유비쿼터스 환경에서 서비스 이동을 이용한 동적 재구성의 설계와 구현

권정호[O1] 정성영[1] 김재훈[1] 조위덕[2]
[1]아주대학교 정보통신전문 대학원
[2] 유비쿼터스컴퓨팅사업단
jungho[O]@dmc.ajou.ac.kr, {inyoureyes98, jaikim}@ajou.ac.kr, chowd@ajou.ac.kr

Design and Implementation of Dynamic Reconfiguration using Service Migration In Ubiquitous Environment

Jung-Ho Kwon[O1] Sung-Young Jung[1] Jai-Hoon Kim[1] We-Duke Cho[2]
[1]Graduate School of Information and Communication, Ajou University
[2]Center of Excellence for Ubiquitous System

요 약

유비쿼터스 환경에서 사용자는 언제 어디서나 원하는 서비스를 제공받을 수 있어야 한다. 이를 위해, 사용자는 데스크탑, 노트북, 그리고 PDA와 같은 다양한 디바이스들을 이용하여, 사용자가 원하는 서비스를 특정 상황의 제약 없이 끊임없이 제공받을 수 있는 기술이 필요하다.

우리는 유비쿼터스 환경의 사용자에게 끊임없는 서비스를 제공하기 위하여 재구성 시스템과 시나리오를 설계 및 구현하였다. 재구성 시스템은 상황에 맞는 서비스를 제공하기 위한 최적의 디바이스를 선택할 수 있다. 또한 실행중인 서비스를 새로 선택한 최적의 디바이스에서 실행하는 서비스의 이동이 가능하다. 본 논문의 시나리오에서는 음악 서비스를 실행 중인 디바이스로부터 사용자가 지속적인 서비스를 제공받지 못할 경우, 변화된 상황에 따른 최적의 디바이스로 이동하여 사용자에게 기존의 음악 서비스를 지속적으로 제공할 수 있음을 보여준다.

1. 서 론

유비쿼터스 환경에서 사용자는 언제 어디서나 원하는 서비스를 제공받을 수 있어야 한다. 이를 위해, 사용자는 데스크탑, 노트북, 그리고 PDA와 같은 다양한 디바이스들을 이용하여, 사용자가 원하는 서비스를 특정 상황의 제약 없이 끊임없이 제공받을 수 있는 기술이 필요하다 [3]. 예를 들어, 사용자가 유선의 오디오 디바이스를 이용하여 미디어 서비스를 제공받는 상황에서 기존 오디오 디바이스가 제공할 수 없는 환경으로 이동하게 될 경우에 사용자는 계속해서 기존 서비스를 끊임없이 제공받아야 한다.

우리는 유비쿼터스 환경의 사용자에게 끊임없는 서비스를 제공하기 위하여 재구성 시스템과 이를 증명할 수 있는 음악플레이어를 설계 및 구현하였다. 재구성 시스템은 상황에 맞는 서비스를 제공하기 위한 최적의 디바이스를 선택할 수 있다. 또한 실행중인 서비스를 새로 선택한 최적의 디바이스에서 실행하는 서비스의 이동이 가능하다. 이를 구현하기 위해서 모니터링 모듈, 추론 모듈, 재구성 모듈, 그리고 다양한 디바이스에서 미디어 플레이가 가능한 어플리케이션들이 필요하다.

모니터링 모듈은 서비스의 동적 재구성을 위하여 센서를 이용하여 사용자의 위치와 디바이스들의 상태와 같은 주변 환경 정보를 모니터링 한다. 추론 모듈은 모니터링 모듈로부터 주위 환경의 컨텍스트 정보를 받고, 이를 이용하여 가장 알맞은 서비스와 디바이스를 선택할 수 있도록 현재의 상황을 추론한다. 그리고 재구성 모듈은 추론 모듈로부터 받은 상황정보를 통해 새로운 디바이스와 서비스를 결정하여 기존 디바이스에서의 실행 중인 서비스를 중지시키고, 새로 선택한 디바이스로 기존 서비스를 이동하는 기능을 한다. 마지막으로 노트북과 PDA의 디바이스의 미디어 플레이어는 새로운 미디어의 리스트와 현재 실행 중인 상태를 서버로부터 제공받고, 이를 동적으로 실행할 수 있는 기능을 가진다.

본 논문의 시나리오에서는 기존 디바이스가 실행 중인 음악 서비스를 제공하지 못할 경우에 사용자에게 해당 서비스를 지속적으로 제공할 수 있는 새로운 디바이스에 기존의 음악 서비스를 변화된 상황에 따른 최적의 디바이스로 이동하여 사용자가 끊임없이 서비스를 제공받을 수 있음을 보여준다.

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술 개발사업의 지원에 의한 것임

2. 관련 연구

2.1 동적 재구성

동적 재구성(Dynamic Reconfiguration)은 시스템을 재시작 하지 않고, 해당 서비스를 재구성할 수 있는 기술로, 재구성의 대상은 보다 효율적이고 안정적인 QoS와 원하는 서비스를 제공받을 수 있다. 일반적으로 동적 재구성은 재구성하기 위한 오브젝트의 추가, 갱신, 삭제, 그리고 이동(migration)과 같이 네 가지 동작으로 이루어진다.

동적 재구성 이후에도 시스템이 올바르게 실행되어야 한다. 이를 위해 재구성 이후에도 반드시 " 정확성" 이 보장되어야 한다. 정확성을 유지하기 위해서는 다음의 세 가지 조건을 만족해야 한다[1]. 재구성 이후에도 시스템의 구조적 통합이 유지되어야 하고, 상호배제(mutual exclusion)를 통하여 일관성 있는 상태를 만족해야 하고, 관련된 어플리케이션과의 상태가 변화 없이 유지되어야 한다.

위와 같이, 동적 재구성은 시스템의 재 시작이 허용되는 정적 재구성보다 정확성을 보장하기 위해 더욱 많은 사항을 고려해야 한다. 또한, 동적 재구성은 실행 도중에 시스템이 변화하므로 매우 복잡하다. 그림. 1은 은행 시스템의 컴포넌트 재구성을 보여주는 예이다. 그림. 1은 OLTP(Online Transaction Processing) 컴포넌트를 위해 Account database, Update account, 그리고 Replica 컴포넌트의 재구성을 보여준다. OLTP는 네트워크상의 여러 이용자가 실시간으로 데이터베이스의 데이터를 갱신하거나 조회하는 등의 단위 작업을 처리하는 방식을 말한다. OLTP는 트랜잭션을 이용하여 프로세스 처리가 가능하여, 보다 효율적인 시스템의 운영을 가능하게 한다.

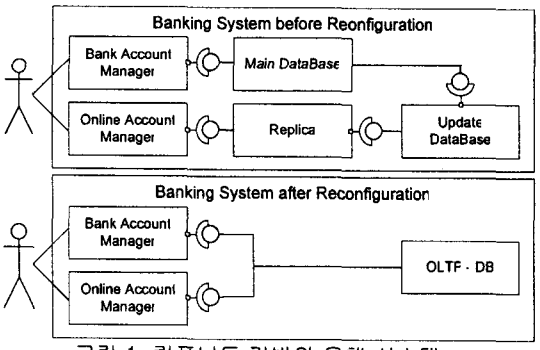

그림 1. 컴포넌트 기반의 은행 시스템

2.2 Migration기법

이동은 동적 재구성의 네 가지 동작 중 하나로, 실행 중인 오브젝트의 노드를 옮기는 것으로 기존 노드가 해당 오브젝트를 실행할 수 없는 상태가 되거나, 이동할 노드에서의 실행이 더욱 빠를 경우에 수행된다.

동적 재구성의 이동은 기존의 노드에 있는 오브젝트를 삭제하고, 이동하는 노드에 해당 오브젝트를 생성한다[1]. 여기서 이동되는 오브젝트의 세션 상태가 그대로 유지되 어야 하고, 사용자는 재구성으로 인한 변화와 상관없이 서비스를 끊임없이 제공받아야 한다.

3. 시스템 구조 설계

3.1 전체 구조 개요

본 시스템은 기존의 실행 중인 서비스를 동적으로 재구성하여 새로운 어플리케이션에 제공한다. 유비쿼터스 환경에서는 사용자의 입력 없이 자동으로 서비스와 실행을 담당하는 어플리케이션의 주체가 변경되어야 한다. 때문에 자동으로 서비스와 어플리케이션을 결정할 수 있는 추론 모듈과 추론이 가능케 하는 컨텍스트 정보를 생성해 주는 모니터링 모듈이 필요하다.

그림 2는 서비스 재구성을 가능케 하는 시스템의 전체 개요이다. 어플리케이션을 선택하고 서비스를 재구성하여 제공해주는 서비스 재구성 모듈은 추론 모듈과 모니터링 모듈과 함께 동작한다. 모니터링 모듈은 센서를 이용하여 주위 환경을 모니터링하고, 추론 모듈은 모니터링된 컨텍스트 정보를 이용하여 현재의 상황을 재구성 모듈이 현재의 상황을 분석 할 수 있는 형태로 상황정보를 생성한다. 우리는 JESS를 이용하여 추론 모듈을 구현하였다. 재구성 모듈은 위의 상황정보를 통해 최적의 어플리케이션을 찾아서 서비스를 재구성하여 제공하여 준다.

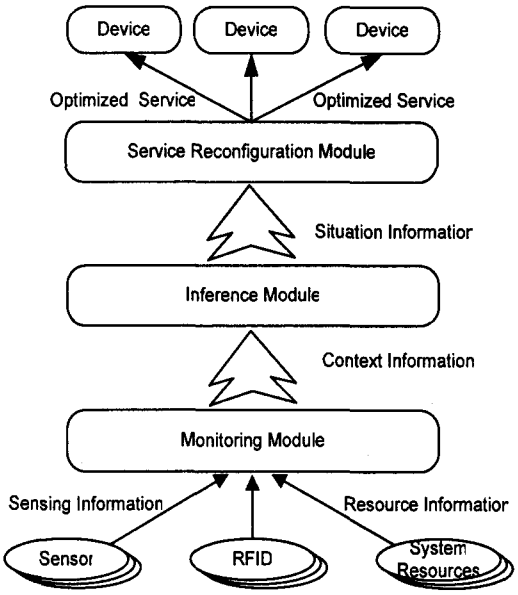

그림 2. 시스템의 전체 개요

서비스 재구성 모듈은 다른 모듈과 연계하여 수행해야 한다. 추론 모듈이 추론한 상황 정보를 전달 받기 위해 추론 모듈과 통신을 해야 한다. 이를 통해 재구성 모듈은 추론 모듈로부터 사용자의 상황 정보, 시스템의 자원상황 정보, 그리고 주변 환경의 상황 정보를 제공받는다. 재구성 모듈이 제공받은 정보들을 받고 상황 정보를 올바르게 인식하였는지에 대한 결과와 인식하지 못하였을 경우에 어떠한 정보에 문제가 있는지를 추론 모듈에 반납한다. 또한 모니터링 모듈에 모니터링 임무와 관련된 서비스 정책을 전송하여 다양한 종류의 서비스를 재구성한다. 서비

스 재구성 모듈은 이 정책을 전달하기 위해서 모니터링 모듈과 통신을 해야 하며, 모니터링 모듈은 이 명령을 기반으로 모니터링 임무를 수행한다. 그림 3은 위의 각 모듈의 상호작동을 위한 명령의 통신의 흐름을 나타낸다.

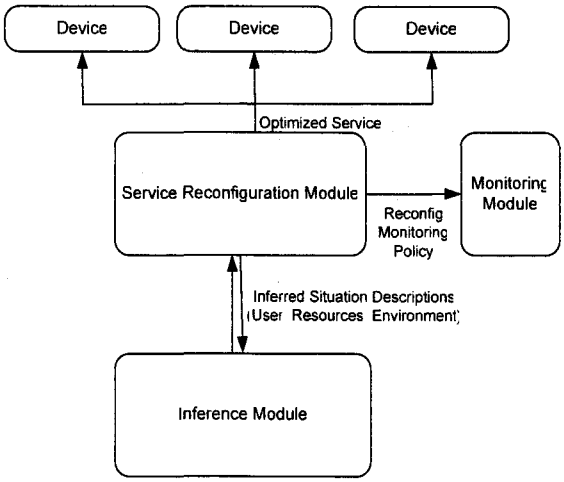

그림 3. 각 모듈의 통신 흐름

3.2 재구성 모듈의 구조

그림 4는 서비스 재구성 모듈의 세부 구조를 나타낸다. 서비스 재구성 모듈은 크게 서비스 결정자(Service Determiner) 모듈과 서비스 제공자(Service Provider)로 구성된다. 서비스 결정자는 디바이스에 제공할 서비스를 결정하는 역할을 하며, 서비스 제공자는 타겟 오브젝트에 재구성 서비스를 제공하는 역할을 한다. 외부의 어플리케이션의 서비스 재구성의 경우 해당 디바이스와의 통신을 통해 재구성 정보와 메시지를 전송하여 재구성을 실행한다.

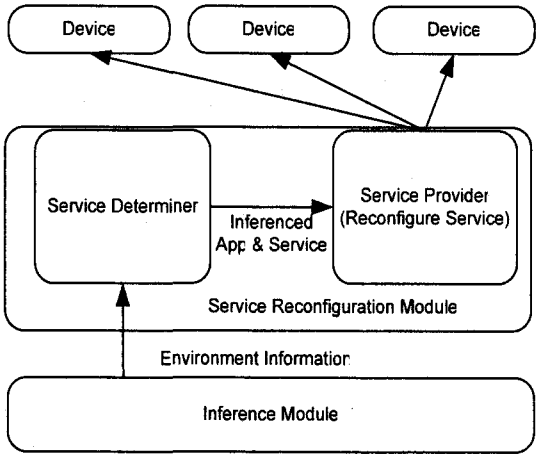

그림 4. 서비스 재구성 모듈의 구조

3.2.1 서비스 결정자(Service Determiner)

추론 모듈로부터 받은 환경정보, 사용자의 위치 정보, 자원 정보 등을 바탕으로 서비스 결정자는 현재 서비스를 제공받고 있는 디바이스에 더 나은 서비스를 제공할 수 있는지 분석한다. 이를 위해 서비스 재구성 모듈 내에는 디바이스 정보 테이블을 유지한다. 즉, 각 디바이스가 제공받을 수 있는 서비스와 현재 제공받고 있는 서비스, 그리고 디바이스의 자원 상황, 디바이스 ID, 디바이스와의 통신을 위한 주소 등이 테이블에 저장되어 있다. 서비스 결정자는 현재의 추론된 환경 정보를 이용하여 해당 디바이스에게 더욱 적합하다고 추론되는 서비스와 관련 정보를 결정한다. 그리고 선택한 디바이스에 새로운 서비스를 제공할 경우 생기는 장단점을 비교하여, 새로운 서비스 제공 유무를 결정한다. 서비스 결정 모듈이 새로운 서비스를 제공해야겠다는 결정을 내린 경우 해당 디바이스의 이름(ID)과 어플리케이션과 제공할 재구성 서비스 정보를 서비스 제공자에게 전달한다.

3.2.2 서비스 제공자

서비스 제공자는 서비스 결정자로부터 받은 디바이스 명과 어플리케이션, 서비스 명을 이용해 해당 디바이스에 재구성이 가능한 상황인지 확인한다. 재구성이 가능한 상황이라면 새로운 서비스를 제공할 디바이스에게 기존에 제공하던 서비스를 중단 시키고, 최적의 서비스를 제공한다. 새로운 서비스를 제공한 후, 디바이스로부터 재구성 서비스의 실행 성공 여부를 리턴 받는다. 서비스 재구성의 실행의 성공을 확인한 서비스 제공자는 서비스 재구성 모듈의 디바이스 정보 테이블에 해당 디바이스의 제공받고 있는 서비스 정보 값을 갱신한다.

3.2.3 서비스 재구성의 단계

서비스 재구성은 다음의 세 단계로 구분된다. 첫째로, 매개변수의 재구성이다. 매개변수의 값을 재구성하여 실행 중인 특정 디바이스에 전송하면, 해당 디바이스는 제공받은 값에 따라서 실행의 조건과 서비스가 변경될 수 있다. 둘째로, 구성정보이다. 재구성된 새로운 구성정보를 받은 디바이스는 매개변수보다 더욱 많은 부분을 재구성할 수 있다는 장점이 있지만, 매개변수의 재구성보다 더욱 복잡하다는 단점이 있다. 마지막으로 프로그램의 재구성이다. 재구성된 프로그램을 디바이스로 전송하여 실행하도록 하는 것이다. 디바이스는 기존과는 완전히 다른 프로그램을 수행하여 기존과 다른 서비스를 제공받을 수 있다.

3.3 외부 모듈과의 상호연동을 위한 인터페이스

서비스 재구성 모듈을 위한 인터페이스로는 기본적으로 추론 모듈로부터 상황정보를 받아들이는 인터페이스 사이의 약속과, 모니터링 모듈의 정책을 재구성할 수 있는 인터페이스, 그리고 외부 디바이스에 재구성한 서비스를 제공해주는 인터페이스가 필요하다. 위의 세 가지 인터페이스는 다음과 같이 정리할 수 있다.

3.3.1 추론 모듈과 통신을 위한 프로토콜

필수 인터페이스는 recvContext(type, time, context) 이다. 서비스 재구성 모듈은 추론 모듈로부터 추론된 상황정보를 받는다. recvContext 인터페이스에는 다음과 같은 매개변수가 있다.

-type : 상황 정보의 종류(예, 환경, 유저 등).

-time : 상황 정보가 추론된 시간.

-context : 추론된 정보의 내용.

3.3.2 모니터링 모듈과 통신을 위한 프로토콜

필수 인터페이스는 reconfigMonitor(policy)이다. 모니터링 모듈에게 자원 수집 정책을 변경하는 명령을 보내기 위한 인터페이스이다. 매개변수는 다음과 같다.

-policy : 모니터링을 하는데 필요한 재구성 정보(예, 모니터링 주기변경).

3.3.3 외부 디바이스와 통신을 위한 프로토콜

필수 인터페이스는 sendService(id, type, service)이다. 디바이스에게 재구성할 서비스를 전송하기 위한 인터페이스이다. 매개변수는 다음과 같다.

-id : 재구성할 해당 디바이스의 아이디.

-type : 재구성되는 서비스의 종류(예, 매개변수, 구성정보, 프로그램 등의 종류 구분).

-service : 재구성되는 새로운 서비스의 이름.

4. 시나리오

본 논문의 시나리오에서는 기존 디바이스가 실행 중인 음악 서비스를 제공하지 못할 경우에 사용자에게 해당 서비스를 지속적으로 제공할 수 있는 새로운 디바이스에 기존의 음악 서비스를 변화된 상황에 따른 최적의 디바이스로 이동하여 사용자가 끊임없는 서비스를 제공받을 수 있음을 보여준다.

우리는 위에서 설계한 시스템을 구현하였고, 서비스를 제공받는 디바이스로 노트북과 PDA를 선택하였다. 그리고 각 디바이스에서 서비스를 실행하기 위한 프로그램으로 음악 플레이어를 구현하였다. 시나리오를 위한 사용자의 위치 이동 정보는 시뮬레이션을 이용하여 수행하였다.

그림 5는 음악플레이어의 구조를 보여준다. 음악 플레이어는 기본적으로 MP3, WAV 등의 음악 파일 연주와 음악 파일 및 리스트와 플레이어의 제어 메시지를 외부로부터 전송 받아 동적으로 연주한다. 또한 자신의 디바이스의 자원상태 및 자신의 디바이스에서 재구성 가능한 서비스의 목록 정보와 현재의 서비스 상태를 재구성 시스템에 보내는 기능을 한다. 노트북과 PDA의 다른 디바이스는 서로 지원하지 못하는 서비스와, 디바이스에 따른 재생할 수 있는 음악의 볼륨이 다를 수 있으므로, 이에 대한 디바이스의 재구성 가능한 서비스 정보를 재구성 시스템에 전송하여, 올바른 서비스를 제공받을 수 있도록 하였다.

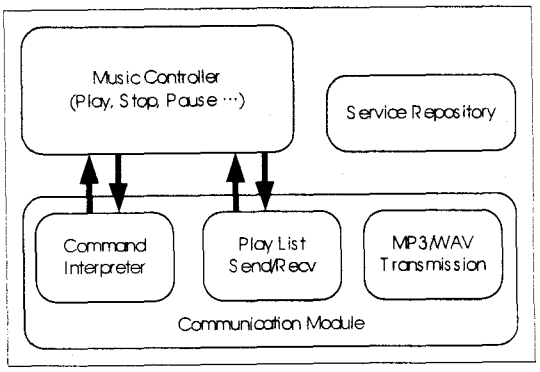

그림5. 음악 플레이어의 구조

그림6은 센서 네트워크를 통해 사용자의 상황을 파악하고, 그 위치에 해당하는 컴퓨팅 장치에서 음악 플레이어를 통해 음악을 재생하는 시나리오를 보여준다. 여기서 사용자의 위치 이동 정보는 시뮬레이션을 이용하였다. 음악 플레이어는 각 컴퓨팅 장치에 존재하고, 재구성 시스템은 센서 정보를 수집하는 서버, 즉 센서 싱크 노드와 연결된 서버에 존재한다.

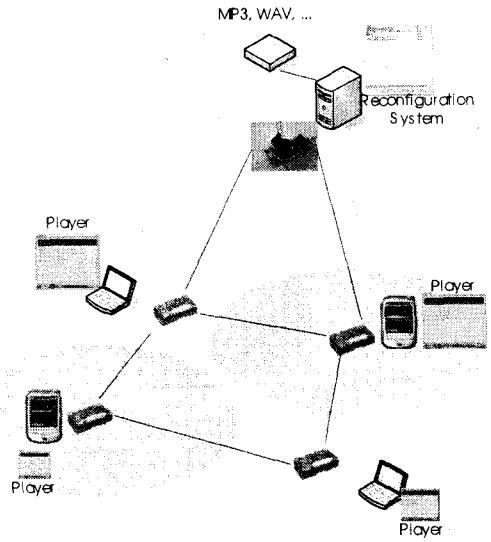

그림 6. 재구성 시스템과 어플리케이션의 구성

사용자의 위치가 변화하여 기존의 위치 영역에서 음악을 연주하던 음악 플레이어에서 사용자에게 음악을 제공하지 못할 경우에, 재구성 시스템은 기존 플레이어에 stop 명령어를 전달하고, 실행 중인 음악 리스트와 현재 실행 중인 상태 정보를 요청한다. 위의 정보를 수신한 후, 새로운 디바이스에 알맞은 형식의 음악 리스트와 서비스 정보로 재구성하여 새로운 디바이스에 전송한다. 새로운 디바이스는 새로 받은 재구성 서비스 정보를 통해 음악리스트를 수행하여, 사용자가 듣던 음악을 계속해서

제공받을 수 있도록 하였다. 새로운 디바이스에 음악 파일이 없을 경우에, 서버로부터 음악파일을 요청하고 음악파일을 수신하여 연주할 수 있도록 하였다. 위의 과정을 통해 사용자의 위치영역에 있는 다른 디바이스가 기존의 이동성이 부족한 디바이스의 단점을 보완하고, 사용자가 디바이스를 가지고 다니지 않아도 위치의 제약을 받지 않고 재구성을 이용하여 서비스를 계속해서 받을 수 있다.

5. 결과 및 향후 연구

위의 설계한 시스템과 시나리오를 통하여 사용자가 이동한 위치 영역이 변화하더라도, 사용자의 위치 영역에 서비스를 제공해 주는 디바이스가 있을 경우에 위치에 따른 제약을 받지 않고, 끊임없이 기존 서비스를 제공받을 수 있게 하였다. 위의 수행을 위해 중요한 기술은 추론과 동적 재구성 기술이다. 본 논문에서는 JESS를 이용하여 추론 모듈을 구현하였고, 서비스의 이동을 이용하여 여러 디바이스가 하나의 서비스를 끊임없이 사용자에게 제공할 수 있게 하였다. 유비쿼터스 환경에서는 컴퓨팅 디바이스가 언제 어디서나 사용자의 위치 영역에 존재하게 되므로 위와 같은 서비스의 이동을 주로 이용하는 동적 재구성 기술은 매우 유용해 질 것이다.

향후, 추론 모듈과 서비스 재구성 모듈을 개선할 것이다. 사용자의 영역에 두 개 이상의 디바이스가 존재할 경우에 현재의 상황에 따른 최적의 디바이스를 찾아야 하고, 같은 종류의 서비스를 다른 디바이스에 제공해야 하기 때문에, 시스템이 자동으로 다양한 디바이스에서 수행 가능한 형태로 재구성하는 방법을 추론하는 기술이 필요하다. 앞으로 베이지안 네트워크를 이용하여 사용자의 행동 패턴을 고려하여 재구성 서비스와 재구성할 디바이스의 선택을 더욱 효과적으로 할 수 있도록 할 것이다. 재구성 모듈은 보다 다양한 디바이스와 프로그램에 재구성 서비스를 적용할 것이다. 또한 음악 리스트와 구성 정보만 재구성 하는 것이 아니라 코덱과 같은 보다 다양한 요소를 재구성하고, 음악과 영화와 같은 다양한 미디어 사이에서도 재구성이 가능하도록 연구 및 구현을 할 것이다.

6. 레퍼런스

[1] Maarten Wegdam, " Dynamic Reconfiguration and Load Distribution in Component Middleware" , Publisher. Telematica Institut, pp. 71-112, 2003.

[2] Arianna Bassoli, Cian Cullinan, Julian Moore, Stefan Agamanolis, " TunA: A Mobile Music Experience to Foster Local Interactions" , UbiComp 2003 Fifth International Conference on Ubiquitous Computing, 2003.

[3] Hidenori Yamamoto, Shigetoshi Sameshima, Katsumi Kawano " Service reconfiguration using Super Distributed Objects(SDO) in context-aware service systems " , The 2nd IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems, pp. 163-165, 2004.

[4] Wang, Schmidt, O'Ryan, " Overview of the COBRA Component Model " , Component-based software engineering: putting the pieces together, pp. 557-571, 2001.