

상황인식 처리를 위한 미들웨어 및 컨텍스트 서버를 이용한 응용시스템의 구현

(Implementation of an Application System using Middleware and Context Server for Handling Context-Awareness)

심 춘 보 [†] 태 봉 섭 ^{**} 장 재 우 ^{***} 김 정 기 ^{****} 박 승 민 ^{****}
 (Choon-Bo Shim) (Bong-Sub Tae) (Jae-Woo Chang) (Jeong-Ki Kim) (Seung-Min Park)

요 약 상황인식 처리기술은 사용자의 컨텍스트에 근거하여 사용자와 장치간의 상호 운용성을 지원해 줌으로써, 사용자로 하여금 정보 획득 및 실행을 보다 용이하게 하도록 해주는 기술이다. 이는 유비쿼터스 컴퓨팅 환경에서 가장 중요한 핵심 기술중의 하나이다. 따라서 본 논문에서는 유비쿼터스 컴퓨팅에서 상황인식 처리를 위한 미들웨어 및 컨텍스트 서버를 제안하고, 이를 이용한 응용시스템을 구현한다. 본 연구에서 제안하는 미들웨어는 블루투스 무선 통신 기술을 이용하여 이동성을 지닌 이동 노드를 발견하고 컨텍스트 서버로부터 해당 컨텍스트에 적합한 실행 모듈을 실행하는 기능을 담당한다. 한편, 제안하는 컨텍스트 서버는 사용자의 현재 상태, 물리적 환경, 컴퓨팅 시스템의 리소스 등의 상황 정보를 효율적으로 데이터베이스 서버에 저장하는 관리자의 역할을 수행한다. 마지막으로 본 연구에서 구현된 응용시스템은 상황 정보에 근거하여 음악 재생 서비스를 제공하는 응용 시스템으로써, 아울러 본 연구에서 구축된 미들웨어 및 컨텍스트 서버의 유용성을 보이고 있다.

키워드 : 유비쿼터스 컴퓨팅, 상황인식, 미들웨어, 컨텍스트 서버

Abstract Context-awareness is a technology to facilitate information acquisition and execution by supporting interoperability between users and devices based on users' context. It is one of the most important technologies in ubiquitous computing. In this paper, we propose a middleware and a context server for dealing with context-awareness in ubiquitous computing and implement an application system using them. The middleware proposed in our work plays an important role in recognizing a moving node with mobility by using a Bluetooth wireless communication technology as well as in executing an appropriate execution module according to the context acquired from a context server. In addition, the proposed context server functions as a manager that efficiently stores into a database server context information, such as user's current status, physical environment, and resources of a computing system. Finally, our application system implemented in our work one which provides a music playing service based on context information, and it verifies the usefulness of both the middleware and the context server developed in our work.

Key words : Ubiquitous computing, context-awareness, middleware, context-server

1. 서 론

급속한 기술 발전 추세에 맞춰 Thin-Client, 휴대용 정보 단말기(PDA), 스마트 폰, 인터넷 TV, 차세대 게임기 등과 같이 컴퓨팅 모듈이 장치에 내장된 정보 기기(Post PC)들이 최근 들어 다양한 분야에서 활발히 이용되고 있다. 또한 컴퓨터와 통신 기술(Bluetooth, IEEE 1394, Power Cable Network, IEEE 802.11b 등)은 수많은 다른 분야의 기술들과 접목되어 일상 생활에 활용이 가능한 컴퓨팅 환경을 제공하고 있다. 한편, 제록스 팔로 알토(Palo Alto) 연구소의 마크 와이저(Mark

[†] 정 회 원 : 순천대학교 정보통신공학부 교수
 cbsim@sunchon.ac.kr
^{**} 정 회 원 : NHN(주) 게임엔진팀 연구원
 saebaryo@nhncorp.com
^{***} 종신회원 : 전북대학교 컴퓨터공학과 교수
 jwchang@chonbuk.ac.kr
^{****} 정 회 원 : 한국전자통신연구원 임베디드SW연구단 연구원
 jkk@etri.re.kr
 minpark@etri.re.kr
 논문접수 : 2004년 7월 15일
 심사완료 : 2005년 10월 26일

Weiser)는 '사람을 포함한 현실 공간에 존재하는 모든 대상을 물리적, 공간적으로 연결해 사용자에게 필요한 정보나 서비스를 즉시에 제공할 수 있는 컴퓨팅 환경 즉, 필요한 일부 기능만을 추가하고 부가적으로 필요한 자원들을 주변 환경 내부에 내장시켜 무선 네트워킹 기술을 이용해 상호 연결이 이루어지는 컴퓨팅 환경을 유비쿼터스 컴퓨팅[1]이라고 정의하고 있다. 이러한 유비쿼터스 컴퓨팅 기술은 우리가 일 외의 대부분의 여가 시간을 보내는 홈 환경, 이동성, 자가 발전, 고정된 상호 작용 공간 등의 특성을 고려한 자동차에 기반을 둔 텔레매틱스, 여러 사람과 정보 공유 및 협업이 필요한 사무실 등 우리 일상생활 곳곳에서 활용될 수 있다.

사용자가 필요로 하는 서비스를 제공하기 위해서는, 일상생활 곳곳에 편재된 센서 및 컴퓨터들이 수집한 각종 환경 정보를 효과적으로 상호 공유하여 사용자 및 주변 환경의 컨텍스트를 알아내는 상황인식 처리 기술이 필요하다[2]. 이는 유비쿼터스 컴퓨팅 기술 가운데 기반 기술에 속한다. 상황인식 처리 기술은 사용자를 중심으로 하는 주변 환경과 사용자간 혹은 사용자 와 장치간의 상호 운용성을 지능적, 자동적으로 선택하여 지원해 줌으로써, 사용자로 하여금 정보 획득 및 실행을 보다 용이하도록 지원한다. 이러한 기술은 '일상 환경 속에 편재된 언제 어디서나 이용 가능한 컴퓨팅 환경'인 유비쿼터스 컴퓨팅 환경에서 가장 중요한 핵심 기술 중에 하나이다[3].

본 논문에서는 유비쿼터스 컴퓨팅에서 상황인식 처리를 위한 미들웨어 및 컨텍스트 서버를 제안하고, 이를 이용한 응용시스템을 구현한다. 먼저, 본 논문에서 제안한 상황인식 기반의 미들웨어는 블루투스 무선 통신 기술[4,5]을 이용하여 이동성을 지닌 이동 노드를 발견하고, 컨텍스트 서버로부터 해당 컨텍스트에 적합한 실행 모듈을 실행하는 기능을 담당한다. 아울러 제안하는 컨텍스트 서버는 사용자의 현재 상태, 물리적 환경, 컴퓨팅 시스템의 리소스 등의 상황 정보를 효율적으로 데이터베이스 서버에 저장하는 관리자의 역할을 수행한다. 프로필을 지닌 이동노드가 미들웨어로 접근하면, 미들웨어는 이동노드의 프로필을 이용하여 컨텍스트 서버에 접속한 후, 수행해야 하는 상황 정보를 검색하고, 이를 해당 미들웨어로 전달한다. 또한 상황인식을 위해 요구되는 일련의 복잡한 과정을 간략하고 명료하게 기술하고 상황에 대한 정의를 미리 규격화된 구문으로 표현함과 동시에 자동적으로 처리할 수 있는 컨텍스트 정의 스크립트 언어 및 언어 처리기[6]를 이용한다. 마지막으로 본 연구에서 구현된 응용시스템은 상황정보에 근거하여 음악 재생 서비스를 제공하는 응용 시스템으로써, 본 연구에서 구축된 미들웨어 및 컨텍스트 서버의 유용

성을 보이고 있다.

본 논문의 구성은 다음과 같다. 제2장에서는 유비쿼터스 컴퓨팅 하에서 상황인식 처리와 관련된 기존 연구들을 소개한다. 제3장에서는 본 논문에서 제안하는 상황인식 처리를 위한 미들웨어 및 컨텍스트 서버에 대해서 설명한다. 제4장에서는 제3장에서 제안된 미들웨어 및 컨텍스트 서버를 이용한 응용 시스템의 구현에 대해서 기술한다. 마지막으로 제5장에서는 결론 및 향후연구를 제시한다.

2. 관련 연구

본 논문의 관련 연구는 유비쿼터스 컴퓨팅에서 상황인식 처리 기술과 밀접한 관련이 있는 국내외의 연구를 간략히 정리하여 소개한다.

국외의 연구는 다음과 같다. 첫째, IRISA/INRIA의 연구[7]는 이동성을 가진 사용자에 대해서 사용자의 주변 환경을 이용하여, 그에 대한 서비스의 레벨을 향상시키기 위해 상황 객체(Contextual Object:CO)에 기반한 하부 구조(infrastructure)를 제안하였다. 사용자의 주변 환경 즉, 상황을 감지하여 그에 적합한 최적의 서비스를 제공하기 위해 CO를 정의하고 있다. 전체적인 구조는 크게 서버 측과 클라이언트 측으로 나뉜다. 서버 측에서는 CO를 저장 및 검색할 수 있는 역할을 담당하며, 클라이언트 측에서의 요청이 있을 경우 CO를 서버 측으로 전송한다. 아울러 CO의 Variant들을 저장하고 있다. 클라이언트 측에서 Variant의 요청이 있으면 그에 따른 응답을 한다. 클라이언트 측은 크게 응용 계층(Application Layer), 적응 계층(Adaptive Layer), 탐지/전송 계층(Detection/Notification Layer)으로 나뉘어져 있고, 여기서 적응 계층은 COs 관리자(COs Manager), 상황 관리자(Context Manager), 선택 관리자(Selection Manager)로 나뉜다. 먼저, 응용 계층은 최상위 계층에 존재하며 적응적 시스템에서의 상황에 따른 처리 결과를 브라우징하거나 속성 정보를 상황 관리기에 전달하는 역할을 담당한다. 아울러 적응 계층의 COs 관리기는, 각 응용에 현재 사용되고 있는 CO의 모든 정보를 위한 데이터 구조를 관리하고, 상황 관리기는 탐지/전송 계층에서 현재 환경에 대한 상황의 변화가 감지되었을 때 그에 대한 정보를 넘겨받는 역할을 수행한다. 선택 관리기는 사용자 프로파일내의 취향이나 선호도를 기반으로 현재 상황에 최적의 결과를 얻을 수 있도록 불필요한 정보를 필터링하는 역할을 담당한다. 마지막으로 탐지/전송 계층은 사용자의 관심에 따른 시스템, 네트워크의 상황의 변화, 주변의 환경을 감지하여 하위 레벨 정보를 상위 레벨의 이벤트로 변환하는 역할 및 그 이벤트를 통지하는 기능을 수행한다. 둘째, AT&T 연구

[8]는 실내(indoor)에서 사용자의 위치를 탐지 및 추적하여 사용자의 주변 환경에 맞는 상황인식 응용 서비스를 제공하고 있다. 실내에서 사용자의 위치를 센싱하기 위해 각 사용자는 배지(Bat unit)를 착용한 상태로 이동하며, 수신자(receiver)는 각 사용자의 배지를 감지하여 사용자 위치를 파악한다. 이때 오직 하나의 수신자만 배지를 감지하는 것이 아니라, 다수개의 수신자가 배지를 탐지하고 각 수신자와 배지 사이의 전파의 세기를 계산하여 사용자의 위치를 파악한다. 아울러, AT&T의 연구는 three-tier 아키텍처에 기반을 두고 배지의 센싱 정보, 기기의 작동상태, 네트워크의 대역폭이나 전달 지연 시간 등의 자원 모니터링 정보가 모바일 응용에 적용된다. 여기서 각 객체의 위치 정보가 공간 인덱싱 프락시(Spatial Indexing Proxy)에서 관리되며, 위자 프락시 서버(Ouija Proxy Server)에서는 객체지향 모델링 언어를 이용하여 다수개의 객체들을 오라클 관계형 데이터베이스의 튜플들로 변환하는 작업을 수행하며, 시스템의 오류에 대비하여 두 개의 서버가 동작하도록 설계하였다. 또한 생성되는 모든 데이터는 오라클7 관계형 데이터베이스에 저장되어 관리된다. 셋째, Lancaster 대학의 GUIDE 프로젝트[9]는 영국의 Lancaster 도시를 방문하는 여행객의 편의와 효율적인 여행 서비스를 제공하기 위해 수행되었다. 즉, 도시를 방문한 여행객에게 휴대용 PC를 제공하고, 여행객의 위치 정보를 탐지 및 추적하여 여행객에게 정보를 제공하는 위치기반 서비스 시스템을 제안하였다. 휴대용 장치는 윈도우 CE가 탑재된 펜 기반 타블렛(pen-based tablet) PC로서, 사용자가 위치한 주변의 흥미있는 관광 명소를 HTML 브라우저를 통해 브라우징 하도록 설계하였다. 사용자의 위치 탐지는 도시의 각 지역마다 많은 셀(cell)들을 설치하고 또한 다수 개의 셀들을 관리하는 셀 서버(cell server)를 두어 각각의 셀들 간의 통신을 통해 가능하게 한다. 도시의 각 지역에 설치되어 있는 셀에는 무선 랜의 Access Point가 있고, AP는 휴대용 단말기와 주기적으로 통신을 함으로써 여행객의 위치를 파악한다. 또한 여행객의 위치에 따라 GUIDE 웹 서버에서는 각 위치에 관련된 HTML 문서를 전송한다. 휴대용 장치에는 클라이언트 에이전트가 상주되어, 사용자의 프로파일이나 위치에 따라 적절히 웹 서버에 데이터를 요구하며, 전송받은 데이터는 주어진 사용자의 상황에 맞게 필터링한다.

국내의 연구는 다음과 같다. 첫째, 광주과학기술원의 연구[10]에서는 유비쿼터스 컴퓨팅 환경에서 상황인식 정보를 기반으로 개인화된 서비스를 구축하기 위한 컨텍스트 인식 프레임워크인 ubi-UCAM 2.0을 제안하였다. 유비센서와 유비서비스로 구성된 ubi-UCAM 2.0은 개체화된 컨텍스트를 이용하여 5W1H(Who, What,

Where, When, How, Why)의 각 항목별 세부요소를 지정함으로써 여러 서비스에서 동시에 사용할 수 있는 컨텍스트 구조를 제공한다. 또한 멀티 캐스팅을 지원하는 ad-hoc 네트워크 모드를 통해 유비센서와 유비서비스의 동적 연결 및 분산 관리 기능을 제공한다. 유비센서는 사용자 및 사용자 환경에 대한 변화를 감지하는 센서모듈과 감지된 정보를 컨텍스트로 생성하는 초기 컨텍스트 결정(preliminary context decision) 모듈로 구성되어 있다. 또한 유비서비스는 컨텍스트 통합기, 컨텍스트 관리기, 해석기(interpreter), 그리고 서비스 관리기(service provider) 등으로 구성되어 있다. 둘째, 순천향대학교 연구[11]에서는 블루투스를 사용하여 유비쿼터스 환경하의 이동 단말에 상황인식 서비스를 제공하는 방식을 제안하였다. 이동 단말이 서비스를 받기 위해 블루투스 AP를 탐색하고 연결설정을 하는 초기 단계에서부터 위치나 서비스 가용 유무 등의 상황인식 정보를 내장하여 전송한다. 따라서 이동 단말이 서비스 유무나 위치정보를 즉시 획득하여 반응할 수 있고, 블루투스 무선 연결 접속 후에 상황인식 정보를 획득하기 위한 부가적인 메시지 교환을 제거하였다. 아울러 제안된 방식의 검증을 위해 홈 네트워크 환경 하에서 정보사전기기를 제어하는 상황을 설정하고, 블루투스 스택에 내장된 서비스 탐색 프로토콜(SDP) 메시지에 위치 및 기기들의 정보들을 XML로 표현하여 상황인식 서비스를 제공한다. 셋째, 부산대학교의 연구[12]에서는 유비쿼터스 컴퓨팅 환경에서 사용자에 대한 상황 변화에 적응적인 서비스를 제공하기 위해 사용자에 대한 행동, 감정상태, 위치와 같은 상황 정보를 인식하고 검색을 통해 사용자의 위치정보, 활동형태 등의 컨텍스트를 질의할 수 있는 JINI 미들웨어 기반의 상황인식 채팅 프로그램을 구현하였다. JINI 미들웨어는 서비스 기반의 발견(discovery) 및 진입(join) 프로토콜이 내장된 부스트래핑(bootstrapping) 메카니즘으로 수행되는 분산 네트워크 하부구조이며, API 형태로 지원한다. JINI 미들웨어 구조는 가장 자바다운 특징을 이용하고 있으며, 네트워크 플러그 앤 워크(network plug and work) 기능은 분산 응용에 있어 간단하면서도 유연한 네트워크 환경을 제공해 준다. 컨텍스트 정보로는 위치정보(location), 감정 및 분위기 정보(mood), 활동 정보를 이용하여 이를 XML로 표현한다. 마지막으로 상황인식 처리를 위한 국내외의 관련 연구들을 요약 정리하면 다음 표 1과 같다.

3. 상황인식 처리를 위한 미들웨어 및 컨텍스트 서버

본 장에서는 유비쿼터스 컴퓨팅 환경에서 상황인식 처리를 위해 제안된 미들웨어 및 컨텍스트 서버의 전체

표 1 상황인식 처리 기술에 대한 관련 연구 비교

	국외			국내			
	IRISA/INRIA	AT&T	GUIDE	광주과기원	순천향대	부산대	본 연구
상황인식 정보	일반적인 상황	사용자 상황 (신원, 위치, 활동)	사용자 상황 (신원, 위치)	사용자 상황, 물리적 환경 상황, 시공간 상황	사용자 상황(신원, 위치)	위치, 감정, 활동	사용자 상황, 물리적 환경 상황, 시공간 상황
센싱	실외(WLAN)	실내(Active Badge)	실외(GPS)	실내 (유비센서)	실내 (블루투스)	실내	실내 (블루투스)
미들웨어	자체 개발	.	.	자체 개발	.	JINI	자체 개발
컨텍스트 서버	자체 개발	.	자체 개발	자체 개발	자체 개발	.	자체 개발
상황인식 정보 표현	상황객체 정의	없음	HTML	5WIH	XML	XML	상황 정의 스크립트 언어
응용 서비스	적용적 웹 서비스	위치 추적, Shopping Assistant	Tour Guide	c-MP, TMCS, cPost-it	정보가전기 제어	채팅	음악 재생 (Sound, Voice)
특징	<ul style="list-style-type: none"> - 상황 객체(CO) 정의 개념 제안 - 상황인식 기반 적용적하부구조 제안 	<ul style="list-style-type: none"> - 최초의 Indoor 기반 사용자 상황인식 서비스 - 센싱을 위해 Active Badge 사용 	<ul style="list-style-type: none"> - 윈도우 CE를 탑재한 펜 기반 타블렛 PC를 이동 단말로 사용 	<ul style="list-style-type: none"> - 컨텍스트 인식 프레임워크 제안 - 멀티캐스팅을 지원하는 ad-hoc 네트워킹 모듈을 제공 	<ul style="list-style-type: none"> - 홈 네트워크 환경 형에서 정보가전기 제어 서비스 제안 	<ul style="list-style-type: none"> - 서비스 기반의 분산 네트워크인 JINI를 이용한 상황인식 채팅 프로그램 제안 	<ul style="list-style-type: none"> - 상황인식을 위한 미들웨어 및 컨텍스트 서버 개발 - 상황 정의를 위한 상황정의 스크립트 언어 제안

적인 구조에 대해서 기술한다. 그에 앞서 본 논문에서 언급하는 컨텍스트의 의미는 기존의 Dey[13], Chen[14], Schilit[15] 등의 상황인식 컴퓨팅에 관한 연구에서 다양한 모습으로 정의가 제시되었으나, 적용하는 응용 분야에 따라 달라 명확한 정의가 쉽지 않다. 그러나 본 논문에서는 최초로 상황인식 정의를 내린 Schilit가 정의한 "관심이 있는 실체의 '위치'와 근접한 사람 및 사물의 확인 그리고 이러한 실체에 대한 변화"로써 상황을 의미하며, 또한 응용에 따라 상황 정보가 다르지만, 현재로는 사용자 상황, 신원 상황(ID, 이름), 물리적 환경 상황, 공간 상황(위치), 시간 상황(시각), 컴퓨팅 시스템 상황을 고려한다. 제안하는 시스템은 크게 3개의 컴포넌트로 구성되며, 이는 컨텍스트 서버(context server), 고정노드(fixed node), 이동노드(moving node)이다. 전체적인 구조는 그림 1과 같다. 먼저, 컨텍스트 서버는 원격 객체 정보와 컨텍스트 정보를 각각 객체 데이터베이스 및 컨텍스트 데이터베이스에 저장하고 검색하는 역할을 담당한다. 둘째, 고정노드는 미들웨어 역할을 담당하며, 블루투스 장치를 사용하는 환경에서 상황인식 처리를 위한 원격 객체 발견, 등록, 실행을 수행하는 모듈과 상황정의 스크립트 및 처리기로 구성되어 있다. 마지막으로, 이동노드는 원격 객체로서 고정노드와 블루투스 통신을 통하여 데이터를 교환하며, 미리 정의된 상황에 따

라 미들웨어와 상호작용을 통해 내장된 프로그램을 실행하거나 지정된 처리를 수행한다. 한편 서버와 고정노드 사이는 TCP/IP를 통한 유선 네트워크로 구성되어 있고, 고정노드와 이동노드 사이에는 무선 통신인 블루투스를 이용하여 데이터를 교환한다. 제안하는 시스템의 구조는 기존의 관련 연구인 INISA/INRIA의 연구 및 AT&T의 연구의 장점을 결합한 구조이다. 즉, IRISA/INRIA의 연구에서는 다양한 응용에 맞는 상황 정보를 정의하기 위해 상황 객체라는 Contextual Object(CO)를 정의하고 있으며 또한 정의한 상황 객체를 통해 클라이언트에서는 그에 해당하는 적용적 응용 서비스를 사용자에게 제공하며 서버측에서는 상황 객체들과 그에 해당하는 다양한 값들을 저장 및 관리하는 상황인식 기반 적용적 하부 구조를 제안하였다. AT&T 연구에서는 사용자가 착용한 배지와 배지를 감지하는 리시버를 통해 사용자의 위치 탐지 및 추적을 가능하게 하며, 리시버를 이용해 감지한 사용자의 위치 및 사용자 상황 정보를 공간 인덱싱 프락시 서버에서 데이터베이스의 튜플들로 관리하고 있다. 따라서 본 연구에서는 이러한 기존의 두 개의 관련 연구들의 장점을 결합하여, 상황인식에 기반한 다양한 응용 서비스를 위한 컨텍스트 정의 방법 및 상황인식 컴퓨팅 시스템의 전체 구조를 제안한다. 그림 1은 본 논문에서 제안한 전체 시스템 구조를

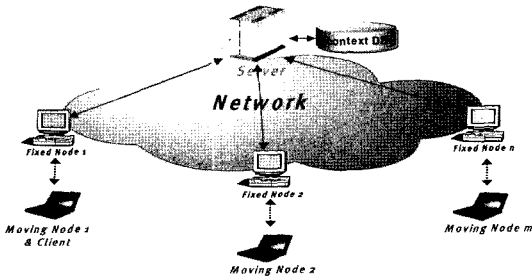


그림 1 상황인식 처리를 위한 전체 시스템 구조

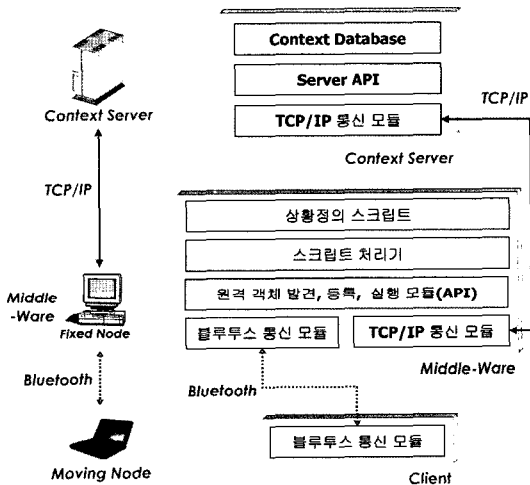


그림 2 세부컴포넌트 및 컴포넌트간 통신을 위한 모듈

나타내며, 그림 2는 전체 시스템의 세부 컴포넌트 및 컴포넌트간 통신을 위한 모듈을 나타낸다.

3.1 미들웨어의 설계

제안하는 상황인식 처리를 위한 미들웨어는 전체적으로 그림 3에서와 같이 3 단계의 개념적인 계층으로 구성되며, 각각은 탐지 및 모니터링 계층, 상황인식 계층, 응용계층으로 구성된다. 첫째, 탐지 및 모니터링 계층은 원격 객체의 위치정보나 응용 프로그램과 상호작용에 따른 이벤트, 그리고 CPU 사용량, 메모리의 사용량, 가용한 대역폭, 블루투스 장치를 포함한 장치에 관련된 이벤트 정보 등과 같은 컴퓨팅 자원이나 네트워크 상태를 모니터링 하는 기능을 담당한다. 둘째, 응용 계층은 하부의 상황인식 계층인 미들웨어와 독립적으로 동작하면서 미들웨어의 응용 프로그래밍 인터페이스(API)를 통해 상황인식 처리 기술에 기반을 둔 다양한 응용 프로그램을 개발하기 위한 기능을 수행한다. 마지막으로, 상황인식 계층은 유비쿼터스 컴퓨팅을 위한 상황인식 처리를 위한 핵심 계층으로서 미들웨어 역할을 담당한다. 이 계층은 크게 스크립트 처리기, 원격 객체 관리기, 컨

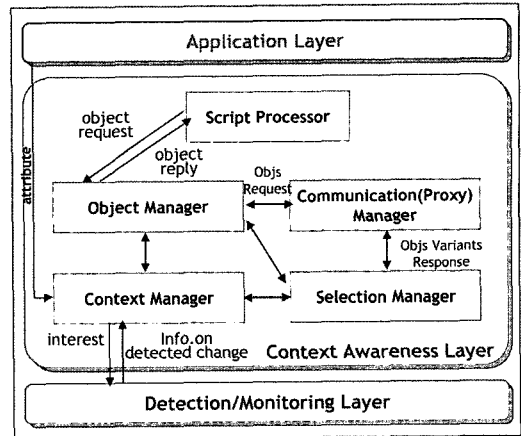


그림 3 미들웨어(고정노드)의 구조

텍스트 관리기, 컨텍스트 선택 관리기, 통신 프락시 관리기의 5 가지의 관리자로 구성된다(그림 3). 각각의 역할을 간단히 기술하면 다음과 같다.

- **스크립트 처리기(Script Processor)** : 응용 프로그램의 상황인식 정의 스크립트의 각 내용을 분석하여 규약에 명시된 역할을 수행하도록 한다.
- **원격 객체 관리기(Object Manager)** : 각 응용 프로그램에서 현재 사용하고 있는 원격 객체(컨텍스트 객체)에 포함되어 있는 모든 정보를 위한 데이터 구조를 관리한다.
- **컨텍스트 관리기(Context Manager)** : 사용자의 취향, 이동 단말 기기의 성능, 현재 위치정보 등을 포함하고 있는 주어진 환경이나 상황에 대한 컨텍스트 정보를 관리한다.
- **컨텍스트 선택 관리기(Context Selection Manager)** : 응용 프로그램을 통해 사용자에게 전송될 가장 유사한 조건에 부합되는 컨텍스트 정보를 선별한다.
- **통신(프락시) 관리기(Communication(Proxy) Manager)** : TCP/IP를 통하여 컨텍스트 서버와 통신을 담당하며, 예상하지 못한 네트워크의 장애로 인한 문제에 대비하여, 즉 재전송을 위해 일시적으로 내용을 보관하는 기능을 수행한다.

예를 들어, 제안하는 미들웨어(고정노드)와 컨텍스트 서버를 통해, 원격 객체 즉, 이동성을 지닌 이동노드를 블루투스 무선 통신을 이용하여 감지하고, 감지된 이동노드에 대한 정보 및 감지된 상황에 적절한 서비스를 실행하는 과정은 그림 4와 같다. 이는 3단계 9과정을 포함한다.

• 단계 1 : 원격 객체의 발견

원격 객체의 발견은 그림 4에서와 같이 이동노드는 블루투스 무선 통신을 위한 고정 노드와의 연결(con-

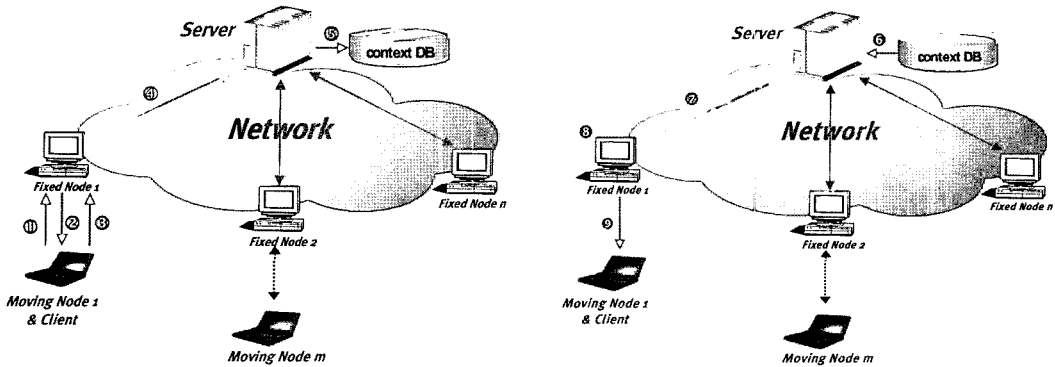


그림 4 원격 객체 발견, 등록 및 서비스 실행

nection)을 위해 접속 신호를 브로드캐스팅(broadcasting) 한다(①). 해당 영역을 담당하고 있는 고정노드는 그 신호를 파악하고 접속을 수락(accept)한다(②). 서로간의 접속이 맺어지면 이동노드는 고정노드에게 자신의 정보를 전달한다(③).

• 단계 2 : 원격 객체의 등록

원격 객체의 등록은 그림 4에서와 같이 원격 객체가 발견되어 원격 객체와 접속이 이루어지면, 원격 객체의 정보를 유선 네트워크인 TCP/IP를 통해 서버로 전달하고(④), 전달받은 정보를 컨텍스트 데이터베이스에 저장한다(⑤).

• 단계 3 : 원격 객체의 서비스 실행

원격 객체의 실행은 응용에 따라서 처리과정이 달라진다. 원격 객체의 실행은 원격 객체의 발견과 등록이 완료된 상태이면, 이동객체에 이동에 따른 변화되는 상황에 따라 컨텍스트 데이터베이스의 컨텍스트 정보를 서버를 통해 고정노드로 전송한다(⑥, ⑦). 고정노드는 컨텍스트 데이터를 바탕으로 상황에 적합한 미리 정해진 처리과정을 실행하며(⑧), 이동노드가 실행할 서비스를 통지한다(⑨). 이동노드는 고정노드로부터 요청받은 서비스를 실행하여 전체적인 상황인식 처리를 마무리한다.

이동노드의 수행 흐름은 메인 모듈에서 클라이언트의 시작 및 블루투스의 통신 소켓 준비를 통해 시작되며 환경 설정 모듈에서 각 고정노드 주소 리스트를 로딩해서 메모리의 고정리스트 주소 변수에 각각 할당한다. 그 다음 블루투스 소켓 모듈에 의해서 앞에서 언급한 순차적 탐색 접속 방법과 동시 접속 방법 중 한 가지를 선택하여 전파가 닿는 고정노드에 접속하게 된다. 접속이 완료되면 블루투스 소켓 모듈에 의해서 고정노드와 주기적인 통신을 통하여 고정노드와 접속 상태에 이상이 있는지 감지하여 이상이 발생한다면 전파가 닿는 다른 고정노드로 접속 시도를 하게 된다. 전체적인 이동노드

의 수행 흐름은 그림 3-10과 같다. 미들웨어의 전체적인 시작은 메인 모듈부터 시작하고 그 밖의 모듈을 호출하여 전체적인 상황을 파악하게 된다. 즉, 블루투스 소켓 모듈을 통하여 이동노드의 접속에 대해서 대기하고, 접속이 이루어지면 접속된 이동노드 클라이언트가 사용자 이름을 전송하여 블루투스 소켓 모듈을 통해서 전송받게 되고, 이 사용자 이름에 대해서 컨텍스트 API와 TCP/IP 소켓 모듈을 통하여 질의 처리에 의해서 인증 절차를 거치게 됨에 따라 컨텍스트 서버에 의해서 등록된다. 이후는 고정노드와 이동노드간의 주기적인 통신에 의해 고정노드가 이동노드의 접속 해제나 통신 장애와 같은 상황을 파악하게 되며 응용 시스템에서 적절한 함수를 호출한다. 이러한 흐름은 그림 3-7과 같다. 고정노드의 미들웨어가 실행되어 이동노드의 블루투스 접속에 대해서 수락할 준비가 되어 있는 상태가 되면 이동노드는 각 고정노드 리스트를 가지고 각 고정노드에 대해서 접속 시도를 한다. 고정노드의 접속 신호에 대해서 전파가 닿아 그 신호를 수락할 수 있는 고정노드가 있으면, 고정노드는 접속 신호를 수락하여 양쪽 노드가 연결되고 컨텍스트 서버 측으로 이동노드를 등록한다. 이후에 양쪽 노드 간 주기적인 통신을 통하여 접속 상태에 대해서 확인을 하며 상황에 따라 적절한 조치를 취하게 된다.

3.2 컨텍스트 서버

상황인식 처리를 위해 원격 객체에 대한 정보 및 원격 객체로부터 파생되고 추출된 상황정보를 서버에 저장하고 검색 할 수 있는 컨텍스트 서버가 요구된다. 이를 위해 본 논문에서는 기존 MySQL DBMS에 기반하여 원격 객체에 대한 정보 및 상황정보를 효율적으로 저장 및 관리할 수 있는 컨텍스트 서버를 설계한다. 한편 설계된 컨텍스트 서버는 이미 규정된 약속을 가지고 미들웨어에서 전달되는 패킷의 내용을 분석하고 패킷의 내용을 수행한다. 즉, 패킷의 내용을 분석한 후 그 패킷

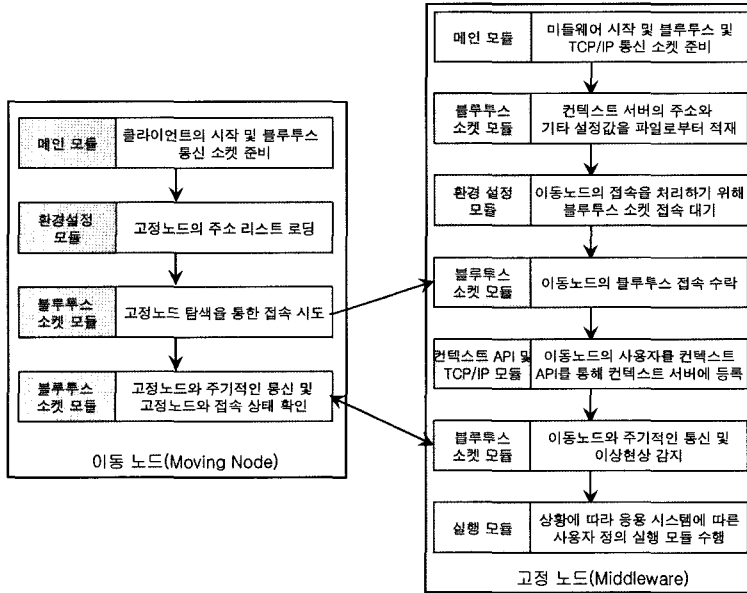


그림 5 시스템의 전체적인 수행 흐름

내용이 Context DB에 저장해야 할 컨텍스트인지, 아니면 컨텍스트 객체인지 구별한 후에 이에 해당하는 데이터베이스에 저장 및 검색한다. 그림 5는 컨텍스트 서버의 구조를 도식화한 것이며, 전체적으로 4개의 관리기로 구성된다. 각각의 관리기에 대한 기능은 다음과 같다.

- **통신 관리기(Communication Manager:CM)** : CM은 서버와 미들웨어간의 통신을 담당한다. 미들웨어에서 전달되어진 패킷의 내용을 분석하기 위해 해당 패킷을 PAM에 전달하고, 서버에서 작성된 결과 패킷을 다시 미들웨어에 전송한다.
- **패킷 분석 관리기(Packet Analysis Manager:PAM)** : PAM은 CM으로부터 전달된 패킷을 분석하여, 현재 어떤 행동을 수행하고자 하는지 분석한다. 이 분석을 통해 COM에 있는 적합한 해당 함수를 호출한다.
- **컨텍스트/객체 관리기(Context/Object Manager:COM)** : COM은 PAM으로 부터 전달된 내용을 SQL문으로 변환하며, 이 SQL문을 SQM에 전달해 실행한다. 이를 위한 인터페이스(API)는 ContextDefine, Context-Destroy, ContextInsertTuple, ContextDeleteTuple, ContextSearch, ContextSearchTuples, Context-Custom이 있다.
- **데이터베이스 질의 관리기(mysql Query Manager:SQM)** : SQM은 COM으로부터 받은 SQL문을 DBMS에 반영하기 위한 작업을 수행하며, 그 결과 값을 CM을 통해 다시 미들웨어에 전송한다. 이를 위한 인터페이스는 mysql_Reader와 mysql_NonQuery 등이 있다.

표 2는 구현된 컨텍스트 서버의 세부 구성 모듈을 정리한 것이다. 메인 모듈은 각각의 서브 모듈인 패킷 분석 모듈, 컨텍스트 API 모듈, mySQL 모듈, 파일 수신 모듈 그리고 TCP/IP 소켓 모듈을 이용한다. 또한 파일 수신 모듈은 미들웨어와 TCP/IP 소켓 통신을 통해 메시지를 주고 받기 때문에 TCP/IP 소켓 모듈에 의존한다. 메인 서버가 TCP/IP 소켓을 통해 준비 중에 있으면, 미들웨어로부터 패킷을 받게 된다. 받은 패킷을 패킷 분석 모듈이 분석하여 어떠한 컨텍스트 API를 호출해야 할 지를 판단한다. 호출 컨텍스트 API는 패킷을 SQL문으로 변환하고 mySQL API에게 전달한다. mySQL API는 mysql사에서 제공하는 mysql C 라이브러리를 이용해서 구현되며, 생성된 SQL문은 mySQL 실행 API에 의해서 실행된다. 즉, 생성된 SQL문에 따라 상황에 맞는 mySQL 실행 API가 실행된다. 만약 생성된 SQL문이 결과를 받아오는 select 문인 경우 mysql_Reader의 함수를 호출하고, 다른 경우에는 mysql_NonQuery의 함수를 사용한다. 아울러 결과 값은 직접 미들웨어에 전송된다.

4. 응용시스템 구현 및 성능 고찰

유비쿼터스 컴퓨팅 환경에서 상황인식 처리를 위해 설계된 미들웨어와 컨텍스트 서버는 Pentium-IV 1.7GHz CPU 와 메인메모리 512MB, Redhat Linux 7.3(kernel version 2.3.20), GCC 컴파일러 2.95.4를 이용하여 구축하였다. 또한 컨텍스트 서버의 구축을 위해

표 2 컨텍스트 서버의 세부 구성 모듈

구성 요소	기능
메인 서버 모듈	컨텍스트 서버의 시작 및 TCP/IP 소켓을 통해 통신하기 위한 통신 관리자(CM) 모듈
패킷 분석 모듈	PAM 부분에 속하는 모듈로써, 미들웨어로 부터 받은 패킷들을 분석한 후, 해당 컨텍스트 API를 호출
컨텍스트 API	DBMS 및 미들웨어와의 상호작용을 위한 라이브러리 - 서버측 API
mysql API	데이터베이스 질의 관리자(SQM)에 속하는 모듈로써, mysql DBMS와의 통신 및 해당 질의 결과문을 미들웨어로 전송하는 기능을 담당
파일 수신 모듈	컨텍스트 서버가 미들웨어로 파일을 전송하기 위한 라이브러리
TCP/IP 소켓 모듈	컨텍스트 서버와 TCP/IP 소켓 통신을 위한 소켓 라이브러리
기타 실행 모듈	원격 객체의 상황에 대한 사용자 정의 실행 모듈

사용된 기존 관계형 데이터베이스는 앞에서 언급한 것처럼 MySQL를 이용한다. 그 이유는 처음부터 저장 시스템을 개발하는 데 소요되는 시간과 어려움을 줄이고, 시스템의 안정성 및 효율성을 증대시킬 수 있기 때문이다. 한편, 상황인식 처리를 위해 구축된 미들웨어 및 컨텍스트 서버를 이용하여 응용 시스템을 구현한다. 응용시스템은 주어진 상황에 적합한 음악 서비스를 제공하는 것으로, 응용 시스템에서 사용자는 하나의 이동노드를 소유하고 있으며, 사용자의 위치에 따라 고정노드는 사용자의 접근을 감지해 사용자에 대해 미리 정의해 둔 음악을 연주하는 역할을 수행한다. 이 때, 음악은 사용자별로 서로 다르게 정의되어 있으며 같은 사용자에 대해서도 오전, 오후, 야간과 같은 시간대에 따라 다른 곡이 정의되어 있다. 따라서 응용 시스템은 각 사용자에 따라 정의된 음악을 연주함에 있어서 그림 6과 같이 사용자 1이 장소 A에서 장소 B로 이동한다고 가정할 때, 장소 A에서 사용자 1을 위한 음악이 연주되고 있다면, 사용자 1이 장소 B로 옮겨감에 따라 장소 A의 음악이 정지되고 장소 B에서 사용자 1을 위한 음악이 연주된다. 한편, 고정노드는 사용자를 구분하여 사용자에 따른 미리 정의된 음악을 연주하며, 사용자가 고정노드 영역에 들어온 시각을 파악하여 시간대에 따라 정의된 음악을 연주한다.

한편 음악 연주 응용 시스템을 위한 컨텍스트 서버의 사용자 데이터베이스 스키마와 튜플 예들은 표 3과 같다. 각 필드는 User_ID, User_Name, Location, Music_M, Music_A, Music_E로 구성된다. 여기서, User_ID는 각 사용자의 식별자로서 각 사용자를 구분하기 위한 주 키(primary key) 역할을 담당하며, User_Name은 각 사용자 이름을 나타낸다. Location은 각 사용자의 위치로서 미리 지정된 값이 아니며, 고정노드가 이동노드의 위

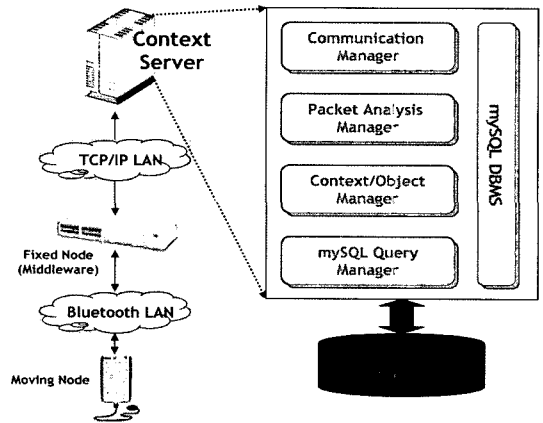


그림 6 컨텍스트 서버의 구성도

치를 파악했을 때 사용자의 위치에 따라 빈번히 갱신되는 값이다. Music_M, Music_A, Music_E는 각각 사용자가 선호하는 음악을 위한 파일명으로서 오전(morning), 오후(afternoon), 저녁(evening)에 따른 음악 파일명에 대응된다. 컨텍스트 서버는 이와 같은 데이터베이스 스키마에 따라 사용자에 대한 음악 파일들을 유지하고 있으며, 미들웨어의 질의를 처리하고, 미들웨어의 파일 요청에 따라 파일들을 미들웨어로 전송한다. 즉, 이동노드를 소유한 사용자가 고정노드에 접근했을 때 이동노드는 주위의 가까운 고정노드를 탐색하고, 고정노드에 접속한 후 고정노드는 컨텍스트 서버에 등록된 사용자인지 질의를 수행한다. 고정노드는 사용자 인증절차를 거친 후 사용자에 대해 미리 정의된 음악을 컨텍스트 서버로부터 요청하여, 다운로드 받아 음악 파일을 수행한다. 또한, 컨텍스트 서버를 제외한 고정노드와 이동노드는 블루투스 장치를 이용하여 무선 통신을 수행한다.

표 3 컨텍스트 서버의 데이터베이스 스키마와 튜플의 예

User_ID	User_Name	Location	Music_M	Music_A	Music_E
1	'김철수'	NULL	filename11	filename12	filename13
2	'이영희'	NULL	filename21	filename22	filename23

블루투스 장치는 블루투스 Version1.1/Class1의 스펙 (specification)을 만족하며 USB 인터페이스를 통하여 PC에 접속된다. 전체적인 응용 시스템의 환경은 시스템 클럭 866Mhz 펜티엄 3 시스템에 64MB의 메모리를 탑재하였다. 운영체제는 Redhat Linux 7.3(kernel verion 2.4.20)이고, 블루투스 통신을 위한 장치 드라이버 및 프로토콜 스택은 affix 2.0.2, 컴파일러로서 GCC 2.95.4를 사용하였다. 아울러 전체 응용 시스템을 모듈식으로 구성하여, 향후 필요한 통신 모듈 및 DBMS의 모듈을 변경 및 교체함으로써, 응용시스템의 특성에 맞는 범용적인 시스템을 구축하기에 용이하도록 하였다.

4.1 응용 시스템의 검증

본 논문에서는 구현된 응용 시스템이 정상적으로 동작하는지 검증한다. 이러한 검증 방법은 기존의 연구 중에 MIT Oxygen 프로젝트 중에 하나인 'Cricket' [16]에서 검증한 시나리오 테스트 방식을 참고하여 본 논문에서도 그와 유사한 방식을 적용하였다. 검증 항목은 이동노드를 소유한 사용자가 고정노드로 접근하거나 멀어질 때, 각각 다른 사용자가 고정노드에 접근할 때 미들웨어의 동작, 사용자가 고정노드에 접근할 때 시간대별 미들웨어의 수행동작 등 다양하다. 단, 본 논문에서는 이동노드를 소유한 사용자가 고정노드로 접근할 때와 다른 고정노드로 멀어져 갈 때에 대한 가장 일반적인 검증을 수행한다. 그림 7은 응용 시스템의 검증 환경을 보여준다. 검증을 위한 장소로는 대학원 내의 '데이터베이스 연구실'과 '미디어 통신 연구실'을 이용했으며, 각 장소에는 블루투스 이동노드를 감지할 수 있는 고정노드가 설치되어 있다. 각 장소간 거리는 직선거리로 약 60m 가량으로서 복도로 연결되어 있다. 사용자는 이동노드를 가지고 '데이터베이스 연구실'에서 '미디어 통신 연구실'로 이동하거나 '미디어 통신 연구실'에서 '데이터베이스 연구실'로 이동을 하며, 각 상황에 대해서 정상적으로 동작하는지 검증한다.

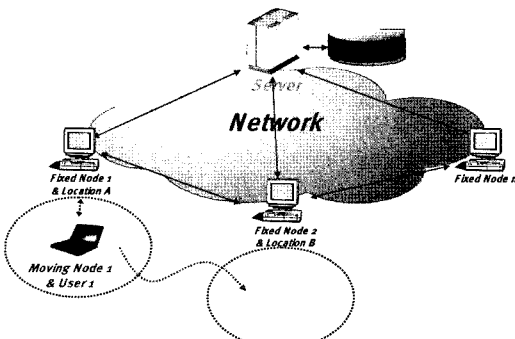


그림 7 사용자 이동 상황에 따른 동작 과정

• 사용자의 고정노드로의 접근 시에 대한 검증

이동노드를 소유한 사용자가 고정노드로 접근했을 때의 동작 화면은 그림 8과 같다. ①은 사용자 접근 감지와 함께 사용자의 이름을 이동노드로 부터 획득한 모습이고, 아울러 인증절차를 수행하는 것을 확인할 수 있다. 인증절차가 끝나면 사용자에게 따라 정의된 음악을 검색하고, ②의 모습과 같이 시각에 따른 음악을 서버로부터 다운로드 받는다. 다운로드가 완료되는 시점에서 ③과 같은 사운드 플레이어가 수행되어 음악이 연주되는 것을 확인할 수 있다.

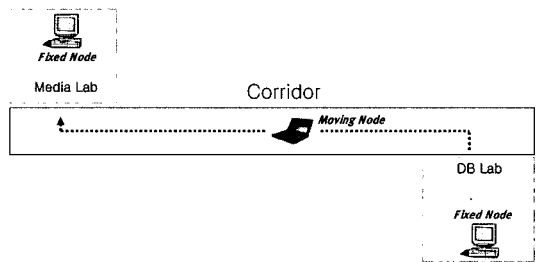


그림 8 응용 시스템을 위한 검증 환경

• 고정노드로부터 사용자가 멀어져갈 때에 대한 검증

그림 9는 사용자가 고정노드로부터 멀어져 갈 때, 미들웨어가 수행하는 동작 화면을 나타낸다. 사용자가 고정노드로 부터 멀어져서 더 이상 블루투스와 무선 통신이 불가능하다고 미들웨어가 감지하면, ①과 같이 예러 메시지를 출력하게 되며, 현재 음악을 연주하고 있는 사운드 플레이어의 프로세스 번호를 탐지하여 프로세스를 종료시킨다. ②는 사운드 플레이어의 프로세스가 종료되어 사라진 모습을 나타내고 있다. 이와 같이 이동노드의 고정노드로의 접근 및 멀어지는 상황인식 처리 기술에 대해, 장소 A와 장소 B를 기반으로 블루투스 장치의 전파가 도달하는 범위가 서로 인접해 있을 때, 사용자의 장소 이동에 따라서 그림 8과 같이 한 쪽 장소(장소 A)에서는 음악이 재생되고, 다른 한 쪽 장소(장소 B)에서는 그림 9와 같이 음악이 정지됨을 확인 할 수 있다.

4.2 응용시스템의 성능 분석

응용 시스템에서 가장 중요한 성능분석 대상인 이동노드에 대한 성능을 고찰한다. 표 4는 응용 시스템에서 일어나는 각 상황을 인식하는 데 걸리는 시간을 보인다. 첫째, 이동노드 접근에 따른 고정노드와 이동노드의 접속 설정에 있어서, 상황 인식을 하는 데 걸리는 시간은 평균 약 1.34초 이다. 이는 이동노드가 고정노드의 영역에 들어왔을 때, 미들웨어가 이동노드의 접속을 감지하는 시간을 의미한다. 이동노드가 고정노드에 접속하기 위해서는 블루투스를 통한 무선 통신에 의해 서로 간의

표 4 이동 노드에 대하여 상황인식을 위해 걸리는 시간 (단위:초)

평가 요소	시간
이동 노드 접근에 대한 평균 접속 설정	1.34
이동 노드 접근에 대한 해당 서비스 평균 시작(Sound Play)	0.50
이동 노드가 멀어짐에 따른 평균 접속 단절	1.45

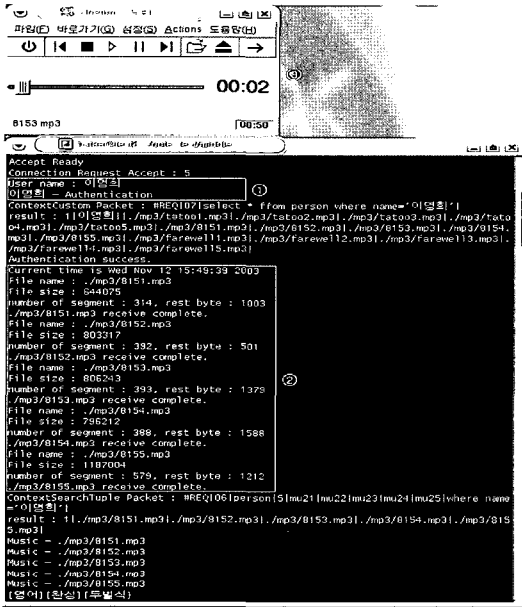


그림 9 고정노드로 접근 시 시스템 동작화면

접속이 설정됨에 따라, 블루투스 자체의 명세에 따라 상황인식 시간이 결정된다. 둘째, 이동노드 접근에 대한 해당 서비스를 시작하는 데 걸리는 평균 시간은 약 0.5 초이다. 이는 미들웨어가 컨택스트 서버를 통해 사용자의 프로파일을 검색하고 음악 모듈을 호출하는 시간을 의미한다. 여기서, 사용자의 프로파일을 검색하는 시간은 TCP/IP의 패킷 전송 시간 및 컨택스트 서버의 DBMS 성능 및 프로세서에 영향을 받는다. 음악 모듈을 호출하는 시간은 운영체제 자체가 음악 모듈 혹은 프로세스를 적재하는 시간으로 커널 자체의 가용 메모리와 하드디스크 속도에 영향을 받는다. 마지막으로, 이동노드가 멀어짐에 따른 접속 단절을 위한 평균 시간은 약 1.45초가 소요된다. 이는 이동노드가 고정 노드를 벗어났을 때, 상황인식을 하는 데 걸리는 시간을 의미한다. 이동노드는 고정 노드의 영역을 벗어나서 서로간의 통신이 불가능함에도 불구하고, 리눅스 커널은 내부적으로 계속적인 통신을 시도하려고 한다. 이에 따라 커널이 소켓 자원을 반환하는 데 오랜 시간이 걸리기 때문에 전체적으로 상황인식을 하는 데 많은 시간이 요구된다. 따라서 미들웨어 자체적으로 스레드(thread)를 생성하

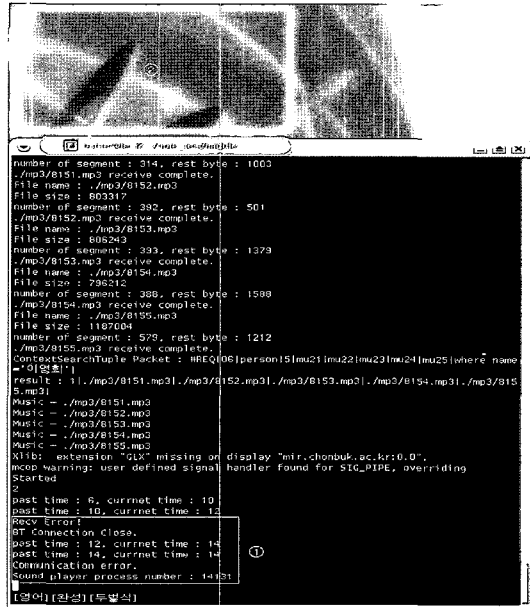


그림 10 고정노드로부터 멀어질 때의 시스템 동작화면

여, 서로간의 통신이 되지 않을 때 약 2초간의 타임 한계를 두면 효과적이다.

본 논문에서의 성능 분석에서 소요시간 측정은 기존의 MIT Oxygen 프로젝트 'Cricket'의 연구에서 성능 실험을 위해 Boundary 성능 평가 부분에서 비콘(beacon)의 영역(boundary) 감지 능력을 평가하기 위해 시간을 측정할 것을 유사하게 적용한 것이다. 특히 소요시간은 상황인식에서 컨택스트를 인식하고 접속을 단절시키는 것이 매우 중요한 기준이므로 이 부분에 대한 시간을 측정하였으며, 이 시간이 너무 늦어지면 사용자는 이를 오류상태로 간주하므로 그 이전에 수행을 완료해야 한다. 따라서 컨택스트를 인식하는 시간으로서 1.5초의 시간은 일반적인 오류로 인식하는 시간 범위보다 작으므로 프로그램 동작에는 문제가 없다고 간주한다.

5. 결론 및 향후 연구

본 논문에서는 유비쿼터스 컴퓨팅에서 상황인식 기반의 미들웨어 및 컨택스트 서버를 제안하고 이를 이용한 응용시스템을 구현하였으며 연구의 특징을 간략하게 요약하면 다음과 같다. 첫째, 제안된 상황인식 기반의 미

들웨어는 블루투스 무선 통신 기술을 이용하여 이동성을 지닌 이동노드를 발견하고, 컨텍스트 서버에 등록하며 해당 컨텍스트에 적합한 실행 모듈을 서비스하도록 구축되었으며 아울러 컨텍스트 서버는 주어진 원격 객체의 정보와 상황 정보를 효율적으로 DBMS 서버에 저장하고 검색하는 역할을 수행함으로써 다양한 응용에 적용할 있는 프로토타입 시스템을 구성하였다. 둘째, 구현된 시스템의 유용성을 검증하기 위해 상황인식 기반의 음악 재생 서비스가 가능한 응용 시스템을 구현하였다. 셋째, 상황인식을 위해 요구되는 일련의 복잡한 과정을 간략하고 명료하게 기술하고 상황에 대한 정의들 미리 규격화된 구문으로 표현함과 동시에 자동적으로 처리할 수 있는 컨텍스트 정의의 스크립트 언어 및 언어 처리기를 이용하였다. 그에 반해, 연구의 한계점으로는 첫째, 미리 정의된 상황정보 뿐만 아니라 상황정보에 대한 추론 기능을 통해 센서로부터 센싱된 컨텍스트 정보를 분석하고 다양한 상황정보를 추론하는 방법을 통해 외부의 다양한 환경 변화에 적응적, 지능적 서비스가 가능하도록 하는 연구가 요구된다. 둘째, 본 연구에서 응용 시스템의 성능 분석은 구현된 시스템의 정상적으로 동작하는 지 검증하는 차원에서 성능 분석을 수행하였다. 보다 나은 컨텍스트 서비스를 위해서는 상황정보의 QoS를 고려해야 할 것이다.

향후 연구로는 본 연구의 한계점을 보완하기 위해 미리 정의된 상황정보 뿐만 아니라 다양한 상황정보를 추론할 수 있는 기법에 관한 연구와 더불어 시스템의 검증은 보다 다양한 환경에서 보다 의미있는 컨텍스트 서비스가 제공될 수 있는 상황정보의 QoS에 관한 연구가 이루어질 필요가 있다.

참 고 문 헌

[1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM*, Vol. 36(7), pp. 75-84, 1993.
 [2] G. Banavar, A. Bernstein, "Issues and challenges in ubiquitous computing: Software infrastructure and design challenges for ubiquitous computing applications," *Communication of ACM*, 2002.
 [3] C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner and W. Newstetter, "The Aware Home: A Living Laboratory for Ubiquitous Computing Research," *Proc. of the 2nd Int'l. Workshop on Cooperative Buildings*, 1999.
 [4] "Bluetooth Version 1.1 Profile", <http://www.bluetooth.com>.
 [5] "Affix : Bluetooth Protocol Stack for Linux," <http://affix.sourceforge.net>.
 [6] 심춘보, 김용기, 장재우, 김정기, "유비쿼터스 컴퓨팅에

서 상황인식을 위한 컨텍스트 스크립트 언어 및 언어 처리기", *한국정보처리학회 논문지*, 제11-A권 제7호, pp. 537-546, 2004.
 [7] P. Couderc, A. M. Kermarrec, "Improving Level of Service for Mobile Users Using Context-Awareness," *18th IEEE Symposium on Reliable Distributed Systems*, pp. 24-33, 1999.
 [8] A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, "The anatomy of a Context-aware application," *Wireless Networks Vol. 8, Issue 2/3*, pp. 187-197, 2002.
 [9] K. Cheverst, N. Davies, K. Mitchell, A. Friday, "Experiences of developing and deploying a context-aware tourist guide: the GUIDE project," *Proceedings of the sixth annual international conference on Mobile computing and networking*, pp. 20-31, 2000.
 [10] 장세이, 우운택, "ubiHome을 위한 컨텍스트 기반 응용 서비스 모형", *정보과학회논문지:소프트웨어 및 응용*, 제30권 제6호, pp. 550-558, 2003.
 [11] 송재훈, 김동균, 이상정, "블루투스를 이용한 상황인식 서비스", *한국통신학회 추계종합학술발표집 제28권*, pp. 254-257, 2003.
 [12] 박한솔, 최태욱, 정기동, "jini 기반의 context-aware chatting program", *한국정보처리학회 추계 학술발표 논문집*, 제10권 제2호, pp. 1177-1180, 2003.
 [13] Anind K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing Journal*, Vol. 5, No. 1, pp. 4-7, 2001.
 [14] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," *Dartmouth Computer Science Tech. Report TR2000-381*, 2000.
 [15] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," *In Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 85-90, 1994.
 [16] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *6th ACM/IEEE Int'l Conf. on Mobile Computing and Networking(MOBICOM)*, pp. 32-43, 2000.



심 춘 보
 1996년 전북대학교 컴퓨터공학과(공학사). 1998년 전북대학교 컴퓨터공학과(공학석사). 2003년 전북대학교 컴퓨터공학과(공학박사). 2004년~2005년 1월 부산 가톨릭대학교 컴퓨터정보공학부 전임강사. 2005년 2월~현재 순천대학교 정보통신공학부 전임강사. 관심분야는 멀티미디어 DB, 멀티미디어 IR, 상황인식



태 봉 섭

2002년 전북대학교 컴퓨터공학과(공학사). 2004년 전북대학교 컴퓨터공학과(공학석사). 2004년 롯데정보통신 롯데마트 IS 사업팀. 2005년~현재 NHN(주) 게임 엔진팀 연구원. 관심분야는 리얼타임 3D 엔진, 3D 공간분할 색인구조, CBD



장 재 우

1984년 서울대학교 전자계산기공학과(공학사). 1986년 한국과학기술원 전산학과(공학석사). 1991년 한국과학기술원 전산학과(공학박사). 1996년~1997년 Univ. of Minnesota, Visiting Scholar. 2003년~2004년 Penn State Univ., Visiting Scholar. 1991년~현재 전북대학교 컴퓨터공학과 교수. 관심분야는 공간 네트워크 데이터베이스, 상황인식, 하부저장구조



김 정 기

1992년 전북대학교 컴퓨터공학과(공학사). 1994년 전북대학교 컴퓨터공학과(공학석사). 1999년 전북대학교 컴퓨터공학과(공학박사). 1998년~현재 한국전자통신연구원 임베디드SW연구단 선임연구원. 관심분야는 임베디드 시스템, 파일시스템, 병렬 정보검색



박 승 민

1981년 울산대학교 전자공학과(공학사) 1983년 홍익대학교 전자공학과(공학석사). 1998년~현재 충남대학교 전자공학과 박사과정. 1984년~현재 한국전자통신연구원 임베디드SW연구단 편제형컴퓨팅 미들웨어연구팀장/책임연구원. 관심분야는 임베디드 시스템, 편제형 컴퓨팅, 이동통신망