

# P-VoT을 사용한 유비쿼터스 컴퓨팅 시뮬레이터 제작

## Creating Ubiquitous Computing Simulators Using P-VoT

서진석, 오세웅, 고광훈\*, 김정현\*  
 동의대학교, 포항공과대학교\*

Seo Jin-Seok, Oh Sei-Woong, Goh Gwang-Hoon\*,  
 Kim Gerard J.\*

Dong-eui Univ., Pohang Univ. of Sci. and  
 Tech.(POSTECH)\*

### 요약

가까운 미래에 우리의 생활환경이 유비쿼터스 컴퓨팅 환경으로 변모될 것으로 기대되고 있다. 다가올 환경이 제공하는 다양한 서비스를 제공받기 위해, 인간과 유비쿼터스 컴퓨팅 환경 사이의 상호작용에 대한 많은 시나리오와 예측이 나오고 있지만, 대부분은 아직 실제 구현된 상태가 아닌 문서상의 검증되지 않은 아이디어에 불과하다. 이 논문은 가상의 유비쿼터스 컴퓨팅 환경을 위한 시뮬레이터를 쉽고 빠르게 제작할 수 있도록 개발된 P-VoT이라는 컴포넌트 기반의 대화식 가상환경 제작도구를 소개한다.

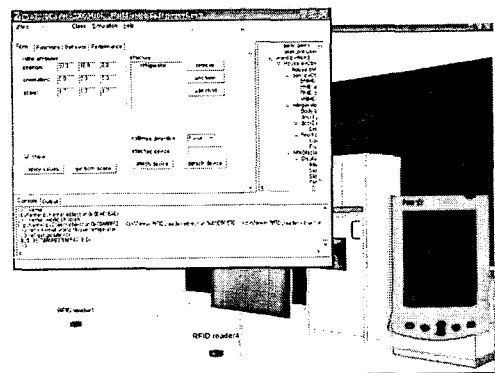
### Abstract

It is projected that, in a near future, our living environment will transform into the ubiquitous computing environment. Many scenarios and projections of how humans will actually interact with such an environment to receive various services have been made, but most of them remain to be untested ideas on paper without implementation. In this paper, we propose to use P-VoT, a component based interactive authoring toolkit for virtual environments, to quickly build a virtual ubiquitous computing environment.

## I. 서론

가까운 미래에는 우리의 일상생활을 위한 환경 곳곳에 컴퓨팅이 기존의 환경과 경계 없이(seamlessly) 산재하게 될 것이다. 이러한 유비쿼터스 컴퓨팅 환경에서 사람들은 언제, 어디서든 컴퓨터와 상호작용하게 될 뿐만 아니라, 컴퓨터가 자동으로 사람들의 요구를 예측하여 서비스하게 될 것이다. 현재는 미래의 유비쿼터스 컴퓨팅 환경을 미리 상상해 볼 수 있는 많은 예상 시나리오들이 나오고 있지만, 대부분은 아직 검증되지 않은 아이디어에 불과하다. 이는 그 환경을 실제 물리적 환경으로 구축하기 위한 기술이 아직 존재하지 않거나 혹은 성숙되지 않았기 때문이다. 더불어, 구축될 환경을 유용성, 사생활 보호, 보안, 비용 대비 효과 등의 측면에서 종합적으로 검증할 수 있는 방법도 없는 것이 현실이다. 만약 다양한 형태

의 유비쿼터스 컴퓨팅 환경을 미리 검증해 볼 수 있다면, 유비쿼터스 컴퓨팅 관련 분야의 종사자들이나 회사의 입장에서 보면 연구 및 개발하는 데 있어서 많은 노력과 비용을 절감할 수 있을 것이다.



▶▶ 그림 1. P-VoT의 동작 화면

이 논문은 P-VoT(POSTECH Virtual reality system development Tool)이라는 컴포넌트 기반의 대화식 저작도구를 사용하여 가상의 유비쿼터스 컴퓨팅 환경을 쉽고 빠르게 구축할 있다는 것을 보여준다.([그림 1] 참조) P-VoT을 이용하여 구축된 가상의 환경을 통해 시스템 설계자들은 다양한 상호작용의 유용성을 평가하거나 시나리오를 검증해볼 수 있게 된다.

P-VoT은 원래 저자가 제안한 "CLEVR (Concurrent and LLevel by Level Development of VR System)"[1]라는 가상현실 시스템 개발을 위한 방법론을 지원하기 위해 개발된 저작도구이다. P-VoT을 사용하여 개발자들은 파이썬(Python) 언어로 된 간단한 스크립트 명령과 GUI를 통해 대화식으로 가상환경을 구축할 수 있다. P-VoT의 통합개발환경(IDE)은 가상 객체의 행위를 묘사하기 위한 Statecharts(David Harel[2]이 제안한 상태도) 편집기, 재사용 가능한 가상 객체 및 상호작용 객체들의 라이브러리, 스크립트 편집기, XML 기반의 마크업 언어인 PVML 해석기 등을 포함한다[3]. CLEVR/P-VoT을 사용하면 가상환경 내의 객체들의 기능이나 행위를 실행시간에 동적으로 재구성 가능하며 그 결과를 즉석에서 확인할 수 있다. 이러한 대화식 방식을 통해, 객체 행위의 정확도, 상호작용의 유용성 등을 빠른 시간에 최적화 할 수 있다. P-VoT에서는 가상의 유비쿼터스 환경을 구축하기 위해 필요한 컴포넌트들을 "(1) 센서 컴포넌트"와 "(2) 디스플레이 컴포넌트"와 같이 2가지 부류로 나누어 라이브러리 형태로 제공한다. 이 컴포넌트들은 기존의 형태로도 재사용 가능하지만, 내부의 기능이나 행위를 개발자가 원하는 형태로 쉽게 변경할 수 있도록 고안되어 있다.

## II. 관련 연구

본 연구와 관련된 분야라고 할 수 있는, 유비쿼터스

컴퓨팅, 가상환경의 프로토타이핑 및 저작, 유용성 평가 등과 관련된 연구는 많이 찾아볼 수 있지만, 이 논문에서는 지면 관계상 "유비쿼터스 컴퓨팅 환경을 위한 시뮬레이터"를 직접 언급하고 있는 연구만 소개하고자 한다.

미국 HP(Hewlett Packard) 연구소에서는 2002년에 새로운 형태의 무선 디바이스, 디바이스 간의 상호작용, 무선 인프라스트럭처 및 이를 바탕으로 하는 서비스를 시뮬레이션 할 수 있는 WISE[4]라는 시뮬레이션 도구를 발표한 바 있다. 그리고 WISE와 "웨이크 II 아레나" 3차원 게임 엔진을 통합하여 UbiWise[5]라는 유비쿼터스 컴퓨팅 환경을 위한 가상 시뮬레이터를 개발하였다. UbiWise는 사용자를 중심으로 하는 1인칭 시점뿐만 아니라 사용자가 조작하는 장치나 물체를 자세히 보여주기 위한 클로уз업 시점까지 가시화하여 준다. UbiWise는 또한 여러 명이 동시에 시뮬레이터에 참여할 수 있도록 하기 위해 다중 클라이언트를 지원한다. 이 연구는 본 논문에서 말하고자하는 연구 동기와 많은 부분이 일치하고 있지만, UbiWise는 무선 디바이스 및 무선 환경을 위한 소프트웨어와 디바이스들 간의 네트워킹 프로토콜의 검증에 중점을 두고 있는 반면, 본 연구는 유비쿼터스 컴퓨팅 환경을 위한 시뮬레이터의 쉽고 빠른 저작에 중점을 두고 있다. [5]의 저자인 Barton도 역시 유비쿼터스 컴퓨팅 환경을 위한 실용적이고 효과적인 시뮬레이터가 되기 위해서는 쉬운 저작이 중요하다고 기술하고 있다.

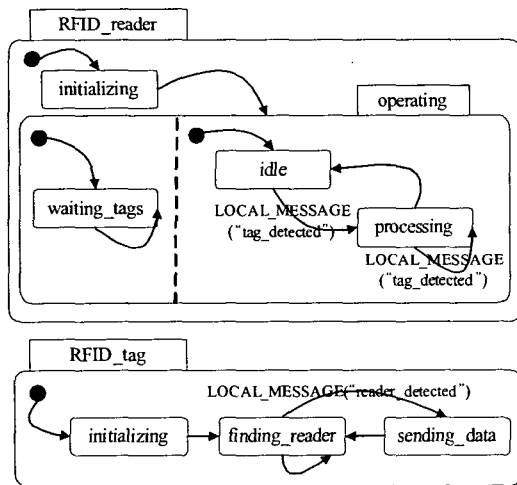
## III. 컴포넌트: 센서와 디스플레이

P-VoT에서는 유비쿼터스 컴퓨팅 환경을 구성하게 될 여러 가지 디바이스들을 센서와 디스플레이라는 두 부류의 컴포넌트들로 제공하는데, 이번 장에서는 그 중 몇 가지만 소개하고자 한다. 컴포넌트의 행위 모델은 Statecharts와 파이썬 기반의 스크립트언어로 정의되는데, 보통 여러 추상화 단계를 가지고

있다. 이는 Statecharts가 기존의 다른 상태전이도와는 달리 계층적 구조(hierarchy)를 지원하기 덕분인데, P-VoT에서는 한 객체를 여러 추상화 단계로 나누어 행위 및 기능을 정의할 수 있도록 지원하고 있다.

### 1. 센서 컴포넌트: RFID

RFID(Radio Frequency Identification)는 기본적으로 한 개의 리더와 여러 개의 태그로 구성되는데, 리더가 발생시킨 전파를 수신하는 태그는 다시 리더에게 자신만의 데이터를 송신하는 식으로 작동한다.



▶▶ 그림 2. RFID 리더와 태그의 Statecharts

[그림 2]는 RFID 리더와 태그의 행위 모델 중 특정 단계의 추상화를 보여주고 있다. 태그가 리더로부터 미리 정의된 특정 영역 안으로 들어가게 되면 "waiting\_tags"라는 상태(state)에서 "tag\_detected"라는 지역 메시지(local message)를 발생시키는데 (주: 메시지를 발생시키는 파이썬 스크립트가 "waiting\_tags"라는 상태의 정의에 포함되어 있다.), 이 메시지에 의해 "idle" 상태에서부터 "processing" 상태로의 전이(transition)가 이루어진다. 한편, 태그는 주기적으로 리더로부터 발생되는 전파를 탐지하

다가, 리더를 찾게 되면 자신의 데이터를 리더에게 전송한다. Statecharts의 각 상태에서 구체적으로 어떤 동작과 연산을 수행하게 될지는 스크립트 언어로 정의하게 되는데, 다음 [그림 3]이 한 예를 보여주고 있다. 이 스크립트는 특정 태그가 가상 공간상에 존재하는 여러 개의 리더 중에서 어떤 리더로부터 전파를 수신하고 있는지를 찾는 코드이다.

```
for reader in self.kernel.world.readers:
    if self.distance(reader) < reader.range:
        self.detected_readers.append(reader)
```

▶▶ 그림 3. RFID 태그의 스크립트 코드

위의 [그림 3]에서 볼 수 있듯이 태그는 미리 정의되어 있는 "distance"라는 함수를 사용하여 리더를 찾고 있다. 이 함수는 [그림 2]에서 볼 수 있는 태그의 행위 모델에서 보다 하위의 추상화 단계에 정의되어 있다. 다시 말해서, 하위의 추상화 단계에서는 보다 기초적인 알고리즘, 하위 상태들(sub-states), 세부 이벤트나 데이터 교환, 네트워크 프로토콜 등을 정의하게 된다. 게다가, (데이터를 무선으로 전송하는) 본래의 RFID 기능을 다른 용도로 사용 할 수 있는데, 예를 들어, 여러 개의 RFID 리더를 사용하여 물체의 위치를 추적하는 용도(주: 이 예는 다음 장에서 볼 수 있다.)로 사용할 수도 있다. 이 경우 [그림 2]에서 볼 수 있는 Statecharts보다 상위 추상화 단계의 행위 모델을 정의하면 된다.

현재 P-VoT에는 RFID 외에도 가상의 유비쿼터스 컴퓨팅 환경을 구축하는데 유용하게 사용할 수 있는 여러 종류의 가상 센서 컴포넌트들이 존재하는데, 자이로(gyro) 센서, 화학 센서, 온도 및 습도 센서, 초음파 센서, 터치 센서 등이 그 예이다. 추후에는 음성 인식 센서, 동작 인식 센서 등 보다 다양한 형태의 상호작용이 가능하도록 확장할 계획이다.

### 2. 디스플레이 컴포넌트: LCD 디스플레이

유비쿼터스 컴퓨팅 환경에서는 다양한 정보를 사용

자에게 제공하기 위해 여러 종류의 디스플레이 장치가 사용될 것이다. 그 대표적인 예가 바로 영상을 표현하기 위한 장치로, 아주 작은 크기의 LCD로부터 대형 프로젝션 스크린 등 다양한 형태의 장치가 이에 포함된다. 하지만, 대부분의 영상을 표현하기 위한 장치들은 각자의 “형태”(예들 들어, 평판, CRT, LED, 플렉시블 디스플레이 등)만 다를 뿐, 영상이나 정보를 표현하고자 하는 기능이나 행위는 동일하다. 영상 디스플레이 장치는 보다 유용한 유비쿼터스 컴퓨팅 환경의 구성 요소가 되기 위해서, 일상생활의 다른 물체(창문, 유리, 벽면 등) 혹은 다른 센서나 장치(터치스크린, PDA, 태블릿 PC 등)와 결합되어 복합적인 디바이스로 정의 될 수 있다. 영상 디스플레이 컴포넌트 외에도, 향후에는 사운드, 음성, 햅틱(haptic), 촉각(tactile)을 표현하기 위한 디스플레이 컴포넌트도 추가하여 보다 다양한 유비쿼터스 컴퓨팅 환경을 위한 시나리오의 시뮬레이션도 가능하게 할 계획이다.

#### IV. 예시 환경

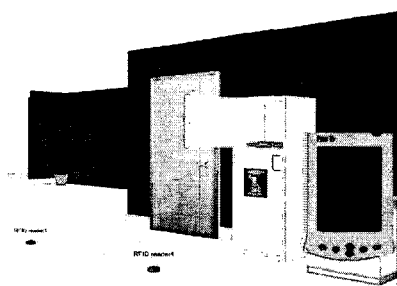
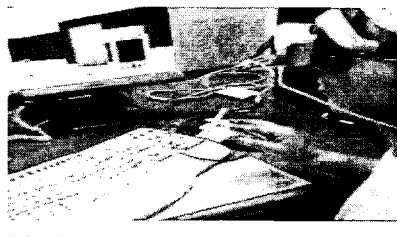
이번 장에서는 전 장에서 기술한 컴포넌트들을 사용하여 가상의 유비쿼터스 컴퓨팅 환경을 구축하는 예를 보여준다. 예시하는 환경([그림 4] 참조)은 가까운 미래의 집안을 상상한 것으로 다양한 유비쿼터스 기술이 일상생활을 위한 환경(가전제품, 가구, 벽 등)에 스며들어 있다. 이 환경에서 우리는 다음과 같은 시나리오를 시뮬레이션하고 검증할 수 있다.

- 스마트 냉장고 : 포장에 RFID 태그가 내장된 음식물을 냉장고에 넣게 되면, 냉장고의 RFID 리더가 음식물에 대한 정보를 수신하여 내부 저장 장치에 보관한다. 사용자는 냉장고에 문에 부착된 LCD 디스플레이와 터치스크린을 통해 냉장고에 들어 있는 음식물에 관한 정보(종류, 양, 유통기한 등)를 검색할 수 있다.([그림 5] 참조)
- 물체 및 사람 위치 추적: RFID는 방, 벽, 바닥, 가구 등 집안 곳곳에 내장되어 있다. 그 결과,

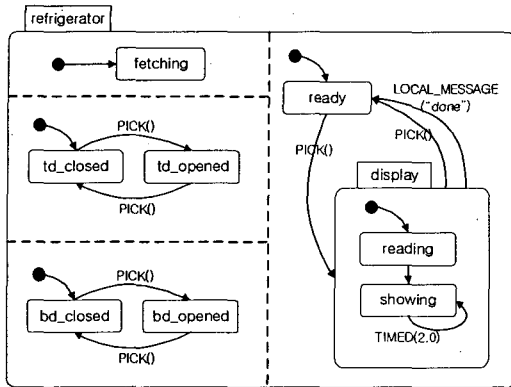
RFID 태그가 내장된 물체나 혹은 물체를 가지고 있는 사람의 위치를 추적할 수 있다.

- 스마트 공조 시스템 : 온도 및 습도 센서와 사람 위치 추적 시스템을 이용하여 집안의 여러 공조 시스템을 제어한다.
- 정보 키오스크(kiosk) : 터치스크린 센서와 영상 디스플레이 장치가 벽, 거울, 창문 등에 내장되어 각종 정보를 검색하거나 집안의 여러 장치를 제어할 수 있다.
- 유비쿼터스 PDA : PDA는 집안의 여러 유비쿼터스 디바이스들과 연결 되어, 위에서 열거한 여러 서비스를 집안에서 혹은 집밖에서 제공할 수 있다.

[그림 4]는 위에서 열거한 시나리오를 가상환경에서 검증해보는 장면을 보여준다. 이 환경은 P-VoT에서 미리 정의된 재사용 가능한 컴포넌트들을 조합하고 간단한 스크립트를 추가하여 구축된 것이다. 사용자는 6-DOF(Degrees-of-Freedom)를 갖는 트래커(tracker)와 HMD(Head Mounted Display)를 사용하여 구현된 몰입형 가상환경에서 시나리오를 검증하거나 유용성을 평가할 수 있다.



▶▶ 그림 4. 가상 유비쿼터스 환경의 예



▶▶ 그림 5. 냉장고의 Statecharts

“스마트 냉장고” 시나리오의 시물레이션을 위해, 일단, 미리 정의 된 RFID 리더와 태그, 영상 디스플레이, 터치스크린 등의 컴포넌트를 불러와서(import) 적당한 객체(냉장고, 음식물 등)의 자식 객체(sub-object)로 추가하여 준다. 다음에, 각 음식물에 추가된 RFID 태그 컴포넌트의 속성인 “tag\_ID”에 각 음식물의 종류, 유통 기한, 구입 일자 등을 기록하여주고, 냉장고의 행위 모델을 정의하면 시나리오가 완성된다. [그림 5]에 보이는 냉장고의 Statecharts는 네 부분으로 구성된 병렬(주: 본래 Statecharts의 정의에서는 “concurrent”라는 용어를 사용한다.) 프로세스를 표현하고 있다. Statecharts의 좌측 하단에 있는 2개의 병렬 프로세스는 냉장고 문이 열리고 음식물이 냉장고 안으로 들어가거나 나가면 냉장고가 지니고 있는 음식물에 대한 정보를 갱신하는 것을 표현하고 있다. 그리고 우측의 프로세스는 사용자가 냉장고의 터치스크린을 만지면 “reading” 상태에서 음식물에 관한 정보를 읽어 오고 “showing” 상태에서 그 정보를 LCD 화면에 출력한다는 것을 보여준다. 이상과 같이 전체 저작 과정은 매우 쉽고 간단한데 그 과정을 요약하면, 객체를 조합하고, 냉장고의 행위 모델을 위한 Statecharts를 정의하고, 스크립트 코드를 추가하는 방식으로 진행된다.

## V. 결론

이 논문에서 우리는 가상의 유비쿼터스 컴퓨팅 환경을 구축하기 위한 P-VoT이라는 컴포넌트 기반의 대화식 저작도구를 소개하였다. P-VoT에서는 쉽고 빠른 가상 유비쿼터스 컴퓨팅 환경 저작을 위해 “센서”와 “디스플레이”라는 2 부류의 컴포넌트를 미리 정의하였는데, 각 컴포넌트는 여러 추상화 단계의 모델을 지니고 있고, 각 추상화 단계별로 행위와 기능을 Statecharts와 스크립트를 통해 정의한다. 예시 환경의 저작을 통해 보여주었듯이, 우리는 P-VoT과 같은 대화식 저작도구가 유비쿼터스 시나리오의 검증 및 유용성 평가에 매우 효과적으로 활용될 수 있다고 확신하고 있으며, 향후에는 P-VoT의 사용자 환경(UI)을 단순화 하고, 스크립트 언어 입력이 필요한 부분을 줄이는 등 더욱 더 쉽고 실용적인 도구로 발전시킬 계획이다.

### ■ 참고 문헌 ■

- [1] Seo, J. and Kim, G. J., Explorative Construction of Virtual Worlds: An Interactive Kernel Approach, Proceedings of the ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Application in Industry (VRCAI), 2004.
- [2] Harel, D., STATECHARTS: a Visual Formalism for Complex Systems, Science of Computer Programming, 8, 231-274, 1987.
- [3] Seo, J. and Kim, G. J., Design for Presence: A Structured Approach to Virtual Reality System Design, Presence: Teleoperators and Virtual Environments, 11(4), 378-403, 2002.
- [4] Vijayaraghavan, V. and Barton J. J., WISE - A Simulator Toolkit for Ubiquitous Computing Scenarios, UbiTools 01 workshop, 2001.
- [5] Barton, J. J. and Vijayaraghavan, V., UbiWise, A Simulator for Ubiquitous Computing Systems Design, Technical Report, HP Labs, HPL-2003-93, 2003.