

센서들의 시스템으로의 통합과 자동설정을 지원하는 프레임워크\*

차영록<sup>o</sup>, 정유철, 이동만  
 한국정보통신대학교  
 {u00cha<sup>o</sup>, chungyc, dlee}@icu.ac.kr

A framework for supporting auto-configuration and integration of location sensors

Youngrock Cha<sup>o</sup> Yoo Chul Chung, Dongman Lee  
 Information and Communication University

요 약

본 연구에서 소개하는 프레임워크는 유비쿼터스 컴퓨팅 환경에서 위치센서가 새롭게 배치 될 때, 그 위치센서를 위치관리자(location manager)에 통합시키고, 통합에 필요한 설정을 지원한다. 그 프레임워크는 (1) 사용자를 위해 설치 및 설정에 대한 복잡성을 줄여줘야 하며, (2) 통합에 필요한 여러 설정을 다루어야 하고, (3) 다양한 종류의 센서배치를 지원할 수 있어야 한다. 본 연구에서 제안한 프레임워크는 센서에서 최소한의 설정을 가지고 위치 관리자로의 통합을 지원한다. 또한 통합에 필요한 설정으로 변환행렬(transformation matrix)과 식별자(adapter ID)를 다룬다. 마지막으로 센서의 타입을 분류하여 그것에 맞는 설정 및 통합을 지원 한다.

1. 서 론

유비쿼터스 컴퓨팅 환경에서는 컴퓨터 및 네트워크 기능이 내재된 지능객체(smart object)들이 서로 상호 작용하며 사용자에게 최적의 서비스를 제공한다[1]. 유비쿼터스 컴퓨팅 환경에서 사용자의 위치(location)는 가장 중요한 상황정보(context) 중 하나이다. 때로는 이 위치상황정보만을 유일한 판단기준으로 하여 지능객체들이 사용자에게 서비스를 제공하기도 한다. 이를 지원하는 위치 관리자(location manager)는 사용자의 위치 정보를 관리하는 중요한 컴포넌트로서, 여러 센서들이 보고하는 위치 정보를 융합하여 보다 정확한 위치 정보를 생산한다. 대부분의 위치 시스템에서 사용되는 어댑터(adapter)의 역할은 이러한 기기종의 각 센서들이 하나의 위치 관리자에 통합 될 수 있도록 적합성을 제공해주는 것이다. 위치 센서시스템이 위치 관리자의 한 구성요소로써 통합되기 위해서는 각 어댑터마다 필요한 변수가 설정되어야 한다.

위치 관리자의 속성상 다양한 센서들을 그 환경 내에 배치할수록 더 정확한 위치 정보를 얻게 된다. 앞으로 센서의 가격이 대량구입이 가능할 정도로 충분히 낮아진다면, 사용자가 개별적으로 구입하여 그 센서들을 자신이 원하는 환경에 충분히 배치하게 될 것이다. 하지만 가정 내의 모든 사용자가 새롭게 배치되는 센서들의 어댑터를 설정할 수 있는 숙련된 관리자는 아니다. 그리고

다양한 센서들의 통합을 위해 수동적으로 설정하는 것도 쉬운 작업은 아니다. 또한 무선 통신 기술의 사용으로 센서들이 이동될 수 있고, 동적으로 그 시스템에 추가 및 삭제가 빈번하게 일어날 수 있을 것이다. 따라서 유비쿼터스 환경에서 일반적인 요구사항은 가능한 한 센서의 쉬운 설치 및 자동 설정을 지원하는 것이다.

본 논문은 위치센서 어댑터의 자동적 설정 및 위치 관리자로의 통합을 지원하는 프레임워크를 소개한다. 제안하는 프레임워크는 센서에서 최소한의 사용자 관여만을 가지고 위치 관리자로의 통합을 지원한다. 또한 위치 시스템의 통합을 위해서 어댑터에서는 각자의 좌표 시스템을 위치 관리자가 다루는 통합된 좌표 시스템으로 변환해준다. 그리고 위치 관리자가 다양한 센서들을 구별할 수 있도록 어댑터가 식별자(adapter ID)를 부여 받을 수 있도록 한다. 마지막으로 센서의 타입을 나누어서 그것에 맞게 설정 및 통합을 지원한다. 이 제안된 프레임워크는 유비쿼터스 컴퓨팅 환경을 가정 내에 구현하는 프로젝트인 Active Surroundings [2]에서 중요한 상황정보인 위치 정보의 생산을 다루는 기반으로 사용되고 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 연구에 관련된 기술과 연구들에 대해 알아보고, 3장에서는 디자인시 고려사항과 접근 방법에 대해, 4장에서는 프레임워크의 구성 및 상호작용 등의 접근법에 대해 기술한다. 5장에서는 구현 및 테스트에 대해, 마지막으로 6장에서는 결론 및 향후 연구 과제에 대해 기술한다.

\* 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업과 정보통신부 및 정보통신연구진흥원의 디지털미디어연구소 지원사업의 지원에 의한 것임

2. 관련 기술

기존 시스템으로의 통합에 필요한 설정을 자동적으로 지원해주는 프로젝트로는 DHCP[3], SECURE project[4], UPnP[5] 등이 있다. DHCP는 네트워크 장비의 네트워크 설정을, SECURE에서는 지능형 객체간의 보안 설정을, UPnP나 OSGi[6]에서는 홈네트워크에 지능형 객체의 설정을 자동화한다.

다양한 센서들을 지원할 수 있는 위치 관리 시스템(location management system)들로는 Gaia에서 제시하는 MiddleWhere[6]와 IBM China에서 제시하는 LORE(Location Operating Reference Model)[8] 등이 있다. 이들은 다중의 센서로부터 인지된 위치 데이터를 확률을 기반으로 융합하여 사용자 및 지능 객체의 위치를 보다 정밀하게 추적하는 것을 목표로 한다. 이들이 제안하는 위치 어댑터는 센서 테크놀로지에서 제공해주는 원시의 인터페이스와 통신하면서 위치 관리자와 끊임없이 상호 작동한다.

위치 관리 시스템에 새로운 센서가 구성요소로써 통합되기 위해, 어댑터에서 필요한 설정으로, 각 센서 고유의 좌표 시스템을 위치 관리자에서 다루는 좌표 시스템으로 변환해 주어야 한다. MiddleWhere와 LORE에서는 사용자가 일일이 이러한 설정을 해주거나 그 값을 구하기 위해서 필요한 정보를 기입해주어야 하는 단점이 있다. 이럴 경우, 숙련된 관리자가 아니라면 사용자가 센서를 설치하고 설정하는 것이 어렵게 된다.

따라서 우리는 LORE와 MiddleWhere가 가지고 있는 이러한 제약조건을 고려하여 어댑터의 자동 설정 및 위치 시스템으로의 자동 통합을 지원하는 프레임워크를 제안한다.

### 3. 디자인 고려사항

어댑터의 자동 설정 및 위치 시스템으로의 자동 통합을 위한 프레임워크의 요구사항은 다음과 같다.

첫째, 숙련된 관리자가 아닌 일반 사용자를 위해 위치 센서의 설치 및 설정에 대한 복잡성을 줄여야 한다.

둘째, 통합에 필요한 여러 설정을 다룰 수 있어야 한다. 새로운 위치 센서가 기존의 위치 시스템에 통합되기 위해서는 여러 가지 설정을 필요로 한다. 그 위치 센서가 특정 영역을 다루는 센서라면, 각 센서가 갖는 고유의 좌표계로 표현된 좌표를 위치 관리자에서 쓰는 좌표계에 맞추어 변환해 주어야 한다. 각 어댑터에서는 이러한 변환을 위해 Aura[9]에서 소개하는 변환행렬(transformation matrix)을 사용한다. 이러한 변환행렬을 기존의 위치 관리 시스템에서는 사용자가 설정해주어야 했다. 본 논문에서 제안하는 프레임워크에서는 이러한 설정값을 어댑터가 쉽게 얻을

수 있도록 지원해주는 것을 목표로 한다. 어댑터 식별자(adapter ID)는 위치 관리자가 센서를 구분할 수 있게 해주는 것으로 마찬가지로 어댑터가 필요로 하는 설정값이다.

셋째, 다양한 종류의 센서에 대해서도 지원이 가능해야 한다. 위치 센서 시스템은 그들이 제공하는 태그(tag) 형식의 객체를 추적하여 위치 정보를 얻는다. 이들 간의 통신 방식에 따라, 위치 센서 시스템은 query같은 pull 방식을 따르는 것들과 callback mechanism 같은 push 방식[10]을 따르는 것으로 나뉜다. 이러한 센서의 통신방법에 따라, 어댑터에 적절한 인터페이스를 설정해주어야 한다. 또한 위치 센서는 스마트리더나 IR센서 같은 point 센서 시스템과 Ubisense와 Cricket 같이 일정 영역을 다루는 range 센서 시스템으로 나뉜다. point 센서 시스템은 사용자 또는 객체가 그 point 센서 근처에 있는지 여부만을 인식한다. 반면 range 센서 시스템은 사용자 또는 객체가 range 센서 시스템의 영역 안 어느 위치에 있는지를 인식한다. 이러한 센서의 차이가 어댑터에 반영 되어야 한다.

## 4. 접근법

### 4.1 개요

위치 센서 시스템이 어떤 환경에 새롭게 배치되면 그 센서의 어댑터는 브로드캐스트를 통해 그 환경에 있는 자동설정자(automatic configurator)에게 자신의 참조자(reference)를 알린다. 자동설정자는 이에 응답하여, 어댑터에게 자신의 참조자를 알리고 어댑터는 자신에 설정되어 있는 정보를 자동설정자와 교환을 하고 위치 관리자로의 통합에 필요한 값들을 설정 받는다. 우리는 위치 센서시스템의 제조사에서 프레임워크에 제시한 어댑터의 규칙을 따라 센서의 어댑터를 구현할 것을 제안한다. 그리고 제조사에서 그 어댑터에 정확도(accuracy), 정밀도(precision), 통신 모드, 센서 종류, 최대 report rate 등을 설정 하게 하려 한다. 제조사에서 어댑터를 제공해주지 않는다면 사용자가 어댑터에 위치 센서시스템의 설명서를 가지고 GUI를 통해 설정 한다. 본 연구는 이렇게 설치 및 설정에 복잡성을 줄이려 한다.

하지만 range 센서 시스템의 어댑터 경우, 어댑터와 자동설정자간의 정보교환만으로 모든 설정이 완료 될 수는 없다. 대표적으로 이전 연구에서는 변환행렬을 어댑터에 설정하기 위해 필요한 정보를 사용자가 직접 기입해주어야 했다. 이러한 수동적 설정을 지양하기 위해 제안하는 프레임워크에서는 사용자에게 태그(tag)를 제공해준다. 각 위치 센서시스템에서 제공하는 태그는 현재 그 태그를 소지한 사용자의 위치를 알려주는 역할을 한다. 사용

자는 그 태그와 새롭게 배치하려는 위치 센서 시스템의 태그를 가지고 프레임워크에서 지정하는 방향으로 움직인다. 프레임워크에서는 두 센서 시스템으로부터 오는 위치정보들을 모은다. 같은 시간에 들어온 두 값은 물리적 관점에서 동일한 좌표지만, 각각의 좌표시스템으로 다르게 표현된 것이다. 이러한 값을 연관(bind)시켜 모아서 변환행렬을 구하기 위한 정보로 사용하고, 얻어진 변환행렬을 어댑터에 설정한다. 이러한 어댑터의 설정을 돕는 프레임워크의 시스템 구조는 그림 1에서 볼 수 있다.

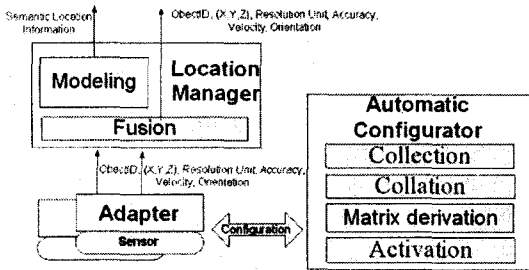


그림 1. 제안하는 프레임워크의 시스템 구조

#### 4.2 시스템 컴포넌트

제안하는 프레임워크는 크게 위치 관리자(location manager)와 어댑터(adapter) 그리고 자동설정자(automatic configurator)로 구성된다.

자동설정자는 어댑터가 위치 관리자로의 통합을 위해 요구되는 값들의 자동 설정을 지원한다. 어댑터에 필요한 식별자를 할당해주고, 앞서 말한 두 태그로부터 보고되는 위치 정보를 이용하여 변환행렬을 구한 다음, 그 값을 어댑터에 설정을 해준다.

어댑터는 위치 센서 시스템을 위치 관리자에 연결 시켜주는 역할을 한다. 특히, 어댑터에서는 각 센서가 가지는 고유의 좌표 시스템을 위치 관리자가 다루는 통합된 좌표 시스템으로 변환 해주는 변환행렬을 포함한다.

위치 관리자는 여러 센서의 어댑터들로부터 받은 위치 정보를 융합하여 보다 정확한 위치 정보를 생산한다. 또한 위치 관리자는 물리적 위치 정보(physical location)를 의미론적 위치 정보(symbolic location)로 바꾸어 주는 역할을 한다. 위치 관리자는 다양한 응용 프로그램의 요구에 응답하여 물리적 위치 정보 또는 의미론적 위치 정보를 생산한다.

#### 4.3 behavior and 상호작용

각 컴포넌트의 상태간의 이동은 메시지의 전송 및 수신

에 의해 일어난다. '/'자를 기준으로 '/'앞의 메시지는 현 상태에서 수신된 메시지를 의미하고, '/' 뒤의 메시지는 전송한 메시지를 의미한다. 그리고 각 상태에서 밑에 기술된 문장은 그 상태에서의 컴포넌트 행동을 가르킨다. 한 상태에서 같은 상태로의 재귀이동은 여러 메시지가 그 상태에서 수신 될 수 있다는 것을 의미한다.

그림 2는 프레임워크에서 자동 설정자의 상태 변이 다이어그램이다. INIT 상태에서 자동설정자는 어댑터로부터 자신을 찾는 메시지에 응답(REPLY 상태로 변이)하여 자신의 참조자를 요청한 어댑터에 알려준다. 자동설정자는 LISTEN 상태로 이동하여 어댑터로부터 정보를 받기를 기다린다. 어댑터로부터 필요한 정보를 받은 자동설정자는 DETERMINE 상태로 이동하고, 어댑터로부터 받은 정보에 근거하여 어댑터의 통신방법을 결정한다. 또한 센서의 종류가 Range 센서인 경우, 변환행렬의 필요성을 인지하여 사용자에게 위치 관리자가 관리하는 태그 하나를 제공해주고 변환행렬 설정에 필요한 행동을 취해 줄 것을 요청한다. 이후 자동설정자는 SETUP 상태로 이동하여, 어댑터와 위치 관리자로부터 위치 정보를 받는다. 연관된 값들을 근거로 자동설정자는 변환행렬 값을 이끌어 내어, 어댑터에 그 값을 설정해주고 CONFIGURED 상태로 이동한다. 자동설정자는 어댑터로부터 설정 완료 메시지(CONFIG\_DONE)을 받고, 다시 INIT 상태로 이동해 다른 어댑터로부터의 요구를 기다린다.

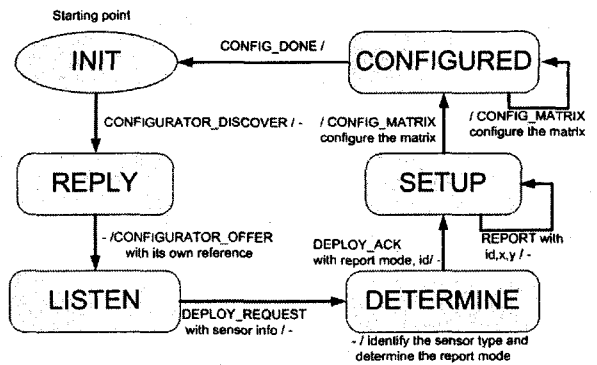


그림 2. 자동 설정자의 상태 변이 다이어그램

그림 3은 어댑터의 상태 변이 다이어그램이다. INIT 상태에서, 어댑터는 배치된 환경에 존재하는 자동 설정자의 참조자를 찾는 메시지를 브로드캐스트하고 나서, 자동설정자로부터 응답을 기다리는 LISTEN 상태로 이동한다. 자동설정자로부터 응답을 받으면, 자신에 설정되어 있는 정보를 자동설정자에게 요청 메시지 함께 보내고,

REQUESTING 상태가 된다. 자동설정자에서 통신모드, 식별자 그리고 변환행렬의 필요유무를 결정해주면 어댑터는 DETERMINED 상태가 된다. 그 후, SETUP 상태로 이동하여 자신의 고유 좌표계의 위치를 자동설정자에게 식별자와 함께 알려준다. 자동설정자로부터 변환행렬이 설정되면 CONFIGURED 상태로 이동하게 된다. 변환행렬의 갱신은 계속되지만 그 오차(표준편차 기반)가 일정 이하가 되면, 갱신은 끝나게 된다. CONFIGURED 상태에서 환경 내의 위치관리자에 통합되기 위해서, 위치관리자를 찾는 메시지를 브로드캐스트하여 찾고, 등록 과정을 거쳐, 위치 관리자에 통합되게 된다. 이 과정에서 REGISTER, INTEGRATION, INTEGRATED의 상태 변이 과정을 거친다.

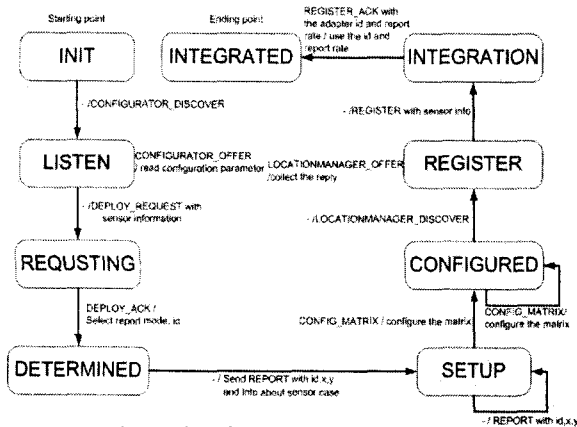


그림 3. 어댑터의 상태 변이 다이어그램

그림 4는 위치 관리자의 상태 변이 다이어그램이다.

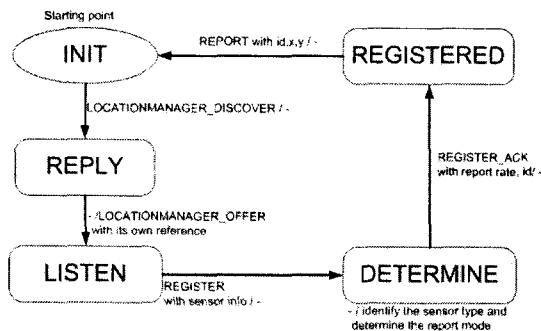


그림 4. 위치 관리자의 상태 변이 다이어그램

INIT 상태에서, 위치관리자는 어댑터로부터 자신을 찾는 메시지에 응답과 함께 REPLY 상태로 변이한다. 그리고 자신의 참조자를 요청한 어댑터에 알려주고, LISTEN 상

태로 이동하여 어댑터로부터 정보를 받기를 기다린다. 어댑터로부터 REGISTER 메시지와 함께 필요한 정보를 받은 위치 관리자는 DETERMINE 상태로 이동한다. 이 상태에서 어댑터로부터 받은 정보에 근거하여 위치 관리자는 어댑터의 통신방법, 어댑터의 식별자와 현재 어댑터에 요구되어지는 최대 report rate를 결정한다. 이러한 결정 사항들을 REGISTER\_ACK 메시지와 함께 어댑터에 전송하고 REGISTERED 상태로 이동한다. 위치 관리자는 어댑터로부터 첫 번째 위치를 보고 받으면, 어댑터의 등록이 완수되었음을 알고, 다시 INIT 상태로 이동하여 다른 어댑터의 등록을 기다린다.

그림 5는 어댑터와 자동설정자 간의 상호작용 다이어그램이고, 표1은 이러한 두 컴포넌트 간의 상호작용에서 교환되는 메시지에 관한 설명이다.

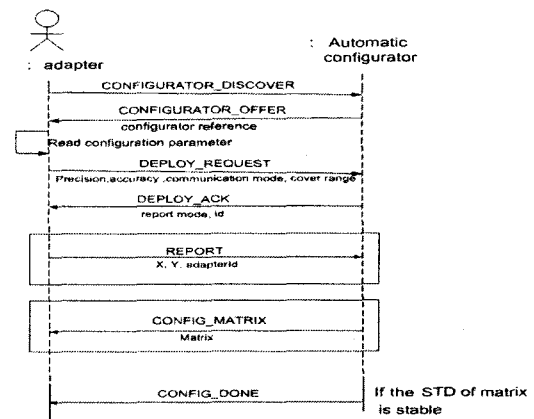
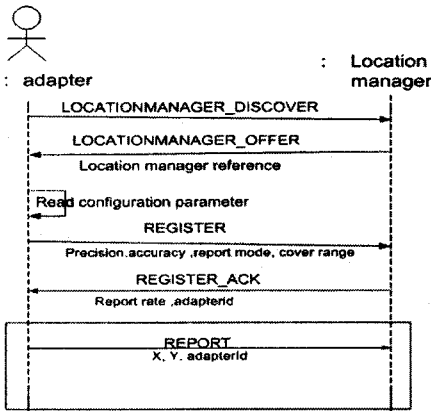


그림 5. 어댑터와 자동설정자간의 상호작용 다이어그램

표 1 어댑터와 자동설정자간의 교환 메시지

Message type	Function	Option	From	to
CONFIGURATOR_DISCOVER	Find the reference of a configurator	None	adapter	configurator
CONFIGURATOR_OFFER	Offer the reference of a configurator	The reference of a configurator	configurator	adapter
DEPLOY_REQUEST	Request its existence to the configurator	Sensor information	adapter	configurator
DEPLOY_ACK	Respond to the DEPLOY_REQUEST	AdapterId, Report mode	configurator	adapter
CONFIG_MATRIX	Config a transformation matrix to the adapter	The elements of the transformation matrix	configurator	adapter
CONFIG_DONE	Notify the end of configuration to the configurator	None	adapter	configurator
REPORT	Report adapter's location	X,Y coordinates and adapter Id	adapter	configurator

그림 6은 어댑터와 위치관리자 간의 상호작용 다이어그램이고, 표 2는 이러한 두 컴포넌트 간의 상호작용에서



교환되는 메시지에 관한 설명이다.

그림 6 어댑터와 위치관리자간의 상호작용 다이어그램

표 2. 어댑터와 위치관리자간의 교환 메시지

Message type	Function	Option	From	to
LOCATIONMANAGER_DISCOVER	Request finding a reference of a location manager	None	adapter	location manager
LOCATIONMANAGER_OFFER	Offer the reference of a location manager	The reference of a location manager	location manager	adapter
REGISTER	Request adapter's registration to the location manager	Sensor information	adapter	location manager
REGISTER_ACK	Respond to the REGISTER	adapter id, report mode and report rate	location manager	adapter
REPORT	Report adapter's location	X, Y coordinates and adapter id	adapter	location manager

#### 4.4 오차 극복 기법

위치 센서들 간에는 정밀도(precision)와 정확도(accuracy)의 차이[10]가 있다 정밀도는 위치 센서가 어느 정도까지의 단위를 표현할 수 있는가 하는 것이다. 정확도는 그 위치 센서가 보고하는 위치 값을 얼마나 신뢰 할 수 있는지 확률로 나타내는 것이다. 새로운 위치 시스템과 기존의 위치 시스템간의 정밀도 및 정확성의 차이는 정확한 변환행렬 값을 구하는 것을 어렵게 만든다.

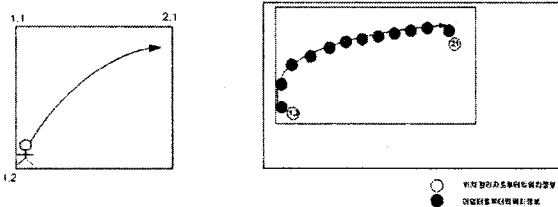


그림 7. 정밀도의 차이로 인한 binding의 어려움  
정확한 값을 얻기 위해서 자동설정자에서는 몇 가지 기법을 사용한다.

첫째, 그림 7에서처럼 자동설정자의 지시대로 사용자가 두 태그를 들고 움직일 때, 센서 간 정밀도의 차이로 인해 정밀도가 낮은 센서의 위치 정보는 동일함에도 정밀도가 높은 센서의 위치 정보는 바뀌게 된다. 이럴 때 자동설정자의 입장에서는 어느 두 위치정보를 연관(bind)시켜야 하는지 판단하기 어렵게 된다. 이러한 점을 극복하고자, 자동설정자는 정밀도가 낮은 위치정보를 기준으로 그 값이 바뀌자마자 가장 근접한 시간에 정밀도가 높은 센서로부터 들어온 위치정보를 연관 시키는 방법을 사용한다. 이것은 자동설정자의 지시 하에 사용자가 직선 방향으로 움직인다는 것을 전제로 한다.

둘째, 센서의 정확성 차이로 인해 생기는 오차를 줄이기 위해, 위치정보들을 가지고 multiple regression 방식을 사용하여 변환행렬을 구한다. 이 방식은 통계적 유추 방법 중의 하나이기 때문에 연관된 위치정보들이 많으면 많을수록 보다 정확한 값을 이끌어 낼 수 있다. 따라서 자동설정자가 어댑터에 변환행렬 값을 한번만 설정해주는 것이 아니라, 계속적인 갱신(update)을 해주어야 한다. 기존 값과 갱신 값 간의 차이가 안정화 될 때까지 이 작업은 계속 된다.

### 5. 구현 및 테스트

#### 5.1 구현환경

Active Surroundings 환경 내 WLAN 센서 시스템[11]을 기반으로 구성된 위치 시스템에 Ubisense 센서 시스템[12]을 우리의 프레임워크의 도움을 받아 새롭게 배치하였다. WLAN 센서 시스템은 학교 전체를 다루지만

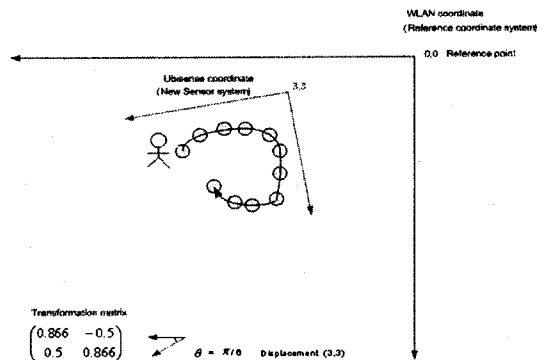


그림 8. 테스트 환경

그 오차는 1m 이상이다. 우리는 Active Surroundings 프로젝트의 연구 및 시현을 위한 데모룸(9m \* 6m)을 가지고 있다. 낮은 정밀도의 WLAN 센서 시스템을 데모룸을 위한 위치시스템으로 이를 사용하기에 불충분 하다. 반면,

Ubisense 센서 시스템의 경우는 오차가 8cm를 넘지 않아 데모를 위한 위치 시스템으로 사용하기 적합하다.

## 5.2 테스트

본 연구에서 제안하는 프레임워크가 성공적으로 작동하는지 확인하기 위해 사용자는 프레임워크에서 제공하는 태그인 HP사의 iPaq RX 3700 PDA와 Ubisense 센서 시스템에서 제공하는 태그를 함께 들고 움직인다. 프레임워크는 PDA 화면을 통해 사용자에게 이동 및 그 방향을 요구한다. 프레임워크가 어댑터의 설정 및 위치 관리자로의 통합을 마치면, 사용자에게 PDA 화면을 통해 그 완료를 알린다. 그 통합이 끝나게 되면 위치 관리자는 Active Surroundings 데모를 제외한 학교 내부에서는 WLAN 센서 시스템을 기반으로, 1m 이내의 오차로 사용자 또는 객체를 추적하지만, Active Surroundings 데모 안에서는 8cm 이내의 오차로 사용자 또는 객체를 추적할 수 있게 되어 가정 내의 위치 인식 응용프로그램(location awareness application program)을 지원 할 수 있게 된다.

## 6. 결론 및 향후계획

본 연구를 통해 우리는 유비쿼터스 환경내에서 새로운 센서 시스템을 배치하는 것의 복잡성을 줄일 수 있게 되었다. 또한 그 배치로 인해 Active Surroundings 프로젝트가 좀 더 많은 위치 인식 응용프로그램을 지원할 수 있게 되었다.

차후 본 연구의 후속으로는 이러한 프레임워크의 도움으로 얻어진 설정값들이 사용자가 수동적으로 설정하는 값보다 얼마나 더 정확한지에 대한 비교 연구 분석이 있어야 할 것이다. 또한, 현재의 구현은 2차원 위치 센서시스템의 경우에 관해서만 생각을 하였기 때문에, 3차원 위치 센서 시스템의 배치 지원에 관해서도 확장 할 예정이다. 앞으로의 유비쿼터스 환경에서 위치센서 시스템은 무선을 기반으로 작동이 되기 때문에, 환경 내에 추가 및 삭제가 동적으로 빈번하게 일어 날 것이다. 그러므로 설치 후, 상황에 대한 고려가 필요하다. 마지막으로, 센서의 배터리를 효율적으로 관리하면서, 정확성 및 신속성을 높이기 위한 위치관리자와 어댑터간의 report rate의 재조정에 대해서도 생각해 볼 수 있다. 이와 같은 통합 후에 어댑터의 관리에 대한 연구가 앞으로 진행될 것이다.

## 7. 참고자료

- [1] D. Saha, A. Mukherjee, Ubiquitous Computing: "A Paradigm for the 21st Century", IEEE Computer, IEEE Computer Society Press, pp. 25-31, March 2003.
- [2] D. Lee, S. Han, Insuk Park, S.K., K. Lee, S.J. Hyun, Y.H. Lee, and G.H. Lee, "A Group-Aware Middleware for Ubiquitous Computing Environments," ICAT 2004, November 2004.
- [3] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, Lachman Technology, Inc., Bucknell University, October 1997.
- [4] Jean-Marc Seigneur, Christian Damsgaard Jensen, Stephen Farrell, Elizabeth Gray, Yong Chen, "Towards Security Auto-Configuration for Smart Appliances."
- [5] UPnP, "Universal Plug and Play", Website, www.upnp.org.
- [6] "OSGi Service Platform Release 2 Specification", The Open Services Gateway Initiative, 2001,
- [7] A.R than, Jalal A.M, S. Chetan, R. Campbell, M. Dennis Mickunas, "A Middleware for Location Awareness in Ubiquitous Computing Applications". In ACM/IFIP/USENIX 5'th International Middleware Conference , Toronto, Ontario, Canada, October 18th - 22nd, 2004.
- [8] IBM J. RES. & DEV, "LORE: An infrastructure to support location-aware" services. VOL. 48 NO. 5/6 SEPTEMBER/NOVEMBER 2004
- [9] C. Jiang and P. Steenkiste, "A hybrid location model with a computable location identifier for ubiquitous computing", presented at Lecture Notes in Computer Science, 2498, UbiComp, 2002
- [10] J. Hightower and G. Borriello, "A Survey and Taxonomy of Location System for Ubiquitous Computing" University of Washington, Computer Scientist and Engineering August 24, 2001
- [11] Ekahau (2002), "Ekahau Positioning Engine 2.0: 802.11-based Wireless LAN positioning system", An Ekahau Technology Document (<http://www.ekahau.com>), Nov 2002.
- [12] UbiSense, "Local position system and sentient computing." <http://www.ubisense.net/>.