韓國컴퓨터情報學會 論文誌
第15卷 第10號, 2010. 10.

2010-15-10-4-3

유비쿼터스 센서 네트워크 환경에서 온톨로지와 규칙을 이용한 지능형 서비스 미들웨어

박 종 현*, 강 지 훈**

An Intelligent Service Middleware Using Ontology and Rule in Ubiquitous Sensor Network Environments

Jong-Hyun Park*, Ji-Hoon Kang**

요 약

USN 미들웨어에 대한 몇몇 연구가 이미 존재하지만 아직까지 미들웨어 표준이 정의되지는 않은 상태이므로 어떠한 미들웨어가 가장 효과적일지는 알 수 없다. 특별히 본 논문에서는 미들웨어의 많은 기능들 가운데 센서 네트워크 환경에서 지능형 서비스를 제공하는 것을 목표로 한다. 센서를 기반으로 하는 많은 응용들은 단순히 센서의 값을 취하거나 모니터링 하기 위해서 센서를 사용하기보다는 이를 기반으로 상황을 인지하고 지능형 서비스를 제공하기를 원한다. 그러나 현재 센서 네트워크 미들웨어 수준에서 지능형 서비스를 지원하기 위한 필요성은 인식하고 있으나, 구체적으로 어떻게 지능형 서비스를 제공할 것 인지에 대한 연구는 찾아보기 힘들다. 그러므로 본 논문에서는 범용의 USN 미들웨어에서 지능형 서비스를 지원하기 위한 컴포넌트를 설계하고 구체적인 서비스지원 방법을 제안한다. 이러한 목적을 위하여 본 논문에서는 USN 미들웨어 응용에서 제공하는 서비스와 센서 네트워크를 정의하고 두 개념들을 연계하기 위해서 센서-서비스 온톨로지를 제안한다. 센서-서비스 온톨로지는 저수준의 정보로부터 고수준의 정보를 추론하기 위하여 사용된다. 또한 응용의 상황 정보를 추론에 반영하기 위하여 규칙을 이용한 추론 방법을 제안한다. 논문은 제안한 지능형 서비스 미들웨어는 프로토타입을 구현하고 성능평가를 통하여 다양한 응용을 위해서 사용될 수 있음을 보인다.

Abstract

There are some of the studies on sensor middleware. However the standard middleware has not yet been defined. Especially, this paper focuses on the processing an intelligent service of the main functions of middleware. Several applications in the sensor network environment support not only monitoring services, but also sensor-based context-awareness and intelligent services based on sensors. However, the previous studies about USN middleware only mentioned the need for intelligent service and did not discuss the architecture and method for supporting the intelligent service in detail. Therefore this paper designs a USN middleware for providing intelligent services

• 제1저자 : 박종현 교신저자 : 강지훈
• 투고일 : 2010. 07. 14, 심사일 : 2010. 08. 03, 게재확정일 : 2010. 08. 12.
* 큐슈대학교 정보기반연구센터 방문연구원 ** 충남대학교 전기정보통신공학부 교수

www.kci.go.kr

and proposes the method for processing the services. For this purpose, this paper proposes the Sensor-Service ontology to define the concept of services and sensors for USN applications and the relationship between them. The Sensor-Service ontology is used to infer high-level information from low-level information. To apply a variety of environmental context to intelligent services, the paper uses the rule-based reasoning. This paper implements the proposed intelligent service middleware as a prototype and then shows that the middleware can be used for a variety of USN applications through the performance evaluation.

▸ Keyword : 유에스엔 미들웨어(USN Middleware), 지능형 서비스(Intelligent Service), 서비스추론(Service Reasoning), 지능형 센서(Intelligent Sensor)

Ⅰ. 서 론

센서를 기반으로 사용자에게 다양한 고급의 서비스를 제공하기 위한 유비쿼터스 센서 네트워크 (Ubiquitous Sensor Network: USN) 환경은 이질적인 수많은 센서 장치 및 노드들로 구성되며 네트워크들 마다 다양한 특성을 갖는다. 이러한 이종의 센서 네트워크들을 기반으로 다양한 서비스를 제공하기 위한 응용들 또한 그 요구사항이 서로 상이하다. 그러므로 센서 네트워크와 USN 응용 서비스 시스템 사이에서 독립성을 유지하며 통합을 유연하게 이루어지도록 하는 역할을 수행하는 USN 미들웨어는 반드시 필요하다. 이러한 요구사항에 발맞추어 국내외 적으로 USN 미들웨어를 개발 중에 있지만 아직까지 USN 환경에 최적으로 동작할 수 있는 미들웨어 표준은 정의되지 않은 상태이다. 그러므로 USN 미들웨어를 개발하고 실제 센서 네트워크 환경에 적용하기 위한 연구는 반드시 수행되어야 할 연구들 가운데 하나이다.

다양한 네트워크와 응용들을 연계하는 USN 미들웨어의 특성상, 미들웨어는 센서 데이터 관리, 메타데이터 관리, 보안 관리 등과 같은 많은 기능들이 필요하며 지능형 서비스 관리 역시 중요한 기능들 가운데 하나이다. 특별한 경우를 제외하고 대부분의 센서 네트워크 기반 응용들은 단순히 센서의 값만을 취하거나 모니터링 하는 것 이외에 센서 데이터로부터 상황을 추론하고 연관되는 서비스를 제공하고자 한다. 본 논문에서는 이러한 상황 추론과 이를 기반으로 하는 서비스들을 지능형 서비스로 정의하고 미들웨어에서 이를 지원하기 위한 방법을 모색한다. 물론, 현재 센서를 이용하여 상황을 추론하고 서비스하는 다수의 응용들이 이미 존재 한다[1, 2, 3, 4, 5]. 그러나 이러한 응용들 대부분은 미들웨어 수준에서 지능형 서비스를 제공하는 것이 아니라 응용 시스템 지역적으로 혹은 응용 시스템의 도메인에 의존적인 서비스를 지원하는 것이 현실이다. 그 이유는 아직까지 미들웨어 수준에서 지능형 서비스를 지원하기 위한 구체적이고 범용적인 방법이 정의되지 않았기 때문이다. 본 논문에서는 미들웨어 수준에서 지능형 서비스를 지원하기 위한 USN 지능형 미들웨어를 설계하고 구현한다. 이를 위하여 USN 미들웨어에서 지능형 서비스 관리 컴포넌트를 정의하고 그 역할과 동작을 보인다.

USN 미들웨어 수준에서 지능형 서비스를 제공하기 위해서는 크게 두 가지 문제를 해결해야한다. 첫 번째는 다양한 응용들이 제공하는 서비스를 지원할 수 있어야 하며, 각 서비스를 제공하기 위해서 필요한 센서 정보와 연계해야한다는 것이다. 이를 위하여 본 논문에서는 서비스-센서 온톨로지 (Sensor-Service Ontology: SS-Ontology)를 정의하여, 응용 서비스와 센서 정보를 정의하고, 이들 사이의 연관 관계를 기술한다. 또한 SS-Ontology를 기반으로 저수준의 센싱 데이터로부터 고수준의 정보를 추론한다. 두 번째 문제는 다양한 응용들의 개별적 요구사항을 고려해야 한다는 것이다. 다시 말하면, 각 응용마다 상이한 응용의 특성 또는 환경 등을 서비스에 반영해야한다는 것이다. 예를 들면 동일한 화재 감시 응용이라 할지라도 '한라산 화재 감시'와 '63빌딩 화재 감시'를 위한 요구사항은 동일할 수 없으며, 한라산에 존재하는 온도 센서들 사이에도 주변 상황에 따라 다양한 특성을 가질 것이다. 그러므로 이러한 특성들을 반영하기 위하여 본 논문에서는 규칙을 이용하여 상황과 서비스를 추론하는 방법을 제안한다.

본 논문의 나머지 구성은 다음과 같다. 2절에는 본 논문과 관련된 연구들을 기술하고 있으며, 3절에서는 본 논문에서 제안하는 지능형 미들웨어의 구성과 지능형 서비스 관리 컴포넌트의 구체적인 구조와 흐름을 설명한다. 4절은 지능형 서비스를 제공하기 위해서 사용하는 온톨로지 기반 추론 방법과 규칙기반 추론 방법에 대해서 시나리오를 기반으로 설명한다. 5절에서는 각 추론 방법의 성능을 평가하고, 마지막으로 6절에서 결론과 함께 향후 연구의 진행 방향에 대해서 기술한다.

www.kci.go.kr

II. 관련 연구

센서 네트워크의 발전과 함께 미들에어에 관한 연구[6, 7, 8, 9], 메타데이터 관리에 관한 연구[10, 11], 그리고 지능형 서비스에 관한 연구[1, 2, 3, 4, 5] 등 많은 연구가 진행되었다. ETRI에서 개발 중인 COSMOS(COmmon System for Middleware Of Sensor network)는 다양한 유형의 USN 응용 서비스에 공통적으로 필요한 미들웨어의 핵심기능을 추출하고, 이들을 표준화된 방식으로 제공하기 위한 기술개발 및 표준화를 진행하고 있다 [6, 7]. COSMOS의 주요 기능으로는 다양한 유형의 질의 지원(일시성, 연속성, 이벤트), 대용량 센서 네트워크 환경에 대하여 대량의 동시질의 처리지원, 이종의 센서 네트워크에 대한 추상화 기능지원 등이 있다. 또한 COSMOS 역시 지능형 서비스를 지원하기 위한 필요성을 인지하고 이를 위한 컴포넌트를 제안하고 있으나, 구체적인 모습은 없는 실정이므로 본 논문의 결과가 COSMOS와 같은 범용의 미들웨어의 개발을 위해 사용될 수 있을 것이다. [2] 역시 미들웨어 수준에서 지능형 서비스의 지원이 필요하다는 것을 주장하고 있으나, 필요성만 언급하고 있을 뿐 구체적으로 어떻게 지원할 것 인지에대해서는 언급이 없는 실정이다.

[3]은 센서 기반 상황 인지 서비스를 제공하기 위한 미들웨어를 제안하고 있다. [3]에서 제안하고 있는 미들웨어인 SENSORD는 센서를 기반으로 미들웨어 수준에서 상황을 인지하고 서비스를 제공하기 위한 방법을 제안하고 있으므로 논문의 목적이 본 논문의 목적과 일치한다. 그러나 SENSORD는 상황 인지만을 위해 특별히 개발된 미들웨어이므로 범용의 USN 미들웨어로 보기는 힘들다. 또한 그 자체가 미들웨어이므로 범용의 미들웨어에 컴포넌트로 활용되는 것 역시 어려운 일이다. [1, 5]은 센서를 기반으로 현재의 다양한 상황을 인지하여 사용자에게 서비스를 제공하기 위한 미들웨어를 제안하고 있다. 이를 위하여 온톨로지와 규칙을 사용하는 방법은 본 논문에서 지능형 서비스를 제공하기 위한 방법과 기술적으로 유사해 보인다. 그러나 이들 방법은 응용이나 응용의 도메인에 의존적인 방법이다. 그러므로 본 논문에서처럼 범용의 유비쿼터스 센서 네트워크 환경에서 사용하기 위해서는 검증이 필요할 것이다.

[4, 5]은 센서 네트워크 환경에서 휴대용 단말기를 기반으로 상황인지 및 지능형 서비스를 제공하기 위한 미들웨어를 제안하고 있다. 그러나 미들웨어가 언제나 휴대용 단말에서 동작할 수는 없으므로 본 논문의 목적을 위해서는 좀 더 범용의 접근 방법이 필요하다.

[12, 13, 14, 15]에서는 센서를 기술하기 위한 방법으로 온톨로지를 제안하고 있으므로 이를 기반으로 추론을 수행하는 것이 가능하다. 그러나 앞선 연구들은 단순히 센서만을 고려한 온톨로지를 제안하고 있을 뿐 다양한 응용의 서비스와 센서들 사이에 연관관계를 정의하고 있지는 않다. 본 논문에서는 센서 온톨로지는 물론이고 서비스 온톨로지를 정의하고 이들 사이의 연관관계를 정의한다. 또한 기존의 센서 온톨로지를 본 논문에서 제안하고 있는 SS-온톨로지의 센서를 기술하기 위한 온톨로지로 활용할 수도 있다.

III. USN 미들웨어 및 지능형 서비스 관리 컴포넌트

<그림 1>은 본 논문에서 제안하는 USN 미들웨어의 구조를 보인다. USN 미들웨어는 USN 서비스 계층 (USN Service Layer), 센서 정보 처리 계층 (Sensor Information Processing Layer), 그리고 USN 추상화 계층(USN Abstraction Layer)으로 구성된다. USN 서비스 계층은 재난 감시, 유비쿼터스-실버케어 등과 같은 USN 응용들과 통신을 위한 OPEN API 컴포넌트가 존재하며, 모든 응용들은 이를 통하여 미들웨어에서 제공하는 서비스를 접근할 수 있다. 보안 관리 (Security Management) 컴포넌드는 사용자 인증 및 보안관련 서비스를 관리하며, 메타데이터 관리(Metadata Management) 컴포넌트는 USN 메타데이터를 저장 관리하며 응용에게 메타데이터 관련 서비스를 제공한다. 본 논문에서 초점을 맞추고 있는 지능형 서비스 관리(Intelligent Service Management) 컴포넌트는 센서 데이터 및 상황 정보를 기반으로 응용에게 지능형 서비스를 제공하는 역할을 담당한다. 센서 데이터 관리 (Sensor Data Management) 컴포넌트는 센서 정보 처리 계층에 존재하며, 실시간 센싱 데이터를 대상으로 일시성, 연속성, 이벤트 질의 등을 처리한다. USN 범용 인터페이스(USN Common Interface)는 센서 네트워크의 어댑터와 통신을 위한 컴포넌트로 센서 네트워크에 관련된 모든 요청 및 데이터 이동은 이를 통해 수행된다. USN 관리 (USN Management) 컴포넌트는 센서 네트워크 정보를 모니터링하고 관리하기 위한 컴포넌트로 현재 본 논문에서 제안하는 미들웨어에는 USN 모니터가 함께 포함되어있다.

www.kci.go.kr

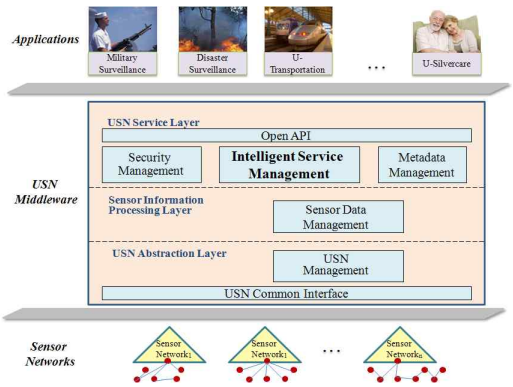

그림 1. USN 미들웨어의 구조
Fig. 1. An Architecture of the USN Middleware

지능형 서비스를 제공하기 위하여 지능형 서비스 관리 컴포넌트와 직접적으로 통신하는 컴포넌트는 센서 데이터 관리 컴포넌트이다. 센서 데이터 관리 컴포넌트는 센서 네트워크로부터 연속적으로 발생하는 센싱 데이터를 기반으로 질의를 처리하고 그 결과를 반환한다. 그러므로 지능형 서비스 처리를 위해 현재의 센싱 데이터가 필요한 경우, 이를 획득하기 위해서 사용된다. 또한 필요에 따라 지능형 서비스 관리 컴포넌트는 메타데이터 관리 컴포넌트와의 통신도 이루어진다. 예를 들어, 특정 센서의 물리적인 위치 정보나 센서가 설치된 시간 정보 등을 얻기 위해서 메타데이터 관리 컴포넌트와 통신한다. 물론 이러한 정보들은 추론을 위하여 온톨로지 문서 개체(individual)에 직접 기술할 수도 있으며 메타데이터 관리 컴포넌트에서 메타데이터로 관리할 수도 있다. 본 논문에서는 현재 추론의 편이를 위하여 온톨로지에 이러한 몇몇 정보들을 직접 기술하고 있다. 그러나 궁극적으로 추론에 직접적으로 사용되지 않는 정보들은 메타데이터로 저장하여 관리하고, 만약 지능형 서비스 제공을 위하여 메타데이터가 필요하다면 통신을 통해 획득하여 사용하는 것이 바람직할 것으로 사료된다. 그 이유는 메타데이터 기반 서비스 역시 미들웨어 수준에서 지원해야하는 서비스이므로 정보의 중복 저장의 측면에서나 미들웨어 서비스의 일원화 측면에서 효율적이지 못할 것이기 때문이다. 물론 정보의 중복 저장으로 인하여 추론에 필요한 시간을 좀 더 효과적으로 줄일 수는 있으나 이는 항상 데이터 관리 측면에서의 균형 (Trade-Off)의 문제로 남아 있다.

<그림 2>는 본 논문에서 제안하고 있는 지능형 서비스 관리 컴포넌트의 구조이다. 지능형 관리 컴포넌트는 OPEN API를 통해 두 가지 종류의 입력을 받는다. 첫 번째는 컴포넌트의 주목적인 지능형 서비스의 요청이고, 두 번째는 미들웨어 관리자 혹은 각 응용 시스템 관리자에 의한 온톨로지 및 규칙의 삽입, 삭제, 검색 그리고 갱신의 요청이다. 지능형 서비스 관리 컴포넌트의 Request Manager는 이러한 요청을 분석하여 해당 모듈에게 처리를 요청한다.

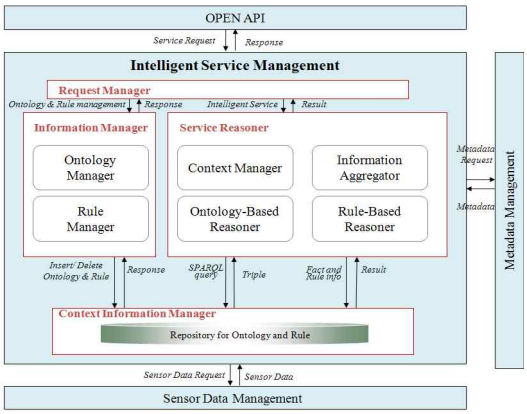

그림 2 지능형 서비스 관리 컴포넌트의 구조
Fig. 2. An Architecture of the Intelligent Management Component

Information Manager는 Ontology Manager와 Rule Manager를 통하여 각각 온톨로지와 규칙을 삽입, 삭제, 검색, 그리고 갱신하는 기능을 수행한다. Service Reasoner는 실제 지능형 서비스를 처리하는 역할을 담당한다. Service Reasoner의 하위 모듈인 Context Manager는 지능형 서비스 요청을 받고, 이를 처리하기 위하여 다른 하위 모듈들과 통신하며 각 모듈들 사이의 처리 및 데이터 흐름을 관리한다. Ontology-Based Reasoner는 Ontology를 기반으로 지능형 서비스에 필요한 센서 정보들과 상황 정보들을 추론한다. 또한 하위 수준의 정보로부터 상위 수준의 정보를 추론하는 역할을 담당한다. Rule-Based Reasoner는 규칙을 이용하여 각 응용의 특성과 현재 상황을 반영하여 서비스의 결과를 추론한다. Information Aggregator는 센서 데이터 관리 컴포넌트 그리고 메타데이터 관리 컴포넌트와 통신하며 추론에 필요한 정보를 획득한다. Context Information Manager는 저장소를 두어 온톨로지와 규칙을 저장, 관리한다.

IV. 지능형 서비스 추론

1. 지능형 서비스 추론의 처리 흐름

지능형 서비스 추론의 입력은 서비스를 요청하는 서비스 이벤트이다. 예를 들어, 지리산 산불 감시 응용 시스템은 "지

www.kci.go.kr

리산 A지구 산불 감시"와 같은 서비스 이벤트를 요청한다. 센서 미들웨어의 특성 상, 많은 수의 응용들은 응용의 시작과 함께 초기에 이벤트를 한번 발생시키고 그 결과를 지속적으로 기다린다. 즉, 산불 감시 응용은 초기 서비스 요청 후 산불의 발생 확률이 나타날 때 까지 결과를 지속적으로 기다린다. 응용이 요청한 서비스의 처리를 위하여 Service Reasoner는 두 단계의 추론 과정을 거친다. 일차 추론은 서비스를 제공하기 위해서 핵심이 되는 센서들을 추론하고 이를 기반으로 초기 상황을 추론하는 단계이고, 이차 추론은 보다 정확한 상황을 파악하고 해당 서비스의 결과를 생성하기 위한 단계이다. 이와 같이 추론을 두 단계로 구분한 이유는 센서 데이터를 실시간으로 처리할 때 필요한 자원의 사용을 줄이기 위해서이다. 즉, Sensor Data Management는 실시간으로 생성되는 센서 데이터를 지속적으로 관리하며 질의를 처리하므로 이때 처리해야할 센서의 수는 그 성능에 많은 영향을 미칠 수밖에 없다. 그러므로 본 논문에서는 일차 추론에서 핵심 센서들만을 기반으로 초기 상황을 추론하고, 이차 추론에서 보다 정확한 상황을 추론하고 서비스 결과를 생성한다. 물론 응용의 특성이나 미들웨어의 능력에 따라 일차 추론에서 모든 최종 결과를 생성하는 것 또한 가능하다.

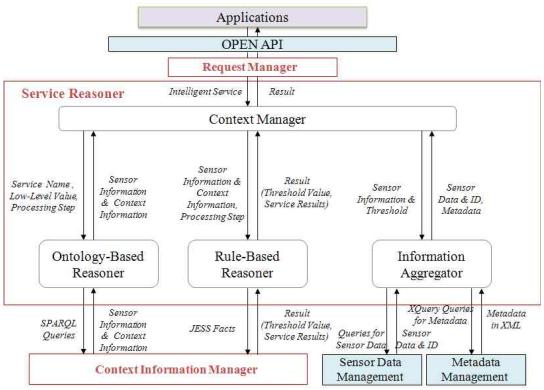

그림 3. 서비스 추론기의 구조 및 흐름
Fig. 3. An Architecture of The Service Reasoner and Action and Data Flow

<그림 3>은 Service Reasoner의 세부 구성 및 흐름을 보인다. 응용의 서비스 요청은 OPEN API를 통해 이루어진다. Service Reasoner 내부 모듈들 사이의 흐름 및 데이터를 제어하는 Context Manager는 Request Manager로부터 서비스의 요청을 받고 가장 먼저 Ontology-Based Reasoner에게 해당 서비스를 처리하기 위해 필요한 센서가 무엇인지 추론을 요청한다. 이때 제공하는 입력은 서비스명과 처리 단계이다. Ontology-Based Reasoner는 Context Information Manager를 통해 요청된 서비스를 위해 필요한 핵심 센서들과 현재의 상황 정보를 획득한다. 이때 Context Information Manager는 온톨로지 기반 추론을 수행하기 위한 도구로 JENA[16]를 사용한다. 이렇게 얻어진 센서 정보와 상황 정보들은 다시 Rule-Based Reasoner의 입력으로 제공된다. Rule-Based Reasoner는 입력된 정보들을 기반으로 현재 상황에서 해당 센서의 한계 값 (Threshold Value)을 추론한다. 물론 각 센서의 한계 값을 온톨로지에 미리 정의하고 이를 활용할 수 도 있다. 그러나 동일한 센서들이라 할지라도 응용의 특성이나 센서가 위치한 환경 상황등과 같이 여러 상황에 의해 한계 값은 상이할 수 있으므로 본 논문에서는 규칙 추론에 의해 핵심 센서의 한계 값을 추론한다. 이렇게 추론된 센서와 한계 값은 Information Aggregator를 통해 Sensor Data Management에게 요청되고, Sensor Data Management는 해당 센서가 한계 값을 만족하는 값을 센싱했을 경우 그 결과를 반환한다.

Service Reasoner의 이차 추론은 Sensor Data Management가 값을 반환하면서 시작된다. Context Manager는 이차 추론을 위해 메타정보가 필요하면 Information Aggregator를 통해 센서 장치, 센서 노드, 그리고 센서 네트워크의 메타데이터를 얻는다. 현재 본 논문에서 제안한 지능형 미들웨어에서는 XML을 기반으로 SensorML[18]을 이용하여 메타데이터를 기술하고 있으므로 XQuery 질의를 이용하여 메타데이터를 획득한다. 이차 온톨로지 기반 추론의 입력은 서비스명, 처리 단계, 그리고 센서 데이터(메타데이터 포함)이다. Ontology-Based Reasoner는 이차 추론 과정에서 해당 서비스를 처리하기 위해 필요한 추가 센서 정보 및 상황 정보를 추론하고 반환한다. 이차 추론 과정에서는 더 이상 각 센서 장치들의 한계 값을 추론할 필요가 없으므로, Context Manager는 Information Aggregator를 통해 추가 센서들의 현재 센서 데이터를 획득한다. 이렇게 획득한 센서 데이터와 온톨로지 기반 추론 과정에서 획득한 상황 정보는 이차 규칙 기반 추론의 입력으로 제공되고 Rule-Based Reasoner는 현재의 정확한 상황과 서비스의 결과를 추론하여 반환한다.

2. 서비스-센서 온톨로지

온톨로지 기반 추론을 위해 본 논문에서는 서비스-센서 온톨로지 (SS-Ontology)를 제안한다. 센서를 기술하기 위한 온톨로지는 이미 앞선 몇몇 연구에서 제안된바 있다[12, 13, 14, 15]. 그러나 SS-온톨로지는 센서들만을 기술하기 위한 온톨로지가 아니고 센서 네트워크 기반 지능형 서비스와 센서들 사이의 연관 관계를 기술하는 것이 주목적이다.

www.kci.go.kr

일반적으로 센서를 기술하는 온톨로지는 센서의 기능 및 능력을 전문적이고 구체적으로 표현한다. 그러나 센서에 대한 전문적인 지식이 부족한 응용 사용자들은 서비스를 얻기 위해 필요한 기능들을 일반적이고 추상적으로 표현하는 경우가 많다. 그러므로 SS-온톨로지는 이 둘 사이의 관계를 온톨로지를 이용하여 정의하고 추론에 이용한다.

SS-온톨로지는 <그림 4>와 같이 서비스 온톨로지(Service Ontology), 센서 온톨로지(Sensor Ontology), 그리고 환경 상황 온톨로지(Environmental Context Ontology)로 구성된다. 서비스 온톨로지는 센서 네트워크 기반 서비스를 정의한다. 또한 해당 서비스를 제공하기 위해서 필요한 요구사항들을 사용자 측면에서 기술한다. 센서 온톨로지는 각 센서를 기술하며 센서 온톨로지의 Capability 클래스는 해당 센서의 센싱 능력을 기술적이며 전문적으로 표현한다. 서비스 온톨로지의 Requirement 클래스는 needsCapability 속성으로 Capability 클래스와 연결되어있으며 이를 통하여 해당 서비스에 필요한 물리적인 센서를 추론할 수 있다. 환경 상황 온톨로지는 응용의 환경 상황을 기술하여 환경 상황을 고려한 추론을 수행할 수 있도록 한다. 본 논문에서는 온톨로지 기반 추론을 위하여 센서 온톨로지, 서비스 온톨로지, 그리고 환경 상황 온톨로지를 개별적으로 정의하여 사용하고 있지만, 각 온톨로지 대신 기존에 정의된 온톨로지를 적용할 수 있다.

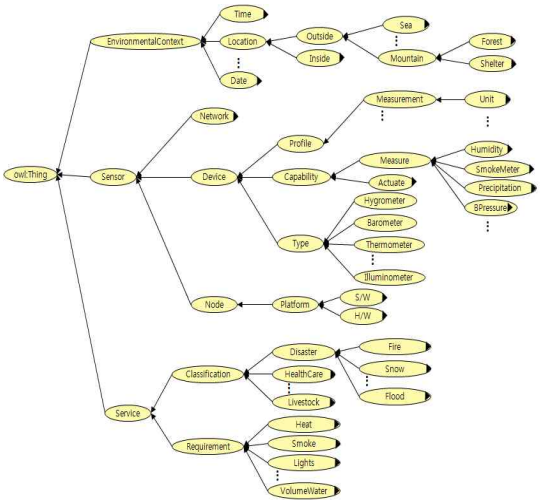

그림 4. 서비스-센서 온톨로지
Fig. 4. Service-Sensor Ontology

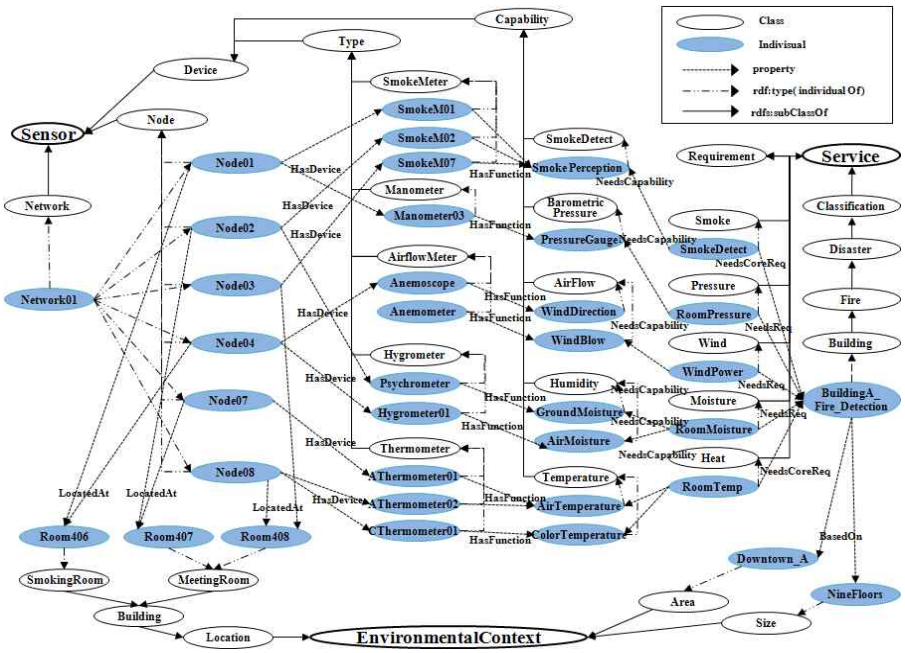

그림 5. 화재 감시를 위한 온톨로지 문서
Fig. 5. Ontology Instance for Fire Detection of the Building 'A'

www.kci.go.kr

3. 시나리오기반 온톨로지 및 규칙 기반 지능형 서비스 추론

본 논문에서는 재난 감시 응용들 가운데 빌딩 화재 감시 서비스 시나리오를 작성하고, 이를 기반으로 지능형 서비스의 처리 과정을 보인다. 응용 시스템으로부터 빌딩 화재 감시 서비스 요청이 들어오면, 첫 번째 추론 단계인 온톨로지 기반 추론에 의해 핵심 센서 및 상황 정보를 추론한다.

<그림 5>는 빌딩A의 화재 감시를 위한 온톨로지 문서의 예이다. 온톨로지에는 "needsCORERes" 속성을 이용하여 빌딩 A의 화재 감시를 위한 핵심 요구 사항으로 연기(Smoke Detect)와 방 온도(RoomTemp) 감지가 필요하다는 것을 추론할 수 있다. 또한 온톨로지를 통하여 현재 건물에 존재하는 센서들 가운데 연기 감지(SmokePerception)와 공기 온도(AirTemperature) 그리고 색 온도(ColorTemperature)를 감지할 수 있는 기능을 보유한 센서들이 총 6개 존재하며, 각 센서 장치들은 5개의 센서 노드에 설치되어있다는 것을 추론할 수 있다. Node01은 SmokeM01, Node02는 SmokeM02, Node03은 SmokeM03, Node07은 AThermometer01, Node08은 AThermometer02와 CThermometer01 센서 장치들을 포함하고 있으며, 모든 센서 노드들은 Network01에 구성요소로 동작하고 있다. 또한 Node01은 Room406, Node02와 Node07은 Room407, Node03과 Node08은 Room408에 존재함을 추론할 수 있으며, Room406은 흡연실이며, 나머지 방들은 미팅실임을 추론할 수 있다. 즉, 하위 수준 정보로부터 상위 수준 정보를 추론할 수 있는 것이다.

온톨로지 추론의 대표적인 추론 기능들 가운데 하나는 동등 혹은 유사한 개념을 추론에 이용하는 것이다. 예를 들어 실내 온도(RoomTemp)를 얻기 위한 센서의 능력으로 "Needs Capability" 속성에 의해 공기 온도(AirTemperature)만 연관되어 있다할 지라도 만약 현재 상황에서 공기 온도를 측정할 수 있는 센서가 존재하지 않는다면, 대체 센서 기능으로 색온도(ColorTemperature) 기능을 추론에 이용할 수 있다. 일차 온톨로지 추론에 의해 핵심 센서와 환경 정보들이 추론되면, 일차 규칙기반 추론에 의해 각 센서들의 한계 값을 추론한다. 연기 감지 센서 SmokeM01은 흡연실에 설치되어있다. 그러므로 미팅실에 설치된 연기 감지 센서와의 임계값은 서로 다를 수 있다. 그러므로 <표 1>과 같이 SmokeM01 연기 감지 센서의 위치 정보를 기준으로 임계값을 추론할 수 있다.

표 1. 흡연실에 연기센서의 임계값을 추론하기 위한 규칙
Table 1. Rule for Reasoning the Threshold Value of Smoke Meters in a Smoking Room

JESS Rule
(defrule ThresholdValueOfSmokeMInSmokingRoom (Service BuildingFire) (needsDevice SmokeMeter) (Location SmokingRoom) => (assert (SmokeMeterThreshold ?value)))

일차 규칙 기반 추론에 의해서 모든 센서들의 임계값이 추론되면, Sensor Data Management는 센서 장치들로부터 센싱 데이터을 지속적으로 취하면서 임계값을 만족하는 경우 그 값을 반환한다. 예를 들어, SmokeM01의 센싱 데이터가 임계값인 50을 넘었을 경우, 그 값을 반환하고 Context Manager는 다시 이차 온톨로지 추론을 통해 추가 센서들의 정보를 추론한다.

<그림 5>의 예제에서 이차 온톨로지 추론에 의해서 추론된 센서들은 실내 기압(RoomPressure)을 얻기 위한 Manometer03, 바람의 강도(WindPower)를 얻기 위한 Anemoscope와 Anemometer, 방안의 습도(RoomMoisture)를 측정하기 위한 Psychrometer와 Hygrometer01이다. 또한 노드의 위치 및 건물의 규모, 위치한 지역 등과 같은 추가적인 상황 정보도 함께 추론한다. 이렇게 추론된 센서 장치들의 정보는 곧바로 현재의 센싱 데이터를 획득하기 위해 사용된다. Context Manager는 각 센서 장치들의 현재 센싱 데이터를 획득한 후, 이를 이차 온톨로지 추론에 의해서 얻어진 상황 정보와 함께 이차 규칙 추론의 입력으로 제공한다.

이차 규칙 추론은 보다 정확한 현재의 상황을 추론하여 서비스 결과를 생성한다. 또한 특별한 경우 사용자에 의해서 미리 정의된 규칙을 기반으로 다음 상황을 예측한 결과를 추론하기도 한다. <표 2>는 보다 정확한 현재의 화재 상황과 화재가 발생한 방의 상황을 추론하기 위한 이차 규칙 추론의 예이다. <표 2>의 상위에 기술된 규칙은 Room406의 연기, 온도, 그리고 습도를 기반으로 보다 정확한 화재의 상황 판단을 추론하기 위한 규칙이다. 그리고 하위에 기술된 규칙은 Room406의 현재 상황을 추론하기 위한 규칙으로, Room406에 화재용 스프링클러가 존재하지만 습도 수치가 매우 낮으므로 스프링클러가 동작하지 않음을 추론할 수 있다.

www.kci.go.kr

표 2. 빌딩 화재 상황과 화재가 발생한 방의 상황을 추론하기 위한 규칙
Table 2. Rules for the Building Fire and the Room Situation

JESS Rule for the Fire Detecting of the BuildingA
(defrule TheFireDetectingServiceforTheBuildingA (Service BuildingFire) (SmokeMeter (Value ?Smoke_Value)) (Thermometer (Value ?Thermo_Value)) (Hygrometer (Value ?Hygro_Value)) (Location MeetingRoom) (SmokingRoom (Room406)) (test (> ?Smoke_Value 50) (> ?Thermo_Value 70) (< ?Hygro_Value 50)) => (assert (currentSituation ?Result)))
JESS Rule for the Room situation
(defrule TheRoomSituationBasedOnMoisture (Hygrometer (Value ?Hygro_Value)) (SmokingRoom (Room406) (hasFireSprinklers TRUE)) (test (< ?Hygro_Value 20)) => (assert (currentSituation ?Result)))

이와 같이 이차 규칙 추론은 현재의 정확한 상황은 물론 선택적으로 추가적인 예측 상황을 추론하기 위해서 사용된다.

V. 성능 평가

본 논문에서는 성능 평가를 위하여 Intel(R) Core(TM)2 Duo CPU 2.20GHz, 2.0GB 메모리를 사용하며, 온톨로지 기반 추론을 위하여 JENA[16]를 규칙기반 추론을 위하여 JESS[17]를 사용한다. 실험을 위한 센서 네트워크 환경은 시뮬레이터를 구현하여 가상의 센서 장치들을 생성하여 시험한다.

일반적으로 지능형 서비스를 제공하기위한 추론의 비용은 매우 높다. 또한 본 논문의 목적은 지능형 서비스를 지원하기 위한 범용의 USN 미들웨어를 개발하는 것이다. 결국 본 논문에서 제안하는 지능형 서비스 처리 방법은 유비쿼터스-헬스케어, 재난감시, 장치 협업 응용 등과 같은 센서 네트워크 환경에서의 다양한 응용 시스템에서 수용할 만한 성능을 보여야 한다. 그러므로 본 연구를 실제 응용에 적용하기 위해서는 추론의 수행 시간이 참조되어야할 것이다. 이를 위하여 본 논문에서는 성능 평가를 제안된 지능형 서비스 추론 방법이 특정 서비스나 센서 장치에 의존적이지 않고 어느 정도 수준에서 수용할 만한 성능을 보이는지를 평가한다.

표 3. 성능평가를 위한 서비스와 센서 장치
Table 3. Services and Sensor Device for Evaluation

Service	Core Sensor Device (Context Info)
	Additional Sensor Device (Context Info)
Building Fire	SmokeMeter, Thermometer (Location)
	Humidity, Barometer, Illumination, Sound, Motion (BuildingInfo, Extinguisher)
Mountain Fire	SmokeMeter (Location, Date)
	Thermometer, Color_Temperature, Humidity, Altitude, CO2, Anemoscope, Anemometer (Time, ForestInfo)

온톨로지 기반 추론의 성능에 직접적인 영향을 미치는 요인은 온톨로지에 기술된 각 클래스들 사이의 연관 관계의 수이다. <표 3>은 온톨로지 기반 성능 평가를 위해 선택된 빌딩 화재 감시 서비스와 산불 감시 서비스이며, 이를 추론하기 위해서 필요한 센서 장치 및 환경 상황 정보를 보인다. 본 논문에서는 온톨로지 기반 추론의 성능을 평가하기 위하여 센서 장치의 수를 100개에서 500개까지 100개씩 증가시키면서 추론 시간을 측정했다. 각 센서 장치는 최소 5개에서 최대 15개까지의 속성을 갖는다. 그러므로 결국 100개의 센서 디바이스는 최소 500개에서 1500개까지의 연관 관계를 포함한다. 물론 실제 응용에서 500개의 센서가 특정 서비스를 추론하기 위해서 사용될 확률은 많지 않다. 그러나 향후 센서 환경의 확장에 따라 필요할 수 있으며 미들웨어의 성능 평가의 관점에서 이러한 환경을 구성하여 평가한다.

<그림 6>은 일차 온톨로지 추론과 이차 온톨로지 추론의 수행 시간을 보인다. 두 서비스의 일,이차 추론 시간 모두 센서 장치의 개수 증가에 비례하는 것을 볼 수 있다.

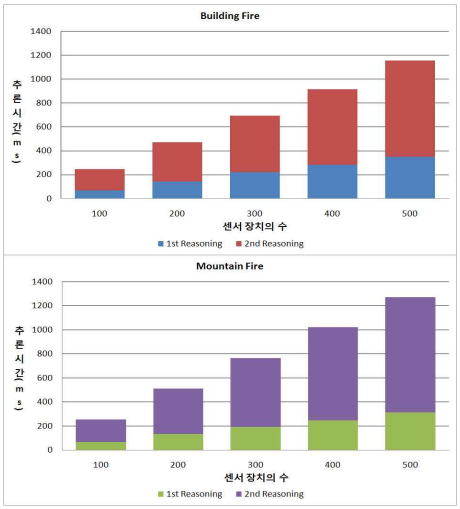

그림 6. 온톨로지 기반 추론의 수행 시간
Fig. 6. Ontology-Based Reasoning times

www.kci.go.kr

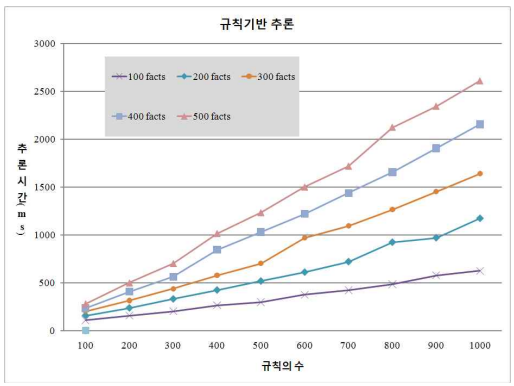

그림 7. 규칙 기반 추론의 수행 시간
Fig. 7. Rule-Based Reasoning times

규칙 추론의 입력은 규칙(Rule)과 사실(Fact)이다. 그러므로 본 논문에서는 규칙 기반 추론의 성능 평가를 위하여 규칙의 수를 100개에서 1,000개까지 100개씩 증가시키면서 그 성능을 평가한다. 사실의 개수는 온톨로지 추론에 의해 추론된 모든 정보들이 참인 경우를 가정하여 100개에서 500개까지 100개씩 구분하여 성능을 평가 한다. 규칙기반 추론은 일차, 이차 추론의 단계나 서비스의 종류에 구분 없이 모두 규칙과 사실의 수에 의존적인 성능 평가를 보였으므로 이를 구분하지 않고 하나의 그래프를 이용하여 성능 평가의 결과를 보인다. <그림 7>에서 볼 수 있는 것처럼 규칙 기반 추론 역시 규칙이나 사실의 수에 비례하여 증가하는 것을 볼 수 있다. 그러나 500개의 사실에 약 2,000개의 규칙을 적용했을 경우, 메모리의 한계 범위를 벗어나는 결과를 보였다. 그러므로 이러한 경우, 응용의 필요에 따라 규칙 기반 추론의 단계를 더욱 세분화하는 것을 고려해야할 것이다.

VI. 결론

본 논문에서는 유티쿼터스 센서 네트워크 환경에서 지능형 서비스를 지원하기 위한 방법을 제안하고, 이를 위한 미들웨어를 제안하였다. 특별히 센서 네트워크 미들웨어에서 지능형 서비스를 처리하기 위한 방법을 구체적으로 보였으며, 성능평가를 통해 유효성을 보였다. 또한 SS-온톨로지를 정의하여 미들웨어 수준에서 지능형 서비스를 일반화 하고 다양한 종류의 지능형 서비스와 이종의 센서 네트워크 환경 사이에 연관관계를 정의했다. 이와 함께 규칙 기반 추론 방법을 이용하여 각 응용의 다양한 특성이나 상황을 추론에 반영하기 위한 방법을 제안했다. 본 논문의 결과는 현재 활발히 연구되고 있는 센서 네트워크 환경의 미들웨어 표준을 위한 참고자료로 사용될 것이며, 유비쿼터스 헬스케어, 재난 감시, 자원 협업 등과 같은 센서 기반 지능형 서비스를 지원하기 위한 응용 시스템에서도 참고할 수 있을 것이다.

현재 본문에서 사용하고 있는 규칙 기반 추론 방법은 시스템 관리자나 응용 사용자에 의해서 미리 정의된 규칙들을 사용하고 있다. 그러나 향후 응용이나 상황의 변화, 또는 [19]와 같이 동적으로 변화하는 센서 네트워크의 상황을 규칙에 반영하기 위한 방법을 연구하여 추론에 반영할 계획이다.

참고문헌

[1] E. Bouillet, M. Feblowitz, Z. Liu, A. Ranganathan, A. Riabov, and F. YeA, "Semantics-based Middleware for Utilizing Heterogeneous Sensor Networks," Proc. DCOSS 2007, USA, pp.174-188, June 2007.

[2] T. S. Lopez, and D. Y. Kim, "A Context Middleware Based on Sensor and RFID Information," Proc. PerComW'07, USA, pp.331-336, March 2007.

[3] A. Sashima, Y. Inoue, and K. Kurumatani, "Spatio-Temporal Sensor Data Management for Context-Aware Services," Proc. ADPUC 2006, Australia, pp.4-9, November 2006.

[4] S.S. Yau, and F. Karim, "Context-Sensitive Middleware for Real-time Software In Ubiquitous Computing Environments," Proc. ISORC 2006, Korea, pp.163-170, April 2001.

[5] E. J. Ko, H. J. Lee and J. W. Lee, "Ontology-Based Context Modeling and Reasoning for U-HealthCare," IEICE Transactions on Information and Systems, Vol.E90-D, no.8, pp.1262-1270, August 2007.

[6] Y. B. Kim, M. Kim, and Y. J. Lee, "COSMOS: a middleware platform for sensor networks and a u-healthcare service," Proc. SAC 08, Brazil, pp. 512-513, March, 2008.

[7] M. Kim, J. W. Lee, Y. J. Lee, and J. C. Ryou, "COSMOS : A Middleware for Integrated Data Processing over Heterogeneous Sensor Networks," ETRI journal, vol.30, no.5, pp.696-706, October 2008.

[8] C. Hermann and W. Dargie, "Senceive: A Middleware for a Wireless Sensor Network," Proc. AINA 2008,

www.kci.go.kr

Japan, pp.612-619, March 2008.
[9] K. Aberer, M. Hauswirth, and A. Salehi,"A middleware for fast and flexible sensor network deployment," Proc. 32nd VLDB, Korea, pp.1199-1202, September 2006.
[10] Y. Obashi, T. Kokogawa, Y. Zheng, H. Chen, H. Mineno, and T. Mizuno, "A Meta-Data-Based Data Aggregation Scheme in Clustering Wireless Sensor Network," Proc. KES 2007, pp.477-483, September 2007.
[11] D. Ballari, M. Wachowicz, and M. A. M. Callejo, "Metadata behind the Interoperability of Wireless Sensor Networks," Sensors, Vol.9, no.5, pp.3635-3651, May 2009.
[12] S. Avancha, C. Patel, and A. Joshi, "Ontology-driven Adaptive Sensor Networks," Proc. Mobi Quitous 2004, USA, pp.194-202, August 2004
[13] R. Jurdak, C. V. Lopes, and P. Baldi, "A Frame work for Modeling Sensor Networks," Proc. OOPSLA'04, Canada, pp.1=5, October 2004.
[14] M. Eid, R. Liscano, and A. E. Saddik, "A Universal Ontology for Sensor Networks Data," Proc. CIMSA 2007, Italy, pp.59-62, June 2007.
[15] D.J. Russomanno, C. Kothari and O. Thomas, "Building a sensor ontology: A practical approach leveraging ISO and OGC models," Proc. IC-AI2005, UAS, pp.637-643, June 2005.
[16] JENA2: A Semantic Web Framework, "*http://jena.sourceforge.net/*".
[17] JESS, the Rule Engine for the JavaTM Platform, "*http://herzberg.ca.sandia.gov/jess/*".
[18] OGC Sensor Model Language, OpenGIS Standard 2007. "*http://www.opengeospatial.org/standards/sensorml*".
[19] 조영복, 최재민 이상호, "센서 네트워크 환경에서 스카이라인 질의를 이용한 효율적인 동적 예측 클러스터링 기법", 한국컴퓨터정보학회논문지, 제 13권, 제 7호, 39-148쪽, 2008년 12월.

저 자 소 개

박 종 현

2002년 : 충남대학교 컴퓨터과학과 석사 졸업
2007년 : 충남대학교 컴퓨터과학과 박사 졸업
2007년 ~ 2008년 : 충남대학교 소프트웨어연구소 전임연구원
2009년 ~ 2009년 : 거제대학 조선정보계열 초빙교수
2009년 ~ 현재 : 큐슈대학교 정보기반 연구센터 방문연구원
관심분야 : 상황인지, 추론시스템, XML, XQuery, Ontology, 분산 데이터베이스, 유비쿼터스 컴퓨팅, 웹정보시스템

강 지 훈

1981년 : 한국과학기술원 전산학과 석사졸업
1996년 : 한국과학기술원 전산학과 박사졸업
2000년 ~ 2002년 : 충남대학교 정보통신원장
1985년 ~ 현재 : 충남대학교 전기정보통신공학부 교수
관심분야 : 시멘틱웹, 추론, XML, XQuery, 데이터베이스 시스템, 웹정보시스템

www.kci.go.kr