

유비쿼터스 컴퓨팅 환경에서의 능동적 디스플레이 디바이스 변경*

박종민⁰, 김광훈, 이경민, 이동만

한국정보통신대학교

{alrogi⁰, nuly17, kmlee, dlee}@icu.ac.kr

ADDC: Active Display Device Change in Ubiquitous Computing Environments

Jongmin Park⁰, Gwanghun Kim, Kyungmin Lee, Dongman Lee
School of Engineering, Information and Communications University

요 약

유비쿼터스 컴퓨팅 환경에서 사용자는 언제 어디서나 서비스를 제공받기를 원한다. 특히 멀티미디어 서비스의 경우 사용자의 위치가 변함에 따라 사용자와 가장 가까운 디스플레이를 통한 끊김없는(seamless) 멀티미디어 서비스를 요구한다. 본 논문은 이와 같이 다양한 디스플레이 디바이스를 통한 끊김없는 멀티미디어 서비스를 지원하기 위한 ADDC (Active Display Device Change) 프레임워크를 제안한다. 우리가 제안하는 프레임워크는 사용자의 위치에 따라 더 좋은 성능을 가진 최적의 디스플레이 디바이스를 검색하기 위하여 시맨틱(semantic) 서비스 디스커버리 기법을 이용하였다. 또한, 디스플레이 변경 시 능동적인 멀티미디어 세션 이동을 지원하는 핸드오프 관리자를 각 디스플레이 디바이스에 둔다. 마지막으로 디스플레이 디바이스의 특성을 고려한 멀티미디어 품질 적응(adaptation)을 위하여 트랜스크딩 서버를 이용한다. 제안한 프레임워크는 ICU에서 개발한 유비쿼터스 컴퓨팅 환경을 위한 미들웨어인 Active Surroundings를 이용하여 구현하였다.

1. 서론

모바일 컴퓨팅 기술의 발전에 힘입어 언제, 어디서나 개인 휴대단말을 통한 멀티미디어 서비스의 제공이 가능해졌다. 예를 들어, 무선 네트워킹이 가능한 곳이라면 언제 어디서나 PDA나 휴대폰을 통하여 비디오 스트리밍을 받아 볼 수 있다. 유비쿼터스 컴퓨팅은 이보다 한걸음 더 나아가 단순히 사용자의 개인 휴대 단말뿐 아니라 주변 환경에 편재되어 있는 다양한 디스플레이를 통한 끊김없는(seamless) 멀티미디어 서비스를 요구한다. 예를 들어, 회사에서 집으로 이동하면서 비디오 스트리밍을 받는 경우를 생각해 보자. 사무실에서 주차장에 있는 자동차로 이동할 때까지는 휴대폰을 통해, 운전 중에는 자동차의 텔레매틱스 단말을 통해, 그리고 집에 도착해서는 거실에 있는 커다란 PDP 디스플레이를 통해 계속해서 비디오 스트리밍을 받을 수 있어야 한다.

유비쿼터스 환경에 편재된 다양한 디스플레이를 통한 끊김없는 멀티미디어 서비스를 제공하기 위해서는 아래의 세가지 요구사항을 고려해야 한다.

- 첫째, 사용자가 명시적으로 디스플레이를 선택하지 않아도 사용자의 위치에 따라 최적의 디스플레이를 자동으로 선택해야 한다.

- 둘째, 사용자의 이동에 따라 디스플레이의 변경이 필요할 때 기존 디스플레이에서 새로운 디스플레이로 멀티미디어 세션을 능동적으로 옮겨주어야 한다.
- 셋째, 서로 다른 디스플레이의 화면 크기, 해상도, 네트워크 대역폭, 컴퓨팅 능력과 같은 특성에 따라서 멀티미디어 서비스의 품질(quality)을 동적으로 조절해야 한다.

GAIA 프로젝트[9]에서는 끊김없는 멀티미디어 서비스 제공을 위한 연구를 진행하였다. 이 연구는 위의 세가지 요구사항을 대부분 만족하고 있지만 다음과 같은 문제점이 있다. 첫째, 새로운 디스플레이로의 멀티미디어 세션 이동 시 디스플레이 디바이스의 화면 크기에 따른 비디오 적응(adaptation) 기능이 없다. 따라서, 디스플레이 디바이스의 크기가 서로 다를 경우 화질 저하나 스트리밍을 하지 못하는 경우가 발생한다. 예를 들어, PDA로 받아 보던 비디오의 화면 크기를 좀 더 커다란 화면 크기를 제공할 수 있는 PDP 디스플레이에서도 그대로 볼 수 밖에 없다. 이는 주위 환경에 편재하는 좋은 성능과 화면 크기를 가지는 디스플레이에서 멀티미디어 서비스를 제공하고자 하는 사용자의 욕구를 만족시키지 못하게 된다. 둘째, 디스플레이의 변경에 따른 멀티미디어 세션 이동 시 사용자 메타데이터 서버를 이용하여 사용자의 상태 정보를 저장한다. 멀티미디어 세션 이동시마다 기존 디스플레이 그리고 새로운 디스플레이는 모두 이러한 유저 메타데이터 서버에 대한 액세스가 필요하다. 이러한 액세스는 매번 유저가 이동할 때마다

* 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업과 정보통신부 및 정보통신연구원진흥원의 디지털미디어연구소 지원사업의 지원에 의한 것이다

발생하기 때문에 사용자가 자주 이동하는 유비쿼터스 환경에서는 많은 오버헤드가 될 수 있다.

본 논문에서는 GAIA 연구의 문제점을 해결하기 위해 ADDC(Active Display Device Change in Ubiquitous Computing Environments) 프레임워크를 제안한다. ADDC 프레임워크는 유비쿼터스 환경에 편재된 다양한 디스플레이를 통한 끊임없는 멀티미디어 서비스 제공을 위한 위의 세 가지 고려사항을 다음과 같이 해결하였다. 첫째, 자동적으로 최적의 디스플레이 디바이스를 찾기 위해서 요청된 디스플레이 디바이스의 이름 또는 속성을 가지고 검색을 하는 것이 아니라, 시맨틱 서비스 디스커버리를 이용하여 의미를 기반으로 검색을 한다. 둘째, 멀티미디어 세션의 능동적 이동을 위해 각각의 디스플레이 디바이스에 핸드오프 관리자가 존재한다. 핸드오프 관리자는 위치 인식 시스템과 시맨틱 서비스 디스커버리를 이용하여 멀티미디어 세션의 이동을 지원한다. 위치 인식 시스템을 통해 사용자의 위치 변화를 감지하면 시맨틱 서비스 디스커버리로부터 현재의 위치에서 가장 적절한 디스플레이 디바이스를 검색하여 새로운 디스플레이에 멀티미디어 세션의 이동(핸드오프)을 요청한다. 셋째, 멀티미디어 세션 이동 시 디스플레이 디바이스의 특성에 따른 품질 적응을 위해 트랜스코딩 기술을 이용한다. 이를 위하여 각 디스플레이 디바이스의 특성은 트랜스코딩 서버에 보내지고 이후에 트랜스코딩 서버는 현재 사용되는 디스플레이 디바이스의 특성에 맞게 비디오 스트리밍의 품질을 변화시킨다. GAIA 연구와는 달리 트랜스코딩 서버에서 비디오 품질 적응을 할 때 비디오 화면 크기에 따른 적응을 지원한다. 또한 우리는 사용자 메타데이터 서버를 없애고 디스플레이 디바이스 간에 직접적으로 상태 정보를 전송하도록 하였다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 유비쿼터스 환경에 편재하는 다양한 디스플레이를 통한 끊임없는 멀티미디어 서비스 시나리오를 구체적으로 기술한다. 3장에서는 이러한 시나리오를 지원하려고 했던 기존 연구에 대해 기술한다. 4장에서는 본 논문에서 제안하는 ADDC의 설계 고려 사항을, 5장에서 ADDC의 전체 구조 및 세부 설계 내용을 기술한다. 6장에서는 ADDC의 구현에 대해 기술한다. 마지막으로 7장에서 결론 및 향후 연구 방향에 대해 기술한다.

2. 시나리오

이번 섹션에서 제안된 프레임워크를 통해서 제공될 수 있는 시나리오에 대해서 설명한다. 시스템은 현재 사용자의 위치와 가장 가까운 곳에 있는 디스플레이 디바이스를 선택한다는 것을 가정으로 한다.

철수는 보고 싶은 우리나라 축구팀의 월드컵 축구 경기를 제시간에 보지 못했다. 그래서 철수는 VoD(Video On Demand) 서비스를 이용하여, 학교에서 수업이 끝나고 집에 오는 길에 그 경기를 PDA에 다시 보고 있다. 그는 집에 들어온 후에도 계속적으로 경기를 보고 싶어한다. 집에 들어왔을 때 그는 더 넓고, 더 성능이 좋은 디스플레이에서 그 축구 경기를 보기를 원하고 있다.

그가 집에 들어오자마자 ADDC 프레임워크는 앞에서

기술한 가정에 따라서 가장 좋은 디스플레이 디바이스를 발견한다. 우리의 가정에서처럼 그와 가장 가까운 디스플레이를 선택하고, 그 디바이스에서 축구 경기를 이어서 보여준다. 만약 그가 소파에 앉는다면, 축구 경기는 PDP 디스플레이에서 보여질 것이고, 만약 그가 거실로 이동한다면, 그 경기는 거실에 있는 LCD 디스플레이에서 보여질 것이다. 마지막으로 만약 그가 침실로 이동해서 침대에 눕는다면, 그 경기는 침실에 있는 LCD 디스플레이에서 보여질 것이다.

3. 관련 연구

본 절에서는 제안하는 기법과 관련된 연구에 대하여 기술한다.

Seamless User-level Handoff in Ubiquitous Multimedia Service Delivery [1]: 이 GAIA의 끊임 없는 비디오 스트리밍 연구에서는 유비쿼터스 멀티미디어 서비스 전달에서의 사용자 이동성을 고려한다. 사용자 이동성을 지원하기 위하여 이 논문의 저자들은 GAIA [9]에 기반을 둔 미들웨어 프레임워크를 디자인한다. 사용자 이동성은 호스트 이동성 뿐만 아니라 하나의 호스트에서 또 다른 호스트로의 스위칭하는 것도 포함한다. 여기서 호스트 이동성의 의미에 대하여 설명하면, 사용자가 여행하는 동안에 같은 서비스를 어느 장소에서나 액세스가 가능하다는 것을 의미한다. 하지만 사용자는 항상 동일한 모바일 호스트를 사용하는 것이 요구되는 문제점이 존재한다..

사용자 이동성에 초점을 맞추기 위하여, 이 논문에서는 2가지의 가정을 세운다. 첫째, 사용자 위치 인식 시스템이 있다는 가정이다. 이러한 위치 인식 시스템은 사용자의 현재 위치를 수정하고, 각각의 사용자의 ID와 프로파일에 관한 정보를 관리한다. 둘째, 서비스 디스커버리가 이용 가능하다는 가정이다.

그림1은 클라이언트 호스트의 미들웨어 아키텍처를 보여준다.

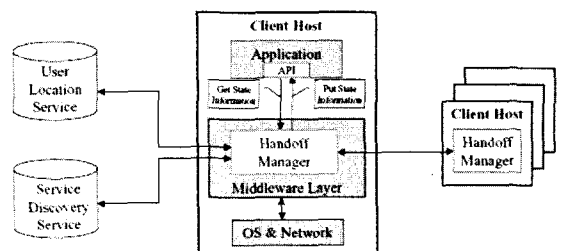


그림 1. 클라이언트 호스트에 있는 미들웨어 아키텍처

클라이언트 호스트에는 핸드오프 관리자의 기능을 하는 미들웨어 컴포넌트가 존재한다. 핸드오프 시에 그것은 상태 정보 관리 작업을 수행한다. 매번 사용자의 이동 이벤트가 발생하게 되면 사용자의 클라이언트 호스트는 실행중인 애플리케이션의 상태 정보를 사용자 메타데이터 서버에 저장한다. 이동한 위치의 새로운 클라이언트 호스트는 사용자 메타데이터 서버로부터 필요한 상태 정보를 얻어 온다. 수신 받은 상태 정보를 기반으로 하여,

새로운 클라이언트 호스트는 같은 애플리케이션을 시작하고 상태 정보를 복원한다. 여기서 사용자 메타데이터 서버는 이전의 클라이언트 호스트와 새로운 클라이언트 호스트 사이의 중간자 역할을 한다.

위의 연구는 우리의 기법과 매우 비슷하다. 그러나 우리의 기법에서는 디스플레이 서비스 디스크립션에 맞게 해당 디바이스에 적절한 비디오 품질을 보여줌으로서 다양한 디바이스에 적응하는 것뿐만 아니라, 이 논문에서는 해결하지 못했던 디스플레이 디바이스의 화면 크기에 따른 비디오 적응 기능, 즉, 비디오 화면 크기를 해당 클라이언트 호스트의 화면 크기에 알맞게 조정할 수 있는 기능을 가능하게 한다.

위에서 설명했던 것처럼 GAIA의 끊임 없는 비디오 스트리밍 논문 [1]에서는 상태 정보를 저장하기 위하여 사용자 메타데이터 서버를 필요로 한다. 그것은 주로 실행중인 애플리케이션의 상태 정보와 모든 상황(context) 정보를 포함하고 있다. 반면에 우리의 연구는 이러한 사용자 메타데이터 서버에 비디오 데이터 포맷, 현재의 프레임 혹은 순서 번호 등의 상태 정보를 저장할 필요가 없다. 그리고 우리가 고려하는 상황은 시스템에 의하여 자동적으로 알 수 있는 위치 정보이기 때문에 이러한 위치 상황 정보는 특별하게 저장할 필요가 없다.

Activity Policy-based Service Discovery for Pervasive Computing [2]: 이 논문에서는 다른 환경에서 다양한 서비스를 시맨틱하게 기술하고 발견할 수 있는 시맨틱 서비스 디스커버리 아키텍처를 제안한다. 그것은 퍼베이시브 컴퓨팅 환경에서 서비스의 다양성을 기술하기 위하여 OWL을 이용한 온톨로지를 디자인한다. 그것은 통신 방법으로서 HTTP 프로토콜을 이용한다. JRacer와 nRQL이 쿼리 조정과 시맨틱 매칭을 수행하기 위하여 사용된다. 우리는 시맨틱 서비스 디스커버리를 이루기 위하여 이 논문의 아키텍처를 부분적으로 사용한다.

A middleware Infrastructure for Active Surroundings [3]: 이 논문에서는 Active Surroundings를 위한 미들웨어 인프라스트럭처의 개요를 나타내고 있다. 이러한 미들웨어는 개인의 상황에 초점을 맞추기 보다는 그룹의 상황에 초점을 맞춘다. Active Surroundings 미들웨어는 5개의 주요한 컴포넌트로 구성되어 있다. 첫 번째는 환경 센싱 컴포넌트이다. 그것은 Active Surroundings에서 상황 변화를 센싱을 쉽게 하기 위한 컴포넌트이다. 두 번째는 상황 관리 컴포넌트이다. 그것은 상황 정보를 기술하고, Active Surroundings에서 상황 변화를 결정하는 것에 초점을 맞추고 있다. 세 번째는 현재 이용 가능한 자원과 서비스가 다이나믹하게 로드 되고, 언로드 되고, 만들어지고, 그리고 소실되는 것을 사용자의 간섭없이 가능하게 하는 다이나믹한 적응 프레임워크 컴포넌트이다. 네 번째는 적용하는 서비스 디스커버리 컴포넌트이다. 이것은 사용자의 프로파일, 위치, 그리고 근처의 컴퓨팅 환경 같은 상황 정보에 따라서 적절한 서비스를 발견하고 선택한다. 마지막으로 다이나믹한 적응을 하는 네트워크는 디바이스 사이에 정보를 전달하기 위한 디바이스 사이의 연결성을 가능하게

한다.

4. 설계 시 고려사항

4.1 시맨틱 서비스 디스커버리 기법에 대한 고려사항

ADDC 프레임워크는 사용자의 위치가 변할 때 자동적으로 적절한 디스플레이 서비스를 찾아 줄 수 있어야 한다. 이전의 서비스 디스커버리 기법들은 서비스의 구문상의(syntactic) 디스크립션, 혹은 키워드 기반의 디스커버리 기법에 의존한다. 이러한 디스커버리 기법은 몇 가지 제한점을 갖는다. 먼저 서비스의 기본적인 정보만 기술되기 때문에 서비스의 기능 및 특징에 대해 파악하기 힘들다. 그리고 서비스를 사용하기 위하여 인터페이스에 대한 자세한 정보까지 알아야 한다. 또한 현재의 방법으로는 질의에 정확하게 일치하는 서비스만 검색 가능하고, 부정확한 매칭(inexact matching)에 대한 고려가 없다. 따라서 우리는 요청된 서비스의 의미를 바탕으로 부정확한 매칭을 지원하기 위하여 시맨틱 서비스 디스커버리 기법을 고려한다.

시맨틱 서비스 디스커버리 기법에는 몇 가지 요구사항이 있다. 첫 번째로, 모든 서비스의 속성과 기능이 기술되어야 한다. 두 번째로, 사용자가 구문상의 표현을 정확히 알지 못해도 필요한 서비스의 요청이 가능해야 한다. 마지막으로, 본 연구에서는 가장 가까운 위치에 있는 디스플레이를 선택하는 것을 가정하기에 서비스 디스커버리 기법은 사용자의 위치정보를 얻을 수 있어야 한다.

4.2 핸드오프 관리에 대한 고려사항

본 연구에서의 핸드오프 의미는 사용자의 위치가 다른 장소로 이동 했을 때, 비디오는 이전의 디스플레이에서 새로운 디스플레이로 이동할 수 있어야 한다는 것을 말한다. 이러한 기능을 지원하기 위하여 시스템은 채널 정보 즉, 콘텐츠 서버(CS)의 주소 같은 비디오 세션에 필요한 상태 정보를 이전의 디바이스에서 새로운 디바이스로 전송할 수 있어야 한다.

4.3 비디오 스트리밍 전달에 대한 고려사항 [8]

사용자의 서비스가 새로운 디스플레이 서비스로 이동했을 때 비디오 스트리밍 전달의 관점에서 몇 가지 사항을 고려해야 한다. 예를 들면, 사용자가 PDA에서 PDP 디스플레이로 비디오 세션을 이동하려고 시도할 때, 다음의 몇 가지 문제가 발생한다. 첫 번째로, PDP는 PDA보다 더 좋은 컴퓨팅 능력과 더 높은 대역폭을 가지고 있기 때문에 PDA에서의 비디오 품질은 PDP에서는 부적절하다. 따라서 PDP는 콘텐츠 서버로부터 더 좋은 품질의 비디오 스트림을 수신 받아야 한다. 두 번째로, PDP가 PDA 보다 훨씬 더 큰 화면 크기를 가지고 있기 때문에 클라이언트 디바이스의 디스플레이 크기에 기반을 두는 화면 적응을 고려해야 한다. 마지막으로 유비쿼터스 컴퓨팅 환경에서 VoD 서비스의 경우에 핸드오프 전의 디스플레이 디바이스에서 보던 장면의 다음 장면이 핸드오프 후에 새로운 디스플레이 화면에서 보여야 한다. 즉, 비디오 스트리밍의 Pause/Play 기능이 지원되어야 한다.

5. 설계

5.1 시스템 개요

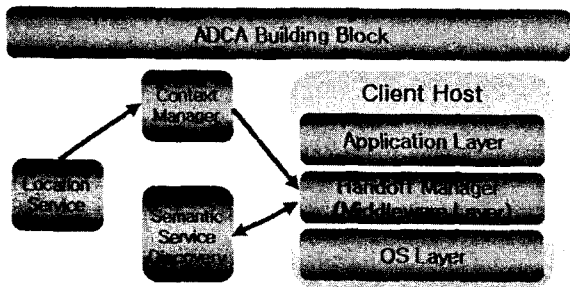


그림 2. ADCC 빌딩 블럭

그림2에서 보는 것처럼, 제안된 프레임워크는 핸드오프 관리자, 상황 관리자, 위치 서비스, 그리고 시맨틱 서비스 디스커버리로 구성된다. 핸드오프 관리자는 모든 컴포넌트를 제어하는 역할을 수행한다. 핸드오프 관리자는 미들웨어 계층으로 클라이언트 호스트에 존재한다. 핸드오프 관리자는 위치 변화 이벤트를 받자마자 적절한 서비스 찾기 위하여 시맨틱 서비스 디스커버리에게 요청한다. 위치 서비스는 위치를 감지하는 역할을 한다. 위치를 감지한 후에 그것은 위치 정보를 상황 관리자에게 보낸다. 상황 관리자는 이벤트를 발생시키고, 관리하는 역할을 수행한다. 상황 관리자가 위치 정보를 받았을 때, 상황 관리자는 ‘거실에 들어옴’ 혹은 ‘침실에 들어옴’ 같은 위치 변화 이벤트를 발생시킨다. 시맨틱 서비스 디스커버리는 사용자가 원하는 서비스를 요청 할 때, 사용자에게 적절한 서비스를 제공하는 역할을 한다.

모든 컴포넌트는 Active Surroundings에서의 서비스 상호 작용 방법인 SIB(Service Interaction Broker)를 이용하여 서로 통신을 한다. 이후에 시맨틱 서비스 디스커버리, 비디오 스트리밍 전달, 그리고 핸드오프 절차 같은 제안된 프레임워크의 중요한 고려 요소의 설계에 관하여 각각 자세히 설명한다.

5.2 시맨틱 서비스 디스커버리

온톨로지란 공유된(shared) 개념들의 개념화(conceptualization)를 정형적(formal)이고 명시적으로(explicit) 설명한 명세서(specification)을 의미한다 [11, 12]. 우리는 서비스들을 공통된 규칙으로 표현하기 위하여 또는 서비스를 더 잘 찾기 위하여 온톨로지를 이용한다. 디스플레이 서비스를 표현하기 위해서 서로간에 계층적 관계를 가지는 서비스 클래스를 정의한다. 마지막으로 서비스의 특성을 나타내기 위해 애트리뷰트 클래스를 정의한다. 우리는 온톨로지 언어로서 웹 온톨로지 언어를(OWL)을 사용하였고, 디스플레이 서비스를 표현하기 위한 간단한 온톨로지를 만들었다.



그림 3. 서비스 디스크립션 - (예: LCD 디스플레이 서비스의 Protégé 화면)

그림3에서처럼, 서비스 디스크립션은 많은 속성을 이용하여 서비스를 기술한다. 이러한 서비스 디스크립션은 사용자에게 적절한 서비스의 쉬운 검색을 돕기 위하여 시맨틱 서비스 디스커버리에 등록을 한다. 우리의 서비스 디스크립션은 서비스의 레퍼런스, 서비스의 물리적 위치, 서비스의 비디오 코덱, 비디오 프레임 비율, 비디오 비트율, 비디오 화면의 넓이, 그리고 비디오 화면의 높이 같은 속성들을 가지고 있다.

타입 매칭이 아니라 시맨틱 매칭을 가지고 적절한 서비스를 찾기 위해서는 인스턴스 간의 관계를 추론할 수 있는 추론 엔진이 필요하다. 추론 엔진은 데이터 저장소로서 온톨로지를 사용한다. 우리는 Racer 추론 엔진을 사용한다.

5.3 비디오 스트리밍 전달

우리는 비디오 품질, 비디오 화면 사이즈 같은 클라이언트 디바이스의 특성에 적응하기 위하여 콘텐츠 서버 앞에 트랜스코딩 서버를 놓는다. 우리의 트랜스코딩 서버는 VLC(Videolan Client) 비디오 미디어 플레이어 [6]의 트랜스코딩 기술을 사용한다. 우리는 비디오 코덱, 비트율, 프레임 비율, 전체 픽셀 크기(width × height)에 따라서 비디오 콘텐츠를 조정한다. 이러한 비디오 트랜스코딩을 사용함으로써, 우리는 이질적인 네트워크와 호스트 디바이스에 적응 위하여 압축된 비디오 스트림을 하나의 형태에서 또 다른 형태로 변환 할 수 있다.

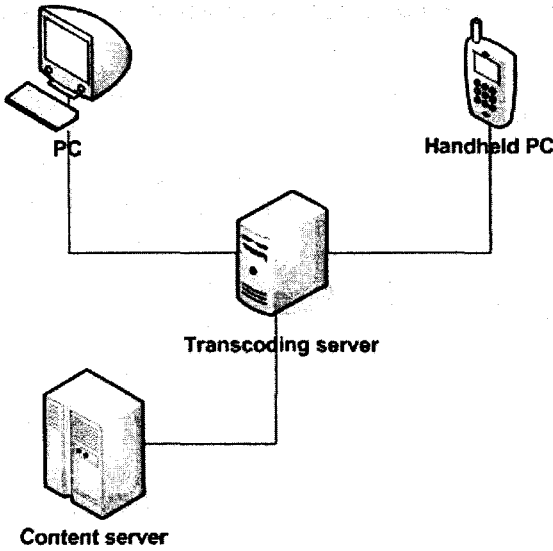


그림 4. 비디오 스트리밍 전달을 위한 각 컴포넌트

그리고 클라이언트의 디바이스에 적응하기 위하여, 우리는 서비스 디스커버리의 서비스 저장소에 저장된 서비스 디스크립션을 사용한다. 서비스 디스커버리의 탐색 결과가 하나 이상 발견될 때 이러한 서비스 디스크립션 또한 사용될 수 있다.

마지막으로 앞의 고려사항에서 언급했던, Pause/Play 기능을 제공하기 위하여, 우리는 핸드오프가 발생했을 때 콘텐츠 서버는 VLC 미디어 플레이어의 중지(pause) 명령어를 사용한다. 그리고 핸드오프 절차가 끝났을 때, 콘텐츠 서버에 있는 중지 명령어를 사용하여 VLC 미디어 플레이어를 다시 시작한다. 그러므로 위에서 언급했던 GAIA의 끊임 없는 비디오 스트리밍 연구 [1]와는 다르게, 우리는 유저 메타데이터 서버에 비디오 데이터 포맷, 현재의 프레임과 비디오 재생 순서 번호 같은 상태 정보를 저장할 필요가 없다.

5.4 ADDC 프레임워크의 핸드오프 절차

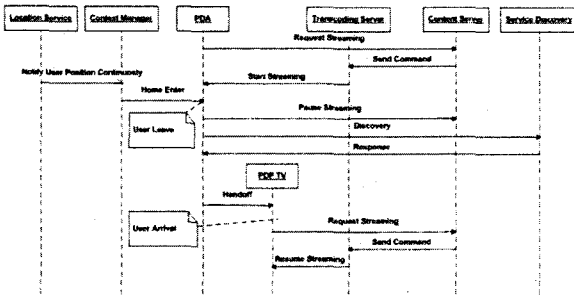


그림 5. ADDC 프레임워크의 시퀀스 다이어그램

그림5는 ADDC 프레임워크의 시퀀스 다이어그램이다. 이해를 돕기 위하여 시퀀스의 흐름대로 작동을 설명한다.

1. 사용자가 집밖에 머무를 때, PDA 디스플레이 디바이스는 콘텐츠 서버에게 스트리밍을 요청한다. 그 때 직접 자신의 서비스 디스크립션 정보를 콘텐츠 서버에게 전달한다. 이 때 콘텐츠 서버는 트랜스코딩 서버에게 해당 디스플레이 디바이스에 알맞은 트랜스코딩 정보를 전달한다. 그 후에 서버로부터 받은 비디오 스트리밍을 이용해서 트랜스코딩 서버는 스트리밍을 릴레이 해준다.

2. 위치 서비스는 사용자의 위치를 계속적으로 상황 관리자에게 통보한다.

3. 상황 관리자는 위치 서비스로부터 받은 사용자의 위치를 가지고, '거실에 들어옴' 혹은 '침실에 들어옴' 같은 위치 변화 이벤트를 발생시킨다. 이러한 위치 변화 이벤트를 해당 디스플레이 서비스에게 통보한다.

4. 위치 변화 이벤트가 도착했을 때 이전의 디스플레이 서비스는 Pause 요청을 콘텐츠 서버에게 전달하면서 자신이 보고 있던 VLC 미디어 플레이어를 종료 시킨다.

5. 이전의 디스플레이 서비스는 적절한 디스플레이 서비스를 찾기 위하여 시맨틱 서비스 디스커버리에 요청한다. 시맨틱 서비스 디스커버리는 미리 등록된 서비스 디스크립션 정보를 통하여 사용자가 원하는 서비스를 찾아주고, 서비스 디스크립션 정보 또한 응답해준다.

6. 이전의 디스플레이 서비스는 새로운 디스플레이의 서비스 디스크립션 정보를 포함하는 핸드오프 메시지를 새로운 디스플레이 서비스에 전달하고, 이 후에는 1번 단계에서 수행했던 작업과 비슷한 작업이 진행된다.

6. 구현



그림 6. Active Surroundings 데모룸

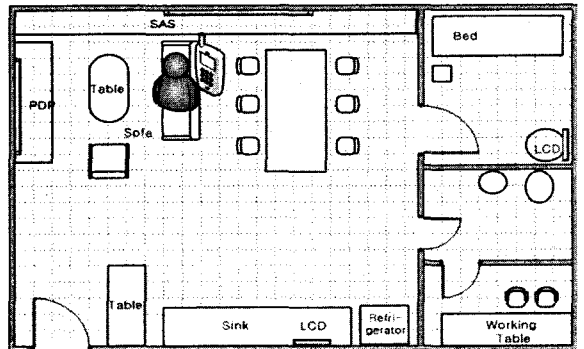


그림 7. Active Surroundings의 전체 레이아웃

ADDc 프레임워크가 성공적으로 작동하는지 확인하기

위하여, 그림6의 Active Surroundings [3] 데모룸에 ADDC 프레임워크를 구현했다. 그림7에서처럼 우리는 PDA, PDP, 및 LCD 같은 3개의 디스플레이 디바이스에서 구현 및 테스트를 수행했다. PDA는 휴대용이 가능하고, PDP 디스플레이 디바이스는 소파 근처에 위치되어 있고, LCD 디스플레이 디바이스는 침실에 위치되어 있다.

일반 디스플레이 디바이스를 위하여, ADDC는 Java 언어(J2SE 5.0 JDK)로 구현됐고, 소형기기(PDA)를 위하여 우리의 프로그램을 포켓 PC를 위한 J9 IBM JVM에 알맞게 변환하는 작업 또한 수행되었다. 소형기기는 HP사의 iPAQ rw6100 스윙 폰을 이용하였다.

온톨로지는 오픈 소스 온톨로지 에디터인 Protégé [4]를 사용하였다. 온톨로지 데이터를 저장하고 쿼리하기 위하여 우리는 RacerPro [5]를 사용했다. 비디오를 전송하고 수신받기 위하여, 우리는 VLC 미디어 플레이어 [6]를 이용했고, VLC의 트랜스코딩 기능이 이용되었다.

이외에도 각각의 컴포넌트 사이의 프로토콜은 구현되었다. 비디오 스트리밍 소스의 코덱은 DIV3가 사용되었다. 마지막으로 Active Surroundings 데모룸 내에서 각각의 사용자는 액티브 배지(Active badge)를 휴대하고, 사용자의 위치를 인식하기 위하여 Ubi-Sense [7]가 사용되었다.

7. 결론 및 향후 과제

본 연구를 통하여 우리는 환경의 제약 없이, 사용자의 위치에 기반을 둔 디스플레이 디바이스 이동을 가능하게 하는 끊임 없는 비디오 스트리밍 애플리케이션을 구현했다. 우리의 프레임워크는 사용자의 위치가 변했을 때, 시맨틱 서비스 디스커버리를 이용하여 찾은 디스플레이로 해당 세션이 이동하면서 스트리밍 서비스를 제공해줄 수 있다. 이외에도 다양한 클라이언트 디바이스에 맞는 다이나믹한 미디어 품질 적응을 제공한다. 마지막으로, 우리는 유비쿼터스 컴퓨팅환경을 구현하기 위한 Active Surroundings 프로젝트 환경에서 성공적으로 테스트를 수행하였다.

차후에 본 연구에서는 몇 가지 사항에 대하여 보완, 발전시킬 것이다. 첫째, 각 디스플레이 디바이스의 특성에 맞게 미디어의 품질을 적응한다. 이때 다이나믹하게 적응하기 위하여 기존에는 서비스 디스크립션 정보를 이용하는 수동적인 트랜스코딩 기법을 사용한다. 그러나 네트워크의 대역폭과 같은 디바이스의 특성은 수시로 변할 수 있기 때문에 현재 우리는 계층적인 비디오 오버레이 멀티캐스트 기법을 사용하여 이 문제를 해결하려고 시도하고 있다. 둘째, 우리는 개발된 서비스 디스커버리를 좀 더 발전시켜 상황 인식이 가능한 서비스 디스커버리로 구현할 예정이다. 상황 인식 서비스 디스커버리는 사용자의 선호도, 시간, 그리고 사회적인 관계 같은 상황을 가지고 더욱더 적절한 서비스를 찾아줄 수 있다. 이러한 차후의 시도를 통하여 우리의 ADDC 프레임워크는 좀 더 효율적인 시스템이 될 것이다.

참고 문헌

- [1] Y. Cui, K. Nahrstedt, D. Xu, "Seamless User-level Handoff in Ubiquitous Multimedia Service Delivery," In

Multimedia Tools and Applications Journal, Kluwer Publisher, Special Issue on Mobile Multimedia and e-Commerce, 2003.

- [2] W. Kim, S. Kang, Y. Lee, D. Lee, and I. Ko, "Activity Policy-based Service Discovery for Pervasive Computing," 2nd International Workshop on Pervasive Information Management, March 2006.
- [3] D. Lee, "Active Surroundings: A Group-Aware Middleware for Embedded Application Systems," IEEE COMPSAC 2004, pp. 404 - 405, September 2004.
- [4] Protégé, <http://protege.stanford.edu/>
- [5] RacerPro, <http://www.racer-systems.com/>
- [6] VLC media player, <http://www.videolan.org/vlc/>
- [7] Ubisense, <http://www.ubisense.net/>
- [8] Z. Lei and N.D. Georganas, "Context Based Media Adaptation in Pervasive Computing," Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2001, Toronto, Canada, May 2001.
- [9] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A Middleware Infrastructure for Active Spaces," IEEE Pervasive Computing, vol. 1, no. 4, pp. 74-83, Oct./Dec. 2002.
- [10] P. Borst, H. Akkermans, and J. Top, "Engineering Ontologies," International Journal of Human-Computer Studies, pp. 365-406, 1997.
- [11] A. Maedche and S. Staab, "Ontology Learning for the Semantic Web," IEEE Intelligent Systems, Vol. 16, No. 2, pp. 72-79, 2001.