韓國 컴퓨터情報學會 論文誌
第11卷 第5號, 2006. 11.

2006-11-5-1-12

분산 유비쿼터스 환경을 위한 상황 인식 미들웨어의 설계 및 구현

김 은 영*, 오 동 열**

Design and Implementation of Context-Aware Middleware for Distributed Ubiquitous Environments

Kim Eun Young *, Oh Dong Yeol **

요 약

상황 인식은 유비쿼터스 환경에서 여러 가지 상황 정보를 인식하고 이를 기반으로 자동으로 서비스를 제공하기 위한 요소 기술이다. 유비쿼터스 환경을 지원하기 위한 기존 미들웨어들은 중앙 집중화된 저장 장치나 DBMS를 이용하여 획득된 상황 정보와 서비스 컨텐츠를 저장·관리하였다. 집중화된 상황 정보 및 서비스 컨텐츠의 관리는 이동 노드의 자율성과 서로 다른 미들웨어 간에 상호 운영을 저해하는 경우가 있다. 이에 본 논문에서는 분산된 유비쿼터스 환경에서 상황 정보를 이동 노드에 분산 관리하고 서비스를 위한 컨텐츠를 미들웨어 간에 공유할 수 있는 시스템을 설계한다. 이와 더불어 상황 정보의 정의와 서비스의 추론 및 호출을 지원하기 위한 상황 인식 스크립트를 제공한다. 본 연구에서 설계된 응용 시스템을 상황 인식 기반의 음악 재생 서비스를 제공하는 시나리오에 적용함으로서 유용성을 보이고 있다.

Abstract

Context-Awareness is a important technology to support optimized services automatically by recognizing various context informations in ubiquitous environments. Previous middlewares which supports ubiquitous environments used a centralized storage and DBMS to store and manage context informations and service contents. Centralized management of context informations and service contents sometimes hinders the autonomy of moving node and interoperability between difference middlewares. In this paper, we design the systems which stores context informations in moving node by distributed form and shares service contents between middlewares in distributed ubiquitous environments. And it provides Context-Aware scripts to supports the definition of context informations , reasoning and execution of services. It verifies the usefulness of the designed systems by applying the scenario of music playing service based on context awareness

▸ Keyword : 유비쿼터스(Ubiquitous), 미들웨어(Middleware), 상황 인식(Context-Aware), 분산 컴퓨팅 (Distributed Computing), P2P(Peer to Peer)

• 제1저자 : 김은영
• 접수일 : 2006.10.14, 심사일 : 2006.10.18, 심사완료일 : 2006.10.24
* 안산공과대학 디지털 미디어과 교수 ** 숭실대학교 컴퓨터 공학과 박사 수료
※ 이 논문은 2005학년도 안산공과대학 학술연구비에 의하여 연구된 것임

Ⅰ. 서 론

사이버 스페이스가 컴퓨터와 네트워크로 구성된 가상 공간상에 사람이 개입하는 방식이었다면 제록스 팔로 알토(Palo Alto) 연구소의 마크 와이저(Mark Weiser)에 의해 처음 발표된 유비쿼터스 컴퓨팅은 사람이 존재하는 공간에 다양한 형태의 센서를 통해서 컴퓨터 군이 개입하는 방식으로 사용자는 시공의 제약을 받지 않고 공간 내에 편재해 있는 여러 가지 형태의 컴퓨팅 자원을 이용하여 다양한 서비스를 제공받을 수 있다.[1]

기존 컴퓨팅 환경에서는 키보드나 마우스와 같이 정형화된 형태의 입력 장치를 통해서 자신의 의도를 전달하고, 프린터나 모니터 혹은 사운드 카드와 같이 사용자의 오감으로 인식할 수 있는 형태의 출력 장치로 그 결과를 인지하였다. 이에 비해, 유비쿼터스 컴퓨팅 환경에서는 사용자의 실생활에 편재되어 있는 다양한 센서와 컴퓨팅 자원들이 사용자의 의도와 주변 환경을 인식하고 이를 근거로 사용자에게 최적의 서비스를 제공해야 한다. 이와 같이 사용자가 필요로 하는 서비스를 제공하기 위해서는 일상 공간에 편재되어 있는 다양한 형태의 센서와 컴퓨터들이 수집한 각종 정보를 효과적으로 공유하여 사용자와 사용자가 위치한 주변 환경의 정보를 자동으로 인식하고 처리할 수 있는 상황 인식 기술이 요구된다.[2] 상황 인식 기술은 센서나 기타 컴퓨터 군으로부터 정보를 획득하고 획득된 정보로부터 상위 상황 정보를 유추할 수 있어야 한다. 이와 더불어 획득된 정보로부터 유추된 상황 정보를 기반으로 사용자에게 최적화된 서비스가 제공되어야 하며 이러한 일련의 과정을 관리 및 운영할 수 있는 일관된 처리 기술이 요구된다.

유비쿼터스 환경에서 제공되는 다양한 형태의 서비스 중에 음악 재생 서비스나 동영상 재생 시버스와 같이 컨텐츠를 기반으로 하는 서비스의 경우, 서비스를 제공하기 위한 해당 컨텐츠가 서로 다른 네트워크의 노드에 다양한 형태로 편재되어 있을 수 있다. 따라서 컨텐츠 기반 서비스는 서비스를 제공하는 미들웨어가 해당 컨텐츠를 가지고 있지 않더라도 미들웨어간의 분산된 형태의 컨텐츠를 효과적으로 공유함으로서 사용자에게 양질의 서비스를 제공할 수 있어야 한다.

이에 본 논문에서는 분산 환경에서 컨텐츠 기반의 서비스를 위한 상황 인식 미들웨어를 제안하고, 이를 이용한 응용 시스템을 구현한다. 본 논문에서 제안한 상황 인식 미들웨어는 다음과 같은 관점에서 연구 한다. 첫째, 상황 정보를 성격에 따라 분산 관리한다. 획득된 상황 정보를 상황 정보 미들웨어와 이동 노드에 분산 저장함으로서 정보의 이동성 및 관리를 용이하게 한다. 둘째, 네트워크에 산재해 있는 다양한 컨텐츠를 상호 공유 할 수 있는 P2P 구조를 확장·적용함으로서 중앙 집중형의 서버 기반 서비스에 비해 보다 더 많은 컨텐츠를 제공하고 장애 발생 시 이를 해결한다. 셋째, 센서 혹은 사용자 인터페이스를 통해서 획득된 정보는 컴퓨터가 이해할 수 있는 형태로 기술한다. 또한 기술된 상황 정보를 이용하여 상위 상황 정보의 유추와 서비스의 추론이 가능토록 하고 이와 같은 일련의 작업들은 쉽게 모델일이 가능토록 한다. 본 논문의 구성은 다음과 같다. 제 2장에서는 상황 인식 처리와 관련된 기존 연구와 상황 정보를 표현하기 위한 모델링 방법 및 분산 컴퓨팅 환경에서 컨텐츠 기반의 서비스를 제공하기 위한 기존 연구를 소개한다. 제 3장에서는 본 논문에서 제안하는 상황 인식을 위한 미들웨어 및 모델링 방법과 분산 환경에서의 서비스 자원 공유를 위한 구조를 설명한다. 제 4장에서는 제 3장에서 제안된 미들웨어 및 응용 시스템의 구현하고 스트리밍 기반의 음악 재생 시스템을 통해서 이를 검증한다. 마지막으로 제 5장에서는 결론과 향후 연구를 제시한다.

Ⅱ. 관련 연구

본 장에서는 상황 인식에 관련된 기존 연구와 분산 환경에서 컨텐츠 기반 서비스와 관련된 연구를 살펴본다.

2.1 상황 인식의 정의 및 요구 사항

사전적 의미의 '상황'이란 무언가가 존재하거나 발생한 경우, 상호 연관성 있는 상태를 의미한다.[3] 이와 같이 사전적인 의미를 컴퓨터 환경과 연관 지어 의미 있는 정의를 내리려는 노력으로 Schilit는 최초로 상황 정보를 사용자의 위치, 사용자의 정보, 공간 내에 사용 가능한 물리적 대상의 정보, 사용자와 상호 작용하는 대상의 상태로 정의하였다.[4] 그 이후, Dey는 대상을 특정화 할 수 있는 정보로서 애플리케이션과 사용자 사이에 사용

자, 사물, 대상물 등의 개체 상태를 나타내는 정보로 정의 하였다.[5] 국내에서는 광주과학 기술원의 우운택은 상황 정보를 응용 서비스에 따라 5W1H(Who, What, Where, When, Why, How)의 조합으로 정의하였다.[6] 이와 같은 정의를 기본으로 상황 인식에 필요한 요구 사항을 정립하였다. 먼저 Schilit는 상황 인식 시스템의 구성 요소로 첫째, 사용자와 가까이에 있는 대상물과의 선택을 용이하게 하는 인터페이스, 둘째, 상황 정보가 변화함에 따라 새로운 서비스 컴포넌트가 추가, 삭제 및 새로운 연관 관계를 형성하는 것 셋째, 상황 정보에 따라서 다른 결과를 제공하는 것과 마지막으로 상황 정보 시스템이 작동해야하는 관계를 명시한 간단한 룰을 언급하였다.[4] 이후에 Pascoe는 상황 인식에 필요한 요소를 상황 정보 센싱, 상황 정보 적응, 자원 발견, 상황 정보의 디지털 데이터로 변환으로 구분하였다.[7] Dey는 이러한 개념을 정리, 통합하여 상황 인식 응용 프로그램이 지원해야 하는 요소로서 첫째, 사용자에게 정보와 서비스를 제공 하는 기능, 둘째, 사용자를 위하여 서비스를 자동으로 수행하는 기능, 셋째, 이후 검색을 위한 상황 정표의 표시 기능으로 정의하였다.[8]

2.2 상황 인식 기반 응용 연구

상황 인식을 기반으로 한 응용 연구는 다음과 같다.

첫째, 미국 일리노이 대학의 Gaia는 물리적인 공간과 소프트웨어 인프라가 함께 자연스럽게 융화되는 액티브 공간을 보다 용이하게 구현할 수 있는 프레임워크를 제공해주는 소프트웨어 미들웨어로서 운영체제의 개념을 도입하였다.[9] <그림 1>은 Gaia의 시스템 구조이다.

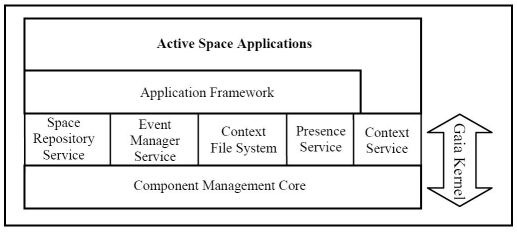

그림 1. Gaia 시스템 구조
Fig 1. System Architecture in Gaia

Gaia의 커널은 컴포넌트 관리 커널과 5종류의 기본 서비스(Space Repository, Event Manager Service, Context File System, Presence Service, Context Service)를 제공하고 있다.

둘째, 미국의 카네기 메론 대학에서 수행하고 있는 Aura 프로젝트는 하드웨어에서부터 운영 체제, 미들웨어, 응용 프로그램, 사용자를 망라하는 프로젝트로서 사용자 환경을 Aura라는 추상적인 개체로 모델링한다. Aura는 사용자의 요구를 파악하고 이를 시스템 전반에 표현해 주는 Prism, 서로 다른 공간상에서 필요한 데이터를 공급해주는 Coda, 가용 자원의 모니터링과 적응을 수행하는 Odyssey, 이동 단말을 지원하기 위하여 자원의 인식 및 지원을 담당하는 Spectra로 구성되어 있다[10]. <그림 2>는 Aura 프로젝트의 시스템 구조이다.

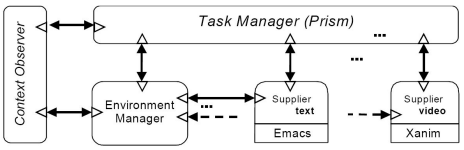

그림 2. Aura의 시스템 구조
Fig 2. System Architecture in Aura

셋째, 국내 연구로 광주 과학 기술원의 우운택이 제시한 ubi-UCAM이 있다. ubi-UCAM은 상황 정보 인식 응용 서비스 통합을 위한 상황 정보 통합기로, 각 센서로부터 5W1H 형태의 초벌 상황 정보를 획득하고 이를 상황 정보 통합기가 통합 상황 정보로 생성한다. 이를 관리기가 전달받고 해당 서비스를 결정하여 처리기가 처리하는 형태로 작동한다.[11] <그림 3>은 ubi-UCAM의 시스템 구조이다.

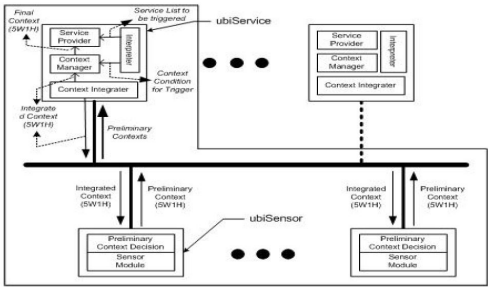

그림 3. ubi-UCAM의 시스템 구조
Fig 3. System Architecture in ubi-UCAM

2.3 상황 정보의 모델링

본 장에서는 상황 정보 모델링에 대한 기존 연구를 데이터의 구조 관점에서 분류 한다.

첫째, 상황 정보를 키와 해당 값(Key=Value)모델로 표현하는 방식이다. 키와 해당 값 형태의 상황 정보 모델링 방법은 묘사하고자 하는 요소들을 나열하고 이에 해당하는 값을 할당하는 방법이다. Schilit는 최초로 키와 해당 값 모델 형태로 상황 정보를 정의하여 이를 서비스에 적용하였다.[4] 이와 같은 형태는 관리하기 쉽다는 장점이 있으나 실세계에서 발생하는 복잡한 상황 정보를 효과적으로 모델링하기에는 한계가 있다.

둘째, 마크업 스킴(Markup Scheme) 모델이다. 마크업 스킴은 마크업 태크를 이용하여 속성과 내용을 계층적으로 표현할 수 있다. ConteXtML은 서버와 사용자간에 상황 정보를 교환하기 위하여 XML을 확장하였고[12], Held는 CSCP(Comprehensive Structured Context Profiles)을 이용하여 이를 상황 인식 서비스에 적용하였다.[13] 마크업 스킴 모델의 경우, 확장성과 더불어 사용의 편의성을 제공하나 상황 정보들 간에 연관 관계를 표현하는데 취약점이 있다.

셋째, 그래픽 기반의 상황 정보 모델링으로 일반적으로 잘 알려진 UML(Unified Modeling Language) 형태를 확장하여 표현한다. 관련 연구로 Henrickse은 객체의 역할을 중심으로 이를 모델링한 ORM(Object - Role Modeling)이 있다.[14] 이와 같은 그래픽 기반의 경우, 상황 정보를 도식적으로 표현하여 사용자에게 개념적인 정보 전달에는 효과적이나 실제 시스템에 도입하기 위해서는 로직 기반 모델이나 온톨로지 기반 모델과 같이 시스템이 이해할 수 있는 형태로 변환해야 한다는 단점이 있다.

넷째, 객체 지향(Object Oriented)모델이 있다. 객체 지향 모델은 기존 객체 지향이 제공하는 정보 은닉이나 상속, 재사용성을 고려한 모델링 방법으로 TEA 프로젝트에서 개발한 cues에서 적용하였다.[15] TEA 프로젝트에서는 물리적이나 논리적인 센서들을 추상화 하였으며 cue는 이러한 센서 값들을 인자 값으로 받는 함수로 묘사하였다. 객체 지향 모델의 경우, 객체에 대한 추상화를 통해서 상황 정보와 시스템 개발 사이에 추상화를 통한 개념적인 분리가 가능하며 재사용성을 높일 수 있다는 장점이 있다.

다섯째, 로직 기반 모델이 있다. 로직 기반 모델은 사실이나 기호적 표현들을 바탕으로 어떠한 사실에 대하여 정의하기 위해 사용된다. 로직 기반의 모델에서는 상황 정보를 상황의 사실 묘사와 여러 가지 연관 기호가 결합된 형태로 정의한다. 상황 정보를 로직 기반 모델로 표현하기 위한 시도로는 Akman이 제시한 'Extended Situation Theory'가 있다.[16] Akman은 상황에 대한 정보를 보다 자연어에 가까운 형태의 규칙과 가정을 연관하여 표현하였다. 이와 더불어 Gaia의 경우, 센서를 통해서 획득된 상황 정보를 기술하기 위하여 로직 기반 모델을 XML 형식으로 변환하여 사용한다.[9] 이와 같은 로직 기반의 경우, 사용자가 정의된 모델에서 상황 정보를 도출하기 위하여 사용된 규칙 문법이나 예약어들의 정의를 이해해야 한다는 단점이 있으나, 컴퓨터가 이해하기 가장 수월한 형태로 기존 인공지능 분야에서 다양한 형태로 활용하고 있다.

2.4 분산 환경 기반의 컨텐츠 서비스

다양한 멀티미디어 기반의 컨텐츠를 서비스하기 위한 대부분의 환경은 대용량 서버를 구축하고 이를 기반으로 하는 클라이언트/서버 모델로 구성되고 있다. 이와 같은 모델의 경우 서버의 과부화로 인하여 네트워크의 다운이나 속도 저하 등을 야기 시킬 수 있다. 이러한 문제를 해결하기 위해 분산된 환경에서 P2P 네트워크를 도입하여 자원의 공유와 배분을 보다 원활하게 서비스 할 수 있다. P2P 모델은 인덱스 서버의 유무에 따라 <그림 4>와 같이 두 가지로 구분된다.

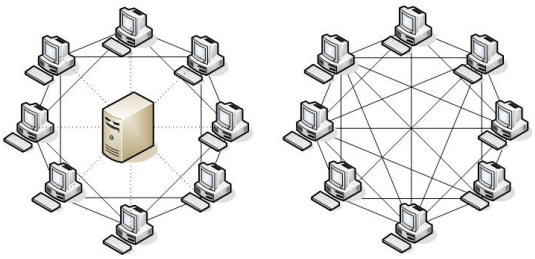

그림 4. 혼합형 P2P와 순수 P2P 네트워크
Fig 4. Hybrid P2P and Pure P2P Network

혼합형 P2P(Hybrid P2P)의 경우, 중앙에 자원에 대한 위치 정보를 가지고 있는 인덱스 서버가 존재하며

이러한 인덱스 서버가 검색에 사용되는 형태의 네트워크이다.[17] 순수 P2P(Pure P2P)는 어떠한 서버도 중간에 존재하지 않으며 호스트 간에 직접 검색 과정을 수행하는 네트워크이다.[18]

III. 분산 환경을 위한 상황 인식 미들웨어의 설계

본 장에서는 유비쿼터스 컴퓨팅 환경에서 상황 인식 처리를 위해 제안된 상황 인식 미들웨어와 서비스를 위한 전체적 구조에 대하여 설명한다. 제안 시스템은 상황 인식 미들웨어, 스트리밍 전송 및 재생 애플리케이션과 이동 노드 디바이스 세 가지로 구성된다. 상황 인식 미들웨어는 여러 가지 상황 정보를 수집하고 이에 적합한 서비스를 추론하기 위한 미들웨어를 의미하고 스트리밍 전송 및 재생 애플리케이션은 서비스를 실제 제공하기 위한 애플리케이션을 의미하며, 이동 노드 디바이스는 사용자의 휴대형 디바이스를 의미한다. 이동 노드 디바이스는 여러 가지 형태가 될 수 있으나 상황 정보의 저장이 가능한 형태의 디바이스로 제한 한다(단순 USB 저장 매체에서부터 휴대형 노트북등이 이에 해당한다). 서비스 구역은 각각의 미들웨어가 서비스를 제공하기 위한 물리적 지역을 의미하며 본 논문에서 제안한 전체 시스템 구조는 <그림 5>와 같다.

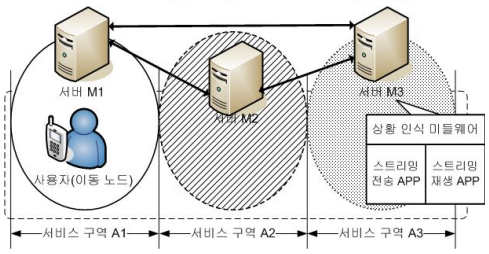

그림 5. 분산된 환경에서의 상황 인식 기반 서비스 구조
Fig 5. Context-aware based service structure in distributed environments

3.1 자생적 상황 정보의 응용

본 논문에서는 상황 정보를 미들웨어와 이동 노드 디바이스에 분산 및 저장 관리하기 위해서 숭실대에서 제시한 자생적 상황 모델을 이용한다.[19] 자생적 상황 모델은 기본적으로 상황 정보를 그 특징에 따라서 표 1과 같이 네 가지로 구분한다.

표 1. 상황 정보의 정의 및 분류
Table 1. Definition and Classification of Context Informations

상황 정보	정의	예
사용자 상황정보	사용자에 관련된 일련의 정보	성별, 나이, 교육,직업, 음성, 수화등
환경 상황정보	서비스 공간 내에 존재하는 물리적인 환경에 관련된 정보	위치, 시간, 날씨, 요일, 책상 위치 등
자원 상황정보	서비스 공간내 에 컴퓨팅 시스템에 관련된 하드웨어 및 소프트웨어 자원	RFID, 인터넷망, 핸드폰, PDA, 센서, 카메라 등
서비스 상황정보	사용자와 서비스 사이에 상호 작용에 관련된 정보	재생 채널 및 영화 정보, 방문 기록 등

사용자에게 제공된 일련의 서비스와 관련된 서비스 상황 정보는 이동 노드 디바이스에 저장되며 필요에 따라 상황 인식 미들웨어가 이를 갱신 및 삭제하게 된다.

3.2 상황 정보의 모델링과 서비스 추론

상황 정보 모델링과 서비스 추론을 위해서 일차 술어 논리(First-Order Predicate Logic)를 확장 적용한다. 상황정보의 모델링은 기본적으로 '상황 정보'라는 집합을 이용한다. (1)은 상황 정보의 서술적인 표현이다.

상황 정보{정보 종류, 주체, 관계, 대상}.....(1)

첫째, '상황 정보' 집합은 상황 정보의 상위 개념을 의미한다. '정보 종류' 속성은 '상황 정보'가 실제로 구체화된 것을 의미하며 '주체' 속성은 해당 상황 정보의 주어에 해당한다. 주체와 대상 간을 연결하는 '관계' 속성은 주체와 대상과의 연관성을 의미하며 동사나 보어, 혹은 술어 연산 기호를 의미한다. 사용 가능한 술어 연산 기호는 크기의 대소를 비교하는 '>', '<'와 등식 기호 '='가 있다. 다음은 일차 술어 논리를 확장하여 표 1에서 분류한 각 상황 정보를 제시한 방법으로 표현한 예이다.

- Location(철수, entering, room 501) : 철수가 501호를 들어간 상황[환경 상황 정보]
- APP(192.168.10.12, running, powerpoint.exe) : IP가 192.168.10.12인 네트워크 노드에 실행중인

애플리케이션이 파워 포인트인 경우[자원 상황 정보]

- Audiable(192.168.10.12/고백.mp3, stop, 1:03) : IP가 192.168.10.12인 네트워크 노드에 고백.mp3라는 파일은 듣기가 가능하고 현재 1분 3초에서 정지하고 있음[서비스 상황 정보]
- Sex(철수, is , man) : 철수라는 이름의 사람이 남성이라는 것을 의미[사용자 상황 정보]

하나의 상황 정보를 묘사하는 '상황 정보' 집합은 연관관계를 나타내는 기호로 집합 군으로의 표현이 가능하다. '상황 정보' 집합은 관계 연산자와 기호에 의해서 상호 연관성을 갖을 수 있다. 사용 가능한 관계 연산자는 표 2와 같다.

표 2. 상황 정보 집합 간에 관계
Table 2. Relations Between Context information Sets

상황 정보	정의
A ∧ B	상황 종류 A와 B가 동시에 발생하는 경우
A ∨ B	상황 종류 A 혹은 B 인 경우
IF A THEN B	상황 종류 A 이면 상황 종류 B(상위 추론) 상황 종류 A 이면 B를 실행(서비스 추론)
NOT A	상황 종류 A가 아닌 경우

상황 종류 집합의 관계 연산자 중에서 *IF ~ THEN*은 일종의 규칙을 나타내는 연산자로서 두 가지 의미로 사용된다. 첫째는 주어진 상황 종류로부터 상위 상황 정보를 유추할 경우이다. 이러한 경우에는 *'IF 전제 THEN 가설'*의 의미로 사용 된다. 두 번째는 주어진 상황 정보에 따른 행동을 정의하는 경우이다. 이러한 경우에는 *'IF 상황 THEN 행동'*의 의미로 사용된다. 일련의 규칙으로 표현된 상황 정보 집합 간에 상위 상황 정보의 유추와 행동을 결정하기 위하여 정방향 추론(Forward Chaining)이 사용된다.[20] 각각의 상황 정보와 지식베이스에 규칙들의 조건 부분과 매치하여 일치하는 규칙이나 상위 상황 정보를 찾아 해당 상위 정보로 유추하거나 해당 서비스를 실행한다.

3.3 미들웨어의 설계

제안하는 미들웨어는 전체적으로 <그림 6>과 같이 3단계의 개념적인 계층으로 구성된다.

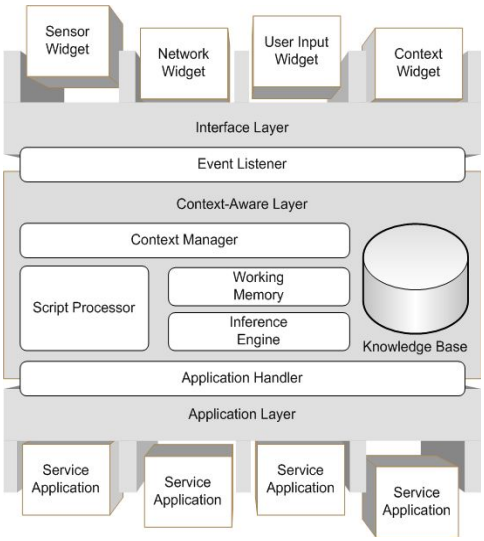

그림 6. 분산된 환경에서의 상황 인식 기반 서비스 구조
Fig 6. Context-Aware Based Service Architecture in distributed environments

첫째, 인터페이스 계층은 미들웨어가 외부로부터 통신하기 위한 계층으로 물리적 특징에 따라서 위젯(Widget)이라는 블랙박스 개념으로 추상화되어 있다. 위젯은 그 특징에 따라서 표 3과 같이 네 가지로 구분된다.

표 3. 위젯의 정의와 분류
Table 3. Definition and Classification of Context Widget

상황 정보	정의
센서 위젯	센서를 통해서 데이터를 획득하기 위한 인터페이스
네트워크 위젯	네트워크 위젯을 다른 미들웨어와의 통신, 혹은 사용자 디바이스와의 통신이 가능하다.
사용자 입력 위젯	단순한 조명on/off나 컴퓨터를 통한 입력, 혹은 음성 인식등 다양한 형태의 사용자 입력을 처리한다.
상황 정보 위젯	기타 상황 정보를 획득할 수 있는 다양한 입력장치와의 연동이 가능하다

센서, 사용자 입력, 상황 정보 위젯의 경우, 상황 정보를 획득하기 위한 입출력을 의미하며 네트워크 위젯은 미들웨어가 외부와 통신하기 위한 위젯을 의미한다.

둘째, 상황-인식 계층은 인터페이스 계층을 통해서 획득된 다양한 상황 정보를 수집하고 이를 상위 정보로 유추하고 이에 맞는 서비스를 제공하는 다음과 같이 기능별로 7가지 구성요소로 이루어져 있다.

- 이벤트 관리자(Event Listener) : 획득된 상황 정보와 통신 정보는 이벤트 관리자를 통해서 상황-인식 계층에 이벤트 형태로 전송 된다.
- 스크립트 처리기(Script Processor) : 정의된 상황 인식 스크립트를 분석하여 명시된 규약에 따라 수행한다.
- 상황정보 관리기(Context Manager) : 인터페이스 계층을 통해서 입력된 상황 정보를 스크립트 형태로 변환하고 이를 실행 메모리에 전송한다.
- 실행 메모리(Working Memory) : 스크립트 처리기를 통해서 분석된 상황 정보를 저장하고, 조건 절에 해당하는 이벤트가 발생할 때 까지 다른 상황 정보의 저장을 관리한다.
- 지식 베이스(Knowledge Base) : 상황 정보에 따른 상위 상황 정보 및 서비스를 관리 및 저장한다.
- 추론 엔진(Inference Engine) : 정의된 지식 베이스(Knowledge Base)로부터 획득된 상황 정보가 조건 절을 만족하는 경우, 해당 서비스를 실행하거나, 상위 상황 정보를 유추한다.
- 응용프로그램 관리자(Application Handler) : 추론된 서비스에 적합한 애플리케이션을 호출하기 위하여 애플리케이션 계층과의 통신을 담당한다.

셋째, 애플리케이션 계층은 상황-인식 계층을 거쳐 추론된 서비스와 이에 적합한 애플리케이션을 호출하여 서비스를 제공하는 계층으로 애플리케이션 API를 통해서 연동 및 구현된다.

3.4 응용 애플리케이션의 설계

본 논문에서 제시한 응용 서비스는 미디어 자원에 대한 위치 정보를 유지하고 있는 인덱스 서버 기반의 혼합형 P2P 기반으로 설계되었다.[17]

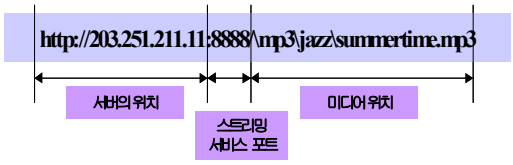

그림 7. 네트워크상에 미디어위치 정보의 표현
Fig 7. Presentation of media locational information on network

각 스트리밍 애플리케이션은 자신이 보유한 MP3 파일에 대한 위치 정보를 인덱스 서버로 전송하며 서버는 실행중인 스트리밍 애플리케이션들이 보유한 각 파일의 위치 정보를 저장 및 관리하게 된다. 파일의 위치 정보는 <그림 7>과 같은 형태로 표현된다. 애플리케이션과 인덱스 서버의 구조는 <그림 8>과 같다.

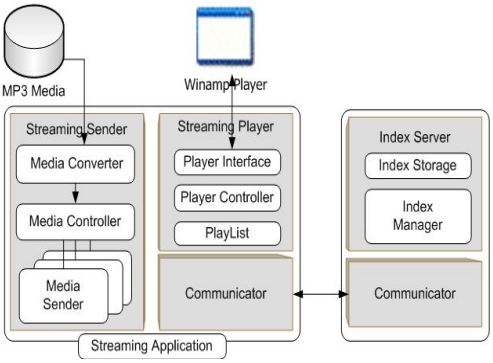

그림 8. 스트리밍 애플리케이션과 인덱스 서버 구조
Fig 8. Architecture of Streaming Application and Index server

스트리밍 애플리케이션은 스트리밍 전송기(Streaming Sender)와 스트리밍 재생기(Streaming Player) 및 통신기 (Communicator)로 분류된다. 첫째, 스트리밍 전송기는 로컬에 위치한 MP3 파일을 다른 애플리케이션의 요청에 따라 스트리밍으로 전송하는 역할을 한다. 스트리밍 전송기는 파일 형태로 저장된 MP3 파일을 버퍼에 일정한 사이즈에 저장 가능하도록 변환하는 미디어 변환기(Media Converter)와 스트리밍 데이터를 전송하기 위한 미디어 전송기(Media Sender), 미디어 변환기를 통해서 변환된 데이터를 미디어 전송기에 전달하고, 여러 개의 미디어 전송기를 제어하기 위한 미디어 제어기(Media Controller)로 구성되어 있다. 둘째, 스트리밍 재생기(Streaming Player)는 윈앰프 애플리케이션과 애플리케이션을 연동하기 위한 재생기 인터페이스(Player Interface)와 재생기의 제어를 담당하는 재생기 제어기(Player Controller), 재생할 곡을 저장 관리하는 재생 리스트(PlayList)로 구성되어 있다. 셋 째, 통신기(Communicator)는 인덱스 서버와의 통신을 담당한다. 인덱스 서버는 스트리밍 애플리케이션이 위치한 물리적 저장 공간 내의 MP3 미디어에 대한 위치 정보를 저장 및 관리하는 역할을 담당한다. 인덱스

서버는 애플리케이션과의 통신을 위한 통신기(Communicator)와 MP3 미디어의 위치 정보를 저장하기 위한 인덱스 저장소(Index Storage) 및 이를 관리하기 위한 인덱스 관리기(Index Manager)로 구분된다.

Ⅳ. 응용 시스템 구현 및 검증

4.1 응용 시스템 구현

논문에서 설계한 상황 인식 미들웨어의 기능과 성능을 실험 및 평가하기 위해서 세 개의 서비스 구역으로 분류하여 각 구역마다 서비스 미들웨어와 스트리밍 전송 및 재생 애플리케이션을 설치하고 이에 대한 실험을 진행하였다. 미들웨어와 애플리케이션은 J2SE 1.4.2를 기반으로 개발 하였으며, 음악 재생을 위한 윈앰프 제어 프로그램은 윈앰프 SDK를 이용하여 DLL 형태로 스트리밍 재생 애플리케이션과의 연계를 위해서 JNI를 이용하였다. 미들웨어간 통신과 애플리케이션과 인덱스 서버간의 통신은 TCP/IP 기반의 소캣 기반으로 한다.

4.2 응용 시스템 검증

본 장에서는 세 군데의 서로 다른 사무실 공간을 <그림 9>와 같이 서비스 공간(A1, A2, A3)으로 설정한다. 각각의 공간에는 상황 인식 미들웨어(M1, M2, M3)와 스트리밍 전송기(S1, S2, S3), 음악 스트리밍 재생기(P1, P2, P3)가 설치되어 있다.

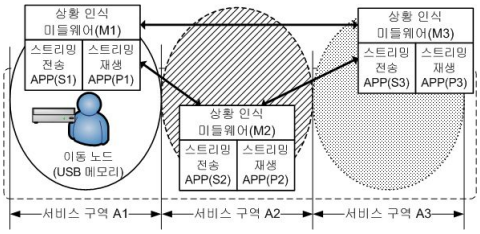

그림 9. 응용 시스템을 위한 테스트 환경
Fig 9. Test Environments for Application

세 개의 서로 다른 미들웨어 중에서 하나의 미들웨어(M1)에 P2P 기반의 인덱스 서버로 설치한다. 따라서 미들웨어들에 설치된 음악 재생 애플리케이션은 초기 구동 시, 물리적으로 같은 위치에 있는 서버의 MP3 파일 리스트를 미들웨어 M_1으로 전송한다. 사용자는 이동 노드로 USB 메모리를 사용하며 이를 미들웨어가 드라이버로 인식할 때 마다 해당 서비스 상황 정보의 유무에 따라 <그림 10>과 같은 시나리오로 진행된다.

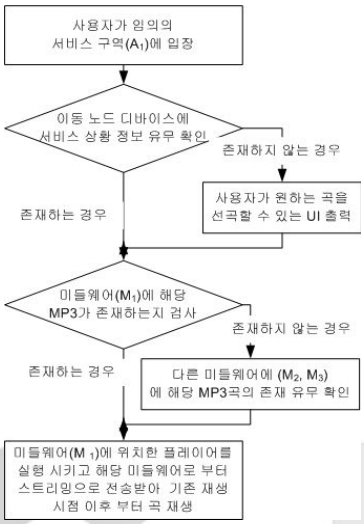

그림 10. 테스트 시나리오
Fig 10. Test Scenario

위 시나리오에 따라 사용자는 서비스 구역(A_1)에 들어 갈 때마다 자신의 USB 메모리를 미들웨어가 설치된 PC에 연결한다.

초기에 서비스 상황 정보를 저장하고 있는 파일이 존재하지 않거나 해당 파일에 서비스 상황 정보에 대한 내용이 존재하지 않는 경우, 미들웨어는 [그림 11-a]와 같이 음악 검색 UI를 호출한다. 사용자는 자신이 원하는 곡 제목의 일부분을 입력하고 인덱스 서버로 질의를 전송한다. 질의의 결과가 출력되면 사용자는 원하는 곡들을 선별하여 재생 리스트를 생성한다. 리스트를 생성한 후 [그림 11-b]와 같이 재생을 선택하여 윈앰프 애플리케이션을 수행한다. 현재 스트리밍 애플리케이션이 정상적으로 스트리밍 데이터를 전송하는지의 여부를 [그림 11-c]와 같이 확인할 수 있다. 사용자가 USB 장치를 제거한 경우, 현재 재생되고 있는 곡의 물리적 위치와

파일 이름 및 재생된 시점까지의 시간을 사용자가 선택한 재생 리스트와 함께 다음과 같은 형태로 기록한다. 이후, 사용자가 다른 서비스 지역(A_2)로 입장하면, 상황 인식 미들웨어(M_2)가 USB 장치로부터 서비스 상황 정보를 입력받고 기존에 재생된 곡의 이후 시점부터 곡을 재생하며 곡 리스트를 윈앰프 애플리케이션에 설정한다. 이와 같은 일련의 과정은 [그림 11-d]에서 확인할 수 있다.

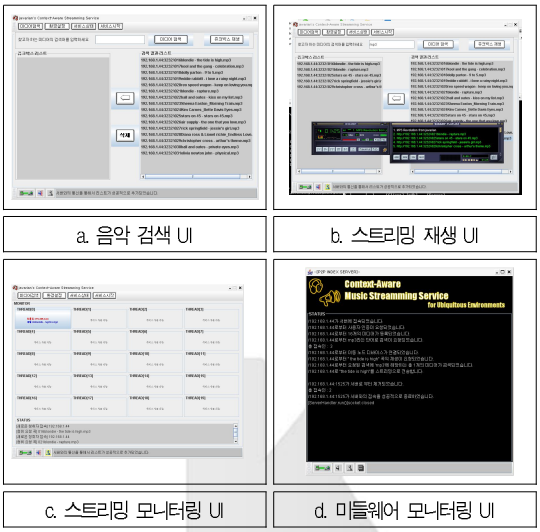

그림 11. 서비스 구현 화면 및 결과
Fig 11. Implemented Service UI and Results

제안된 시나리오는 이동 노드의 휴대용 디바이스에 서비스에 관련된 상황 정보를 분산 관리함으로서 다른 서비스 지역으로 이동 노드가 움직이는 경우에도 연속적인 서비스가 가능함을 알 수 있다. 또한 해당 서비스 지역을 담당하는 애플리케이션에 서비스를 위한 미디어가 존재하지 않더라도, 다른 지역의 미들웨어를 통해서 해당 미디어를 스트리밍으로 서비스 하는 것을 알 수 있다.

V. 결론 및 향후 연구

본 논문에서는 분산된 유비쿼터스 환경에서 상황 인식 기반의 미들웨어를 제안하고 이를 이용하여 응용 시스템을 구현하였으며 연구의 특징을 간략하게 요약하면 다음과 같다. 첫째, 제안된 상황 인식 미들웨어는 이동 노드 디바이스에 서비스에 관련된 상황 정보를 분산 관리함으로서 노드가 서로 다른 서비스 지역으로 이동하는 경우 미들웨어 간에 불필요한 정보 교환을 최소화 하였다. 둘째, 컨텐츠가 분산된 환경에서도 서비스를 지원하기 위해서 컨텐츠를 미들웨어간에 공유할 수 있는 구조를 제시하였고 멀티미디어 음악 컨텐츠를 파일이 아닌 스트리밍으로 공유함으로서 미들웨어간 불필요한 컨텐츠의 다운로드를 최소화 하였다. 셋째, 상황 정보의 획득 및 서비스의 추론을 가능하게 하는 상황 정보 스크립트를 제시하여 서비스 개발에서 하드 코딩을 최대한 배제하였으며 서비스의 추가 및 확장을 용이하게 하였다. 이는 기존 연구에 비해서 다음과 같은 이점을 제공한다. 첫째, 서비스를 위한 상황 정보를 분산 관리함으로서 미들웨어간 통신에서 발생하는 불필요한 과정을 배제하고 사용자의 자율성을 보장한다. 둘째, 사용자의 휴대 디바이스를 이용하여 이동 노드가 다른 서비스 지역으로 이동하는 경우에도 연속적인 서비스가 가능하다. 셋째, 기존 멀티미디어 서비스를 위한 미디어를 분산된 환경에서 공유할 수 있는 환경을 제공함으로서 보다 더 많은 미디어의 확보와 서비스 애플리케이션의 장애에도 유연하게 대처할 수 있다.

향후 연구로는 첫째, 보다 더 다양한 서비스를 적용함으로서 제시된 상황 인식 스크립트의 유용성을 평가하고 다양한 서비스 및 상황 정보를 수용할 수 있는 연구가 요구된다. 둘째, 상황 정보를 수집할 수 있는 다양한 센서들과 사용자의 입력을 멀티 모달 관점에서 수용할 수 있도록 상황 위젯에 관한 연구가 요구된다. 셋째, 제공되는 컨텐츠를 MP3 파일 포맷에만 국한하지 않고 미들웨어간에 다양한 형태의 멀티미디어 컨텐츠를 실시간으로 변환하여 이를 스트리밍으로 전송할 수 있는 연구가 이루어질 필요가 있다.

참고문헌

[1] Marc Weiser. “The computer for the 21st century”, Scientific American, pp.94-104, 1991.
[2] G. Banavar, A. Bernstein, "Issues and Software infrastructure and design challenges for ubiquitous computing applications",

Communication of ACM, 2002
[3] 브리테니어 사전, http://www.britannica.com
[4] Bill Schilit, Norman Adams, and Roy Want. “Context-aware computing applications.”, In IEEE Workshop on Mobile Computing Systems and Application, 1994.
[5] Anind K. Dey and Gregory D. Abowd, “Towards a Better Understanding of context and context -awareness”, Georgia Institute of Technology Colleage of Computing, June 1999.
[6] 장세이, 우운택, “유비쿼터스 컴퓨팅 환경을 위한 센싱 기술과 컨텍스트-인식 기술의 연구 동향”, 정보과학회지 제 21권 제 5호, pp. 18-28, 5, 2003.
[7] Jason Pascoe. "Adding generic contextual capabilities to wearable computers", In Proceedings of the 2nd International Symposium on Wearable Computers, Pittsburgh, Pennsylvania, IEEE Computer Society Press, 1998
[8] Daniel Salber and Gregory D. Abowd "The Design and User of a Generic Context Server", Technical Report , Microsoft Research, 1998.
[9] Christopher K. Hess and Roy H. Campbell, "A Context File System for Ubiquitous Computing Environments", Technical Report, University of Illinois at Urbana-Champaign, July 2002.
[10] David Garlan, Dan Siewiorek, Asim Smailagic, and Peter Steenkiste, "Project Aura : Towards Distraction - Free Pervasive Computing Environm ents", IEEE Pervasive Computing, special is on Intergrated Pervasive Computing Environments, Volumn 1. Number 2, pp22-31, April-June 2002.
[11] 장세이, 우운택, “유비 쿼터스 컴퓨팅 환경을 위한 컨텍스트 기반 애플리케이션 구조”, 한국 정보과학회 HCI 논문집, 제 2권, pp. 346-351, 2003.
[12] Nick Ryan, "ConteXtML : Exchanging Contextual information between a Mobile Client and Field Note Server", "http://www.cs.kent.ac.uk /projects/mobicomp/" ,1999
[13] Held, A., Buchholz, S,, and Schill, A. "Modelling of context information for pervasive computing applications.", In Proceedings of SCI, 2002
[14] Henricksen, K., Indulska, J., "A Software Engine ering Framework for Context-Aware Pervasive Computing", Proc. of the 2nd IEEE International Conference on Pervasive Computing and Communication", pp. 77-86, 2004
[15] Schmidt, A., and Laerhoven, K. V., "How to build Smart Applicances", IEEE Personal Communications, 2001
[16] Akman, V., and Surav, M, "The use of situation theory in context modeling", Computational Intelligence 13, pp 427-438, 1997
[17] Schollmeier, R., "A Definition of Peer to Peer Networking for the classification of Peer to Peer Architectures and Application", Proceedings of the IEEE 2001 International Conference on Peer to Peer Computing
[18] Milojicic, D. et al. "Peer to Peer Computing", HP Labs Technical Report HPL-2002-57, 2002
[19] 오동열, 오해석, “유비쿼터스 환경에서의 컨텍스트 인식을 위한 자생적 컨텍스트 모델과 서비스의 설계,” 한국 통신 학회 Vol. 24, pp 266-275, 2005
[20] 이재규 외 5인, “전문가 시스템 원리와 개발”, 법영사, pp 62-75, 2000

저 자 소 개

김 은 영

1987년 : 숙명여자대학교 전산학과
1993년 : 숙명여자대학교 전산학과
2001년 : 숭실대학교 컴퓨터 공학박사
현 안산공과대학 디지털 미디어과 교수
관심분야 : 인터넷표준, 멀티미디어통신

오 동 열

1999년 : 경희대학교 컴퓨터 학사
2002년 : 숭실대학교 컴퓨터 석사
2004년 : 숭실대학교 컴퓨터 박사수료
관심분야 : 유비쿼터스 컴퓨팅, P2P, 멀티미디어, 음악 검색 시스템