

유비쿼터스 컴퓨팅 환경에서의 상황 인식을 위한 확률 확장 온톨로지 모델

정헌만*, 이정현**

Probability-annotated Ontology Model for Context Awareness in Ubiquitous Computing Environment

Jung Heon Man *, Lee Jung Hyun **

요약

유비쿼터스 컴퓨팅 환경에서 현재의 상황 인식 어플리케이션은 다루고 있는 상황 정보가 정확하다고 가정하지만, 실제로 센서로 입력되고 해석된 상황 정보들은 종종 모호하거나 불확실하다. 본 논문에서는 상황 정보의 모호성을 해결하기 위하여 베이지안 네트워크를 사용하고 상황 정보를 표현하기 위해 온톨로지 기반 모델을 확장한 확률 모델을 제안한다. 이 논문에서 제시한 확률 확장 온톨로지 기반 상황 인식 미들웨어는 유비쿼터스 환경에서 요구되는 다양한 상황 인식 서비스의 개발 및 운용을 효과적으로 지원할 수 있다.

Abstract

Current context-aware applications In ubiquitous computing environments make the assumption that the context they are dealing with is correct. However, in reality, both sensed and interpreted context informations are often uncertain or imperfect. In this paper, we propose a probability extension model to ontology-based model for representing uncertain contexts and use Bayesian networks to resolve about uncertainty of context informations. The proposed model can support the development and operation of various context-aware services, which are required in the ubiquitous computing environment.

▶ Keyword : 상황인식 모델(Context-Aware Model), 상황 온톨로지(Context Ontology), 모호한 상황 정보(Uncertain contexts), 상황 인식 미들웨어(Context-Aware Middleware), 유비쿼터스 컴퓨팅(Ubiquitous Computing)

• 제1저자 : 정헌만
• 접수일 : 2006.07.11, 심사일 : 2006.07.14, 심사완료일 : 2006.07.19
* 인하대학교 컴퓨터공학부 강사 ** 인하대학교 컴퓨터공학부 교수

I. 서론

유비쿼터스 컴퓨팅은 일상생활 속에 편재해 있는 컴퓨팅 자원을 이용하여 사용자가 언제 어디서나 동적인 서비스를 받을 수 있는 환경을 제공하며 조용한 컴퓨팅(calm computing)[1], 보이지 않는 컴퓨팅(invisible computing)[2], 사라지는 컴퓨팅(disappearing computing)[3] 등의 용어는 유비쿼터스 컴퓨팅에 관한 사용자 인터페이스 관점을 잘 설명해 주고 있다.

유비쿼터스 환경에서는 기존 컴퓨팅 환경에서의 사용자와 컴퓨터간의 대화형 상호작용이 아닌 물리적인 환경, 상황(context)등을 시스템이 인식하고 이를 기반으로 사용자와의 상호 작용을 지원하는 상황 인식 기술이 필수적인 요소로 자리 잡고 있다. 또한 상황 인식 서비스는 다양한 상황 정보를 수집 및 해석을 통해 인식하고, 추론과정을 거쳐 기존의 수동적인 서비스에서 벗어나 사용자 명령 없이도 자동으로 실행되는 지능형 서비스를 지원하며, 각 사용자에게 맞춰진 개인화된 서비스 등을 제공한다. 상황 인식 서비스는 의료, 교육, 재난, 구호, 쇼핑 등 사회 전 분야에 걸쳐 응용될 수 있어 사회 전반에 걸쳐 많은 영향을 줄 것이다[4,5].

기존의 상황 인식 서비스 기반 기술 중 상황에 대한 체계적인 관리를 위한 온톨로지 기술과 사용자 맞춤형 서비스 제공을 위한 추론 기술에 대한 연구는 어느 정도 진행되고 있으나 사용자 및 주변 환경으로부터 입력되는 불확실하거나 모호한 상황 정보에 대한 표현과 추론에 대한 연구는 부족한 실정이다[4,5,6,7,8].

본 논문에서는 상황 인식 모델링 및 센싱에 필요한 기반 기술과 온톨로지 기반 기존 연구에 대해 알아보고 상황 인식을 위한 확률 확장 온톨로지 모델을 설계한다. 제안한 모델은 사용자의 행동 및 환경의 변화에 따른 센서 입력 데이터를 바탕으로 온톨로지를 구성하고 다양한 종류의 모호한 상황(Uncertain Context)을 위해 확률 정보를 온톨로지 내에 표현 및 추론할 수 있도록 베이저안 네트워크를 이용하여 온톨로지 모델을 확장 설계한다.

II. 관련 연구

이 장에서는 유비쿼터스 컴퓨팅에서 필수적으로 요구되는 상황 인식 관련 기술과 모델링 방법, 상황 인식 관련 기존 연구, 불확실한 데이터의 확률적인 추론을 제공하는 베이저안 네트워크에 대해서 기술한다.

2.1 상황 인식 개요

상황에 대한 다양한 정의가 있지만 사용자와 유비쿼터스 컴퓨팅 환경 사이의 관계와 연관 지어지는 사용자 주위의 상황이나 상태(circumstance) 또는 객체(object)들에 대한 정보를 통칭한다[4]. 상황의 본질적인 정의는 “실세계(Real World)에 존재하는 실체(Entity)의 상태를 특정화하여 정의한 정보”라고 정의할 수 있으며, 여기서 실체란 인간, 장소 또는 사람과 서비스간의 상호 작용을 의미한다고 할 수 있다.

Schilit와 Theimer는 그들의 연구에서 “상황 인식”이란 용어를 처음으로 사용하였는데, 그 연구에서는 상황을 장소, 사람이나 사물들을 구별 짓는 특징인 아이덴티티(identity), 사람이나 사물들을 포함하는 환경의 변화 등으로 설명한다[5]. Dey는 상황을 사용자가 속해 있는 환경 내에서 사용자의 감정적인 상태, 주의력, 위치와 방향, 날짜와 시간, 사람과 사물 등으로 정의한다[6].

위와 같이 관점에 따른 상황 정의에 약간의 차이가 있으나 일반적인 상황 정보는 사용자 상황, 물리적 환경 상황, 컴퓨팅 시스템 상황, 사용자-컴퓨터 상호 작용 이력, 기타 상황으로 분류할 수 있으며 사용자의 현재 상황에 따라 적절한 정보 혹은 서비스를 제공하기 위해 상황을 이용하는 것을 상황인식(Context-Awareness)이라 한다[4,7].

상황 모델링은 상황 정보에 대한 높은 수준의 추상적 개념을 제공하기 위해 필요하며 센서와 액추에이터(actuator)의 추상화를 제공함으로써, 개발자가 다양한 하드웨어 장치와 인터페이스하는 부담을 줄일 수 있다. 이러한 상황 모델링 기법은 각 시스템에서의 상황 정보 교환을 위해 사용되는 데이터 구조에 관한 스키마에 의해 다음의 네 가지로 분류된다[4,5,6,7]. 키-값(Key-Value) 모델, 마크업 스키마(Markup Scheme) 모델링, 객체 지향 모델링 기법, 온톨로지 기반 모델링 등이 있다. 온톨로지 기반 모델링에서 온톨로지는 개념과 상관 관계를 기술하는 도구로 상황 정보를 표현하고 공유하기 위한 어휘 및 용어를 제공하며 상황 정보의 다양성 때문에 특정 도메인별 온톨로지(domain specific ontology)를 정의하고 있다. 이 모델은 유비쿼터스 컴퓨팅 환경에서 상황 지식의 공유와 재사용을 지원하며, 또한 계층적 상황 온톨로지 모델에서는 상위 계층의 온톨로지를 이용하여 도메인에 맞는 하위 계층의 온톨로지를 생성해 낼 수 있다.

2.2 상황 인식 서비스 미들웨어

현재 상황 인식과 관련된 연구는 상황 인식 미들웨어와 미들웨어를 이용한 서비스 즉, 응용으로 분류할 수가 있다.

상황 인식 서비스 미들웨어와 관련된 연구에서는 상황인식 응용 개발에 필요한 공통 기능을 응용 레벨에서 분리하여 미들웨어 형태로 개발자에게 제공하는 방안이 연구되고 있으며, 개발자에게는 응용과 관계된 기능에만 집중할 수 있도록 하고, 사용자에게는 일반화된 응용 서비스를 제공한다.

이 절에서는 온톨로지 기반의 상황 인식 미들웨어에 관련된 대표적인 연구에 대해 요약하였다.

CoBrA(Context Broker Architecture)[7]는 유비쿼터스 환경에서 상황 정보의 생성, 관리, 배포를 위한 아키텍처로 공간상에 존재하는 모든 컴퓨팅 객체들을 위한 상황 공유 모델을 관리하는 broker agent 개발을 목표로 하고 있다. Policy language를 통해 사용자 및 기기의 상황 정보에 대한 접근 관리를 제공하며 상황을 RDF와 OWL[9]을 기반으로 온톨로지로서 구성하여 지식 공유 및 기본적인 추론이 가능하도록 설계 하였다. SOCAM(Service Oriented Context-aware Middleware)[10]은 유비쿼터스 환경에서 제공 되어지는 다양한 상황의 상호의존적 개념에 대한 정의 및 모바일 환경에서의 상황인식 서비스 제공을 위해 구현되었으며 OWL 기반의 온톨로지를 사용하여 의미기반 상황 표현과 다양한 형태의 상황에 대한 추론, 지식 공유, 상황의 분류와 상호 의존성과 관련된 문제를 다루고 있다. Gaia[11]는 응용이 다양한 상황정보를 얻고 추론할 수 있게 해주며, 상황 처리를 위해 논리 추론과 기계 학습 방법이 폭넓게 활용되며, 서로 다른 유비쿼터스 컴퓨팅 환경뿐만 아니라 이종 에이전트간의 시맨틱한 상호 운용성을 보장하기 위해서 DAML+OIL로 기술된 온톨로지를 사용한다. Gaia의 모든 에이전트는 CORBA 환경 위에서 작동되며, 이를 위해 CORBA의 다양한 미들웨어 서비스를 활용하고 있다. 인하대의 DyCAM에서는 유비쿼터스 환경에서 서비스의 이동성을 효과적으로 지원하기 위해서 상황 정보의 동적 관리 및 서비스간의 상호 작용을 위한 서비스 검색 및 조합, 서비스 이동성을 지원하는 모델 및 미들웨어를 제시했으며 OSGi 프레임워크위에서 구현함으로써 UPnP, Jini 등의 표준 인터페이스 기술을 이용할 수 있도록 했다[12][13].

상황 인식 응용 서비스로 MicroSoft의 Easy Living, AwareHome, Musex등 현재 상황 인식 서비스에 대한 다양한 연구가 진행되고 있으나 사용자와 환경의 상황을 반영한 사용자간에 발생하는 서비스 충돌 해결 및 불확실하고 모

호한 상황 데이터의 처리에 대한 연구가 부족한 실정이다[8].

2.3 베이지안 네트워크

베이지안 네트워크는 인공지능분야에서 확률과 불확실성을 표현하는 수단으로 많이 사용되고 있는 확률 기반 추론을 제공하는 방법 중 하나이다. 이러한 특징으로 베이지안 네트워크는 불확실한 상황에서 통계적 추론을 이용한 의사 결정에서 좋은 결과를 내고 있다[14]. 그 적용 분야로는 확률기반 전문가 시스템, 의학분야에서 진단, 언어 이해 분야, 비전 분야, 휴리스틱 검색등이 있다[15].

베이지안 네트워크는 DAG(directed acyclic graph, DAG)로서, 네트워크를 구성하는 각 노드는 확률변수를 나타낸다. 각 노드는 다수의 상태를 가질 수 있으며, 한 노드는 여러 개의 속성을 가질 수 있고 각 상태에 대한 확률 값의 합은 1이 된다. 노드와 노드를 연결하는 아크(arc)는 부모에서 자식으로의 인과관계를 나타내며 변수의 확률적 인과관계로 네트워크를 구성하고 특정 조건이나 증거가 주어진 경우의 확률 즉, 조건부 확률 테이블(conditional probability table)을 가지고 베이지안 규칙(Bayes's rule)을 이용해 결과를 추론한다. 베이지안 네트워크에서 추론한다는 것은 일반적으로 관측 가능한 노드의 증거(evidence)가 설정된 상태에서 추론의 대상이 되는 노드인 쿼리 노드(Query node)의 각 상태에 대한 확률을 계산하는 것을 의미한다.

X의 부모(상위) 노드들은 parent(X)로, 자식(하위) 노드들은 children(X)라고 하면, 각 부모 노드의 초기 확률 값은 P(Parent Node)로, 자식 노드의 확률은 조건부 확률인 P(Child/Parent)로 나타낼 수 있다.

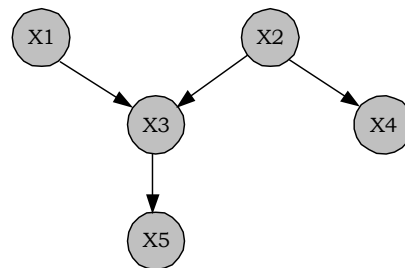


그림 1. 간단한 베이지안 네트워크 구조의 예
Fig 1. Bayesian Network Structure

그림 1에서 네트워크가 조건부 독립이라고 가정하고 체인규칙을 이용하면, 네트워크를 구성하는 각 노드에 대한 결합 확률(joint probabilistic) $P(x_1, x_2, x_3, x_4, x_5)$ 은 식 2와 같이 나타낼 수 있다.

$$P(x_i|x_j) = \frac{P(x_i|x_j)P(x_j)}{P(x_j)} \dots\dots\dots (식1)$$

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1, x_2)P(x_4|x_2)P(x_5|x_3) \dots\dots\dots (식2)$$

일반적인 노드들 사이의 확률 분포는 식 3과 같이 나타낸다.

$$P(x_1, x_2, x_3, \dots, x_n) = \prod_{i=1}^n P(x_i | Parent(x_i)) \dots\dots\dots (식3)$$

또한, 베이지안 네트워크에서의 추론은 식 4에 의해 이루어지며 정밀 추론(exact inference)과 근사 추론(approximate inference) 방법이 있다.

$$P(X_i | X_j) \propto \sum_{k \neq i, j} P(X_i | X_j, X_k) \dots\dots\dots (식4)$$

여기서 X_j 는 관측된 변수들의 집합을 의미하며, X_i 는 우리가 추정하고 싶은 숨겨진 변수들의 집합을, X_k 는 관계 없는 숨겨진 변수들을 의미한다.

III. 확률 확장 온톨로지 모델

온톨로지를 기반으로 상황을 모델링함으로써 쉬운 추론 및 패턴 분석, 사람과 기계, 기계사이의 원활한 커뮤니케이션, 확장성 및 상황정보 공유의 용이성, 다양한 응용 서비스 관리의 편리함을 얻을 수 있다. 그러나 현재 상황인식 시스템에서 사용되는 로직 및 룰 기반 추론 메카니즘은 모호하고 불확실한 데이터에 대한 처리를 지원하지 않으므로 모호한 상황 데이터에 대한 관리 및 추론을 위해 기존의 온톨로지 모델을 확장한 확률 확장 온톨로지 모델을 제안한다. 또한 상황 온톨로지 모델을 계층적으로 구성하여 상위 계층의 공통 온톨로지를 상속받아 특정 도메인에 적합한 (domain-specific) 하위 계층의 온톨로지를 생성해 낼 수 있는 계층 상황 온톨로지 관리 방법을 제안한다.

3.1 온톨로지 기반 상황 모델

상황은 특정 시간에 센서로부터 수집된 여러 속성들이 일정 영역에 포함되는지에 따라서 결정되므로 시스템은 센

서로부터 받은 입력 데이터 값을 통해서 상황을 판단할 수 있는 방법을 가져야 한다. 식 5는 이 논문에서 설계하는 미들웨어에 연결되어 있는 센서들로부터 특정시간 (t)에 전달 받은 속성들의 집합을 표현한 것이다.

$$S(t) = \sum_{i=1}^n A_i(t) \dots\dots\dots (식 5)$$

위 식에서 $A_i(t)$ 는 특정 시간 t에 속성 값을 나타낸 것이다. 상황 속성은 상황을 추론하는데 사용되어질 수 있는 요소들을 기술하는데 사용된다. 상황 속성은 가상 또는 물리적인 센서들과 연결되어 있다.

온톨로지는 개념(concept)과 관계(relation)들로 구성된 사전으로서 특정 도메인에 관련된 객체들을 계층적 구조로 표현하고 추가적으로 이를 확장할 수 있는 추론 규칙을 포함한다. 유비쿼터스 환경에서는 모든 객체들과 객체의 속성들을 특정 종류의 도메인 안에 계층적으로 표현해야 하고, 이벤트 상황에 따라서 속성 값들을 변화시키는 규칙이 필요하므로 온톨로지를 사용하여 유비쿼터스 환경내의 모든 객체들을 계층적으로 구성, 관리해야 한다.

온톨로지 기반 상황 모델은 RDF triple의 집합인 OWL로 구성되며, 상황은 1차술어논리(first-order predicate calculus)로 표현된다. 기본 모델의 상황 표현은 다음과 같은 형식으로 표현된다.

$$\begin{aligned} & \text{Predicate}(\text{subject}, \text{value}) \\ & \text{subject} \in S^* : \text{set of subject names} \\ & \text{Predicate} \in V^* : \text{set of predicate names} \\ & \text{value} \in O^* : \text{set of all values of subject} \in S^* \end{aligned}$$

subject는 person이나, room 또는 device 같은 object등이 될 수 있고, Predicate(or Property)는 location in과 has status, value는 living room, open, numeric value등이 될 수 있다. 예를 들면, Location(Jung, bathroom)은 Jung은 bathroom에 있고 Temperature(bedroom, 24)은 bedroom의 온도가 24°C, Status(door, open)은 door 상태가 open이다.

그림 2는 이 논문에서 설계한 계층적 상황 온톨로지를 표현한 것이다. 미들웨어상에서 관리되는 상황 정보는 상위 계층의 공통 상위 온톨로지 상황 정보와 도메인별 특성에 맞게 상위 온톨로지를 상속하여 별도로 정의되는 하위 도메인 상황 정보로 구성되며, 상황 및 상황의 속성은 웹 온톨로지 언어인 OWL로 온톨로지를 정의한다. 서비스가 미들웨어에 새롭게 배치되는 경우에는 미들웨어는 기존의 도메인 상

황 정보를 상속받아 서비스에서 필요로 하는 개별 상황 정보를 추가하여 운용할 수 있어야 한다.

공통된 상위 상황 정보는 상황 인식 응용에서 필요로 하는 기본 요소를 최상위 클래스인 Context 클래스에서 상속받아 실행객체(RunEntity), 위치(Location), 사람(Person), 서비스(Service), 행위(Activity) 클래스로 정의하고 도메인별 상황 정보는 상위 클래스의 정보를 상속받아서 설계한다.

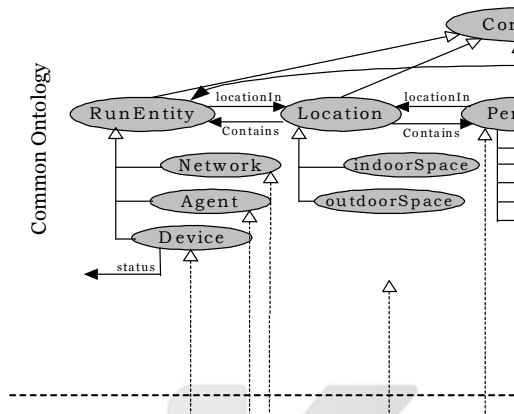


그림 2. 계층 상황 온톨로지 구성
Fig 2. Hierarchical Context Ontology Structure

각 상황 클래스는 해당 상황을 설명하기 위한 다수의 속성을 포함할 수 있으며, 이 속성은 다른 클래스와의 관계(relation)를 나타낼 때도 사용되며, 다른 상황을 프로퍼티로 포함할 수도 있다. 예를 들어서, 홈 네트워크 응용 온톨로지 에서 Location 클래스로부터 상속받은 indoorSpace의 Room 상황과 RunEntity의 하위 클래스인 Device 클래스로부터 상속받은 Temperature 상황이 있는 경우 Temperature 상황은 Room 상황의 프로퍼티로 이용될 수 있다.

다음은 OWL로 표현한 상위 상황 온톨로지의 일부분이다.

```
<owl:Ontology rdf:about="">
  <rdfs:comment>CommonOntology</rdfs:comment>
</owl:Ontology>
<owl:Class rdf:ID="#Context">
  <rdfs:label>Context</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="#RunEntity">
  <rdfs:label>RunEntity</rdfs:label>
```

```
<rdfs:subClassOf rdf:resource="#Context"/>
</owl:Class>
<owl:Class rdf:ID="Location">
  <rdfs:label>Location</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Context"/>
</owl:Class>
<owl:Class rdf:ID="Person">
  <rdfs:label>Person</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Context"/>
</owl:Class>
<owl:Class rdf:ID="Activity">
  <rdfs:label>Activity</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Context"/>
</owl:Class>
...
<owl:Class rdf:ID="Device">
  <rdfs:label>Device</rdfs:label>
  <rdfs:subClassOf rdf:resource="#RunEntity"/>
</owl:Class>
</rdf:RDF>
```

3.2 확률 확장 온톨로지 모델

OWL은 확률 정보의 표현을 지원하지 않기 때문에 모호한 상황 정보를 표현하고 추론하기 위해 온톨로지를 구성하는 클래스의 프로퍼티 의존성을 이용하여 베이지안 네트워크를 구성하고, 온톨로지 기반 모델에서 확률 정보를 나타낼 수 있는 클래스와 프로퍼티를 정의하여 상황 모델을 확장한다.

3.2.1 확률 확장 모델

온톨로지 기반 상황 모델에서 각 클래스의 프로퍼티 간의 인과관계(casual relationship)를 기반으로 서브네트워크(sub-network)를 구성한다. 이러한 클래스의 프로퍼티 사이의 인과관계는 dependOn 속성으로 나타내며 클래스들은 확률을 나타내는 속성과 그밖에 일반적인 데이터를 나타내는 속성들로 구성될 수 있으며 온톨로지의 속성 값의 의존적 관계에 따라 계층적인 네트워크를 구성할 수 있다. 계층적 네트워크는 상위의 온톨로지 레벨과 상위 레벨의 상황 변수간의 의존성을 바탕으로 생성할 수 있는 하위의 의존 레벨(dependent level)로 구성된다.

그림 3은 온톨로지 모델에서 베이지안 네트워크 구성을 위한 클래스 사이의 의존 관계를 통하여 유효한 2개의 서브

네트워크의 구성을 보여 주고 있다.

클래스 X의 프로퍼티 중에 다른 클래스의 프로퍼티들과의 의존성을 나타내는 dependOn 프로퍼티가 클래스 Y의 프로퍼티 중 어느 하나 이상의 프로퍼티와 의존성이 있을 경우 클래스 Y의 프로퍼티와의 확률을 계산하여 확률 의존성이 없으면 노드를 구성하지 않고, 확률이 존재하면 두 프로퍼티를 노드로 하여 아크를 연결한다. 계속해서 다른 클래스의 프로퍼티 종속성을 이용하여 베이지안 네트워크를 구성해 나간다.

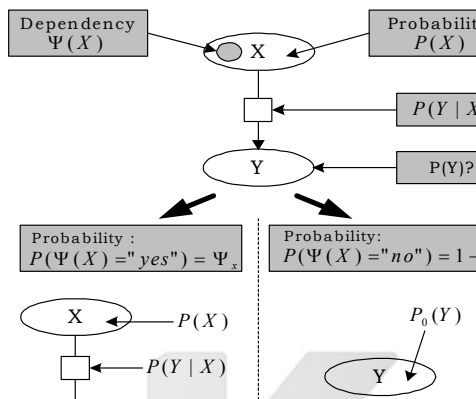


그림 3. 베이지안 네트워크 구성 : 프로퍼티 의존성에 기반한 서브네트워크
Fig 3. Sub Network based on property's dependency

여기에서,
 attributes
 X : set of values $\{x_1, x_2, x_3 \dots x_n\}$,
 attributes
 Y : set of values $\{y_1, y_2, y_3 \dots y_n\}$,
 probabilities : $P(X) = P(Y|X)$
 depend property : $\psi(X) = yes$, X가 의존적 일 때
 $\psi(X) = no$, X가 의존적이지 아닐 때
 depend value : $\psi_x = P(\psi(X) = yes)$

X와 $\psi(X)$ 가 주어졌을 때 Y의 조건적 확률은 식 6과 같다.

$$P(Y|X, \psi(X)) = \begin{cases} P(Y|X); \psi(X) = yes \\ P_0(Y); \psi(X) = no \end{cases} \dots\dots (식 6)$$

여기에서, $P_0(Y)$ 는 Y의 사전 확률 분포(prior

probability distribution)이다.

베이지안 추론을 위한 확률 P_0 는 식 7에 의해 계산된다.

$$P_0(Y=y_j) = \frac{1}{n} \cdot \sum_{i=1}^n P(Y=y_j|X=x_i) \dots\dots\dots (식 7)$$

여기에서, n은 X값의 개수이며, Y의 확률은 식 8과 같이 나타낼 수 있다.

$$P(Y) = \frac{1}{n} \cdot \sum_x (P(Y|X) \cdot (n \cdot \psi_x \cdot P(X) + (1 - \psi_x))) \dots\dots\dots (식 8)$$

3.2.2 확률 확장 마크업 엘리먼트

클래스의 프로퍼티들 사이의 확률적 종속성을 표기하기 위한 마크업 엘리먼트를 다음과 같이 정의한다.

표 1. 확률 확장 마크업 엘리먼트
Table 1. Probabilistic Annotation Markup elements

Class	name	Type	Description
PriorProb	hasPVariable hasPValue	Object Property Value Property	사전 확률 값을 갖는 노드명칭 사전 확률 값
CondProb	hasCVariable hasCond hasCValue	Object Property Object Property Value Property	조건적 확률값을 갖는 노드명칭 조건적 확률을 가지는 부모 노드의 명칭 부모노드의 조건적 확률 값
PriorProb CondProb	dependOn	Object Property	다른 property와의 의존성 표현
All	prob(nTriple)	Value Property	Class의 Predicate에 확률 값을 표현

PriorProb는 확률 정보를 가지고 있는 클래스로 프로퍼티와 제약사항(restriction)를 가지고 있는 일반적인 클래스와 유사하며 확률 정보를 포함하지 않는 고정적이고 독립적인 값을 가지는 프로퍼티와 확률 정보를 가지고 있는 프로퍼티로 구성된다. hasPVariable는 확률 정보를 가지고 있는 프로퍼티로 베이지안 네트워크에서 루트 노드에 대응하는 프로퍼티 hasPValue를 사용하여 확률 분포 값을 지정한다. CondProb는 조건적 확률정보를 가지고 있는 클래스로 hasCVariable는 베이지안 네트워크에서 중간 또는 자식노드에 대응되며 조건적 확률 노드의 값은 프로퍼티 hasCValue를 사용하여 부모 노드들과의 조건적 확률 의존성을 표시한다. Prob(Predicate(subject,value))는 각 각

의 class의 predicate에 확률 값을 표현하고 확률 값은 0 과 1사이의 값을 가지며 어떤 상황정보(sensed contexts, defined contexts, derived contexts)에도 적용될 수 있다. Prob(Status(Jung,Showering))=0.9은 Jung이 현재 샤워하고 있을 확률이 0.9라는 것을 의미한다.

온톨로지에 RDF triples 형태의 클래스 프로퍼티 A,B,C가 있을 때 사전 확률(prior probability) P(A)는 클래스 PriorProb의 인스턴스로 정의되며, 조건적 확률은 P(A|B,C,D)는 클래스CondProb의 인스턴스로 정의된다. P(A)=0.8의 확률적 마크업을 사용한 OWL 표현은 다음과 같다.

```
<inha:PriorProb rdf:ID="A">
<inha:hasPVariable><rdf:value>A</rdf:value>
</inha:hasPVariable>
<inha:hasPValue>0.8</inha:hasPValue>
</inha:PriorProb>
```

P(A|B,C,D)=0.6의 확률적 마크업을 사용한 OWL 표현은 다음과 같다.

```
<inha:CondProb rdf:ID="ABCD">
<inha:hasCond><rdf:value>B</rdf:value>
</inha:hasCond>
<inha:hasCond><rdf:value>C</rdf:value>
</inha:hasCond>
<inha:hasCond><rdf:value>D</rdf:value>
</inha:hasCond>
<inha:hasCVariable><rdf:value>A</rdf:value>
</inha:hasCVariable>
<inha:hasCValue>0.6</inha:hasCValue>
</inha:CondProb>
```

dependOn은 종속성 정보를 마크업하며 종속성은 상황 정보의 중요한 특성으로 한 객체 및 데이터 타입과 연관된 다른 객체와의 의존성을 프로퍼티 종속성으로 나타내며 모호한 상황에 대한 추론을 위해 확률과 베이지안 네트워크의 통합 시 사용된다.

3.3 베이지안 네트워크로의 구조적 변환

상황 정보를 이용하여 베이지안 네트워크로 구조적 변환을 할 경우 서로 다른 상황 이벤트 사이의 인과 관계를 기초로 하여 변환되며 구조적인 변환 규칙은 다음과 같다.

상황 온톨로지를 표현하는 RDF 그래프는

rdfs:dependOn 프로퍼티를 이용하여 베이지안 네트워크의 DAG로 변환 된다. 클래스의 프로퍼티는 베이지안 네트워크의 노드(node)로 매핑되고 다른 클래스의 프로퍼티와 의존성이 존재하는 노드 사이에 방향성 아크가 그려진다.

베이지안 네트워크를 구성한 후, 여러 노드의 조건적 확률(conditional probability)을 얻기 위하여 실제 데이터로 학습한다. 베이지안 네트워크가 일단 학습되면 컨텍스트의 이벤트와 컨텍스트 조건적 확률을 추론하는데 사용하며 CPT(conditional probability table)을 이용하여 베이지안 네트워크의 확률분포(probability distribution)를 계산한다.

데이터 집합의 학습으로 모든 루트 노드의 확률과 자식 노드들의 조건적 확률을 얻을 수 있어 다양한 컨텍스트 이벤트의 확률 분포를 얻을 수 있다. 이러한 확률은 액세스와 쿼리를 위해 상황 온톨로지에 저장되며 모호한 상황처리를 위하여 서비스 개발자는 미리 정의된 룰(pre-defined rules)에 의해 트리거(triggered)되는 액션을 명시해야 한다. 룰은 컨디션의 집합과 모호한 상황 정보를 포함한다.

아래 그림 4는 도메인 온톨로지로부터 실제 상황 온톨로지를 정의하고 베이지안 네트워크로의 변환을 보여주고 있다.

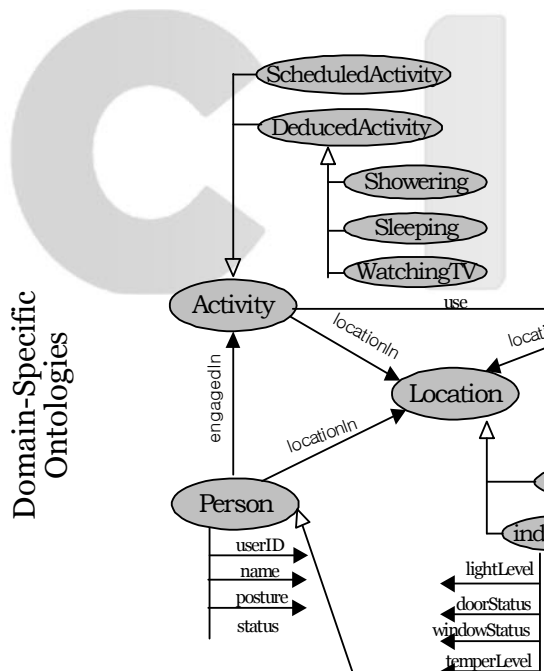


그림 4. RDF 그래프의 베이지안 네트워크 변환
Fig 4. Translate RDF graph to Bayesian Network

통합된 상황 온톨로지 모델은 확률 추론을 위해 사용되

는 확률 지식과 구조적인 지식을 표현할 수 있다. 상황 온톨로지 모델은 공통 온톨로지와 도메인에 특성에 맞는 온톨로지 로 구성되며 도메인 온톨로지는 실제 상황 온톨로지와 프로퍼티 사이의 조건적 확률 의존성을 표시하는 확률 모델로 구성된다.

IV. 상황 추론

설계한 확률 확장 온톨로지 모델에서 도메인 내의 상황 정보를 추론하는 방법으로 온톨로지 추론(ontology reasoning), 규칙기반 추론(rule-based reasoning), 베이 지안 추론을 지원한다.

4.1 온톨로지 추론

상황에 대한 추론은 1차 술어 논리 기반의 규칙 기반 추론을 하며 온톨로지 리스너(reasoner)와 규칙 기반 리스너는 Jena API를 기반으로 구현했다. 온톨로지 추론은 도메인 온톨로지가 전환(switching)되거나 통합될 때 내부 상황 클래스의 일관성과 그들 간의 함축된 관계등을 찾아내는데 사용된다.

온톨로지 추론은 RDFS 추론과 OWL 추론을 포함하며 다음은 온톨로지 추론의 종류를 나타낸 것이다.

표 2. 온톨로지 추론 규칙
Table 2. Ontology inference rules

Constructs	Ontology inference rules
rdfs:subClassOf	(?A rdfs:subClassOf ?B), (?B rdfs:subClassOf ?C) ⇒ (?A rdfs:subClassOf ?C)
owl:TransitiveProperty	(?P rdf:type owl:TransitiveProperty), (?A ?P ?B), (?B ?P ?C) ⇒ (?A ?P ?C)
rdfs:subPropertyOf	(?A rdfs:subPropertyOf ?B) ∧ (?B rdfs:subPropertyOf ?C) ⇒ (?A rdfs:subPropertyOf ?C)
owl:disjointWith	(?A owl:disjointWith ?B) ∧ (?X rdf:type ?C) ∧ (?Y rdf:type ?D) ⇒ (?X owl:differentFrom ?Y)
owl:inverseOf	(?A owl:inverseOf ?B) ∧ (?A ?X ?Y) ⇒ (?Y ?B ?X)

만약 현재 'Jung'이라는 사용자가 BathRoom에 있다는 상황 정보가 발생하면 온톨로지 TransivieProperty 규칙을 이용한 상황 추론을 통해서 BathRoom이 indoorSpace의 부분임을 유추함으로써 'Jung'이 집안에 있다는 것을 알 수 있다.

아래의 OWL 표현은 프로퍼티 locationIn의 전이 프로퍼티를 나타낸다.

```
<owl:ObjectProperty rdf:ID="locateIn">
  <rdf:type rdf:resource=
    "http://www.w3c.org/2002/07/owl#TransitivePro
    perty">
  <rdfs:range rdf:resource="#Location"/>
</owl:ObjectProperty>
<inha:Room rdf:ID="BathRoom">
  <inha:locateIn rdf:resource="#indoorSpace"/>
</inha:Room>
```

4.2 규칙 기반 추론

사용자 정의 규칙 기반 추론은 순방향 체이닝(forward chaining)방법을 사용하며 다음과 같은 사용자 정의 규칙 셋(rule-set)에 의하여 사용자가 취침중이라는 것을 추론할 수 있다.

표 3. 사용자 정의 추론 규칙
Table 3. User defined Inference rules

Context	User defined Inference Rules
Sleeping	(?p locatedIn Bedroom), (Bedroom lightLevel LOW), (Bedroom noiseLevel LOW) ⇒ (?p status Sleeping)
Showering	(?p locatedIn Bathroom), (Bathroom lightLevel HIGH), (Bathroom noiseLevel High), (WaterHeater status ON), (Bathroom DoorStatus CLOSED) ⇒ (?p status Showering)
WashingFace	(?p locatedIn Bathroom), (Bathroom lightLevel HIGH), (Bathroom noiseLevel NORMAL) ⇒ (?p status WashingFace)
WatchingTV	(?p locatedIn LivingRoom), (TV status ON) ⇒ (?p status WatchingTV)

4.3 확률 기반 추론

인스턴스의 프로퍼티에 대해 질의하기 전에 도메인 내 상황 온톨로지의 확률 확장 모델에 대응하는 하나 이상의 베이 지안 네트워크를 생성하고 베이 지안 리스너는 모든 노드의 확률을 추론하여 온톨로지의 인스턴스의 프로퍼티를 직접 수정한다. 그림 4의 상황 온톨로지를 통해 사용자의 현재 상태 정보 및 환경의 현재 상태를 나타내는 사용자의 위치, WaterHeater의 동작 여부, BathRoom의 조명, BathRoom 문의 상태에 대한 증거(evidence)를 알 수 있으므로 대응하

는 프로퍼티의 확률을 세팅하고 온톨로지를 통해 베이지안 네트워크를 구성하고 모든 노드에 대해 확률을 추론 한 후 각 노드의 추론된 확률 값을 인스턴스 프로퍼티에 다시 반영한다.

확률 추론을 위한 도메인 온톨로지는 그림 4의 구성과 같이 Person은 "Jung"이며, Person의 프로퍼티는 name, status, locatedIn으로 구성하고, Location은 "indoorSpace"의 내부에 "BedRoom", "LivingRoom", "BathRoom"으로 구성된다. Device는 "TV", "WaterHeater", "Light"등을 가지며, 디바이스 및 환경 정보를 사용하는 사용자의 활동은 "Showering", "Sleeping", "WatchingTV", "WashingFace"등이 있다.

그림 5는 사용자가 BathRoom안에 있을 때, WaterHeater, light, door, noiseLevel등과 같은 환경의 상황 정보를 가지고 현재 사용자의 상태에 대한 확률을 계산하여 사용자의 행동을 추론하는 것을 보여주고 있다.

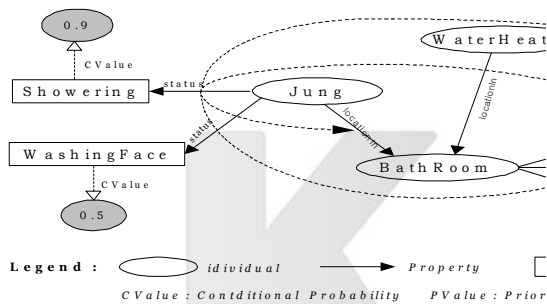


그림 5. 확률 추론 과정
Fig 5. Probabilistic inference Process

사용자가 BathRoom안에 존재할 때 doorStatus가 Closed 되어 있어 확률 값은 1.0으로 결정되고, noiseLevel High에 대한 확률 값은 0.9로 되어있어 Showering의 추론된 확률이 WashingFace의 확률 보다 높아 사용자의 현재 상태가 Showering이라는 것을 알 수 있다.

V. 결론

본 논문에서는 유비쿼터스 컴퓨팅 환경에서 모호하거나 불확실한 상황에 대한 관리와 추론을 지원하는 확률 확장 계층적 온톨로지 기반의 상황인식 모델을 제시하였으며 확률 기반 추론을 통하여 제시한 모델이 유효성을 입증했다. 제시한 인식 모델은 모호한 상황 정보를 표현하고 추론하기 위해 상황 클래스의 프로퍼티 의존성을 이용하여 베이지안

네트워크를 구성하고, 기존 온톨로지 모델에서 확률 정보를 나타낼 수 있도록 설계하였으며 확률 정보를 나타낼 수 있도록 클래스와 프로퍼티를 정의하여 상황 모델을 확장 설계 하였다.

또한, 온톨로지 기반 상황 모델은 공통 상위 온톨로지를 상속받아 응용에 적합한 하위 온톨로지를 생성할 수 있으며 동적 상황 변화에 대해 유연하게 인식하고 변경할 수 있다.

이 논문에서 제시한 확률 확장 온톨로지 모델은 홈 네트워크, 스마트 오피스 등의 실내 환경 및 다양한 유비쿼터스 환경에서 동적 상황 인식 미들웨어 개발 시에 상황 인식 모델로 적용될 수 있을 것으로 보인다.

향후 연구 방향은 불확실한 상황 처리를 더욱 효과적으로 하기 위해 퍼지논리나 결정네트워크등과 같은 다른 방법과의 통합에 대한 연구이다.

참고문헌

- [1] M. Weiser and J. S. Brown, The Coming Age of Calm Technology, In P. J. Denning & R. M. Metcalfe (Eds.), Beyond Calculation: The Next Fifty Years of Computing, pp. 75-85, 1998.
- [2] A. D. Norman, The Invisible Computer: Why Good Products Can Fail, The Personal Computer Is So Complex, and Information Appliances Are the Solution, MIT Press, 1998.
- [3] J. Weichert, "The Disappearing Computer,"Information Document, IST Call for proposals, European Commission, Future and Emerging Technologies, 2000.
- [4] Dey, A.K., et al. " A conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications" anchor article of a special issue on Context-Aware Computing, Human-Computer Interaction (HCI) Journal, Vol. 16, 2001
- [5] B. N. Schilit, N. Adams, and R. Want, "Context-aware computing applications," Proceedings of the Workshop on Mobile Computing System and Applications, pp. 85-90, 1994.

[6] A. K. Dey, "Supporting the Construction of Context-Aware Applications," Dagstuhl seminar on Ubiquitous Computing, 2001.

[7] Chen, Harry, Tim Finin, and Anupam Joshi. "An Intelligent Broker for Context-Aware Systems." Adjunct Proceedings of UbiComp 2003, Seattle, Washington, USA, October 12-15, 2003.

[8] B. A. Truong, Y. K. Lee, S. Y. Lee, "Modeling and Reasoning about Uncertainty in Context-Aware Systems", ICEBE, pp.102-109, 2005.

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

[10] Tao Gu, Hung KP, Da QZ, "A Service-oriented middleware for building context-aware services", Journal of Network and Computer Applications, Vol.28, 2005

[11] M. Roman, C. K. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt. "Gaia: A Middleware Infrastructure to Enable Active Spaces", In IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.

[12] 이승근, 김영민, "계층적 상황 온톨로지 관리를 이용한 상황 인식 서비스 미들웨어 설계", 한국컴퓨터정보학회 학회지, 제11권 제1호, p186-193. 2006.

[13] 이승근, "유비쿼터스 컴퓨팅 환경을 위한 온톨로지 기반 상황 인식 서비스 미들웨어", 인하대학교 박사학위 논문, 2006.

[14] F. V. Jenesn, "Introduction to Bayesian Networks", New York: Springer Verlag, 1996

[15] Eugene Charniak, "Bayesian Networks without tears", American Association for Artificial Intelligence, pp 50-63, 1991.

저자 소개



정헌만

1996년 2월 서울산업대학교
전자계산공학과
2001년 2월 인하대학교 전자계산공
학과 (공학석사)
2004년 2월 인하대학교
컴퓨터정보공학과
(박사수료)
2001년~ 경인여대, 인하전문대
현재 강사
관심분야 상황인식, 시맨틱웹,
웹서비스, 유비쿼터스
센서네트워크



이정현

1977년 2월 인하대학교 전자공학과
1980년 2월 인하대학교 전자공학과
(공학석사)
1988년 2월 인하대학교 전자공학과
(공학박사)
1979년~ 한국전자기술연구소
1981년 시스템 연구원
1984년~ 경기대학교
1989년 전자계산학과 교수
1989년~ 인하대학교
현재 컴퓨터공학부 교수
관심분야 HCI, 유비쿼터스
컴퓨팅