유비쿼터스 환경에서 상황기반의 커뮤니티 컴퓨팅 모델을 위한 멤버 프레임워크

김한욱[O1], 김희수[1], 이정태[2], 김민구[2]
[1]아주대학교 정보통신전문대학원
{hanoogi, heemanz}@ajou.ac.kr
[2]아주대학교 정보 및 컴퓨터공학부
{jungtae, minkoo}@ajou.ac.kr

Member Framework for Situation-Based Community Computing Model in Ubiquitous Environment

Han-Wook Kim[O1], Hee-Soo kim[1], Jung-Tae Lee[2], and Min-Koo Kim[2]
[1]Graduate School of Information and Communication, Ajou University
[2]Department of Information and Communication Engineering, Ajou University

요 약

유비쿼터스 지능 공간(Ubiquitous Smart Space)에서 동적으로 발생하는 다양한 문제를 협업을 통하여 해결할 수 있는 방법론으로 제안된 커뮤니티 컴퓨팅(Community Computing) 모델을 기반으로 하는 개발 도구(Community Computing Developing Tool Kit : CDTK)를 사용하면 특정 문제를 해결할 수 있는 커뮤니티 컴퓨팅 어플리케이션이 생성된다. 이 커뮤니티 컴퓨팅 어플리케이션이 실제로 유비쿼터스 지능 공간에 존재하는 uT-entity에 이식되어 동작하기 위해서 uT-entity의 종류에 상관없이 커뮤니티 컴퓨팅 어플리케이션이 배포될 수 있는 환경을 필요로 한다. 본 연구에서는 CDTK를 이용하여 생성된 커뮤니티 컴퓨팅 어플리케이션이 uT-entity에 배포되어 각 uT-entity가 커뮤니티의 멤버로 참여하여 멤버간의 협업을 통해 목적을 달성할 수 있도록 지원하는 어플리케이션 프레임워크인 멤버 프레임워크(Member Framework)를 제안한다.

1. 서 론

유비쿼터스 지능 공간에서는 서로 다른 벤더에 의해 만들어진 다양한 디바이스들이 서로 다른 네트워크 방식으로 이질적인 환경을 구성하게 된다. 본 연구에서는 네트워킹 능력, 초소형 프로세싱 능력을 가지고 있는 오브젝트를 uT-entity라 명명한다. 이런 uT-entity들은 다양한 네트워크를 통해 연결되어 있고, 환경 정보를 수집할 수 있는 센싱 능력을 가질 수 있으며, 단독으로 고유한 작업을 수행할 수 있다.

유비쿼터스 지능 공간에서 동적으로 발생하는 다양한 문제를 협업을 통하여 해결할 수 있는 방법론으로 제안된 커뮤니티 컴퓨팅 모델[1]을 기반으로 하는 MDA 방식의 CDTK를 사용하면 특정 문제를 해결할 수 있는 커뮤니티 컴퓨팅 어플리케이션이 생성된다. 이 커뮤니티 컴퓨팅 어플리케이션이 실제로 유비쿼터스 지능 공간에 존재하는 다양한 uT-entity에 배포되어 동작하기 위해서는 uT-entity의 종류에 상관없이 커뮤니티 컴퓨팅 어플리케이션이 배포될 수 있는 환경을 필요로 한다.

본 연구에서는 커뮤니티 컴퓨팅 모델을 지원하고, 이질적인 유비쿼터스 환경을 지원하며 CDTK를 이용하여 생성된 커뮤니티 컴퓨팅 어플리케이션이 uT-entity에 배포되어 각 uT-entity가 커뮤니티의 멤버로 참여하여, 멤버간의 협업을 통해 목적(Goal)을 달성할 수 있도록 하는 어플리케이션을 위한 멤버 프레임워크를 제안한다.

2. 관련연구

2.1 OSGi

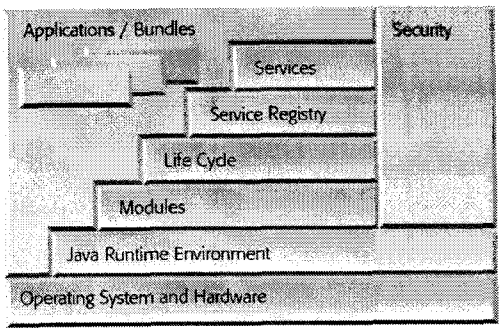

그림 1 OSGi 구성요소

OSGi(Open Service Gateway initiative)는 1993년 썬 마이크로시스템, IBM, 에릭슨 등에 의해 구성된 개방형 표준단체에서 홈네트워크를 구성하는 기술의 표준화를 제안하였다. OSGi는 번들(Bundle)를 동작하도록 하는 소프트웨어 프레임워크와 프레임워크와 통신하도록 하는 인터페이스를 정의 하고 있다. 외부망과 내부망, 각 기술

과 서비스들 간의 교량이자 관문 역할을 하게 된다.

OSGi의 구성 요소는 그림1과 같이 서비스, 번들, 프레임워크로 나눌 수 있다. 서비스는 특정 기능을 수행하는 자바 인터페이스와 실제 구현 객체로 되어 있다. 번들은 서비스 제공하기 위한 기능적 배포 단위로써 JAR파일의 형태로 존재하고 JAR파일에는 하나 이상의 서비스의 구현 객체와 리소스 파일 및 매니페스트 파일을 포함하고 있다. 프레임워크는 번들 컨텍스트라고도 하며 번들의 라이프사이클을 관리하는 번들이다.[2]

2.2 이클립스 리치클라이언트 플랫폼

이클립스는 구조화된 아키텍처로 개발자가 지원하고자 하는 툴 관련 어플리케이션의 종류에 따라 재사용될 수 있는 프레임워크를 지원한다. 리치클라이언트 어플리케이션을 개발할 때 필요한 플러그인들을 모아놓은 이클립스 플랫폼의 한 부분으로 존재한다. [3][4]

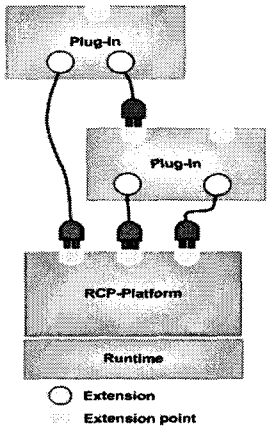

그림2 RCP의 Plug-In

그림2와 같이 이클립스의 플러그인은 리치클라이언트 플랫폼에 확장점(Extension Point)를 두어 다른 플러그인이 확장(Extension)할 수 있도록 하는 구조이다. 리치클라이언트 플랫폼으로 개발된 어플리케이션은 OSGi 런타임을 기반으로 하기 때문에 플러그인들을 결합하여 확장이 가능하고, 배포한 어플리케이션은 플러그인의 동적인 업데이트가 가능하기 때문에 추후에 유지 및 보수가 용이하다.

3. 멤버 어플리케이션

3.1 정의

그림 3과 같이 uT-entity는 외부의 개입 없이 혼자서 본래의 기능(Own Action)을 수행한다. 만약 커뮤니티가 생성되면 특정 커뮤니티가 가지고 있는 목표(Goal)를 해결하기 위해 유비쿼터스 지능 공간에 존재하는 uT-entity를 멤버화 과정을 통해 커뮤니티의 멤버로 임명한다. 이때 선택된 uT-entity에 멤버화를 위해 배포되는 어플리케이션을 멤버 어플리케이션이라고 한다.

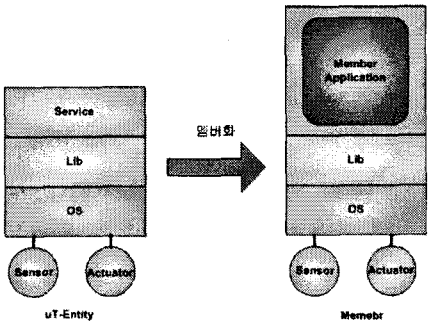

그림 3 멤버화 과정

3.2 소사이어티 멤버와 커뮤니티 멤버

위에서 말하고 있는 멤버화 과정은 소사이어티(Society) 멤버화와 커뮤니티 멤버화로 나누어진다. 즉, 유비쿼터스 지능 공간을 구성하는 다양한 uT-entity는 특정 소사이어티에 포함되면 소사이어티 멤버(Society Member)가 된다[1].

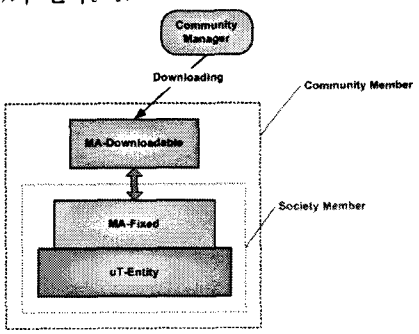

그림 4 커뮤니티 멤버와 소사이어티 멤버

멤버화된 uT-entity에 멤버 프레임워크의 고정된 부분(Fixed Part)가 배포되어 그림 4와 같이 소사이어티 멤버가 된다. 이후에 소사이어티의 멤버가 커뮤니티 매니저(Community Manager)에 의해 특정 커뮤니티의 멤버로 캐스팅되면 멤버 프레임워크의 다운로드 가능한 부분(Downloadable Part)가 커뮤니티 매니저로부터 다운로드되어 배포되면 그림4와 같이 커뮤니티 멤버가 된다.

4. 멤버 프레임워크

4.1 정의 및 요구 사항

이질적인 유비쿼터스 환경을 지원하는 멤버 어플리케이션을 보다 쉽게 개발하기 위한 소프트웨어 프레임워크를 멤버 프레임워크라 한다.

멤버 프레임워크의 요구 사항은 크게 커뮤니티 컴퓨팅 모델 지원, 유비쿼터스 지능 공간의 이질적인 환경을 지원, MDA 방식에 의한 CDTK를 지원으로 나눌 수 있다.

첫째, 커뮤니티 컴퓨팅 모델 지원해야 한다. uT-entity가 소사이어티의 멤버로 등록되어 고유의 기

능(Own Action)을 수행하다가 특정 목표(Goal)를 해결하기 위한 커뮤니티가 생성되면, 커뮤니티 매니저의 요청에 의해 커뮤니티에 참여하게 된다. 이때, 역할(Role)을 동적으로 다운로딩하여 커뮤니티에 참여하게 된다. 커뮤니티로부터 상황정보(Situation)를 받아서 그 상황정보에 맞는 액션을 수행하는 Situation-Aware를 만족해야 한다. 또한 여러 커뮤니티의 멤버로 참여가 가능해야 한다.

둘째, 유비쿼터스 환경에서의 이질적인 환경을 지원해야 한다. 유비쿼터스 지능 공간에서 uT-entity는 센서, 엑추에이터, 커뮤니티 접속 방식이 다양하고 각각 다를 수 있다.

셋째, MDA방식에 의한 CDTK를 지원해야 한다. 그림 5와 같이 CDTK는 CCM, CIM-PI, CIM-PS 단계를 거쳐 프레임워크와 플러그인들이 결합된 멤버 어플리케이션이 생성된다. 이를 지원하기 위해서는 표준화된 어플리케이션 형식이 필요하며 동기화 및 비동기화 방식의 다양한 데이터 전달 방식을 지원해야 한다.

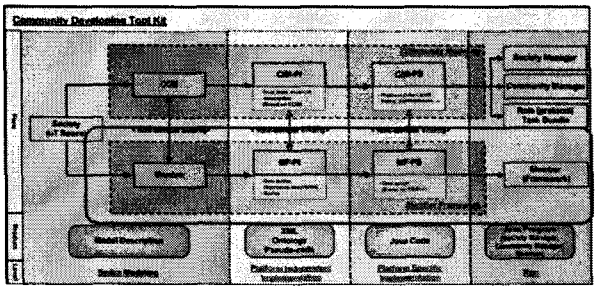

그림 5 CDTK 구성도

4.2 아키텍처

멤버 프레임워크는 그림6과 같이 소사이어티 멤버가 되기 위해 미리 uT-entity에 이식되는 멤버 어플리케이션 고정된 부분(Fixed Part)와 커뮤니티 매니저에 의해 다운로드되는 멤버 어플리케이션 다운로드 가능한 부분(Downloadable Part)로 나눌 수 있다. 여러 커뮤니티에 참여시 각 커뮤니티 마다 멤버 어플리케이션 다운로드 가능한 부분을 다운로딩한다.

Fixed Part는 크게 모든 어플리케이션에 공통된 기능이 포함되는 모듈인 Main, Downloadable Manager, Role Member Binder, Function Mapper, Society Binder, Configuration Manager, Message Broker와 uT-entity 마다 다르게 장착되는 모듈인 Sensor Driver, Actuator Driver, Communication Driver, Own Action, Context Manager 로 나눌 수 있다. 각각 플러그인의 주요기능은 표1과 같다.

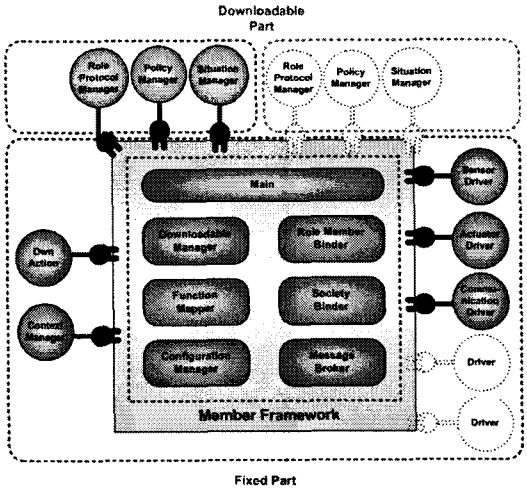

그림 6 멤버프레임워크 아키텍처

표 1 Plug-In의 기능

플러그인		설명
Fixed Part	Main	Member Application의 시작점, Community 생성 전에 Own Action의 본래의 임무를 수행하고, Community 생성 될 때 커뮤니티의 Role을 수행하는 두 가지 상태를 스위칭하는 역할을 하는 모듈
	Society Binder	uT-entity가 Society의 멤버화 되기 위한 바인딩을 하는 모듈
	Role Member Binder	Community Manager와 Society/Community Member 간에 Role을 통한 바인딩 작업을 지원하는 모듈
	Downloadable Manager	Community Manager로부터 Downloadable Part를 다운로드 및 설치하는 모듈
	Function Mapper	Role Protocol의 액션과 Own Action의 액션을 매핑하는 모듈
	Configuration Manager	Plug-In의 상태 정보를 저장
	Message Broker	내부 플러그인들 간의 메시지 교환 통로 역할을 하는 모듈
	Context Manager	Sensor로부터 Raw Data와 Own Action의 결과로 Member의 Context를 생성하고 관리하는 모듈
	Own Actions	uT-entity가 고유의 임무를 수행하는 액션들이 모인 모듈
	Sensor Driver	uT-entity에 장착된 sensor device를 작동할 수 있는 driver
	Actuator Driver	uT-entity에 장착된 actuator device를 작동할 수 있는 driver
	Communication Driver	Network 접속장치와의 창구 역할을 하는 Driver
Downloadable Part	Role Protocol	Role기반으로 프로그램을 실행하기 위한 모듈
	Policy Manager	Situation간의 충돌과 우선순위를 정하는 Policy를 실현시켜주는 모듈
	Situation Manager	Member의 Context 를 Community Manager에게 송신하고 Community의 Situation을 수신하기 위한 모듈

멤버 프레임워크의 아키텍처는 요구사항을 다음과 같이 만족하고 있다.

첫째, 커뮤니티 컴퓨팅 모델을 지원하기 위해서 멤버 프레임워크의 중심에 있는 Society Binder, Role-Member Binder, Downloadable Manager가 커뮤니티 컴퓨팅의 소사이어티 멤버화와 커뮤니티 멤버화, RoleProtocol 다운로드를 지원하게 된다. 역할(Role)을 동적으로 다운로딩하여 커뮤니티에 참여하게 됨으로 서비스의 점진적인 추가기능을 제공하게 되고 여러 커뮤니티에 멤버로 참여가 가능하게 된다.

둘째, 유비쿼터스 지능 공간의 이질적인 환경을 지원할 수 있다. uT-Entity는 센서, 엑추에이터, 커뮤니케이션 방식, Own Action, 컨텍스트 매니저를 플러그인 방식으로 장착할 수 있고 추후에 플러그인을 교체할 수도 있다. 이는 유비쿼터스 지능 공간의 다양한 uT-Entity를 지원할 수 있는 장점이 있다.

셋째, MDA방식에 의한 CDTK를 지원해야 한다. 각 플러그인의 구조는 CDTK에 의해 플러그인이 생성되기 위해서는 그림7과 같이 공통적인 형식이 필요하다. 플러그인의 구성은 플러그인의 고유의 기능과 메시지를 전달받는 메시지 큐(Message Queue), 멤버프레임워크 플러그인에 확장(Extension)을 담당하는 부분으로 나눌 수 있다. 모든 플러그인들은 하나의 쓰레드로 동작하고 주기적으로 자신의 메시지 큐를 폴링하여 들어온 메시지를 처리한다. 각각 플러그인들간의 메시지는 프레임워크 플러그인의 메시지브로커를 통해 전달을 할 수 있으며, 메시지의 타입에 따라 메시지의 전달 방식도 이벤트를 이용한 비동기화 방식과 Request/Response을 이용한 동기화 방식 모두 지원하게 된다. CDTK를 이용하여 멤버 어플리케이션을 생성하게 되면 개발비용이 적게 들고 개발속도가 빨라지게 되는 장점이 있다.

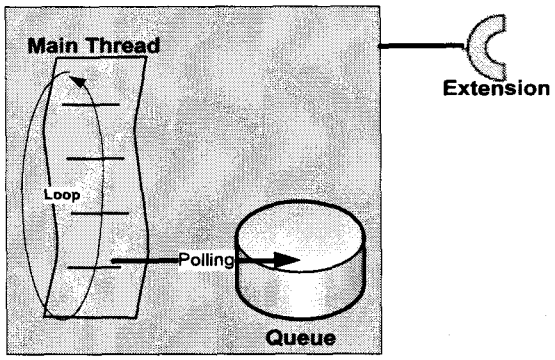

그림7 플러그인의 공통구조

4.3 상황기반의 Role-Protocol 수행

상황기반(Situation-Based) 커뮤니티 컴퓨팅 모델은 커뮤니티의 상황에 따라 멤버들이 각각의 임무를 수행하여 커뮤니티의 목표를 달성하게 하는 모델이다. 커뮤니티 매니저에 의해 멤버에게 전달되는 상황은 각 멤버가 해야 할 액션들로 이루어져 있다.[1]

멤버의 컨텍스트는 그림8과 같이 센서의 Raw Data와 OwnAction의 결과를 바탕으로 컨텍스트 매니저에 의해 생성이 되며, 시추에이션 매니저를 거쳐 커뮤니티 매니저에 전달되게 된다. 커뮤니티 매니저는 커뮤니티의 모든 멤버의 컨텍스트를 확인하여 시추에이션을 결정하게 된다.

커뮤니티 매니저는 그림8과 같이 시추에이션을 각 멤버의 시추에이션 매니저에 전달하게 되면, 멤버는 멤버화 과정에서 다운로딩 받은 Role-Protocol에 정의된 시추에이션에 따라 액션들을 Own Action을 통해 실행하게 된다. 이런 시추에이션의 변화에 따라 각 멤버의 액션들을 수행하여 멤버간의 협업을 통해 커뮤니티의 목표(Goal)를 성취하게 된다.

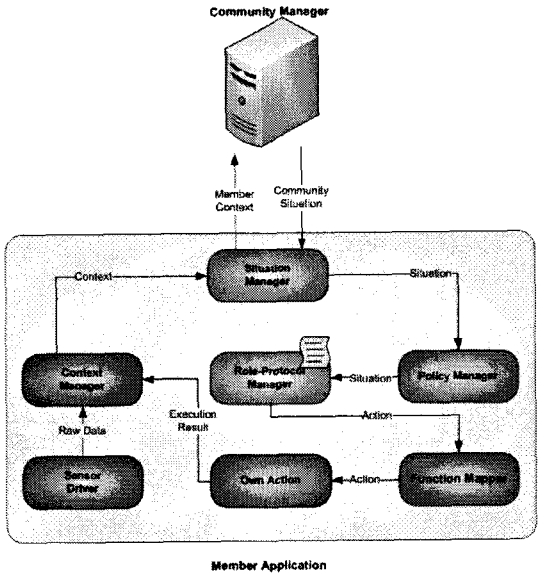

그림8 Situation과 Role-Protocol 처리 과정

5. 결론 및 향후 연구 방향

본 연구는 유비쿼터스 지능 공간에서 발생하는 다양한 문제를 협업을 통해 해결하려는 커뮤니티 모델을 실제화하는 연구로써 유비쿼터스 지능 공간에 존재하는 이질적인 형태의 uT-entity에 이식 가능하도록 지원하는 멤버 어플리케이션을 위한 멤버 프레임워크를 제안한다. 멤버 어플리케이션은 MDA방식에 의한 커뮤니티 컴퓨팅 개발툴인 CDTK를 이용하여 생성되고, 이를 사용하여 다양한 uT-entity에 커뮤니티 컴퓨팅 멤버 어플리케이션에 배포되어 실제로 커뮤니티 컴퓨팅 구현에 도움을 줄 것으로 판단된다. 향후 연구는 프레임워크 아키텍처에 대한 보완이 이루어져야 할 것이며, 다양한 시나리오를 바탕으로 CDTK로부터 다양한 이질적인 환경을 지원할 수 있는 멤버 어플리케이션이 생성되는지를 연구해야 할 것이다. 또한, 멤버의 인텔리전스에 대한 연구가 진행되어야 할 것이다.

참고문헌

[1] Y. Jung, J. Lee, M. Kim. "Multi-agent based community computing system development with the model driven architecture", AAMAS 2006, pp.1329-1331, 2006.
[2] OSGi, http://www.osgi.org
[3] Eclipse.org, "Notes on the Eclipse Plug-in Architecture", http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin_architecture.html, 2003.
[4] Eclipse.org, "Dev guide: Updating a product or extension", http://help.eclipse.org/help31/index.jsp? topic=/org.eclipse.platform.doc.isv/guide/product_update.htm, 2006.

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