

## 사용자 중심의 상황 인지 시스템의 개발\*

장인우\*\* · 우종우\*\*\*

### Development of User-Centered Context Awareness System\*

In-woo Jang\*\* · Chong-woo Woo\*\*\*

#### ■ Abstract ■

Recently, a smart space with Ubiquitous Environment is expanding rapidly due to the development of Ubiquitous Sensor Network. Therefore, more appropriate and intelligent services of the context awareness system is being required. The previous context awareness system can provide a service to the user through the inference only on the current situation. But, it does not handle certain situation properly when the system provides abnormal result. Also it does not have any proper method of generating reliable semantic data from sensed raw data. In this paper, we are trying to solve the problems as the following approaches. First, the system recognizes abnormal result and corrects it by learning feedback from the user. Second, we suggest a method of converting sensed data into more reliable semantic data. Third, we build the system based on an Ontological context model that is capable of interoperability and reusability. Therefore, the context awareness system of our study can enhance the previous system that can generate more reliable context data, can provide more effective inference method, and can provide more intelligent system structure.

Keyword : Context Awareness, Ontology, Case-based Reasoning, Intelligence Agent

## 1. 서론

최근 활발히 연구되는 유비쿼터스 컴퓨팅의 기본 개념은 언제 어디에서나 인터넷 망을 통해 여러 자원을 접근 및 제어 할 수 있다는 것을 의미한다 [2]. 이와 같이 유비쿼터스 컴퓨팅이 가능한 공간을 스마트 공간이라 정의하며 최근의 스마트 공간은 유비쿼터스 네트워크의 발달로 인해 급격히 확장되고 있다. 상황 인지 시스템의 목적은 위와 같은 스마트 공간 내의 사용자의 의도를 파악하여 시스템이 자동적으로 사용자에게 적합한 서비스를 제공하는 것에 있다. 또한 상황 인지 시스템은 스스로 유비쿼터스 컴퓨팅 자원을 사용할 수 없는 사용자 및 자동화된 서비스를 요구하는 사용자를 대상으로 한다. 이러한 시대적 흐름 및 요구를 바탕으로 스마트 공간에서의 상황 인지 시스템의 개발은 가속화되고 있다[1].

상황 인지 시스템의 구성은 크게 상황 모델링 방법과 추론 방법으로 나눌 수 있다. 상황 모델링에 관한 기존의 연구로는 Context-Toolkit[2], CoolTown [3], Karen이 제안한 ER and UML Model[4], CONON [5] 등이 연구되었으며 최근에는 재사용성 및 상호 운영성의 이유로 온톨로지를 활용하는 CONON 모델이 많이 사용되고 있다. 상황 인지 추론에 관한 기존의 연구로는 상황 모델링의 구조를 활용한 규칙 기반 추론방법이 대다수를 이루며 최근에는 Naive Bayes 알고리즘 등의 다양한 추론 기법을 활용한 연구가 진행되고 있다[6, 7, 8]. 이러한 연구에도 여전히 다음과 같은 문제점들이 존재한다.

첫째, 상황 모델에 효율적으로 상황 정보를 표현하기 위해서는 센서로부터 수집된 데이터를 시맨틱 데이터로의 변환이 필요할 뿐 아니라 데이터에 따라 필터링 및 가공 과정이 필요하다. 기존의 시스템에서는 이러한 상황 데이터들이 추론에 적합한 상황 정보로의 변환 방법이 정형화 되어 있지 않아 적합지 못한 데이터의 사용이 발생되기도 하였다. 둘째, 기존의 시스템에서는 추론의 과정이 항상 고정되어 있어서, 적합하지 못한 추론이 수행

되었을 경우 이를 수정할 수 있는 방법이 존재하지 못하였다. 따라서 사용자나 주변상황의 변화에 적절하게 대처하지 못하는 문제점이 발생된다.

본 논문에서는 기존 시스템의 구조를 다음과 같이 확장함으로써 이러한 문제점들을 개선하고자 한다. 첫째 신뢰성 있는 상황 정보의 생성을 위해 정형화된 센서 데이터 변환 알고리즘을 제안하고, 둘째, 사용자 친화적인 서비스를 제공하기 위하여 사용자의 피드백을 반영함으로써 시스템 스스로 잘못된 추론을 수정할 수 있는 방법을 제안하고자 한다.

## 2. 관련 연구

상황 인지 시스템은 상황 정보를 구성하는 상황 모델링과 구성된 정보를 기준으로 적합한 서비스를 찾아내는 상황 추론으로 구분 지을 수 있다. 본 절에서는 기존의 상황 모델링 방법론 및 시스템에 대해서 분석된 결과 및 상황 추론 방법론을 서술한다. 또한 기존의 상황 인지 시스템의 공통적인 구조를 통해 내재되어 있는 문제점을 분석한다.

### 2.1 상황 모델링(Context Modeling)

사용자 주변에 산재한 각각의 상황 정보(Context)를 취합 및 추론한 것을 사용자가 처한 상황(Situation)이라 하며 이와 같이 사용자가 처한 상황을 효율적으로 분석 및 추론할 수 있도록 각각의 상황 정보를 구성하는 방법을 상황 모델링이라 한다. 논문에서 표현하고 있는 ‘상황(Context)’이란 단어의 정의는 사용자를 둘러싼 각각의 자원 정보로써 위치, 시간 등의 물리 정보, 빛, 온도, 습도 등의 환경 정보, 사용자의 기분, 건강 상태 등의 개인적 정보 등을 통틀어 나타낸다[10, 11]. 이와 같은 상황 정보를 추론 및 분석에 효율적인 구성을 하고자 하는 상황 모델링 방법에 관한 여러 연구가 진행되었으며 그 내용은 다음과 같다.

- Attribute-Value Model : 상황 정보를 속성과 해당 속성에 대한 값으로 상황 모델을 구성하

는 모델이다. 이와 같은 상황 모델을 사용한 대표적인 시스템으로는 Context-Toolkit[2]이 존재하며 상황 모델에 저장되는 상황 정보는 Widget이라 불리는 모듈에 의해 주변 정보가 습득되고 습득된 정보는 Interpreter 모듈을 통해 해석되어 진다.

- Web-based Model : 상황 정보를 Web 기반으로 표현하여 상황 정보에 대한 접근성을 높인 상황 모델이다. Web Presence로 표현되는 형식을 통해 스마트 공간상의 정보를 사용자에게 제공하며 Web Presence를 통하여 상황 정보의 종류 및 관계에 대한 정보를 습득할 수 있다. Web-based Model의 대표적인 시스템으로는 HP Smart Space의 프로젝트의 일환으로 개발된 Cool-Town[3]이 존재한다.
- ER and UML Model : 상황 정보를 구성하는 개체 및 그 관계에 따라 표현한 상황 모델로 데이터베이스에서 사용되는 ER 및 UML 모델을 상황 모델 구축 시에 활용한 방법이다. Karen[4]이 제안한 상황 모델에서 사용되었으며 계층관계를 표현할 수 없다는 단점을 가지고 있다.
- Ontology Model : 상황 정보의 개체 표현, 개체 간의 관계표현, 개체 간의 계층적 표현이 가능한 상황 모델 방법이다. 온톨로지로 구축한 상황 모델은 이 기종 시스템 간의 상호 운용성, 구축한 모델의 재사용성, 지식의 재사용성의 장점을 가진다. 온톨로지 상황 모델 방법의 효율적인 구축을 위하여 CONON[5]에서는 상위 온톨로지와 하위 온톨로지의 구분과 같은 방법을 제시하여 여러 도메인에 쉽게 적용할 수 있는 장점이 있다. 최근에는 위와 같은 이유로 온톨로지로 구축한 상황 모델이 대부분 사용되고 있다.

## 2.2 상황 추론

온톨로지 상황 모델이 사용되지 않은 상황인지

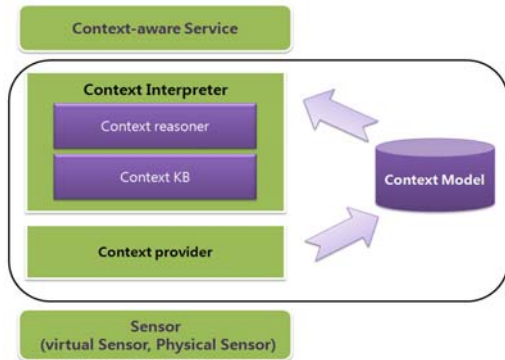
시스템의 경우, 추론 방법이 취약하며 제공하는 서비스로는 스마트 공간내의 활용할 수 있는 기기의 정보를 보여주거나, 사용자가 작성한 스케줄링에 따른 부가 서비스가 주를 이루었다[4, 5]. 그러나 온톨로지 상황 모델의 등장으로 온톨로지 구조가 제공하는 계층적 추론 및 규칙 기반 추론, 사용자의 반응을 저장한 로그 정보 학습에 따른 추론 등의 다양한 추론 방법론이 연구되고 있다. 이러한 상황 모델을 활용한 추론 방법은 아래와 같다.

- 규칙 기반 추론 : 상황 인지 시스템에서 가장 많이 사용되는 상황 추론 방법이다. 온톨로지 상황 모델 뿐 아니라 다양한 상황 모델에 모두 적용할 수 있으며, 설정된 규칙에 의하여 추론한다. 단점으로는 Rule의 생성시 비용이 많이 들며 잘못된 생성된 Rule의 경우 시스템 스스로 수정할 수 없다는 단점이 있다. 대표적으로 SOCAM[6], CoBrA[7] 등이 존재한다. CoBrA의 경우 ContextBroker라 불리는 Agent를 통해 추론되는데, 정해진 규칙에 따라 추론하여 서비스를 제공한다. SOCAM의 기능도 유사하지만, 모바일을 통해 서비스를 제공하는 특징이 있다.
- Naive Bayes : 상황 인지 시스템에서 사용자에게 제공한 서비스의 로그를 분석하여 상황을 추론 하는 방법이다. 사용자는 제공된 서비스에 대하여 만족하거나 불만족할 수 있으며 상황 인지 시스템은 이를 로그로 기록하여 분석 추론을 수행할 수 있게 지원한다. GAIA가 대표적인 시스템이다[8].

위와 같은 기존의 상황 추론 방법의 경우 상황 인지 시스템 동작 시에 제공된 서비스에 대하여 사용자의 피드백을 반영할 수 있는 방법이 없는 문제점을 내재하고 있다.

## 2.3 상황 인지 시스템의 구조

기존 상황 인지 시스템의 공통 구조는 [그림 1]



[그림 1] 기존 시스템의 공통 구조

과 같으며 아래의 네 단계를 통해 사용자에게 서비스가 제공된다.

- 첫째, 센서 계층에서는 물리, 혹은 가상의 센서를 통해 데이터를 수집하여 상황 정보 제공자(Context provider)에게 전달한다. 물리 센서 데이터의 경우 스마트 공간 내의 환경 정보를 수집하는 온도, 습도 등의 정보를 나타내며, 가상 센서 데이터의 경우 인터넷 망을 통하여 얻는 시간 정보, 날씨 정보, 날짜 정보 등을 의미한다.
- 둘째, 상황 정보 제공자는 전달 받은 데이터를 각각의 시스템의 내부에 있는 모듈을 통해 상황 정보로 가공하며 상황 모델(Context Model)에 저장한다.
- 셋째, 상황 해석자(Context Interpreter)는 상황 모델에 저장 되어 있는 상황 정보를 기반으로 현재의 상황을 추론한다.
- 넷째, 상황 인지 서비스(context-aware Service) 계층에서는 추론된 결과를 사용자에게 전달한다.

이와 같은 기존의 상황 인지 시스템 구조에서는 상황 해석자 내부의 Context KB를 통해 추론되며 추론된 결과가 사용자에게 제공된다. 이는 상황에 따라 정의된 서비스가 규칙으로써 존재한다는 의미로 만약 잘못 정의되어 있는 규칙 혹은 개개의 사용자에게 맞지 않은 규칙이 존재 할 경우 시스템 스스로 오류를 수정할 수 없는 문제점이

발생하게 된다. 또한 각각의 시스템에 존재하는 상황 정보 제공자의 경우 상황 정보를 생성하는데 있어 정형화된 방법 및 신뢰성 있는 데이터를 생성하기 위한 방법이 존재하지 않았다. 이에 따라 본 연구에서는 상황 추론 에이전트(Situation Reasoning Agent : SRA)를 통한 시스템의 학습 알고리즘 및 상황 관리 에이전트(Context Management Agent : CMA)를 기반으로 하는 신뢰 있는 상황 정보 방법을 설계 및 구현하고자 한다.

### 3. 시스템 구조

본 연구의 시스템은 주변 자원에 대한 정보를 수집할 수 있는 스마트 공간에서의 동작으로 가정한다. 스마트 공간에서의 상황 정보 수집 방법 및 학습 방법을 전체 시스템의 구조를 중심으로 기술한다.

#### 3.1 시스템 설계의 주안점

본 논문에서 제안하는 온톨로지 상황 모델 기반의 상황 인지 시스템의 설계 주안점은 다음 세 가지로 요약할 수 있다.

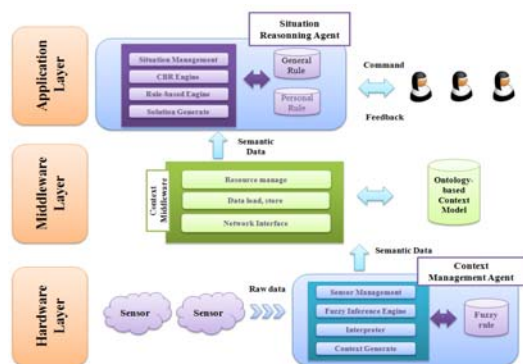
- 센서로부터 수집되는 다양한 상황 데이터의 신뢰 있는 상황 정보로의 변환 방법
- 시스템이 제공한 서비스에 대한 사용자의 피드백을 반영하여 학습할 수 있는 방법
- 시스템의 확장성, 재사용성, 상호 운용성을 위한 온톨로지 상황 모델의 구축

위와 같이 제안하는 시스템의 설계 주안점은 이전 연구에서 언급했던바와 같이 기존의 시스템이 가지는 신뢰 있는 상황 정보의 생성[9] 및 시스템 학습에 대한 문제[8]를 해결하고자 고안하였으며 이는 기존의 시스템과 기능적인 측면 및 구조적인 측면에서 <표 1>과 같은 차별성을 가지고 있다.

본 논문에서 제안하는 상황 인지 시스템의 전체적인 구조는 [그림 2]와 같다. 시스템의 구조는 분산 환경에 적용 가능하도록 하드웨어 레이어, 미

〈표 1〉 기존 시스템과의 차별성

	기존 시스템	제안하는 시스템
구조적 측면	상황 데이터의 종류에 따른 프로세싱 불가	상황 데이터의 종류에 적합한 프로세싱 가능
기능적 측면	학습 모듈 부재 및 상황 데이터 변환 모듈 부재	시스템 자동 학습 및 상황 데이터 변환 가능



[그림 2] 시스템 구조도

들웨어 레이어, 어플리케이션 레이어로 구분하였다. 하드웨어 레이어는 수집된 데이터의 가공 및 상황정보의 생성의 역할을 하며, 미들웨어 레이어는 상황정보를 저장하기 위한 온톨로지 기반의 상황모델과 상황정보의 입출력을 제어하는 모듈로 구성된다. 어플리케이션 레이어는 추론된 결과를 사용자에게 전달하는 역할을 한다.

시스템의 전체적인 흐름은 먼저 센서로부터 주변 자원에 대한 데이터를 수집하며 수집한 데이터를 신뢰 있는 상황 정보로 변환하여 온톨로지 기반의 상황 모델에 저장한다. 다시 상황 모델에 저장된 상황 정보를 기반으로 추론을 진행하며 추론의 결과를 사용자에게 제공한다. 이 때 제공된 서비스에 따라 학습이 진행되며 학습의 결과는 다음 상황의 발생시 적용된다.

### 3.2 시스템 구조

본 절에서는 시스템을 구성하는 세 레이어 중

상황정보 저장을 제어하는 미들웨어 레이어를 제외한 하드웨어 레이어와 어플리케이션 레이어의 세부 모듈에 대하여 기술한다.

#### 3.2.1 하드웨어 레이어

하드웨어 레이어의 주 역할은 수집된 데이터를 변환 하여 추론에 적합한 상황 정보를 생성 및 미들웨어 레이어의 상황 모델에 저장하는 것을 목표로 한다.

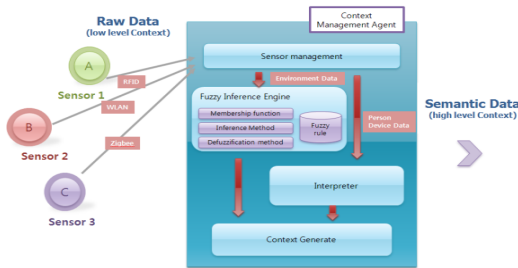
스마트 공간상의 센서는 특성에 따라 다양한 형식의 데이터 및 데이터 습득 시기를 가져야 한다. 습득 시기는 데이터의 특성에 기인하는데 RFID와 같이 사용자의 입·출입을 항상 감지해야 하는 센서(Person) 및 시스템 및 사용자의 조작에 의해 상태가 변하는 센서(Device), 일정 시간마다 환경 정보를 업데이트 하는 센서(Environment)로 구분할 수 있다. 이와 같이 상황 데이터의 습득 시기를 설정 하는 이유는 각 특징에 맞는 상황 데이터의 업데이트 주기를 설정하여 불필요한 생성을 줄여 시스템의 과부하를 줄이기 위해서이다. 또한 각각의 데이터 습득 시기에 따라 서로 다른 변환 과정을 거치게 되는데 그 내용은 아래와 같다.

- Person 인지 센서의 경우 데이터를 항상 습득하며 해석자 모듈을 통해 데이터를 변환한다.
- Device 센서의 경우 이벤트에 따라 데이터를 습득하며, 해석자 모듈을 통해 데이터를 가공한다.
- Environment 센서의 경우 일정 시간마다 데이터를 습득하며, 퍼지 추론 엔진 모듈을 통해 데이터를 변환한다.

상황 정보 변환 과정은 상황관리 에이전트(CMA)의 센서 데이터 수집 모듈(Sensor Management), 해석자 모듈(Interpreter), 퍼지 추론 엔진 모듈(Fuzzy Inference Engine), 상황 정보 생성자 모듈(Context Generator)을 통해 이루어지며 그 동작 과정은 아래 [그림 3]과 같고, 각 세부 모듈의 기능은 다음과 같다.

3.2.1.1 센서 데이터 수집 모듈

센서 데이터 수집 모듈은 다양한 특성의 데이터를 구분하기 위한 분류자로 정의된다. Person 및 Device 속성 데이터의 경우 해석자 모듈을 통해 상황 모델 사례가 생성되어야 하며 Environment 속성 데이터의 경우 퍼지 추론 엔진 모듈을 통해 상황 모델 사례가 생성되어야 한다. 센서 데이터 수집 모듈에서는 이와 같이 센서 데이터의 특성에 따라 추론엔진을 선택하게 된다.



[그림 3] 상황 관리 에이전트

3.2.1.2 해석자 모듈

Person 및 Device의 속성 데이터를 시맨틱 데이터로 가공하는 모듈이다. 상황 관리 에이전트 내부의 데이터베이스를 통해 사용자의 ID 및 Device의 상태 정보를 해석하며, 생성된 시맨틱 정보를 상황 정보 생성자 모듈에 전송한다. 예를 들어 사람의 ID 정보를 습득하는 RFID 리더기의 경우 고정된 위치 및 ID 정보가 데이터베이스에 있어 입출입시 사람의 위치 정보를 생성하여 전송한다.

3.2.1.3 퍼지 추론 엔진 모듈

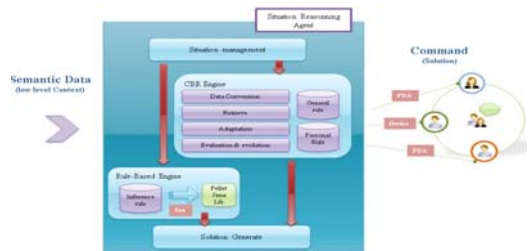
퍼지 추론 엔진 모듈에서는 Environment 속성 데이터를 통하여 두 종류의 시맨틱 데이터를 생성한다. 온도와 습도 같은 실측된 데이터의 경우 퍼지 소속함수를 통해 소속도가 가장 큰 소속함수를 시맨틱 데이터로 생성하며, 불쾌지수와 같은 실측된 데이터로부터 추론되는 종속된 데이터의 경우 퍼지 추론을 통해 시맨틱 데이터를 생성한다. 퍼지 추론 엔진 모듈에서 생성하는 시맨틱 데이터는

해당 속성의 상태 뿐 아니라 상태의 신뢰도를 나타내는 정규화된 값(1부터 0사이의 실수값)으로 표현된다. 실측 정보의 변환 과정 및 종속 정보의 변환 과정은 다음과 같다.

- 실측 상황 정보의 생성 : 주변 센서로부터 습득된 데이터는 1차적으로 퍼지소속 함수를 거치게 된다. 퍼지 소속함수는 각 상태를 연산하는 수식을 나타내며 습도의 상태를 7단계로 표현하고자 한다면 퍼지 소속함수의 개수는 7개가 된다. 각 소속함수의 수식에 의해 0과 1사이의 상태로 계산되는 퍼지수의 경우 해당 상태의 소속도 즉, 현재 상태의 신뢰도를 반영하게 된다.
- 종속적 상황 정보의 생성 : 퍼지 추론 엔진의 목적은 센서로부터 습득된 습도와 온도 등의 실측 값을 기반으로 불쾌지수 등의 종속적 속성의 시맨틱 데이터를 생성하는데 있다. 본 논문에서는 종속적 상황 정보를 생성하기 위해서 퍼지 추론방법인 Mamdani의 min-max 중심법을 사용한다[12]. 추론을 위한 퍼지규칙은 퍼지 추론 엔진 내부의 지식베이스(Knowledge Base)에 저장되어 있으며, 추론을 통해 얻어지는 값은 종속적 속성의 비 퍼지수(불쾌지수의 예측값)로, 실수값을 가지게 된다. 이 실수 값은 다시 퍼지소속함수를 통해 변환된다.

3.2.1.4 상황 정보 생성자

해석자 모듈과 퍼지 추론 엔진 모듈을 통해 생성된 시맨틱 정보는 상황 정보 생성자 모듈을 통해 미들웨어의 온톨로지 상황 정보 모델에 저장된다.



[그림 4] 상황 추론 에이전트

### 3.2.2 어플리케이션 레이어

어플리케이션 레이어에서는 온톨로지 기반의 상황 모델로부터 상황 정보를 추출하고 추출한 상황 정보를 추론하며 추론된 결과를 사용자에게 서비스로서 제공하는 과정을 수행한다. 이 같은 과정의 동작 구조는 [그림 4]와 같다.

상황추론 에이전트는 상황 관리(Situation Management), 사례 기반 추론 엔진(CBR Engine), 규칙 기반 추론 엔진(Rule-based Engine), 서비스 생성자(Solution Generator)의 네 모듈로 구성되어 있으며 위 모듈을 통해 추론 및 학습이 이루어진다. 아래에서는 각 세부 모듈별 기능을 기술한다.

#### 3.2.2.1 상황 관리

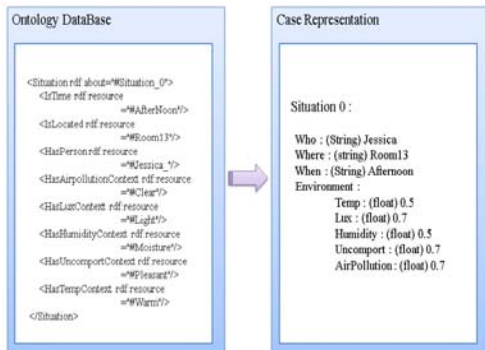
상황 정보의 특성 및 도메인에 따라 학습이 요구되는 상황과 그렇지 않은 상황을 구분한다. 현재는 학습이 요구되는 상황의 경우 사례 기반 추론 방법을 선택하고 그렇지 않을 경우 규칙 기반 추론 방법을 선택한다. 또한 도메인의 특성에 따라 또 다른 추론 알고리즘을 추가로 구성이 가능하다.

#### 3.2.2.2 사례 기반 추론과 규칙 기반 추론

규칙 기반 추론은 대부분의 기존 시스템에서 사용되었으며, 본 연구에서는 Pellet과 Jena2와 같은 오픈 라이브러리를 활용함으로써 개발하였다. 사례 기반 추론은 사용자의 피드백에 따른 학습이 요구되는 도메인 및 상황 정보를 추론하는 역할을

한다. CBR 추론 과정은 순서에 따라 다음과 같은 단계로 수행된다.

- 1) Representation 단계 : 상황 정보를 사례 기반 추론에서 연산 가능한 사례로 표현한다. 이에 따라 추론의 바탕이 되는 상황 정보에 대한 사례의 표현은 [그림 5]와 같이 구성한다. 사례의 표현은 사용자를 둘러싼 정보를 수집할 수 있는 센서의 종류 및 도메인에 따라 더 세분하여 구성 될 수 있으며 본 논문에서는 다섯 가지 환경 센서 및 위치, 시간, 사람 정보로 구성하였다. 표현된 사례 중 누가, 어디에서, 언제를 표현하는 속성 정보는 String 형태로 나타내며 이를 제외한 환경 정보는 Float 형태로 변환하여 나타낸다.
- 2) Retrieve 단계 : 사례베이스로부터 현재 사례와 가장 유사한 사례 (최근접 사례)를 검색하는 단계이다. 최 근접 사례의 검색은 k-nearest 알고리즘을 사용하며 이때의 유사도는 아래의 수식을 통해 계산된다. 수식에서처럼, 유사도는 사례의 각 속성간의 차이 및 속성 가중치의 곱을 속성 가중치의 합으로 나눈 수치가 된다. 위의 수식 (1)에서  $W$ 는 속성 가중치를 의미하며 함수  $f$ 는 속성간의 차이를 연산하는 함수로 환경을 구축하고 있는 정보의 경우 속성 차의 절대값으로 연산되며, 그 외의 who, when, where의 차이는 유사도 테이블을 통해 연산된다. 함수  $f$ 의 두 파라미터는 최근접 사례 및 현재 사례의 각 속성의 상태를 나타낸다.



[그림 5] 사례의 생성

$$similarity = \frac{\sum W_i \cdot f_d(C_{h_i}, C_c)}{\sum W_i} \quad (1)$$

또한 유사도 비교시 속성별 가중치를 설정하는데, 이는 상황의 종류에 따라 속성의 중요도를 설정하기 위함이다. 본 논문에서 제안하는 속성 가중치 설정 방법은 속성 중 민감한 속성을 찾아내는 방법으로 각 속성의 차



이 값을 모든 속성의 차이 값의 합으로 나누어 각 속성이 가지는 비중만큼 가중치를 가지도록 하는 것이며 아래의 수식 (2)와 같이 표현된다. 수식 (2)의 함수  $f$  및 파라미터의 역할은 위의 수식 (1)에서와 동일하다.

$$Weight_i = \frac{|f_d(c_{s_i}, C_{c_i})|}{\sum |f_d(c_{s_i}, C_{c_i})|} \quad (2)$$

3) Adaptation 단계 : Retrieve 단계에서 검색된 최근접 사례와 현재 사례와의 차이를 규칙을 통해 해결하는 단계이다[13]. Adaptation 단계는 <표 2>의 규칙을 통해 이루어지며 이 규칙은 처음 시스템 설계시 정의되거나 혹은 학습에 의해 자동 생성된다.

Match rule의 경우 현재 사례와 최근접 사례의 차이가 규칙에 정의된 대로 정확히 일치하는 경우 적용되며 Range Rule의 경우 두 사례의 차이가 일점 범위를 만족할 경우 적용되는 형태이다.

<표 2> 두 가지 타입의 규칙

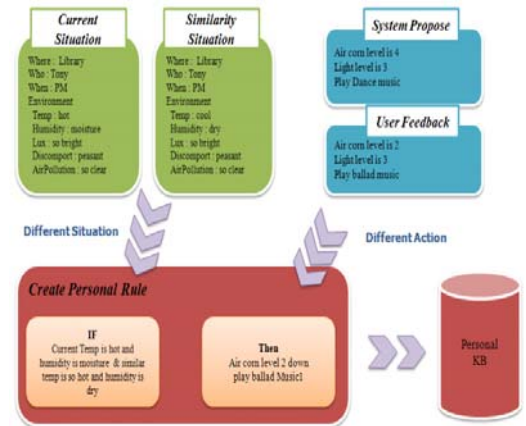
Num	Type	Rule	Priority
Rule 1	Match rule	IF Current Temperature is warm, and Similar Temperature is Cold and Current Humidity is dry Then set the Air conditioner level 2 down	0
Rule 2	Range rule	IF Current Lux is very dark and Similar Lux level less than normal Then set Lux level 4	0

4) Evaluation and Store 단계 : 제공된 서비스에 대한 사용자의 피드백을 반영하여 학습하는 단계이다. CBR에서는 기본적으로 새로운 사례를 저장함으로써 학습이 진행되지만, 본 연구에서는 범용적인 사례 학습을 위해 [그림 6] 과 같은 학습 알고리즘을 제안한다[14]. [그림 6]에서

처럼 시스템이 현재 사례에 따른 서비스를 제공하였을 때 사용자가 ‘air con level’ 및 ‘music type’을 직접 조작하였다고 가정하자. 이 때 시스템은 사용자의 행동을 인지하여 현재 사례와 최근접사례와의 차이인 ‘Temp’와 ‘Humidity’의 상태 차이를 [그림 6]과 같은 조건절로 구성하며 또한 사용자의 조작과 시스템이 자동 제공한 서비스와의 차이인 ‘air con level’ 및 ‘music type’를 결과절로 구성함으로써 적합화 규칙을 생성한다. 이 같이 생성된 규칙은 기존 규칙과의 충돌 위험성 때문에 개별적 Rule DB에 저장된다. 만일 개별적 Rule DB에서도 충돌 발생시에는 우선 순위등을 통해 충돌을 방지한다. 학습의 결과로 생성되는 사례 및 규칙은 다음 상황의 발생시 실시간으로 적용된다.

### 3.2.2.3 서비스 생성자

사례 기반 추론 모듈과 규칙 기반 추론 모듈에서 추론된 서비스 리스트를 다양한 Device를 통해 제공한다. 추론된 결과는 미들웨어의 상황 모델에 전송되어 저장된다.



[그림 6] 적응된 케이스 규칙의 학습

## 4. 실험

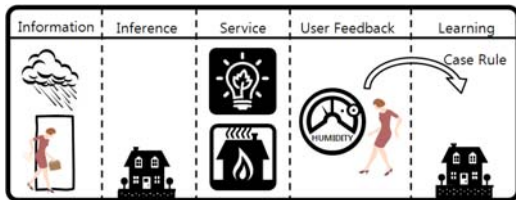
본 논문에서 제안하는 상황인지 시스템의 구현



을 위해서 아래와 같은 시나리오를 통해 시물레이션 하였다.

“제시카는 일을 마치고 집에 들어올 때, 밖에서는 비가 내리고 있었으며 그에 따라 집안의 온도와 습도는 매우 낮은 상태를 가지고 있었다. 제시카의 집에서 동작하는 상황인지 시스템은 제시카가 집에 들어옴을 감지하고 상황 정보를 기반으로 현재 상황을 추론하여 조명 및 난방 서비스를 제공한다. 하지만 제시카는 제공된 서비스에 습도측면에서 만족하지 못해 제습기를 가동시킨다. 며칠 후 다시 비슷한 상황에서 제시카가 집에 들어왔을 때 집 안의 시스템 학습된 결과로 조명과 난방 및 제습 서비스를 제시카의 의도에 맞게 제공하여 제시카를 만족시킨다.”

위 시나리오 상에서 제한하는 상황 인지 시스템의 학습 동작은 [그림 7]과 동일하며 실질적인 시스템의 수행 내역은 아래와 같다.



[그림 7] 학습 시나리오

- ① 제시카의 집안 출입을 센서가 감지함에 따라 상황관리 에이전트를 통한 위치 정보 생성 및 환경 정보를 생성 한다.
- ② 상황추론 에이전트를 통한 상황 정보 추출 및 CBR 추론 방법에 따라 제시카에게 조명 및 히터를 가동한다.
- ③ 제공한 서비스에 따른 제시카의 피드백을 시스템이 인지한다.
- ④ CBR 학습 모듈에 따라 피드백을 반영한 학습 알고리즘을 수행한다(사용자의 요구를 반영함).

위의 시나리오 상에서 시스템이 사용하는 상황 정보 중 제시카의 입출입 및 위치 정보의 경우 상

황관리 에이전트의 해석자 모듈을 통해 변환되며 제시카 집안의 온도, 습도, 조도, 공기 오염도 등의 환경정보의 경우 상황관리 에이전트의 퍼지 추론 엔진 모듈을 통해 변환된다. 이렇게 생성된 상황 정보는 온톨로지 기반의 상황 모델에 저장되며 <표 3>와 같이 구성된다.

상황관리 에이전트를 통해 저장된 상황 정보는 상황추론 에이전트의 사례 기반 추론 엔진을 통해 추론된다. 이는 현재의 시나리오가 사용자의 피드백이 요구되는 도메인이기 때문에 규칙 기반 추론보다 사례 기반 추론이 더 적합하기 때문이다. [그림 8]은 사례 기반 추론 과정을 검증하는 시물레이션 어플리케이션의 모습을 나타낸다.

<표 3> 상황 정보의 저장

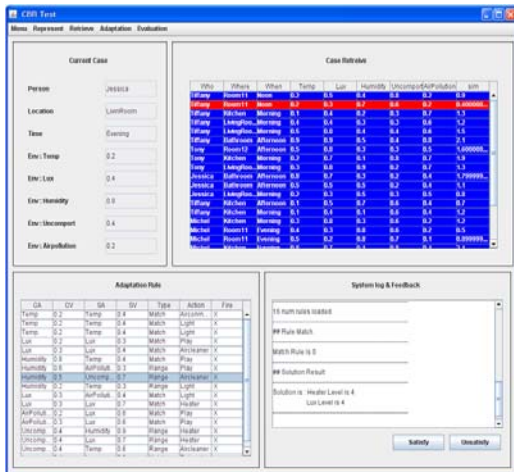
```

<Location rdf:about = "#CurrnetLocation">
  <rdf: type rdf: resource = "&owl; Thing"/>
  <Located rdf: datatype = "&xsd; string">LivingRoom
</Located>
</Location>
<owl: Thing rdf: about = "#CurrnetTemp">
  <rdf: type rdf: resource = "#Temp"/>
  <State rdf: datatype = "&xsd; string">Cold
  </State>
</owl: Thing>
<owl: Thing rdf: about = "#CurrnetUser">
  <rdf: type rdf: resource = "#Person"/>
  <Name rdf: datatype = "&xsd; string">Jessica
  </Name>
</owl: Thing>
...
    
```

시물레이션 프로그램은 네 개의 윈도우로 구성 되어있다. 먼저 왼쪽 위의 창은 상황 모델로부터 상황 정보를 추출 및 가공한 현재의 사례를 나타낸다. 오른쪽 위의 창은 이전에 발생했던 사례들의 목록을 나타내며 이는 사례를 구성하는 속성 정보를 표현한다. 또한 그림에서 적색으로 표시된 사례는 현재 사례와 가장 유사한 사례를 나타낸다.

왼쪽 아래의 창은 현재 시스템이 정의한 Adaptation Rule과 규칙의 적용 유무를 나타낸다. 오른쪽 아래 창은 사용자에게 제공되는 서비스의 리스트 및 사용자의 피드백을 직접적으로 수용할 수 있는 버튼이 존재한다.

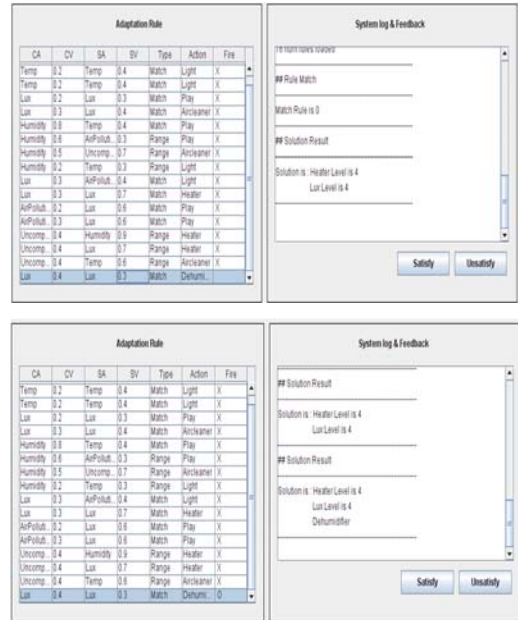
위의 예제에서는 우선, 현재 사례와 최 근접 사례와의 차이를 해결할 수 있는 Adaptation Rule이 존재하지 않기 때문에 사용자가 원하는 서비스를 정확히 제공할 수 없음을 나타낸다. 이러한 문제점을 해결하기 위하여 시뮬레이션에서는 사용자의 피드백을 통해 현재의 사례 및 피드백이 반영된 서비스 리스트를 사례 베이스에 기록하고, 개인화된 규칙을 개별 Rule DB에 추가하게 된다.



[그림 8] 시스템 시뮬레이션

[그림 9]의 상위 그림에서는 사용자의 피드백에 따른 규칙 생성 모습을 나타내며 [그림 9]의 아래의 그림에서는 임의의 시간 후에 같은 상황의 발생 시 생성된 규칙이 적용되는 모습을 확인 할 수 있다. 이를 서비스의 결과로써 비교하면 처음 상황의 발생시에 제공되었던 서비스 리스트(Heater level as 4 and , set the Lux level as 3)에서 사용자에게 적합한 서비스 리스트로 변화하는 것을 확인 할 수 있다. (Heater level as 4 and , set the Lux level as 3 and Dehumidifier) 위 실험에서는 생성된 규

칙을 검증하기 위해서 사례의 저장은 배제하였다.



[그림 9] 규칙의 생성 및 적용 모습

## 5. 결 론

본 연구의 주안점은, 정형화된 상황 정보 생성의 및 사용자 피드백을 반영한 시스템의 학습을 통해 기존의 시스템보다 효과적인 상황인지 시스템을 제안하고자 하였다. 본 연구의 결과를 요약 하면, 첫째, 데이터의 종류에 따라 단순추론과 퍼지추론을 병행함으로써 보다 신뢰성 있는 상황 정보를 생성하였다. 둘째, 규칙 기반 추론 엔진과 사례 기반 추론 엔진을 병행함으로써 상황 정보에 따른 추론을 수행하였다. 셋째, 사용자의 요구사항을 피드백으로 받아서 학습함으로써 보다 효과적인 추론기법을 제시하였다. 넷째, 시스템의 주요기능을 에이전트화 함으로서 보다 지능적인 상황인지 시스템의 구조를 제시하였다.

향후 연구로 개선되어야 할 점은 다음과 같다. 첫째, 상황 인지 시스템의 학습에 따른 규칙의 생성시 기존의 규칙과 충돌이 발생하는데 이를 해결하기 위한 더 효율적인 알고리즘이 필요하다. 둘

제, 시뮬레이션환경이 아닌 실제 센서 및 사용자를 대상으로 실험을 진행하여 알고리즘의 타당성을 증명해야 한다. 셋째, 시스템의 추론 시기 및 센서의 데이터 수집 시기를 결정하고 이를 조절 할 수 있는 효율적인 미들웨어를 구축해야 한다. 추후 연구에서는 이와 같은 관점을 중심으로 보다 효율적인 상황 인지 시스템이 개발되어야 할 것이다.

## 참 고 문 헌

- [1] Karen Henricksen, Jadwiga Indulska and Andry Rakotonirainy, "Infrastructure for Pervasive Computing : Challenges", *Workshop on Pervasive Computing INFORMATIK 2001*, Vienna, Vol.157(2001), pp.214-222.
- [2] Anind K. Dey and Gregory D.Abowd, "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), pp.97-166.
- [3] Philippe Debaty, Patrick Goddi and Alex Vorbau, "Integrating the physical world with the web to enable context-enhanced mobile services", *Mobile Networks and Applications*, Vol.10(2005), pp.385-394.
- [4] Karen Henricksen, Jadwiga Indulska and Andry Rakotonirainy, "Modeling Context Information in Pervasive Computing Systems", *Proceedings of the First International Conference on Pervasive Computing*, (2002), pp.167-180.
- [5] Xiao Hang wang, Tao Gu, Da Qing Zhang and Hung Keng Pung, "An Ontology-based Context Model in Intelligent Environments", *In Proceedings of Communication Networks and Distributed System Modeling and Simulation Conference*, San Diego, California, USA, (2004), pp.270-275.
- [6] Xiao Hang Wang, Tao Gu, Da Qing Zhang, and Hung Keng Pung, "A service-oriented middleware for building context-aware services", *Journal of network and Computer Applications*, Vol.28(2005), pp.1-18.
- [7] Harry Chen, Tim Finin and Anupam Joshi, "An Ontology for Context-Aware Pervasive Computing Environments", *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, Vol.18(2003), pp.197-207.
- [8] Anand Ranganathan and Roy H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments", *International Federation for Information Processing*, (2003), pp.143-161.
- [9] Kwang-Eun Ko and Kwee-Bo Sim, "Context Aware System based on Bayesian Network driven Context Reasoning and Ontology Context Modeling", *Interanational Journal of Fuzzy Logic and Intelligent Systems*, Vol.8, No.4(2008), pp.254-259.
- [10] Anind K. Dey and Gregory D.Abowd, "Towards a Better Understanding of Context and Context-Awareness", *Workshop on the what, who, where, when and how of context-awareness at CHI*, April, (2000), pp.304-307.
- [11] Steven A. N. Shafer, Barry Brumitt and JJ Cadiz, "Interaction Issues in Context-Aware Intelligent Environments", *Human-Computer-Interaction*, Vol.16(2001), pp.363-378.
- [12] E.H. MamDani and S.Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal of Human-Computer Studies*, (1999), pp.135-147.
- [13] Wolfgang Wike and Ralph Bergmann "Techniques and Knowledge Used for Adaptation

- During Case-Based Problem Solving”, *Tasks and Methods in Applied Artificial Intelligence*, Vol.1416(1998), pp.497-503.
- [14] Kathleen Hanney and Mark T. Keane, “The Adaptation Knowledge Bottleneck How to Ease it by Learning from Cases”, International Conference on Case-Based Reasoning, Vol.1266 (1997), pp.359-370.

## ◆ 저 자 소 개 ◆

**장 인 우 (zx1018@naver.com)**

국민대학교에서 컴퓨터공학을 전공하고, 현재 국민대학교 컴퓨터공학부 석사 과정에 재학 중이다. 주요 관심분야는 컨텍스트 컴퓨팅, 온톨로지, 시맨틱 웹, 데이터마이닝 등이다.

**우 종 우 (cwwoo@kookmin.ac.kr)**

미국 Illinois Institute of Technology 에서 전산학 박사를 받고, 현재 국민대학교 컴퓨터공학부 교수로 재직중이다. 주요 연구관심분야로 컨텍스트 컴퓨팅, 증강형실, 지능형 에이전트, 모델링과 시뮬레이션 등이며, Artificial Intelligence in Medicine, 한국정보과학회, 한국 정보처리학회등에 다수의 논문을 실었다.