

OSGi 프레임워크 기반의 상황인지 서비스 모델링

박준상^o 박영택
 송실대학교 컴퓨터학과

theartist@ailab.ssu.ac.kr^o, park@computing.ssu.ac.kr

OSGi-based Context Reasoning Service Modeling

Junsang Park^o Youngtack Park
 Dept. of Computer Science, Soongsil University

요약

유비쿼터스 컴퓨팅 시대의 도래와 함께 홈네트워크상의 장비들의 연결과 제어를 통해 다양한 서비스를 제공하는 기술이 차세대 정보기술 산업을 주도할 태풍의 핵으로 부상하고 있다. 본 논문에서는 OSGi 표준을 기반으로 한 지능형 에이전트를 구성하며, 가정 내의 가전기기, 통신기기를 제어하고 상황에 적합한 서비스를 추론하여 제공해주는 기술을 제안한다. 헬스케어(health-care) 서비스, 베이비케어(baby-care) 서비스, 사용자 편의를 위한 개인화 서비스, 방범(security) 서비스등이 상황 추론을 통해 제공되는 서비스의 대상이 된다. OSGi 플랫폼상에서 지능형 에이전트는 각 서비스에 의존적으로 구현되며, OSGi 번들 형태로 존재하여 프레임워크나 다른 에이전트의 변화없이 쉽고 간편하게 OSGi상에 등록(register)하고 취소(unregister)하는 것이 가능하다. 이러한 에이전트에 규칙기반의 상황 추론이 가능한 엔진을 장착하여, 유비쿼터스 컴퓨팅 환경의 서비스 컴포넌트들의 역할을 돕는 지능형 시스템을 연구하였다.

1. 서론

유비쿼터스 컴퓨팅 환경에서는 누구나 기기, 시간, 장소에 구애 받지 않고 상황에 적합한 개인화 서비스를 제공받는 것이 가능하다. 사용자의 건강상태를 생체신호 센서를 통해 실시간으로 확인하여 적절한 조치를 취하는 헬스케어 서비스, 가정 내의 보호자가 필요한 아기들의 모니터링을 위한 베이비케어 서비스, 개인의 취향과 생활패턴을 고려하여 상황에 맞는 자동화된 서비스를 제공하는 개인화 서비스, 그리고 가정 내의 도둑이나 불법 침입자로부터 보호하는 방범 서비스등이 스마트 공간(smart space)을 기반으로 하여 여러 연구기관과 대학에서 활발한 연구가 진행 중이다. 그리고 가정 내의 가전, 텔레콤, 컴퓨팅등의 기기를 통일된 프로토콜을 사용하여 제어하고 관리하기 위해 홈 네트워크 표준화 작업이 진행중인 가운데 OSGi가 여러 글로벌 기업의 참여와 지원에 힘입어 선두주자의 입지를 굳히고 있다.

본 논문에서는 OSGi 스펙(specification)을 이용한 프레임워크를 구축하여 유비쿼터스 센서로부터 각종 데이터를 입력받고, 사용자에게 제공될 각종 서비스를 동작(invocation)시키는 시스템을 제안한다. 환경에 맞는 컨텍스트 모델링, 추론에 효율적인 형식으로의 변환, 컨텍스트의 추론을 진행하는 엔진등의 기능은 OSGi 프레임워크[1]에 각각 번들형태로 구현되어 동작하게 된다. 번들은 JAVA언어 어플리케이션 기능을 하며 다른 번들들과는 독립적인 성질로 구현된다.

2. 지능형 에이전트 시스템 아키텍처

그림1은 센서를 이용한 컨텍스트 수집부터 서비스 트리

거빙(triggering)되는 전체적인 흐름을 표현한 구조도[2]이다. 유비쿼터스 컴퓨팅 환경에 존재하는 각종 환경 센서, 위치 센서, 생체신호 센서등의 장치로부터 컨텍스트를 위한 소스 데이터를 얻을 수 있다. 이러한 데이터는 컨텍스트 생성기(context provider)에 의해 일정한 형태로 구성된다. 해당 도메인 온톨로지의 참조를 통해 시맨틱 웹 언어인 OWL 형태의 컨텍스트 모델을 구성하며, 컨텍스트 추론 엔진에 적절한 입력 형태로 변환된다. OWL로 모델링함으로써 사용자 프로파일 및 실시간 센싱 데이터를 쉽게 수정하고 공유할 수 있으며, 추론에 효율적으로 사용될 수 있다. 각 서비스를 위한 규칙들은 지식베이스(knowledge base)에 저장되며, 컨텍스트 모델

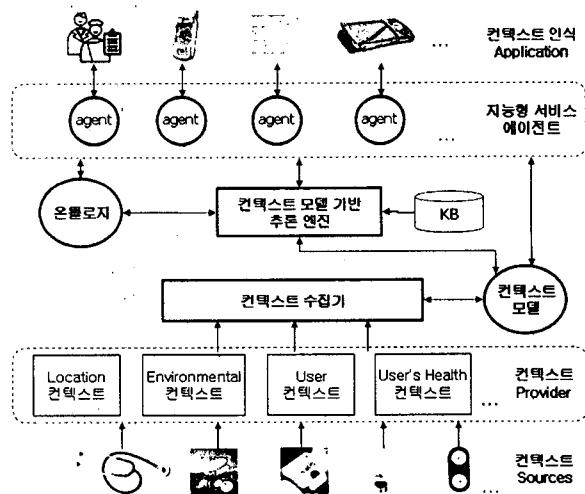


그림 1 OSGi를 기반으로 컨텍스트 수집, 컨텍스트 모델링, 컨텍스트 추론을 통한 서비스를 제공하는 지능형 에이전트 시스템 구조

이 규칙에 적용됨으로써 장치마다 적절한 서비스 제공을 명령을 하는 역할을 한다. 새로운 서비스를 추가하기 위해서는 전체 플랫폼에 영향을 미치지 않고, 서비스 규칙을 동반한 번들형태의 에이전트만 구현하여 등록하면 된다.

3. OSGi 프레임워크 구성

OSGi[3]는 JAVA지향의 서비스 플랫폼이다. JAVA VM을 사용하므로 어느 형태의 운영 체제 위어나 탑재가 가능하고, 관리와 개발의 편리성이 뛰어나다는 장점이 있다.

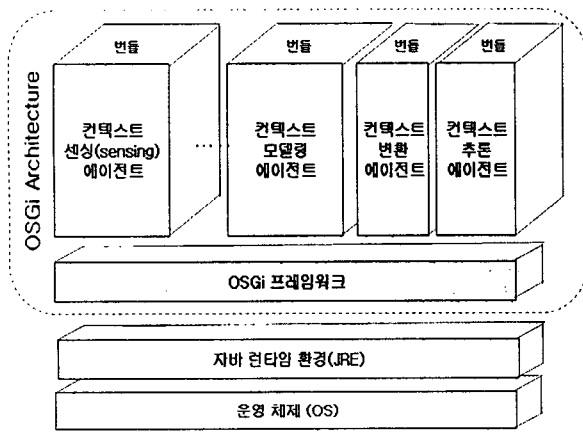


그림 2 OSGi 프레임워크상에 번들화된 에이전트 모듈

그림 2는 OSGi 프레임워크와 프레임워크를 구성하는 주요 번들을 나타내고 있다. 지능형 에이전트를 번들형태로 구현함으로써 에이전트의 라이프 사이클(life-cycle) 관리가 수월하다. 컨텍스트 센싱 에이전트는 유비쿼터스 센서 노드로부터 전달되는 센싱 데이터를 저장하고 관리하는 역할을 한다. 센싱 데이터의 특징에 따라 분류하고 컨텍스트 모델링 에이전트에게 전달하는 기능도 하고 있다. 각종 컨텍스트는 모델링, 변환, 추론의 과정을 통해 서비스 제공이라는 마지막 단계까지 진행된다.

3.1 컨텍스트 모델링 에이전트

유비쿼터스 컴퓨팅 환경을 구성하는 물리적인 요소들을 센서를 통해 감지하고, OWL이라는 시멘틱 웹 언어를 사용하여 컨텍스트 모델을 구성한다. OWL은 개체의 특징을 나타내는 프로퍼티(property)를 정의할 수 있으며, 개체와 개체간의 관계(relation)를 표현할 수 있다. 그러므로 센싱되는 데이터들의 복잡한 특징과 관계를 계층구조로 명확하게 표현할 수 있으며, 계층에 의해 추론이 가능해 지므로 정의하지 않은 데이터마저 추출해 낼 수 있다. 그리고 어느 OSGi 플랫폼 상에서든지 OWL로 표현된 컨텍스트 모델에 접근할 수 있어서 사용자는 가정 내뿐만 아니라 사무실, 학교, 공공기관등에서도 같은 서비스를 받을 수 있다.

그림 3은 감지한 센싱 데이터를 OWL 형태로 표현한 예를 보이고 있다. 사용자에게 대한 신상정보인 이름, 나이, 키, 몸무게, 전화번호 같은 값들과 그의 생체정보인 혈압치 같은 값들, 소유하고 있는 장치들을 클래스와 프로퍼티의 구조로 표현하고 있다.

```
<Person rdf:ID="person00">
  <hasWeight rdf:datatype="#int">45</hasWeight>
  <ownsDevice>
    <PDA rdf:ID="pda01">
      <exerMode rdf:datatype="#boolean">true</exerMode>
    </PDA>
  </ownsDevice>
  <hasPhoneNum rdf:datatype="#string">010-777-7777</hasPhoneNum>
  <healthValue>
    <BloodPressure rdf:ID="hongBP">
      <ownsBy rdf:resource="#person00"/>
      <currentMaxValue rdf:datatype="#int">73</currentMaxValue>
      <normalMaxValue rdf:datatype="#int">120</normalMaxValue>
      <normalMinValue rdf:datatype="#int">80</normalMinValue>
    </BloodPressure>
  </healthValue>
  <hasHeight rdf:datatype="#int">160</hasHeight>
  <hasAge rdf:datatype="#int">23</hasAge>
  <hasName rdf:datatype="#string">hong</hasName>
</Person>
```

그림 3 OWL로 표현된 컨텍스트 모델의 예

3.2 컨텍스트 변환 에이전트

컨텍스트 변환 에이전트는 컨텍스트 추론 엔진에 적합한 형태로의 변환을 수행하는 역할을 한다. 규칙 기반의 추론은 술어(predicate)의 구조가 효율적이므로, OWL 인스턴스를 트리플(triple)형태로 변환한다. 트리플 형태는 (<predicate> <subject> <object>)로 구성되어, 개체와 프로퍼티의 관계를 명확히 표현할 수 있다.

3.3 컨텍스트 추론(context reasoning) 에이전트

OSGi의 여러 번들 가운데 가장 핵심이 되는 것이 컨텍스트 추론 에이전트[4]이다. 컨텍스트 추론 에이전트는 제공하는 서비스에 비례하여 다수가 존재하게 되며, 각자 독립적으로 자기 역할을 수행하므로 사용자의 구미에 맞게 얼마든지 수정이 가능하다. 추론 엔진은 JAVA 기반의 JESS라는 선언적 언어를 사용하여 구동하였다. JESS는 1초에 수만 개의 규칙을 처리할 수 있어서 서비스 엔진을 구성하기에 적합하다. 동시에 수십만 개의 트리플이 작업 메모리(working memory)상에 공존 가능하므로, 유비쿼터스 컴퓨팅 환경의 센싱 데이터를 처리하는데 무리가 없다.

그림 4는 컨텍스트 추론의 과정을 보이고 있다. JESS는 Rete 알고리즘을 사용한 engine으로 전방향(forward chaining) 추론을 수행한다. 변환된 JESS Fact형태의 트리플이 작업 메모리에 실시간으로 삽입(assert)된다. 서비스마다 정의된 서비스 규칙에 JESS Fact가 적용됨으로써 적절한 서비스에 트리거링이 수행된다.

그림 5는 헬스케어 도메인에 대한 서비스 규칙의 예이다. 혈당과 혈압의 관리가 필요한 환자의 경우, 환자의 혈당치에 대한 임계값, 혈압치에 대한 임계값을 OWL형태의 프로파일(profile)로 미리 정의해 둔다. 비슷한 경우

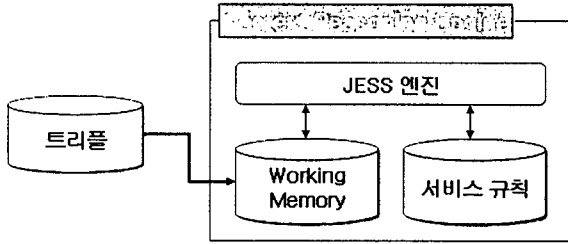


그림 4 컨텍스트 추론 과정으로, 응급상황에 대한 규칙을 정의함으로써 자동화된 응급서비스를 적용하여 대처하는 서비스를 제공할 수 있게 된다.

건강 및 질병 관리	1.혈당관리대상 IF 환자의 혈당치가 정상이지 않다 THEN 주치의에게 알린다 2.혈압관리대상 IF 환자의 혈압이 정상이지 않다 THEN 주치의에게 알린다
응급상황	IF 환자의 체온이 정상이지 않다 AND 환자의 심장박동수가 정상이지 않다 THEN 병원 응급실로 알린다

그림 5 헬스케어 서비스에 관한 규칙 예

4. 실현

실험은 OSGi 오픈소스인 OSCAR[5]를 이용하여 진행을 하였다. 그림 6은 OSCAR의 GUI환경으로, 센서 번들, 서비스 에이전트 번들등을 등록하는 화면을 보이고 있다.

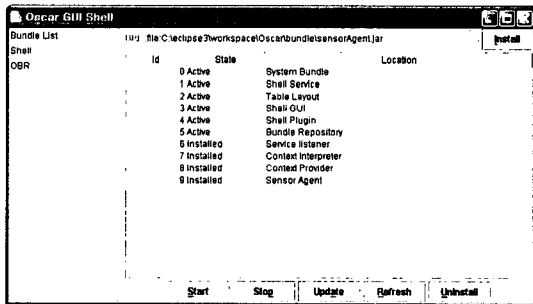


그림 6 OSGi 번들의 등록(register)과 취소(unregister), 시작(start)과 멈춤(stop)을 수행하는 OSCAR 환경이다.

모든 번들의 등록이 끝나고 시작을 실행하면, 그림7과 같이 서비스 트리거링을 위한 과정이 진행된다. 헬스케어 서비스를 위한 센싱 데이터가 입력되고, 헬스케어 서비스에 적절한 헬스케어 OWL 인스턴스가 생성된다. 이는 다시 JENA기반 하에 JESS Fact에 적용할 트리플로 변환되어지고 JESS 작업메모리에 쌓이게 된다. 컨텍스트 추론엔진은 사용자의 생체 정보를 분석하여, 건강상태를 알아내고 적절한 조처를 취하기 위해 서비스 제공을 명령하게 된다.

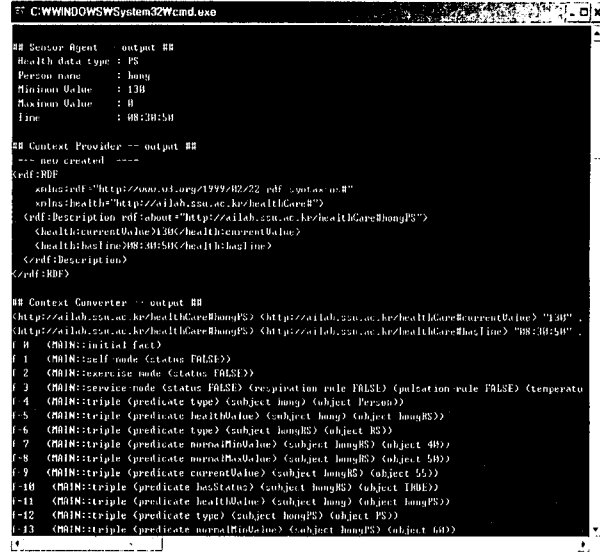


그림 7 센싱 데이터, OWL 인스턴스, 트리플 변환, JESS 작업 메모리로의 삽입의 과정을 보이고 있다.

5. 결론 및 향후 연구

본 논문에서는 OSGi를 기반으로 하여 필요한 서비스에 맞는 컨텍스트의 추론을 돕는 컨텍스트 모델링을 기법을 기술하였다. 결론적으로, OSGi 번들 형태의 에이전트를 구성하는 방법을 제시하여 서비스의 특징에 관계없이 휴대워크상에 하나의 플랫폼을 구성할 수 있게 되었다. 원하는 서비스를 추가하고자 할 경우, 장치를 그 서비스 번들과 함께 등록하기만 하면 원활한 서비스를 제공할 수 있어 서비스 관리 측면에서 매우 효율적이다. 향후에는 거의 무한적으로 입력되는 센싱 데이터의 효율적인 관리 방안, 독립적으로 동작하는 서비스 번들간의 통합, 서비스 트리거링에 대한 연구를 진행할 것이다.

6. 참고 문헌

[1] Valtchey, D, Frankov, I, "Service gateway architecture for a smart home", Communications Magazine, IEEE, Vol.40, Issue.4, April 2002
 [2] Tao Gu, H.K. Pung, D.Q. Zhang, "Toward an OSGi-based Infrastructure for Context-Aware Applications", Published by the IEEE CS and IEEE ComSoc, 2004
 [3] "Open Service Gateway Initiative", <http://www.osgi.org>
 [4] Schilit, B., Adams, N., Want, R., "Context-Aware Computing Applications.", Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, p. 85-90, 1994.12
 [5] Oscar 홈페이지, <http://oscar-osgi.sourceforge.net>