

## 유비쿼터스 컴퓨팅 환경 하에서 개체 간 상호 이해를 위한

### Bridge Ontology 생성 및 제공 기법 연구

강태훈<sup>o</sup> 박경량 김신덕  
연세대학교 컴퓨터과학과 슈퍼컴퓨팅연구소  
{thkang<sup>o</sup>, lanx, sdkim}@parallel.yonsei.ac.kr

#### Bridge Ontology Generation and Provision for Mutual Understanding In Ubiquitous Computing Environment

T.H. Kang<sup>o</sup> K.R. Park S.D. Kim  
SCL, Department of Computer Science, Yonsei University

#### 요 약

유비쿼터스 컴퓨팅 환경을 위해서는 그 구성 개체들 간의 의사소통이 원활히 이루어져야 한다. 이를 위한 중요한 기술 중 하나가 사람이 아닌, 컴퓨터가 데이터를 이해하고 직/간접적으로 처리할 수 있도록 하는 기술이며, 그래서 도입된 개념이 온톨로지(Ontology)이다. 본 논문에서는 이러한 온톨로지를 기반으로 하는 유비쿼터스 컴퓨팅 환경 하에서 서로 다른 온톨로지를 기반으로 하는 구성개체들이 원활하게 서로 이해하고 상호작용할 수 있도록 Ontology Mediator와 Ontology Bridging Service를 기반으로 하는 Bridge Ontology 생성 및 제공 기법을 제안한다.

#### 1. 서 론

유비쿼터스 컴퓨팅이란 라틴어로 '편재하다(보편적으로 존재하다)'라는 의미를 가진 'Ubiquitous'라는 단어에서 유래된 개념으로, 물이나 공기처럼 주변 환경 모든 사물 및 사람이 보이지 않는 네트워크로 연결되어 이를 의식하지 않고 사용할 수 있는 환경을 말한다.[1] 이러한 유비쿼터스 컴퓨팅 환경을 위해서는 그 구성 개체들 간의 의사소통이 원활히 이루어져야 한다. 이를 위한 중요한 기술 중 하나가 사람이 아닌, 컴퓨터가 데이터를 이해하고 직/간접적으로 처리할 수 있도록 하는 기술이며, 그래서 도입된 개념이 온톨로지(Ontology)이다.

온톨로지란 시멘틱웹(Semantic Web) 분야에서 크게 사용되고 있는 개념으로, 공유된 개념화에 대한 정형화된 명세를 의미한다.[2] 이와 같은 맥락에서 유비쿼터스 컴퓨팅에서의 온톨로지란 그 구성 개체 및 개념들에 대한 공통적인 표현에 대한 표준화를 의미하며, 이를 통해 관계들에 대한 명확한 표현 및 의미 부여를 가능하게 함으로서 구성개체들이 서로를 이해하고 상호작용할 수 있도록 한다. 하지만 기존의 연구들은 각각의 연구에서 제시하는 온톨로지만을 고려할 뿐 서로 다른 온톨로지를 사용하는 개체간의 상호작용에 대해서는 크게 신경 쓰지 않고 있다.

본 논문에서는 유비쿼터스 컴퓨팅 환경 하에서 서로 다른 온톨로지를 기반으로 하는 구성개체들이 원활하게 서로 이해하고 상호작용할 수 있도록 Ontology Mediator와 Ontology Bridging Service를 기반으로 하는 Bridge Ontology 생성 및 제공 기법을 제안한다.

#### 2. 관련 연구

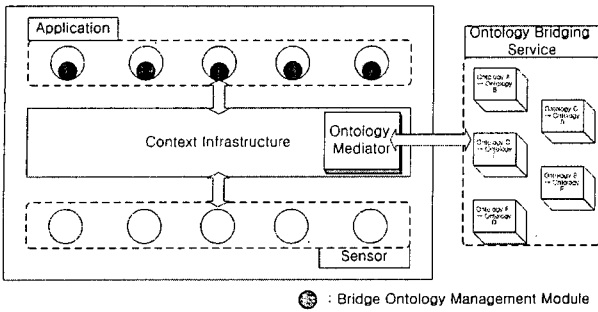
유비쿼터스 컴퓨팅에서 온톨로지의 필요성이 강조됨에 따라 여러 프로젝트에서 온톨로지 모델에 대한 연구가 진행되고 있다. CoBrA(Context Broker Architecture)의 SOUPA(Standard Ontology for Ubiquitous Pervasive Applications)[3][4], Semantic Space의 ULCO(Upper Level Context Ontology)/ECO(Extended Context Ontology)[5], 등을 그 연구 예로 들 수 있다. 또한 그 외의 여러 프로젝트에서도 별도의 온톨로지 모델에 대해 연구하고 있지는 않지만 암묵적으로 나름의 온톨로지를 정의하여 사용하고 있다.

하지만 이러한 프로젝트들은 각각의 연구에서 제시하는 온톨로지를 기반으로 하는 환경만을 가정할 뿐 서로 다른 온톨로지를 사용하는 경우에 대해서는 거의 고려하고 있지 않은 실정이다. 즉, 서로 다른 온톨로지를 사용하는 응용과 인프라가 만나는 경우 서로 전달하는 정보의 의미를 이해하지 못하므로 원활하게 동작할 수 없게 된다. 따라서 온톨로지 모델뿐 아니라 서로 다른 온톨로지간의 이해에 대한 연구도 필요하다.

#### 3. Bridge Ontology 생성을 통한 개체 간 상호 이해

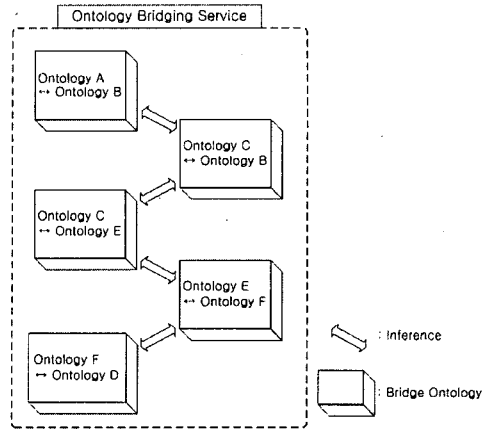
##### 3.1. Ubiquitous Computing Architecture

본 논문에서 제시하는 유비쿼터스 컴퓨팅 환경의 구조는 [그림 1]과 같으며, 크게 네 부분으로 나눌 수 있다.



[그림 1] Ubiquitous Computing Architecture

As(A[X ], C[?])와 같은 형태의 쿼리를 통해 sameClass As(A[X ], C[Z])와 같은 Bridge Ontology를 간단히 생성할 수 있다.



[그림 2] Ontology Bridging Service

센서(Sensor)는 컨텍스트 정보를 생성하여 인프라를 통해 제공하는 역할을 하며, 각자의 온톨로지를 기반으로 구성된다. 즉, 센서로부터 제공되는 컨텍스트는 각자의 온톨로지를 기반으로 표현된다. 이때 센서는 컨텍스트 인프라의 Ontology Mediator에게 자신의 온톨로지가 무엇인지 알린다.

응용(Application)은 컨텍스트 인프라에게 컨텍스트 정보를 요청하고 이 정보를 바탕으로 서비스를 제공하는 역할을 한다. 응용은 각자의 온톨로지를 기반으로 구성되며 센서와 마찬가지로 Ontology Mediator에게 자신의 온톨로지가 무엇인지 알린다. 또, 서로 다른 온톨로지를 기반으로 하는 개체와의 정보교환을 위해 Ontology Bridging Service에서 제공하는 Bridge Ontology를 저장, 관리하는 Bridge Ontology Management Module을 갖는다.

컨텍스트 인프라(Context infrastructure)는 센서로부터 제공되는 컨텍스트를 수집, 가공하여 응용에게 제공한다. 이 과정에서 응용과 센서가 서로 다른 온톨로지를 기반으로 하는 경우 발생할 수 있는 문제를 해결하기 위해 Ontology Mediator를 갖는다. Ontology Mediator는 응용과 센서의 온톨로지가 무엇인지 파악하고 이를 바탕으로 Ontology Bridging Service에 Bridge Ontology를 요청하여 이를 응용의 Bridge Ontology Management Module에 제공한다.

Ontology Bridging Service는 서로 다른 온톨로지간의 맵핑 정보를 저장 및 관리하며 컨텍스트 인프라의 요청을 바탕으로 Bridge Ontology를 제공한다. 그 구조는 [그림 2]와 같다. Ontology Bridging Service는 OWL(Web Ontology Language), DAML-OIL(DARPA Agent Markup Language-Ontology Inference Layer)과 같은 온톨로지 언어를 사용하여 구성되며, 이를 통해 다음과 같은 장점들을 갖게 된다.

우선 서로 다른 여러 온톨로지들을 맵핑 하는데 있어서 그 경우의 수를 크게 줄일 수 있다. 즉, 각 온톨로지의 관계 중 클래스에 관한 정보를 OWL의 sameClassAs를 통해 정의한다고 하면, sameClassAs(A[X ], B[Y])와 sameClassAs(B[Y ], C[Z])에서 sameClassAs(A[X ], C[Z])를 추론해 낼 수 있다. 또한 이와 관련해서 여러 온톨로지간의 Bridge Ontology를 생성하는데 있어서도 일일이 모든 경우를 저장하고 관리할 필요가 없다. 앞의 예에서와 같이 sameClassAs(A[X ], B[Y])와 sameClass As(B[Y ], C[Z])가 존재한다고 할 때, sameClass

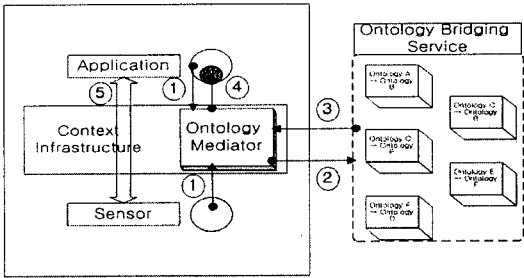
### 3.2. Bridge Ontology Generation

서로 다른 온톨로지를 기반으로 하는 응용과 센서 간에 정보교환이 일어날 때, 이러한 정보교환을 가능하게 하는 Bridge Ontology의 생성은 Ontology Mediator와 Ontology Bridging Service간의 상호작용을 통해 일어난다. Ontology Mediator는 여러 Bridge Ontology가 등록되어 있는 개별적인 공간(ex. Web Services, Grid)에서 저장된 정보, 즉 응용과 센서의 온톨로지를 기반으로 적절한 Ontology Bridging Service를 찾아 구성하고, 구성된 Ontology Bridging Service는 적절한 Bridge Ontology를 생성하여 다시 Ontology Mediator를 통해 응용에 전달한다. 예를 들어 [그림 2]와 같은 Bridge Ontology들로 구성된 공간을 가정하고, 응용은 온톨로지 B를 센서는 온톨로지 D를 기반으로 할 때, Ontology Mediator는 [Ontology B ↔ Ontology C], [Ontology C ↔ Ontology E], [Ontology E ↔ Ontology F], [Ontology F ↔ Ontology D] 이 네 가지 Bridge Ontology만으로 Ontology Bridging Service를 구성하며, 이렇게 구성된 Ontology Bridging Service는 제공받은 응용과 센서의 온톨로지 정보를 기반으로 앞에서 설명한 sameClassAs와 같은 추론기능을 통해 [Ontology B ↔ Ontology D]의 Bridge Ontology를 생성, Ontology Mediator를 통해 응용의 Bridge Ontology Management Module전달한다.

### 3.3. Operation Flow

[그림 3]은 이러한 구조 하에서의 동작의 흐름을 설명한 그림이다. 응용과 센서는 컨텍스트 정보를 요청하거나 제공하기에 앞서 ①에서와 같이 각자의 온톨로지가 무엇인지를 인프라에게 알리며, Ontology Mediator는 이 정보를 받아서 보관하게 된다. 이후 응용과 센서간의 정보교환이 필요한 경우가 발생하면 Ontology Mediator는 ②에서와 같이 저장된 정보를 바탕으로 적절한 Ontology Bridging Service를 찾아 구성하고, 응용과

센서의 온톨로지 정보를 제공한다. 이렇게 제공된 정보를 바탕으로 Ontology Bridging Service는 ③에서와 같이 두 온톨로지를 맵핑해줄 수 있는 Bridge Ontology를 생성하여 이를 다시 Ontology Mediator에게 제공하며, Ontology Mediator는 ④에서와 같이 이 Bridge Ontology를 응용의 Bridge Ontology Management Module에게 제공한다. 그러면 응용은 ⑤에서와 같이 Bridge Ontology Management Module에 저장되어 있는 Bridge Ontology를 바탕으로 센서와 서로 전달하는 정보의 의미를 명확히 이해하면서 상호작용할 수 있다.

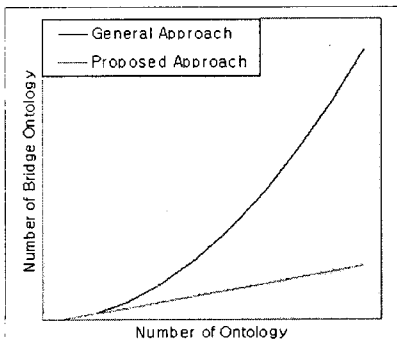


● : Bridge Ontology Management Module

[그림 3] Operation Flow

3.4. 성능 분석

[그림 4]는 Ontology Mediator와 Ontology Bridging Service를 통한 방법과 일반적인 방법 사이의 차이를 비교하기 위해 온톨로지의 숫자가 늘어남에 따라 이러한 온톨로지 간의 맵핑 정보를 제공하는 Bridge Ontology의 숫자가 어떻게 늘어나는지 보여주는 그래프이다. 그래프에서 보이는 바와 같이 일반적인 방법은 서로 다른 모든 온톨로지 간의 Bridge Ontology를 생성해야하므로 온톨로지의 숫자가 늘어남에 따라 급격하게 증가하는 모습을 보이는 반면, 제안하는 방법은 서로 다른 모든 온톨로지가 증가함에 따라 일정한 비율로 증가하는 모습을 보인다. 이와 마찬가지로 의미에서 서로 다른 모든 온톨로지 간의 Bridge Ontology를 저장하거나 관리하기 위한 공간 또한 [그림 4]의 그래프와 같은 결과를 보인다.



[그림 4] Performance Comparison

3. 결론 및 향후 연구

유비쿼터스 컴퓨팅을 위해서 요구되는 여러 다양한 기술들 중 중요한 한 가지는 사람이 아닌 컴퓨터가 데이터를 이해하고 직/간접적으로 처리할 수 있게 하는 기술이다. 이를 위해 도입된 개념이 온톨로지이며, 여러 다양한 프로젝트들에서 사용되고 있다. 하지만 현재까지는 단순히 하나의 온톨로지를 사용하는 경우만을 가정할 뿐, 서로 다른 여러 온톨로지를 사용하는 경우에 관해서는 거의 고려되고 있지 않다. 본 논문에서는 이와 같이 유비쿼터스 컴퓨팅 환경 하에서 하나의 온톨로지만을 사용하는 경우가 아닌 서로 다른 여러 온톨로지를 사용하는 경우의 해결방법을 제시하였다.

향후 과제로 컨텍스트 인프라와 Ontology Bridging Service를 좀 더 구체화하여 구현해보아야 할 것이다.

참고문헌

- [1] Weiser, M. "The computer for the 21st century.", Scientific American, 265(3), 94-104, 1991.
- [2] Web-Ontology Working Group <http://www.w3.org/2001/sw/WebOnt/>
- [3] Harry Chen, Tim Finin, and Anupam Joshi, "An Ontology for Context-Aware Pervasive Computing Environments", Workshop on Ontologies and Distributed Systems, Knowledge Engineering Review, August 09, 2003
- [4] Chen, H, Perich, F, Finin, T, Joshi, A, "SOUPA: standard ontology for ubiquitous and pervasive applications", Mobile and Ubiquitous Systems: Networking and Services, 2004.
- [5] Wang, X, Dong, J.S., Chin, C.Y., Hettiarachchi, S.R., Zhang, D., "Semantic Space: an infrastructure for smart spaces" Pervasive Computing, IEEE Volume 3, Issue 3, Page(s):32-39, July-Sept. 2004