

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/314859928>

# A Development of Intelligent Context-Awareness Middleware

Article · November 2012

DOI: 10.9716/KITS.2012.11.sup.165

CITATION

1

READS

28

2 authors:



Joohee Suh

University of Oklahoma

7 PUBLICATIONS 12 CITATIONS

SEE PROFILE



Chong-Woo Woo

Kookmin University

28 PUBLICATIONS 239 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Context Aware Learning Model (CALM) [View project](#)



CIRCSIM-Tutor [View project](#)

## 지능적 상황인지 미들웨어의 개발\*

서주희\*\* · 우종우\*\*\*

### A Development of Intelligent Context-Awareness Middleware\*

Joohee Suh\*\* · Chong-woo Woo\*\*\*

#### ■ Abstract ■

Context-Awareness system provides an appropriate service to user by recognizing situation from surrounding environment. There are many successful studies on this framework, but still has some limitations. In this paper, we are describing a context-awareness middleware that can enhance the limitation of the previous approaches. We first defined a new concept of context-awareness environment as a social intelligence. This concept implies that intelligent objects can make relationships, can aware of situation from surrounding environment, and can collaborate to accomplish a given task. The significance of the study is as follows. First, the system is capable of multi context-awareness since it is designed with a structure that supports multiple lines of reasoning. Second, the system is capable of context planning by adapting AI planning mechanism. Third, the system is capable of making the intelligent objects as a group for collaboration, and provides adaptive service to user. We have developed a prototype of the system and tested with a virtual scenario.

Keyword : Context-Awareness, Middleware, Context Modeling, Context Planning, Blackboard, Social Intelligence

## 1. 서 론

1994년 Bill N. Schilit은 환경과 객체 또는 사람을 둘러싼 조건에 대한 모든 정보를 상황(context)이라고 정의 하였으며, 사람을 둘러싼 조건이라는 것은 사람과 유비쿼터스 환경이 서로 상호작용 하는 것과 관련된 모든 정보라 하였다[15]. Anind K. Dey는 Bill N. Schilit에 의해 정의된 상황이 경우에 따라서는 전혀 관계없는 정보가 될 수 있는 물리적인 환경에 대한 모든 정보까지 포함 하게 되므로, 어플리케이션 개발자들이 보다 쉽게 실제 시스템에서 상황을 기술하기 어렵다고 주장하며 상황을 다음과 같이 재정의 하였다[2]. 즉, 상황이란 환경에 내재된 엔티티(Entity)의 특성을 기술하기 위해 사용 될 수 있는 모든 정보를 의미하며, 엔티티는 사람과 어플리케이션이 상호작용 하거나 어플리케이션들끼리 상호작용하는 것에 관련된 객체, 장소 또는 사람 자체를 지칭 한다[14]. 상황 인지 컴퓨팅(context-awareness computing)이란 이러한 상황을 이용하여 사용자에게 적절한 정보나 서비스를 제공하는 것이며, 이러한 컴퓨팅을 할 수 있는 환경을 상황 인지 컴퓨팅 환경이라 한다.

기존에 연구된 상황인지 컴퓨팅 환경과 환경을 구성하고 있는 구성 요소들을 분석하고 몇 가지 제한점들을 살펴보면 다음과 같다. 첫째, 지금까지 연구된 상황인지 시스템은 모두 중앙집중식의 처리 방식을 사용한다. 즉, 중앙 집중 서버에서는 환경 곳곳에 산재된 다양한 컴퓨팅 기기들로부터 데이터를 수집한 후 상황모델링과 상황추론을 통하여 한곳에서만 상황인지를 하게 된다. 둘째, 환경에 내재된 다수의 컴퓨팅 기기들은 매우 수동적인 성향을 가지고 있다. 즉, 기기들이 스스로 상황을 판단하고 이에 대한 지능적 서비스를 수행하는 것이 아닌, 단순히 중앙 집중 서버가 필요한 자료를 수집하는 역할만 수행한다. 셋째, 환경을 구성하고 있는 개체들 간의 협업에 대한 방안은 포함하지 않는다. 즉, 대부분의 기기들은 정보를 단순히 수집하며 수집된 정보에 대한 분석기능은 부재하다.

본 논문에서는 기존의 상황인지 컴퓨팅 환경에서 상황을 인지할 뿐만 아니라, 상황을 인지한 후 그에 맞는 계획을 수립하여, 지능적인 개체들에 의해 서비스가 수행될 수 있는 사회적 지능성 기반의 상황인지 컴퓨팅 환경(social intelligence based context-awareness environment)을 제시하고자 한다. 사회적 지능성 이란, 사물이나 현상을 보고 현상에 대한 적합한 상황을 판단할 수 있으며, 다른 객체들과의 관계를 맺고, 서로가 가진 정보를 공유하거나 협업과 같은 활동을 수행할 수 있는 능력을 의미한다. 이러한 시스템은 다음과 같은 특성들을 만족시킨다. 첫째, 환경은 지능형 개체들의 존재를 인지하고, 지식을 서로 공유할 수 있다. 둘째, 공유된 지식과 존재의 인지를 통해서 서로 긴밀한 협업이 이루어 질 수 있다. 셋째, 환경정보를 통하여 상황을 판단하고, 인지한 상황에 대한 적절한 수행계획을 수립할 수 있다. 넷째, 지능적 행위 및 협업을 수행 할 수 있는 환경 독립적 지능적 개체를 포함한다.

이러한 지능적 개체는 다음과 같은 특징을 가진다. 첫째, 스스로 상황인지를 할 수 있다. 둘째, 인지한 상황을 기반으로 지능적 사고를 통하여 지능적인 서비스를 제공한다. 셋째, 자신만의 고유 지식과 환경에 대한 지식을 분리하여 갖는다. 그러한 분리된 지식체계가 있음으로써 환경에 대한 지식과 자신이 가지고 있는 지식을 함께 이용하여 서로 다른 환경에 적응하며, 환경 독립적인 개체가 될 수 있다. 넷째, 지능적 행위를 함에 있어서 환경 및 다른 개체들과 협업을 할 수 있다. 본 연구에서는 이러한 지능적 개체들을, 지능형 상황로봇(intelligent context-awareness robot)으로 정의하였다. 따라서 기존의 환경을 변화 시킬 수 있는 개체는 사람뿐만이 아닌 지능적 개체인 상황로봇까지 확장하여, 모든 사건 및 상황에 사람이 개입하지 않아도 상황이 종결되거나 시스템이 다른 상황으로 변화될 수도 있게 된다. 또한, 이렇게 상황로봇들이 포함된 환경에서는 상황을 중앙 집중 서버 한 곳에서만 인지하는 것이 아닌, 지능적 개체들

에 의하여 여러 곳에서 동시에 인지하는 다중 상황인지가 가능하게 된다. 본 논문에서는 이러한 환경을 지원하기 위하여 사회적 지능성 기반의 상황인지 미들웨어(Social Intelligence based Context-Awareness Middleware : SI-CAM)를 제시하며, SI-CAM의 기반 구조로서 다중 추론이 가능한 블랙보드(blackboard) 모델을 사용하여 상황인지 미들웨어를 구축한다.

본 논문의 구성은 다음과 같다. 제 2장 관련 연구에서는 설계에 필요한 요소기술들에 관하여 기술한다. 제 3장에서는 전체 시스템 구조와 각 시스템 세부 기능들에 대하여 기술하고, 제 4장에서는 실험 시나리오 및 이에 관한 시뮬레이션 과정을 소개한다. 제 5장에서는 결론 및 향후 연구에 대해 기술한다.

## 2. 관련 연구

### 2.1 상황인지 컴퓨팅 환경과 온톨로지

현재 대표적 상황인지 시스템들인 Context Toolkit[3], HP's Cooltown[9], CoBrA[1], SOCAM[5, 18], GAIA[12]들을 분석하면, 상황인지 서비스 수행을 위한 과정들이 다음과 같이 요약된다. 첫째, 환경에 산재된 다양한 컴퓨팅 기기들로부터 일차적인 정보를 수집한다. 수집된 정보는 단순한 수치 데이터 값을 의미한다. 둘째, 시스템은 수집한 원시 데이터들을 가공하여 시스템이 읽고 이해할 수 있는, 보다 고차원적인 정보를 생성한다. 셋째, 생성된 정보와 시스템이 가지고 있는 환경 정보 모델을 통하여 현재의 상황을 추론 한다. 넷째, 추론한 상황에 맞게 사용자에게 적절한 서비스를 제공한다.

이러한 상황인지 절차에 대한 명칭과 모델링 기법은 각 응용 시스템마다 다르지만, 시스템을 구성하는 상황인지 미들웨어의 구조는 일반적으로 다음과 같다. 즉, 데이터를 얻는 물리적 계층(Physical Layer), 상황을 인지하는 상황추론 계층(Context Reasoning Layer), 그리고 인지한 상황에 기반하여 서비스를 수행하는 응용 계층(Application Layer)으로 구성되며, 문제영역에 대한 모델링 기법과 구축된 모델에 적합한 추론기법으로 각 상황인지 시스템이 구성된다.

text Reasoning Layer), 그리고 인지한 상황에 기반하여 서비스를 수행하는 응용 계층(Application Layer)으로 구성되며, 문제영역에 대한 모델링 기법과 구축된 모델에 적합한 추론기법으로 각 상황인지 시스템이 구성된다.

### 2.2 상황추론과 상황모델링

실세계는 매우 다양하고 복잡한 상황들의 집합체이며, 쉽게 예측할 수 없는 상황들이 매우 빈번하게 발생되기 때문에 상황 인지 컴퓨팅 시스템의 개발은 단순하지 않다. 따라서 대부분의 기존 상황인지 컴퓨팅 시스템에서는, 이러한 현실적인 문제들에 접근하기 위한 필수적인 기반구조로 상황모델링(context modeling)과 상황추론(context reasoning)이 주축을 이루어왔다[4-6, 18]. 특히, 상황 정보가 실질적으로 표현, 관리, 조작, 그리고 인지되기 위한 상황모델링에 관한 연구는 Attribute-Value 모델, Web Based 모델, ER과 UML 모델 및 온톨로지(Ontology) 모델 등으로 다양하게 진행되고 있으나, 그 중 지식의 공유, 확장, 재사용 그리고 논리적인 추론이 가능한 장점을 가지고 있는 온톨로지 기반 모델링이 가장 핵심적인 기법으로 연구되고 있다[17].

이와 같이 상황모델과 상황추론을 통하여 적절한 서비스를 제공 하는 응용연구들은 활발히 진행 중이며 몇 가지 사례를 분류해보면, 거주자의 상황에 맞게 편의를 제공하는 홈 자동화 시스템 및 스마트 홈 시스템[8, 19], 사용자의 건강 정보 및 신체 상태를 분석하여 상황에 맞는 서비스를 제공하는 U-Healthcare 시스템 등이 있다[10]. 또한 스스로 상황을 인지하고 이를 기반으로 지능적 행위를 할 수 있는 지능형 서비스 로봇에 대한 연구도 활발히 진행 중이다. 예를 들면, 지능형 로봇 환경에서 상황 정보를 베이지안 네트워크를 이용한 서비스 모델링 연구[4], 환경과 상호작용 하는 모바일 로봇이 객체를 인지하고 그 객체에 대한 로봇의 상황 이해력에 관한 연구 및 로봇의 지식체계 구축

관련 연구들이 있다[7].

### 2.3 지능형 서비스 로봇과의 협력

지능형 서비스 로봇에 관한 연구는 다양한 연구가 진행되고 있지만, 지능형 서비스 로봇이 환경을 이해하고 상황을 추론하기 위해서는 로봇의 지식체계가 필요하다. 기존의 연구 사례들에서는 로봇의 지식체계를 온톨로지 기반으로 구축하여 로봇이 상황을 추론하여 상황에 맞는 적합한 서비스 수행할 수 있는 방법을 제시하였다[3]. 그러나 이러한 연구들은 하나의 로봇에 대해 하나의 지식체계를 이용하여 상황을 인지하며, 다양한 지식체계를 가진 다수의 로봇들로 구성된 환경은 고려하지 않았다. 다만, 모델링과 지식체계 보다는 통신을 위주로 하여 로봇 팀을 구성하는 연구가 진행되고 있다. 예를 들어, Rodic과 Engelbrecht에 의하면, 다중 로봇 팀은 특정한 하나의 로봇 보다 작업을 훨씬 빠르고 효율적으로 수행할 수 있다는 가정 하에 다중 로봇 팀을 위한 협력 방법으로 social network를 제시하였다[13]. 또한 Labella 외는 's-bots'라 불리는 몇 개의 작고 독립적인 모듈들을 제시하여, 그들 사이의 통신을 이용한 협업 방법을 제시하였고[11], 다중 로봇들 팀들이 협력을 위한 작업할당을 위한 작업계획(task planning) 기법을 제시하고 있다[16].

### 2.4 계획기법(Planning)

계획기법이란 주어진 목적을 달성하기 위하여, 실행 가능한 작업순서들의 나열을 얻는 행위로서, 주어진 초기 상태가 최종적인 목적 상태로 가기 위한 일련의 계획(plan)을 생성한다. 계획기법은 전통적인 비 계층적 계획기법(non-hierarchical planning), 계층적 기법(hierarchical planning)에서부터, 최근의 기회적 계획기법(opportunistic planning), 동적 계획기법(dynamic planning)에 이르기 까지 다양하게 연구되고 있다. 계획기법의 핵심적인 기능

은 주어진 목적에 따라, 상황을 인지하여 자동적으로 목적 달성을 위한 계획을 생성하는데 있으며, 이러한 기능을 지능적 행위의 일부로 정의한다.

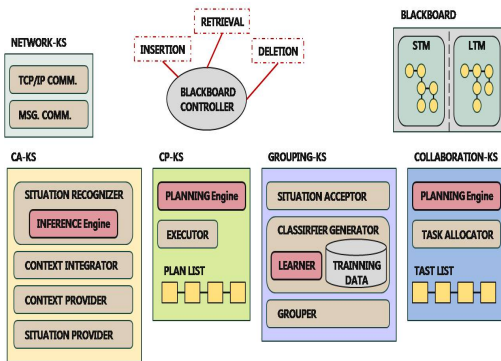
### 2.5 블랙보드 시스템(Blackboard System)

블랙보드 시스템은 Hearsay-2 음성이해 시스템에서 처음 소개 되었다. 블랙보드 모델은 다양한 지식이 요구되고 복잡한 문제의 해결을 위한 병렬적으로 처리하기 위한 모델이다. 블랙보드 시스템의 일반적인 구조는, 문제 해결에 필요한 지식들이 표현되고 저장되는 공유된 지식 저장소인 블랙보드, 블랙보드의 지식을 이용하여 문제를 해결하는 다수의 독립된 모듈들(knowledge sources), 블랙보드를 관리, 감독하는 제어기(controller)등으로 구성 된다. 각 모듈들은 독립적으로 주어진 작업을 수행하며, 작업들의 수행 결과는 공유된 저장소인 블랙보드에 저장된다. 모듈들 간의 직접적인 통신도 가능하지만, 블랙보드를 통하여 통신도 가능하게 된다. 모듈들의 수행 시기는 블랙보드를 통하여 결정되며 제어는 블랙보드에 나타나 있는 현재의 상태를 항상 감시하여, 이를 바탕으로 문제 해결을 위해 적용하여야 할 모듈을 결정한다. 블랙보드에 표현된 지식 체계는, 일반적인 지식베이스구조인 장기 기억(long-term memory) 및 단기 기억(short-term memory)으로 구성 되지만, 복잡한 시스템 구성을 위한 계층형(hierarchical layer)과 같이 다양하게 구성 될 수도 있다. 블랙보드 구조의 장점은, 시스템 구성의 유연성, 유연한 문제 해결 능력, 모듈의 선택적 사용, 다중 문제 해결 기법 적용, 그리고 계층적 지식의 표현화의 운영 가능성 등을 들 수 있다.

## 3. 시스템 구성

상황인지 시스템의 기본적인 구성은 주변 환경으로부터 입력된 정보를 해석하는 모듈, 해석된 정보를 모델링하여 상황으로 결정하는 모듈, 그리

고 정의된 상황을 응용 시스템에 따라 처리하는 모듈등으로 구성된다. 본 연구의 상황인지 미들웨어 구조는 이러한 기본적인 시스템의 흐름에 따라 상황인지 모듈을 설계하였으며, 다중 상황인지를 위하여 계획모듈, 그룹핑 모듈과 협업모듈을 추가하여 [그림 1]과 같이 다섯 개의 핵심적 모듈로 구성하였으며, 네트워크모듈은 시스템간의 통신에 해당하며 본 논문에서는 설명을 생략하였다. 시스템의 구성은 블랙보드, 블랙보드 제어기, 상황인지 모듈(Context-Awareness Knowledge Source), 상황계획 모듈(Context-based Planning Knowledge Source), 그룹핑 모듈(Grouping Knowledge Source), 협업 모듈(Collaboration Knowledge Source), 계획표시 모듈(Plan Indicator Knowledge Source)로 구성된다. 각 모듈들은 서로 다른 기능과 목적을 가지고 있으며 블랙보드를 통하여 임무를 수행한다. 각 모듈들의 역할 및 세부 사항은 아래와 같다.



[그림 1] 상황인지 미들웨어의 구조

### 3.1 상황인지 모듈

상황인지 모듈은 환경으로부터 수집되는 다양한 센서 데이터들을 인지하는 기능을 담당한다. 인지된 상황은 상황카드(Context Card)를 생성하여 블랙보드에 저장한다. 상황카드는 하나의 인지된 상황을 의미하며 보다 정형화된 상황을 의미하기 위하여 상황카드로 정의하였다. 기존 시스템과의 차이점은 인지된 상황을 직접 제공받을 수 있다는

점과 생성된 상황카드를 블랙보드에 저장하여 다른 모듈들이 이를 활용할 수 있게 한다는 점이다. 이렇게 생성된 상황은 블랙보드에 저장하여 상황기반의 계획 수립을 가능하게 한다. 상황인지 모듈의 세부 모듈은 아래와 같다.

- Context Provider : 외부로부터 끊임없이 원시 데이터를 입력 받는 부분으로써, 센서 등으로부터 받은 단순한 수치 데이터를 하위레벨 상황으로 가공하는 역할을 한다.
- Context Integrator : Context Provider로부터 생성된 하위레벨 상황들을 상황모델을 통하여 통합하고 추론하여 상위레벨 상황을 생성하는 역할을 한다.
- Situation Recognizer : Context Integrator에서 생성된 상위레벨 상황에 대한 상황카드를 생성하는 부분으로써, 상황카드에는 상황id, 장소 및 시간 등이 생성되어 카드에 저장된다.
- Situation Provider : 다른 지능적 개체로부터 인지된 상위레벨 상황을 전달받는 부분으로써, 전달받은 하나의 상황에 해당하는 상황카드를 생성하여 블랙보드에 저장한다.

본 연구에서는 대표적 온톨로지 언어인 OWL 기반으로 상황모델을 구축하였으며, 온톨로지 기반 규칙의 예는 <표 1>과 같다.

<표 1> 가상의 상황과 온톨로지 규칙의 예

Possible Context	Ontology Rules
Customer Arrived	IF Sensor hasLocation Entrance Sensor hasSensorValue $\geq$ 40 THEN Context "A Customer Arrived" occurred.
	IF Sensor hasLocation Table Sensor hasSensorValue $\geq$ 40 THEN Context "A Table Called" occurred.

### 3.2 상황계획 모듈

상황인지 모듈은 블랙보드에 저장된 상황카드로부터 인지된 상황에 맞는 상황계획을 수행하는 역할을 한다. 상황계획들은 기존의 상황카드에 기록하여 새로운 상황카드를 생성하고 이를 다시 블랙보드에 저장함으로써 다른 모듈들이 활용할 수 있게 한다. 이러한 구조는 SI-CAM이 인지한 상황에 맞는 지능적 계획을 수립할 수 있게 한다. 본 논문에서는 STRIPS 기반의 상황계획을 XML로 작성하여 블랙보드의 공통 지식베이스에 저장하고, 이는 상황계획모듈이 상황계획을 수행할 때 사용하게 된다. XML로 작성된 계획 규칙의 형태는 <표 2>와 같고, 각 상황에 대해 생성되는 계획은 <표 3>와 같다.

<표 2> 상황계획 규칙의 XML 표현

```
<rule>
<operator>
<action> UsheringInGuests </action>
<precondition> ComingInPeople </precondition>

<effect>
<add_list> AfterHavingSeats </add_list>
<delete_list> ComingInPeople </delete_list>
</effect>
<description> </description>
</operator>
<operator>
<action> BeingOrdered </action>
<precondition> AfterHavingSeats </precondition>

<effect> <add_list> AfterOrdering </add_list>
<delete_list> </delete_list>
</effect>
<description> </description>
</operator>
...
</rule>
```

### 3.3 그룹핑 모듈

그룹핑 모듈은 저장된 상황계획 카드로부터 해당 계획을 수행할 수 있는 지능형 개체들을 그룹핑 역할을 수행 한다. 세부 모듈인 Situation ac-

<표 3> 가상의 상황과 상황기반 계획의 예

Possible Context	Context-based Plan
CustomererArrived	UsheringInGuests BeingOrdered Serving HelpingPayBill
TableCalled	GoingToGuests CatchingDemanding PerformingTasks

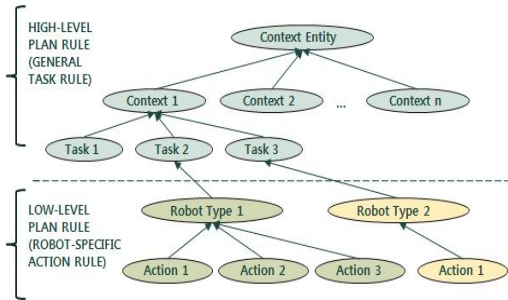
ceptor는 상황인지 모듈로부터 상황을 전달받아 Classifier generator에게 그룹 형성을 명령한다. Classifier generator는 활동 중인 지능형 개체들 중 해당 상황을 수행하기에 적합한 로봇을 분류하기 위하여 ID3 알고리즘 기반으로 접근하였고, 트레이닝 데이터는 XML 형태로 블랙보드에 저장하여 사용한다. 최종적으로, Grouper는 주어진 상황에 적합하게 Classifier에서 분류된 개체들에 대한 정보를 저장하는 역할을 하게 된다. 세부적인 연구내용은 선행된 사전연구에서 정의되나 있으며, 이를 기반으로 그룹핑 모듈을 구축하였다.

### 3.4 협업 모듈

협업모듈은 블랙보드에 저장된 협업그룹 카드로부터 그룹핑 된 로봇들과의 협업을 담당하는 역할을 한다. 동시에 인지된 상황에 대한 계획을 수립함에 있어서, 계속 진행해야 할 계획이 있으면 협업 그룹카드를 상황대기 카드(waiting context card)로 변경하고, 더 이상 진행해야 할 계획이 없어서 인지된 상황에 대한 서비스가 종료된 것이면 상황 종료 카드(finished context card)로 변경하여 블랙보드에 저장 한다([그림 3] 참조).

협업 모듈은 세부적으로 협업 계획기와, 임무 할당기(task allocator)로 구분된다. 협업 계획기는 상위레벨의 계획 규칙을 생성한 후 이를 수행할 수 있는 행동(action)들을 생성한다. 예를 들면, 만약 인지된 상황이 “cleaning”이라면, 이는 “sweeping, mopping” 등으로 하위 계획이 생성된다. 또한 세부계획 중의 하나인 “sweeping”은 이를 수행하기

위한 “move, lift, sweep, release” 등의 행동들을 생성한다. Task allocator는 생성된 세부계획을 상황 로봇에게 할당하는 역할을 수행한다. 협업 계획기의 구조도는 [그림 2]와 같고, 협업 규칙의 예는 <표 4>와 같다.



[그림 2] 협업계획기의 구조도

<표 4> 협업 규칙의 예

High-level Planning Rule	IF (Context = “Cleaning”) THEN (needTask (Sweeping, Mopping))
	IF (Context = “Serving”) THEN (needTask (Take_Order, Food_Serve, Check_out))
Low-level Planning Rule	IF (Task = “Sweeping”) THEN (needAction (Move, Lift, Sweep, Release))
	IF (Task = “Take_Order”) THEN (needAction (Give_Menu, Help_order))

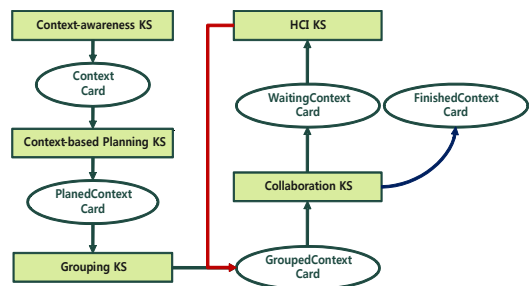
### 3.5 계획표시 모듈

계획표시 모듈은 블랙보드에 저장된 상황대기 카드를 통하여 현재 사용자로부터 혹은 서비스 제공자로부터 다음 계획을 진행해도 되는지에 대한 신호를 기다리고, 그 신호 여부에 따라 상황대기 카드의 변경 여부를 결정하는 역할을 한다. 예를 들어, CustomerArrived의 상황을 기반으로 생성된 계획들 중, Serving이라는 계획을 수행한 후, 다음 계획인 HelpingPayBill이라는 계획을 수행하기 위해서는 서비스 수혜자들이 식사를 다 마칠 때까지

기다려야 한다. 본 논문에서는 이를 위해 계획 표시 모듈을 구축함으로써 사용자에게 보다 지능적 서비스를 제공할 수 있게 한다.

### 3.6 블랙보드(Blackboard)

블랙보드 시스템의 블랙보드 자체는 전통적인 지식베이스의 형태인 장기 메모리와 단기 메모리로 구성된다. 장기 메모리에 저장되는 정보는 수시로 변하지 않는 정보들로서, 온톨로지, 계획 규칙, 환경정보, 훈련데이터, 등이 저장된다. 단기 메모리에는 시스템 실행 시에 수시로 변화되는 데이터들이 저장되며, 환경에 내재된 원시 센서 데이터 값들, 상황 계획 카드, 상황 그룹핑 카드, 상황대기 카드, 상황종료 카드 등이 저장된다. 블랙보드에 저장되는 상황카드 들의 흐름을 도식화 하면 [그림 3]과 같다. 즉, 블랙보드에 입력된 상황카드를 보고 상황계획이 수행되어 상황카드가 생성되며, 생성된 상황계획에 따라 팀을 구성하고, 상황에 따라 대기 또는 완료상태가 진행 된다.



[그림 3] 블랙보드의 상황카드 흐름도

## 4. 시뮬레이션

본 논문에서는 유비쿼터스 레스토랑을 도메인으로 하여 SI-CAM의 프로토타입 을 설계 및 구현하였다. 본 논문에서는 서로 다른 기능을 가진 가상의 지능형 개체들의 훈련 데이터 파일 및 Instance 파일을 미리 생성하여 블랙보드에 저장하고 다음과 같은 시나리오에 대하여 실험하였다.



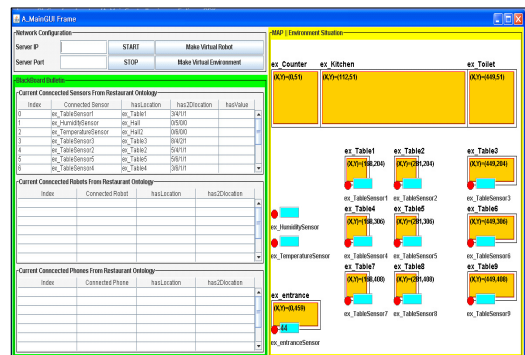
#### 4.1 시나리오

실험을 위한 시나리오는 GUI를 통하여 센서 정보를 입력하여, 어떻게 상황을 인지하고, 인지한 상황에 대하여 계획을 수립한 후, 수립한 계획이 어떻게 진행되는지를 수행하기 위한 시나리오이다. [그림 4]에서 오른쪽 창은 가상의 레스토랑 환경을 의미하며, 왼쪽의 창들은 시뮬레이션의 시작/종료 및 블랙보드의 내용을 의미한다. 시뮬레이션은 다음과 같은 순서로 진행된다.

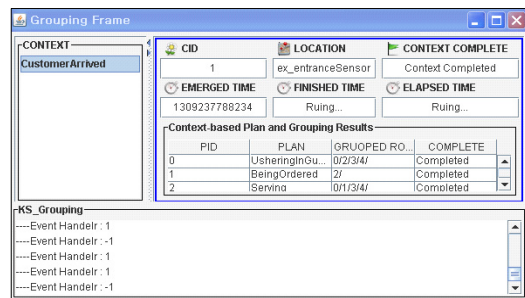
- 1) [그림 4] GUI 창에서 Start 버튼을 누른다.
- 2) 블랙보드에 저장된 환경 정보 온톨로지를 통하여 개체들을 [그림 4]의 GUI 오른쪽 화면에 보여준다. 환경정보 개체들은 kitchen, entrance, counter, toilet, 그리고 다수의 table들로 구성되며, 숫자 데이터들은 센서 데이터 값을 의미한다.
- 3) [그림 4]의 GUI에 입구(entrance) 센서 필드에 센서 값을 44 이상으로 입력한다.
- 4) 상황인지 모듈은 블랙보드에 저장된 상황모델을 통하여 CustomerArrived 상황을 발생시키고 이에 해당하는 상황카드를 생성하여 저장한다.
- 5) 상황계획 모듈은 블랙보드에 상황카드가 생성되면 해당되는 상황계획카드를 생성하고 이를 블랙보드에 저장한다.
- 6) 그룹핑 모듈은 상황계획 카드가 생성되면 각 계획에 맞는 지능형 개체들을 그룹핑하고 상황그룹 카드를 생성한 후 이를 블랙보드에 저장한다. 예를 들면, 생성된 상황그룹카드는 [그림 5]에서 처럼 context 창에서 CustomerArrived 상황이 발생 되면 해당 상황 계획이 Serving일 때, 그룹 된 로봇들의 ID가 0, 1, 3, 4임을 볼 수 있다.
- 7) 협업모듈은 블랙보드에 생성된 상황그룹 카드로부터 UsheringGuests 계획을 지능형 개체들이 수행할 수 있도록 로봇들의 협업을 도모한다. 로봇들의 작업이 끝나면, 계속 수행되어야 할 계획인 BeingOrdered, Serving, HelpingPayBill 등이 있으므로, UsheringGuests 계획에 대하여 계획완료를 기록한 후, 해당 카드를 상황대기

카드로 변경한다.

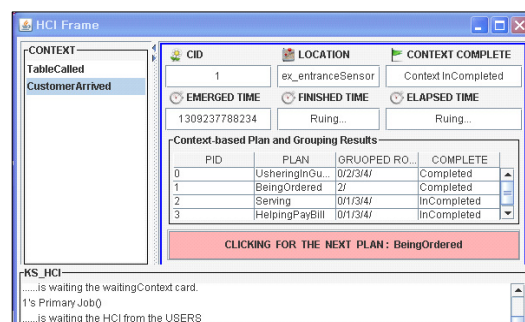
- 8) 계획표시 모듈은 블랙보드에 생성된 상황대기 카드로부터 다음 계획을 수행하기 위한 사용자 상호작용을 기다린다. 다음 계획의 진행 여부는 버튼 클릭 이벤트로 대체하며, 버튼 클릭 시 다음 계획이 “BeingOrdered”임을 볼 수 있다([그림 6] 참조).



[그림 4] 상황인지 미들웨어의 시뮬레이션 환경



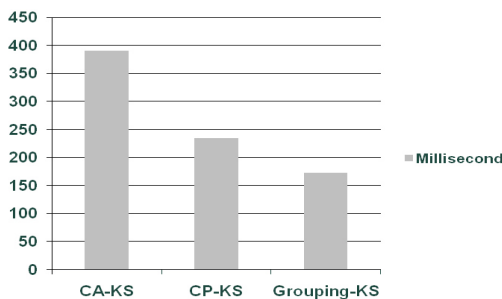
[그림 5] 그룹핑 모듈의 구성



[그림 6] 계획 표시모듈의 구성

## 4.2 시뮬레이션 실험 결과

본 연구의 상황인지 미들웨어는 블랙보드 제어기와 7개의 모듈이 스레드로 구현되어 작동되는 멀티스레드 시스템이며, 각 스레드의 우선순위는 동등하게 하였다. 개발도구는 JAVA JDK1.6이며, 채택한 실험을 위해서 55개의 클래스와 3개의 object properties, 그리고 6개의 data properties, 29개의 인스턴스, 74개의 규칙을 사용하였다. 또한 상황모델의 생성을 위하여 대표적 온톨로지 언어인 OWL API와 protégé\_4.1\_beta를 사용하였고, 추론을 위해서 Pellet 2.2.2를 사용하였다.



[그림 7] 상황인지 측정실험

개발된 상황인지 미들웨어는 우선 각 모듈의 성능을 중심으로 실험 하였다. [그림 7]에서, 온톨로지를 이용하는 상황인지 모듈(CA-KS)의 데이터는 상황 인지 시간을 측정한 실험으로 0.04초에 가까운 측정시간을 보였다. 즉, 사용자로부터 가상의 센서 데이터를 받고, 상황인지가 완료되는 시간의 측정치를 의미한다. 또한 상황계획 모듈(CP-KS) 데이터는 주어진 하나의 상황에 따라 10개의 규칙을 사용하여 상황계획의 완료시간을 측정한 것이다. 그룹핑 모듈(Grouping-KS)의 데이터는 ID3기반 엔진이 객체를 구분하여 분류되는 시간을 측정한 것으로 0.015에 근접한 속도로 가장 작은 수행시간을 보여주고 있다. 가상의 레스토랑을 문제영역으로 실험한 환경에서 총 8대의 가상로봇과 상황인지 미들웨어의 총 수행시간의 결과는 0.08초를 보여주었다. 시간 측정은 Celeron(R) Dual-Core CPU,

1.90GHz, 2.96GB Ram 환경에서 측정하였다.

## 5. 결 론

본 연구에서는 기존의 상황 인지 컴퓨팅 환경에 사회적 지능성의 개념을 도입하여 새로운 상황 인지 환경을 정의하고 이를 위한 상황인지 미들웨어를 설계 및 구현하였다. 본 연구의 중요성은 다음과 같다. 첫째, 본 연구의 시스템은 블랙보드기반으로 설계되어 다수의 모듈들이 독립적으로 주어진 임무를 수행함으로써 지능적 서비스를 제공한다. 둘째, 시스템은 상황 계획에 따라 생성된 계획을 수행한다. 셋째, 시스템은 주어진 상황에 따라 지능적 객체들을 그룹으로 구성하여 협력을 통해 임무를 수행한다. 시스템의 타당성을 검증하기 위하여 가상의 시나리오를 설정하여 실험을 하였으며, 실제 시스템 구축의 가능성을 보여주었다. 또한 각 모듈의 실제 구현 성능을 비교하기 위하여 시간을 기준으로 실험하였다.

본 연구의 제한점 및 향후연구는 다음과 같다. 첫째, 실험을 위한 데이터는 가상의 데이터를 사용하였기 때문에, 향후에는 센서기기를 통한 실제 데이터를 사용하여 실험할 계획이다. 둘째, 본 연구의 상황인지 미들웨어는 상황모델을 확장 가능한 온톨로지 기반으로 구성하였지만 향후에는 보다 다양한 지식체제로 설계하여 여러 종류의 상황인지 응용 시스템에의 적용 가능성을 모색할 것이다. 또한 개별적 객체인 상황인지 로봇의 구체적인 설계를 통하여 좀 더 효율적인 사회적 지능성을 실험할 것이다. 마지막으로 협업을 위한 계획 생성 및 협업전략에 대한 보다 구체적인 메커니즘이 요구된다.

## 참 고 문 헌

- [1] Chen, H., T. Finin, and A. Joshi, *An Ontology for a Context Aware Pervasive Computing Environment*, Proceedings of the

- IJCAI workshop on ontologies and distributed systems, (2003), pp.197-207.
- [2] Dey, A. and G. Abowd, *Towards a better understanding of context and context-awareness*, In Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, (2000), pp.304-307.
- [3] Dey, A., G. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", *Human Computer Interaction (HCI) Journal*, Vol.16, No.2/4(2001), pp.97-166.
- [4] Go, Y. and J. Sohn, *Probabilistic approach to model intelligent robot services*, Advanced Communication Technology, ICACT, (2006), pp.1221-1224.
- [5] Gu, T., X. Wang, H. Pung, and D. Zhang, *An ontology-based Context Model in Intelligent Environment*, Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, (2004), pp.270-275.
- [6] Gu, T., H. Pung, and D. Zhang, "Toward an OSGi-Based Infrastructure for Context-Aware Applications", *IEEE Pervasive Computing*, Vol.3, No.4(2004), pp.66-74.
- [7] Hwang, W., J. Park, H. Suh, H. Kim, and I. Suh, *Ontology-Based Framework of Robot Context Modeling and Reasoning for Object Recognition*, LNAI 4223, (2006), pp.596-606.
- [8] Jaygarl, H., K. Oyama, J. Xia, and C. Chang, *HESA : A Human-Centric Evolvable Situation-Awareness Model in Smart Home*, ICOST, LNCS 5129, (2008), pp.153-160.
- [9] Kindberg, T and J. Barton, "A Web-based nomadic computing system", *The International Journal of Computer and Telecommunications Networking*, Vol.35, No.4(2001), pp.443-456.
- [10] Ko, E., H. Lee, and J. Lee, *Ontology-Based Context Modeling and Reasoning for U-Health Care*, IEICE Trans Inf. and Syst., (2007), pp.1262-1270.
- [11] Labella, T., M. Dorigo, and J. Deneubourg, *Self-Organized Task Allocation in a Group of Robots*, Technical Report No. TR/IRIDIA/2004-6, Universite Libre de Bruxelles, 2004.
- [12] Ranganathan, A. and R. Campbell, *A Middleware for Context-Aware Agents in Ubiquitous Computing Environments*, In proceedings of AC/IFIP/USENIX International conference on Middleware, (2003), pp.143-161.
- [13] Rodic, D. and A. Engelbrecht, *Social network as a Coordination Techniques for Multi-Robot Systems*, Proceedings of the International Conference on System Design and Applications, (2003), pp.503-513.
- [14] Schilit, B. and M. Theimer, *Disseminating Active Map Information to Mobile Hosts*, IEEE Network, (1994), pp.22-32.
- [15] Schilit, B., N. Adams, and R. Want, *Context-aware computing applications*, In IEEE Workshop on Mobile Computing Systems and Applications, (1994), pp.85-90.
- [16] Sterns, M. and N. Windelinckx, *Combining Planning with Reinforcement Learning for Multi-robot Task Allocation*, Proceedings of adaptive agents and MASII, LNAI 3394, (2006), pp.260-274.
- [17] Strang, T. and C. Linnhoff-Popien, *A context modeling survey*, Workshop on Advanced Context Modeling, Reasoning and Management as Part of UbiComp, The 6th International Conference on Ubiquitous Compu-

- ting, (2004), pp.33-40.
- [18] Wang, X., D. Zhang, T. Gu, and H. Pung, *Ontology Based Context Modeling and Reasoning using OWL*, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, (2004), pp.18-22.
- [19] Zhang, D., T. Gu, and X. Wang, "Enabling Context-aware Smart Home with Semantic Technology", *International Journal of Human-friendly Welfare Robotic Systems*, Vol.6, No.4(2005), pp.2-20.

## ◆ 저 자 소 개 ◆



**서 주 희 (crazyDMP@gmail.com)**

국민대학교 컴퓨터공학부에서 학사, 석사 학위를 하고 현재 미국 University of Oklahoma에서 컴퓨터공학 박사과정 중이다. 한국정보과학회, 한국정보처리학회, 한국IT서비스학회지, IEEE International Conference on Tools with AI(ICTAI), LNEE 등의 학술지에 논문을 게재하였다. 주요 관심분야는 지능형 에이전트, 컨텍스트 어웨어니스 시스템, 지능형 휴머노이드 로봇 등이다.



**우 종 우 (cwwoo@kookmin.ac.kr)**

미국 Illinois Institute of Technology에서 컴퓨터공학 박사를 받고, 현재 국민대학교 컴퓨터공학부 교수로 재직 중이다. Artificial Intelligence in Medicine, ITS, PRICAI, 한국정보과학회논문지, 한국정보처리학회논문지, 한국 IT서비스학회지 등에 다수의 논문을 실었다. 주요 관심 분야로 컨텍스트 어웨어니스 컴퓨팅, 증강현실 시스템, 지능형 에이전트, 모델링&시뮬레이션 등이다.