은 로봇 명령어를 생성하는 과정을 보여준다.

그림 10은 서브워크플로우에서 첫 번째 노드(MoveIntoTheDoor)의 실행을 위한 상황 정보의 판단 이후, XML 기반의 로봇 제어 명령어를 생성하는 과정을 보여주는 것이다. 그림의 상단에 위치한 서비스 호출 부분([service invocation])의 4번째 항목에 웹 서비스 호출을 위한 입력 값으로 RobotCommand가 전송되었음을 타나내고 있으며, 명령어의 자세한 내용은 다음 붉은 색 박스 부분에 있다. 그 다음은 URC 서버에 명령어가 전송되어, 로봇 서비스가 실행되었다는 결과를 알려주고 있다.

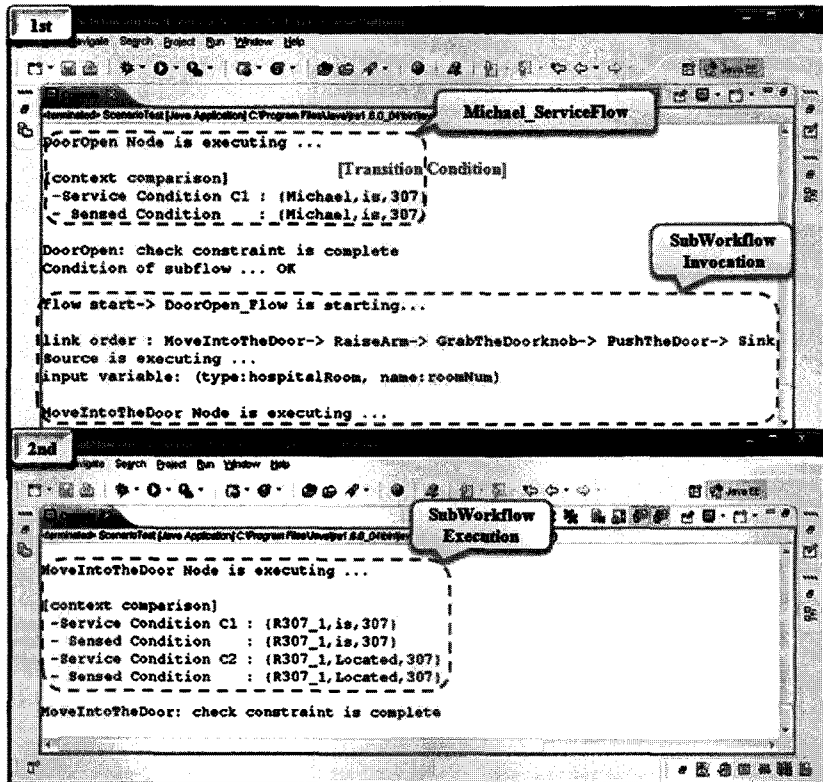


그림 9 로봇 서비스 제공을 위한 상황인지 과정

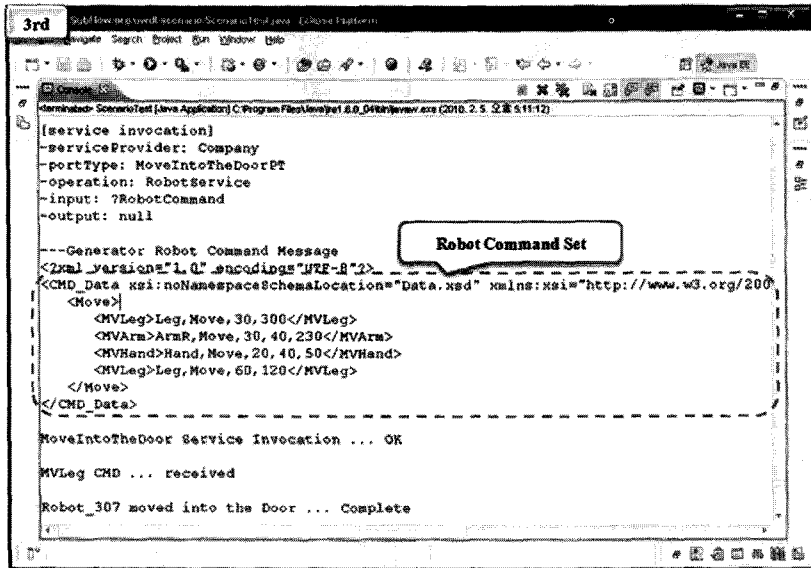


그림 10 로봇 제어 명령어 생성 과정

5. 결론

본 논문에서는 URC 로봇을 통해 사용자의 다양한 상황 정보에 따라 표준 웹 서비스 기반의 상황인지 서비스를 제공하기 위한 워크플로우 시스템을 제안하였다. 또한 제안하는 시스템의 실현 가능성을 입증하기 위해 시뮬레이션 실험을 하였고, 이를 통해 실제 로봇 서비스를 제어하여 로봇 서비스를 제공하는 과정을 보였다. 제안하는 시스템은 크게 두 가지 기능을 갖는다. 하나는 표준 XML 형태의 객체화된 제어 명령어 모델에 작성된 로봇 명령어에 따라 로봇을 제어하는 것이고, 다른 하나는 자원의 접근성이 뛰어난 표준 웹 서비스 기술을 바탕으로 상황인지 기반의 URC 서비스를 제공하는 것이다. 따라서 제안하는 시스템은 다양한 로봇 하드웨어 플랫폼에 대한 이식성을 높일 수 있고, XML을 이용하여 정의한 로봇 명령어에 대한 높은 응용성을 제공할 수 있다.

본 논문에서 제시한 시나리오는 상황인지 기반의 워크플로우를 이용하여 로봇을 제어할 수 있다는 가성을 보이기 위한 것이므로, 문을 여는 수준에서의 간단한 내용을 표현하였다. 따라서 실제 URC 로봇에로의 적용을 위해 사용자에게 의미 있는 시나리오에 대한 연구가 요구되며, 제안하는 시스템의 추가적인 검증 을 위해 의미 있는 시나리오를 바탕으로 실제 가용한 로봇 및 스마트 디바이스를 활용한 실험이 지속적으로 이루어져야 한다.

참고 문헌

[1] 김현, 조영조, 오상복, "URC(Ubiquitous Robotic Com-

panion : 네트워크 기반 서비스 로봇", 정보과학회지 제24권 제3호, pp.5-11, 2006.
 [2] 유원필, 박승환, 채희성, "u-City 로봇시스템기술 동향", 전자통신동향분석, 제24권, 제5호, pp.98-108, 2009.
 [3] 최종선, 조용윤, 최재영, "다중-워크플로우를 지원하는 상황인지 워크플로우 언어의 설계", 한국인터넷정보학회 논문지, 제10권, 제6호, 2009.
 [4] Alexandre Alves, Assaf Arkin, Sid Askary, Curbera, Charlton Barreto, Francisco Curbera, et al, "Web Services Business Process Execution Language Version 2.0," <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>, 2007.
 [5] Sucha S., L. Sea, et al., "A survey on context-aware workflow adaptations," *Proceedings of the 6th International Conference on Advances in MoMM08, ACM*, pp.414-417. 2008.
 [6] Abu Zafar, A. and A. S. Zubair, "A Conceptual Framework for Smart Workflow Management," *Proceedings of the 2009 International Conference on Information Management and Engineering, IEEE Computer Society*, pp.574-578. 2009.
 [7] M Wieland, O. K., D Nicklas, F Leymann, "Towards context-aware workflows," *CAiSE07 Proc. of the Workshops and Doctoral Consortium*, 2007.
 [8] Jun Li, Yingyi Bu, Shaxun Chen, Xianping Tao, Jian Lu, "FollowMe: On Research of Pluggable Infrastructure for Context-Awareness," *AINA06*, vol.1, pp.199-204, 2006.
 [9] Joohyun Han, Yongyun Cho, Eunhoe Kim, Jaeyoung Choi, "A Ubiquitous Workflow Service Framework," *ICCSA06*, pp.30-39, 2006.
 [10] Wieland M. Kaczmarczyk P., Nicklas D., "Context Integration for Smart Workflows," *Proceedings of*

the 6th Annual IEEE International Conference on PerCom08, IEEE Computer Society, pp.239-242, 2008.

- [11] Y. C. Ngeow, D. C., A. K. Mustapha, E. Goh, H. K. Low, "A Context-Aware Architecture for Smart Space Environment," *MUE07, 26-28 April 2007, Seoul, Korea, IEEE Computer Science*, pp.908-913. 2007.
- [12] J Han, Y Cho and J Choi, "Context-aware Workflow Language based on Web Services for Ubiquitous Computing," *LNCS 3481-ICCSA05*, pp.1008-1017, 2005.
- [13] 백문홍, 백승호, 박재한, "정보화시대의 지능형로봇 연구동향", 정보과학회지, 제26권, 제1호, pp.46-52, 2008.
- [14] 홍충성 외 5명, "URC를 위한 상황 정보 관리 기술", 전자통신동향분석 제22권 제2호, pp.10-19, 2007.
- [15] 정연구, 조현규, "지능형 로봇의 국제 표준화 동향", 전자통신동향분석, 제22권, 제2호, pp.70-78, 2007.
- [16] Hyun Kim, Young-Jo Cho, Sang-Rok Oh, "CAMUS: a middleware supporting context-aware services for network-based robots," *Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on*, pp.237-242, 2005.
- [17] Jong-Hwan Kim, In-Bae Jeong, In-Won Park, Kang-Hee Lee, "Multi-Layer Architecture of Ubiquitous Robot System for Integrated Services," *Int J Soc Robot*, pp.19-29, 2009.
- [18] OpenRTM-aist, <http://www.openrtm.org/OpenRTM-aist/html-en/Documents2FRT-Middleware20Overview.html>
- [19] Tetsuo Kotoku, Makoto Mizukawa, "Robot Middleware and its Standardization in OMG," *SICE-ICASE, IEEE*, pp.2028-2031, 2006.
- [20] Ando. N., Suehiro T., Kitagaki K., Kotoku T., Woo-Keun Yoon, "RT-Middleware: Distributed Component Middleware for RT (Robot Technology)," *IEEE/RSJ International Conference on IROS05*, pp.3555-3560, 2005.8.



곽 동 규

2001년 서경대학교 응용수학과(학사). 2004년 숭실대학교 컴퓨터학과(공학석사). 2004년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 프로그래밍 언어, 컴파일러, XML



최 재 영

1984년 서울대학교 제어계측공학과(학사) 1986년 미국 남가주대학교 컴퓨터공학(석사). 1991년 미국 코넬대학교 컴퓨터공학(박사). 1992년~1994년 미국 국립 오크리지연구소 연구원. 1994년~1995년 미국 테네시 주립대학교 연구교수. 2001년~2002년 미국 국립 슈퍼컴퓨팅 응용센터(NCSA) 초빙연구원. 1995년~현재 숭실대학교 정보과학대학 컴퓨터학부 교수. 관심분야는 병렬/분산처리, 고성능컴퓨팅(HPC), 유비쿼터스컴퓨팅



조 용 윤

1995년 인천대학교 전산학과 졸업(학사) 1998년 숭실대학교 대학원 컴퓨터학과 졸업(석사). 2006년 숭실대학교 대학원 컴퓨터학과 졸업(박사). 2009년~현재 국립순천대학교 정보통신공학부 조교수. 관심분야는 시스템 소프트웨어, 임베디드 소프트웨어, 유비쿼터스 컴퓨팅



최 중 선

2000년 숭실대학교 컴퓨터학부(학사). 2002년 숭실대학교 컴퓨터학과(공학석사). 2003년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 시스템 소프트웨어, 유비쿼터스 컴퓨팅, 지능형 로봇 미들웨어