

커뮤니티 컴퓨팅 모델에 기반한 개발도구 구현*

김준영^o, 정유나, 김민구

아주대학교 정보통신전문대학원
{tristan^o, serazade, minkoo}@ajou.ac.kr

The Implementation of CDTK(Community Developing Tool Kit) Based-on Model Driven Architecture

Junyoung Kim^o, Yuna Jung, Minkoo Kim
Graduate School of Information and Communication, Ajou University

요 약

커뮤니티 컴퓨팅 시스템은 3단계 모델(CCM, CIM-PI, CIM-PS)로 이루어져 있다. 모델은 이 전의 모델을 기반으로 틀의 지원을 받아 코드를 상세화하면서 실제 구현단계에 이르게 된다. 3가지 모델 중 가장 먼저 기술해야 하는 CCM(Community Computing Model)은 커뮤니티 컴퓨팅 시스템에 대한 가장 높은 수준의 추상화 모델로서, 시스템의 환경과 요구사항을 기술하는 부분이다. 기술된 CCM을 기반으로 생성되는 CIM-PI(Platform Independent Community Implementation Model)에서는 시스템의 구현을 고려하여 컴퓨팅 요소들과 또 이들 간에 생길 수 있는 협업관계를 자세히 기술한다. 이렇게 기술된 CIM-PI를 멀티 에이전트 플랫폼 위에서 작동할 수 있도록 CDTK를 이용해 변환해 놓은 것이 CIM-PS(Platform Specific Community Implementation Model)이다. 본 논문에서는 커뮤니티 컴퓨팅 시스템 개발을 더욱 쉽고 체계적으로 개발하기 위해 만든 CDTK와 이 개발 틀을 통해 얻을 수 있을 수 있었던 여러 장점들에 대해 소개한다. CDTK를 이용한 적용사례로 유비쿼터스 환경내에서 발생할 수 있는 가상시나리오에 적용하여 CDTK의 실현성과 효율성을 검증해보았다.

1. 서 론

커뮤니티 컴퓨팅에서의 커뮤니티(Community)라 함은 목적 지향적 협업 1)조직의 동적 생성과 해체 및 조직 내에서 또는 조직들 간의 동적인 협업 등의 유비쿼터스 특징들을 표현하기 위한 조직을 뜻한다. 커뮤니티를 기반한 커뮤니티 컴퓨팅(Community Computing)은 다음과 같이 정의할 수 있을 것이다.

“유비쿼터스 환경위의 이질적인 하드웨어와 에이전트간의 협업과 의사소통이 동적이고 연속적으로 이루어 질 수 있도록 지원해 줄 수 있는 하나의 소프트웨어 플랫폼”

우리는 이런 커뮤니티 컴퓨팅 시스템 개발을 위해서 모델 기반 방식(Model Driven Architecture)[2]의 개발 방법을 사용하고 한다. 이 방법을 통하여 우리는 모델의 자동화와 변환(transformation)을 통해 여러 플랫폼을 쉽게 지원하고 개발자의 입장에서 시간을 많이 소비하는 코드 작성부분을 줄여 줄 수 있도록 하고자 한다.

이에 우리는 제안한 모델, 즉, CCM, CIM-PI, CIM-PS를 기반으로 하여, 개발자들이 구현에 관련된 복잡한 문제에서 벗어나 시스템을 고수준의 추상화 개념으로부터 시스템에 적용될 소스 코드를 얻을 때의 과정을 분리하고, 또 일정부분은 자동화함으로써 보다 쉽고 체계적으로

개발 할 수 있도록 개발 과정 전반을 지원해줄 수 있는 커뮤니티 개발 킷 (CDTK: Community Developing Tool Kit)을 소개하도록 하겠다. CDTK를 이용한 적용사례로 유비쿼터스 환경내에서 발생할 수 있는 가상시나리오에 적용하여 CDTK의 실현성과 효율성을 검증해보았다.

2. 커뮤니티 컴퓨팅

2-1. 커뮤니티 컴퓨팅의 정의

커뮤니티 컴퓨팅은 동적으로 변화하는 유비쿼터스 환경의 특징을 반영하여 초기 고수준의 추상화 모델단계에서부터 실제 환경에 배포를 위한 구현 단계수준까지를 모델 중심으로 디자인하여 유비쿼터스 환경을 멀티 에이전트 기반의 시스템으로 효율적으로 표현 및 구현하기 위해 제안된 모델링 기법이다. 이 모델을 통해 우리는 특정 유비쿼터스 도메인을 하나의 사회(Society)로 보고 이 사회와 사회를 구성하는 구성원(Member)을 기술한 후 특정한 목표(Goal)가 발생했을 때 이를 해결하기 위해 동적으로 생성되는 커뮤니티(Community)와 이 커뮤니티를 구성하는 구성원 간의 상호작용(Interaction)에 통해 목표를 달성하고 있다고 제안하고 있다.

2-2. 모델기반방식(Model Driven Architecture)의 이점

모델 기반 방식은 최근에 각광받고 있는 개발 방식으로써, 1989년에 설립되어 객체 기술에 대한 사용증진 및 표준안을 만들고 있는 산업계 컨소시엄인 OMG를 통해 처음 알려지게 되었다.

모델 기반 방식의 기본적인 개발방식은 비즈니스 프로세스나 소프트웨어에 대한 기술서를 바탕으로 추상적인 모델을 추출하고, 추상적인 모델을 실행 가능한 구현 모

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

델로 변환시키는 과정을 통해 전체 시스템을 개발해 나가는 것이다[3].

이런 개발 과정에 관한 표준을 적용해서 우리는 MDA를 적용한 시스템은 모델의 자동화와 변환(transformation)을 통해 여러 플랫폼을 쉽게 지원할 수 있게 하며, 개발자의 입장에서는 반복되는 코드 작성부분에서 시간을 줄일 수 있으며, 개발 프로세스의 측면에서도 유지보수가 쉬워 품질 관리를 수월하게 할 수 있을 것이다.

2-3. 모델기반방식의 커뮤니티 컴퓨팅 시스템의 모델

위에서 설명한 모델 기반 개발방법을 기반으로 하여, <그림 1>과 같이 커뮤니티 컴퓨팅 시스템을 3단계과정, 즉, CCM(Community Computing Model), CIM-PI(Platform Independent Community Implementation Model), CIM-PS(Platform Specific Community Implementation Model)으로 모델을 상세화하면서 시스템 구현을 해 나가게 된다.

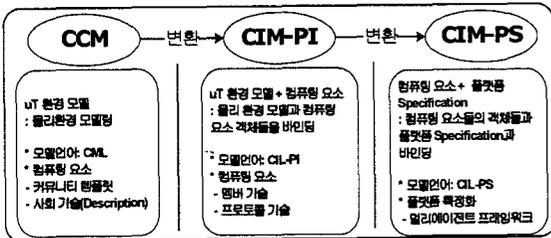


그림 1 커뮤니티 컴퓨팅 시스템 개발을 위한 추상화 모델들

2-3-1. 커뮤니티 컴퓨팅 모델(CCM)

CCM은 커뮤니티 컴퓨팅 시스템에 대한 가장 높은 수준의 추상화 모델로서, 시스템의 환경과 요구사항을 기술하는 것을 목적으로 한다. CCM을 기술하기 위해서 자체적으로 정의한 CML (Community Modeling Language)를 사용한다.

2-3-2. 플랫폼 독립적 커뮤니티 컴퓨팅 구축 모델(CIM-PI)

CIM-PI는 시스템의 구현을 고려하여 컴퓨팅 요소들과 이들 간의 협업관계를 자세히 기술하는 것을 목적으로 한다. 이때, 특정 에이전트 플랫폼에 국한되지 않고 어떠한 플랫폼에도 적용될 수 있는 수준으로 기술하도록 한다. CIM-PI는 CCM의 커뮤니티와 사회 기술(Society Description)을 세부적인 내용으로 확장한다.

2-3-3. 플랫폼 의존적 커뮤니티 컴퓨팅 구축 모델(CIM-PS)

CIM-PS는 CIM-PI를 특정 플랫폼 환경에서 어떻게 구현할지를 기술한 것으로, 여기에서의 플랫폼은 JADE[2]와 같은 에이전트 플랫폼에 코드를 변환기 시켜 작성한 것을 CIM-PS라고 할 수 있다.

3. 커뮤니티 개발도구 (Community Developing Tool Kit)

CDTK는 커뮤니티 컴퓨팅을 개발하기 위한 개발도구로써 각 모델언어로 기술된 커뮤니티 모델들의 기술, 다른 모델로의 자동변환 및 특정 장치 혹은 에이전트로의 배포(Deployment)를 쉽고 빠르게 해주기 위해 개발이 되었다. CDTK를 이용해 사용자는 자동으로 변환된 모델에 약간의 수정을 하는 것으로 시스템을 개발할 수 있게 되며, 또한 상대적으로 편리하게 모델을 완성할 수 있어 커뮤니티 컴퓨팅 시스템 개발 시간을 크게 단축시킬 수

가 있다.

3-1. 시스템 구조 (System Architecture)

<그림 2>는 CDTK의 시스템 구조를 나타내고 있다. CDTK는 GUI를 위에서 커뮤니티 모델언어를 사용해 코드 작성을 할 수 있도록 하고 있으며, 기존의 파일을 읽어서 편집하거나 다른 커뮤니티 모델로 자동으로 변환시킬 수 있는 기능을 제공한다. 또한 최종적으로 플랫폼에 맞게 작성된 에이전트코드(CIM-PS)를 특정 장치로 배포할 수 있는 기능을 제공한다. 이를 위해 CDTK는 사용자와의 상호작용을 위한 GUI 인터페이스를 제공하며, 내부적으로 시스템 구조는 크게 커뮤니티 모델 생성을 위한 모델빌더 (Model Builder) 모듈과 이로 인해 생성되는 커뮤니티 모델(Community Model), 생성된 모델을 변환시키고 다른 모델 스크립트로 저장할 수 있는 기능을 제공하는 변환기 (Convertor) 모듈, 그리고 장치로의 코드 배포를 위한 배포기(Deployer) 모듈로 구성된다.

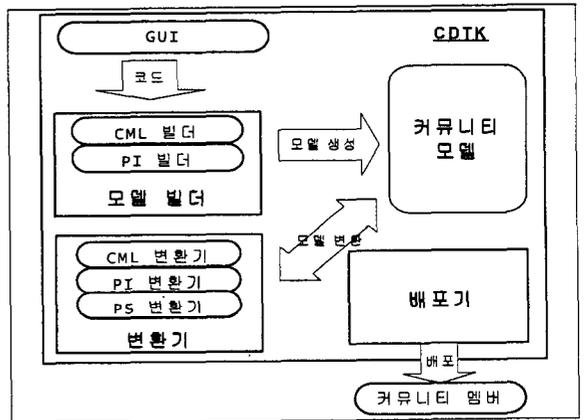


그림 2 CDTK 시스템 구조도

모델 빌더 (Model Builder)

모델 빌더는 CCM 혹은 CIM-PI의 내용으로부터 커뮤니티 모델을 만들어내는 클래스들의 집합이다. 크게 CML빌더와 PI빌더로 이루어져있으며 CML빌더는 CML로부터 커뮤니티 모델을 생성하고, PI빌더는 CIM-PI로부터 커뮤니티 모델을 생성한다. 마지막으로 변환기 클래스는 생성된 커뮤니티 모델로부터 CIM-PI 혹은 CIM-PS를 생성해주는 클래스들의 집합이다.

커뮤니티 모델 (Community Model)

커뮤니티 모델은 CCM에 기술된 사회, 커뮤니티, 멤버, 프로토콜을 저장하기 위한 클래스들의 집합이다. 커뮤니티 모델이 각각의 클래스들을 포함하는 멤버를 가지며, 모델 빌더에 의해 생성되며 모델 간 변환을 위한 기초가 된다.

변환기 (Convertor)

변환기 모듈은 CML 변환기와 PI 변환기로 이루어져있다. 모델의 변환은 CML→CIM→PI→CIM-PS의 순서로 이루어지는데 CML→CIM-PI로의 변환과정에서 CML 변환기가 이용되고, CIM-PI→CIM-PS로의 변환과정에서 PI 변환기가 이용된다. 현재의 CDTK는 멀티 에이전트 플랫폼

인 JADE위에서 구현이 되어있다. 모델 빌더 모듈과 변환기 모듈은 커뮤니티 모델링 언어의 레벨에 따라 해당 언어를 위한 하위 모듈을 가지게 되지만, 내부적으로는 하나의 커뮤니티 모델을 공유하게 된다.

배포기 (Deployer)

배포기 모듈은 CDTK를 이용해 최종적으로 만들어진 캐스팅된 클래스들을 적합한 멤버들에게 배포, 동작하게 하는 역할을 한다.

4. CDTK를 이용한 시스템 구현

4-1. 아이 보호 시나리오

시스템 구현을 가정한 시나리오인 아이보호 시나리오를 자세히 설명하자면 다음과 같다. 어린 아이에게 있어 안전을 보장받을 수 있는 위치 범위가 있다고 하고, 이를 벗어나면 안전을 위협받을 수도 있는 위험 범위에 있다고 정의한다. 이러한 위험범위의 수준은 다양하게 나누어 질 수 있겠다. 아이가 집을 나설 때 아이의 엄마는 아이의 정보를 담은 스마트벨트 (Smartbelt)를 채워주고 그때부터 스마트벨트는 유비쿼터스 환경에서 아이를 대변하는 디바이스가 된다. 집을 나선 아이가 근처의 안전범위 내에 있다가 주택가를 벗어나 차가 다니는 위험범위로 들어가게 되면, 스마트벨트가 현재 위치가 위험범위를 감지하고, 스마트벨트의 요청을 받아 시스템을 통해 아이보호 커뮤니티를 생성한다. 아이보호 커뮤니티가 생성되면 아이가 위험 범위에 있다는 사실을 가족에게 알리고, 아이가 데려올 것인지를 묻는다. 이때 집에 있는 아이의 엄마가 아이를 데려오고자 하면, 커뮤니티는 자신에 속한 가족과 친한 이웃들 중에서 아이의 위치에 가장 가까이에 있는 사람을 선택하고 아이를 데려올 수 있는지 의향을 묻는다. 가까이에 있는 이웃이 이러한 요청을 수락하여 아이를 안전하게 집으로 데려오면 커뮤니티의 목적이 달성되고, 아이보호 커뮤니티는 소멸된다.

4-2. CDTK를 이용한 시나리오 구현

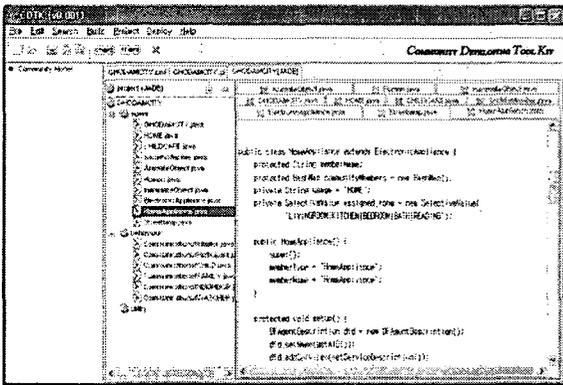


그림 3 CIM-PI 모델로부터 변환된 JADE 기반의 CIM-PS

먼저 커뮤니티 컴퓨팅 시스템을 기술하기 위한 최상위 추상화 모델인 CCM을 자체적으로 정의한 CML을 가지고 기술한다. 이렇게 기술된 CCM 모델을 가지고 CDTK를 통해 CIM-PI로 변환하게 된다. CIM-PI 모델에서는 CCM단위에서 기술하지 못한 시스템 세부적인 내용을 추가적으로 기술하게 된다. 변환, 수정한 CIM-PI 모델을 통해

CIM-PS 모델로 다시 한번 CDTK를 통해 멀티 에이전트 플랫폼위에 동작할 수 있도록 변환하게 되며, <그림 3>은 최종적으로 작성된 CIM-PS의 클래스들 중 하나이다. 현재 우리는 멀티 에이전트 플랫폼으로 JADE를 사용하고 있다. 이렇게 작성된 파일들을 배포기를 통해 파일에 적합한 커뮤니티 멤버들에게 배포하고 커뮤니티를 동작하게 된다.

5. 결과 및 결론

아이 보호 시나리오에 대한 시스템을 커뮤니티 컴퓨팅 시스템 개발 도구를 이용하여 실제로 개발해 본 결과 우리는 다음과 같은 이점을 얻을 수가 있었다.

- 모델의 자동화와 변환(transformation)을 통해 여러 플랫폼을 쉽게 지원하고 개발자의 입장에서 시간을 많이 소비하는 코드 작성 부분을 줄일 수 있다. 즉, 사용자는 본 도구를 이용하여 자동으로 변환된 모델에 약간의 수정을 하는 것으로 시스템을 개발함으로써 상대적으로 편리하게 모델을 완성할 수 있어 커뮤니티 시스템 개발 시간을 크게 단축할 수 있게 된다.
- 개발 프로세스의 측면에서도 손쉬운 유지보수를 통해 품질 관리를 수월하게 할 수 있다.
- 구현 시에 유연한 컨트롤이나 커스터마이징 가능하도록 하였다.

이러한 장점으로 인하여 기존의 유비쿼터스 시스템을 개발하는 방식에 비해 커뮤니티라는 일관된 개념을 가지고 사고를 분리하여 체계적으로 진행할 수 있어서 개발 효율성을 높이고 있다고 볼 수 있겠다.

6. 향후 계획

CDTK는 앞으로 이클립스(Eclipse)[3] 툴 위로 플러그인 형식으로 올려질 예정이다. 커뮤니티 컴퓨팅 모델 개선에 더 시간을 집중하고 코드 작업을 위해 필요한 코드 인덴테이션, 하이라이팅, 자바 파일 디버깅, 컴파일등의 작업은 이클립스가 IDE로써 제공해주는 많은 이점을 통해 이용하기 위해서이다. 또한, 시나리오 구현을 위한 시스템 자체가 소프트웨어적인 가정으로만 설계되었기 때문에 실제 환경에서의 시뮬레이션이 필요할 것이다. 그리고, JADE 플랫폼외의 다른 멀티 에이전트 플랫폼위에도 배포(Deployment)를 고려중이다.

7. 참고 문헌

[1] Yuna Jung, Jungtae Lee, Minkoo Kim. "Multi-agent based Community Computing System Development with the Model Driven Architecture" AAMAS, 2006.
 [2] Site : www.omg.org/mda/
 [3] Site : <http://www.zdnet.co.kr/itbiz/reports/trend/0,3903,4651,39130721,00.htm>
 [3] F. Bellifemine, A. Poggi, and G. Rimassa. "JADE - A FIFA-compliant Agent Framework" In Proc. of Practical Application of Intelligent Agents and MultiAgents (PAAM' 99), London, UK, April, p.97-10 (1999)
 [4] Site : <http://www.eclipse.org/>