

커뮤니티 컴퓨팅을 위한 문맥적 요구사항

지경환^o 양정진

가톨릭 대학교

{sshine106^o, jungjin}@catholic.ac.kr

Contextual Requirements for Community Computing

KyengWhan Jee^o JungJin Yang

The Catholic University of Korea

요 약

본 논문에서는 능동적인 상호협력으로 목적에 맞는 서비스를 제공하는 컴퓨팅 환경에서 가장 중요한 역할을 하게 될 문맥(Context)을 중점적으로 설명한다. 문맥은 정보를 주고받는 기본 단위가 될 것이며 이를 위한 표준이 갖추어져 할 요구사항과 이것을 수용할 수 있는 기술적 토대(XML, XML Schema, RDF, Ontology Language)를 소개 한다.

1. 서론

급속한 정보통신의 발달은 인간의 삶에 직접적인 영향을 끼치는 단계에 접어들게 되었다. 기존의 컴퓨팅은 사용자의 입력에 의지하여 동작 하지만 유비쿼터스 시대의 커뮤니티 컴퓨팅은 사용자의 의향과 상황을 인지해 동적으로 장치를 설정하고 자동으로 협력해 목표를 해결한다. 앞으로 다가올 정보통신의 복합적인 채널인 커뮤니티 컴퓨팅 환경을 구축하기 위한 방안으로 커뮤니티 컴퓨팅의 모델, 기술 그리고 환경에 대해 연구할 필요가 있다.

본 논문은 커뮤니티 컴퓨팅 환경을 이루는 개체들이 능동적으로 상호간 협력하여 목적에 맞는 다양한 서비스를 제공하는 환경을 위한 문맥적인 요구사항을 설명한다. 궁극적인 목표는 단순하게 서비스를 나열하는 것에서 부터 일시적으로 협력하고, 주어진 상황을 효과적으로 인지할 뿐만 아니라 다수의 장치가 협력 관계를 추구하는 성공적인 커뮤니티 컴퓨팅의 모델을 확립하는 것이다.

2. 관련 연구

2.1 Ontology

온톨로지(Ontology)^[1]는 철학의 한 갈래로 존재의 본질을 연구하는 형이상학이다. 인공지능 영역에서의 온톨로지는 개체와 각 개체간의 관계성을 기술하는 언어로써 인지능력을 위한 기본 구조가 될 수 있다. 온톨로지는 주어, 술어, 목적어 구문(Triple)의 형태로 자원을 기술하는 기반구조인 RDF(Resource Description Framework)^[2]를 바탕으로 추론능력과 강화된 표현력을 제공해 줄 수 있는 DAML+OIL(DARPA Agent Markup Language +

Ontology Inference Layer)^[3]으로 Description Logic을 기술하게 된다. 이 모든 언어는 XML(eXtensible Markup Language)을 기반으로 구법(Schema)을 가지게 되며 기계가 읽을 수 있는(Machine Readable) 특성을 가지게 된다.

2.2 GAIA

GAIA^[4]는 물리적인 공간을 관리하여, 상위 레벨의 프로그래밍을 할 수 있도록 Active Space를 제공하는 Pervasive Computing Environment를 위한 미들웨어(Middleware)로써 커뮤니티 컴퓨팅을 위한 레이어, 혹은 플랫폼이라고 할 수 있다. 또한 CORBA를 사용하여 분산 컴퓨팅을 가능하게 하였고 "Context Awareness"와 "Semantic Interoperability"를 위하여 온톨로지 사용을 영두에 두었다.

3. 커뮤니티 컴퓨팅을 위한 문맥적 요구사항

커뮤니티 컴퓨팅은 사용자에게 현재 Context에 알맞은 서비스를 제공하고 시간이 지남에 따라 특정 Context가 인도되면 그 상황에 맞는 적응적인(Adaptive) 서비스를 제공하기 위하여 각 개체 간 협동, 협상을 할 수 있어야 한다. 이를 위하여 Context는 조합, 분해, 전달, 처리, 저장, 회수를 할 수 있게 설계되어야 한다.

(1) Context Model

Context는 커뮤니티 컴퓨팅 환경에서 가장 중요한 정보가 될 것이며 다양한 개체들은 많은 종류의 Context를 생성, 소비 할 것이다. 이를 위하여 Context의 종류를 구분하는 것은 중요하며 [표 1] 예로 볼 수 있다.

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

[표 1]

정보는 특성에 따라 분류되고 다른 술어(Predicate)의 형태로 Context Model을 정의하게 된다. 커뮤니티 컴퓨팅에 참여하는 개체들 간에 Context와 메타데이터를 함께 교환하는 것은 Context의 저장, 관리, 처리, 수집, 검색, 교환을 용이 하게 한다.

(2) Community Model

성공적인 커뮤니티 컴퓨팅을 위해선 반드시 Context-aware Adaptive Behavior가 있어야 한다. Behavior는 신중한 결정을 위하여 다음과 같은 정보를 얻어낼 수 있어야 한다. "어떤 개체가 존재하는가?", "그 개체의 특징(목적, 위치 등)은 무엇인가?", "어떤 정보를 요청, 기대 하는가?", "주고받는 Context의 의미는 무엇인가?". 특정 개체는 목적에 따라 특정 Context들을 함께 요청할 수 있으며 이를 위한 Community Model을 염두에 두어야 한다.

Community Model은 특정 Context들이 집단적으로 생성, 저장, 교환될 수 있게 하고 개체간의 추상화된 질의, 응답을 수용할 수 있게 한다. 이를 위해선 다양한 종류의 Community Model들을 표현하고 생성, 조합, 분해를 염두에 둘 수 있는 기본방향이 될 스키마가 필요하다.

(3) Context-aware Adaptive Behavior

Behavior는 환경의 동적인 변화에 따른 적응의 단계가 필요하고 이를 위해서 Context Repository로 부터 최신의 Context Model, Community Model, Context를 유지할 수 있어야 한다. 또한 Context의 내용에 따라 Behavior가 알맞은 동작을 할 수 있게 기술하는 사람 혹은 기계를 위한 메커니즘 역시 필요하다. 이 기술(Description)이 Community Model에 포함된다면 고정적인 구현 정보를 가지는 개체와 비교해 더 적응적으로 동작할 수 있게 된다.

(4) Resource Awareness

개체(예. Embedded, Host Device)의 자원적인 제약으로 인한 서비스의 적응과 재배치를 위해서 제공 할 수

있는 최대 가용자원에 관한 충분한 정보(Processing power, Memory, Bandwidth, Battery life time)가 필요하다. 제공된 정보는 Community Model에 포함될 수 있으며 다른 개체와 교환될 수 있다.

(5) Code Generation

다양한 장치로 서비스를 배치하기 위해서 추상화된 high-level 명세(Spec.)를 각 장치에 알맞은 구현(Implementation)으로 생성할 수 있어야 한다. 이를 위해 Operating system, Drivers, Softwares, Libraries, Virtual Machines, APIs등의 정보를 필요로 한다.[5]

(6) Mobile Services

시간에 따른 위치의 변화로 서비스의 전체 혹은 부분이 짧은 시간 내로 이주할 필요가 있다. 이때 플랫폼 실행에 관한 자세한 정보가 자동적으로 이주하여야 하며 다양한 플랫폼에 호환 가능한 VM(Virtual Machine)이 필요하다.

(7) Semantic Service

커뮤니티 컴퓨팅에 참여하는 개체들은 Community Model, Context Model, 개체의 정보(예. 제공하는 서비스, 자원 등)의 제공자(Provider)이며, 소비자(Consumer)이다. 좀 더 전역적인 정보의 제공을 위하여 개체로부터 제공된 정보를 광고(Advertisement), 이벤트(예. 다른 개체의 등장)의 공고(Notification), 개체의 서비스를 중재하는 레지스트리 혹은 저장소(Repository)가 필요하다. 개체가 소비자의 입장에서 레지스트리 혹은 저장소에 Community Model 수준의 추상화된 질의를 할 수 있는 메커니즘이 필요하다.

다양한 개체 간 상호운용성을 위해 메시지의 해석에 앞서 스키마를 알아야 하며 의미적 상호운용성(Semantic Interoperability)을 위해 Ontology Repository가 필요하다. 간단한 차폐형 시스템(closed system)에서 모든 요구된 스키마는 컴포넌트 내부에 번역 될 수 있지만 개방형 시스템(open system)에서는 각 집단이 자동적이고, 다양하고, 진보적으로 다룰 수 있어야 한다. 이를 위해서 스키마를 교환(게시, 검색), 합성, 정의하기 위한 개방형 모델이 필요하며 표준화된 어법과 어휘, 기계가 읽을 수 있는 특성을 가져야 한다.[6] 또한 온톨로지에 포함될 수 있는 구조를 가져야 하며 이를 위해 XML Schema를 고려 할 수 있다.

(8) Context-aware User Interfaces

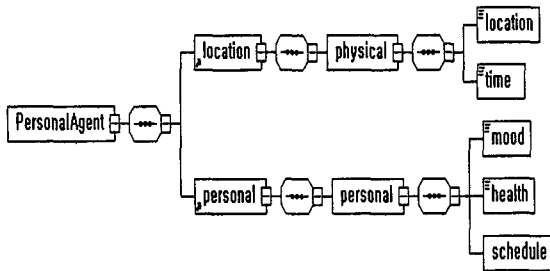
UI(User Interface)는 최종사용자를 위한 장치가 그 시점에서 제공 할 수 있는 최대 가용자원 내에서 제공될 수 있어야 하며 시간에 따른 Context의 변화에 적응하며

재설정이 될 수 있어야 한다. 이를 위한 대안으로써 XML문서로 데이터를 표현하고 XSLT를 이용하여 표현을 위한 태그트리를 재구성 하는 방식을 생각 할 수가 있다.

4. 기반 기술

4.1 다양한 Model을 정의 할 수 있는 XML Schema

XML은 자기 설명적 언어로써 기계가 읽을 수 있는 특성을 가지고 있다. 또한 표현력이 풍부하여 문자열을 비롯해서 이진코드까지 다양한 종류의 데이터를 기술할 수가 있다. XML Schema는 메타언어로써 XML기반의 어휘와 어법을 기술하여 새로운 언어를 작성할 수 있다.



[그림 1]

[그림 1]은 지능형 에이전트를 개체로 가정하고 사용되는 Community Model(PersonalAgent)을 Physical Context Model(location, personal)을 포함(Include)하여 구성하였다. Community Model은 작성되어진 다른 모델의 스키마를 바탕으로 확장 될 수 있고 특정 개체의 목적을 위한 또 다른 Community Model로 재구성 되어질 수 있다.

4.2 RDF를 사용한 Model 의미 표현

RDF는 URI(Uniform Resource Identifier)로 유, 무형의 자원을 지칭할 수 있으며 Context Model, Community Model, 커뮤니티 컴퓨팅에 참여하는 다양한 개체를 기술하는데 사용된다.

```

1: <?xml version="1.0" encoding="EUC-KR"?>
2: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3: xmlns:cc="http://CommunityComputing/terms/0.1/"
4: xml:base="http://idis.catholic.ac.kr/CommunityOntology.owl">
5:   <cc:CommunityModel rdf:ID="PersonalAgent">
6:     <cc:useContextModel rdf:parseType="Literal">
7:       <physical/>
8:       <personal/>
9:     </cc:useContextModel>
10:   </cc:CommunityModel>
11: </rdf:RDF>
    
```

[그림 2]

[그림 2]의 RDF구문은 CommunityModel의 인스턴스인 PersonalAgent를 기술하고 해당 온톨로지를 명시하고

있다. 또한 useContextModel 이라는 술어와 함께 그 밖의 정보역시 기술할 수 있음을 보여준다. RDF는 트리플 형태로 RDB(Relation Data Base)에 저장될 수 있기 때문에 빠른 회수가 가능 하다.

4.3 Ontology를 이용한 Semantic Interoperability

각 Community Model은 스스로가 기술된 온톨로지의 정보를 포함한다. 사용되는 Context는 용어의 URI를 비교함으로써 같은 의미로 사용되는 것인지 알 수 있다. 다른 온톨로지에 기술된 용어가 사용 될 경우 온톨로지 간의 의미를 유추하기 위한 General Ontology 혹은 Reference Ontology를 활용 할 수 있다. 온톨로지는 Description Logic을 바탕으로 단순 키워드 검색을 벗어난 의미적인 추론을 통하여 다른 언어로 기술된 목적 어휘를 찾는 것이 가능하다.

5. 향후 계획

본 논문에서 설명한 문맥적 요구사항을 수용 할 수 있는 Context, Community Model을 자동적으로 구성하는 메커니즘을 연구할 것이며 더 나아가서는 커뮤니티 컴퓨팅의 플랫폼(Platform)의 요구사항과 이것을 수용할 수 있는 미들웨어를 구성하고 이것을 기반으로 한 커뮤니티 컴퓨팅의 모델을 확립할 것이다.

6. 참고문헌

- [1] OWL Web Ontology Language Semantics and Abstract Syntax <http://www.w3.org/TR/2004/REC-owl-semantic-20040210/>
- [2] RDF Primer, W3C Working Draft 23 January 2003, <http://www.w3.org/TR/rdf-primer>
- [3] Reference description of the DAML+OIL (March 2001) ontology markup language <http://www.daml.org/2001/03/reference>
- [4] Manuel Román, Christopher Hess, Renato Cerqueira, Anand Ranganat, Roy H. Campbell, Klara Nahrstedt "Gaia: A Middleware Infrastructure to Enable Active Spaces (2nd Revision)", 7/1/2002
- [5] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere "Towards an extensible context ontology for Ambient Intelligence" EUSAI 2004, LNCS 3295, pp. 148-159, 2004
- [6] Robert E. McGrath, Anand Ranganathan, Roy H. Campbell and M. Dennis Mickunas "Use of Ontologies in Pervasive Computing Environments", UIUCDCS-R-2003-2332, April 2003