

MDA에 기반한 커뮤니티 컴퓨팅 시스템 개발을 위한 모델 변환

김성택*, 김민구**

*아주대학교 컴퓨터공학과

**아주대학교 정보 및 컴퓨터공학부

e-mail:{blackie, minkoo}@ajou.ac.kr

Model Transformation for Community Computing System based on MDA

Sung-taeg Kim*, Min-koo Kim**

*Computer Engineering, Ajou University

**Information and Computer Science, Ajou University

요 약

유비쿼터스 환경에서 서비스를 개발하는데 협업모델이 중요한 주제로 연구되어 왔다. 본 연구에서는 이를 위한 방법으로 커뮤니티 컴퓨팅 모델을 MDA(Model Driven Architecture)에 기반하여 개발하고 있다. MDA에 기반한 커뮤니티 컴퓨팅 모델을 PICM(Platform Independent Community Model)에서 PSCM(Platform Specific Community Model)을 거쳐 최종 프로그램으로 개발된다. 본 논문에서는 PICM에서 PSCM으로 변환되는 방법을 규칙에 기반하여 제안하고 이를 구현한다.

1. 서론

MDE(Model Driven Engineering)는 새롭게 떠오르는 소프트웨어 개발 방법으로 주목받고 있다. 모델변환을 통해서 시스템을 구현하는 과정을 자동화함으로써 모델의 재사용성을 높여주고 개발과정을 간편하게 해준다. MDE를 이용한 개발 방법을 적용한 기존연구[2,3]에서 발견되고 있다. 이런 연구들은 MDE를 복잡한 멀티에이전트 시스템을 구현하는 방법론으로, 유비쿼터스 시스템을 구축하는 수단으로 적용되고 있다. 이러한 시도들과 개발방법이 추상화 단계가 고도화되는 방향으로 발전해온 사실을 볼 때, 모델 중심 개발방법이 향후 주류로 자리매김할 가능성이 높다고 판단된다[8].

잘 알려진 MDE의 한 예가 OMG(Object Management Group)의 MDA(Model Driven Architecture)[1]이다. MDA는 모델을 PIM(Platform Independent Model)과 PSM(Platform Specific Model)으로 구분하고 있다. 이것은 플랫폼에 따른 구분으로 PIM에서 PSM으로의 모델변환 과정을 거치고 PSM을 실행 가능한 코드로의 변환을 통해 MDA는 개발 과정을 자동화한다. 이러한 모델변환을 위해서 모델의 추상화단계의 구분이 필요하며 MOF(Meta-Object Facility)와 UML(Unified Modeling Language)에서 정의하고 있다.

본 논문에서는 멀티에이전트 시스템에 기반한 협업모델인 커뮤니티 컴퓨팅 시스템[4]을 개발하는 과정을 MDA에

적용하는 과정에서 모델 변환을 목적으로 한다. 커뮤니티 컴퓨팅 모델은 MDA를 고려하여 최초 PICM(Platform Independent Community Model)을 개발했고, 이것은 모델 변환을 통해 PSCM(Platform Specific Community Model)이 된다. 최종적으로 PSCM은 실행 가능한 프로그램으로 변환될 수 있다. 커뮤니티 컴퓨팅 모델을 MDA에 적용하는 것을 통해 복잡한 시스템을 구현하는 과정을 자동화하고 유지 보수 과정에서 모델의 재사용성을 높일 수 있다.

본 논문은 다음과 같이 구성된다. 2장은 관련연구를 설명하고 이전 연구에서 수행한 커뮤니티 컴퓨팅을 정리한다. 3장은 커뮤니티 컴퓨팅 모델을 설명하고, 4장에서 본 연구가 제안하는 모델변환에 필요한 환경과 방법을 설명한다. 5장에서는 본 연구의 결론과 향후 연구 방향을 제시할 것이다.

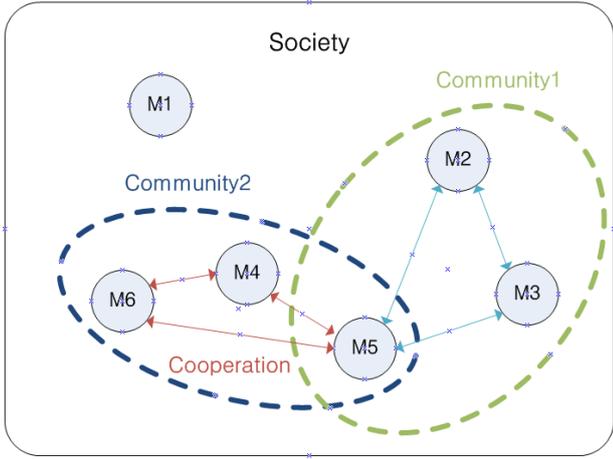
2. 관련연구

2.1 커뮤니티 컴퓨팅

커뮤니티 컴퓨팅은 유비쿼터스 컴퓨팅 환경에서 사용자에게 제공되는 서비스가 환경을 구성하는 컴퓨팅 요소들의 협업에 의해 제공된다는 사실에 입각하여 시스템을 개발하고, 개발된 시스템의 동작을 설명하기 위해 제안되었다. 커뮤니티를 메타포(metaphor)로 하여 협업을 하는 컴퓨팅의 개념으로 소개되었다. 커뮤니티는 사람 간의 소셜 네트워크가 아닌 유비쿼터스 환경에서 제공되는 서비스 또는 디바이스들 간의 자율적인 협업을 의미한다. 커뮤니티 컴퓨팅 환경에서는 발생하는 문제를 해결하기 위해 필요한 컴퓨팅 요소들이 (그림 1)과 같이 M2, M3 등과 같

본 연구는 지식경제 프론티어 기술개발사업의 일환으로 추진되고 있는 지식 경제부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술개발사업의 지원에 의한 것임.

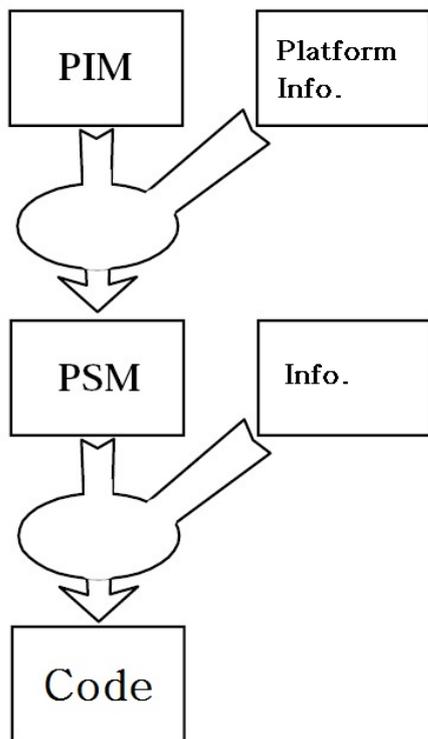
은 디바이스들이 커뮤니티를 형성, 상호 협업한다.



(그림1) 커뮤니티 컴퓨팅 뷰

2.2 MDA(Model Driven Architecture)

MDA는 OMG(Object Management Group)의 표준으로 소프트웨어 시스템을 개발하기 위한 소프트웨어 디자인 방법이다. 이는 모델로 표현되는 개발과정의 구조적 가이드라인을 아래 (그림 2)의 변환으로 제공한다.

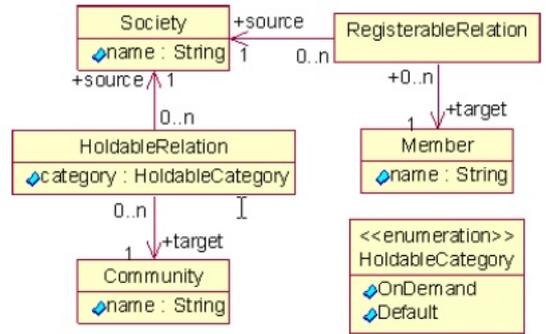


(그림 2) MDA변환

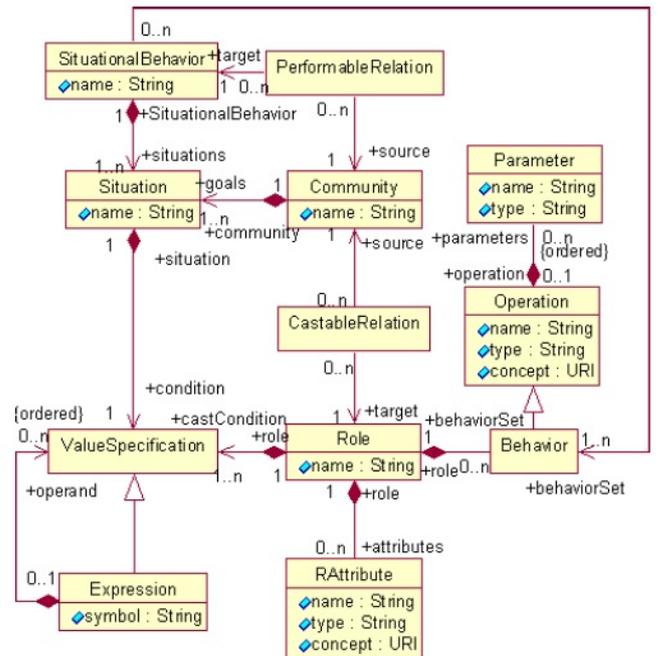
여기서 PIM(Platform Independent Model)은 시스템이 구현되는 플랫폼의 의존성에 독립적으로 디자인된 모델을 의미한다. PSM(Platform Specific Mode)은 PIM에서 보다 세부적인 구현과 관련된 플랫폼 정보를 포함하고 있는 모델이며, PSM은 변환을 통해 실행 가능한 코드로 생성된다.

3. 커뮤니티 컴퓨팅 메타모델

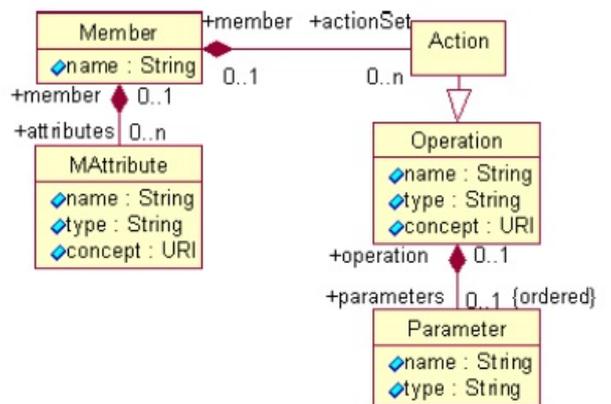
커뮤니티 컴퓨팅을 구현하기 위한 모델은 관심의 분리 (Separation of Concern)을 고려하여 소사이어티 모델과, 커뮤니티 모델, 멤버 모델로 구분된다. (그림 3~5)는 각각 소사이어티, 커뮤니티, 멤버 모델을 표현하기 위한 메타모델[5]들을 보여준다. 각 모델의 구성요소들은 <표 1>과 같이 설명된다.



(그림 3)PICM 소사이어티 메타모델



(그림 4) PICM 커뮤니티 메타모델



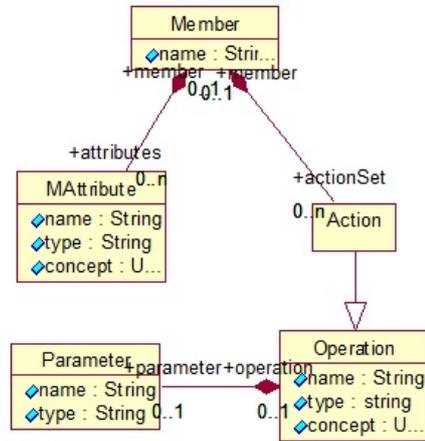
(그림 5) PICM 멤버 메타모델

메타모델	요소	설명
Society Meta-Model	Society	유비쿼터스 컴퓨팅 시스템을 표현하는 요소
	Holdable Relation	Society와 그것이 생성할 수 있는 Community와의 관계를 표현
	Hodable Catagory	Community 타입을 설명
	Registerable Relation	Society와 그것이 포함할 수 있는 Member와의 관계를 표현
Community Meta-Model	Community	유비쿼터스 컴퓨팅 시스템의 서비스를 표현하는 요소
	Situational Behavior	커뮤니티 상황에 따른 역할의 임무
	Performable Relation	Community와 SituationalBehavior의 관계를 표현
	Situation	Community 내부상황 표현
	Role	Community가 필요로하는 역할
	Castable Relation	Community와 Role과의 관계를 표현
	Value Specification	Role의 캐스팅 조건과 Community의 Situation을 표현
	Expression	Role의 캐스팅 조건과 Community의 Situation 표현
	RAttribute	Role의 정보를 표현
Behavior	Role이 수행하는 행동을 표현	
Member Meta-Model	Member	컴퓨팅 장치를 표현하는 요소
	MAttribute	Member가 제공할 수 있는 정보 표현
	Action	Member가 수행할 수 있는 행동을 표현
Common	Parameter	멤버 메타모델의 Action과 커뮤니티 메타모델의 Behavior가 취하는 파라미터
	Operation	멤버 메타모델의 Action과 커뮤니티 메타모델의 Behavior의 메타모델

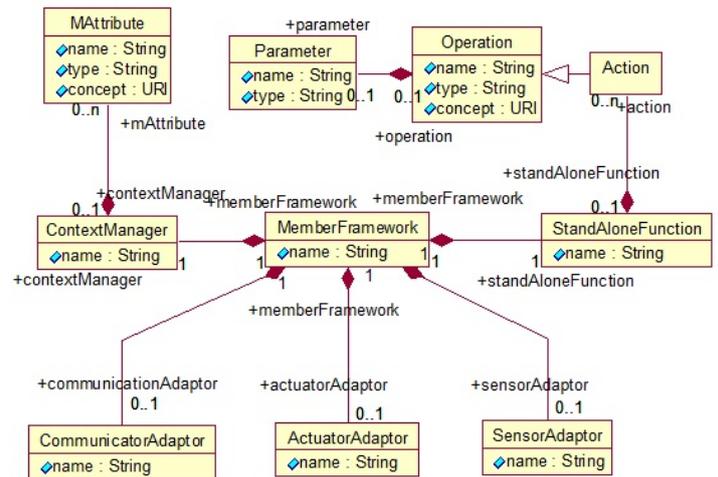
<표 1> PICM 메타모델 구성요소

4. PICM에서 PSCM으로 모델 변환

PICM(Platform Independent Community Model)에서 PSCM(Platform Specific Community Model)으로의 변환은 (그림 4)와 같이 정의된 모델을 (그림 5)로 변환시키는 과정을 말한다. 모델변환틀은 Eclipse프로젝트의 M2M중 ATL(Atlas Transformation Language)[7]을 사용했다.



(그림 4) Platform Independent Member Meta Model



(그림 5) Platform Specific Member Meta Model

PSMM은 커뮤니티 멤버 프레임워크[6]의 구조에 의해 각각의 플러그인을 나타내는 ContextManager, StandAloneFunction, CommunicatorAdaptor, ActuatorAdaptor, SensorAdaptor를 추가했으며 Member를 MemberFramework로 대체했다. 플랫폼을 멤버프레임워크로 보기 때문에 소사이어티 모델과 커뮤니티 모델은 별도의 플랫폼에 종속적인 모델을 가지지 않는다.

```

-- @path pimmm=/newModel/model/pimmm.ecore
-- @path psmm=/newModel/model/psmm.ecore

module pimmm2psmm;
create OUT : psmm from IN : pimmm;

rule M2M{
    from s : pimmm!Member

```

```

to t :
psmm!MemberFramework(name<-s.name+'MF'),
      x : psmm!StandAloneFunction(
name <- s.name + 'SAF'),
      y : psmm!ContextManager ( name
<-s.name +'CM'),
      z :
psmm!CommunicatorAdaptor(name <- s.name +'CA'),
      a : psmm!ActuatorAdaptor(name
<- s.name +'AA'),
      b :
psmm!SensorAdaptor(name<-s.name+'SA')
}

rule Mat2Mat{
  from s : pimml!MAttribute
  to t : psmm!MAttribute(name<-s.name,
type<-s.type, concept<-s.concept)
}

rule A2A{
  from s : pimml!Action
  to t : psmm!Action(name<-s.name,
type<-s.type, concept<-s.concept)
}

```

<표2 ATL변환>

<표2>의 모델변환을 통해 (그림4)는 (그림5)로 변환된다.

5. 결론

커뮤니티 컴퓨팅 시스템은 유비쿼터스 시스템이 가진 특징으로 인해 크고 복잡하다. 이런 복잡한 시스템을 구현하기 위한 방법으로 MDA를 적용하여 모델변환을 통해 어느 정도 자동화된 개발과정을 지원함으로써 개발자의 편의를 돕고 차후 시스템의 유지보수에도 도움이 될 수 있는 모델변환 틀에 대한 연구를 수행하였다.

본 논문은 커뮤니티 컴퓨팅 모델의 PICM을 변환하는 틀을 기술하였다. 현재 후속연구로 모바일 기반의 커뮤니티 컴퓨팅 서비스를 지원하는 연구가 진행되고 있으며, 커뮤니티 개발도구를 사용하여 서비스를 구현 중에 있다. 자동화 수준을 높일 수 있도록 하는 모델변환에 대한 연구를 수행할 것이다.

참고문헌

- [1] OMG(2003). MDA Guide version 1.0.1 GMG document omg/2003-06-01
- [2] Carlos Cetina, Estefania Serral, Javier Munoz, Vicente Pelechano(2007). "Tool support for Model Driven Development of Pervasive Systems", MOMPES'07

[3] Juan C. Garcia-Ojeda, Scott A. DeLoach, Robby, Walamitien H. Oyenand and Jorge Valenzuela(2008). "O-MaSE: A Customizable Approach to Developing Multiagent Development Processes", LNCS AOSE VIII vol4951/2008 1-15p, 2008

[4] Yuna Jung(2007). "Community Situation based Community Computing Model for Ubiquitous Environemnt," Ph.D. Thesis, Graduate School of Information and Communication, Ajou University 2007

[5] 김희수(2008). "커뮤니티 컴퓨팅의 역할 할당 방법," 한국 정보과학회 2008년 가을 학술발표논문집 Vol. 35, No. 2(B), pp. 27-32

[6] 김한욱, 김희수, 이정태, 김민구, "유비쿼터스 환경에서 상황기반의 커뮤니티 컴퓨팅 모델을 위한 멤버 프레임 워크", 한국정보과학회 2007 추계 학술발표 논문집 제 34 권 제 2호, pp531-535, 2007.

[7] M2M/Atlas Transformation Language(ATL) - Eclipsepedia(2010). [http://wiki.eclipse.org/M2M/Atlas_Transformation_Language_\(ATL\)](http://wiki.eclipse.org/M2M/Atlas_Transformation_Language_(ATL))

[8] Jacques Ferber, Olivier Gutknecht, Fabien Michel(2003). "From Agents to Organizations: an Organizational View of Multi-agent Systems," AAMAS'03