

상황인지 기반 로봇 서비스 개발도구

조용성*, 최종선*, 최재영*, 조용운**

*숭실대학교 컴퓨터학부

**국립순천대학교 정보통신공학부

e-mail : yongseong.cho@ssu.ac.kr, jongsun.choi@ssu.ac.kr, choi@ssu.ac.kr,

yycho@sunchon.ac.kr

A Development Tool for Context-Aware Robot Services

Yongseong Cho*, Jongsun Choi*, Jaeyoung Choi*, Yongyun Cho**

*School of computer science and engineering, Soong sil University

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

요 약

최근 지능형 로봇 서비스의 개발을 위한 여러 형태의 개발도구들이 활발히 연구되고 있다. 지능형 로봇 서비스의 개발도구는 다양한 주변 환경을 인지하여 사용자의 상황에 적합하도록 로봇 서비스를 기술하고, 처리할 수 있어야 한다. 이에 본 논문에서는 유비쿼터스 컴퓨팅 환경에서 사용자의 상황정보를 바탕으로 로봇 서비스를 기술하고, 이를 처리할 수 있는 로봇 서비스 편집 도구를 제안한다. 제안하는 편집 도구는 개발자가 직관적인 GUI를 이용하여 상황인지 워크플로우 기반의 로봇 서비스를 쉽게 작성할 수 있도록 해주며, 워크플로우의 흐름에 따라 로봇을 제어할 수 있는 기능을 포함한 웹 서비스를 이용하여 로봇 서비스를 제공하는 워크플로우 실행 엔진의 연동 기능을 제공한다. 실험에서는 제안한 편집 도구의 효용성을 검증하기 위하여 제안하는 편집 도구를 통한 간단한 로봇 서비스 개발 과정 및 워크플로우 실행 엔진의 연동 과정을 보여준다.

1. 서론

초고속 통신망과 센서 하드웨어의 발전은 유비쿼터스 시대를 불러왔다. 사용자들이 의식하지 않아도 시스템은 사용자의 요구를 인식해 처리해주는 것이 당연한 시대가 된 것이다. 자동화된 유비쿼터스 시스템은 로봇의 운영 방향에도 큰 영향을 미쳤다. 단순 동작 또는 조작의 개념에서 자동화된 서비스를 제공해야만 하는 과제에 직면한 것이다. 하지만 현실적으로 그러한 서비스를 제공해주는 로봇을 개발하기에는 많은 난관들이 존재한다. 인간과 유사한 인공지능을 가진 로봇의 개발은 아직도 먼 훗날의 얘기이고, 특정한 서비스를 탑재한 로봇은 해당 서비스에 종속적이게 되고, 또 다른 서비스를 개발하고 탑재하기 위한 추가적인 비용의 문제 또한 발생한다.

이러한 문제의 해결 방법으로 통신모듈을 탑재한 로봇과 간편한 접근성을 제공하는 웹 서비스의 결합이 부각되었다. 로봇을 제어하는 모듈을 제공하는 웹 서비스를 개발하고, 웹 서비스의 실행과정을 기술하는 워크플로우를 사용함으로써 개발을 간소화할 수 있게 된 것이다. 하지만 단순한 순차처리로서 처리하기엔 실시간으로 급변하는 현실의 문제를 해결하기에 부적합한 면들이 발생하였다.

이러한 문제로 인해 최근 유비쿼터스 컴퓨팅 환경에 적합한 상황인지 워크플로우 모델을 개발하기 위한 다양한 연구가 진행되고 있다. 상황인지 워크플로우 모델이란, 기존의 비즈니스 프로세스를 조정하는 워크플로우 모델에

주변 요소의 영향을 고려할 수 있는 기반을 갖추어 주변 상황을 판단하여 프로세스를 처리할 수 있도록 하는 워크플로우이다. 기존의 워크플로우 모델은 처리해야 하는 프로세스를 직관적으로 표현하여 알아보기 쉬운 장점이 있지만 워크플로우 실행의 전이조건이 프로세스의 입/출력 매개변수를 한정되어 있다는 단점이 있기 때문에 상황인지라는 기능을 추가한 것이다.

하지만 이러한 상황인지 워크플로우 언어를 사용하기 위해서는 상황인지라는 개념과 해당 모델에 대한 포괄적인 이해가 필수적이다. 새로운 개념의 언어를 배우고, 능숙하게 사용하기까지 상당히 많은 시간이 소요되므로, 해당 모델의 사용을 도와줄 수 있는 개발도구의 도입은 필수적이다.

이에 본 논문에서는 웹 서비스와 상황인지 워크플로우 언어 CAWL(Context-Aware Workflow Language)[1]을 바탕으로 유비쿼터스 컴퓨팅 환경에서 로봇 서비스를 손쉽게 그리고 빠르게 개발할 수 있는 개발도구인 CAWL 편집 도구를 제안한다. CAWL 편집 도구는 인터넷 상에 배포된 웹 서비스를 사용하여 로봇이 사용할 수 있는 프로세스를 검색하고, 각각의 웹 서비스들 간의 관계를 GUI의 형태로 표현하여 작성할 수 있도록 한다. GUI로 작성된 워크플로우는 검증된 상황인지 워크플로우 언어인 CAWL을 바탕으로 하여 신뢰성있는 워크플로우 문서를 생성하게 되고, CAWL 편집 도구에 삽입되어있는 CAWL 엔진과 연동하

여 직접 실행 할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 상황인지 워크플로우 모델에 대한 연구에 대해서 살펴본다. 3 장에서는 상황인지 워크플로우 모델을 기반으로 하여 동작하는 CAWL 편집 도구의 구성 모듈에 대해 살펴본다. 4장에서 실험에 관한 내용을 설명한다. 5장에서는 결론을 언급한다.

2. 관련 연구

본 절에서는 유비쿼터스 컴퓨팅 환경에 적합한 로봇 서비스를 개발하기 위한 편집 도구와 관련된 연구 및 개발에 필요한 환경에 대해서 살펴본다.

워크플로우는 작은 단위의 로직을 구성하고 있는 액티비티 또는 다른 워크플로우를 일의 처리순서에 따라 처리하는 기술이다. 현재 사용되고 있는 워크플로우 언어로는 WSBPEL (Web Services Business Process Execution Language), XPD (XML Processing Description Language), BPML (Business Process Modeling Language) 등이 있다. 하지만 본 논문에서 제안하는 CAWL 편집 도구를 통해 로봇서비스를 기술하기 위한 언어로서는 적합하지 않다. 그 이유는 위의 언어들은 워크플로우를 실행하기 위한 전이 조건으로 프로세스의 입출력 매개변수를 사용하지만 그것만으로는 유비쿼터스 환경의 상황정보를 명확하게 표현할 수 없기 때문이다. 그 때문에 위의 언어에 맞춰 설계된 다양한 개발도구(BPEL4WS Design Tool, uFlow scenario editor 등)들이 있지만 유비쿼터스 컴퓨팅 환경에 적용시키기에 부족한 면이 있다.

BPEL4WS Design Tool for Ubiquitous Workflows는 WSBPEL을 기반으로 유비쿼터스 컴퓨팅 환경에 맞춰 워크플로우를 기술하기 위한 개발도구로서 WSDL과 온톨로지 서버를 중점적으로 사용한다. WSDL의 메소드와 오퍼레이션을 온톨로지 서버를 통하여 동적 바인딩하는 방법을 사용한다. 다만 WSDL 문서를 확장하는 방법을 사용하고 있어 WSDL 문서에 대한 의존성이 높다.

uFlow Framework는 uWDL은 서비스의 전이조건으로 상황정보를 사용하여 유비쿼터스 컴퓨팅 환경에 최적화시켰다. uWDL은 다른 워크플로우 언어와는 다르게 언어 자체에 상황정보를 표현할 수 있다. 하지만 uWDL을 사용하여 개발할 수 있는 uFlow Framework 편집 도구는 다양한 워크플로우 패턴을 제공하지 않고 있다.

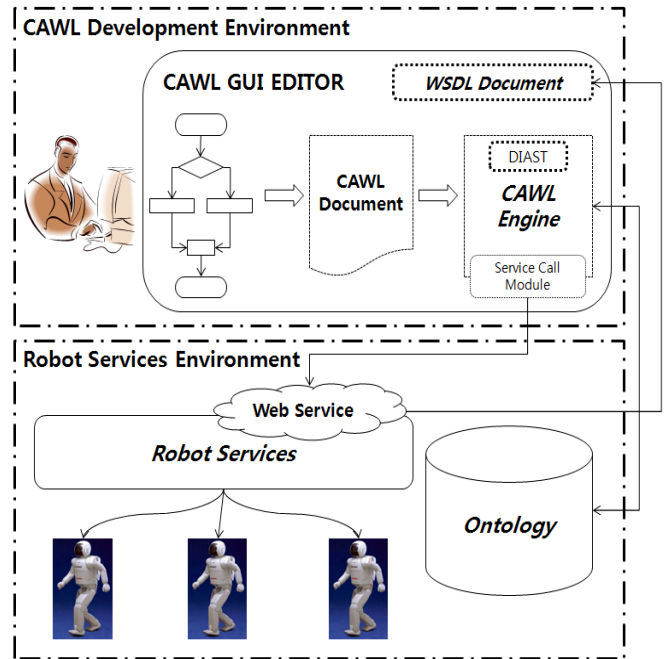
이클립스는 Java의 개발도구로 시작되었지만 현재는 범용적인 Java 플랫폼으로 자리 잡았다. 이클립스 플랫폼을 사용하여 개발된 어플리케이션들은 이기종간의 호환성을 지원하고 Swing을 대체할 수 있는 SWT/JFace를 제공하여 가벼운 GUI 프로그램을 개발할 수 있도록 도와주며, 각 운영체제의 고유의 모습을 표현할 수 있어 산출물의 품질을 향상시켜 준다.

본 논문에서 제시하는 CAWL 편집 도구는 이클립스 플랫폼의 RCP와 GEF 플러그인을 사용하여 그래픽 기반의

편집환경을 구현하도록 하겠다.

3. 제안하는 개발도구

본 절에서는 유비쿼터스 컴퓨팅 환경에서 사용할 로봇 서비스를 개발하기 위한 개발도구의 구성에 대해서 살펴본다.



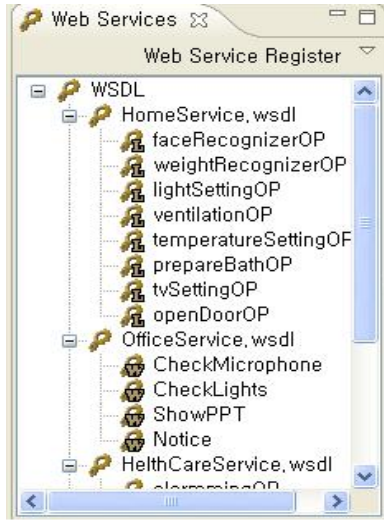
(그림 2) CAWL 기반 편집 도구 구조

그림 2는 CAWL 엔진을 기반으로 하여 만들어진 CAWL 편집 도구를 사용하여 상황인지 워크플로우 문서를 작성하고, 실행하는 과정을 나타낸 것으로 일련의 과정은 다음과 같다.

사용자는 편집 도구를 사용해 웹 상에 배포된 웹 서비스의 WSDL 문서를 파싱하여 확인된 웹 서비스들의 목록을 사용해 순서도 형태의 워크플로우로 작성한다. 작성된 워크플로우는 CAWL 편집 도구의 변환 기능을 사용하여 CAWL 문서로 변환된다. 변환이 완료된 문서는 CAWL 엔진의 워크플로우 파서를 사용해 올바른 CAWL 문서인지를 확인한 후 CAWL 엔진의 워크플로우 엔진이 사용할 객체인 DIAST(Document Instance Abstract Syntax Tree)를 생성한다. 생성된 DIAST는 CAWL 엔진의 워크플로우 엔진으로 전달되어 실행된다. CAWL의 워크플로우 엔진은 온톨로지에 저장된 실시간 상황정보를 바탕으로 DIAST의 루틴을 순차적으로 처리하고, 웹 서비스를 호출하여 로봇을 제어하게 된다.

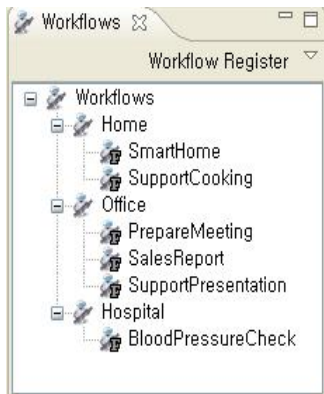
4. 실험

본 절에서는 제안하는 개발도구의 검증을 위해 각 모듈의 모습과 기능에 대해 살펴본다.



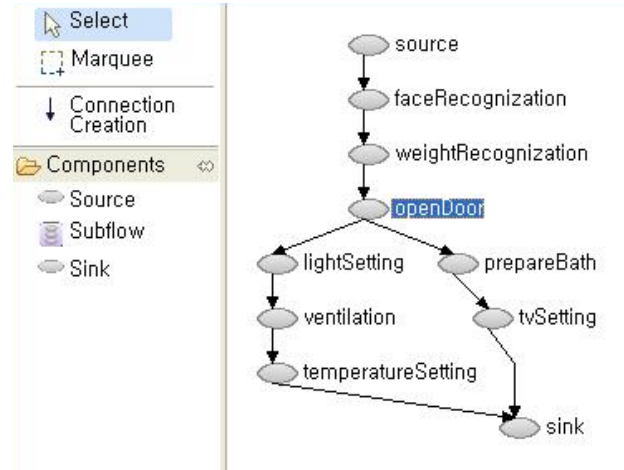
(그림 3) WSDL 문서를 파싱한 목록

그림 3은 CAWL 편집 도구에서 WSDL 문서를 파싱하여 사용자에게 보여주는 화면으로 로봇개발자는 웹 서비스의 WSDL 문서를 배포하여 로봇서비스 개발자가 사용할 수 있도록 할 수 있다. CAWL 편집 도구는 배포된 WSDL 문서에 기술되어 있는 웹 서비스를 읽어 들여 상황인지 워크플로우를 개발할 수 있게 된다.



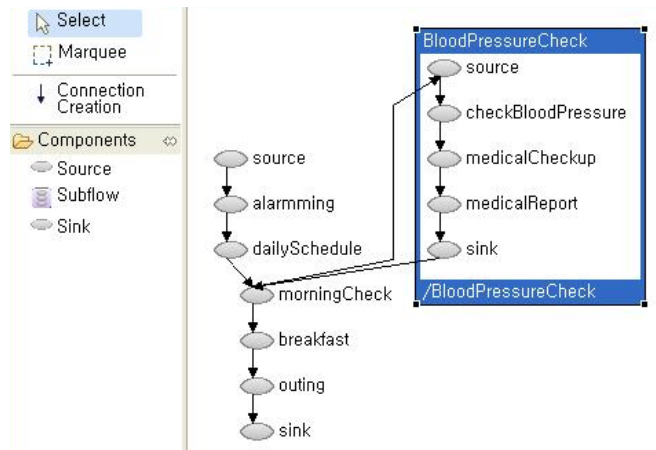
(그림 4) 생성된 워크플로우 목록

그림 4는 개발된 로봇서비스 워크플로우의 목록을 보여준다. 로봇서비스 개발자는 두 가지의 요소를 사용해 로봇서비스 워크플로우를 생성할 수 있는데, 첫 번째는 앞서 설명한 웹 서비스의 항목이고, 두 번째는 그림 3의 워크플로우 항목이다. 반복되는 서비스는 워크플로우로 작성한 후에 재사용이 가능하다.



(그림 5) 상황인지 워크플로우 작성 예시

그림 5는 워크플로우를 작성한 예시이다. 사용자는 로봇서비스의 시작을 나타내는 Source 노드와 종료를 나타내는 Sink 노드 사이에 워크플로우의 흐름을 기술한다. 사용자는 그림 2와 그림 3의 목록을 사용하여 워크플로우를 작성할 수 있다.



(그림 6) 로봇서비스가 재사용 되는 예시

그림 6은 로봇서비스의 재사용을 나타내는 그림이다. 사용자가 정의한 BloodPressureCheck 로봇서비스를 다른 로봇서비스 워크플로우 안에 삽입하여 사용하고 있는 예시이다. 삽입되는 로봇서비스는 노드로서 취급될 수도 있고, 결과 값을 돌려주는 메소드의 형태를 가질 수도 있다

그림 7은 사용자가 작성한 워크플로우를 바탕으로 편집도구가 산출해내는 CAWL 문서의 일부분을 발췌한 것이다. CAWL 문서는 CAWL 엔진의 워크플로우 파서에 의해 파싱된 후 DIAST(Document Instance Abstract Syntax Tree)를 생성하게 되고, CAWL 엔진의 워크플로우 엔진에 의해 온톨로지 정보와 결합되어 알맞은 로봇서비스를 실행하게 된다.

```

<uflow:flow ...>
  <uflow:documentation> ... </uflow:documentation>
  <uflow:variable ...>
    <uflow:initialize ...>
      <uflow:from .../>
    </uflow:initialize>
  </uflow:variable>

  <uflow:source>
    <uflow:documentation> ... </uflow:documentation>
    <uflow:input ... />
  </uflow:source>

  <uflow:node ...>
    <uflow:documentation> ... </uflow:documentation>
    <uflow:condition ... >
      <uflow:context>
        <uflow:rule ... >
          <uflow:constraint ... >
            <uflow:subject ... > ... </uflow:subject>
            <uflow:verb ... > ... </uflow:verb>
            <uflow:object ... > ... </uflow:object>
          </uflow:constraint>
        </uflow:rule>
      </uflow:context>
    </uflow:condition>
    <uflow:invoke ... />
  </uflow:node>

  <uflow:link ... />

  <uflow:sink>
    <uflow:output ... />
    <uflow:synchronize/>
  </uflow:sink>
</uflow:flow>

```

(그림 7) CAWL 로 작성된 시나리오 문서의 형식

[7] Junwei Cao*, Stephen A. Jarvis, Subhash Saini and Graham R. Nudd, "GridFlow: Workflow Management for Grid Computing", Department of Computer Science, University of Warwick, 2003 - ieeexplore.ieee.org

5. 결론

본 논문에서는 유비쿼터스 컴퓨팅 환경에 적합한 로봇 서비스를 개발하기 위한 개발도구인 CAWL 편집 도구를 소개하였다. CAWL 편집 도구는 CAWL과 RDF 기반의 온톨로지, 웹서비스를 바탕으로 설계되어 직관적인 GUI 형태로 로봇서비스를 개발할 수 있다. 사용자는 직관적인 UI로 워크플로우를 작성할 수 있고, 작성된 워크플로우는 CAWL 문법에 맞춰 문서로 생성되며, 편집 도구의 CAWL 엔진은 주변상황을 인지하고 알맞은 로봇 서비스를 실행하여 자동화된 개발환경을 제공한다.

참고문헌

- [1] 최종선, 조용윤, 최재영, "다중-워크플로우를 지원하는 상황인지 워크플로우 언어의 설계" 인터넷정보학회 논문지 제10권 제6호, 2009 년 12 월
- [2] OASIS, WSBPEL v2.0, <http://www.oasis-open.org/committees/wsbpel/>
- [3] Workflow Management Coalition, XPD, <http://www.wfmc.org/>
- [4] OMG, BPMI, <http://www.bpmi.org/>
- [5] W3C, WDSL, <http://www.w3.org/TR/wsd/>
- [6] Howard Foster, Sebastian Uchitel, Jeff Magee, Jeff Kramer, "LTSA-BPEL4WS: Tool Support for Model-based Verification of Web Service Compositions", Department of Computing, Imperial College London, <http://www.doc.ic.ac.uk/ltsa/bpel4ws>, 2007 - en.scientificcommons.org