유비쿼터스 컴퓨팅 환경에서의 접근제어 모델을 위한 요구사항 분석

오 세 종[†] · 박 제 호[††]

요 약

유비쿼터스 컴퓨팅 환경은 보안의 강화와 사생활 보호라는 과제를 안고 있다. 접근 제어는 보안 분야의 하나인데, 유비쿼터스 컴퓨팅 환경은 전통적인 정보시스템과는 여러 면에서 특성을 달리하기 때문에 기존의 접근제어 모델을 그대로 적용하기에는 무리가 있다. 본 연구에서는 접근제어 측면에서 유비쿼터스 컴퓨팅 환경을 정의하고 그 환경에서 접근제어의 특성을 분석한 뒤, 그 환경을 위한 접근제어 모델을 개발할 때 필수적으로 고려해야할 요구사항을 제시하였다. 또한 접근제어 모델의 구현시 가능한 세가지 유형에 대해서도 제시하였다.

Requirements Analysis for Access Control Model on Ubiquitous Computing Environment

Sejong Oh[†] · Jeho Park[††]

ABSTRACT

Ubiquitous computing environment requires strong security and privacy. Access control is one of security areas. Access control on Ubiquitous computing is different from it on traditional information systems so that traditional access control models are not suitable for Ubiquitous computing environment. This research defines Ubiquitous computing environment as an access control point of view, and shows requirements to consider for developing access control model for Ubiquitous computing environment. It also brings up three implementing types of access control on Ubiquitous computing environment.

키워드 : 정보보호(Security), 접근제어(Access Control), 유비쿼터스 컴퓨팅(Ubiquitous Computing)

1. 서 론

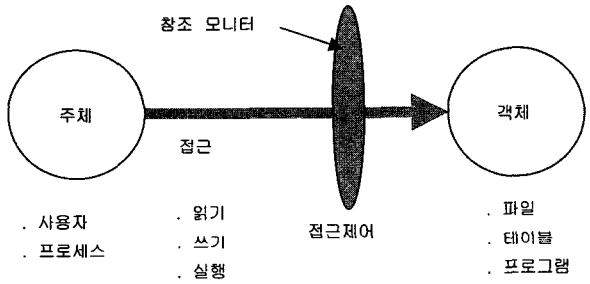
유비쿼터스 컴퓨팅은 Mark Weiser에 의해 1991년 소개된 이후 미국, 유럽, 일본등 세계 각국에서 모바일, 브로드 밴드, 극소형 컴퓨터, IPv6 등이 창출해 내는 유비쿼터스 컴퓨팅 혁명을 차세대 정보통신의 새로운 패러다임이라고 인식하고 정부, 기업, 대학을 중심으로 많은 예산과 인력을 투입하여 연구 및 개발을 진행하고 있다. 이러한 유비쿼터스 환경은 기밀성(security)의 보장과 사생활 보호(privacy)라는 측면에서 적절한 보안기술의 개발을 필요로 한다. 먼저 기밀성의 보장이라는 측면에서 보면 보안 기술 자체가 유비쿼터스 컴퓨팅 기술의 한 부분으로서 인식되고 있다. 유비쿼터스 컴퓨팅에서는 다수의 사용자가 다수의 정보객체, 센서, 시스템과 커뮤니케이션을 하면서 금융거래, 신용 거래등을 할 수 있기 때문에 정보의 안전한 유통은 유비쿼터스 컴퓨팅의 필수적인 요소가 된다. 또한 사용자의 입장에서 볼 때 유비쿼터스 환경이 편리성을 가져다주지만 사생활 보호가 충분히 되지 않는다면 유비쿼터스 컴퓨팅의 이용에 소극적이 될 수밖에 없기 때문에 보안 기술의 개발이 필요하다.

유비쿼터스 컴퓨팅에 관련된 보안 기술에는 사용자 인증, 데이터 보호, 보안 프로토콜 등의 분야가 있으나 본 연구는 접근제어(access control) 분야에 초점을 맞춘다. 접근제어란 주체(Subject ; 사용자, process, intelligent agent)가 정보 객체(Object ; file, database, program, machine)에 접근(Access ; read, write, execute)하려고 할 때 주체가 가지고 있는 권한에 기초하여 접근을 허용할 것인지 차단할 것인지를 결정하는 것을 말한다(그림 1). 예를 들면 지능형 홈(intelligent home)에서 아버지와 어머니는 케이블 TV의 성인 채널에 접근할 수 있어야 하지만 청소년인 자녀들은 접근이 차단되어야 한다. 일반적인 정보시스템 환경에 비해 유비쿼터스 컴퓨팅 환경은 매우 다이나믹하고 접근제어 판단의 기본 자료가 되는 컨텍스트(context) 정보들이 시시각각 변하기 때문에 일반적

† 정 회 원 : 단국대학교 컴퓨터과학과 교수
†† 종신회원 : 단국대학교 컴퓨터과학과 교수
논문접수 : 2004년 7월 16일, 심사완료 : 2004년 11월 11일

인 접근제어 모델을 적용하는데 무리가 있다. 따라서 유비쿼터스 환경에 적합한 새로운 접근제어 모델의 개발이 필요하다. 본 연구에서는 유비쿼터스 컴퓨팅 환경을 위한 접근제어 모델을 개발하기에 앞서 유비쿼터스 컴퓨팅 환경과 전통적인 정보시스템 환경에서 접근제어가 특성상 어떤 차이를 보이는지 살펴보고 유비쿼터스 컴퓨팅 환경에서 접근제어에 대한 요구사항을 분석한다. 이러한 연구는 지능형 홈, 지능형 사무실 등 접근제어가 중요시되는 유비쿼터스 컴퓨팅 환경에 필요한 접근제어 모델을 개발하는데 필수적으로 필요하다[16].

본 논문의 구성은 다음과 같다. 2장에서는 접근제어에 관련된 연구를 살펴본다. 특히 전통적인 접근제어 모델들의 특징에 대해 살펴보고 유비쿼터스 컴퓨팅 환경에서 접근제어에 대한 연구동향을 소개한다. 3장에서는 접근제어의 관점에서 유비쿼터스 컴퓨팅 환경을 정의하고 이 환경에서의 접근제어 모델은 어떤 특징이 있는지를 전통적인 환경에서의 접근제어 모델과의 비교를 통해 분석한다. 4장에서는 유비쿼터스 환경에서의 접근제어 모델이 갖는 특징을 기초로 접근제어 모델에 대한 요구사항을 도출하고, 그중 한가지 요구사항에 대해 해결방법을 사례를 통해 제시한다. 그리고 5장에서 결론을 맺는다.



(그림 1) 접근제어의 개념[1)]

2. 관련 연구

다수의 사용자가 하나의 정보시스템을 공유하는 환경이 보편화 되면서 운영체제, 데이터베이스 및 응용 시스템에서 사용자의 권한을 관리하고 권한에 따라 정보 객체에 대한 접근을 제한하는 접근제어 분야가 연구되고 구현되어 왔다. 사용자 및 정보객체의 수가 많으면 많을수록 효율적인 접근제어 모델의 필요성이 증가한다. 지금까지 여러 가지 접근 제어 모델들이 개발 되었는데 대표적인 것으로는 접근제어 리스트(ACL : Access Control List) 모델, 자율적 접근제어(DAC : Discretionary Access Control) 모델, 강제적 접근제어 (MAC : Mandatory Access Control) 모델 등이 있다[11, 12]. 접근제어 리스트는 Unix나 Window 같은 운영체제에 사용되며, 사용자에게 부여된 권한 리스트에 따라 접근제어를 수행한다. 접근제어 리스트는 (그림 2)와 같이 접근제어 행렬(ACM : Access Control Matrix)로도 표현된다. 자율적 접근제어 모델에서는 정보 객체마다 소유자(owner)가 존재하며 소유자들이 자율적으로 사용 권한을 다른 사용자에게 부여하거나 회수할 수 있는 모델이다. 강제적 접근제어는 군사환경과 같이 엄격한 정보보호가 필요한 환경에서 사용되며 (그림 3)과 같이 사용자와 정보 객체에 보안 등급을 부여하고 사용자는 자신의 보안 등급에 적합한 정보 객체에만 접근할 수 있도록 제한한다. 역할기반 접근제어(RBAC : Role-Based Access Control) 모델은 사용자들에게 직접 권한을 할당(assign)하던 기존의 모델들과는 달리 (그림 4)와 같이 현실세계에서 수행하는 업무적 역할에 따라 인가권한(permission)을 역할(role)에 할당하고, 사용자들은 적당한 역할에 소속되도록 함으로써 사용자들의 권한 관리를 효율적으로 할 수 있도록 지원한다. 현실세계에서는 사용자들의 권한 관리를 담당하는 보안 관리자(security administrator)가 존재하며, 큰 조직의 경우 한명의 보안 관리자가 아닌 여러 보안 관리자들이 자신에게 주어진 권한 범위 내에서 보안 업무를 수행하는 분산 보안 관리가 일반적이다. 보안 관리자들에 대한 관리에 대해 RBAC의 방법을 적용하고 이를 RBAC 모델과 통합한 모델이 관리적 역할기반 접근제어(ARBAC : Administrative Role-Based Access Control) 모델이다. 현재 ARBAC97 모델[14]과 ARBAC02 모델[15]이 제안되어 있다.

접근제어 모델은 접근제어가 수행되는 환경에 의존한다. 접근제어는 현실의 요구사항이 반영된 것으로 환경이 바뀌면 접근제어의 방법도 바뀌어야 한다. 유비쿼터스 컴퓨팅 환경이 대두되면서 유비쿼터스 컴퓨팅에 필요한 보안 분야가 활발히 연구되고 있는데 접근제어도 그중의 한 분야이다. Stajano와 Anderson는 3대 보안 원리인 기밀성(confidentiality), 무결성(integrity), 유용성(availability)의 측면에서 유비쿼터스 컴퓨팅 환경의 특징에 대해 살펴보고 Resurrecting Duckling이라는 새로운 보안 원리를 제안하였다[1]. Varshney는 무선 기반 구조(wireless infrastructure)를 갖는 유비쿼터스 컴퓨팅 환경은 여러 부분에서 보안 취약성이 존재할 수 있으며 인증(authentication), 권한부여(authorization), 접근성(accessibility)의 어려움을 설명하였다[2]. 또한 서로 다른 보안 레벨을 갖는 다양한 무선 장치들이 공존하는 상황을 고려할 것을 제안하였다. Tuchinda는 유비쿼터스 컴퓨팅 환경에서는 사용자의 위치 정보나 신상 정보가 쉽게 드러날 수 있기 때문에 사생활보호(privacy)의 문제가 심각함을 지적하였고 보안에 대한 하나의 응용으로서 기존의 접근제어 모델인 RBAC 모델을 유비쿼터스 컴퓨팅 환경에 적용하려는 연구를 하고 있다[3]. 이외에도 여러 연구들이 유비쿼터스 컴퓨팅 환경에서의 보안에 대해 다루고 있다[4-7, 13].

유비쿼터스 컴퓨팅 환경에서는 센서 네트워크로부터 수집된

1) (그림 1)에서 참조 모니터(reference monitor)란 실제 접근제어를 수행하는 프로세스 혹은 시스템을 의미한다. 참조 모니터는 주체가 객체에 접근을 할 때 이를 탐지하여 자신이 가지고 있는 권한 정보와 비교한 뒤 접근에 대한 허용 여부를 결정한다.

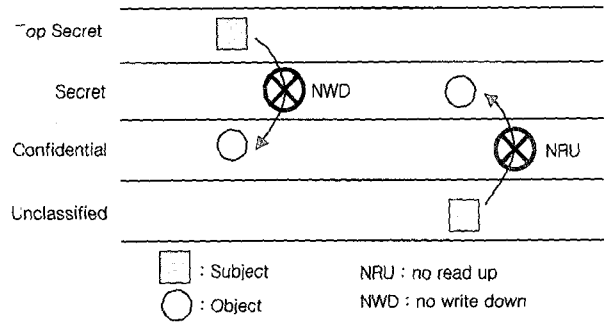
다양한 컨텍스트 정보가 접근제어시 참조되어야 한다. 이러한 컨텍스트 정보를 다루기 위해 여러 방법이 제안되었다[8-10]. Bacon는 역할 및 권한 정보를 표현할 수 있는 의사 자연어(pseudo natural language)를 제시하였는데, 이 의사 자연어는 논리(Logic)로 변환되어 처리 되도록 하였다. Covington은 역할의 개념을 확장하여 컨텍스트 정보를 환경 역할(environment)로 표현하고 역할을 다루는 방법과 유사한 방법으로 컨텍스트 정보를 다룰 수 있도록 하였다[9]. Kumar는 역할 컨텍스트와 컨텍스트 필터라는 개념을 도입하여 사용자가 처한 컨텍스트에 따라 접근제어가 달리 될 수 있는 모델을 제안하였다[10].

지금까지의 연구를 살펴보면 유비쿼터스 컴퓨팅 환경에 대한 접근제어의 적용이 개별 응용 환경에 기초한 경우가 많고 유비쿼터스 컴퓨팅 환경에서 접근제어의 특징에 대한 전반적인 고려가 부족하였다. 유비쿼터스 컴퓨팅 환경에 적합한 접근제어 모델을 개발하기 위해서는 유비쿼터스 컴퓨팅 환경에서의 접근제어 모델의 특징을 이해하고 요구사항을 분석하는 작업이 필요하다.

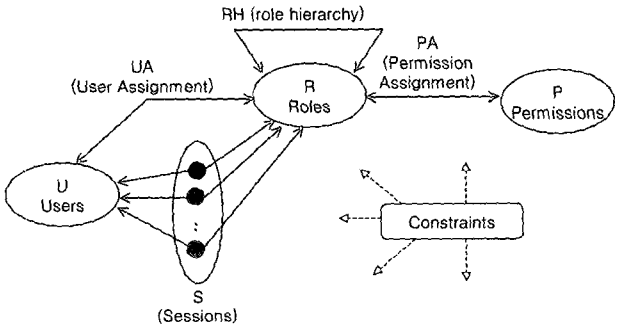
	File1	File2	File3
John	R	RW	R
Tom			RW

(R : read, W : write)

(그림 2) 접근제어 행렬의 예



(그림 3) 강제적 접근제어 모델

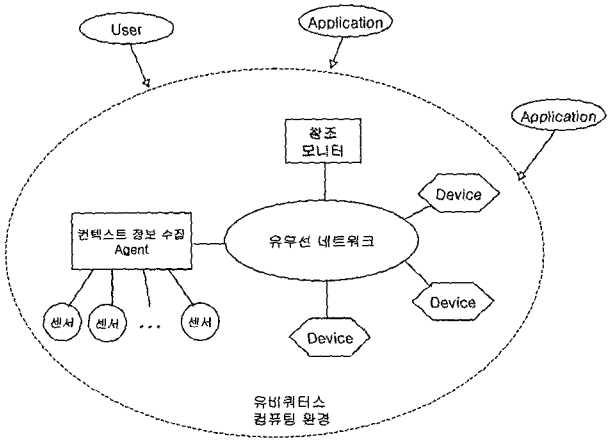


(그림 4) 역할기반 접근제어 모델

3. 유비쿼터스 컴퓨팅 환경에서 접근제어의 특징

3.1 접근제어 관점에서의 유비쿼터스 컴퓨팅 환경

유비쿼터스 컴퓨팅 환경은 연구자나 응용 프로젝트의 종류에 따라 정의와 내용을 달리하고 있다. 현재 추진되고 있는 대표적인 유비쿼터스 컴퓨팅 프로젝트를 보면 MS사의 이지리빙(Easy Living), UC Berkeley의 스마트 먼지(Smart Dust), NIST의 스마트 공간(Smart Space), 카네기 멜런의 아우라(Aura)등이 있는데 이러한 프로젝트들은 내용, 물리적인 환경, 적용 기술 등이 다르다[21, 22]. 이러한 개별 프로젝트들은 그들 환경에 맞는 접근제어를 필요로 할 것이다. 개별 프로젝트들은 여러 측면의 상이성에도 불구하고 접근제어의 관점에서 볼 때 유비쿼터스 컴퓨팅 환경은 (그림 5)와 같이 단순화 하여 표현될 수 있다. 유비쿼터스 컴퓨팅 환경은 기본적으로 컨텍스트에 대한 수많은 센서들을 포함한다. 이러한 센서들은 시간, 기온, 습도, 빛의 밝기 및 사람의 위치, 행동의 변화 등을 실시간으로 감지하고 이를 컨텍스트 정보수집 에이전트에 넘겨준다. 컨텍스트 정보수집 에이전트는 개념적인 요소로서 센서의 정보를 수집하여 관련된 장치나 참조 모니터(reference monitor)에 전달하는 역할을 한다. 유비쿼터스 컴퓨팅 환경에서의 장치(device)는 대개 인터넷 냉장고와 같은 지능형 장치들로서 유·무선 네트워크를 통해 장치들 상호간 통신이 가능하고 센서정보나 사용자 요구에 따라 반응을 보인다. 이러한 센서와 디바이스들로 구성된 유·무선 네트워크상에서 사용자, 응용 프로그램, 디바이스 등이 어떤 정보 객체에 대한 접근을 요구하게 되고 참조 모니터는 접근을 모니터링하고 있다가 접근을 허용해야 할 것인지 거부할 것인지를 결정하게 된다. 참조 모니터를 구현하는 과정에서 구체적인 접근제어 모델이 사용될 수 있다.



(그림 5) 유비쿼터스 컴퓨팅 환경과 접근제어

3.2 유비쿼터스 컴퓨팅 환경에서 접근제어의 특징 분석

앞 절에서 살펴 본 바와 같이 유비쿼터스 컴퓨팅 환경은 기존의 일반적인 정보시스템 환경과는 여러 가지 면에서 상이하

다. 이러한 상이성은 다음과 같은 접근제어 영역에서 기존의 정보 시스템에서와는 다른 특징을 보이게 된다.

3.2.1 주체(Subject)

전통적인 정보 시스템에서 접근의 주체는 대개 인간 사용자이거나 실행중인 프로세스(process)이다. 유비쿼터스 컴퓨팅 환경에서는 유·무선 네트워크에 연결된 지능형 장치들도 접근의 주체로 참여한다. 지능형 장치를 프로세스의 일종으로 볼 수도 있겠으나 보다 독립적이고 가변적이라는 점에서 메모리상에서 실행되는 프로세스와는 구분이 된다. 지능형 장치들은 사용자의 요구에 의해, 혹은 자체의 판단에 따라 정보 객체들에 접근할 수 있다. 예를 들면 지능형 오븐은 새로운 요리법을 인터넷에서 다운 받기 위해 네트워크 장치에 접근할 수 있는데 보안을 생각할 때 이에 대한 제어가 필요하게 된다.

3.2.2 객체(Object)

전통적인 정보 시스템에서 접근의 대상이 되는 정보 객체들은 파일, 데이터베이스 내의 테이블, 프로그램 등이다. 이들이 존재하는 위치는 하드 디스크와 같은 저장장치 혹은 메인 메모리이다. 유비쿼터스 컴퓨팅 환경에서는 보다 다양한 객체들이 존재하는데 전화기, Fax, TV, 냉장고, 전등 스위치등 유·무선 네트워크에 연결된 여러 장치들이 접근의 대상이 된다. 이러한 객체들은 독립적으로 존재하며 활동하기 때문에 유비쿼터스 컴퓨팅 환경에서의 접근제어는 분산 환경에서의 접근제어의 특성을 지닌다. 또한 주체와 객체가 명확히 구분되는 것이 아니라 시점에 따라 주체가 객체가 될 수도 있고 객체가 주체가 될 수도 있다.

3.2.3 접근의 형태(Access Type)

접근의 형태(혹은 접근 권한의 형태)는 접근의 대상이 되는 객체의 특성과 관련이 있다. 전통적인 시스템에서 객체들은 하드 디스크나 메모리상에 존재했기 때문에 접근의 형태는 기본적으로 읽기(read)와 쓰기(write) 연산이며 쓰기는 추가(insert), 갱신(update), 삭제(delete) 연산으로 세분화해서 관리할 수 있다. 이러한 연산들은 시스템 혹은 참조 모니터에 의해 쉽게 식별될 수 있다. 유비쿼터스 컴퓨팅 환경에서는 객체의 형태가 보다 다양해지기 때문에 접근의 형태 역시 다양해진다. 전원의 온/오프(on/off), 터치(touch), 누르기(push), 인터넷 접속과 같은 행위들이 접근의 형태가 될 수 있다. 이러한 다양성은 시스템 혹은 참조 모니터가 접근을 식별하는 것을 어렵게 할 수 있는데, 예를 들면 단순히 전원을 온/오프 하는 연산이 아닌 '실내의 조도(照度)를 x % 올려라'와 같은 가변적인 연산이 가능할 수 있기 때문에 이러한 접근에 대한 허용 여부를 결정하는 규칙 역시 복잡해진다.

3.2.4 세션(Session)

세션이란 일반적으로 사용자(주체)가 접근 권한을 가지기 위해 시스템에 로그인한 상태 혹은 기간을 말한다. 세션은 사용자가 로그 오프 하거나 시간 초과에 의해 시스템이 강제로 세션을 종료 시킬때 까지 유지 된다. 세션을 통해 사용자는 자신에게 부여된 여러 접근 권한을 합법적으로 행사할 수 있다. 유비쿼터스 컴퓨팅 환경에서는 주체가 명시적으로 로그인 하거나 로그 오프를 하지 않는 경우가 대부분이기 때문에 세션의 생성, 유지, 종료가 불명확하다. 예를 들면 지능형 홈(intelligent home) 환경에서 사용자는 특별한 로그인 없이 TV를 켜거나 오디오를 동작시킨다. 이러한 특징으로 인해 유비쿼터스 컴퓨팅 환경에서는 사용자가 접근을 할 때 마다 사용자를 식별하고 권한을 검사해야 하는 상황이 될 수 있다.

3.2.5 권한의 유효성(validity)

주체가 객체에 접근할 수 있기 위해서는 접근 권한을 가지고 있어야 하며, 참조 모니터는 객체가 접근 권한을 가지고 있는지를 확인할 수 있기 위해서 접근 권한 정보를 관리한다. (그림 2)의 접근제어 행렬은 접근 권한 정보를 저장하고 관리하는 가장 단순한 방법이다. 전통적인 정보시스템에서는 주체가 권한을 부여 받으면 주체는 언제든지 필요할 때 자신의 접근 권한을 행사할 수 있다. 예를 들면 (그림 2)에서 John은 file1을 읽을 권한을 가지고 있기 때문에 언제든지 원하면 file1을 읽을 수 있다. 이렇게 권한이 한번 부여되면 권한을 다시 회수하기 전까지는 유지키는 접근제어 방법을 정적(static) 접근제어라고 하며 2장에서 설명한 접근제어 모델들은 모두 정적 접근제어 모델이다. 물론 정적 접근제어에서도 권한의 행사에 제약조건(constraints)를 두기도 하지만 단순하고 제한적으로 사용된다. 유비쿼터스 컴퓨팅 환경에서도 접근제어를 위해 권한 정보가 사용되지만 센서들로부터 수집된 컨텍스트 정보에 따라 권한의 행사가 영향을 받는다. 예를 들면 TV를 볼 권한이 부여된 어린이가 12세 이상 관람가 영화를 보려고 할 때 어른이 함께 있다면 권한은 행사될 수 있지만 어른이 없다면 권한은 행사될 수 없다. 따라서 주체의 입장에서 볼 때 부여된 권한은 임시적인 것으로 인식된다. 이러한 접근제어를 동적(dynamic) 접근제어라고 하는데 유비쿼터스 컴퓨팅의 특성상 동적인 접근제어가 필요로 하게 된다.

3.2절에서는 전통적인 정보시스템에서의 접근제어와 유비쿼터스 컴퓨팅 환경에서의 접근제어를 비교함으로써 유비쿼터스 컴퓨팅 환경에서의 접근제어가 갖는 특징이 무엇인지를 살펴보았다. 이를 도표로 정리하면 <표 1>과 같다.

4. 유비쿼터스 컴퓨팅 환경을 위한 접근제어의 요구사항

4.1 접근제어 요구사항

3장에서 살펴본 바와 같이 유비쿼터스 컴퓨팅 환경에서의 접근제어는 전통적인 정보시스템에서의 접근제어와는 여러 가

〈표 1〉 환경에 따른 접근제어의 특성 비교

비교항목	전통적인 정보시스템 환경에서의 접근제어	유비 쿼터스 컴퓨팅 환경에서의 접근제어
주 체	인간 사용자, 프로세스	인간 사용자, 프로세스 지능형 장치들
객 체	파일, 데이터베이스, 프로그램	지능형 장치들(지능형 장치의 특정 기능들)
접근의 형태	read, write, execute	read, write, execute turn on/off, push, touch, increase, …
세 션	생성, 유지, 종료의 안정성	세션 개념의 적용이 어려움
권한의 유효성	정적으로 유지됨	컨텍스트에 따라 동적으로 변화함

지 면에서 차이가 있다. 따라서 유비쿼터스 컴퓨팅 환경을 위한 접근제어 모델을 개발하거나 접근제어 시스템을 구축할 때는 다음과 같은 세가지 점을 고려해야 한다.

REQ(1). 유비쿼터스 컴퓨팅 환경에 포함된 주체, 객체, 접근 형태(권한)를 식별(identify)하고 정보로 표현할 수 있는 방법이 있어야 한다.

참조 모니터가 접근제어를 수행할 수 있기 위해서는 기본적으로 주체, 객체, 접근 형태(권한)가 식별될 수 있어야 한다. 유비쿼터스 컴퓨팅 환경에서는 주체, 객체, 접근의 형태가 일반 정보 시스템과는 다르므로 어떻게 이들을 식별하고 정보를 관리할 것인가를 고민해야 한다. 주체나 객체의 식별은 비교적 쉽겠지만 접근의 형태에 대한 식별은 쉽지 않다. 특히 전원을 켜고 끄는 것과 같은 단순한 동작이 아니라 실내등의 밝기를 일정 비율로 올린다던가 사람이 자동문 앞에 다가서는 것과 같은 동작은 접근 형태로서 표현하기도 어렵고 관리하기도 어렵다. 유비쿼터스 컴퓨팅 환경에서는 단순히 하나의 장치만 포함된 것이 아니라 기능과 성격이 다른 여러 장치들이 포함되어 있고 장치마나 접근 특성이 다르므로 이들에 대해 전체적 차원에서 접근제어를 시행하는 것은 어려운 일이다. 주체, 객체, 접근의 형태에 대한 식별과 표현 방법에 대한 연구가 필요하다.

REQ(2). 컨텍스트 정보를 기반으로한 동적 제어 규칙을 표현할 수 있는 방법이 있어야 한다.

유비쿼터스 컴퓨팅 환경에서는 기본적으로 컨텍스트 정보에 의해 접근제어가 이루어지는 매우 동적인 접근제어를 필요로 한다. 따라서 컨텍스트 정보를 어떻게 효과적으로 접근제어에 접목 시킬 수 있는가가 문제해결의 관건이다. 유비쿼터스 컴퓨팅 환경에서 접근제어에 대한 기존의 연구가 이 부분에 집중되어 있다는 사실은 컨텍스트 정보의 처리가 매우 중요함을 입증한다. 컨텍스트 정보가 접근제어에 접목되기 위해서는 제어 규칙의 형태로 표현되어야 한다. 제어 규칙은 "문서 보관실에 들어갈 때는 반드시 두사람 이상이 함께 들어가야 한다"와 같은 내용인데, 참조 모니터가 이와 같은 규칙을 이해하고 처리할 수 있기 위해서는 자연어 형태가 아닌 단순하고 형식적(formal)인 형태로 표현되어야 한다. 기존의 연구에서는 "if … then …" 형태로 이루어진 룰(rule)을 주로 사용하였는데, 유비쿼터스 컴퓨팅 환경에 존재하는 주체, 객체, 접근 형태의 특성을 충분히 반영하기 어렵기 때문에 제어 규칙을 보다 효과적으로 표현하고 다루기 위한 연구가 필요하다.

REQ(3). 분산환경의 특성을 구현하기 위한 방법이 제공되어야 한다.

유비쿼터스 컴퓨팅 환경은 기본적으로 분산 환경이다. 하나의 단일 하드웨어 혹은 단일 시스템으로 구현되는 것이 아니라 다양한 기능과 역할을 하는 지능형 장치들과 센서들이 네트워크로 연결되어 하나의 '환경'을 형성한다. 따라서 접근제어 모델 역시 이러한 분산 환경의 특성을 지원해야 한다. 분산환경이라는 특성은 접근제어를 실제 구현할 때 다양한 유형의 구현이 가능하게 한다. 이에 대해서는 4.2절에서 자세히 논의한다.

REQ(4). 명시적 로그온/로그오프가 없는 상태에서의 세션 관리 방법이 제공되어야 한다.

3.2절에서 살펴본 바와 같이 유비쿼터스 컴퓨팅 환경에서는 주체가 명시적 로그온 과정 없이 객체에 접근하는 경우가 많으므로 세션의 개념이 재정의 되어야 하고 세션을 유지할 수 있는 방법이 제시 되어야 한다.

REQ(5). 이질적인 장치들 간의 접근제어 커뮤니케이션을 위한 프로토콜이 있어야 한다.

유비쿼터스 컴퓨팅 환경에서 접근제어가 이루어지기 위해서는 참조모니터와 지능형 장치들 혹은 분산된 참조모니터들 간의 커뮤니케이션이 필수적이다. 유비쿼터스 컴퓨팅 환경은 서로 다른 업체에서 생산된 상호 이질적인 장치들로 구성되는 경우가 많을 것이므로 원활한 커뮤니케이션을 위해서는 약속된 프로토콜이 있어야 한다. 현재 유비쿼터스 관련 표준 활동은 주로 플랫폼에 대한 표준 제정을 위주로 하고 있으나 접근제어를 위한 프로토콜의 개발도 필요로 한다.

4.2 접근제어의 구현 형태

유비쿼터스 컴퓨팅 환경에서 접근제어 모델은 참조 모니터를 어떤 형태로 구현할 것인가에도 영향을 받는다. (그림 6)은 참조 모니터의 여러 가지 구현 형태를 보여준다. (그림 6)(a) 형태는 중앙 집중형으로서 모든 객체에 대한 접근은 하나의 참조 모니터에 의해 처리된다. 이 경우 지능형 장치는 접근제

어 기능이 없으며 중앙의 참조 모니터에게 접근의 허용 여부를 질의하고 중앙의 참조 모니터가 허용 여부를 결정하여 장치에게 통보하거나 참조모니터를 통해서 장치에 접근이 이루어지도록 한다. 중앙 집중형 접근제어는 효율성이 뛰어나지만 참조 모니터의 기능 이상시 모든 객체들에 대한 접근이 불가능해질 수 있고, 트래픽이 참조모니터에 집중된다는 단점이 있다. 또한 서로 다른 기업에 의해 만들어진 지능형 장치들에 참조 모니터와의 접근제어를 위한 표준 커뮤니케이션 프로토콜이 존재해야 한다.

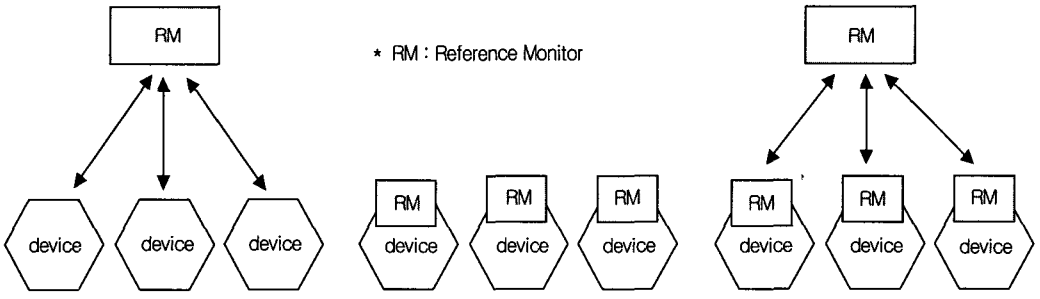
(그림 6)(b) 형태는 분산형으로서 참조 모니터가 개별 장치에 포함되어 있다. 즉, 접근제어를 개별 장치들이 각각 수행하게 된다. 각각의 장치들은 자신에게 접근이 요구될 때 허용여부를 스스로 판단한다. 이 경우 참조모니터의 구현은 장치를 만드는 개별 기업의 몫이며 개별 장치가 아닌 전체 차원에서의 접근제어 정책은 구현되기 어렵다.

(그림 6)(c) 형태는 앞의 두가지 형태를 절충한 것으로 개별 장치들도 부분적으로 접근제어를 수행하고, 전체 차원의 접근제어 정책은 중앙의 참조 모니터를 통해 수행된다. 이 경우 접근제어에 대한 부하가 분산되는 장점이 있으나 중앙의 참조 모니터와 개별 정치의 참조 모니터의 역할 분담의 문제, 표준 커뮤니케이션 프로토콜의 문제가 있다.

어떤 형태로 접근제어를 구현할 것인가는 구현하고자 하는 유비쿼터스 컴퓨팅 환경의 특성 및 형태에 달려 있다. 경우에 따라서는 개별 장치간 혹은 개별 장치와 중앙의 참조 모니터간의 접근제어를 위한 통신 프로토콜이 필요하다.

다양한 장치들로 구성된 유비쿼터스 컴퓨팅 환경에서 접근제어의 주체, 객체, 접근형태를 식별할 수 있어야함을 요구한다. 이러한 요구사항을 해결하는 한가지 방법은 객체 지향 개념에서 클래스(class)의 개념을 사용하여 접근제어의 세가지 요소를 모델링 하는 것이다. 접근제어 혹은 보안 분야에 객체지향 개념을 접목한 여러 연구들은 접근제어 분야에서 객체 지향 개념의 유용성을 잘 보여준다[17-20].

(그림 7)은 본 논문에서 제안한 방법을 나타낸다. (그림 7)에서 전자레인지와 N/W Hub는 유비쿼터스 컴퓨팅 환경에 포함된 지능형 장치를 나타내며 두 장치 모두 경우에 따라 주체, 혹은 객체의 역할을 할 수 있다. 두 장치는 클래스의 속성(attribute)으로서 ip주소, 온도와 같은 상태(status) 정보를 포함하고 있으며 이는 접근제어의 허용여부 판단시 컨텍스트 정보로 사용될 수 있다. 클래스에서 메소드(method)는 접근형태를 나타낸다. 즉 start(), stop(), set_menu()는 전자레인지에 대한 가능한 접근형태를 의미하며 전자레인지의 메소드를 호출하면 이것이 곧 전자레인지에 접근(access)함을 의미한다. 메소드는 다양한 이름으로 명명되고, 메소드는 그 요구 기능에 따라 자유로이 프로그래밍 될 수 있으므로 유비쿼터스 컴퓨팅 환경에 필요한 접근형태를 표현하는데 적합하다. 클래스 개념을 가지고 접근제어에서의 주체, 객체, 접근형태를 표현하는 이라한 방법은 유비쿼터스 컴퓨팅 환경에 포함된 지능형 장치들이 독립된 객체로서의 특성을 가지고 있기 때문에 매우 유용한 방법이다.



(a) 중앙 집중형 (b) 분산형 (c) 중앙 집중형 + 분산형

(그림 6) 접근제어의 구현 형태

4.3 접근제어 요구사항의 적용 사례

4장에서는 유비쿼터스 컴퓨팅 환경에 적합한 접근제어 모델을 위한 요구사항에 대해 기술하였다. 이러한 요구사항들은 구체적인 대상 환경에 따라 선택적으로 혹은 그 환경에 맞게 수정하여 적용되어야 한다. 본 절에서는 실제 모델의 설계시 REQ(1)이 어떻게 만족될 수 있는지에 대한 사례를 보여 주고자 한다. 이러한 사례는 본 연구에서 제시한 요구사항들을 만족시키는 방법을 찾는데 도움을 줄 것이다. REQ(1)의 내용은

전자레인지	N/W Hub
ip주소 온도 현재메뉴	ip주소 max. speed
start() stop() set_menu()	http_access() block_traffic() reset()

(그림 7) 클래스로 모델링된 장치의 예

5. 결 론

본 논문에서는 유비쿼터스 컴퓨팅 환경에 적합한 접근제어 모델을 개발하기 위한 선행 연구로서 유비쿼터스 컴퓨팅 환경의 특성 및 수행될 접근제어의 특성을 분석하고, 접근제어 모델의 개발시 필수적으로 고려해야할 요구사항에 대해 제시하였다. 유비쿼터스 컴퓨팅 환경은 많은 센서들과 지능형 장치들이 유.무선 네트워크로 연결된 환경으로서 전통적인 정보시스템과는 특성이 많이 다르기 때문에 접근제어 역시 기존의 방법을 그대로 적용하기에는 무리가 있다. 보안성의 강화라는 추세에 비추어 볼 때 적절한 접근제어 모델이 개발되어야 하는데, 마땅한 참조 자료가 없었다는 점에서 본 연구의 의의가 있다. 유비쿼터스 컴퓨팅 환경을 위한 접근제어 모델을 개발하기 위해서는 본 논문에서 제시한 세가지의 요구사항 및 구현시 어떤 형태로 할 것인지를 고려해야 한다. 4.3절에서 제시한 접근제어의 주체, 객체, 접근형태를 식별하는 방법은 다른 요구사항들을 해결하는데도 참고가 될 수 있다. 본 논문에 대한 추가적인 연구 내용은 본문중에서 제시한 연구 주제들을 보다 심도 있게 살펴보는 것과 구체적인 대상(target) 유비쿼터스 컴퓨팅 환경을 정하고 본 연구의 결과를 반영하여 접근제어 모델을 개발하고 구현하는 것이다.

참 고 문 헌

[1] Frank Stajano and Ross Anderson, "The Resurrecting Duckling : Security Issues for Ubiquitous Computing," Proc. of 7th International Workshop on Security Protocols, 1994.

[2] Upkar Varshney, "Network Access and Security Issues in Ubiquitous Computing," Proc. of Workshop on Ubiquitous Computing Environment, 2003.

[3] Rattapoom Tuchinda, "Security and Privacy in the Intelligent Environment," http://www.ai.mit.edu.

[4] Colin English, Paddy Nixon, Sotirios Terzis, Andrew McGettrick and Helen Lowe, "Dynamic Trust Models for Ubiquitous Computing Environments," Proc. of Workshop on Security in Ubiquitous Computing, 2003.

[5] Laurent Bussard and Yves Roudier, "Authentication in Ubiquitous Computing," Proc. of Workshop on Security in Ubiquitous Computing, 2003.

[6] Frank Stajano, 'Security for Ubiquitous Computing,' Wiley, 2002.

[7] Lalana Kagal, Tim Finin and Anupam Joshi, "Moving from Security to Distributed Trust in Ubiquitous Computing Environments," IEEE Computer, 2001.

[8] Jean Bacon, Michael Looyd, and Ken Moody, "Translating Role-Based Access Control Poly within Context," Proc. of International Workshop, Policies for Distributed Systems and Networks, 2001.

[9] Michael J. Covington, Wende Long, "Securing Context-Aware Applications Using Environment Roles," Proc. of Sixth ACM Symposium on Access Control Models and Technologies, 2001.

[10] Arun Kumar, Neeran Karnik, an Girish Chafle, "Context sensitivity in role-based access control," ACM SIGOPS Operating Systems Review, 2002.

[11] Charles P. Pfleeger and Shari L. Pfleeger, 'Security in Compting,' Prentice Hall, 3rd edition, 2003.

[12] Matt Bishop, 'Computer Security,' Addison Wesley, 2003.

[13] James B. D. Joshi, Elisa Bertino and Arif Ghafoor, "Temporal Hierarchies and Inheritance Semantics for GTRBAC," Proc. of 7th ACM Symposium on access Control Models and Technologies, 2002.

[14] Ravi Sandhu, Venkata Bhamidipati and Qamar Munawer, "The ARBAC97 Model for Role-Based Administration of Roles," ACM Transactions on Information and System Security, 1999.

[15] Sejong Oh, Ravi Sandhu, "A Model of Role Administration Using Organization Structure," Proc. of 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002), 2002.

[16] Mark Evered, Serge Bögeholz, "A case study in access control requirements for a Health Information System," Proc. of the 2nd workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation, Vol.32, 2004.

[17] S. A. Demurjian, T. C. Ting and M. Y. Hu, "Role-Based Access Control for Object-Oriented/C + + Systems," Proc. of first ACM Role-Based Access Control Workshop, 1995.

[18] Yan Han, Liu Fengyu, and Zhang Hong, "An Object-Oriented Model of Access Control based on Role," ACM SIGSOFT Software Engineering Notes, Vol.25, No.2, 2000.

[19] Gustaf Neumann and Mark Strembeck, "Design and Implementation of a Flexible RBAC-Service in an Object-Oriented Scripting Language," Proc. of the 8th ACM conference on Computer and Communications Security, 2001.

[20] Mark Evered, "Flexible Enterprise Access Control with Object-oriented View Specification," Proc. of Australasian Information Security Workshop 2003 (AISW2003), 2003.

[21] 이성국, 김완석, '세계 각국의 유비쿼터스 컴퓨팅 전략', 전자신문사, 2003.

[22] 김완석, 김정국, 박범수, 박태웅, 이성국, "유비쿼터스 컴퓨팅 전략 및 정책", 한국디지털정책학회 창립학술대회 학술논문지, 2003.

오 세 종

e-mail : sejongoh@dankook.ac.kr

1989년 서강대학교 컴퓨터학과(학사)

1991년 서강대학교 대학원 컴퓨터학과 (공학석사)

2001년 서강대학교 대학원 컴퓨터학과 (공학박사)

1991년~1997년 대우정보시스템 근무

2001년~2003년 미국 George Mason University Post Doc. 연구원

2003년~현재 단국대학교 공학대학 컴퓨터과학과 전임강사

관심분야 : 정보시스템, 정보보호, 데이터베이스, 유비쿼터스 컴퓨팅

박 제 호

e-mail : dk_jhpark@dankook.ac.kr

1985년 서강대학교 전자계산학과 학사,

1993년 N.Y. Polytechnic Univ. Computer Sci. M.S.

2001년 N.Y. Polytechnic Univ. Computer Sci. Ph.D.

2001년 System Architect at Voicemate (NYC, USA)

2003년~현재 단국대학교 컴퓨터과학과 전임강사

관심분야 : DB 시스템 구조, 분산 DB, Information Architecture, Bioinformatics