

상황 인식 모바일 멀티미디어 응용을 위한 미들웨어

오선진* · 배인한**

1. 서 론

최근의 PDA(personal digital assistants)나 모바일 폰(mobile phone)은 다양한 새로운 응용들을 개발할 수 있는 뛰어난 기능들을 제공한다. 최근에 상황인식 모바일 멀티미디어는 연구자들의 주의를 매료시켰던 응용영역이다. 이 응용영역은 사용자들이 모바일 장치로부터 멀티미디어 콘텐츠와 상황 인식 서비스를 액세스할 때 큰 비즈니스 잠재력을 제공한다.

그러나 모바일 장치들은 제한된 대역폭, 작은 스크린 크기, 낮은 컴퓨팅 능력을 갖는다. 따라서 멀티미디어 콘텐츠는 장치의 제약에 맞게 적응되어야 한다.

이동성과 상황인식은 더 많은 도전을 요구한다. 응용은 인식된 상황과 같이 스스로 변화하는 환경에 적응해야만 한다. 이동성과 상황인식과 관련된 태스크를 실행하고 멀티미디어 지원을 제공하는 중간 소프트웨어 계층은 발생하는 문제를 해결할 수 있다. 그것은 응용들의 증가하는 복잡성을 피하고 개발자로 하여금 응용위주 태스크에

집중할 수 있도록 도와준다.

분산 컴퓨팅 분야에서 CORBA[1]나 J2EE[2]와 같은 다수의 미들웨어 솔루션들이 개발되었다. 그러나 이러한 기술들은 고정된 인프라 구조를 가진 네트워크에서 사용되도록 의도되었고 장치가 많은 가용 자원을 가질 것을 요구한다. 따라서 그것들은 모바일 컴퓨팅에 적합하지 않다. 더욱이 그것들은 멀티미디어를 지원하지 않는다.

서비스 위주 상황인식 미들웨어(SCAM, Service-oriented Context-Aware Middleware) 프로젝트[3]는 상황인식 모바일 서비스를 위한 구조를 제공한다. 이것은 상황 관련 기능에 집중하였고, 응용의 이동성을 지원하기 위한 서비스는 거의 없다. 또한 멀티미디어 응용을 위한 기능은 지원하지 않는다. CAMPUS 프로젝트[4]는 상황인식 응용을 위한 미들웨어를 제안한다. 이것의 장점은 퍼지집합 이론에 기초한 상황모델에 있다. 그러나 미들웨어는 모바일이나 멀티미디어 응용에 대한 지원이 없기 때문에 단지 고정된 장치에 적합하다.

CORTEX 구조[5]는 상황인식 분산 응용을 위한 미들웨어이다. 이것은 메시징, 서비스 발견, 자원관리 기능을 가진다. 시스템은 네트워크 자원을 제어하는 기능이 없고 응용에 대해 적응적 UI를 제공하지도 않는다. 더구나, 멀티미디어 응용을

* 교신저자(Corresponding Author) : 오선진, 주소 : 충북 제천시 세명로 117(390-711), 전화 : 043) 649-1279, FAX : 043) 649-1747, E-mail : sjoh@semyung.ac.kr

* 세명대학교 정보통신학부 교수

** 대구가톨릭대학교 컴퓨터정보통신공학부 교수 (E-mail : ihbae@cu.ac.kr)

지원하지 않는다. DISCWorld 시스템[6]은 모바일 로봇 시스템에 초점을 맞추었다. 이것의 미들웨어는 통신을 위한 기능을 제공하는 다수의 구성요소들을 포함한다. DISCWorld 미들웨어의 목표는 로봇들의 집합과 제어하는 PDA들을 하나의 네트워크로 연결하는 네트워크 도전을 해결하는 것이기 때문에 미들웨어는 단지 메시징 기능에 초점을 맞춘다. 더욱이 이것은 멀티미디어 응용에 대한 지원이 없다.

분산 멀티미디어 응용을 위한 미들웨어 서비스는 Lohse[7]에 의해 제공되었는데 분산 환경에서 오디오와 비디오 렌더링을 위한 기능을 제공한다. 이것은 또한 모바일 장치의 제한된 자원에 대한 멀티미디어 콘텐츠의 적응성을 지원한다. 그러나 미들웨어 솔루션은 상황인식 응용을 위한 기능을 제공하지 않는다.

QoSDREAM 프레임워크[8]는 상황인식 멀티미디어 응용의 개발을 지원하는데 초점을 맞추고 있다. 이것은 이벤트 메시징 구성요소, 데이터 저장장치 그리고 분산 멀티미디어 서비스를 가진다. 그림 1은 QoSDream 구조의 주요 4가지 구성요소를 보여준다. 프레임워크는 미디어 스트리밍을 지원하나 상황은 단지 위치 정보에 의해 주어진다. 프레임워크는 응용의 이동성을 지원하지 않는다. 예를 들어, 응용은 한 호스트에서 다른 호스트로 이동할 수 없다. 게다가, 네트워크 연결은 고정되었다고 가정한다. 모바일 장치들은 단지 위치 센서로 사용된다.

Lau와 Lum[9]은 멀티미디어 콘텐츠 적응을 위한 결정 엔진을 개발하였다. 그것들은 PDA를 통해 콘텐츠에 접근할 수 있는 리치 멀티미디어 요소들을 위한 실시간 처리 시스템을 제공한다. 그림 2는 콘텐츠 적응의 전체 구조를 보여준다. 어떤 사용자가 중간 프락시 서버를 거쳐 콘텐츠 제공자

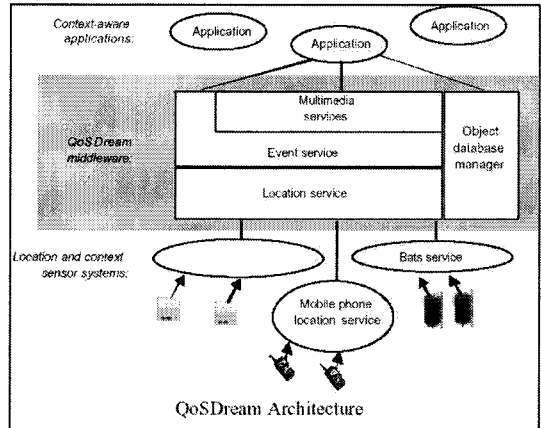


그림 1. QoSDream 구조

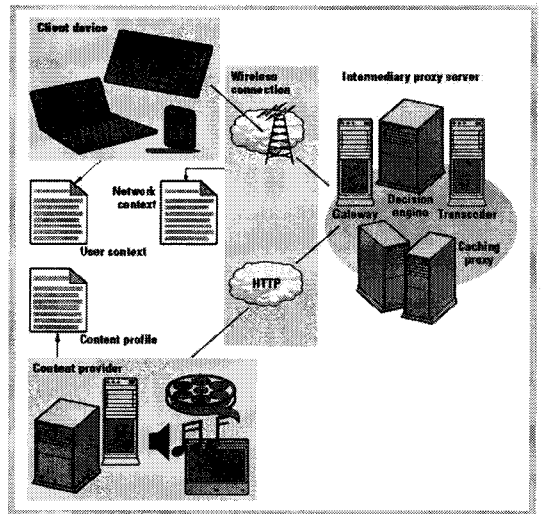


그림 2. 콘텐츠 적응의 전체 구조

에게 어떤 요청을 제출하기 위하여 모바일 장치를 사용할 때 콘텐츠 적응 프로세스는 시작된다. 시스템이 사용자를 확인한 다음, 시스템은 프락시 서버에 있는 결정 엔진에 상황정보를 입력한다. 결정 엔진은 현재 클라이언트 장치와 네트워크 특성으로 표현 가능한 최적 버전을 계산하는 알고리즘을 실행한다. 결정 엔진은 그 결과를 요구되는 콘텐츠 버전을 생성하는 transcoder에게 전송한다. 렌더링을 위해 중간 프락시는 적응된 콘텐츠

츠를 목표 장치에게 전송한다.

Lemlouma와 Layaida[10]은 모바일 장치를 위한 콘텐츠 전달의 적응성을 위한 프레임워크를 제공한다. 그러나 이러한 솔루션들은 비록 프로젝트가 제한된 자원을 가지고 모바일 장치에 멀티미디어 전달에 초점을 맞추었다고 하더라도 모바일 상황인식 응용의 개발에 대한 기능을 제공하지 않는다.

이 짧은 조사로부터 모바일 응용에 대한 상황인식 솔루션과 분산 멀티미디어 응용에 대한 솔루션 사이에는 차이가 있음을 알 수 있다. 첫 번째는 상황정보를 처리하고 하드웨어의 이질성을 감춤으로써 모바일 응용을 지원한다. 그것들은 자원이 지극히 제한된 모바일 환경에서 실행되기에 충분히 가벼운 솔루션을 제공한다. 분산 멀티미디어 응용에 대한 솔루션들은 모바일 장치가 제한된 자원을 가지기 때문에 멀티미디어 콘텐츠 적응성에 대한 기능을 제공한다고 가정한다. 게다가 그것들은 분산 환경에서 동작할 수 있도록 하는 네트워크 투명성을 제공함으로써 모바일 응용을 지원한다. 하지만 이러한 미들웨어 솔루션들은 상황 관련 태스크들을 실행할 수 없다. 상황인식, 분산 그리고 멀티미디어 응용을 지원하는 유일한 프레임워크는 QoSDREAM이다. 그러나 이것은 위에 언급했던 다수의 제약을 가진다.

본 연구에서 우리는 상황인식과 모바일 멀티미디어 응용의 개발을 위한 미들웨어 솔루션을 제안한다. 미들웨어는 상황인식과 이동성을 다루는 대부분의 태스크들을 실행할 수 있고 멀티미디어 지원에 초점을 맞춘다. 이것은 서비스와 구성요소를 찾고, 응용들 사이에 비동기적 메시징을 제공하며, 다양한 소스들로부터 수집된 상황 데이터를 저장 및 처리하고, 지역과 네트워크 자원 등을 관리한다. 멀티미디어에 대한 지원은 미디어 콘텐츠

를 캡처, 렌더링, 처리, 적응, 그리고 저장하는 것이다. 미들웨어는 미디어 경고라 부르는 새로운 이벤트 종류를 소개하고 미디어 메시징 응용의 구축을 용이하게 한다. 미디어 경고는 미리 정해 놓은 상황이 인식되었을 때 생성되는 멀티미디어 메시지이다.

본 논문은 다음과 같이 구성되었다. 2장에서는 제안하는 상황모델과 상황인식 미들웨어를 설명하였고, 3장에서 제안하는 상황인식 모바일 멀티미디어 미들웨어인 M4CMM(Middleware for Context-Aware Mobile Multimedia)의 기능과 구조를 기술한다. 프로토타입 구현에 대한 고찰은 4장에서 설명하고, 그리고 마지막으로, 5장에서 결론과 향후 연구내용을 기술한다.

2. 미들웨어 구조 요구사항

유비쿼터스 모바일 응용의 상황인식은 최근 많은 연구자들로부터 흥미롭고 기대되는 연구 주제로 주목을 받고 있다. 상황인식은 거의 대부분 유비쿼터스 응용들이 동작에서 상황 정보를 사용하기 때문에 모바일 시스템에 핵심적인 특징이다. 우리는 “엔티티의 상황을 특징짓기 위하여 사용될 수 있는 어떤 정보”로 상황을 정의한 Dey의 정의를 사용한다. 이때 엔티티는 사람, 장소 또는 사용자와 응용 자신을 포함한 사용자와 하나의 응용 사이의 상호작용과 관련 있는 객체이다[11]. 상황 정보의 예로 사용자의 위치, 시간, 사용자의 프로파일, 모바일 장치의 지역 자원들, 가용 서비스 등이다.

상황인식은 태스크를 수행할 때 상황 정보를 사용하기 위한 시스템으로 특성을 기술한다. 이 논문에서는 상황인식의 태스크 지향 정의를 사용한다. 즉, “상황인식 시스템은 사용자에게 관련 정

보나 서비스들을 제공하기 위하여 상황을 사용한다. 여기서 관련성은 사용자의 태스크에 의존한다.” 다시 말해, 주된 중요성은 태스크와 관련 있는 상황에 둔다[11].

상황인식은 센서로부터 데이터 획득 수행, 상황 인식, 상황이 실제로 사용되기 전에 완성을 필요로 하는 다른 태스크들을 포함한다. 응용들을 위하여 데이터 획득 태스크와 상황 처리 태스크를 위임하는 것은 그 응용들이 재사용되는 것을 거의 불가능하게 만든다. 이러한 문제에 대한 한 가지 솔루션은 응용으로부터 그 태스크들을 분리하여 요구되는 기능을 하위 계층으로 이동시키는 것이다. 응용의 요구를 서비스하는 그러한 계층은 미들웨어라 불리는 특별한 계층을 일반적으로 형성한다. 우리는 미들웨어에 대한 정의를 “프로그래밍 추상화를 제공할 뿐만 아니라 하부의 네트워크, 하드웨어, 운영체제, 그리고 프로그래밍 언어의 이질성을 감추는 소프트웨어 계층”이라고 한 Coullioris의 정의를 따른다[12]. 이 미들웨어 계층은 상황정보를 측정하는 장비의 이질성과 분산 특징을 감춘다. 상황인식 미들웨어는 응용의 상황 요구를 서비스한다. 그러한 계층의 기능은 다음장에서 상세히 다루기로 한다.

상황인식 미들웨어는 다음의 상황 위주 기능을 응용에 제공해야 한다.

- 다양한 센서 장치의 지원
- 데이터가 서로 다른 소스로부터 발생 때문에 상황 정보의 분산 특징을 지원
- 응용의 투명한 해석과 상황 데이터의 추상화 제공
- 상황 저장장치의 유지
- 상황 데이터 흐름의 제어

이동성은 미들웨어에 다수의 제약을 가져온

다.[14]

- 대역폭은 낮고 호스트들은 네트워크 분할이나 빈약한 유효 송수신 범위로 인해 도달할 수 없다.
- 장치의 메모리 용량이나 CPU 파워와 같은 지역 자원들은 지극히 제한적이다.
- 시스템 구성요소들 간의 통신은 비동기식이다.
- 실행환경은 동적이다.

동적 실행환경은 계속해서 변화하는 분산 모바일 시스템을 일컫는다. 호스트들은 네트워크에 정적으로 연결되는 것이 아니고 그러한 시스템의 환경설정은 예측하기 힘들다. 그러므로 미들웨어는 이와 같이 계속해서 변화하는 시스템에서 호스트와 서비스의 탐색을 지원해야만 한다.

더구나, 멀티미디어 응용들은 통신과 연산을 위한 요구사항을 설정한다. 멀티미디어 알고리즘은 많은 대역폭과 처리 능력을 요구한다. 따라서 미들웨어는 다음 사항들을 처리할 수 있어야 한다.

- 가용 대역폭 사용: 미들웨어는 온갖 다른 가용 연결을 사용해야 하고, 어떤 시점에 요구사항을 가장 잘 수행하는 연결로 전환할 수 있어야 한다.
- 연산의 장소 제어: 제한된 자원에 대처하기 위하여 미들웨어는 상황에 기초한 연산이 수행되어야 할 장소를 결정할 필요가 있다.

더욱이, 미들웨어는 비디오카메라나 마이크와 같은 다양한 멀티미디어 장치들을 지원해야만 한다. 마지막으로, 시스템을 적응적으로 만드는 미들웨어의 요구사항은 다음과 같다.[15]

- 시스템 전역 단계에서 적응성 호출
- 시스템 전역 적응성 정책에 대한 지원
- 장치와 미들웨어 사이에 공통 인터페이스 제공

3. M4CAMM

M4CAMM은 상황인식, 소프트웨어 구성요소의 이동성, 멀티미디어 응용과 적응성으로 미들웨어 구조를 설정하는데 필요한 요구사항을 수행하기 위하여 개발되었다.

M4CAMM은 응용 계층 아래 그리고 기존 기술 계층 위에 위치한다. M4CAMM에 의해 제공되는 기능은 구성요소의 집합으로 나누어져 있다. 각 구성요소는 다른 구성요소들이나 응용에 특별한 서비스를 제공한다. M4CAMM은 그러한 구성요소로 연결 관리, 서비스 발견, 메시징, 구성요소 관리, 사용자 인터페이스, 상황, 미디어와 상황기반 저장장치 등을 포함한다. 그림 3은 M4CAMM 구조를 보여준다.

M4CAMM은 상황인식에 의해 설정된 모든 요구사항을 수행한다. 이것은 위치 데이터를 수집하거나 비디오 스트림을 기록하기 위해 서로 다른 장치를 지원한다. 데이터는 물리적으로 다른 네트워크에 연결된 장치에 위치하는 분산 소프트웨어

구성요소로부터 발생되어진다. 상황 추상화가 중간의 구성요소들을 통해 투명하게 수행되어서 응용들은 러프 측정값 대신 이미 추상화 된 데이터를 수신한다. 미들웨어는 파일을 포함한 어떠한 상황 관련 정보를 저장하는 완전한 상황 저장 기능을 제공한다.

M4CAMM 시스템의 이동성은 경량 구성요소 프레임워크에 의해 지원된다. 구성요소는 충분한 자원을 가진 장치에서 실행된다. 연결 관리 구성요소는 대역폭과 전방적인 트래픽을 제어한다. 메시징 구성요소, 서비스 발견과 구성요소 관리는 모바일 환경에서 응용들이 작동할 수 있게 필요한 수준의 투명성을 제공한다.

멀티미디어 응용에 의해 설정된 요구사항은 역시 구성요소 프레임워크와 제공되는 기본 서비스로 역시 실행된다.

적응성은 멀티미디어 응용의 구축을 쉽게 해주는 멀티미디어 단계에서 수행된다. 각 응용은 응용의 실행 중에 고려되어야 하는 자원이나 네트워크 요구사항을 정의한다. 게다가, 응용들은 연결과 트래픽을 제한하는 네트워크 정책을 정의한다. 마지막으로, 미들웨어 구성요소들은 하위 계층에 대한 공통 인터페이스를 갖는다. 이것이 M4CAMM에 의해 지원되는 하드웨어의 확장을 용이하게 한다.

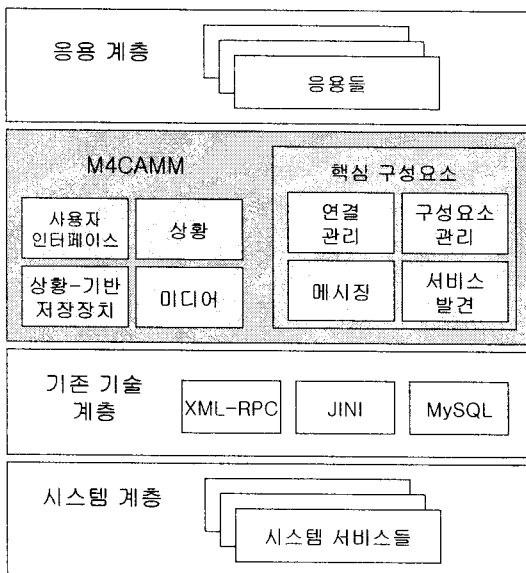


그림 3. M4CAMM의 구조

3.1 M4CAMM 구성요소

M4CAMM 미들웨어 구성요소는 명백하게 정의된 인터페이스를 가진 원자적 소프트웨어 개체이다. 구현의 내부 구조나 복잡성은 구성요소의 인터페이스 뒤에 숨겨져 있다. 모든 구성요소들은 프락시나 스텝 구성요소로 표현된다. 프락시 접근 방법은 환경에서 분산 투명성을 성취하는데 도움을 준다. 따라서 구성요소들은 지역이든 원격이든 그것들의 실제 위치에 상관없이 같은 방법으로

서로 통신한다. 구성요소는 시스템이 시작되거나 응용이 실행되는 프라이 상태에도 탐색되거나 시작될 수 있다. 이것은 그 환경에 미들웨어 적용을 용이하게 한다.

M4CMM 핵심은 구성요소 관리, 메시징, 그리고 서비스 발견 구성요소를 포함한다. 이들은 분산 환경에서 다른 구성요소나 응용을 지원하기 위해 필요한 완전한 서비스를 제공한다.

구성요소 관리는 구성요소들과 그것들의 스텝을 제어한다. 이것은 시동 시 구성요소를 시작하고, 프락시를 초기화하고 생성하며, 서로 구성요소 액세스를 얻기 위하여 요청을 처리한다. 구성요소로 어떤 핸들이 요청되었을 때 구성요소 관리는 실제 구성요소로의 참조 대신 프락시 구성요소로의 참조를 반환한다. 만약 필요한 구성요소가 지역 장치에서 비가용이면 구성요소 관리는 명시된 장소로부터 인터넷으로 그것을 적재한다. 구성요소 관리는 만약 장치가 구성요소를 실행시키기에 충분한 지역 자원(즉 메모리나 CPU 파워가 제한적)을 가지고 있지 않으면 원격 호스트로 구성요소 이동을 결정할 수 있다. 이것은 미디어 구성요소가 멀티미디어 콘텐츠의 처리나 적용을 수행할 필요가 있을 때 요구되어진다.

M4CMM에서 통신은 메시징 구성요소의 책임이다. 이것은 채널을 생성해서 비동기식 통신, 채널 관련 연산, 그리고 환경에서 원격 프로시저 호출을 지원하기 위한 기능을 제공한다. 채널은 네트워크에 연결된 장치들이 위치한 서로 다른 물리적 응용들 간의 논리적 링크이다. 메시징 구성요소는 구성요소 관리와 함께 위치 투명성을 제공한다.

이벤트 통지 메커니즘은 출판과 구독 모델에 기반 한다. 구성요소나 이벤트를 소비하는 응용들은 이벤트 생산 구성요소로부터 특정 타입의 이벤

트를 구독할 수 있다.

동적 분산 환경에서 서비스나 호스트의 가용성은 어떤 지역에서의 네트워크 단절이나 빈약한 유효 송수신 범위에 기인하여 계속해서 변화한다. 따라서 중앙 집중식 서버나 디폴트 서버가 존재하지 않기 때문에 서비스 검색은 중요한 문제가 되었다. 위치 투명성은 서비스 발견 기능으로 달성된다. 이 구성요소는 서비스나 가용 구성요소를 위치시킨다. 클라이언트는 다른 구성요소들에 의해 제공되는 서비스를 찾기 위해 서비스 발견을 요청한다. 서비스 발견은 맞는 항목을 찾아서 클라이언트에 반환하고, 클라이언트는 요구한 구성요소를 목록으로부터 선택하고 선택된 구성요소에 대한 참조를 얻기 위해 구성요소 관리에 요청한다.

연결 관리와 미디어 구성요소는 다음 장에서 좀 더 자세하게 설명하기로 한다. 나머지 미들웨어 구성요소들은 장치에서 실행하는 것이 선택이기는 하지만 상황 인식 모바일 응용을 지원하기 위해서는 필수적이다. 서비스를 가능하게 하는 구성요소들은 상황 기반 저장장치, 상황, 그리고 사용자 인터페이스 구성요소 들이다.

M4CMM에서 저장장치 기능은 상황 기반 저장장치에 의해 제공된다. 이것은 주로 요청에 의해 데이터를 저장하고 탐색한다. 구성요소는 데이터의 동기화를 다루고 데이터베이스의 내용이 변경되었을 때 클라이언트에 경고한다. 상황 기반 저장장치는 사용자의 데이터에 프라이버시와 접근 권한을 정의할 책임이 있다.

상황 구성요소는 상황 정보의 제공: 즉 상황 전달, 처리와 서비스에 책임이 있다. 상황 구성요소는 상황 센서를 위한 래퍼 역할을 한다. 더욱이, 상황 데이터가 분산된 소스로부터 발생될 때, 그것은 서버 구성요소로 사용된다. 상황 기반 저장

장치와 함께 상황 구성요소는 상황 히스토리 유지 기능을 제공한다.

사용자 인터페이스(UI) 구성요소는 모바일 장치에 대한 UI 프런트 엔드 역할을 하며 응용의 설계와 구현을 지원한다. UI는 추상 UI, 플러그인 UI(다운로드 가능한 자바 코드), 그리고 웹 기반 UI(HTML) 등 세 가지 다른 기술을 사용한다. UI 소프트웨어 개발자는 요구에 가장 잘 맞는 기술을 선택할 수 있지만 UI는 응용 논리로부터 분리되어야 한다. 추상 UI에 대해서 구성요소는 UI 구현을 구축하거나 수정할 뿐만 아니라 응용과 UI 구현 사이에 메시지나 이벤트를 교환할 수 있는 단순한 인터페이스를 제공한다. 자바 플러그인 UI가 사용되는 경우 단지 메시지 교환만이 지원된다. 메시지는 M4CMM 메시지 메커니즘을 사용하여 전달된다. 웹 기반 UI는 목표 장치의 웹 브라우저에서 시작되며 응용과 UI간 통신은 웹 기술(웹 서버, CGI 스크립트, 자바 서버 페이지 등)을 사용하여 분리하여 구축되어야 한다.

3.2 연결

연결 관리 구성요소는 모바일 장치의 연결을 제어하고 감시한다. 구성요소는 실제 연결의 자세한 부분을 채널 뒤로 감춘다. 채널은 응용 계층에 대한 네트워크 투명성을 제공하며 계속되는 통신을 방해하지 않고 네트워크 기능의 동적 적응을 허용한다. 즉, 구성요소는 채널을 사용하며 그것들은 어떤 연결이 데이터를 전송하는데 사용되는지 알지 못한다. 심지어 연결은 구성요소에 알려지지 않고 통신 중간에 교체될 수 있다. 현재 연결 관리는 TCP/UDP, HTTP, 멀티캐스트와 TCP 수신 채널을 지원한다.

클라이언트는 예를 들어 연결의 대역폭과 가용성과 같은 채널의 상태에 대한 실시간 정보를 요

청할 수 있다. 구성요소에 직접 요청뿐만 아니라 클라이언트는 이 정보를 구독하고 채널 상태가 변화했을 때 통지를 얻을 수 있다.

연결 관리 구성요소는 정의된 정책에 따라 플라이 상태에서 연결을 교환함으로써 채널을 제어한다. 채널 정책은 플라이 상태에서 사용자나 응용에 의해 공통 인터페이스를 통해 변경될 수 있다. 어떠한 미들웨어 솔루션도 채널 단계에서 연결을 제어할 수 있고, 전반적인 트래픽이나 대역폭을 제한할 수 있으며, 클라이언트가 그것들을 사용하는 동안 연결을 동적으로 교환할 수 있다는 것을 알지 못한다.

연결 관리는 연결 제어기, 정책 관리자, 그리고 연결 모니터 등 세 가지의 하위 구성요소로 구성된다. 연결 관리 구성요소의 구조는 그림 4에서 보여준다.

연결 감시자는 채널의 상황 정보를 감시한다. 이것은 RTT(round trip time)와 같은 채널의 QoS 파라미터뿐만 아니라 지역과 원격 호스트의 IP 주소와 같은 네트워크 특징을 수집한다. 정책 관리자는 채널에 대한 기준과 규칙을 정의할 책임이 있다. 정책 관리자는 규칙을 특정 채널 또는

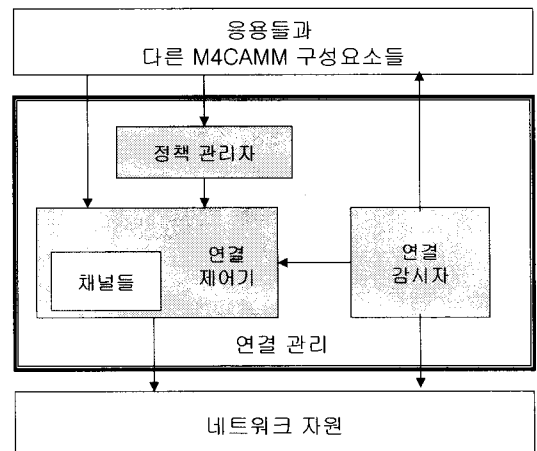


그림 4. 연결 관리 구성요소의 구조

장치에 바인딩 하는 것을 허용한다. 연결 제어기는 정책 관리자와 연결 감시자 하부 구성요소 모두를 관리할 수 있다. 게다가 연결 제어기는 클라이언트가 요구하는 대로 채널을 관리한다.

3.3 미디어 구성요소

미디어 구성요소는 다양한 장치들이 접촉하는 고유한 미디어 기능의 휴대성과 확장성을 용이하게 한다. 목표는 미디어 응용 개발자들에게 이미지, 오디오 그리고 비디오의 캡처 기능을 제공하는 것이다. 미디어는 지역으로나 원격으로 처리되거나 저장될 수 있다. 미디어 콘텐츠는 다양한 장치에 대해 적용될 수 있다. 이러한 적용은 지역에서 또는 원격 장치에서 수행될 수 있다.

미디어 구성요소는 핵심 구성요소와 선택 구성요소로 나뉜다. 핵심 미디어 구성요소는 정적이고 미디어 기능이 사용될 때 항상 존재한다. 선택적 동적 구성요소들은 요청에 의해 적재되고, 장치에서의 응용의 요구와 가용 자원에 의존한다. 그래서 만약 장치가 충분한 자원을 가지고 있지 않다면 동적 미디어 구성요소는 서버 측에서 원격으로 시작된다. 동적 구성요소는 단지 서버 측에서만 실행되도록 제약을 둘 수 있다. 그림 5는 미디어 구성요소의 구조를 보여준다.

핵심 구성요소는 미디어 객체의 연속 간행(serialization)을 생성하고 수행하고, 관련된 메타 데이터를 처리한다. 미디어 객체는 모든 미디어 부요소의 데이터, 미디어 요소의 관계, 그리고 각 부요소의 메타 데이터를 포함한다. 구성요소는 공간적, 시간적, 그리고 향해 관계 등 미디어 요소들 사이에서의 세 가지 타입의 관계를 지원한다. 공간적 관계는 요소들이 공간적으로 어떻게 서로 관계되는가를 정의한다. 시간적 관계는 미디어 요소들이 시간적으로 어떻게 관계되는가를 정의한

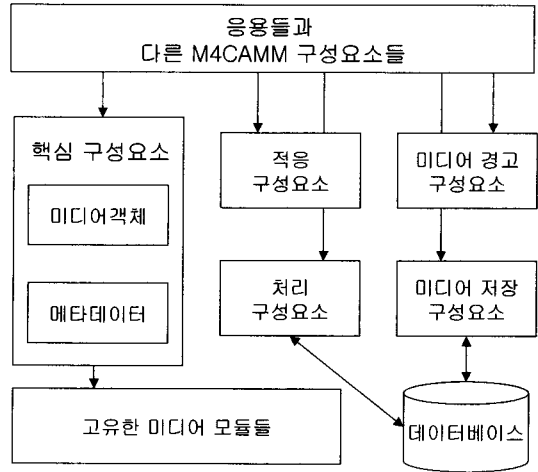


그림 5. 미디어 구성요소의 구조

다. 예를 들어, 요소 B는 요소 A 전에 재생되어야 한다. 향해 관계는 요소들 각각의 링크를 정의한다. 예를 들어, 요소 A는 요소 B를 재생하기 위한 미디어 재생을 요구하는 링크를 포함할 수 있다. 공간적, 시간적, 그리고 향해 관계는 SMIL2.0[16] 표준에 따라 제안되어 진다.

나머지 구성요소들은 동적이고 미디어 요소를 저장하고 탐색하고 필요한 경우 처리할 수 있는 기능을 제공한다. 구성요소는 또한 미디어 객체를 캡처하고 렌더링 할 수 있다. 더구나, 미디어 경고 구성요소는 미디어 객체의 푸시 타입 전달 메커니즘을 지원하는 인터페이스를 제공한다. 미디어 저장 구성요소는 데이터베이스와 미디어 구성요소 사이의 인터페이스이다. 이것은 인덱스를 생성하기 위해 메타 데이터를 사용하고 미디어 객체에 대한 데이터베이스를 질의하는 유연한 방법을 제공한다.

응용은 서로 다른 장치에서 재생하기에 적당하도록 미디어 객체의 적용을 요구한다. 이런 기능은 미디어 구성요소에 의해 지원된다. 클라이언트는 단말기의 미디어 객체와 정보를 미디어 구성요소에게 전달한다. 동적 구성요소는 장치의 기능을

데이터베이스로부터 탐색하고 이러한 제약에 따라 미디어 객체를 적응한다. 적응은 지역이나 원격으로(즉, 서버 측에서) 수행되어진다.

캡처와 렌더링 기능은 미디어 객체의 생성을 쉽게 하고 재생을 지원하기 위하여 미디어 구성요소에 의해 제공된다. 실제 캡처나 렌더링이 시작되기 전 미디어 구성요소는 목적 장치가 미디어를 재생하거나 캡처하는데 충분한 자원을 가지고 있는지를 검사한다. M4CMM 미디어 구성요소 대신 미디어 객체의 캡처나 렌더링을 실행하기 위해 외부의 서드파티 구성요소를 사용할 수 있다.

미디어 경고는 미디어 콘텐츠와 연관된 이벤트이다. 그것들은 예를 들어 카메라가 동작을 감지하거나 마이크가 연설을 감지하는 것과 같은 어떤 미디어 관련 상황이 발생했을 때 생성된다. 구성요소와 응용들은 미디어 경고 구성요소로부터 관심 있는 경고들을 수신할 수 있다. 게다가 미디어 경고 구성요소는 요청에 의해 다른 구성요소와 응용 사이에서 미디어 경고를 전달한다.

미디어 구성요소는 또한 응용에 의해 미디어 경고의 호출을 지원한다. 따라서 각 응용은 자신만의 경고를 호출할 수 있다. 이것은 미디어 메시지를 전달하기 위한 유연한 인프라 구조를 생성하는 것을 허용한다. 미디어 경고는 비동기식으로 전달된다. 만약 클라이언트가 미디어 경고를 수신할 수 없으면 큐에 저장되었다가 클라이언트가 가용일 때 전달된다.

4. 프로토타입 고찰

M4CMM 구조를 증명하기 위해 두 가지 프로토타입이 고려되었다. 그 프로토타입의 기능은 다음과 같다. 구성요소는 동적으로 탐색되고, 다운로드되며, 시작된다. 다른 구성요소에 대한 참조

를 얻기 위한 클라이언트의 요청은 구성요소 관리에 의해 처리된다. 이것은 구성요소를 지역이나 원격 호스트 어디서든 시작할 수 있다. 필요한 호스트에서 만약 구성요소가 비가용이면 구성요소 관리는 인터넷으로부터 그것을 다운로드 할 수 있다. 구성요소의 기능과 요구사항은 XML 서술 형태로 주어진다. 그러한 서술은 구성요소의 인터페이스 이름, 유일한 ID 번호, 구성요소의 코드 URL 등과 같이 구성요소의 선택적인 또는 필수적인 특징들을 포함한다. 이 서술은 구성요소 관리와 서비스 발견 구성요소에 의해 사용된다. 메시징 구성요소는 같거나 다른 호스트에 있는 그것들의 장소에 상관없이 구성요소들 사이에서 원격 프로시저 호출을 실행할 수 있다. 메시징 구성요소에서 구현된 기능은 비동기식 메시지 송신을 포함한다. M4CMM 프로토타입에서 모든 메시지 기반 통신은 XML-RPC 프로토콜에 기반한다. XML-RPC가 경량 프로토콜이기 때문에 XML-RPC를 선택한다. XML-RPC의 단점은 객체 송신을 함수의 매개변수 전달처럼 허용하지 않고, XML 메시지에서 무기명으로 위치된 구조의 사용을 강요하는 것이다[17]. M4CMM 프로토타입에서 이동성은 네트워크 자원을 제한하기 때문에 객체 지향 XML-RPC에 의해 제공되는 모든 기능을 사용할 수는 없다.

연결 관리 구성요소는 응용의 작업을 방해하지 않고 한 타입에서 다른 타입으로 연결을 변경할 수 있다. 이러한 채널 적응은 투명하며 응용으로부터 어떠한 의사 결정 처리를 요구하지 않는다. 응용들과 클라이언트들은 단지 채널 정책을 정의하면 되고 연결 관리는 이 정책에 따라 연결을 변경한다.

서비스 발견 구성요소는 Jini와 내부의 데이터 베이스를 사용하여 서비스나 다른 구성요소를 찾

을 수 있다. 필요할 때 클라이언트는 서비스나 구성요소를 찾기 위하여 서비스 발견을 요청한다. 서비스 발견은 클라이언트에게 찾은 목록을 제공한다. 그러면 클라이언트는 목록으로부터 한 항목을 선택하여 희망 구성요소의 핸들을 얻도록 구성요소 관리에게 요청한다. 서비스 발견은 서비스의 위치 인식 탐색을 실행한다. 예를 들어, 클라이언트는 자신으로부터 가까운 지역의 가용 서비스 목록을 요청하기 위하여 서비스 발견을 제한할 수 있다.

M4CMM 환경은 서로 다른 종류의 이벤트를 지원한다. 상황 관련 이벤트는 푸시 모델로 생성되고 수집된다. 이벤트가 발생할 때 마다 그것은 상황 구성요소로 전달되고, 그것은 이 이벤트에 책임이 있는 구성요소로 전송된다. 이벤트는 출판-구독의 형태로 전달된다. 이벤트 공급자는 어떤 구성요소 이벤트 생산자로부터의 이벤트 타입에 자신의 관심을 등록해야 한다. 미들웨어는 한 구성요소로부터 다른 구성요소로 직접 전달되는 내부 이벤트를 지원한다. 내부 이벤트는 메시지 구성요소의 책임이다.

미디어 구성요소는 비디오를 캡처하고 이미지를 만들 수 있다. 오디오와 비디오 스트림의 동기화가 아직 완성되지 않았기 때문에 사운드로부터 비디오는 분리되어서 저장되고 재생된다. 미디어 구성요소는 미디어 객체를 생성하고 객체의 메타데이터와 관련된 미디어 객체들을 처리할 수 있다. 미디어 경고나 저장 기능뿐만 아니라 렌더링 기능도 구현되어진다. 미디어 객체의 적응, 상황, UI, 그리고 상황 기반 저장과 같은 다른 구성요소들은 구현되어야 한다.

M4CMM 프로토타입을 일상 학습과 비즈니스 미팅에 대해 실험한다. 첫 번째 프로토타입은 사용자의 습관을 학습할 수 있고 위치 기반 기억

메모를 제공할 수 있다. 이것은 프로필 관리자, 기억 메모(reminder), 달력, 그리고 개인 보조(personal assistance) 등 4가지 응용으로 구성된다. 달력 응용은 약속과 미팅을 저장하기 위해 사용된다. 상황 구성요소는 기억 메모를 경유해서 달력으로부터 이벤트를 얻는다. 마지막으로, 프로필 관리자는 상황 구성요소로부터 일과들을 감지한다. 개인 보조 응용은 모바일 장치에서 응용을 검색하는데 사용된다.

프로토타입 시나리오에 따르면 사용자는 자신의 전화기를 구입해서 사용을 시작한다. 사용자가 주간 미팅 동안 전화기를 매너 모드로 전환하는 습관을 학습한다. 전화기가 일과를 감지한 후에 시스템은 만약 사용자가 미팅이 있을 때 자동적으로 전화기를 매너 모드로의 전환을 제공한다. 더군다나, 프로토타입은 위치 기반 기억 메모를 지원한다. 따라서 사용자가 위치 기억 메모를 첨부하면 시스템은 장소를 인식하고 사용자에게 메시지를 통해 통지한다.

비즈니스 미팅에 대한 프로토타입은 다음의 시나리오에 기초한다. 사용자는 발표를 해야 한다. 원격으로 미팅 룸의 가용 서비스를 체크한다. 발표 자료가 생성된 후에 이 파일을 달력 응용의 이벤트에 첨부한다. 아침에 사용자는 미팅에 관한 기억 메모를 받는다. 만약 차례가 되면 미팅 룸의 가용 서비스를 탐색하고 슬라이드 프로젝트를 선택한다. 미팅과 관련된 파일들의 목록이 보여진다. 사용자는 필요한 파일을 찾아 모바일 장치를 원격 제어 기구로 사용하며 슬라이드 쇼를 시작한다. 미팅 중간에 아내가 전화로 통화를 요청한다. 아내는 남편이 미팅 중이라는 통지를 받는다. 그녀는 비디오 메시지를 남기기로 결정한다. 미팅이 끝났을 때 사용자는 비디오 메시지에 대한 통지를 받고 메시지를 확인한다.

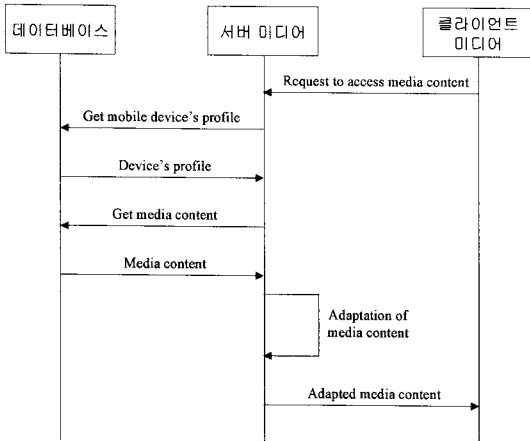


그림 6. 미디어 구성요소 시나리오

비즈니스 미팅에 대한 프로토타입은 전화번호부, 비디오 메시징, 프로젝트, 서비스 탐색기와 파일 탐색기 응용이 필요하다. 비디오 메시징 응용은 비디오 메시지를 녹화하고 보여줄 수 있다. 프로젝트 응용은 모바일 장치를 통해 실제 프로젝트를 제어하도록 지원한다. 서비스 탐색기는 사용자가 가용 서비스에 대한 서비스 발견 요청을 돕는다. 파일 탐색기 응용은 데이터베이스에 대한 인터페이스이다. 그림 6에서 미디어 구성요소에 대한 시나리오를 보여 준다.

사용자는 모바일 장치로 비디오 파일을 액세스하는 것을 시도한다. 그러나 작은 스크린 때문에 파일을 재생할 수 없다. 모바일 장치에 위치한 미디어 구성요소는 비디오 파일을 얻기 위해 서버에 위치한 서버 미디어 구성요소에 요청한다. 서버 구성요소는 데이터베이스로부터 장치 프로파일을 질의하고 난 다음 원래의 비디오를 탐색한다. 적용이 완료된 후 서버는 사용자가 그것을 재생할 수 있는 모바일 장치에 파일을 전송한다.

5. 결 론

우리는 모바일 멀티미디어 응용을 위한 상황인

식 미들웨어인 M4CMM을 제안한다. 미들웨어는 상황인식 멀티미디어 응용과 그것들의 분산 환경 모두에 중요한 특징을 제공한다. 미들웨어는 응용을 위해 어떤 기능을 제공하도록 다수의 구성요소로 나누어진다. 미들웨어는 모바일 환경에서 동작하도록 개발되어진다. 따라서 단말 장치들의 자원은 극히 제한적이다.

미들웨어는 상황인식 응용을 위한 기능을 제공하도록 만들어진다. 이것은 상황 센서를 위한 공통 인터페이스를 가지는데 이것은 새로운 센서를 시스템에 내장하는 것을 쉽게 한다. 센서들은 물리적으로 분산된 장치들에 있을 수 있다. 따라서 상황 정보는 이후 상황 전달과 처리를 실행할 중앙 집중식 상황 구성요소에 전달된다. 게다가 상황 구성요소는 상황 인식의 책임이 있다. 이러한 상황 관련 동작들은 응용을 위해 상황 처리나 인식을 투명하게 하는 미들웨어 계층에서 실행된다. 상황 기반 저장장치는 데이터 저장 기능을 제공하는 구성요소이다. 이것은 데이터 동기화에 책임이 있다. 만약 데이터베이스의 내용이 변경되면 구성요소는 클라이언트에게 통지한다. 그것들은 상황 기반 구성요소로부터 이러한 경고를 수신하기 위해 자신의 관심을 구독할 수 있다. 더욱이, 상황 구성요소와 함께 이것은 이 환경에서 상황 데이터 흐름을 제어한다.

미들웨어는 구성요소의 이동성을 지원한다. 즉, 이것은 시작하는 응용이나 구성요소의 자원 요구량에 따라 구성요소를 클라이언트 측(모바일 장치)에서 시작할 것인지 아니면 서버 측(고정 PC)에서 할 것인지를 결정한다. 실행 중에 이동하는 구성요소들의 기능은 구현되어야 한다. 분산 환경은 메시징, 서비스 발견, 그리고 구성요소 관리 구성요소에 의해 지원된다. 메시징 구성요소는 비동기식 통신과 원격 프로시저 호출의 기능을

제공한다. 서비스 발견 구성요소는 서비스, 구성 요소, 그리고 장치를 찾을 수 있다. 구성요소 관리는 그것들의 생명 주기 동안 구성요소들을 처리한다. 더군다나, 호스트의 이동성은 연결 단절이나 네트워크 분리를 초래할 수 있으며 연결 단절의 예측은 아주 어려운 일이다.

미들웨어는 멀티미디어 콘텐츠 생성과 처리를 위한 기능을 제공한다. 이것은 미디어의 저장, 탐색, 캡처, 그리고 렌더링을 할 수 있다. 미들웨어는 멀티미디어 메시징 응용의 개발을 용이하게 하는 미디어 경로의 생성과 호출에 대한 가능성을 제공한다. 게다가 이것은 멀티미디어 응용에 의해 정의될 수 있는 정책에 따라 네트워크 자원을 제어하고 채널을 교환할 수 있다. 따라서 미들웨어는 상황에 적응할 수 있다.

M4CMM 미들웨어는 자바로 완전히 개발되기 때문에 프로토타입이 모바일 장치에 적응적으로 만든다. 더구나 미들웨어의 구조는 다른 플랫폼에 시스템의 이식성을 증가시키도록 구축된다. 자바 접근 방법은 모바일 환경에서 중요한 문제인 구현된 프로토타입의 호환성을 용이하게 한다. 구현되는 미디어 서비스는 멀티미디어 생성, 전달, 적응, 그리고 처리에 대한 다양한 기능을 지원함으로써 멀티미디어 메시징 응용의 빠른 개발을 가능하게 한다.

향후 연구과제는 더욱 많은 재사용 구성요소를 갖는 구조로 향상하는 것이다. 미들웨어는 다수의 하위 계층으로 나누어지며, 미들웨어의 상위 계층은 서로 다른 운영체제를 쉽게 이식할 수 있다. 차기 프로토타입에서의 메시징 구성요소는 네트워크 분리 문제를 쉽게 해결하는 Session Initiation Protocol에 기초한다. 구성요소 관리는 응용 계층에서부터 보다 많은 책임을 가지며 그것들의 작업에 최소의 간섭으로 그것들의 실행동안

한 장치에서 다른 장치로 구성요소를 이동시킬 수 있다. 서비스 발견 구성요소는 RDF 모델에 기초한다. 이것은 행위-동기 규칙을 기술하기 위하여 이벤트와 조건의 복잡한 순서를 조합하는 쉬운 방법을 제공한다. 상황 기반 저장장치는 데이터베이스를 사용한다. 이것은 자동 데이터 갱신과 동기화를 할 수 있다. 미디어 구성요소는 사운드와 함께 비디오 메시지의 캡처와 렌더링뿐만 아니라 미디어 콘텐츠의 적응을 제공하도록 구현될 것이다.

참 고 문 헌

- [1] Object Management Group, Common Object Request Broker Architecture: Core Specification, v. 3.0.3, 2004.
- [2] Sun Microsystems, Inc., Java 2 Platform Enterprise Edition Specification 1.4, 2004.
- [3] Gu, T., et al., "A Middleware for Building Context-Aware Mobile Services," In Proceedings of IEEE Vehicular Technology Conference (VTC-Spring 2004), Milan, Italy, 2004.
- [4] Hisazumi, K., et al., "CAMPUS: A Context-Aware Middleware," The 2nd CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing, Nara Institute of Science Technology, Nara, Japan, 2003.
- [5] Duran-Limon, H., et al., "Context-Aware Middleware for Pervasive and Ad Hoc Environments," Computing Department, Lancaster University, Bailrigg, Lancaster, UK, 2000.
- [6] Hawick, K.A., James, H.A., "Middleware for Context Sensitive Mobile Applications," In Proceedings of workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing, Adelaide, Australia, pp. 133-14, 2003.

[7] Lohse M., Repplinger M., Slusallek P., "Session Sharing as Middleware Service for Distributed Applications, Interactive Multimedia on Next Generation Networks," Proceedings of First International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2003), Naples, Italy, 2003.

[8] Naguib, H., Coulouris, G., Mitchell, S., "Middleware support for context-aware multimedia applications," In Proceedings of the 3rd IFIP WG 6.1. International Working Conference on Distributed Applications and Interoperable Systems (DAIS 2001), Krakow, Poland, 2001.

[9] Lau, F., Lum, F., "A Context-Aware Decision Engine for Content Adaptation," in Proceedings of the 8th annual international conference on Mobile computing and networking, Atlanta, Georgia, USA, 2002.

[10] Lemlouma, T., Layaida, N., "Adapted Content Delivery for Different Contexts," In Proc. Of 2003 Symposium on Applications and the Internet, IEEE, 2003.

[11] Dey, A., "Providing Architectural Support for Building Context-Aware Applications," PhD thesis, Georgia Institute of Technology, 2000.

[12] Coulouris, G., Dollimore, J., Kindberg, T., "Distributed Systems Concepts and Design," Addison-Wesley, 2001.

[13] Dey, A., et al, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," in Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI-99), Pittsburgh, Pennsylvania, USA, 1999.

[14] Carpa L. et al., "Middleware for Mobile Computing," In Proceedings of the 8th Workshop on Hot Topics in Operating Systems, Elmau, Germany, 2001.

[15] Efstratiou, C., et al., "Architectural Requirements for the Effective Support of

Adaptive Mobile Applications," in Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware-2000, New York, USA, 2000.

[16] W3C, Synchronized Multimedia Integration Language, <http://www.w3.org/TR/smil20/>, 2004.

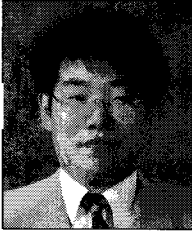
[17] Gabhart, K., Gordon, J., "Wireless Web Services with J2ME," Web Services Journal, Vol. 2, Issue 2, 2002.

[18] Gryazin, E., Burmakin, E., Tuominen, O., "Comparison of CORBA and Web Services middleware operation in wireless environments," in Proceedings of International Workshop on Mobile Computing, (IMC 2003), Rostock, Germany, 2003.



오 선 진

- 1981년 한양대학교 공과대학(공학사)
- 1987년 미국 Wayne State University 컴퓨터학과(이학사)
- 1989년 미국 University of Detroit 컴퓨터학과(이학석사)
- 1993년 미국 Oklahoma State University 컴퓨터학과(박사과정)
- 1999년 曉聖 Catholic University 전자계산학과(이학박사)
- 1994년~2000년 신림대학교 컴퓨터정보학과 교수
- 2000년~현재 세명대학교 정보통신학부 교수
- 관심분야 : 모바일 멀티미디어, 모바일 컴퓨팅, 무선 인터넷 등



배 인 한

- 1984년 경남대학교 전자계산학과(공학사)
 - 1986년 중앙대학교 대학원 전자계산학과(이학석사)
 - 1990년 중앙대학교 대학원 전자계산학과(공학박사)
 - 1996년~1997년 Department of Computer Science and Engineering, The Ohio State University(Post-Doc)
 - 2002년~2003년 Department of Computer Science, Old Dominion University (Visiting Professor)
 - 1989년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수
 - 관심분야 : 모바일 멀티미디어, 모바일 컨버전스, 모바일 컴퓨팅, 무선 인터넷 등
-
-