

시뮬레이터 기반 uT 환경에서의 커뮤니티 컴퓨팅 구현

지경환, 양정진

가톨릭대학교 컴퓨터공학과, 경기도 부천시 원미구 역곡2동 산43-1

Tel: +82-2-2164-4678, E-mail: {sshine106 | jungjin}@catholic.ac.kr

요약

삶의 질을 높여주기 위한 지능적이고 자율적인 컴퓨팅이 uT 환경과 더불어 일상생활에 스며들고 있다. uT 환경을 성공적으로 정착 시키기 위해서는 다양한 사람들과 지능적인 개체들이 조직화되어 조직내부 구성원들과 조직들이 서로 커뮤니케이션을 수행하며 지역적이고 전역적인 목표를 달성하여야 한다. 본고에서는 커뮤니티 컴퓨팅의 개념과 가상 시나리오의 모의실험을 수행하는 시뮬레이터를 이용하여 uT 환경 구성의 초석을 다지고자 한다.

Keywords:

Community Computing, Ubiquitous, Simulation

1. 서론

다양한 환경 속에서 서로 다른 정보 기기와 애플리케이션, 에이전트들이 사람들의 생활 속으로 스며들면서 Mark Weiser[1]가 주장한 유비쿼터스 컴퓨팅 패러다임이 하나씩 실현되어 가고 있다. 유비쿼터스 컴퓨팅 연구자들은 지능적인 개체가 그들을 둘러싼 환경의 변화에 자치적이고 적응적으로 대처하는 것이 사용자의 생활을 향상시킨다는 점을 하나의 전제로 공유하고 있으나[2], 유비쿼터스 컴퓨팅을 구성하는 주된 역할을 수행할 에이전트는 자신이 속한 환경의 인지능력만큼, 분산되어 변화하는 환경에 적응적으로 대응하기에는 아직 미흡하다. 다양하고 이질적인 개체들은 연속적으로 변화하는 상황에 적합하게 조직적으로 구성되어 인지된 상황에 적합한 목표를 해결하기 위하여 협동하여야 한다.

uT 환경을 구성하는 멀티 에이전트 시스템을 효율적으로 설계, 구현하기 위해서는 추상적인 수준의 정형화된 방법론이 필요하다. 따라서 에이전트 시스템 설계의 추상화와 재 사용을 위한 다양한 연구가 제시 되었다. Agent-Object-Relationship Modeling Language (AORML)[3]은 에이전트 시스템을 분석하고 설계하기 위해 제시 되었다. AORML은

에이전트의 정신 상태를 재 표현한 Internal AOR Model을 제시하였고 에이전트 시스템의 정적, 동적인 관점을 반영한 다이어그램들로 구성된 External AOR Model을 제시하여 에이전트 시스템을 위한 분석 모델을 확립하였다. 하지만 기존의 연구는 설계시점에 에이전트 조직을 구성하기 때문에 동적으로 변화하는 상황에 적응적으로 재 조직되기 부족하다.

본 고에서 제시하는 커뮤니티 컴퓨팅은 상황에 적합한 커뮤니티를 적응적으로 조직하여 다음의 주된 기능들을 제공하는 목적을 가진다. [4]

- 가상환경의 창조적인 구현을 통하여 개체들이 서로를 이해한다
- 인터넷을 통해 분산된 컴퓨팅으로 그들의 취향과 지식을 공유한다
- 다양한 커뮤니티를 위한 네트워크 솔루션으로 여론(합의)를 산출 한다
- 자율적 에이전트와 다중에이전트 시스템을 통하여 일상 생활을 지원한다
- 개인생활을 돕는 소프트웨어를 통하여 사회적 사건을 돕는다

본 고에서는 커뮤니티 컴퓨팅 목적이 반영된 시나리오를 제시하고 모의 실험을 위한 시뮬레이터를 구현한다. 시뮬레이터를 바탕으로 커뮤니티 컴퓨팅을 위한 모의실험을 통하여 에이전트 기반 소프트웨어 공학의 초석을 다지는데 도움이 되고자 한다.

2. 관련연구

2.1 가상 환경을 조성한 시스템 사례

Román et al은 [5] uT 환경에서 상황을 인지 하고 적합한 작업을 수행하기 위한 미들웨어인 GAIA를 제시 하였다.

GAIA는 여러 개체들로 구성된 물리적 공간을

프로그램 가능한 Active Space로 변환시켜준다. Active Space는 실 세계를 재 표현한 상황들을 수용 하며 에이전트가 상황인지 서비스를 제공하는 기반구조를 제공한다.

GAIA는 인지한 상황을 재 표현하기 위한 모델로 술어를 사용한다. 술어들의 형태를 분류하고 모델마다 다른 제약을 이용하여 표 1에 표현된 다양한 종류의 상황을 재 표현한다.

표 1 - GAIA에서 재 표현된 상황의 종류

Context Model	information
Physical	location, time
Environmental	weather, light, sound level
Informational	stock quotes, sports scores
Personal	health, mood, schedule, activity
Social	group activity, social relationships, whom one is in a room with
Application	email, websites visited
Systemic	network traffic, status of printers

GAIA는 상황을 감지하고 제공하는 Context Provider Agent, 생성된 상황을 소비하는 Context Consumer Agent 그리고 Context Provider Lookup Service를 제공한다. 또한, 감지된 상황들을 바탕으로 추론된 상황을 제공하는 Context Synthesizer Agent와 상황정보들을 저장하여 데이터마이닝을 수행하기 위한 Context History를 제공한다.

GAIA는 가상환경에서 지식 콘텐츠의 관리와 다른 개체간의 의미적인 상호작용을 보장하기 위하여 시멘틱 웹의 기반기술인 온톨로지를 이용한다.

2.2 개방형 환경에서 의미적 상호운용성의 보장

W3C에서는 데이터를 위한 의미(Meta-data)를 기술하기 위한 표준으로 XML을 권고한다. XML은 기계가 읽을 수 있는 구조로 정의 되었고 XML Schema와 더불어 성분(Element)과 속성(Attribute)으로 메타데이터 어휘를 기술하고 개체 수 제약(Cardinality Restriction)을 이용하여 구문을 기술한다. RDF[6]는 XML을 기반으로 유, 무형의 자원을 주어 술어 목적어 기반의 트리플 형태로 선언하며 URI와 더불어 유일한 메타데이터를 기술하기 위한 기반구조로 사용된다. RDF Schema는 RDF의 용어, 역할의 개념을 정의하는 경량의 온톨로지를 정의하기 위한 메타언어로 사용되며 응용계층은 각 개념들이 선언된 온톨로지를 명시하여 의미적 상호운용성을 보장 받는다. OWL[7]은 W3C에서 권고하는 온톨로지 언어이며 서술 논리(Description Logic)를 기술하여 논리적인

표현력과 합리적인 시스템을 위한 자료구조로 사용될 수 있다.

온톨로지의 메모리 상주 모델 처리 방식은 방대한 설계 지식을 수용하기에 부적절하기 때문에 온톨로지 저장소의 사용을 요구하고 논리적으로 표현된 지식의 일관성과 추론을 위해 추론엔진의 사용이 요구 된다. 온톨로지 저장소의 기능적 요구사항과 성능적인 요구사항을 충족시키기 위해서 KAON[8], DLDB[9], Minerva[10]등이 연구 되었다.

온톨로지 저장소는 개방된 환경에서 이질적인 시스템간의 상호운용성을 보장하기 위해서 사용될 수 있다. 온톨로지는 다양한 개체간의 협동을 위한 지식베이스 역할을 수행하며 다양한 온톨로지 저장소를 사용하기 위한 일반적인 방법론의 연구가 진행 중이다. [11]

다음 절에서는 소개한 선행연구들과 더불어 커뮤니티 컴퓨팅에서 커뮤니티의 구성과 소멸과 협동을 위한 기술들의 요구사항을 모의 실험을 통하여 도출하기 위해 사용되는 시뮬레이터를 구현한다. 유아의 안전 유지 시나리오를 통하여 일상생활에서 다양한 개체들이 조직되고 협동하여 상황에 적합하도록 사람의 활동을 지원하는 커뮤니티 컴퓨팅의 사례를 제시 한다.

3. 커뮤니티 컴퓨팅 시나리오

커뮤니티 컴퓨팅의 메타 모델과 아키텍처의 요구사항 도출의 생산성을 높이기 위해 가상의 시나리오를 설정하여 실험해 보았다. 시나리오에 등장하는 에이전트 커뮤니티는 주어진 업무를 성취하기 위하여 적합한 에이전트 혹은 커뮤니티를 구성원으로서 캐스팅 할 수 있어야 한다. 커뮤니티의 구성원들은 다른 구성원들의 변화에 따라 능동적으로 수행하는 역할을 변경할 수 있어야 한다. 에이전트 커뮤니티를 효율적으로 구성하기 위해 앞서 설명한 요구사항을 포함하는 안전 유지 시나리오를 제시한다.

3.1 유아 보호 시나리오

[시나리오 1 - Safety]

(안전영역 내) 어머니인 김지연은 현재 주방에서 바쁘게 일을 하고 있다. 딸 박수연은 아파트 단지내의 놀이터에서 또래 아이들과 즐겁게 놀고 있다. 어머니는 딸이 혼자밖에 있는 것을 걱정하여 딸의 위치 및 신변을 보호하기 위한 스마트 벨트를 착용하고 나가도록 하였다.

[Level 1 위험 지역]

시간이 얼마 지나지 않아 박수연은 아파트 외부에 사는 친구와 놀기 위해 놀이터를 벗어나서 Level 1

위험 지역으로 이동한다. 집에 있는 김지연에게 Level 1 위험 상황을 알리기 위하여 커뮤니티가 형성되고 현재 김지연이 있는 위치와 가까운 스마트 주방 가전 (밥솥)을 통하여 간단한 경보음과 텍스트로 상황을 통보한다. 경고를 본 김지연은 아파트 단지 내에 있는 것을 확인하고 걱정할 일이 아니라고 생각해서 경고상황을 종료 하고 하던 일을 계속 한다.

[Level 2 위험 지역]

박수연은 Level 1 위험지역을 벗어나 Level 2 위험 지역으로 이동한다. Level 2 위험 상황을 알리기 위해서 커뮤니티가 형성되어 집에 있는 거실의 PDP TV, 스마트 주방가전(밥솥)등 경고사항을 알릴 수 있는 모든 방법을 동원하여 김지연에게 박수연의 위험 상황을 알리기 위해 시도한다. 따라서 스마트 주방가전에는 경보음과 함께 텍스트로 상황을 통보하며 PDP는 자동으로 전원이 켜지고 볼륨을 높여 박수연의 현재 상황을 스마트 가로등의 카메라를 통해 영상으로 보여준다. 상황을 인지하여 PDP TV 앞으로 이동한 김지연은 박수연이 아파트 단지를 벗어나는 것을 확인하고 위험상황으로 판단하여 긴급 상황임을 전달한다. 커뮤니티 매니저는 위험상황을 해결하기 위하여 김지연의 오프라인 커뮤니티 멤버(친목 모임) 중에 박수연의 주위에 있는 사람을 찾아 보니 최홍임, 신성희씨가 있다는 것을 발견한다. 박수연과의 거리를 기준으로 가까운 최홍임씨의 OMD¹에 도움 요청메시지를 전달하자 최홍임씨는 다른 바쁜 일로 인해 불가능하다는 의사를 보낸다. 다시 신성희씨의 OMD에 메시지를 전달하자 신성희씨는 기꺼이 허락한다. 허락과 동시에 현재 박수연의 위치와 영상을 신성희씨의 OMD에 전송하고 해당 정보를 이용하여 박수연을 찾아 집으로 데리고 온다.

3.2 위험지역별 순서도

시나리오에 등장하는 에이전트와 스마트 장치들 중 커뮤니티 컴퓨팅의 기능을 구현하고 시나리오의 목적을 달성하기 위한 에이전트를 선별하였다.

그림 1은 선별된 에이전트를 토대로 아이가 Level 1 위험 지역으로 이동할 때에 커뮤니티 구성과 소멸을 순서도로 표현한 것이다.

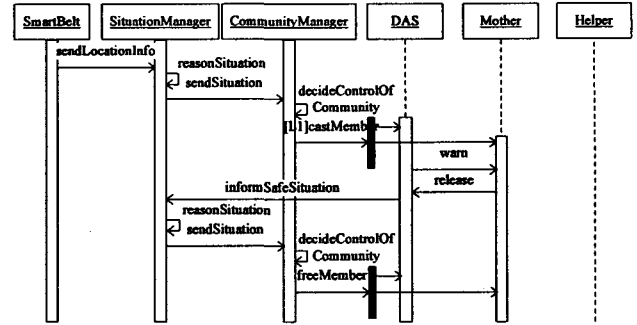


그림 1 - Level-1 위험 지역의 순서도

SmartBelt 에이전트는 아이의 위치 정보를 SituationManager 에이전트에게 주기적으로 전송한다. SituationManager 에이전트는 제공되는 위치정보, 이전 상황 그리고 활동 중인 커뮤니티를 토대로 현재 상황을 추론한다. CommunityManager 에이전트는 SituationManager 에이전트로부터 상황정보를 전달 받고 Level 1 위험 지역에서 아이의 안전을 유도하기 위한 커뮤니티를 형성한다. DAS²는 Mother 에이전트의 위치와 가까운 곳의 주방 가전을 통하여 위험 상황을 경보하고 Mother 에이전트는 Level 1 지역에서는 아파트 단지 내에 있는 것을 확인 하고 안전하다는 상황을 SituationManager 에이전트에게 전달한다. SituationManager 에이전트는 전달된 정보와 위치정보, 이동 상황, 활동 중인 커뮤니티를 토대로 추론한 현재 상황을 CommunityManager 에이전트에게 전달한다. CommunityManager 에이전트는 전달 받은 정보를 토대로 활동중인 커뮤니티를 소멸한다.

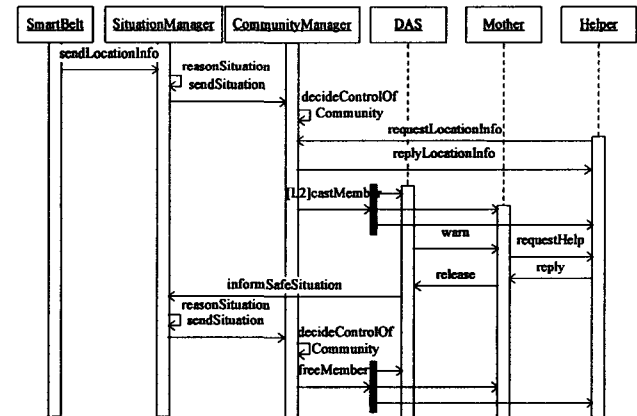


그림 2 - Level-2 위험 지역의 순서도

그림 2는 아이가 Level 2 위험 지역으로 이동하였을 때의 커뮤니티 구성을 순서도로 표현한 것이다. SituationManager 에이전트와

¹ Organic Mobile Device (OMD)

본 논문에서는 개인이 휴대하는 모바일 디바이스들로 구성된 집합체로 가정한다.

² Digital Appliance Server (DAS)

본 논문에서는 가전제품(e.g., 주방 가전, PDP TV)들을 제어하는 통합적인 서버로 가정한다. 멀티 에이전트 플랫폼을 배치하고 Wrapper Agent를 생성하여 수동적인 가전제품을 제어한다.

보이기도 하였다.

4.3 시뮬레이터 디자인

그림 4는 앞서 설명한 시뮬레이터와 에이전트들의 구동화면이며 화면 좌측 상단에는 아이의 위치를 조정하는 제어 패널을 좌측 하단에는 모든 에이전트들의 동작을 모니터링 하는 뷰어를 배치 하였다. 시뮬레이터의 우측에는 메시지들의 전역 공간(i.e. Environment State)을 관찰할 수 있는 뷰어를 배치 하였고 화면 중앙에는 스크린을 배치 하였다. 각 개체들은 이름과 표현을 한 쌍으로 배치 하였고 동일한 커뮤니티 멤버들을 표현하기 위해 멤버들의 이름은 같은 글자 색으로 표현하였다.

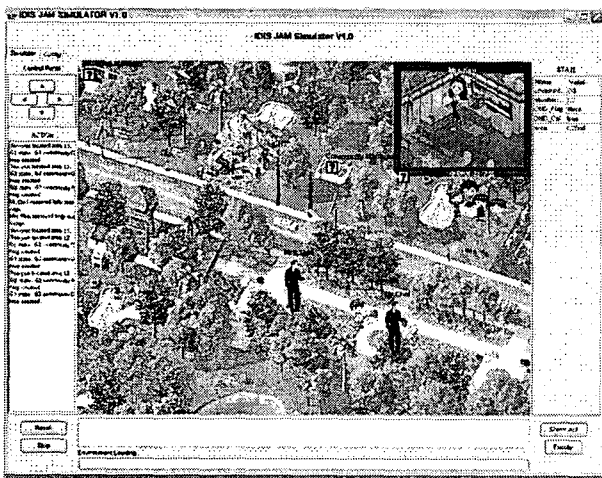


그림 4- 시뮬레이터 패널

그림 4에서 오른쪽 상단 부는 집이고 왼쪽 상단 부는 Level 1 위험 지역이며 차도 하단부는 Level 2 위험 지역이다. 아이의 이름은 SOO_YUN으로 표시되었으며 Level 1, Level 2, Level 1, Level 2, 안전지대, Level 2 지역들로 이동 했었다는 것을 동작 뷰어를 통하여 관찰할 수 있다. 그림 4에서 빨간색으로 표시된 개체들과 그림 5 좌측의 OMD는 아이가 Level 2 지역일 때 캐스팅된 커뮤니티 멤버이다.

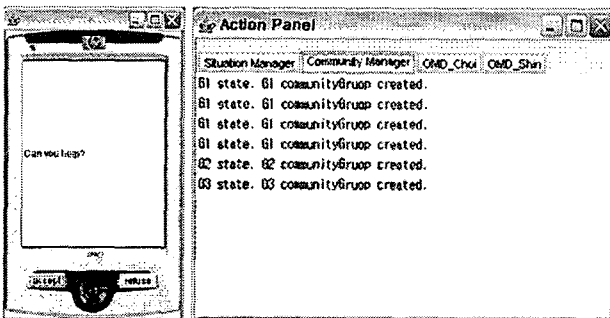


그림 5 - OMD, Action 패널

그림 5의 우측 패널은 각 에이전트들의 동작들을

모니터링 하기 위한 뷰어들이며 좌측 패널은 OMD이다. OMD는 Helper 에이전트로서 커뮤니티에 캐스팅 되고 accept와 refuse버튼으로 사용자가 도움 수락 여부를 제어할 수 있도록 구현 하였다.

5. 결론

소프트웨어 공학 패러다임은 객 체기반에서 에이전트 기반으로 이동하고 있고 에이전트를 이용하여 문제를 해결하는 다양한 방법론이 활발히 연구되고 있다. 에이전트 시스템과 에이전트 시스템간의 상호작용을 위한 표준 명세가 정립되고 이를 수용하고 다양한 컴퓨팅 환경에서 동작할 수 있는 MAS(Multi Agent System) 플랫폼 연구 역시 활발히 진행되어 왔다.

커뮤니티 컴퓨팅의 시나리오 설정과 시뮬레이터 기반의 모의 실험은 커뮤니티 컴퓨팅 기반 소프트웨어 방법론 연구에 도움을 준다. 에이전트를 통한 원격 시뮬레이터 객체의 제어는 가상 환경에서 에이전트 기반의 커뮤니티 컴퓨팅을 가능하게 하며 커뮤니티 컴퓨팅의 메타 모델과 아키텍처의 요구사항 도출의 생산성을 높여준다. 본 연구를 기반으로 앞으로의 목표는 시뮬레이터와 다양한 모의 실험을 바탕으로 커뮤니티 컴퓨팅에 적합한 에이전트 기반 소프트웨어 공학의 초석을 다지는데 있다.

후기

This research is supported by the ubiquitous autonomic computing and network project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea

참고문헌

- [1] Weiser M. (1991). The computer for the 21st Century. Scientific American, 265(3), 66-75
- [2] AK Dey, D. Salber, and GD Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction, 16. 2001
- [3] G. Wagner: The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior. Information Systems 28:5 (2003), pp. 475-504.
- [4] Toru Ishida Ed. Community Computing: Collaboration over Global Information Networks. John Wiley and Sons, 1998.

- [5] Gaia: A Middleware Infrastructure to Enable Active Spaces. Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt, In IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.
- [6] W3C, RDF Primer W3C Recommendation 10 February, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210>, 2004.
- [7] W3C, OWL Web Ontology Language Overview W3C Recommendation 10 February, <http://www.w3.org/TR/2004/REC-owl-features-20040210>, 2004.
- [8] Boris Motik, Daniel Oberle, Steffen Staab, Rudi Studer, Raphael Volz, KAON Server Architecture, Technical Report 421, University of Karlsruhe, Institute AIFB, 76128 Karlsruhe, Germany. 2002.
- [9] SWAT Projects, HAWK - OWL Repository and Toolkit, <http://swat.cse.lehigh.edu/projects/index.html#hawk>.
- [10] IBM, IBM Integrated Ontology Development Toolkit March 15, <http://www.alphaworks.ibm.com/tech/semanticstk>, 2006.
- [11] SungTae Kim, KyengWhan Jee, JungJin Yang, Utilizing Distributed Ontology Repository in Multi-Agent System Environment, Journal of Intelligent Information Systems, Vol.11 (3) pp. 129-139, 2005.
- [12] Marcus J. Huber, Ph.D., Usage Manual for the Jam! Agent Architecture, November 2001, <http://www.marcush.net/IRS/Jam/Jam-man-01Nov01.doc>