

유비쿼터스 환경에서 웹 서비스에 기반한 상황 인식 미들웨어의 설계

A Design of Context-Aware Middleware based on Web Services in Ubiquitous Environment

송영록*, 우요섭**

Young-Rok Song*, Yo-Seob Woo**

요약

유비쿼터스 컴퓨팅을 위한 상황 인식 기술은 수집된 컨텍스트 정보를 효과적으로 구조화하여 표현하고, 이러한 컨텍스트 정보를 이용하여 사용자의 의도를 파악한 후, 서비스를 제공하는 기술 등의 연구가 필요하다. 본 논문에서는 상황 인식 컴퓨팅을 위한 프레임워크인 WS-CAM으로 명명된 웹 서비스에 기반한 상황 인식 미들웨어를 제안한다. WS-CAM은 온톨로지 기반의 컨텍스트 모델을 사용하여 다양한 종류의 컨텍스트 정보에 대한 충분한 표현력과 추론 기능을 제공하고, 미들웨어로부터 응용 서비스로의 컨텍스트 정보 전달에 있어서 웹 서비스를 적용하여 미들웨어 독립적인 응용 개발을 이룰 수 있는 구조로 설계한다. 또한, WS-CAM의 유용성을 검증하기 위하여 유비쿼터스 컴퓨팅 환경에 기반을 둔 강의 서비스 시나리오를 기술하고, 웹 서비스 기반의 장점을 보이기 위하여 미들웨어 독립적인 시스템 확장의 예를 보인다. 강의 서비스에 적용한 WS-CAM은 도메인 환경 내의 컨텍스트 정보를 OWL 기반 온톨로지 모델로 효과적으로 표현하였고, 사용자 정의 규칙에 의하여 고수준의 컨텍스트 정보로 추론됨을 확인하였다. 또한, 웹 서비스에 의해 제공되는 다양한 웹 메서드를 통하여 컨텍스트 정보가 응용 서비스에 미들웨어 독립적으로 전달됨을 확인할 수 있었다.

Abstract

Context-aware technologies for ubiquitous computing are necessary to study the representation of gathered context-information appropriately, the understanding of user's intention using context-information, and the offer of pertinent services for users. In this paper, we propose the WS-CAM(Web Services based Context-Aware Middleware) framework for context-aware computing. WS-CAM provides ample power of expression and inference mechanisms to various context-information using an ontology-based context model. We also consider that WS-CAM is the middleware-independent structure to adopt web services with characteristic of loosely coupling as a matter of communication of context-information. In this paper, we describe a scenario for lecture services based on the ubiquitous computing e e e e e to verify the utilization of WS-CAM. We also show an example of middleware-independent system expansion to display the merits of web-based services. WS-CAM for lecture services represented context-information itodomait as OWL-based ontology model effectively, and we confirmed the information is inferred to high level context-information by user-defined rules. We also confirmed the context-information is transferred to application services middleware-independently using various web methods provided by web services.

Keywords : Context-Aware Middleware, Web Services, Ontology, Context Model

1. 서론

유비쿼터스(ubiquitous) 컴퓨팅 기술 중에 상황 인식 기술은 실세계의 특징을 표현하는 정보 기술에서 시작되었으며, 인간 세계의 의사소통과 거의 동일한 수준으로 인간과 컴퓨터 간의 의사소통이 가능하도록 하는 것이 목표이다[1]. 이러한 상황 인식 기

술은 다음과 같은 두 가지 관점에서 연구되어진다. 첫째는 환경 내에서 발생하는 컨텍스트(context) 정보의 효과적인 표현을 위한 컨텍스트 모델링 기법이고, 둘째는 적절한 응용 서비스의 제공을 위한 컨텍스트 정보의 처리 기법이다. 하지만 기존의 연구들은 Context Toolkit과 같이 표현력과 확장성이 부족한 컨텍스트 모델을 사용하던지, Gaia와 같이 미들웨어와 개별 응용 서비스 사이의 포인트 투 포인트(point-to-point) 통신에 의한 컨텍스트 정보 전달이 제한적이라는 단점이 있다[2-3]. 이에 본 논문에서는 스마트 환경에서 사용자와 주변 환경에 관한 다양한 컨텍스트 정보들을 분석하여 온톨로지(ontology)로 모델링을 하고, 사용자에게 적절한 서비스를 추론하여 제공하는 웹 서비스 기반 미들웨어 프레임워크의 개발을 목적으로 한다. 본 논문에서 제안하는 웹

*인하대학교

**인천대학교

투고 일자 : 2009. 8. 17 수정완료일자 : 2009. 10. 23

계재확정일자 : 2009. 10. 29

※ 본 논문은 인천대학교 2008년도 자체연구비 지원에 의해 연구되었음.

서비스 기반 상황 인식 미들웨어는 WS-CAM (Web Services based Context-Aware Middleware)이라 명명하고 다음과 같은 관점에서 설계한다[4].

첫째, 정보를 구조화하는데 매우 효과적이며 상호 관계성 및 부분적인 컨텍스트 정보를 쉽게 표현할 수 있는 온톨로지 기반 컨텍스트 정보 모델을 온톨로지 웹 언어(Ontology Web Language: OWL)로 설계한다. 컨텍스트 정보 모델은 스마트 환경의 도메인을 구성하는 개체에 대한 개념을 정의하고, 개체 사이의 관계, 개체들의 속성, 그 속성의 논리적인 특징 및 제약 사항 등을 OWL을 이용하여 설계한다[5]. 둘째, 저수준의 정보 수집부터 고수준의 추론 기능과 컨텍스트 정보의 일관성을 유지하고 서로 다른 서비스들 간의 내부 소통 등을 지원하는 미들웨어를 설계한다. 셋째, 미들웨어와 응용 서비스 간의 상호 의존성을 최소화하기 위한 약 결합(loosely coupling)의 특징을 갖는 웹 서비스를 적용하여 다양한 응용 서비스를 위한 오픈 인터페이스를 제공한다[6].

II. 관련 연구

2.1 상황 인식 컴퓨팅 기술

상황 인식 컴퓨팅 기술은 유비쿼터스 환경에서 주요 서비스 기능들을 구현하고 있는 다양한 서비스들을 사용자의 간섭을 최소화하면서 효율적으로 사용할 수 있도록 도와주는 미들웨어이다 [7]. 이는 사용자 환경을 지속적으로 모니터링 하고 이로부터 유익한 컨텍스트 정보를 취득한 다음 그 정보를 기반으로 사용자에게 효율적인 서비스를 능동적으로 제공하는 것을 목적으로 하고 있다. 상황 인식 컴퓨팅 기술에 관련된 대표적인 연구는 다음과 같다.

Gaia는 상황 인식을 위하여 운영 시스템을 확장한 미들웨어 구조로 인간이 생활하는 물리적인 공간에 필요한 정보를 요청하고 이용 가능한 다양한 리소스를 통해 원하는 작업을 쉽게 수행할 수 있는 액티브 공간(active space)을 실현하는 것이 목표이다 [3]. SOCAM은 상황 인식 모바일 서비스를 쉽게 개발하기 위한 프레임워크로서 상황 인식 서비스 구축을 위해 유비쿼터스 환경에서 제공되는 다양한 컨텍스트의 취득, 발견, 해석과 접근을 위한 효율적인 지원을 제공한다[8]. CoBrA는 에이전트를 기반으로 스마트 공간(smart space)에서 상황 인식 컴퓨팅을 지원하는 미들웨어로 스마트 공간 안의 모든 기기, 서비스와 에이전트의 공유 가능한 컨텍스트 모델 제공, 소스로부터의 컨텍스트 정보 취득, 컨텍스트 정보의 추론, 공유된 컨텍스트 모델의 부정확성에 대한 탐지 및 해결, 개인 정보 보호 정책에 대한 책임을 가진다[9].

이러한 상황 인식 컴퓨팅 기술은 상황 인식 미들웨어에서 응용 서비스로의 컨텍스트 정보 전달이 컴포넌트 기반의 일대일 통신을 통하여 수행되므로 응용과 미들웨어 계층이 종속적이라는 단점이 있다. 이에 응용과 미들웨어 계층이 구조적으로 독립적인 특징을 제공해 주는 웹 서비스 기반의 상황 인식 컴퓨팅 기술이 연구되고 있다[10-11].

Chari 등은 웹 서비스 기반의 상황 인식 시스템을 설계하였다. 컨텍스트 정보는 XML(eXtensible Markup Language)을 이용한 마크업 스키마 방식으로 모델링하고, (user, network, device, metadata)로 표현한다. 또한 Context Interpreter를 통하여 간단한

정보 추론을 처리한다[11].

하지만 이러한 연구들은 상황 인식 컴퓨팅에서 웹 서비스의 적용이 필요한 이유와 웹 서비스 처리 과정 등에 대해서 설명하고 있지만, 컨텍스트 모델링 방법이나 추론과 같은 상황 인식 컴퓨팅 기법에 대한 고려가 부족하다.

2.2 온톨로지 기반 컨텍스트 모델링

온톨로지 기반 컨텍스트 모델은 엄밀한(formal) 지식 표현 스킴(scheme)을 사용하여 정보를 표현하는 방법이다. 이 모델은 분산된 시스템간의 지식 공유의 목적을 달성하기 위하여 특정 도메인에서 컨텍스트 정보를 표현하는 온톨로지를 모델링하는 것에 초점을 맞추고 있다. 또한 정보를 구조화하는데 매우 효과적이며 상호 관계성 및 부분적인 컨텍스트 정보를 쉽게 표현할 수 있다[12].

기존의 컨텍스트 모델은 상황 인식 컴퓨팅 환경에서 컨텍스트 정보에 대한 상호 운용성을 제공하는데 한계를 지닌다. 그래픽 기반 상황 모델은 사용자에게 컨텍스트 정보 및 특성에 대한 개념적인 이해를 제공하지만 비정형적인 모델이기 때문에 기계 가독성이 떨어진다. 객체 지향 모델은 컨텍스트 정보의 속성과 행동을 명확하게 표현하지만 컨텍스트 정보의 논리적인 특성을 표현하는 능력이 부족하다. 하지만 온톨로지 기반 컨텍스트 모델은 일관성 있는 포맷을 제공할 뿐만 아니라 컨텍스트 정보에 대한 상속성 및 논리적 특성과 같은 의미 정보를 제공할 수 있다. 따라서 스마트 환경에서 컨텍스트에 대한 의미적인 수준의 상호 운용성을 제공할 수 있으므로, 상황 인식 프레임워크와 어플리케이션에 컨텍스트 정보 공유를 가능하게 할 뿐만 아니라 의미 정보를 바탕으로 추론 능력을 제공할 수 있다[13].

2.3 웹 서비스 기반 상황 인식 컴퓨팅 기술

웹 서비스는 네트워크상에서 서로 다른 종류의 컴퓨터들 간에 상호 작용을 하기 위한 소프트웨어 시스템으로 서비스 지향적인 분산 컴퓨팅 기술의 일종이다. 또한 웹 서비스는 표준화된 XML 메시징을 통하여 접근 가능한 네트워크 명령들의 집합을 기술한 인터페이스로서 모든 메시징에 XML이 사용되어 상호운용성이 높고 기존의 분산 컴퓨팅 기술들인 CORBA(Common Object Request Broker Architecture), DCOM(Distributed Component Object Model), RMI(Remote Method Invocation)와 비교했을 때 약 결합, 이진 부호화가 아닌 XML 유니코드 부호화 사용, 그리고 객체 지향이 아닌 메시지 지향적이라는 특징을 가지고 있다[6].

컴포넌트 결합 구조의 관점에서 기존의 상황 인식 미들웨어는 미들웨어와 개별 응용 서비스가 포인트 투 포인트(point-to-point) 통신에 의한 결합 구조로서 컨텍스트 정보의 전달이 정적이라는 단점이 있다[14]. 중앙 관리자는 상황 인식 시스템 구축 시, 다양한 플랫폼의 서비스 어플리케이션에 적합한 컨텍스트 정보의 분배 및 전송 등을 결정하고, 서비스 어플리케이션과 통신할 컴포넌트를 직접 알려주어 연결한다. 따라서 새로운 기기나 서비스가 스마트 환경에 추가되어질 경우 응용 개발자는 중앙 관리자의 도움 없이 적절한 컨텍스트 정보를 검색하여 사용하기 어렵다. 또한 컴포넌트들 간의 종속성 때문에 확장성이 부족하고 유지 보수도 어렵다. 이에 비해 웹 서비스를 이용한 컴포넌트 결합 구조는 미들웨어와 응용 서비스 간의 상호 의존성을 최소화할

수 있기 때문에 응용 서비스의 유지보수와 확장이 쉽다는 장점이 있다. 미들웨어와 서비스 어플리케이션 간의 컨텍스트 정보 전달을 웹 서비스가 담당하므로 스마트 환경 내에 새로운 기기나 서비스가 추가되어질 경우 응용 개발자는 미들웨어 내부의 정확한 지식 없이도 웹 메서드를 이용하여 필요한 컨텍스트 정보를 검색하여 사용하면 되므로 각종 서비스 어플리케이션에 유연하게 적용할 수 있다[15].

이에 본 논문에서는 미들웨어와 응용 서비스 간의 상호 의존성을 최소화할 수 있는 웹 서비스와 정보의 구조화에 매우 효과적인 온톨로지 기반의 컨텍스트 모델링 기법을 이용한 상황인식 미들웨어 시스템을 설계하고자 한다.

III. WS-CAM 시스템

3.1 WS-CAM의 구조

본 논문에서 제안하는 웹 서비스 기반 상황 인식 미들웨어 프레임워크인 WS-CAM의 구조는 그림 1과 같다.

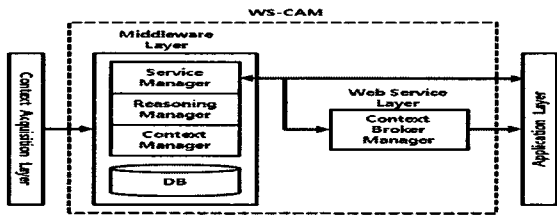


그림 20. WS-CAM의 구조.
Fig. 1. The WS-CAM architecture.

WS-CAM은 컨텍스트 취득 계층에서 전해지는 정보의 처리, 추론, 검색, 그리고 서비스 관리 등을 위한 미들웨어 계층과, 서비스 제공 계층과 미들웨어 사이의 효율적인 정보 전달을 위한 웹 서비스 계층으로 구성된다. 미들웨어 계층은 컨텍스트 정보를 취득하여 온톨로지 모델로 표현하는 Context Manager, 온톨로지 정보를 이용하여 고수준의 정보를 추론하는 Reasoning Manager, 스마트 환경 내의 사용자에게 적절한 서비스를 제공하기 위한 정보의 효율적인 분배를 담당하는 Service Manager, 컨텍스트 정보와 추론을 위한 사용자 정의 규칙을 저장하는 데이터베이스로 구성되고, 웹 서비스 계층은 응용 계층의 미들웨어 독립성을 위한 Context Broker Manager로 구성된다. WS-CAM은 Context Acquisition Layer를 통하여 취득되는 스마트 환경의 컨텍스트 정보를 Context Manager에서 취합하여 컨텍스트 모델로 변경하고 Reasoning Manager에 의해 응용 서비스를 위한 고수준의 컨텍스트 정보로 생성한다. Service Manager는 컨텍스트 정보를 분류하여 Context Broker Manager에 의해 웹 서비스를 이용하여 응용서비스에 제공한다.

3.2 Context Manager

상황 인식 미들웨어는 사용자에게 대한 지능적인 서비스 제공을 위하여 다양한 컨텍스트를 효과적으로 관리할 수 있어야 한다. 이러한 컨텍스트는 스마트 환경 내에 존재하는 다양한 센서들로부터 취득되어지거나, 또는 사용자의 스케줄 정보나 주변 지역의 날씨와 같이 웹에 존재하는 다양한 정보 서버들로부터 취득되어

질 수 있다. Context Manager는 위와 같이 다양한 소스로부터 얻어지는 사용자와 사용자 주변 환경에 대한 정보의 취득, 처리, 표현, 그리고 저장을 위한 기능의 관리를 담당한다. Context Manager는 그림 2와 같이 다양한 소스로부터 얻어지는 정보를 미들웨어에 안정적으로 공급하는 역할을 하는 Context Provider, 센서로부터 전달되는 저수준의 데이터를 사용자에게 이해하기 쉬우면서 직관적인 의미의 정보로 변경시키는 역할을 하는 Context Interpreter, 컨텍스트 정보를 온톨로지 모델로 표현하는 역할을 하는 Context Generator와 Context DB로 구성된다.

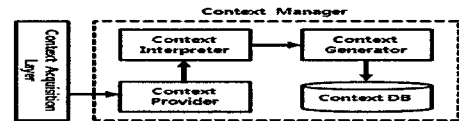


그림 21. Context Manager의 구조.
Fig. 2. The Context Manager architecture.

표 1. WS-CAM 컨텍스트 모델의 구성요소

Table 1. The elements of WS-CAM context Model

최상위	하위 클래스	속성	관계
Person	none	name	own
		role	locatedAt
	act ...
Place	Indoor	Room	temp.
		Etc_Place	noise
	Outdoor	humid.	isPartOf
	contains
	subClassOf
Object	Device	Electric	isOwnedBy
		Computing	locatedAt
	subClassOf

Behavior	Inferred	Scheduled	startTime
		Affair	endTime
		Rest	...
	locatedAt
	isActedBy

본 논문에서는 제안하는 WS-CAM 온톨로지는 OWL-DL 언어로 Protégé 4를 이용하여 구성하였다. WS-CAM 온톨로지는 표 1과 같이 도메인을 구성하는 Person(사람), Place(장소), Behavior(행동), 그리고 Object(기기)라는 4개의 최상위 핵심 클래스들과 하위 클래스들, 그리고 그들의 관계를 표현하는 속성들의 정의로 이루어진다.

각각의 클래스는 모델이 되는 일반적인 개념의 속성들과 다양한 하위 클래스를 정의한다. 클래스들 간의 관계를 표현하는 속성은 owl:objectProperty로 정의하고, 클래스 인스턴스와 연결되는 데이터 값과의 관계를 표현하는 속성은 owl:datatypeProperty로 정의한다. Person 클래스는 스마트 환경 내에 존재하는 사람에 대한 일반적인 속성으로 정의한다. WS-CAM 온톨로지는 owl:datatypeProperty로 정의된 각각의 사람의 이름, 전화 번호, 환경 내에서의 역할 정보 등 특정한 사람의 신원 정보를 요구한다. 또한 owl:objectProperty로 정의된 own, locatedAt, 그리고 act라는 속성에 의해 특정한 기기를 소유하고, 어떤 장소에 존재하며, 그리고 특정한 행동을 하는 중이라는 정보를 표현할 수 있

다. Place 클래스는 실내와 실외를 나타내는 IndoorPlace와 OutdoorPlace를 자식 클래스로 갖는다. 그 중 상황 인식 서비스의 대부분이 수행될 IndoorPlace 클래스는 다시 Room과 Etc_Place라는 자식 클래스를 갖는다. Room과 Etc_Place 클래스는 disjointWith 속성에 의하여 같은 장소가 될 수 없음을 표시한다. 또한 isPartOf라는 속성에 의해 IndoorPlace 클래스의 구성요소임을 표시하고 IndoorPlace 클래스는 isPartOf의 owl:InverseOf 속성인 hasPart 속성에 의해 Room과 Etc_Place를 포함하고 있음을 표현한다. Place 클래스는 locatedAt 속성의 owl:InverseOf 속성인 contains 속성에 의하여 장소 내에 특정한 기기와 사람이 포함되어 있음을 표시한다. Place 클래스와 하위 클래스들은 상속 관계에 의하여 특정한 기기나 사물의 존재 범위를 더욱 세밀하게 표현 할 수 있다. 그림 3은 OWL-DL 구문으로 표현된 WS-CAM 온톨로지의 일부분이다.

```

<owl:Class rdf:ID="IndoorPlace">
  <rdfs:subClassOf rdf:resource="#Place"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasPart"/>
      <owl:someValuesFrom rdf:resource="#Room"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasPart"/>
      <owl:someValuesFrom rdf:resource="#Etc_Place"/>
    </owl:Restriction>
  </rdfs:subClassOf>

```

그림 3. WS-CAM 온톨로지의 예

Fig. 3. An example of WS-CAM ontology represented in OWL-DL

3.3 Reasoning Manager

컨텍스트 정보의 추론은 다양한 소스로부터 획득된 원시 데이터들을 융합하여 고수준의 컨텍스트 정보를 생성시키는 상황 인식 컴퓨팅의 가장 중요한 기술 중 하나이다. Reasoning Manager는 WS-CAM 온톨로지를 이용하여 위치 정보, 환경 정보 등의 추론을 관리하는 역할을 한다. Inference Engine은 그림 4와 같이 Context DB에 저장된 컨텍스트 정보와 Rule DB에 저장된 추론 규칙을 이용하여 현재의 상황을 추론하여 Service Manager에게 제공하는 역할을 한다.

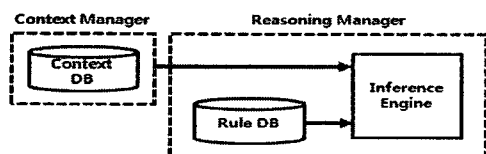


그림 4. Reasoning Manager의 구조

Fig. 4. The Reasoning Manager architecture

WS-CAM은 추론을 위한 사용자 정의 추론 규칙을 작성하기 위하여 온톨로지 기반의 컨텍스트 모델에서 공통적으로 사용하는

RDF(Resource Description Framework) 트리플 형태인 (주어, 동사, 목적어)에 기반을 둔 (Instance, Property, Value)의 형태로 컨텍스트 정보를 표현한다. WS-CAM의 컨텍스트 정보 트리플은 주어와 목적어가 각 클래스의 인스턴스로 구성되고, 동사는 인스턴스 사이의 관계를 나타내는 프로퍼티로 구성된다. 예를 들어 (Leo, locatedAt, Room_404)는 "404호 강의실에 Leo가 존재한다"라는 정보를 간단히 표현할 수 있다. 또한 이러한 정보는 AND, OR, 그리고 NOT 연산자들과 연결되어 합성 컨텍스트 정보를 표현할 수 있다. 다음은 사용자 정의 추론 규칙에 의하여 특정 정보를 추론하는 예이다.

- (a) 사용자의 의도를 파악하기 위한 추론 예
- ① Person("Leo") locatedAt Room("302호")
 - ② Room("302호") contains Lecture("Java") has startTime("2009-06-19T13:00:00")
 - ③ Person("Leo") has Property role("lecturer")
 - ④ ← ① + ② + ③ : Person("Leo")은 Room("302호")에 예정된 Lecture("Java") 강의를 위해 강의실에 도착했다.

- (b) 사용자의 의도에 대한 서비스 제공을 위한 추론 예
- ⑤ Room("302호") contains Computer("com_302") has status("off")
 - ⑥ Room("302호") contains Projector("pro_302") has status("off")
 - ⑦ Room("302호") has Property lighting("veryBright")
 - ⑧ ← ④ + ⑤ + ⑥ + ⑦ : Person("Leo")의 Lecture("Java") 강의를 위하여 Room("302호")의 Computer("com_302")와 Projector("pro_301")의 status("on") 으로 하고 lighting("littleDark") 으로 하라.

3.4 Service Manager

WS-CAM의 Service Manager는 컨텍스트 정보 전달을 위한 모듈로서 서비스 제공 계층과의 인터페이스 역할을 담당한다. Service Manager는 스마트 환경 내의 컨텍스트 정보를 입력받아 각각의 서비스를 위한 적절한 정보를 분배하여 제공하는 기능을 수행한다. 또한 특정한 응용 서비스가 자신이 원하는 현재의 컨텍스트 정보를 요청할 때, 컨텍스트 정보를 검색하여 제공하는 역할을 수행한다. 컨텍스트 정보 중에는 특정한 대상의 위치나 환경 내의 정보 등, 대부분의 응용 서비스에서 필요한 공통 컨텍스트(common context)가 존재한다. Service Manager는 이러한 공통 컨텍스트 정보를 웹 서비스를 통해 제공하기 위하여 데이터베이스로 관리하는 기능을 갖고 있다. 그림 5는 Service Manager의 구성도이다.

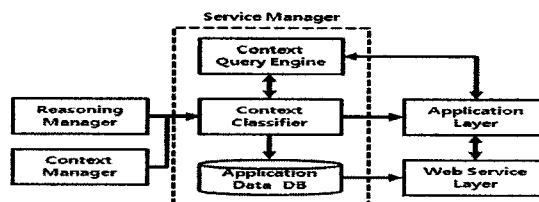


그림 5. Service Manager의 구조

Fig. 5. The Service Manager architecture

먼저 Context Classifier는 Context Interpreter에 의해 전달되는 일차 변환된 컨텍스트 정보와 Reasoning Manager에서 추론

규칙에 의해 추론된 컨텍스트 정보를 미리 정의된 개별 응용 서비스들의 요구에 맞게 분배하여 제공하는 역할을 한다. 또한 미들웨어 독립적인 응용 개발을 지원하기 위해 사용하는 웹 서비스를 위하여, 공통 컨텍스트 정보를 분류하여 Application Data DB에 저장한다.

상황 인식 시스템의 응용 서비스 계층은 일반적으로 상황 인식 미들웨어에서 전해지는 컨텍스트 정보를 바탕으로 서비스를 제공한다. 하지만 응용 서비스가 원하는 정보를 미들웨어에 요청하는 경우가 발생할 수 있다. Context Query Engine은 RDQL (RDF Data Query Language)을 이용하여 응용 서비스로부터 요청되는 질의를 처리하여 적절한 컨텍스트 정보를 검색하여 제공하는 기능을 수행한다. RDQL은 Jena RDF 모델을 위한 질의 언어로서, SQL과 비슷한 문법 구조를 갖기 때문에 쿼리의 작성과 이해가 쉽고, RDF 트리플의 주어, 동사, 목적어를 질의를 통해 알아낼 수 있다[16].

3.5 Context Broker Manager

Context Broker Manager는 WS-CAM에서 처리된 컨텍스트 정보를 웹 서비스를 이용하여 응용 서비스에 전달하는 역할을 한다. 상황 인식 미들웨어에 관한 기존의 연구들은 중앙 관리자의 시스템 초기 설정 시, 응용 서비스들이 커뮤니케이션 컴포넌트를 직접 알려주어 연결시켜주는 방법만을 고려하였다. 이것은 새로운 서비스를 도메인 내에서 제공하려 할 때, 응용 개발자가 중앙 관리자의 도움 없이 미들웨어와의 연결을 확립시킬 수 없다는 단점이 존재한다. 이에 본 논문에서 제안하는 WS-CAM은 응용 서비스에서 필요한 컨텍스트 정보들을 미들웨어 독립적인 응용 개발을 이룰 수 있는 구조인 웹 서비스를 이용하여 시스템의 유연한 확장성을 고려한다.

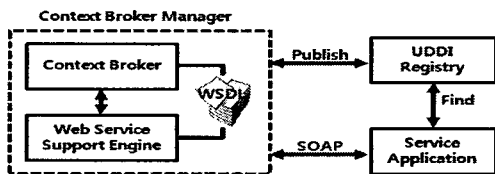


그림 6. Context Broker Manager의 구조
Fig. 6. The Context Broker Manager architecture

표 2. 웹 서비스를 위한 웹 메서드의 예

Table 2. An example of web methods for Web Services

웹 메서드	입력 파라미터	출력 값	설명
UserConfirm_Service	room_num : string	person_id : string[]	해당 공간의 사용자 확인
UserInfo_Service	person_id : string	name : string role : string phone_num : string	사용자의 개인정보 확인
UserLocation_Service	person_id : string	room_num : string	특정 사용자 위치 확인
DeviceInfo_Service	room_num : string	com_status : string pro_status : string	공간의 장치 상태 정보 확인
EnvInfo_Service	room_num : string	temperature : string lighting : string humidity : string	해당 공간의 환경정보 확인

Context Broker Manager의 Web Services Support Engine은 기존의 연구[4]와 비교하여 웹 서비스 플랫폼 관리를 위해 확장된 모듈로서, 그림 6과 같이 Application Data DB에 존재하는 컨텍스트 정보를 취득하기 위한 표 2에서 보인 웹 메서드의 파라미터, 데이터 타입, 전송 프로토콜과 웹 서비스의 종점 URL 등을 명시한 WSDL(Web Service Description Language)문서를 생성한다. 또한, 웹 메서드에 관한 정보를 UDDI(Universal Description, Discovery and Integration) Registry에 공개(publish) 한다. 새로운 서비스를 도메인에 확장하려는 응용 개발자는 자신의 서비스에 필요한 정보를 UDDI Registry에서 검색(find)하여 SOAP(Simple Object Access Protocol) 메시지를 이용하여 WSDL 문서로 구성된 정보를 바인딩(bind)한다. 따라서 응용 개발자는 미들웨어 독립적으로 서비스에 필요한 정보를 호출하여 사용할 수 있다[17-19].

IV. 실험 및 평가

지금까지 설명한 WS-CAM은 다양한 응용 환경에서 상황 인식 처리를 위하여 사용되어질 수 있다. 온톨로지를 이용하여 컨텍스트 정보를 표현하므로 다양한 응용 환경에 알맞은 컨텍스트 모델을 최소한의 수정으로 정의할 수 있고, 또한 웹 서비스를 이용하여 스마트 환경의 정보를 제공하므로 시스템의 변경이나 확장 시, 유연하게 적용할 수 있다는 장점이 있다. 이와 같은 WS-CAM의 특징을 검증하기 위해 본 논문에서는 유비쿼터스 컴퓨팅 환경에 기반을 둔 강의 서비스를 구성하였다.

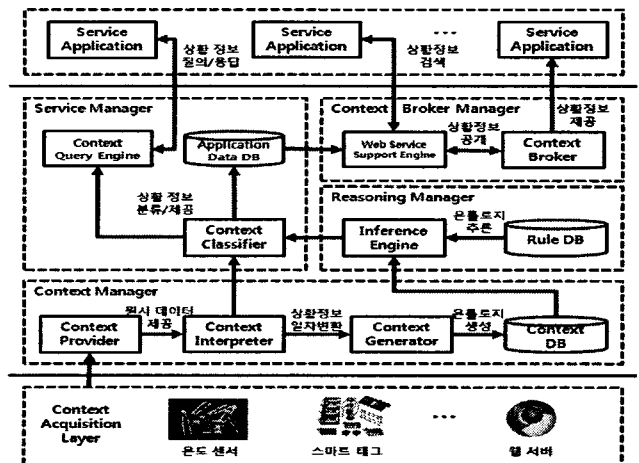


그림 7. WS-CAM 프레임워크의 데이터 관계도

Fig. 7. The data flow diagram of WS-CAM framework

WS-CAM은 그림 7과 같이 RFID 태그를 부착하고 있는 강사가 강의실에 들어오면 RFID 센서는 강사의 정보를 Context Provider에 보낸다. 또한 Context Provider는 강의실의 환경 정보 센싱 값을 주기적으로 입력받아 이들 정보를 Context Interpreter에 제공한다. Context Interpreter는 현재 강의실의 온도나 조도와 같은 센싱 값들을 간단한 형식의 컨텍스트 정보로 일차 변환하여 Context Generator에 전송하고, 이와 같은 정보들을 필요로 하는 응용들을 위하여 해당 정보를 Context Classifier에 전송한다. Context Generator는 미리 정의된 컨텍스트 모델에 맞게 컨텍스트

트 정보 온톨로지를 생성하여 Context DB에 저장하고 Inference Engine에 전달한다. Inference Engine은 도메인 온톨로지와 Rule DB의 사용자 정의 룰을 이용하여 컨텍스트 정보를 추론하고, RDF/XML 추론 모델을 생성하여 Context Classifier에 전달한다. Context Classifier는 다양한 응용들에 제공할 컨텍스트 정보들을 분류하여 서비스들에게 제공하고, 웹 서비스를 통해 제공할 정보를 Application Data DB에 저장한다.

4.1 강의 서비스를 위한 Context Generator

WS-CAM은 강사의 등장을 인식하면 강사의 개인 정보와 강의실 내의 환경 관련 정보를 Context Interpreter에 제공한다. Context Interpreter는 원시 데이터로 제공된 온도나 습도와 같은 환경 정보를 일차 변환하고 Context Generator에 제공한다. Context Generator는 Context Interpreter에서 전달되는 컨텍스트 정보를 WS-CAM 온톨로지 모델에 부합하는 OWL 온톨로지 문서로 생성한다. 그림 8의 OWL 온톨로지 문서는 'Room_404'가 isPartOf 속성에 의하여 '#제1공학관'에 속한 강의실이고, contains 속성에 의하여 'Computer_404', 'Projector_404', 'Aircon_404'와 'Person_4'가 존재함을 알 수 있고, 또한 온도와 습도와 같은 환경 정보도 표현하고 있다.

```
<Room rdf:ID="Room_404">
  <isPartOf rdf:resource="#제1공학관"/>
  <contains>
    <Computer rdf:ID="Computer_404">
      <status rdf:datatype="XMLSchema:string">off </status>
    </Computer>
    <Lecture rdf:ID="Lecture_JAVA">
      <endTime rdf:datatype="XMLSchema:dateTime">
        2009-06-19T15:50:00</endTime>
      <startTime rdf:datatype="XMLSchema:dateTime">
        2009-06-19T13:00:00</startTime>
    </Lecture>
    <Person rdf:ID="Person_4">
    </contains>
    <humidity rdf:datatype="XMLSchema:string">
      littleHumidity</humidity>
  </Room>
```

그림 8. 환경 정보에 관한 온톨로지 문서의 예
Fig. 8. An Example of WS-CAM Ontology for Environment Information

4.2 강의 서비스를 위한 Inference Engine

Inference Engine은 OWL-DL 기반의 자동화된 추론과 사용자 정의 규칙 기반의 추론을 통하여 고수준의 컨텍스트 정보를 추론하는 역할을 한다.

WS-CAM은 온톨로지 추론을 위하여 Jena 룰 엔진[20]을 이용하고, 추론 규칙을 작성하기 위하여 온톨로지 기반의 컨텍스트 모델에서 공통적으로 사용하는 RDF 트리플 형태인 (주어, 동사, 목적어)에 기반을 둔 (Instance, Property, Value)의 형태로 컨텍스트 정보를 표현한다. Inference Engine은 Context Generator에 의해 제공되는 그림 8과 같은 컨텍스트 정보 온톨로지를

DL(Description Logic) 기반의 자동화된 추론과 사용자 정의 규칙 기반 추론을 통하여 그림 9와 같이 RDF/XML 모델로 표현한다. 또한 Inference Engine은 컨텍스트 정보 온톨로지와 사용자 정의 룰을 이용하여 사용자의 위치 정보 추론, 사용자가 위치한 장소의 환경 관련 추론, 그리고 사용자의 상태나 의도를 파악하여 서비스를 제공하기 위한 추론을 진행한다. 표 3은 Person(Leo)의 현재 위치와 개인 정보를 파악한 후, 해당 공간에 예정된 Lecture(Java)의 강의를 위하여 Room(404호)에 입장하였음을 판단하는 예이다.

```
<rdf:RDF>
  <rdf:Description about="Ontology65.owl#Room_404">
    <contains>
      <rdf:Description about="Ontology65.owl#Person_4">
        <hasInfo> lecturer </hasInfo>
      </rdf:description>
      <rdf:Description about="Ontology65.owl#Lecture_JAVA">
        <hasInfo> 2009-06-19T13:00:00 </hasInfo>
      </rdf:description>
    </contains>
  </rdf:Description>
</rdf:RDF>
```

그림 9. RDF/XML 추론 모델의 예
Fig. 9. An example of RDF/XML inference model

표 3. 사용자의 의도 파악을 위한 추론 결과의 예
Table 3. An example of inference result for user intention

웹 메서드	Leo의 의도를 파악하기 위한 추론
사용자 정의 룰	(Person(?x), locatedAt, Room(?y)) AND (Room(?y), contains, Lecture(?z)) AND (Person(?x), hasInfo, role("lecturer")) AND (Lecture(?z), hasInfo, startTime("09-06-19T13:00:00")) => (Person(?x), act, Lecture(?z))
의미	Person("Leo")은 Room("404호")에 예정된 Lecture("Java") 강의를 위해 강의실에 도착했다.

4.3 미들웨어 독립적인 시스템 확장의 예

WS-CAM은 미들웨어 독립적인 시스템의 확장을 위하여 웹 서비스를 이용하여 컨텍스트 정보를 제공한다. Context Broker Manager는 응용 서비스들이 일반적으로 사용하는 공통 컨텍스트 정보를 제공하는 표2와 같은 웹 메서드들을 WSDL 문서로 생성하여 웹 서비스를 위해 UDDI에 등록한다.

예를 들어, 404호 강의실에 새로운 에어컨을 설치하고 관련 서비스를 제공하려는 응용 개발자는 UDDI에서 특정 공간의 실내 온도를 응답 받을 수 있는 기능을 검색하여 TemperatureService()라는 웹 메서드의 정보를 파악하고, Context Broker Manager에 해당 웹 메서드에 관한 WSDL 문서를 요청하여 제공받는다. 응용 개발자는 TemperatureService() WSDL 문서를 토대로 그림 10과 같은 SOAP 메시지를 Context Broker Manager에 요청한다. Context Broker Manager는 요청 SOAP 메시지에 해당하는

404호 강의실의 현재 실내 온도를 그림 11과 같은 응답 SOAP 메시지를 이용하여 응용 서비스에 정보를 제공한다. 응용 개발자는 제공하려는 에어컨 서비스에 필요한 정보를 미들웨어 내부 컴포넌트와의 연결 없이, 웹 서비스를 통해서 얻을 수 있으므로, 미들웨어 독립적으로 시스템의 확장을 이룰 수 있다.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="XMLSchema-instance"
  xmlns:xsd="XMLSchema"
  xmlns:soap="/soap/envelope/">
  <soap:Body>
    <Request xmlns="http://211.119.247.101/Temperature">
      <room_num type="xsd:string">404호</room_num>
    </Request>
  </soap:Body>
</soap:Envelope>
```

그림 10. TemperatureService()를 위한 요청 SOAP 메시지의 예

Fig. 10. An example of request SOAP message for TemperatureService()

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="XMLSchema-instance"
  xmlns:xsd="XMLSchema"
  xmlns:soap="/soap/envelope/">
  <soap:Body>
    <TemperatureResponse xmlns="
      http://211.119.247.101/Temperature">
      <return type="xsd:string">littleHot</return>
    </TemperatureResponse>
  </soap:Body>
</soap:Envelope>
```

그림 11. TemperatureService()를 위한 응답 SOAP 메시지의 예

Fig. 11. An example of response SOAP message for TemperatureService()

4.4 평가

WS-CAM은 웹 서비스에 의한 원격 메시지를 이용하여 다양한 응용 서비스를 위한 오픈 인터페이스를 제공하였다. 이는 기존의 연구들이 미들웨어와 응용 서비스간의 강 결합 컴포넌트 구조로 인하여 컨텍스트 정보의 전달이 제한적이고 시스템의 확장성이 부족하다는 단점을 극복한 것이다[2-3]. 또한, WS-CAM은 OWL DL 기반의 자동화된 추론과 Jena 를 엔진 기반의 사용자 정의 규칙을 통한 고수준의 추론 기능을 제공하였다. 이는 웹 서비스를 적용한 Charri의 연구가 XML 기반의 마크업 스키마 모델을 사용하여 간단한 수준의 추론 기능만을 제공한다는 단점을 극복한 것이다[11]. 기존의 연구[4]와 비교한 WS-CAM은 보다 정확하고 다양한 추론 매커니즘과 서비스 제공을 위하여 사용자 정의 추론 규칙과 웹 서비스를 위한 웹 메시지를 다양한 환경과 시나리오를 고려하여 정의하였고, 웹 서비스 플랫폼의 보다 효율적인 관리를 위해 시스템을 확장하였다.

V. 결론

본 논문에서는 사용자와 주변 환경에 관한 컨텍스트 정보를 이용하는 웹 서비스 기반의 상황인식 미들웨어로서 WS-CAM을 설계하였다. 제안된 WS-CAM은 온톨로지 기반의 컨텍스트 모델을 정의하여 컨텍스트 정보의 저장, 공유, 그리고 추론에 활용하였다. WS-CAM은 온톨로지 표준 언어인 OWL을 이용하여 DL의 추론 규칙을 이용한 자동화된 추론 기능을 제공하였고, Jena 를 엔진 기반의 사용자 정의 규칙 추론에 대응하였다. 또한 특정 도메인에 종속적인 모델이 아닌, 일반적인 스마트 환경의 도메인을 구성하는 개체와 개체 사이의 관계, 개체들의 속성과 논리적인 특징 등을 정의하여 컨텍스트 모델을 정형화하였기 때문에 각기 다른 도메인에서 최소한의 수정으로 컨텍스트 모델을 재사용할 수 있다는 장점을 제공할 수 있었다.

WS-CAM은 센서로부터 수집된 정보를 컨텍스트 모델에 부합하는 온톨로지로 생성하고, 사용자 정의 규칙을 통한 고수준의 추론 기능을 제공하였다. 이에 따라 온톨로지에 의해 일차원적인 RDF 트리플 형태로 표현된 컨텍스트 정보들이 논리 연산을 통하여 사용자의 의도와 같은 고수준의 새로운 컨텍스트 정보로 추론이 가능하였다. 또한 WS-CAM은 RDQL을 이용한 컨텍스트 쿼리 기능을 제공하여 응용 서비스가 단순히 미들웨어로부터 정보를 전달받을 뿐만 아니라 필요한 정보를 검색하여 사용할 수 있는 기능을 제공하였다.

기존의 상황 인식 미들웨어들은 컨텍스트 정보 전달이 수동적이며 제한적인 단점이 존재하기 때문에 WS-CAM은 응용 서비스를 위하여 공통 컨텍스트 정보 제공 기능 APIs를 웹 서비스를 이용하여 제공하였다. 이에 따라 WS-CAM을 적용한 스마트 환경에서 응용 개발자는 미들웨어 독립적으로 서비스의 보수, 유지나 확장 등이 가능하다.

본 논문에서 제안한 온톨로지 기반 웹 서비스를 이용한 상황 인식 미들웨어를 보다 다양한 스마트 환경에 적용하기 위해서는 다음과 같은 연구가 보완되어야 한다. 첫째, 보다 높은 수준의 온톨로지 재사용성을 제공하여야 한다. WS-CAM 온톨로지는 일반적인 도메인을 구성하는 요소들로 정의하였지만, 어느 정도 시물레이션을 위한 강의 도메인에 종속적인 것이 사실이기 때문에 최대한의 재사용성을 제공하기 위한 연구가 필요하다. 둘째, 효율적인 추론을 위한 보다 다양하고 정확한 규칙의 정의가 필요하다. 또한 스마트 환경에서 발생하기 쉬운 문제 중 하나인 개인 정보 보안에 관한 연구가 필요할 것이다.

참고문헌

- [1] B. N. Schilit, etc, "Context-Aware Computing Applications." IEEE Workshop on Mobile Computing Systems and Applications, 1994, pp. 85-90.
- [2] A. K. Dey and G. D. Abowd, "The Context Toolkit: Aiding the Development of Context Enabled Applications" CHI'99, 1999, pp15-20.
- [3] M. Román, etc, "Gaia : A Middleware Infrastructure to Enable Active Spaces." IEEE Pervasive Computing, 2002, pp.74-83.

[4] 송영록, "웹 서비스를 이용한 온톨로지 기반 유비쿼터스 상황 인식 미들웨어 설계", 인천대학교 박사학위논문, 2009.

[5] OWL, <http://www.w3.org/2004/OWL/>.

[6] DAML Services Coalition, "DAML-S: Web Service Description for the Semantic Web." First International Semantic Web Conference, 2002, pp. 348-363.

[7] W. Qin., etc, "CAMPS: A Middleware for Providing Context-Aware Services for Smart Space." Advances in Grid and Pervasive Computing, 2006, pp. 644-653.

[8] T. Gu, etc, "A Service-Oriented Middleware for Building Context-Aware Services" Journal of Network and Computer Applications, Vol. 28, 2004, pp. 1-18.

[9] H. Chen, etc, "Intelligent Agents Meet the Semantic Web in Smart Spaces." IEEE Internet Computing, 2004, pp. 2-12.

[10] G. Gehlen and G. Mavromatis, "Mobile Web Services based Middleware for Context-Aware Applications" Proceedings of the 11th European Wireless Conference, Vol. 2, 2005, pp. 784-790.

[11] T. Chaari, etc, "Design of Context-Aware Applications based on Web Services." UniversitàCáFoscari di Venezia reference: Technical Report CS-2004-5, pp. 1-6.

[12] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition, Vol. 5, 1993, pp. 199-220.

[13] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey." In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004, pp. 1-6.

[14] 정국상. "미들웨어 독립적인 응용개발을 위한 컨텍스트 인지 프레임워크", 전남대학교 박사학위 논문, 2007.

[15] 송영록 등, "무선 센서네트워크를 이용한 교량 안전진단 시스템에서의 XML을 이용한 데이터 처리", 한국멀티미디어학회, 제9권-2호, 2006, pp. 40.

[16] A. Seaborne. "RDQL - A Query Language for RDF." <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>.

[17] D. Booth and C. K. Liu, "Web Services Description Language(WSDL) Version 2.0 Part 0: Primer." <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/>.

[18] UDDI, http://www.uddi.org/pubs/uddi_v3.htm.

[19] N. Miltra and Y. Lafon, "SOAP Version 1.2 Part 0: Primer." <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>.

[20] Jena - A Semantic Web Framework for Java, <http://jena.sourceforge.net/>.



송영록 (Young-Rok Song)

正會員

2001년 인천대 정보통신공학과 (공학사)
2003년 인천대 정보통신공학과 (공학석사)
2009년 인천대 정보통신공학과 (공학박사)

2009년 ~ 현재 인하대 정보전자공동연구소 연구교수

※주관심분야 : 시멘틱 웹, 인공지능, 신호처리



우요섭 (Yo-Seob Woo)

1986년 한양대 전자통신공학과 (공학사)
1988년 한양대 전자통신공학과 (공학석사)
1992년 한양대 전자통신공학과 (공학박사)

1992년 ~ 현재 인천대학교 정보통신공학과 교수

※주관심분야 : 한국어정보처리, 시멘틱 웹, 인공지능