

상황인지 IoT 서비스 제공을 위한 온톨로지 기반 사용자 친화적 서비스 환경

Ontology Based User-centric Service Environment for Context Aware IoT Services

최환석, 이준영, 양나리, 이우섭
국립한밭대학교 멀티미디어공학과

Hoan-Suk Choi(hkrock7904@gmail.com), Jun-Young Lee(wnsdud511@gmail.com),
Na-Ri Yang(nari.yang1104@gmail.com), Woo-Seop Rhee(wsrhee@hanbat.ac.kr)

요약

상황인지 서비스 제공을 위해서는 센서를 이용하여 사용자의 상황, 정황, 상태 정보를 추적하여야 한다. 하지만 센서에서 수집된 데이터는 시스템, 문법, 구조, 의미적인 측면에서 이질성을 가지고 있어 다양한 서비스에 재사용이 어렵다. 이를 위해 센서 데이터는 이와 같은 이질성을 제거하고 상황인지 서비스 제공에 용이한 컨텍스트 정보 형태로 처리 되어야 한다. 또한 기존의 상황인지형 서비스 정의 환경은 상황을 정의하기 위해 필요한 센서의 종류나 상황정보의 기준, 서비스 정의를 위한 프로그래밍 능력 등을 필요로 하여 일반적인 사용자의 이용이 어려운 실정이다. 따라서 본 논문에서는 IoT 패러다임을 통해 습득된 데이터를 이용하여 상황을 인지하고 사용자가 원하는 서비스를 제공하는 상황인지 IoT 서비스의 제공을 위한 온톨로지 기반 사용자 친화적 서비스 환경을 제안한다. 제안하는 환경은 센서 데이터를 컨텍스트 정보로 처리하는 온톨로지 기반 시멘틱 센서 데이터 처리 메커니즘과 사용자 친화적 서비스 정의 환경으로 구성된다.

■ 중심어 : | 상황인지서비스 | 사물인터넷 | 시멘틱 | 센서데이터 처리 | 서비스 매시업 |

Abstract

To provide context aware service, certain phenomena and situation information of the user should be detecting and monitoring using sensors. But, the sensor data has heterogeneity about system, syntax, structure, and semantics. So, it is difficult to apply to various services. To solve this issue, we should remove these heterogeneities and convert to context information that is easy to provide context aware services. Also, the existing context aware service mash-up environment require relevant sensor type, criteria of context, programing ability to define the service situation. So, it is difficult to create service for a non-technical user. Therefore, we propose ontology based user-centric service environment to provide context aware IoT services. It provide context awareness and user-desired services using data that acquired via IoT paradigm. The proposed environment consist of the ontology based semantic sensor data processing and the user-centric service environment. It provide convert the object data to the context information and service mash-up process to create user service in easy way.

■ keyword : | Context Aware Service | Internet of Things | Semantic | Sensor Data Processing | Service Mash-up |

* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(과제번호: 2013025720)

접수일자 : 2014년 03월 24일
수정일자 : 2014년 05월 08일

심사완료일 : 2014년 05월 08일
교신저자 : 이우섭, e-mail : wrhee@hanbat.ac.kr

I. 서론

유비쿼터스, 퍼제형 컴퓨팅 (pervasive computing) 등의 상황인지 환경을 실현하고 사용자 주변 환경의 정황을 인지하기 위해서는 다양한 센서가 광범위하게 배치되어야만 한다. 일반적으로 센서 네트워크는 이와 같은 환경 감지에 효과적인 해결책으로 사용되고 있다. 하지만 기존의 센서 네트워크는 서비스 공급자, 센서와 장비 제조사 및 프로토콜에 의존적인 구조를 가지고 있어 재정적, 기술적인 문제가 존재한다[1]. 이러한 이질성은 크게 시스템, 문법, 구조, 의미적 측면의 4가지로 구분하고 있으며[2] 이기종 환경에서 센서데이터의 재사용이 어렵다[3]. 이를 해결하기 위해 웹 프로토콜을 이용하는 패러다임이 대두되고 있다. IoT (Internet of Things), WoT (Web of Things) 와 같은 새로운 패러다임은 인터넷 프로토콜 기반의 웹 기술을 이용하여 각각의 센서나 장치의 제어, 관리, 모니터링을 제공한다. 또한 웹 프로토콜 표준인 HTTP (Hyper Text Transfer Protocol) 를 이용하여 센서 및 장치를 웹 환경의 자원으로 인지하고 처리할 수 있어 센서 및 장비 제조사, 서비스 제공자, 이기종의 네트워크 구조 등에 독립적인 제어가 가능하다[4]. 이와 같은 패러다임을 통해 시스템, 문법, 구조적 측면의 이질성을 제거할 수 있으며 이는 비용적인 측면에서도 큰 장점이라고 할 수 있다[5]. 또한 사용자의 다양한 욕구를 충족시키기 위한 맞춤형 서비스 혹은 자신의 상황 및 주변 정황 정보를 인지하는 상황인지 기반의 서비스에 관한 요구가 증가하고 있다[6]. 이를 위해 온톨로지 기술을 이용하여 수집된 센서 정보의 의미적 이질성을 제거하고 상황을 인지하는 SSW (Semantic Sensor Web) 기술에 관한 연구가 진행되고 있다[7].

또한 스마트 기기의 보급이 증가하면서 기존의 획일화 된 휴대전화 환경에서 자신만의 환경을 구성하여 사용하는 스마트 환경으로 발전 하였다. 이는 기존의 휴대전화 뿐 만 아니라 스마트 TV를 필두로 가정, 빌딩, 사무실 등의 여러 환경의 다양한 장치로 확산되고 있는 실정이다. 따라서 상황 인지형 서비스의 경우에도 일반 사용자가 자신만의 맞춤형 서비스를 직접 생성하고자

하는 요구사항이 증가할 것이다[8]. 하지만 기존의 온톨로지를 이용한 상황인지 서비스 제공 기술[9]은 대체적으로 특정 활용 영역의 어휘를 상세하게 작성하고 상황의 정의를 위해 다양한 정보를 입력하고 서비스 조건을 정의한 후 센서에서 생성된 데이터를 도입하여 추론을 하는 방식을 따르고 있다. 따라서 이를 기반으로 하는 서비스 매시업 및 컴포지션 기술은 전문적인 지식이 없는 일반 사용자가 서비스 요구사항, 조건, 필요한 정보 등을 정의하고 이를 생성 및 구현하기에 매우 어려운 실정이다[10]. 왜냐하면 사용자가 서비스를 정의하기 위해서는 서비스를 제공하는 플랫폼의 전체적인 구조를 이해할 필요가 있으며 서비스의 조건이 되는 컨텍스트를 정의하기 위해 필요로 하는 관련 센서들의 종류와 각 측정값의 조건 및 기준들을 알아야하기 때문이다. 결과적으로 쉽게 서비스를 정의할 수 있는 환경이 제공된다면 사용자가 원하는 환경에서 원하는 서비스를 쉽게 제공 받을 수 있으며 플랫폼 입장에서보다 보다 정확한 서비스 정의 정보를 통해 사용자의 의도를 정확히 파악할 수 있을 것으로 생각된다.

따라서 본 논문에서는 IoT 패러다임으로 연결된 센서나 장비를 이용하여 상황을 인지하고 이러한 정보를 기반으로 사용자가 원하는 서비스를 제공하는 것을 상황인지 IoT 서비스라고 정의하고 이러한 서비스의 제공을 위한 온톨로지 기반의 사용자 친화적 서비스 환경을 제안한다. 제안하는 센서 데이터 처리 메커니즘은 오브젝트 및 서비스 추상화 (Object & Service abstraction), 시맨틱 어노테이션 (Semantic annotation), 추론 (Reasoning), 서비스 실행 (Service execution) 의 4가지 단계를 통해 센서 데이터를 상황 정보로 변환한다. 또한 제안하는 사용자 친화적 서비스 정의 환경은 장소 설정 (Place setting), 오브젝트 설정 (Object setting), 오브젝트 및 기능 선택 (Object & function selection), 상황 설정 (Condition setting) 의 4 단계로 구성되어 있으며 웹 기반의 서비스 정의 UI를 제공하여 프로그래밍 능력이 없는 일반적인 사용자가 서비스를 쉽게 정의 하도록 한다.

이를 위해, 본 논문에서는 2장에서 온톨로지 기반 리소스 및 상황 모델링 기술, 서비스 매시업 기술, 사용자

친화적 서비스 정의 기술에 관한 관련 연구를 분석하고 3장에서는 센서 데이터를 상황 정보로 변환하는 온톨로지 기반 시멘틱 센서 데이터 처리 기술을 제안한다. 4장에서는 일반적인 사용자의 쉬운 서비스 정의를 위한 사용자 중심 서비스 정의 과정 및 웹 기반 UI를 제안한다. 5장에서는 기존의 연구와 제안하는 방식의 비교 분석을 수행하고 6장에서 결론을 맺는다.

II. 관련연구

1. 온톨로지 기반 리소스 및 상황 모델링 기술

상황인지 서비스를 제공할 수 있는 스마트 환경을 실현하기 위해서는 모든 정보를 기계가 인식 가능한 형태로 제공해야 한다. 이를 위해서는 센서에서 생성된 원시 데이터를 컨텍스트 정보로 변환하는 메커니즘과 변환된 정보를 웹상의 리소스로 표현하는 표현기술이 요구된다. 이러한 컨텍스트 정보를 표현하는 표현 기술로는 표현력의 정도에 따라 대표적으로 RDF (Resource Description Framework), RDFs (RDF Schema), OWL (Web Ontology Language)을 사용한다. 상황인지 서비스 제공을 위해서는 이와 같이 표현된 정보를 이용하여 상황 모델링을 제공해야 한다. 상황은 어떤 한 개체의 상태를 특정화 시킬 수 있도록 사용되는 정보이다. 이러한 상황을 인간의 사고를 통해 인식하는 것과 같이 상황 정보에 대한 높은 수준의 추정적 개념을 제공하기 위해 상황 모델링이 요구된다[11]. 상황인지 서비스를 위한 온톨로지 기반의 상황 모델링 기법을 적용한 연구들이 진행되고 있다. [9]는 사용자 지역의 기상 및 보건 정보를 입력하여 u헬스 서비스에 필요한 상황 정보를 온톨로지 기반으로 도출하고 그에 맞는 건강 지수, 운동 및 식단 추천 서비스를 제공한다. 이는 온톨로지를 건강정보와 같은 특정 영역의 상황을 상세하게 정의하여 서비스를 제공하므로 이를 위해서는 전문적 지식 및 기술적 수준을 요구하며 범용적 서비스를 제공하기 위한 상황 정의로 확장하기에는 구조적 한계가 존재한다.

2. 서비스 매시업 기술

다양한 서비스의 제공을 위해 리소스 및 서비스 정의 기술에 의해 표현된 정보들을 이용하여 서비스를 매시업 하고 제공하는 연구가 활발히 진행되고 있다. E. Avilés-López[10]은 웹 기술기반의 IoT 서비스를 위한 사용자 상호작용 모델을 지원하는 프레임워크를 제안하였다. 이 프레임 워크는 기능적 측면에서 각 서비스 명세를 기술하는 방법, 사용자 환경에서 사용할 수 있는 서비스를 검색하는 방법, 관심 있는 이벤트가 발생할 때 동료와 통신 하는 방법, 서비스에서 데이터와 기능을 결합하는 방법을 제시하고 있다. 또한 Application Manifests라 정의한 컴포지션 언어를 제안한다. 이는 컴포지션에 참여하는 서비스, 서비스 간의 흐름, 매시업 된 결과물, 실행일정, 인증자격증명 및 기타 메타데이터와 같은 응용프로그램 제어 데이터를 포함하고 있다. S. Han[12]는 서비스라고 정의한 표준화된 형태의 독립적인 컴포넌트로 BAS (Building Automation System)의 기능을 분리하여 이를 SOA (Service Oriented Architecture)기반의 분산형 서비스로 제공하는 방법을 제안한다. 또한 미리 정의된 규칙을 기반으로 6단계 컴포지션 처리과정을 통해 해당 장치 및 서비스를 매시업, 모델링하고 처리하여 사용자에게 적합한 서비스를 제공한다. 또한 서비스 캐시를 이용하여 큰 규모의 환경에서 효과적임을 보였으나 액츄에이터의 기능 한 가지를 서비스로 정의하고 규칙에 맞게 여러 서비스를 실행하는 구조로 되어 있어 복잡한 상황 및 다양한 서비스를 매시업할 경우 서비스 레시피를 정의하는데 복잡도가 상승할 것으로 예상된다. 또한 온톨로지를 이용하여 상황을 인지하였으나 대부분 컴포지션 계획에 따라 진행되는 구조로써 보다 적극적으로 상황정보를 이용할 수 있는 구조가 요구된다. C. H. Quyet[13]은 다양한 장치, 사양, 프로토콜을 통합하고 효과적으로 이용하기 위한 클라우드 기반의 스마트 홈 구조를 제안하고 있으며 DPWS (Device Profile for Web Services) 기반의 홈 네트워크 환경에 온톨로지 기술을 적용하여 스마트 홈 서비스를 제공한다. 사용자가 수동적으로 장치를 제어하는 방법, 미리 정의된 규칙을 기반으로 자동적으로 제어하는 방법, 사용자의 행동의 패턴을 조사하여 규칙

에 추가하는 방법으로 구성되어 있다. 이 연구의 가장 큰 특징은 사용자의 패턴을 분석하여 서비스 규칙에 적용시키는 것이다. 이는 사용자의 행위 패턴을 조사하여 특정 행위 자체를 규칙에 저장하게 된다. 하지만 사용자의 모든 행위를 기록, 분석하는 것은 많은 처리 비용을 야기할 것으로 예상된다.

3. 사용자 친화적 서비스 정의 환경

사용자 주변의 상황을 인지하고 그 상황에 맞는 서비스를 동적으로 제공하는 것은 매우 복잡하며 고려해야 할 사항들이 많다. 따라서 사용자에게 원하는 서비스를 손쉽게 정의할 수 있는 환경을 제공하려는 연구가 활발히 진행되고 있다. UbiSOA[10]은 스마트 환경과 사용자간의 상호작용을 할 수 있도록 만든 메시업 에디터로써 사용자는 사용가능한 서비스 혹은 이전에 정의된 서비스를 통해 새로운 서비스를 정의할 수 있다. UbiSOA 에디터는 웹 기반 환경에서 끌어 놓기 (Drag&Drop) 방식의 GUI (Graphic User Interface)를 제공하여 서비스 정의를 지원한다. 하지만 사용자가 서비스를 정의하기 위해서는 서비스 실행 엔진 메커니즘의 동작 방법 및 조건을 나타내는 레시피 (recipes)를 정의해야 한다. [10]에서 제안하는 레시피는 PHP (Personal Hypertext Preprocessor)의 확장된 버전으로 이를 작성하기 위해서는 사용자가 웹 프로그래밍 언어 능력을 가지고 있고 시스템의 명확한 이해가 필요하므로 보다 직관적이고 쉬운 컴포지션 에디터가 요구된다. ClickScript[14]는 서비스 개발자에게 자원 및 연산자를 블록화 하여 쉽게 메시업 기능을 수행 할 수 있게 도와주는 비주얼 프로그래밍 툴이다. 자바스크립트로 구현되어 거의 모든 웹 브라우저에 동작한다. ClickScript를 이용하여 Bluetooth, Zigbee와 같은 하위 수준 프로토콜을 직접적으로 제어 할 수는 없지만 HTTP를 이용하여 각 자원에 접근할 수 있다. 각 자원은 하나의 블록으로 표현되며 연산자들과 더불어 새로운 서비스를 만들 수 있다 [14]. 하지만 이 에디터는 각 자원을 웹 기반으로 연결하여 보여주며 조건을 정의하는 등의 방식으로 수행되기 때문에 기본적인 시스템과 for, while, if, than 등의 제어문의 이해 및 서비스 흐름의 설계 능력이 필요하

다. 따라서 일반적인 사용자가 각각의 장치의 특성을 이해하고 특정 조건을 정의하기 위한 장치를 구성하는데 어려움이 따를 것으로 생각된다. JIGSAW[15]는 태블릿 기반 에디터로 여러 상황을 그림으로 표현한 퍼즐 조각들을 사용자가 결합하여 원하는 서비스를 정의할 수 있는 환경을 제공한다. 퍼즐로 여러 가지 기능적 모듈을 표현하였지만 사용자의 수준 및 특성에 따라 서로 다른 이해를 도출할 가능성이 있어 직관적이지만 오류를 범할 확률이 높다. 또한 관계연산자와 산술연산자를 지원하지 않아 산술이 필요한 서비스는 정의할 수 없다.

III. 온톨로지 기반 시멘틱 센서 데이터 처리 메커니즘

1. 오브젝트 데이터의 상호 운용성 증대를 위한 오브젝트 데이터 포맷

상황인지 서비스의 제공을 위해서는 센서 데이터의 재사용 및 상호 운용성의 제공이 필요하다. 이를 위해서는 서비스 제공 요구사항을 만족하는 공통된 데이터 포맷이 요구된다. 또한 센서는 대체적으로 전원 및 처리능력이 제한적이며 무선 통신 환경에서 데이터를 전송하므로 매우 제약적인 문제점을 가지고 있다. 따라서 적용되는 프로토콜 역시 단순하고 가벼운 특징을 가져야 하며 전송되는 데이터 또한 적은 양을 갖는 것이 유리하다. XML, EXI (Efficient XML Interchange), BXML (Binary XML), Fast Infoset 등 데이터를 표현하는 인코딩 방법은 다양하다. 단순하게 같은 정보를 적은 비트로 표현하는 것도 중요하지만 데이터를 습득했을 때 값을 전달하는 과정의 복잡도 또한 중요하다. 이는 플랫폼 혹은 데이터 처리 단계에서 비용을 낮춰주며 또한 잘못된 값의 전달을 줄여줄 수 있다. 따라서 본 논문에서는 [표 1]과 같은 오브젝트 데이터 포맷을 제안하였으며 이는 JSON 형태로 작성되어 플랫폼에 전달된다. 제안하는 오브젝트 데이터 포맷은 제어하는 오브젝트의 특성에 따라서 크게 센서와 액츄에이터로 구분한다. 일반적으로 센서는 주변의 현상을 전기적인 수치로 반환한다.

센서 데이터 포맷은 센서를 구분, 접근할 수 있도록 하는 URI (Uniform Resource Identifier)와 센서의 종류를 나타내는 Type, 측정된 값의 단위를 나타내는 Unit, 실제 측정된 측정값을 나타내는 Value, 측정 간격을 나타내는 Accuracy, 위치를 나타내는 Location, 소유 및 접근 권한을 나타내는 Ownership을 포함하고 있다. 플랫폼은 전달받은 오브젝트 데이터를 온톨로지에 매핑하게 되며 추후에 이것이 상황정보로 변환된다. 액추에이터 데이터 포맷은 센서와 마찬가지로 URI와 액추에이터의 이름, 현재 상태를 나타내는 Status, 제공해 줄 수 있는 기능 목록인 Function list, 위치를 나타내는 Location, 소유 및 제어 권한을 나타내는 Ownership을 포함한다. [표 1]의 예를 보면 센서의 종류는 온도, 값은 섭씨 28도, Jun 이 소유하고 있으며 침실에 위치하고 있다는 정보를 포함하고 있다. 이와 같은 정보를 습득하면 우리는 침실은 섭씨 28도 라는 사실을 알 수 있게 된다. 또한 액추에이터의 경우에는 Lamp 이면서 현재 상태는 On, 즉 켜져 있는 것을 알 수 있으며 제공되는 기능 목록을 통해 Turn on, Turn off 기능을 이용할 수 있으며 Location 정보를 통해 부엌에 위치하고 있다는 것을 알 수 있다. 결과적으로 우리는 이 액추에이터의 URI를 통해 부엌의 전등을 제어할 수 있다. 하지만 이와 같은 정보만으로는 상황인지 서비스 제공이 용이하지 않다. 예를 들어 [표 1]의 센서로 부터 받은 정보는 섭씨 28도의 온도정보인 것은 판단할 수 있지만 이는 말 그대로 현재 센서가 위치하는 방의 온도가 될 수 있고 특정 사용자의 체온이 될 수도 있다. 이와 같은 이유 때문에 정확한 의미상의 구별이 필요하다. 본 논문에서는 이를 위해 센서의 원시 데이터를 컨텍스트 정보로 변환하는 시맨틱 어노테이션을 제공한다.

표 1. 제안하는 오브젝트 데이터 포맷

Sensor		Actuator	
URI	../sensor/tempsensor1	URI	../actuator/lamp2
Type	Temperature	Name	Lamp
Unit	centigrade	Status	ON
Value	28	Function list	Turn on, Turn off
Accuracy	5	Location	Kitchen
Location	Bedroom	Ownership	Jun
Ownership	Jun		

2. 상황인지를 위한 기반 온톨로지

온톨로지는 특정 영역을 표현하는 데이터 모델로 개념간의 관계를 기술하는 정형화된 어휘의 집합이다. 이는 시맨틱의 기반 기술로써 본 논문에서는 온톨로지를 이용하여 센서 데이터를 상황정보로 변환한다. [그림 1]은 제안하는 기반 온톨로지 모델의 개념도이다. 제안하는 온톨로지는 장소, 오브젝트, 컨텍스트, 서비스 온톨로지의 4계층으로 구성되며 이와 같은 온톨로지의 계층적 개념을 통해 수정, 확장 및 기존 온톨로지의 재사용이 용이하다. 또한 서비스 정의에 필요한 컨텍스트를 온톨로지로 정의하여 제공함으로써 사용자는 자신이 원하는 상황을 정의하기 위해 요구되는 센서의 목록이나 기준 값을 고려하지 않아도 보다 쉽게 조건을 정의할 수 있으며 컨텍스트의 계층적 관리와 수정 및 개인화를 제공한다. 추후 장소 온톨로지를 확장하여 새로운 환경을 지원할 수 있으며 오브젝트가 추가적으로 필요할 경우 오브젝트 온톨로지에 오브젝트 정보를 추가하고 기존에 제안한 온톨로지 계층 간의 프로퍼티를 재사용하여 여러 상황 및 환경을 지원할 수 있다. 각 계층별 설명은 다음과 같다.

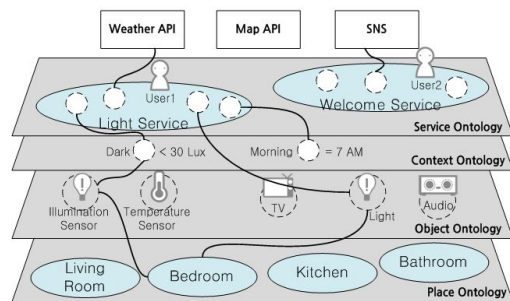


그림 1. 제안하는 기반 온톨로지 모델 개념도

2.1 장소 온톨로지

장소 온톨로지는 서비스 도메인을 구성하는 온톨로지이다. 서비스 도메인은 부엌, 거실, 침실, 서재 등 다양한 종류를 포함 할 수 있으며 각 장소는 각각의 특징을 가지고 있다. 예를 들면 침실의 경우 일반적으로 침대가 배치될 수 있으며 부엌의 경우 주방 가전 및 가스 시설을 포함하고 있다. 만약 서비스 도메인이 회사, 공

장, 농장 등으로 다양하게 확장 될 경우 요구되는 온톨로지를 확장하여 정의함으로써 다양한 서비스 도메인을 지원할 수 있다.

2.2 오브젝트 온톨로지

오브젝트 온톨로지는 서비스 도메인에 위치하는 다양한 센서 및 액추에이터를 구성한다. 오브젝트는 각각의 특징을 가지고 있다. [표 1]에서 정의한 바와 같이 센서나 액추에이터의 목적에 따라 오브젝트를 온톨로지 형태로 정의하였으며 센서의 경우 *hasValue* 프로퍼티를 통해 습득한 값과 연결되게 된다. 액추에이터의 경우에는 *hasStatus* 프로퍼티를 통해 상태 값을 가지게 된다. 또한 위에서 설명한 장소 온톨로지와의 연결을 통해 현재 센서의 위치를 정의한다. 이와 같이 상관관계에 있는 온톨로지 사이의 연결을 통해 오브젝트에서 습득한 정보를 매핑한다.

2.3 컨텍스트 온톨로지

컨텍스트 온톨로지는 서비스 조건이 되는 컨텍스트를 정의한다. 사용자는 특정 상황 (예: 어둡다, 덥다, 밝다, 습하다 등)으로 서비스 제공 조건을 정의하게 되는데 본 논문에서는 이러한 특정 상황을 컨텍스트라 정의한다. 이는 특정 상황을 판단할 수 있는 센서와의 연결을 통해 정의되며 상황에 따라 조건 값을 가지게 된다. 예를 들어 조도 센서에서 습득된 값이 30 lux 보다 적으면 “어둡다” 라는 컨텍스트로 정의할 수 있다. 사용자는 컨텍스트 온톨로지를 통해 자신이 원하는 서비스 제공 조건을 판단하기 위한 사항 (센서의 종류, 컨텍스트 조건 값 등)에 관한 지식이 없어도 기반 온톨로지에 미리 정의된 기본 컨텍스트를 통해서 보다 쉽게 서비스 조건을 정의할 수 있다. 또한 컨텍스트의 계층적 관리를 통해 다수의 컨텍스트를 이용하여 새로운 컨텍스트를 생성하는 컨텍스트 메시지를 용이하게 한다.

2.4 서비스 온톨로지

가장 상위에 위치하는 서비스 온톨로지는 서비스 조건에 해당하는 컨텍스트와 서비스 목적 달성을 위한 액추에이터의 조합으로 이루어진다. 사용자가 선택 혹은

생성한 컨텍스트와 액추에이터의 종류 및 기능, 최종 상태 등의 정보를 통해 서비스를 정의한다. 이는 SWRL (Semantic Web Rule Language)과 Jena rule에 의해 표현된다.

3. 타겟 서비스 시나리오

센서 데이터 처리 단계의 설명에 앞서 보다 명확한 처리 단계를 보여주기 위해 타겟 서비스 시나리오를 정의한다. [그림 2]는 제안하는 타겟 서비스 시나리오 환경을 나타낸다. 본 시나리오는 1인 가구 환경이며 사용자의 위치는 움직임 센서에 의해 구별될 수 있다고 가정하였다. [그림 2]의 번호는 사용자가 이동하는 순서를 나타내며 각 순서의 시나리오는 다음과 같다.

- ① 침실에 설치된 조도 센서에 의해 침실의 조도가 측정되고 있다. 사용자가 아침에 일어났을 때 침실이 어둡다면 램프를 켜고 커튼을 연다.
- ② 기상을 한 사용자는 부엌으로 이동한다. 아침 시간에 부엌으로 사용자가 이동한 것을 감지하게 되면 커피포트가 동작한다.
- ③ 사용자가 거실로 이동하면 TV에서는 주로 보던 채널이 방송된다.
- ④ 사용자는 출근 준비를 위해 욕실로 이동한다. 샤워를 시작하면 욕실안의 습도가 증가한다. 습도 센서는 욕실의 습도를 측정하고 일정 습도 이상이 된 것을 감지하면 환풍기를 동작시킨다.

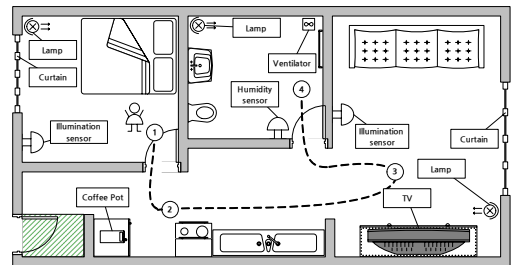


그림 2. 타겟 서비스 시나리오 환경

위의 시나리오는 서비스 조건이 되는 컨텍스트와 서비스 행위가 되는 액추에이터의 기능과 같은 구성요소를 포함하고 있다. 시나리오 4번의 경우 환풍기가 켜지

는 서비스의 조건은 사용자의 위치와 습도에 관한 컨텍스트이다. 또한 1번의 경우 아침이라는 상황과 어둡다는 상황이 합쳐져 새로운 컨텍스트가 서비스 조건이 된 경우이다. 이와 같은 다양한 조건의 시나리오를 서비스로 제공하고자 할 경우 다양한 컨텍스트를 만족시키는 새로운 컨텍스트가 필요하며 이를 위해 컨텍스트 매시업이 요구된다.

4. 온톨로지 기반의 센서 데이터 처리 단계

앞서 설명한 바와 같이 센서 데이터는 시스템, 문법, 구조, 의미적 측면의 이질성을 가지고 있다. 본 논문에서는 이를 효과적으로 제거하기 위해 온톨로지 기반의 센서 데이터 처리 메커니즘을 제안한다. [그림 3]은 제안하는 메커니즘의 구현구조이다. 센서 데이터의 효율적인 처리를 위해서 Java EE 기반의 실험 환경을 구축하였다. Apache Tomcat 6.0 은 RESTful 웹 서비스의 제공 및 관리를 위해서 사용하였고 시멘틱 데이터의 처리를 위해 자바 기반의 Jena framework 2.6.4를 이용하였다. 또한 시멘틱 데이터의 효율적인 질의를 위해 SPARQL API를 이용하였다. 여러 상황정보가 입력된 온톨로지에서는 상황의 판단 및 추론을 위해 Pellet 추론기를 이용하였다. SWRL과 Jena rule을 이용하여 데이터 처리 및 서비스 제공을 위한 서비스 규칙을 정의하였다. 또한 오브젝트 데이터의 손쉬운 습득 및 Jena 프레임워크와의 파싱을 위해 Jackson API 2.2.3을 이용하여 JSON 데이터와 자바 오브젝트간의 변환을 처리하였다. 그리고 온톨로지의 저장 및 관리를 위해 MySQL 5.1 과 JDBC 5.1.25를 이용하였다.

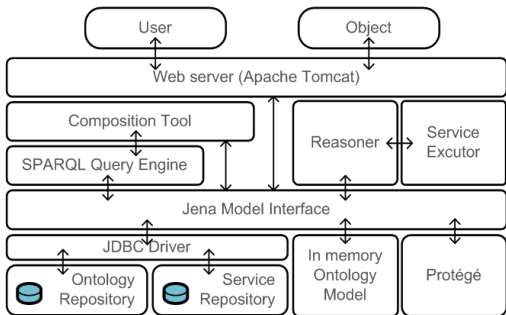


그림 3. 센서 데이터 처리를 위한 구현구조

[그림 4]는 오브젝트 및 서비스 추상화, 시멘틱 어노테이션, 추론, 서비스 실행의 4단계로 이루어진 제안하는 메커니즘의 처리 단계를 보여준다. 다양한 프로토콜 및 메시지 포맷을 가지고 있는 오브젝트는 [표 1]와 같은 형식으로 추상화되며 사용자의 서비스 정의 또한 추상화되어 플랫폼에 전달된다. 이렇게 전달된 오브젝트나 서비스의 정보는 [그림 1]의 기반 온톨로지에 매핑되는데 이 과정을 시멘틱 어노테이션이라고 한다. 이렇게 매핑된 온톨로지에서 상황을 인지하기 위해 정의된 각종 컨텍스트는 추상화 과정을 통해 분류(classification)되며 최종적으로 Jena rule로 정의된 서비스 조건 및 결과 정보에 의해 각종 액츄에이터 혹은 사용자 장치에 상태정보를 전달하고 제어한다.

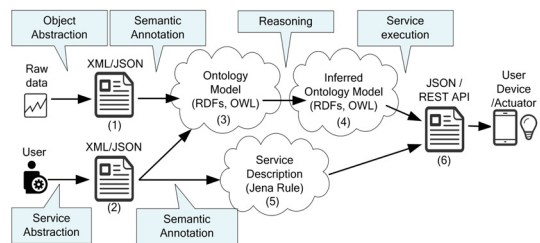


그림 4. 온톨로지 기반의 센서 데이터 처리 단계

4.1 오브젝트 및 서비스 추상화 (Object & Service abstraction)

[그림 5]는 센서 데이터의 시스템, 문법, 구조적 측면의 이질성을 제거하기 위해 추상화를 거친 오브젝트 데이터이다. 제한적인 센서 환경에서의 효율적인 전송을 위해 파싱이 쉽고 데이터양이 적은 JSON 포맷을 사용하였다. 이는 타겟 시나리오 4를 기반으로 한 오브젝트 데이터로 오브젝트의 URI, 유형은 습도, 측정된 값은 70%임을 알아 낼 수 있으며 위치정보를 통해 욕실에 설치된 것을 알 수 있다. 결과적으로 욕실의 습도는 70%임을 파악할 수 있다.

```
{
  "URI" : "HumiditySensor 1", "Type" : "HumiditySensor",
  "Unit" : "Percent", "Value" : 70, "Accuracy" : 5,
  "Location" : "Bathroom_Jun", "Ownership" : "Jun"
}
```

그림 5. 추상화된 오브젝트 데이터 (JSON)

[그림 6]은 서비스 요구사항을 정의한 내용이다. 서비스는 서비스 추상화를 통해 [그림 6]과 같이 기술된다. 이는 컨텍스트와 서비스 규칙을 정의하는데 사용되며 [그림 4]-(2)에 해당된다. 이는 서비스와 컨텍스트의 이름, 하위 컨텍스트의 목록 및 기준 값, 액추에이터의 이름 및 기능 목록을 포함하고 있다. 컨텍스트를 계층적으로 구성하는 이유는 사용자가 원하는 서비스 조건이 다수의 컨텍스트를 만족하는 조건일 경우 이를 정의하기 위한 것이다. 결과적으로 [그림 6]의 메시지를 통해서 사용자가 정의한 서비스의 요구사항을 파악할 수 있으며 온톨로지에 컨텍스트를 생성할 수 있다. 내용을 보면 서비스 조건인 컨텍스트는 습도가 50 이상일 경우 습하다 (Damp)로 정의하고 있으며 이를 만족할 경우 *Ventilator1*을 동작시킨다.

```
{
  "ServiceName" : "bath_Jun", "ContextName" : "Damp_Jun",
  "Context": [
    { "Type" : "TemperatureSense/Damp", "Operation" : ">", "Value" : 50 },
    "Actuator" : [
      { "Name" : "Ventilator1", "Function" : "Turn on" } ]
    }
  }
```

그림 6. 추상화된 서비스 요구사항 (JSON)

4.2 시멘틱 어노테이션 (Semantic annotation)

추상화된 오브젝트 데이터를 수집하면 의미적 이질성을 제거하고 추후 상황 인지를 위한 추론에 이용하기 위해 온톨로지에 매핑 한다. [그림 7]은 제안하는 기반 온톨로지를 Protege의 온톨로지 시각화 툴 *OntoGraf*를 이용하여 나타낸 것이다. [그림 1]의 기반 온톨로지는 장소, 오브젝트, 컨텍스트 로 이루어져 있으며 서비스 온톨로지는 이들을 연결하여 정의한다. [그림 5]의 추상화된 오브젝트 데이터는 시멘틱 어노테이션을 통하여 [그림 7]과 같이 온톨로지에 매핑된다.

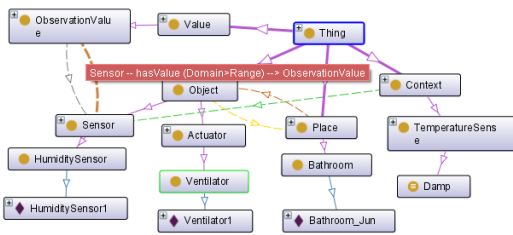


그림 7. 기반 온톨로지

오브젝트 데이터는 Type을 통해 현재 습득한 데이터가 습도임을 알 수 있으므로 온톨로지의 센서 클래스의 하위 클래스인 *HumiditySensor* 클래스에 인스턴스를 생성한다. 이는 미리 정의된 기반 온톨로지의 특성에 따라 *Sensor* 클래스는 수집한 값을 갖기 때문에 *Value* 클래스의 하위 클래스인 *ObservationValue* 클래스와 *hasValue* 프로퍼티를 이용하여 연결된다. 또한 미리 정의된 컨텍스트는 여러 가지 현상과 감각을 분류하여 계층적 클래스로 구성되어 있다. [그림 5]의 데이터의 경우 습도센서에서 수집된 것 이므로 *TemperatureSense* 클래스의 하위 클래스인 *Damp* 클래스와 연결되어 있다. 또한 *Object* 클래스는 *Place* 클래스와 연결되어 위치를 알 수 있다. 결과적으로 센서에서 수집된 정보를 온톨로지에 적용하여 “욕실은 습하다.”와 같은 상황 정보로 처리 한다.

[그림 8]은 특정 컨텍스트(*Damp_Jun*)의 OWL 표현이며 기반 온톨로지의 일부로 컨텍스트의 정의를 보인다. 이는 [그림 4]-(3)의 예로서 컨텍스트 클래스의 인스턴스인 *Damp_Jun* 은 습도 센서 인스턴스인 *HumiditySensor1*과 *HumiditySensorObservationFrom* 프로퍼티를 이용하여 연결된다. 또한 *hasHumiditySensorObservationValue* 프로퍼티를 이용하여 습도 값 정수형 70 을 가진다. 이 상수 값은 *ObservationValue* 클래스의 인스턴스로 존재한다.

```
<!-- http://www.semanticweb.org/dial/ontologies/2013/8/shbo#Damp_Jun -->
<owl:NamedIndividual rdf:about="#shbo:Damp_Jun">
  <rdf:type rdf:resource="#shbo:TemperatureSense"/>
  <hasHumiditySensorObservationValue rdf:datatype="xsd:integer">70
  </hasHumiditySensorObservationValue >
  <shbo:HumiditySensorObservationFrom rdf:resource="#shbo:HumiditySensor 1"/>
</owl:NamedIndividual >
```

그림 8. Damp_jun 컨텍스트의 OWL 표현

4.3 추론 (Reasoning)

서비스 조건인 컨텍스트를 정의하고 센서로부터 전달받은 정보를 온톨로지에 매핑하면 컨텍스트 판단 조건의 만족 유무를 판단해야 한다. 이를 위해서 본 논문에서는 *Pellet* 추론기를 사용한다. 추론기는 조건에 맞는 클래스나 인스턴스를 분류 하는 역할을 한다. 이를 통해 센서 값을 가지고 있는 컨텍스트 인스턴스가 특정

컨텍스트의 조건에 맞는 경우 추론기가 이를 분류한다. [그림 9]는 추론 결과를 보여주는 것으로 Protege 내부에 클래스의 정의 상태를 보여준다.

[그림 9]에서는 *Damp* 컨텍스트를 클래스로 정의하였으며 이때 50 이상의 값을 갖는 인스턴스가 이 클래스에 속하게 된다. 이는 *hasHumiditySensorObservationValue* 프로퍼티를 이용하여 값을 가지며 결과적으로 *Damp_Jun* 인스턴스는 70 이라는 값을 가지므로 [그림 9]의 하단부와 같이 *Damp* 클래스로 구분된 것을 확인할 수 있다.

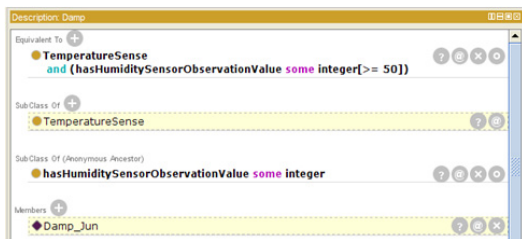


그림 9. Damp 클래스의 추론 결과

4.4 서비스 실행 (Service execution)

기본 온톨로지를 이용하여 서비스 조건의 만족을 판단하면 서비스를 실행해야 한다. 이를 위해 본 논문에서는 Jena 프레임워크를 기반으로 서비스 조건의 판단 및 실행을 한다. [그림 10]은 Jena rule을 이용한 서비스 정의를 보여준다.

```
[bath_Jun:
(shbo:Damp_Jun rdf:type shbo:Damp),
(shbo:Ventilator1 shbo:hasStatus "OFF"^^xsd:string) ->
(shbo:Ventilator1 shbo:hasStatus "ON"^^xsd:string)
]
```

그림 10. Jena rule을 이용한 서비스 정의

이는 [그림 4]-(5)에 해당된다. *Damp_Jun* 인스턴스가 *Damp* 클래스의 구성원 (type)이 된다면 서비스 실행 조건을 만족한 것이다. 이때 *Bath_Jun* 이라고 정의한 서비스가 실행된다. 따라서 *Ventilator1*은 OFF의 상태에서 ON의 상태로 변경된다. 이와 같이 서비스 조건에 의해 액추에이터 상태가 변경되려면 이러한 상태정

보를 액추에이터에 전달해야한다. 이 단계는 [그림 4]-(6)에 해당된다. 만약 전달해야 할 정보의 내용이 다수의 항목을 포함하고 있다면 오브젝트 데이터 포맷과 같이 JSON 형태로 전달된다. 하지만 단순히 상태정보를 변경하는 것이라면 간단하게 REST operation을 이용할 수 있다. 예를 들어 *Ventilator1*의 상태정보를 on으로 변경하고자 한다면 해당 오브젝트의 URI (예 : <http://hostaddr/put/actuator/ventilator1/status/on>)를 이용하여 변경한다. 이는 오브젝트 내부의 REST operation 처리 모듈에 따라 다르게 표현될 수 있다.

IV. 사용자 친화적 서비스 정의 환경

모든 서비스는 특정 조건에서 제공된다. 예를 들면 극장에서 영화표를 구매하고 상영관에 들어가면 시작 시간에 맞추어 영화가 방영된다. 또는 비밀번호 잠금장치에 올바른 비밀번호를 입력한다면 문이 열린다. 이와 같이 어떤 형태의 서비스라 할지라도 서비스를 제공하는 조건 및 제공되는 행위, 동작, 콘텐츠 등의 서비스 행동은 정해져 있다. 상황인지 서비스의 경우에도 마찬가지이다. 다만 서비스 조건의 만족 여부를 플랫폼의 기본 온톨로지를 통해 판단하고 해당 서비스 행동을 제공하도록 지시할 뿐이다. 2장의 관련연구에서 서술한 바와 같이 사용자에게 서비스 정의 환경을 제공하는 연구는 계속되고 있으나 기존의 서비스 제공 환경은 서비스 정의에 필요한 각종 상황의 기준 값, 온톨로지의 추론 결과를 알기 위한 SPARQL, 서비스 정의를 위한 스크립트 기반의 각종 프로그래밍 언어능력을 요구하는 등 일반적인 사용자에게는 수행하기 어려운 실정이다. 따라서 본 논문에서는 플랫폼의 구조, 컨텍스트의 정의를 위해 필요한 센서의 목록이나 조건, 서비스를 정의하기 위한 프로그래밍 능력 등이 없는 일반적인 사용자가 쉽게 자신이 원하는 서비스를 정의할 수 있는 환경을 제안한다. 이 환경에서 사용자는 단지 서비스 제공 조건 및 서비스 제공결과인 액추에이터의 기능 두 가지 요소만을 선택하여 서비스를 정의할 수 있다. 사용자는 서비스를 생각 할 때 내가 어떤 동작, 정보, 행위를 제공받

을 수 있는지 고민하고 또한 그 결과가 어떤 상황에 제공되는 것이 좋을지 판단한다. 이와 같이 서비스 정의 환경은 서비스의 결과를 정의 하는 것으로 시작된다. 또한 서비스 상황은 여러 가지 컨텍스트로 정의되며 사용자는 기 정의된 컨텍스트를 이용하여 자신만의 서비스 조건을 정의할 수 있다.

1. 제안하는 사용자 친화적 서비스 정의 과정

[그림 11]은 본 논문에서 제안하는 사용자 중심 서비스 정의 과정을 나타낸 것으로 단계마다 필요한 입출력을 정의하고 있다. 이는 장소 설정, 오브젝트 설정, 오브젝트 및 기능 선택, 상황 설정의 4 단계로 구성되어 있다. 서비스 정의 과정은 크게 서비스 환경을 정의하는 앞 두 단계와 서비스 자체를 정의하는 뒤의 두 단계로 구분 될 수 있다. 플랫폼 관점에서는 사용자에게 상용인지 서비스를 제공하고자 할 때 고려해야 할 요소들이 매우 많다. 이는 환경적으로 인지해야할 공간, 오브젝트의 종류 및 배치, 서비스 조건 등 매우 다양하다. 이를 사용자가 정확하게 정의한다면 플랫폼 입장에서도 처리(추론)해야 할 정보들이 확연하게 감소되며 정확한 서비스 제공이 용이해 진다. 추후 사용자의 행동을 반영하는 적응형 서비스로 발전시켜 나갈 수 있는 장점도 가지고 있다. 관련연구에서 살펴본 모든 사용자 중심 서비스 정의 환경은 한 화면에 전체적인 서비스를 정의하는 플로우차트 개념으로 전체적인 흐름을 파악하기엔 좋지만 사용자가 전체적인 서비스 흐름을 이해하고 정의해야 하며 단계를 나누지 않고 모든 요소들을 나열하여 실수가 생길 수 있다.

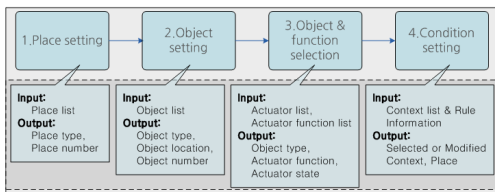


그림 11. 서비스 정의 과정

제안하는 정의 과정은 [그림 11]의 서비스 정의 과정을 단계별로 진행하는 방식으로 서비스 정의의 마법사 방

식을 적용하여 사용자가 정의해야 하는 사항들을 순차적으로 도출하여 어떤 사항을 선택해야 하는지 도움을 준다. 따라서 사용자의 오류를 줄일 수 있으며 전체적인 서비스 흐름의 이해를 요구하지 않고 단순히 가능한 요소들을 선택하는 방식으로 서비스를 정의할 수 있다. 이를 위해 본 연구에서는 [그림 12]의 웹기반 서비스 정의 UI를 제공한다. 이는 HTML5, CSS3, 자바스크립트 등을 이용하여 구현하였으며 데이터 전달을 위해 HTTP와 JSON을 이용한다.

1.1 장소 설정 단계(Place setting)

첫 단계는 [그림 12]와 같이 서비스 제공 환경을 정의하는 장소 설정 단계이다. 같은 가정환경이라도 방의 개수 및 종류의 차이가 존재하며 추후 회사, 공장 등 다양한 환경으로 확장 될 수 있기 때문이다.

본 논문은 가정환경을 타겟으로 기반 온톨로지의 Place 클래스에 일반적인 가정에 존재하는 장소의 특징 및 종류가 정의되어 있다. 장소 설정을 위해 기반 온톨로지에 정의된 Place 클래스의 하위 클래스 목록(장소의 종류)을 UI에 전달한다. 사용자는 [그림 12]와 같이 전달받은 목록 중 자신의 서비스 환경에 맞는 종류를 선택 (Check)하고 만약 한 종류의 장소가 여럿 존재할 경우 장소의 개수를 입력한다. 사용자의 입력이 끝나면 [그림 12]의 하단부와 같이 사용자가 선택한 장소가 생성된다. 이와 같은 단계를 통해 사용자의 서비스 환경이 기반 온톨로지에 적용된다.



그림 12. 웹기반 서비스 정의 UI (장소 설정 단계)

[그림 13]은 [그림 12]와 같이 장소를 설정했을 때 그 결과가 적용된 기반 온톨로지의 상태를 보여준다. 사용자가 선택한 결과에 따라 장소 온톨로지의 클래스 하위

에 인스턴스 형태로 생성된다. [그림 13]을 보면 *Kitchen_Jun* 인스턴스가 *Kitchen* 클래스의 소속으로 생성된 것을 확인할 수 있다.

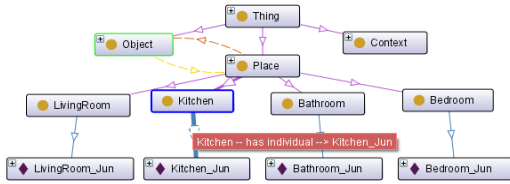


그림 13. 장소 설정 단계가 적용된 기반 온톨로지 상태

1.2 오브젝트 설정 단계 (Object setting)

장소 설정 단계가 끝나면 서비스 제공에 필요한 오브젝트를 등록하는 오브젝트 설정 단계가 진행된다. 사용자에게 실제적인 서비스를 제공하는 것은 어떤 행위 및 동작을 할 수 있는 액추에이터라고 볼 수 있다. 따라서 사용자는 원하는 액추에이터의 기능을 선택하여 서비스를 정의한다. 플랫폼은 범위 안에 있는 오브젝트의 추상화된 데이터를 제공 받아 기반 온톨로지의 *Object* 클래스에 등록한다. 이렇게 등록된 정보 중 사용자가 권한이 있는 액추에이터의 목록을 UI에 제공한다. [그림 14]는 오브젝트 설정단계의 서비스 정의를 위한 UI를 보여준다. 사용 가능한 목록은 상단부에 노출하였으며 사용자는 끌어 놓기 형식으로 원하는 액추에이터를 특정 장소에 배치한다. 배치된 액추에이터는 [그림 7]과 같이 종류에 따라 기반 온톨로지에 인스턴스 형태로 등록된다. 이와 같이 오브젝트 설정 단계를 통해 배치된 액추에이터는 장소 온톨로지와 연결하여 위치 정보를 갖는다.

기존의 연구들은 실제 오브젝트의 배치를 위해 추가적인 메타데이터 입력 또는 스크립트 기반의 프로그래밍 능력을 요구한다. 또한 동일한 환경을 재사용하기 위해서 반복하여 환경 정의를 해야 하는 단점을 가지고 있다. 그러나, 제안하는 환경은 프로그래밍 능력이 없는 사용자가 손쉽게 환경을 정의할 수 있고 환경을 재사용하여 추가적인 설정을 줄여줄 수 있다. 또한 사용자의 서비스 환경이 변경되거나 추가적인 장소를 설정하고자 할 경우 장소 및 오브젝트 설정 단계를 다시 수행하여 새로운 서비스 환경을 추가할 수 있으며 이렇게 설

정된 정보는 온톨로지에 사용자별로 관리된다.

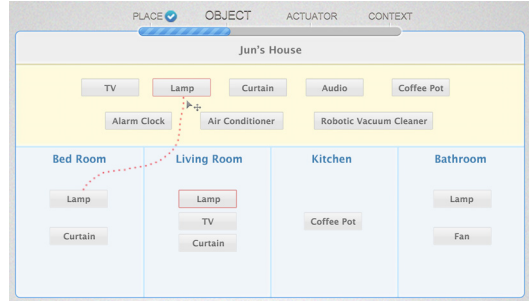


그림 14. 웹기반 서비스 정의 UI (오브젝트 설정 단계)

1.3 오브젝트 및 기능 선택 단계 (Object & function selection)

이 단계에서는 서비스 기능 및 상태를 정의한다. [그림 15]는 오브젝트 및 기능 선택 단계의 서비스 정의를 위한 UI이다. 우선적으로 다양한 서비스를 효과적으로 관리하기 위해 서비스 이름을 입력한다. 서비스 정의 환경은 [그림 15]의 상단부와 같이 장소 및 오브젝트 설정 단계를 통해 온톨로지에 정의된 서비스 환경 정보를 이용하여 사용가능한 액추에이터 목록을 보여준다. 사용자는 제어하길 원하는 액추에이터를 선택한다. 액추에이터가 선택되면 해당 기능 목록이 출력된다. 사용자는 이 목록 중 자신이 원하는 기능 혹은 상태를 선택하여 서비스 행위를 정의한다. 예를 들어 “전등이 켜지고 커튼이 열림” 과 같이 사용자가 다수의 액추에이터 기능을 동시에 제공받고 싶다면 사용자는 가용 액추에이터 목록에서 다수의 액추에이터와 해당 기능을 선택한다.

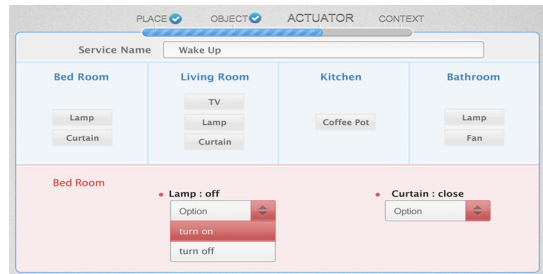


그림 15. 웹기반 서비스 정의 UI (오브젝트 및 기능 선택 단계)

[그림 16]은 Protege를 이용하여 특정 액추에이터 인스턴스의 정의를 표현한 것이다. 앞서 설명한 바와 같이 기반 온톨로지에는 해당 액추에이터가 지원하는 기능 목록 및 위치, 현재 상태 정보가 포함되어 있다. 이러한 정보를 바탕으로 사용자는 액추에이터가 지원하는 목록 및 상태를 알 수 있으며 이를 변경하여 액추에이터를 제어하게 된다.

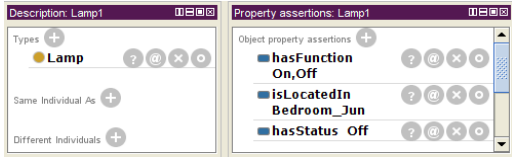


그림 16. 액추에이터 인스턴스의 정의

1.4 상황 설정 단계 (Condition setting)

서비스 제공 조건은 플랫폼에서 기 정의된 다양한 컨텍스트를 통해 정의된다. 이는 컨텍스트의 특성에 따라 분류되어 있으며 다양한 상황 (예: 아침 + 어둡다, 욕실에 사용자가 있을 때 + 습하다)을 정의 할 수 있도록 컨텍스트 메시지를 제공한다. 하지만 컨텍스트의 특성에 따라 동시에 선택된다면 조건을 만족시킬 수 없는 경우가 존재한다. 이를 위해서 상충되는 조건 (예: 춥다, 덥다)을 감각이나 현상의 종류에 따라 분류하여 제공한다. 같은 분류에 존재하는 컨텍스트는 다중 선택 될 수 없으며 다른 분류 간에는 동시에 선택할 수 있게 하였다. 또한 컨텍스트는 기준을 가지고 있는데 이는 사용자마다 그 기준이 다를 수 있다. 예를 들어 한 사용자는 26도의 온도가 덥다고 느끼는 반면 다른 사용자는 28도의 온도가 되어야 덥다고 느낄 수 있기 때문이다. 이러한 상황을 만족시키기 위해 각 컨텍스트의 기준을 변경할 수 있게 하였고 설정된 조건은 [그림 6]과 같은 서비스 요구사항으로 플랫폼에 전달된다. [그림 17]은 상황 설정 단계의 서비스 정의를 위한 UI 이다. 좌측에는 사용 가능한 컨텍스트가 분류되어 제공되며 우측에 선택된 컨텍스트의 기준값이 표현된다. 이와 같이 선택 혹은 수정된 컨텍스트는 컨텍스트 메시지를 통해 새로운 컨텍스트로 생성된다. 컨텍스트는 계층적 구조를 가지므로써 기준에 정의된 기본 컨텍스트 및 기준값을 포함한다. 이와 같은 계층적 관리를 통해 기본 컨텍스트를

기반으로 사용자가 원하는 다양한 컨텍스트를 생성 / 관리 할 수 있다.

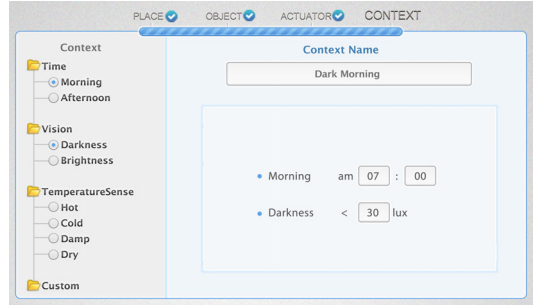


그림 17. 웹기반 서비스 정의 UI (상황 설정 단계)

V. 비교 분석

본 장에서는 제안하는 사용자 친화적 서비스 정의 환경과 관련 연구들을 비교분석한다. [표 2]는 사용자 친화적 서비스 정의 환경을 비교 분석한 결과로써 사용하고 있는 비교 항목은 관련연구의 특징 및 장단점 분석을 통해 도출되었다. 본 논문에서는 사용자 친화적 서비스 정의 환경의 제공을 위해 3가지 사항 (기반 온톨로지, 센서 데이터 처리 메커니즘, UI)을 제안하였다. 이를 위한 비교 항목은 총 6가지로 구성되어 있으며 기반 온톨로지의 적용을 통해 [표 2]의 1-4번 항목을 제공하고 센서 데이터 처리 메커니즘을 통해 4, 6번 항목을 제공한다. 또한 사용자 친화적 UI를 통해 1-6번 항목의 환경을 제공한다.

표 2. 사용자 친화적 서비스 정의 환경 비교

비교항목	UbiSOA [10]	ClickScript [14]	JIGSAW [15]	제안 방식
1. 기본 컨텍스트 제공	x	x	o	o
2. 컨텍스트 수정 및 개인화 제공	x	x	x	o
3. 컨텍스트 메시지 제공	o	x	x	o
4. 환경 및 조건 재 사용 가능	o	x	o	o
5. 순차적 서비스 정의 가능 제공	x	x	x	o
6. 프로그래밍 능력 불필요	x	x	o	o

1. 기본 컨텍스트 제공

사용자가 처한 환경은 매우 다양하며 이를 판단하기 위한 기준도 매우 다양하다. 하지만 일반적인 사용자는 자신이 원하는 상황 정보의 판단을 위해 어떤 정보가 필요한지, 어떤 기준으로 판단할 수 있는지 구상하기란 어렵다고 할 수 있다. 단순히 ‘덥다’와 같은 상황의 경우에는 온도 센서에서 생성된 데이터를 이용할 수 있다고 생각할 수 있지만 다양한 주변 요인에 따라서 판단이 어려울 수 있다. 예를 들어 섭씨 15도의 온도의 경우 계절, 습도등 다양한 요인에 따라 사람이 느끼는 기준은 변할 수 있기 때문이다. UbiSOA, Click-Script 등의 연구에서는 사용자가 컨텍스트를 정의하기 위해 사용할 수 있는 센서를 선택하고 기준 값을 입력하고 연산식을 표현한다. JIGSAW는 퍼즐 조각 형태로 미리 정의된 기능을 사용할 수 있지만 관계연산 및 산술 연산자를 지원하지 않아 세부적인 서비스 정의가 어렵다. 그러나 제안하는 방식은 미리 정의된 기본 컨텍스트를 제공함으로써 사용자는 자신이 원하는 상황을 선택하고 그 상황을 정의하기 위한 추가적인 지식을 요구 하지 않는다.

2. 컨텍스트 수정 및 개인화 제공

사람마다 특정 현상에 대해 느끼는 기준이 다르므로 일반적인 기준을 사용자에게 맞추어 수정 혹은 개인화 할 필요가 있다. 예를 들어 한 사용자는 40lux의 조도에서 어둡다고 느끼는 반면 다른 사용자는 어둡지 않다고 느낄 수 있기 때문이다. 따라서 손쉽게 기존에 정의한 컨텍스트를 수정하고 사용자 마다 다른 기준의 컨텍스트를 사용할 수 있어야 한다. UbiSOA, Click-Script, JIGSAW 모두 사용자 별로 컨텍스트를 관리하지 않으며 수정을 위해서는 새로운 컨텍스트를 생성해야 한다. 또한 [9]는 미리 정의된 컨텍스트의 수정을 위해서 온톨로지에 직접 프로그래밍 하여 추가해야 한다. 그러나 제안하는 방식은 온톨로지에 인스턴스 형태로 사용자 별 컨텍스트를 생성, 관리하고 클래스의 조건을 통해 전체적인 컨텍스트 유형을 정의, 관리한다. 이를 통해 사용자는 기 정의된 컨텍스트를 수정, 개인화 하여 사용할 수 있다.

3. 컨텍스트 매시업 제공

서비스 조건은 매우 다양하기 때문에 이것을 표현하기 위해서는 기존에 존재하는 컨텍스트의 재사용 및 매시업이 필요하다. UbiSOA에서는 부분적으로 기존에 정의한 컨텍스트간의 연결을 통해 새로운 컨텍스트의 생성이 가능하다. 하지만 Click-Script와 JIGSAW는 컨텍스트 간의 연결을 통해 새로운 컨텍스트를 생성하는 것이 아니라 전체적인 정의를 통해 컨텍스트를 생성한다. 그러나 제안하는 방식은 컨텍스트간의 계층적인 구조를 통해 하위 컨텍스트를 가질 수 있다. 따라서 사용자는 기본 컨텍스트를 이용하거나 수정, 및 개인화된 컨텍스트를 재사용 하여 새로운 컨텍스트를 정의할 수 있다.

4. 환경 및 조건 재사용 가능

비슷한 상황 및 환경을 재사용 하여 새로운 서비스를 정의 할 필요가 있다. 예를 들어 기존의 서비스가 A라는 상황에 B라는 동작을 제공했을 때 같은 상황에 다른 장소의 C라는 동작을 제공받길 원하는 경우이다. UbiSOA와 JIGSAW의 경우 기존에 정의한 컨텍스트를 모듈형태로 재사용할 수 있다. 하지만 Click-Script의 경우 환경 / 서비스 구분 없이 한 화면에서 모두 정의해야 하므로 매번 환경 및 서비스를 정의해야 한다. 또한 환경 (예: 거실, 부엌, 침실 등)이 수정되거나 사용자 별로 환경이 변화할 수 있는데 제안하는 방식에서는 온톨로지의 *Place*와 *Object* 클래스로 환경 설정 정보를 관리하여 상황의 재사용뿐만 아니라 환경의 변화에 적용할 수 있다.

5. 순차적 서비스 정의 기능 제공

상황인지 서비스를 정의하기 위해서는 다양한 환경, 관련 센서, 현상에 관한 기준 값, 서비스 행위 및 절차 등 수많은 사항을 고려해야 한다. 일반적인 사용자는 어떤 항목을 우선적으로 고려해야할지 생각하기 어려우며 이는 복잡한 서비스의 경우 더 심각하다고 할 수 있다. 또한 각 플랫폼의 처리 과정에 따라 고려해야 할 사항 및 순서는 달라질 수 있으며 이에 따라 제공되는

서비스는 사용자의 의도 및 요구사항을 만족시키지 못하는 경우가 발생할 수 있다. 이에 대해 관련연구 모두 서비스의 흐름에 관계없이 한 화면에서 서비스를 정의하는 방식만을 제공한다. 그러나 본 논문에서는 서비스 정의의 마법사 방식을 제안하여 사용자가 결정해야 할 사항을 순차적으로 제시하여 사용자의 혼란 및 오류를 줄여준다. 또한 [그림 15]의 상단처럼 현재 처리과정을 확인할 수 있게 하여 사용자가 직관적으로 서비스 정의 방식을 이해할 수 있도록 한다.

6. 프로그래밍 능력 불필요

일반 사용자는 PHP, 자바스크립트 같은 프로그래밍 능력이 없고 if, then, else와 같은 조건문, for 또는 while과 같은 제어문의 이해와 응용능력이 부족하다. UbiSOA는 단위 모듈마다 PHP 기반의 프로그래밍이 필요하며 모듈간의 연결을 통해 서비스를 정의한다. Click-Script의 경우 블록간의 연결 및 연산자를 사용하여 서비스를 정의하며 새로운 블록을 정의하기 위해 자바스크립트 프로그래밍이 요구된다. 이러한 방법은 플로우차트의 작성이 익숙하지 않은 사용자들에게 어려울 수 있으며 연산자의 연결 시 시스템의 이해가 요구된다. JIGSAW의 경우 단위 모듈을 퍼즐조각의 그림으로 표현하여 프로그래밍 능력이 요구되지는 않지만 표현의 다항성이 부족하며 퍼즐 조각의 의미를 다르게 이해 할 가능성이 있고 정보의 전달 및 조건의 설정을 구분하기 힘들다. 그러나 제안하는 방식은 기본 컨텍스트를 선택하고 필요에 의해 기준 값을 변경하고 원하는 동작을 선택하는 방식으로 일반적인 사용자의 프로그래밍 능력이 요구되지 않으며 순차적으로 서비스를 정의할 수 있다.

VI. 결론

상황인지형 서비스 제공을 위해서는 센서를 이용하여 사용자 주변의 상황, 정황, 상태 정보를 추적하여야 한다. 하지만 센서에서 수집된 데이터는 시스템, 문법, 구조, 의미적 측면과 같은 이질성을 가지고 있어 다양한 서비스에 재사용이 어렵다. 또한 사용자의 다양한

요구를 만족시키기 위해 사용자가 원하는 서비스를 정의하는 환경에 관한 요구가 증대되고 있다. 하지만 기존 연구의 경우 서비스를 정의하기 위해 서비스의 전체적인 흐름 및 서비스 조건을 정의하기 위해 필요한 센서 및 기준 값, 이를 표현하기 위한 프로그래밍 능력이 요구되는 등 일반적인 사용자에게 다소 어렵고 복잡한 환경을 제공하였다. 따라서 본 논문에서는 센서 데이터를 상황 정보로 처리하는 온톨로지 기반 시멘틱 센서 데이터 처리 메커니즘과 사용자 친화적 서비스 정의 환경을 제안하였다. 온톨로지 기반 시멘틱 센서 데이터 처리 메커니즘은 오브젝트 및 서비스 추상화를 통해 문법 및 구조적 측면의 이질성을 제거하였으며 시멘틱 어노테이션을 통하여 추상화 된 오브젝트 데이터를 기반 온톨로지에 매핑하여 상황 정보로 전환하였다. 또한 상황의 인지를 위해 기반 온톨로지에 정의된 컨텍스트 조건을 통해 현 상황을 판단하는 추론을 수행하고 서비스 실행단계를 통해 서비스 조건을 만족할 시 해당되는 액티비티를 제어한다. 각 처리 단계에 따라 Java 기반의 Jena 프레임워크를 통해 처리된 온톨로지의 상태와 표현 언어를 제시하여 데이터 처리 단계를 설명하였다. 또한 사용자 친화적 서비스 정의환경은 서비스 환경 정의를 위한 장소 설정단계 및 오브젝트 설정단계, 서비스 정의를 위한 오브젝트 및 기능 선택단계 및 상황설정 단계로 이루어지며 웹 기반 UI를 통해 일반 사용자의 서비스 정의를 용이하게 하였다. 본 논문에서 제안한 메커니즘 및 서비스 정의 환경을 통해 사용자는 보다 손쉬운 서비스 정의가 가능하며 플랫폼 관점에서는 보다 정확하게 사용자의 요구를 파악하고 처리하여 다양한 환경의 상황인지 서비스 제공이 용이할 것으로 생각된다. 추후에는 Zigbee, Bluetooth, Wifi등의 다양한 인터페이스를 통해 전달되는 오브젝트의 데이터를 실제적으로 추상화 하는 방법과 다양한 환경의 서비스 확장을 위해 서로 다른 환경 온톨로지의 적용 기술에 관한 연구가 필요할 것으로 생각된다.

참고 문헌

- [1] V. Huang and M. K. Javed, "Semantic sensor

- information description and processing,” *Sensor Technologies and Applications (SENSORCOMM)*, pp.456-461, 2008.
- [2] A. P. Sheth, “Changing focus on interoperability in information systems: from system, syntax, structure to semantics,” *Interoperating Geographic Information Systems*, pp.1-30, 1999.
- [3] B. Christophe, “Semantic profiles to model the web of things,” *IEEE SKG*, pp.51-58, 2011.
- [4] B. Ostermaier, F. Schlup, and K. Romer, “WebPlug: A Framework for the Web of Things,” *Pervasive Computing and Communications Workshops*, pp.690-695, 2010.
- [5] D. Guinard and V. Trifa, “Towards the web of things: web mashups for embedded devices,” *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*, 2009.
- [6] C. Iván, F. J. Martínez, M. S. Familiar, and L. López, “Knowledge-Aware and Service-Oriented Middleware for deploying pervasive services,” *Journal of Network and Computer Applications*, pp.562-576, 2012.
- [7] 이영설, 양건모, 조성배, “상황인지 스마트 TV 서비스를 위한 온톨로지 기반 스마트 TV 환경 모델링”, *한국정보과학회학술발표회*, pp.195-197, 2012.
- [8] D. Guinard, “*A web of things application architecture-integrating the real-world into the web*,” PhD thesis No.19891, ETH Zurich, 2011.
- [9] 류중경, 김종훈, 김재권, 이정현, 정경용, “상황인식 기반의 유헬스 환경정보 서비스”, *한국콘텐츠학회논문지*, 제11권, 제7호, pp.21-29, 2011.
- [10] E. Avilés-López and J. A. García-Macías, “Mashing up the Internet of Things: a framework for smart environments,” *EURASIP Journal on Wireless Communications and Networking*, 2012.
- [11] 류중경, 정경용, 김종훈, 임기욱, 이정현, “서비스 온톨로지 기반의 상황인식 모델링을 이용한 추천”, *한국콘텐츠학회논문지*, 제11권, 제2호, pp.22-30, 2011.
- [12] S. Han, G. Lee, and N. Crespi, “Semantic context-aware service composition for building automation system,” *IEEE Transactions on Industrial Informatics*, Vol.10, No.1, pp.752-761, 2013.
- [13] C. H. Quyet, “*A hybrid semantic energy saving service for cloud-based smart homes*,” M.S thesis, School of engineering and technology thailand, unpublished, 2013.
- [14] <http://clickscript.ch>
- [15] T. Rodden, A. Crabtree, T. Hemmings, B. Koleva, J. Humble, K. Akesson, and P. Hansson, “Configuring the Ubiquitous Home,” *The Sixth International Conference on Designing Cooperative Systems*, pp.227-241, 2004.

저 자 소 개

최 환 석(Hoan-Suk Choi)

정희원



- 2009년 2월 : 국립한밭대학교 멀티미디어공학과(공학사)
- 2011년 2월 : 국립한밭대학교 멀티미디어공학과(공학석사)
- 2011년 3월 ~ 현재 : 국립한밭대학교 멀티미디어공학과 박사과정

<관심분야> : Mobility Management, Mobile Multicasting, 사물인터넷, WoT (Web of Things), Semantic Processing, Service Composition

이 준 영(Jun-Young Lee)

준회원



- 2009년 2월 : 국립한밭대학교 멀티미디어공학과(공학사)
- 2013년 3월 ~ 현재 : 국립한밭대학교 멀티미디어공학과 석사과정

<관심분야> : 사물인터넷, WoT, service composition, REST, Open API, HTML5

양 나 리(Na-Ri Yang)

준회원



- 2011년 2월 : 국립한밭대학교 경영학과(학사)
- 2013년 3월 ~ 현재 : 국립한밭대학교 멀티미디어공학과 석사과정

<관심분야> : 사물인터넷, WoT, Ontology, Context-aware Service, Service mash-up

이 우 섭(Woo-Seop Rhee)

종신회원



- 1983년 2월 : 홍익대학교(공학사)
- 1995년 8월 : 충남대학교(공학석사)
- 2003년 8월 : 충남대학교(공학박사)

- 1983년 6월 ~ 2005년 3월 : 한국전자통신연구원 팀장/책임연구원
- 2005년 3월 ~ 현재 : 국립한밭대학교 정보통신공학과 교수
- 2006년 1월 ~ 현재 : 한국ITU연구위원회 국제표준전문가 (ITU-T SG13 Editor)
- 2012년 2월 ~ 2013년 1월 : 프랑스 Institute Telecom SudParis 방문교수

<관심분야> : 사물인터넷, WoT, Semantic sensor web, Service mash-up