DOI: 10.3745/KIPSTC.2008.15-C.6.493

u-헬스케어를 위한 상황기반 동적접근 제어 모델 및 응용

정 창 원[†] · 김 동 호[††] · 주 수 종[†††]

요 약

본 논문은 역할 기반 접근 제어 모델 연구를 통해 u-헬스케어 환경의 요구사항을 충족하는 상황 기반의 동적 접근제어 모델을 제안한다. 동적 보안 도메인 관리를 위해 분산객체그룹 프레임워크를 사용하였고, 동적 접근제어를 위한 상황 정보는 기 구축된 데이터베이스를 사용하였다. 본 논문에서는 러프집합 이론을 근거로 의사결정 테이블에서 지식감축을 통해 의사결정 규칙을 정의하고, 생성된 규칙을 동적 접근제어 모델에 적용하였다. 그리고 분산객체그룹 프레임워크를 기반으로 u-헬스케어 응용을 통해 상황기반 동적 보안 서비스의 수행 결과를 보였다. 이 결과, 동적 접근 제어 모델은 u-헬스케어 환경에서 보안 도메인에 따라 적합한 보안 서비스와 보다 유연한 접근제어를 제공한다.

키워드 : 보안, 역할 기반 접근제어, 동적 접근제어, 상황인식, 분산객체그룹 프레임워크, 유비쿼터스

Context-based Dynamic Access Control Model for u-healthcare and its Application

Jeong Chang Won[†] · Kim Dong Ho[††] · Joo Su Chong[†††]

ABSTRACT

In this paper we suggest dynamic access control model based on context satisfied with requirement of u-healthcare environment through researching the role based access control model. For the dynamic security domain management, we used a distributed object group framework and context information for dynamic access control used the constructed database. We defined decision rule by knowledge reduction in decision making table, and applied this rule in our model as a rough set theory. We showed the executed results of context based dynamic security service through u-healthcare application which is based on distributed object group framework. As a result, our dynamic access control model provides an appropriate security service according to security domain, more flexible access control in u-healthcare environment.

Keywords : Security, Role Based Access Control, Dynamic Access Control, Context Aware, Distributed Object Group Framework, Ubiquitous

1. 서 론

유비쿼터스 컴퓨팅 환경은 센서나 태그를 이용하여 개체를 실시간으로 감지하고, 다양한 데이터를 처리하여 생성된 정보를 자동으로 전달하여 사물들의 네트워크화를 지향하며, 궁극적으로 사람, 컴퓨터, 사물들을 네트워크로 연결하고 3차원으로 정보를 전달하는 차세대 기술을 제공한다[1]. 이러한 환경에서는 다양한 기기와 시스템들이 유무선 네트워크로 연결되어 정보 또는 객체들이 분산되어 있는 시스템들 사이를 이동하고, 정보에 대한 접근이 언제 어디서나 어떠한 장치로나 가능함에 따라 보안 문제가 발생할 수 있다. 따라서 이러한 문제점을 해결하기 위해 동적으로 재구성되는 보안 도메인에 따라 시간과 공간 그리고 사용자의 상황에 따라 보안 서비스를 제공하기 위한 새로운 보안 모델이 요구된다.

기존 분산객체그룹 프레임워크에서 제공하는 보안서비스는 ACL(Access Control Lists)을 통해 클라이언트 객체와 서버 객체간의 접근을 제어하였으나 사용자의 위치와 근무시간 그리고 역할의 변화와 같은 사용자의 상황이 바뀌어도 정보에 대한 접근 권한이 유지됨에 따라 유비쿼터스 환경의 응용에 적용하기에 미흡하다.

이러한 문제점을 해결하기 위한 연구로 역할 기반 접근제어(RBAC:Role Based Access Control)[2,3,4] 모델은 접근권한을 역할에 따라 그룹화하고 사용자 개개인의 책임과 권

※ 이 논문은 2008년 교육과학기술부로부터 지원받아 수행된 연구임(지역거점 연구단육성사업/헬스케어기술개발사업단)
† 정 회 원 : 원광대학교 전기전자 및 정보공학부 박사후 연구원
†† 정 회 원 : (주)PC 닥터 부설 연구소 연구원
††† 정 회 원 : 원광대학교 전기전자 및 정보공학부 교수(교신저자)
논문접수 : 2007년 10월 10일
수 정 일 : 1차 2008년 8월 18일
심사완료 : 2008년 9월 2일

한을 역할에 부여하여 자원에 대한 접근 제어를 통해 보안 서비스를 제공함으로써 보안 관리의 효율성을 극대화시켰다. 그러나 이 모델은 유비쿼터스 환경에서 동적으로 재구성이 가능한 보안 도메인에 따른 보안 서비스나 상황에 따라 변하는 사용자의 역할에 부여되는 권한의 변경에 대한 보안 요구 사항을 충족시키기에 한계가 있다. 특히, 동일한 접근 권한이 있는 객체일지라도 보안 도메인이 재구성되거나 객체나 객체의 환경, 또는 시스템의 상황에 따라 접근 가능한 정보를 제한하는 동적 접근제어 기술이 요구된다.

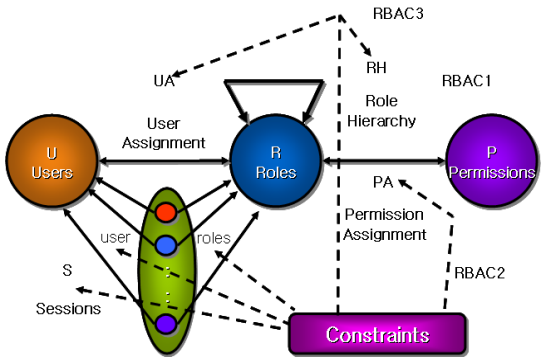
따라서 본 논문에서는 이러한 u-헬스케어 환경에서 발생할 수 있는 보안 취약점과 다양한 보안 요구사항을 만족시키기 위해 역할 기반 접근 제어 기술을 기반으로 하는 상황기반 보안 모델을 제안한다. 이를 위해 분산객체그룹 프레임워크(DOGF :Distributed Object Group Framework)[5,6]의 그룹 개념을 보안 도메인에 적용하였다. 특히, u-병원을 구성하는 보안 도메인 상에서 다루어지는 상황정보를 사용자-역할과 환경 그리고 시스템, 사용자-장치간 상호작용이력상황으로 분류하고, 헬스케어 응용에 적용하기 위한 세부 상황으로 정의한 후, 러프집합 이론을 이용한 동적 접근제어 모델을 제안한다. 그리고 이를 헬스케어 응용에 적용하여 수행성을 확인한다.

2. 관련연구

본 장에서는 관련 연구로 역할기반 접근제어[2,3,4,9]와 상황기반의 접근 제어 모델에 대해 살펴본다. 그리고 기존 모델에 대한 분석을 통해 유비쿼터스 컴퓨팅 환경에서 변화되는 상황에 따라 동적인 보안 서비스를 제공하는 보안 모델의 필요성에 대해 기술한다.

2.1 역할기반 접근제어 모델

역할 기반 접근제어는 전통적인 제어 기법과 달리 정보에 대한 사용자의 권한 부여 여부를 각 사용자나 이미 정의된 접근제어 규칙에 의해 판단하지 않고, 사용자가 소속된 그룹, 즉 역할에 기반을 두고 시스템 자원에 대한 접근 제어를 하는 기법이다. RBAC 모델의 특징은 권한을 부여하는 단위가 사용자 대신 사용자가 할당되도록 분류해 놓은 역할이라는 점이다. 역할과 객체간의 관계로 접근 권한을 관리함으로써, 사용자와 객체의 수가 많고 구성이 수시로 변할 수 있는 유비쿼터스 컴퓨팅 환경에서 효율적인 권한부여 및 권한 관리를 가능하게 한다. 또한, 역할 간 계층구조를 통해 하위 역할에 할당된 권한이 상위 역할에 의해 사용될 수 있는 권한상속(permission inheritance)을 제공한다. 권한상속을 이용하여 계층구조를 가진 역할들에 대한 권한부여를 보다 효과적으로 수행할 수 있다. 이러한 방식은 권한 관리를 단순화시켜 줄 뿐만 아니라, 보안정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다. Sandhu는 역할 기반 접근제어 기술에 대해 다음과 같은 네 가지 모델로 구분하였다[3,4]. 역할 기반 접근 제어의 기본 모델인 RBAC0과 기본



(그림 1) RBAC 모델

모델에 역할의 상속 개념인 역할 계층(role hierarchy)을 추가한 RBAC1, 기본 모델에 제약 조건(constraints)을 추가시킨 RBAC2, 그리고 RBAC1과 RBAC2의 통합 모델인 RBAC3으로 구분하였다. (그림 1)은 이러한 역할 기반 접근제어 모델의 특징을 보인다.

그러나 이러한 보안 모델은 기존 플랫폼 지향적인 시스템에서 적용이 가능하지만, 다중 플랫폼 상에 수시로 재구성이 가능한 구성 요소와 이에 따라 변하는 보안 도메인 그리고 자원 제약성 및 시스템 환경의 가변적인 상황과 같은 특성을 갖는 유비쿼터스 컴퓨팅 환경에서는 보안 문제가 발생할 수 있으므로, 이를 해결하기 위한 동적인 특성을 갖는 보안 메커니즘이 필요하다.

2.2 상황정보를 이용한 접근제어 모델

기존의 역할기반 접근제어는 시간과 위치와 같은 상황에 근거한 접근제어를 수행할 수 없어 이를 해결하기 위해 GRBAC 모델이 제안되었다. GRBAC 모델[10]은 접근제어 결정에 사용자 역할(subject role), 객체 역할(object role), 환경 역할(environment role)을 사용함으로써 기존 역할기반 접근제어를 확장하였다. 사용자, 객체, 환경 요소를 역할로 구조화함으로써 접근제어 정책 기술의 단순함(simplicity)과 융통성(flexibility)을 제공한다.

보안 관리자는 접근권한 정보로 주체역할(subject role), 객체역할(object role), 환경역할(environment role), 연산(operation), 접근기호(sign)의 다섯 가지 구성 요소를 사용하여 기술한다.

<<Jane, medical record, weekdays, write>, +>
<<doctor, history case, weekends, read>, ->

위의 예에서 doctor 역할을 할당받은 사용자는 weekends에 history case를 읽을 수 없음을 나타낸다.

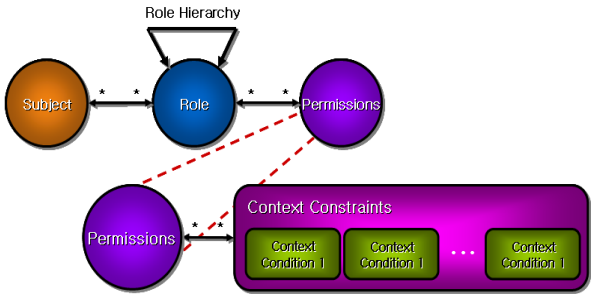
역할계층 구조에 의해 발생하는 비명시적인 권한 부여 해결을 위해 역할 계층 구조에서의 권한상속 개념을 이용한다. 권한상속은 standard, strict, lenient의 세 가지 타입이 있다.

GRBAC 모델은 상황정보를 환경 역할로 정의하여 접근제어 정책에 기술하고 전달 규칙을 적용하여 사용자의 접근 요청을 처리한다. 그러나 접근 권한의 전달로 발생하는 권

한들 사이의 충돌에 대하여 해결 방안을 제시하지 않았고, 사용자가 사용자의 고려사항을 환경 역할로 정의함에 많은 계층 구조가 발생하여 관리의 어려움이 따른다.

Gustarf Neumann과 Mark Strembeck는 상황정보를 접근제어 결정에 이용하기 위하여 역할 기반 접근제어의 제약사항을 사용하는 xoRBAC 모델을 제안하였다[11]. 상황 제약사항(context constraint)은 상황정보 속성의 실제값을 미리 정의된 조건과 비교하는 역할 기반 접근제어 제약사항으로 특정 연산의 수행을 허용하기 위해 상황정보 속성이 충족되어야 할 조건을 기술한다. 상황 제약사항은 속성(context attribute), 함수(context function), 조건(context condition)의 튜플(tuple)을 갖는다. 속성은 시간, 요일과 같은 동적으로 변화하는 속성을 나타내거나 위치, 소유관계, 생일이나 국적과 같은 객체의 인스턴스에 따라 변화하는 속성을 나타낸다. 권한 결정은 특정 주체 또는 역할이 가지는 권한에 따라 이루어짐에 따라 (그림 2)와 같이 상황제약사항도 권한과 관련된다. 권한은 여러 개의 상황 제약과 관련되고 모든 상황 제약이 참값을 가질 때 접근이 허용된다.

이 모델은 상황 정보를 상황 정보제약에 기술하고 각 권한에 대하여 상황 정보제약을 둔다. 사용자의 접근 요청은 해당 객체에 대한 연산의 권한이 갖는 상황정보제약이 모두



(그림 2) 상황제약과 관련된 xoRBAC 모델

참 값을 가질 때 허용 또는 거부된다. 따라서 각 객체의 권한에 대하여 상황 정보제약이 기술되므로 사용자의 상황에 따른 접근 제어를 수행하고자 할 경우 최악의 경우 (사용자수*2^컨텍스트의 수) 의 제약사항이 기술이 필요하다. 사용자의 접근요청이 발생하였을 때 권한을 부여하기 위하여 상황 정보제약의 탐색과 평가하는 지연 문제를 발생시킨다. <표 1>은 상황의 적용 대상, 정보 관리와 적용방법, 종류 및 분류에 있어서 기존 모델의 특징을 나타낸다.

2.3 제안 보안 모델의 필요성

본 논문에서 제안하는 동적 보안 서비스는 분산객체그룹 프레임워크의 보안객체가 제공한다[7,8]. 기존 분산객체그룹 프레임워크에서 제공하는 보안 서비스는 ACL기반의 보안 서비스를 제공하지만 이 방법을 이용하여 보안을 유지하기에는 서버 객체에 대한 접근권한을 얻은 클라이언트 객체에 대한 정보만을 포함하고 있기 때문에 상황에 따라 서버 객체에 대한 접근을 제어하는데 적합하지 못하다. 특히, 유비쿼터스 환경은 다양한 환경과 상황에 따라 동적으로 변하는 보안 도메인을 통해 접근 제어를 해야 하며, 또한 사용자 역할의 변경과 시스템 환경의 변화에 적응적인 접근 제어를 통해 보안을 유지해야 한다.

이를 해결하기 위해 본 논문에서는 관련연구의 기존 보안 모델에서 RBAC0을 기반으로 역할 계층과 상속에 대한 개념을 사용하고, 상황정보에 따라 동적으로 역할 배정과 객체에 대한 접근모드를 할당한다. 그리고 유비쿼터스 컴퓨팅 환경의 특성에 따라 다중 플랫폼 상에 수시로 재구성되는 보안 도메인을 관리하기 위해 분산객체그룹 프레임워크의 객체그룹 개념에 적용한다. 또한 기존 보안 모델에서 상황정보를 하나의 역할이나 제약조건으로 제한하지 않고 보안 모델에서 동적인 특성을 갖는 구성요소로 정의하여, 상황정보를 주체, 역할, 객체 모두에 적용한다. 이를 u-헬스케어

〈표 1〉 접근제어 모델 비교

	항목	RBAC[3]	xoRBAC[11]	GRBAC[12]
모델 평가	보안 도메인	정적	정적	정적
	보안 도메인 관리	수동적	수동적	수동적
	상황 적용 대상	X	객체	주체, 객체
	동적 접근 제어	X	O	O
관리	상황 관리	상황 정보 고려하지 않음	정형화된 구조가 없어 관리가 어려움	계층구조만 이용
	정책 기술	만족되어야 할 조건만 검사	만족되어야 할 조건만 검사	만족되어야 할 환경 역할만 나열
수행	상황 활성화	상황 정보 고려하지 않음	만족해야할 조건만 검사	현재 활성화된 환경 역할과 정책에 기술된 상황과 비교
	권한 탐색	주체, 객체, 역할에 대한 정형화된 규칙 탐색	규칙 탐색 및 객체에 대한 환경 조건을 기술한 제약사항 탐색	주체, 객체, 환경 역할에 대한 규칙 탐색
	권한 평가	탐색한 규칙들의 권한 비교	제약 사항의 조건 검사만으로 권한 부여	탐색한 규칙들의 권한을 비교 및 충돌을 고려한 후 권한 부여
	구현의 복잡성	역할과 권한 정의	상황의 정의와 계층구조, 활성화 프로세서, 탐색과 평가 프로세서 필요	환경 역할의 정의와 계층구조, 환경 역할의 활성화 필요

환경에 적용하기 위해서 각 보안 도메인 상에서 가능한 상황과 적절한 접근 제어를 가능케 하기 위해 러프집합 이론을 이용하여 의사결정 테이블을 통해 규칙을 생성하고, 이를 통해 동적인 접근제어가 가능하도록 한다. 러프집합 이론을 기반으로 보안 도메인의 구성요소로부터 수집되는 다양한 상황정보를 분류하고, 의사결정 테이블을 통해 범주화를 통해 지식 감축을 하여 최적화 규칙을 생성하고, 이를 u-헬스케어 응용에 적용하여 상황에 따라 정보 및 자원에 대한 접근을 동적으로 제어함을 응용의 결과화면을 통해 보인다.

3. 상황기반 동적 보안서비스

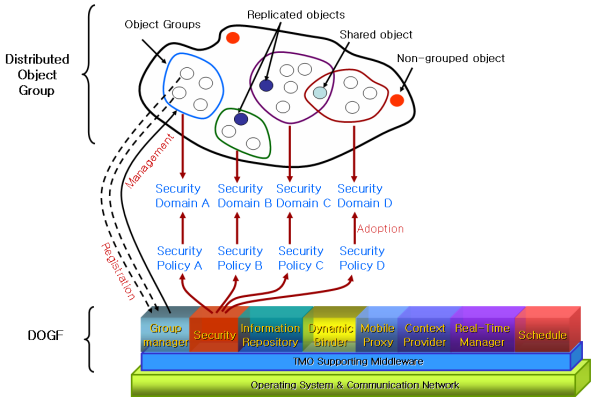
본 장에서는 유비쿼터스 컴퓨팅 환경에서 적용이 가능한 상황정보를 이용한 보안 서비스에 대해서 기술한다. 이를 위해 먼저 u-헬스케어 컴퓨팅 환경에 따르는 보안 도메인에 대해서 살펴보고, 보안 도메인상의 상황정보 그리고 본 논문에서 제안하는 동적 접근제어 모델에 대해 기술한다.

3.1 보안 도메인

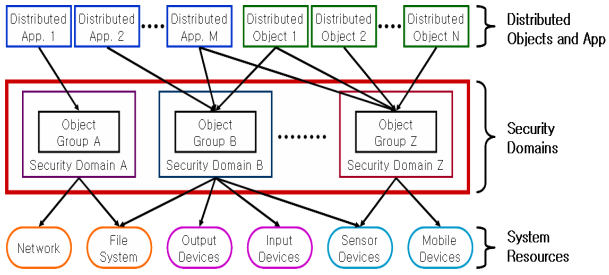
유비쿼터스 컴퓨팅 환경은 진행 중인 프로젝트의 종류에 따라 다양하게 정의되고 있다. 유비쿼터스 관련 프로젝트는 Oxygen Project[12], Smart Dust[13], Aura Project[14] 등이 있는데, 이러한 연구들은 내용, 물리적인 환경, 적용기술이 서로 상이하다. 그러나 개별 프로젝트들은 다양한 측면에서의 이질성에도 불구하고 보안 측면에서 특정 데이터에 대한 접근제어와 관련해 기밀성의 확보라는 공통적인 요구사항을 가지고 있다.

접근제어가 요구되는 보안 도메인은 기존 일반적인 플랫폼 지향 시스템에서 플랫폼에 의존적인 특정 도메인으로 제한되었다. 그러나 유비쿼터스 컴퓨팅 환경은 다중 플랫폼 상에 자원을 포함한 객체들의 집합으로 도메인을 구성한다. 이러한 도메인은 중첩되거나 서브 도메인을 포함하며, 필요에 따라 도메인이 형성되거나 해체가 동적으로 재구성 된다[6]. 이에 따라 보안 정책은 서로 다르게 적용되어야 하며, 도메인이 해체될 경우 각 분리된 도메인이 갖는 보안 정책은 재구성되어야 한다. 따라서, 본 논문에서는 u-헬스케어 환경에서 동적으로 재구성되는 보안 도메인을 관리하기 위해 분산객체그룹 프레임워크[5,6]를 사용하였다.

보안 도메인은 (그림 3)에서 나타난 바와 같이 분산객체 그룹 프레임워크에 의해 관리되는 객체그룹을 기준으로 구성되며, 보안 도메인에 따라 서로 다른 보안 정책을 따른다. 보안 도메인 A의 경우는 객체의 접근권한을 갖는 개별적인 분산객체들로 그룹화되어 동일한 보안 정책의 적용이 가능하다. B와 C의 경우에는 중복된 분산객체를 포함하고 있으며, 중복된 분산객체는 서로 다른 도메인에 소속되고 서로 다른 보안 정책이 적용되어 사용자의 역할과 상황에 따라 상이한 수행결과를 보인다. C와 D의 경우에는 중첩된 분산객체를 포함하고 있으며 서로 다른 보안 정책이 적용되나



(그림 3) 보안 도메인



(그림 4) 분산객체그룹 프레임워크의 보안 도메인

사용자의 역할과 상황에 따라 동일한 수행결과를 보인다. 즉, 분산객체그룹 프레임워크에 의해 형성된 보안 도메인에 따라 분산객체의 접근권한은 일관된 보안 정책에 의해 관리된다.

이에 대해 세부적으로 살펴보면, 보안 도메인은 실행 가능한 응용인 분산객체 혹은 유비쿼터스 응용의 특징에 따라 시스템 자원에 접근할 수 있는 객체들을 분리시켜 단일 시스템 또는 여러 시스템 영역 내에 가상적인 객체그룹에 따라 구성된다. 접근권한이 허용된 분산객체 또는 분산 응용이 접근 가능한 시스템 자원 부분과 접근이 불가능한 경우를 분산객체그룹에 의해 구분할 수 있다. 이는 분산 응용을 구성하는 객체그룹에 따라 다중 보안 정책을 적용할 수 있어 일관된 보안 정책에 유연성 및 안정성을 제공한다.

본 논문에서 보안을 위해 정의된 보안 도메인은 시스템 자원 부분과 분산 응용으로 나누어진다. 시스템 자원 부분으로는 파일 시스템, 네트워크, 모니터, 키보드, 센서, 모바일 장치 등이며, 분산 응용으로는 분산객체 및 분산 응용이 있다. 이에 대한 구조는 (그림 4)와 같다. 객체그룹 구조는 여러 시스템 상에 위치한 분산객체들의 그룹화를 통해 보안 도메인을 형성하고, 이러한 영역은 객체그룹 식별자를 통해 구분한다. 또한 객체그룹을 구성하는 분산객체에 대한 동적 바인딩객체를 통해 하나의 에러가 전체적인 시스템 자원 관리 부분에 치명적인 위협이 되는 문제를 해결한다.

3.2 보안 상황정보

u-헬스케어 환경은 기본적으로 다양한 센서들을 포함하

〈표 2〉 u-헬스케어 환경에서의 상황정보의 계층화

일반상황	세부상황	u-병원 환경 세부상황
사용자-역할 상황 (CI_1)	신원 상황 (CI_11)	성명, ID, 직원번호
	역할 상황 (CI_12)	관리자, 의료진, 담당 의료진, 의사, 간호사, 원무과직원, 보호자, 피보호자, 노인, 어른, 아동, 외부인
	신체 상황 (CI_13)	맥박, 혈압, 혈당, 체온, 음성
환경 상황 (CI_2)	공간 상황 (CI_21)	위치, 방향, 속도
	시간 상황 (CI_22)	일자, 시각, 계절
	환경 상황 (CI_23)	온도, 조도, 습도, 소음, 진동
	활동 상황 (CI_24)	인접인, 이동경로, 일정
시스템 상황 (CI_3)	객체 상황 (CI_31)	객체명
	소속된 객체그룹 (CI_32)	그룹명
	가용자원 (CI_33)	전원, 영상장치, 출력장치, 통신장치
	접근 상황 (CI_34)	사용자, 접근허용정보, 인접성
사용자-장치 간 상호작용 이력 상황 (CI_4)	이력 상황 (CI_41)	사용자, 서비스, 시간
	장애 상황 (CI_42)	시간-사용자-서비스

고 있다. 이러한 센서들은 시간, 위치, 환경, 행동의 변화 등을 실시간으로 감지하고 처리하며, 상황-인식 컴퓨팅 분야에서는 이러한 정보를 가공하여 상황 정보로 사용한다. 그러나 특정 어플리케이션에 적합한 상황 정보가 정의되어 사용되고 있어 상황 정보에 대한 일반화된 정의가 없는 상태이다.

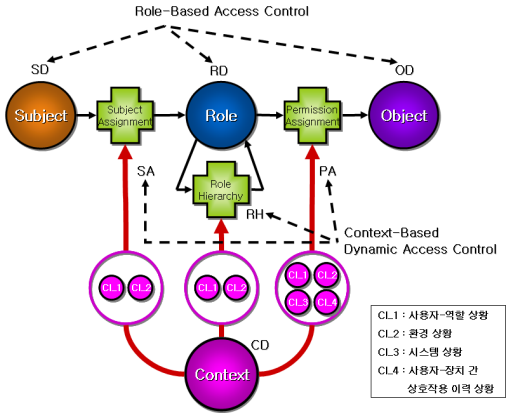
따라서 u-헬스케어 환경에서 동적 보안 서비스를 제공하기 위한 상황정보는 클라이언트 프로그램 개발자가 효율적으로 관리하고, 접근제어 모델에 효과적으로 적용하기 위해 <표 2>와 같이 계층화하고 분류하여 정의하였다.

상황정보는 홈 환경에 위치한 다양한 센서들로부터 수집한 데이터와 이를 가공한 데이터를 처리하는 기 구축된 데이터베이스와 데이터베이스 관리 도구, 상황 정보 생성 GUI를 사용하여 생성하였다. 데이터베이스에서는 기본 정보와 상황정보가 분류되어 관리 된다. 기본 정보는 물리적 센서로부터 얻어지는 위치, 건강, 환경관련의 데이터와 사용자 입력을 통한 개인 건강관련 프로파일 정보로 이루어져 있고, 상황정보는 기본 정보들을 이용하여 혼합 가공한 정보로 응용과 서비스에 따라 구성된 다양한 뷰 스키마를 통해 얻어진다[15].

3.3 동적 접근제어 모델

본 절에서는 상황기반의 동적 보안 서비스를 수행하기 위한 보안 모델에 대해 기술한다.

상황기반의 동적 접근제어 모델은 분산객체그룹 프레임워크[5,6]의 보안객체에 적용되어, 동적으로 재구성되는 보안 도메인에 따라 능동적인 보안서비스를 제공한다. 동적 접근제어 모델에 적용된 보안 규칙은 지식의 감축을 통해 최소 의사결정 규칙을 생성하는 러프집합 이론[16,17]을 적용하여



(그림 5) 동적 접근제어 모델

생성하였다.

제안하는 동적 접근제어 모델은 역할 기반 접근제어 모델을 기반으로 u-헬스케어 환경 내의 다양한 연령의 개별 사용자들, 센서와 모바일기기 및 컴퓨터를 포함하는 장치, 그리고 유비쿼터스 서비스 모델 등의 특성을 고려하여 구성하였다. (그림 5)는 본 논문에서 제안하는 동적 접근제어의 기본 모델로 구성요소에는 주체 요소(S: Subject), 역할 요소(R: Role), 객체 요소(O: Object), 상황 정보 요소(C: Context)가 있다.

주체 요소는 u-헬스케어 환경 내의 시스템 자원을 사용하기 위해 능동적으로 접근 권한을 요청하는 요소이며, 접근권한이 부여되어 있는 하나 이상의 역할 요소에 배정 된다. 또한, 분산객체그룹 프레임워크의 그룹 관리자 객체에 의해 등록되고 관리되는 요소로 사용자, 분산객체 및 분산객체그룹, 유비쿼터스 응용이 주체 요소와 일대일 관계로 연관된다. 이

〈표 3〉 구성요소 정의

SD = {S_1, S_2, ... , S_M, SG_1, SG_2, ... , SG_N}, 1≤i≤M, 1≤j≤N : 주체 요소 도메인(Subject Domain)
RD = {$Role_1$, $Role_2$, ... , $Role_M$}, 1≤i≤M : 역할 요소 도메인(Role Domain)
OD = {$Object_1$, $Object_2$, ... , $Object_M$}, 1≤i≤M : 객체 요소 도메인(Object Domain)
CD = {$Context_1$, $Context_2$, ... , $Context_M$}, 1≤i≤M : 상황정보 요소 도메인(Context Domain)
SA ⊆ SD X RD : 사용자 역할 할당 정의(Subject Assignment)
RH ⊆ RD X RD : 역할 상속 정의(Role Hierarchy)
PA ⊆ RD X OD : 역할-객체 간 접근권한 할당 정의(Object Assignment)

렇게 생성된 주체 요소는 분산객체그룹 프레임워크에 의해 객체 그룹을 형성하고 보안 도메인으로 관리된다.

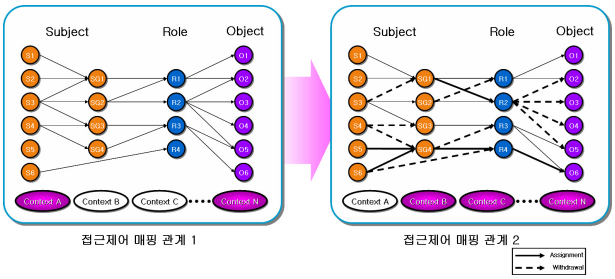
역할 요소는 주체 요소가 소속되는 그룹으로 객체 요소에 대한 접근 권한이 부여되는 구성 요소이다. 역할 요소가 갖는 권한은 그 역할 요소에 소속된 모든 주체 요소에게 동일하게 적용되며, 보안 정책에 따라 역할 요소 간 계층구조를 통해 표준(standard), 엄격(strict), 관대(lenient)의 세 가지 타입으로 하위 역할 요소에 할당된 권한이 상위 역할 요소에 의해 사용될 수 있도록 권한 상속을 제공한다. 주체 요소의 그룹인 역할 요소로 접근 권한을 관리함으로써, 유비쿼터스 컴퓨팅 환경에서 수시로 재구성이 가능한 보안 도메인에 대해 효율적인 권한 부여 및 권한 관리를 가능하게 한다.

객체 요소는 주체 요소가 접근권한을 요청하는 그 대상으로 특정한 기능을 수행하는 수동적 개체인 자원과 수행 객체들의 집합이다. 유비쿼터스 컴퓨팅 환경에서 자원은 전통적인 접근제어 모델에서처럼 주체 요소와 객체 요소가 명확히 구분되지 않고 접근권한을 요청하는 시점에 주체 요소와 객체 요소로 나뉘어 접근제어가 이뤄진다.

상황정보 요소는 제안하는 보안 모델에 동적인 특성을 부여하는 요소로 보안 정책에 부합하는 접근제어 결정에 이용하는 입력값이다. 이는 분산객체그룹 프레임워크의 상황정보 제공자 객체가 제공하며, 동적 접근제어 규칙을 생성하기 위해 사용된다. <표 3>은 동적 접근제어 모델의 구성요소를 정의한 것이다.

(그림 6)은 하나의 보안 도메인 내에서 보안 정책에 따라 구성이 가능한 구성 요소 간의 관계를 그림으로 나타낸 것이며, 상황에 따라 접근제어 매핑관계 1과 접근제어 매핑관계 2와 같이 각각 다양한 방법으로 연관되어질 수 있다.

이러한 구성에서 상황을 고려하지 않은 기존의 역할기반 접근제어는 접근제어 매핑관계 1, 2에 대하여 다음과 같은 접근 제어 결과를 보인다. 접근제어 매핑관계 1에서는 객체 요소 O2의 접근제어 리스트에는 접근이 허가된 역할 요소 R1과 R2가 기록되며, 객체 요소 O6의 접근제어 리스트에는 R3만이 기록되어 객체 요소 O2에 대해 역할 요소 R2는 접근이 허가되고 O6에 대해 R4는 접근이 거부된다. 하지만 접근제어 매핑관계 2에서는 객체요소 O2의 접근제어 리스트에는 역할 요소 R1만이, 객체 요소 O6의 ACL에는 역할 요소 R3과 R4가 기록되어 접근제어 매핑 관계 1에서는 O2에 접근이 허가 되었던 R2가 접근이 거부되고 O6에 접근이 거



(그림 6) 상황에 따라 변하는 구성 요소들 간의 관계

부되었던 R4는 접근이 허가되는 서로 다른 접근제어 리스트를 갖게 된다. 기존의 역할 기반 접근제어 규칙은 상황 정보에 대한 변화를 고려하지 않는 정적인 접근제어 규칙을 사용하여 모순된 접근제어 결과를 보인다.

3.4 보안 정책 및 동적 접근제어 규칙 생성

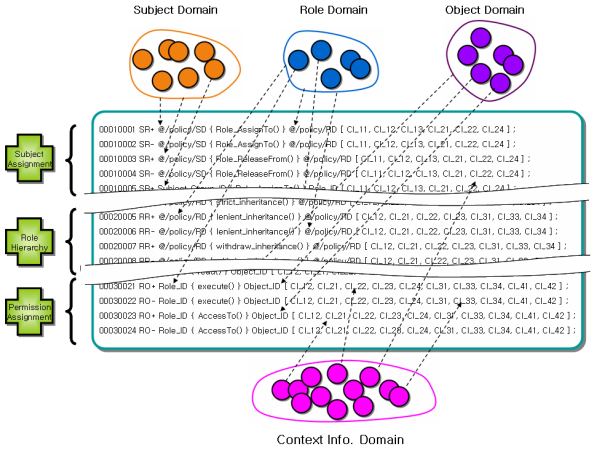
보안 정책은 주체 요소에 할당할 수 있는 역할 요소, 역할 요소가 각 객체 요소에 대해 수행할 수 있는 연산과 이들 간의 관계에 따르는 기능적인 역할을 정의한 것으로, 접근제어 규칙을 생성하는 기준 정보가 된다. 동적 접근제어 모델에서 사용하는 보안 정책은 객체의 보안 정책 관리에 정책 수준의 역할 기반 접근제어 기법을 적용하여 동적으로 재구성되는 보안 도메인에 대한 보안 정책의 변경, 삭제 및 삽입의 보안 정책 관리에 효율성을 제공하고, 다양한 기능의 보안 도메인 특성에 맞는 일관된 보안 정책 구현에 융통성을 제공한다.

보안 정책은 다음과 같은 속성들로 기술한다.

```
Policy_ID mode component1 '{' action '}'
component2 [context] [description] ';'
```

'Policy_ID'는 정의한 정책에 대한 정책식별자이며, 'mode'는 주체-역할 배정 정책(SR)인지 역할 상속 정책(RR)인지 역할-객체 접근권한 정책(RO)인지를 나타내며, 정책 적용 시 허용을 의미하는 태그 '+'와 거부를 표현하는 태그 '-'를 추가로 표기가 가능하다.

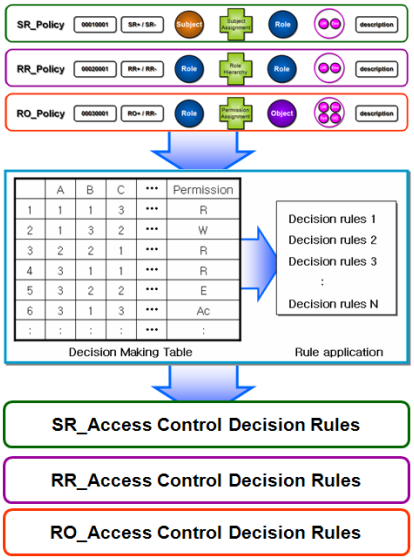
또한, 'component1'과 'component2'는 주체 요소, 역할 요소, 객체 요소에 해당하며 하나 이상의 'action'요소에 의해 'component1'은 'component2'에 대한 행위가 가능하다. '[context]'는 상황정보 요소로 보안규칙 생성 시 기준이 되



(그림 7) 동적 보안 모델 구성요소와 기본 보안 정책

는 상황정보들이며, '[description]'은 정책에 관한 기술항목으로 선택사항이다. 유비쿼터스 컴퓨팅 환경의 구성요소와 보안 정책 간의 관계는 (그림 7)과 같다.

보안 정책은 접근제어 규칙을 생성하는 기준을 제시하며, 보안 정책에서 제시한 상황 정보들과 주체 요소, 역할 요소, 객체 요소를 입력으로 하여 주체-역할 배정 규칙, 역할 상속 규칙, 역할-객체 접근권한 규칙을 생성한다. 접근제어 규칙은 다양한 상황정보와 각 상황정보의 속성값을 모두 고려해야 하기 때문에 객관성이 불충분하거나 애매모호한 데이터들로부터 구체적이고 체계적인 핵심정보를 추출해내는 기법인 러프 집합 이론을 채택하였다[16,17]. (그림 8)은 u-헬



(그림 8) 보안 정책 기반의 동적 접근제어 규칙 생성 과정

스케어 환경에서 보안 도메인의 구성요소와 세부상황 중 핵심 정보만을 추출하고 의사결정표의 간략화를 통해 동적 접근제어 규칙을 생성하는 과정을 보이고 있다.

본 논문에서는 동적 접근제어 규칙 중 역할-객체 접근권한 규칙을 생성하는 방법을 보인다. 먼저 규칙을 생성하기 위해서 의사 결정 데이터 테이블을 구성하기 위한 조건-속성 정의가 필요하다. 데이터베이스의 상황 정보는 정량적 속성값을 정성적 속성값으로 표현하여 역할 요소/객체 요소

〈표 4〉 의사 결정 속성-가치 테이블

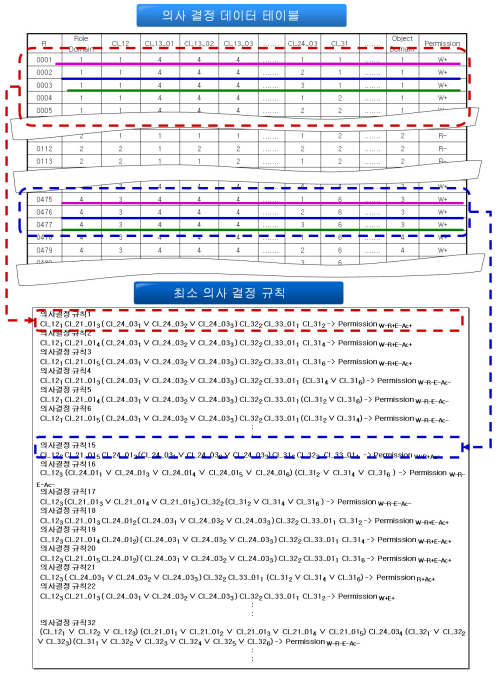
속성 \ 값	1	2	3	4	5	6	7	~	비고
CI_12	의사	담당의사	간호사	환자	담당환자	게스트			역할 상황
CI_13_01	빠름/위급	빠름/주의	빠른/정상	정상	느린/정상	느림/주의	느림/위급		신체 상황(맥박)
CI_13_02	고혈압/위급	고혈압/주의	높은/정상	정상	낮은/정상	저혈압/주의	저혈압/위급		신체 상황(혈압)
CI_13_03	높음/위급	높음/주의	높은/정상	정상	낮은/정상	낮음/주의	낮음/위급		신체 상황(혈당)
CI_13_04	높음/위급	높음/주의	높은/정상	정상	낮은/정상	낮음/주의	낮음/위급		신체 상황(체온)
CI_21_01	의원실	복도	병실 A	병실 B	병실 C	병원/외부			공간 상황(위치)
CI_22	출근시각	퇴근시각							시간 상황
CI_23_01	높음/위급	높음/주의	높은/정상	정상	낮은/정상	낮음/주의	낮음/위급		환경 상황(온도)
CI_23_02	높음/위급	높음/주의	높은/정상	정상	낮은/정상	낮음/주의	낮음/위급		환경 상황(조도)
CI_23_03	높음/위급	높음/주의	높은/정상	정상	낮은/정상	낮음/주의	낮음/위급		환경 상황(습도)
CI_24_01	의사	담당의사	간호사	환자	담당환자	게스트			활동 상황(인접인)
CI_24_02	1-2	2-3	2-4	2-5	2-1	3-2	4-2		활동 상황(이동경로)
CI_24_03	주간근무	저녁근무	야간근무	휴무					활동 상황(일정)
CI_31	A병실 환경정보	A병실 건강정보	B병실 환경정보	B병실 건강정보	C병실 환경정보	C병실 건강정보			객체명
CI_32	위치추적 정보그룹	건강정보 그룹	쾌적환경 지원그룹						소속된 객체그룹
CI_33_01	켜짐	꺼짐	없음						가용자원 (전원)
CI_33_02	영상장치	음성장치	출력장치	통신장치	가전기기				가용자원(종류)
:	:	:	:	:	:	:	:		:

〈표 5〉 역할-객체 간 접근권한 기본 정책

Component 1	Action				Component2	Context
	Write()	Read()	Execute()	AccessTo()		
의사	-	+/-	X	+/-	Temp_Lux_Humi_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	X	+/-	BloodPressure_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	X	+/-	Glycosuria_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	X	+/-	Pulse_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+	X	+	Personal_Location_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	:	:	:	:	:	:
담당의사	-	+/-	X	+/-	Temp_Lux_Humi_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	-	+/-	Light_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	-	+/-	Fan_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	-	+/-	AirCon_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	X	+/-	BloodPressure_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	X	+/-	Glycosuria_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+/-	X	+/-	Pulse_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	+/-	+/-	X	+/-	Prescription_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	-	+	X	+	Personnal_Location_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
	:	:	:	:	:	:
간호사	-	+	X	+/-	Temp_Lux_Humi_TMO	UR_Context, Env_Context, Sys_Context, UD_Context
:	:	:	:	:	:	:

와 함께 의사 결정 속성-가치 테이블을 생성한다. 생성된 의사 결정 속성-가치 테이블로부터 의사 결정 테이블을 구성하고 속성을 적용한다. 규칙 적용 과정은 응용에 적용할 보안 정책에 따라 역할 요소의 연산과 이들 간의 관계에 따르는 기능을 적용하여 분류하고 우선순위를 선정한다. 여기에서 의사결정 속성인 Permission은 이러한 조건 속성정보를 기반으로 결정된 값으로 W는 쓰기를 R은 읽기를 E는 실행을 Ac는 객체 요소에 대한 접근으로 역할 요소의 접근권한 측면에서 결정하였다. <표 4>는 u-헬스케어 응용에 적용하기 위해 작성한 의사 결정 속성-가치 테이블이다.

의사 결정 데이터 테이블을 근거로 하여 러프 집합 이론을 적용하여 규칙을 생성하기 위해서는 불필요한 조건과 조건부 속성 중 불필요한 값을 제거하여 최소 의사 결정 규칙을 찾아낸다. 이와 같이 생성된 최소 의사 결정 규칙은 기구축된 데이터베이스에 저장된 데이터의 다양한 속성 정보를 분석하여 접근제어 규칙을 간략화하고 접근제어 규칙을 생성한다. (그림 9)는 u-헬스케어 응용에 맞게 동적 접근제어 모델의 구성요소와 보안 정책을 정의하여 의사결정 속성-가치 테이블을 참조하여 접근제어 규칙을 생성하는 과정을 보인다.



(그림 9) 러프집합 이론을 이용한 접근제어 규칙 생성 과정

4. 동적 보안 서비스가 적용된 u-헬스케어 응용

본 장에서는 동적 접근제어 모델이 적용된 보안객체의 동적 보안 서비스 수행을 검증하기 위해, 분산객체그룹 프레임워크 상에 u-헬스케어 응용을 구현하여 수행 결과를 확인한다. u-헬스케어 응용은 다양한 역할에 따른 권한관리의 중요성이 요구되고, 객체 간 복잡한 상호작용을 갖는 u-병동환경에 적용하여 실시간 프로그래밍 객체 모델인 TMO 스킴을 사용하여 분산객체그룹 프레임워크 상에 구현하였다.

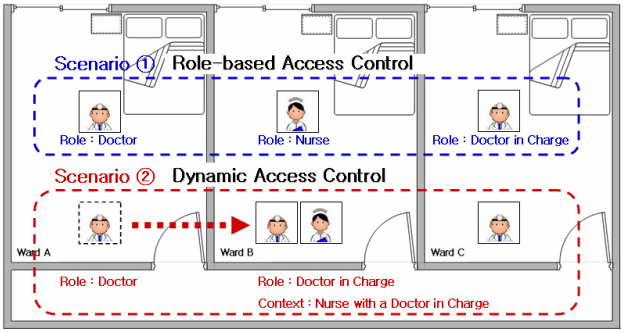
4.1 u-헬스케어 응용 수행 환경

본 절에서는 u-병원을 대상으로 응용을 구현하고, 수행환경과 동적 보안 서비스의 수행 결과를 확인하기 위한 응용 서비스 시나리오와 서비스 구성요소에 대해 기술한다.

u-병원 응용은 병동 내의 다양한 사용자에 대한 위치 추적을 통한 위치 추적 서비스와 환자의 건강 정보 수집을 기반으로 한 건강 정보 서비스 그리고 환자의 병실 환경을 최적의 상태로 유지하기 위한 쾌적 환경 지원 서비스를 제공하고 병실 내 정보가전기기들을 능동적으로 제어할 수 있다. <표 6>은 u-헬스케어 응용에서 적용되는 상황과 역할 그리고 보안 요구사항을 나타낸다.

〈표 6〉 u-헬스케어 응용의 보안 요구사항 정의

상황	역할	보안 요구사항
Office Hours	의사	같이 있는 환자의 건강 정보 열람 가능
	담당의사	같이 있는 환자의 건강 정보 열람 가능 같이 있는 환자의 건강 정보 기록 가능 위치한 곳의 환경 정보 열람 가능 위치한 곳의 정보가전기기 정보 열람 가능
	간호사	위치한 곳의 환경 정보 열람 가능 위치한 곳의 정보가전기기 정보 열람 가능 위치한 곳의 정보가전기기 제어 가능
Round Of Visits	의사	위치한 곳의 환자에 대해 담당의사의 권한 획득
	담당의사	위치한 곳의 환경 정보 열람 가능 위치한 곳의 정보가전기기 정보 열람 가능 같이 있는 환자의 건강 정보 열람 가능 같이 있는 환자의 건강 정보 기록 가능
	간호사	전체 병실의 환경 정보 열람 가능 전체 병실의 정보가전기기 정보 열람 가능 전체 병실의 정보가전기기 제어 가능 같이 있는 환자의 건강 정보 열람 가능
With Doctor In Charge	의사	같이 있는 환자의 건강 정보 기록 가능
	담당의사	같이 있는 환자의 건강 정보 기록 가능
	간호사	같이 있는 환자의 건강 정보 열람 가능
With A Patient In Charge	의사	담당의사 역할과 권한 획득
	담당의사	같이 있는 환자의 건강 정보 열람 가능 같이 있는 환자의 건강 정보 기록 가능 위치한 곳의 환경 정보 열람 가능 위치한 곳의 정보가전기기 정보 열람 가능
	간호사	해당 사항 없음
An Accident Case At Wards	의사	사고 발생 병실의 환경 정보 열람 가능 사고 발생 병실의 정보가전기기 정보 열람 가능 사고 발생 병실의 정보가전기기 제어 가능 사고 발생 병실의 환자 건강 정보 열람 가능
	담당의사	
	간호사	
An Emergency Case In Wards	의사	응급 환자의 건강 정보 열람 가능 응급 환자의 건강 정보 기록 가능
	담당의사	응급 환자의 건강 정보 열람 가능 응급 환자의 건강 정보 기록 가능
	간호사	응급 환자의 건강 정보 열람 가능

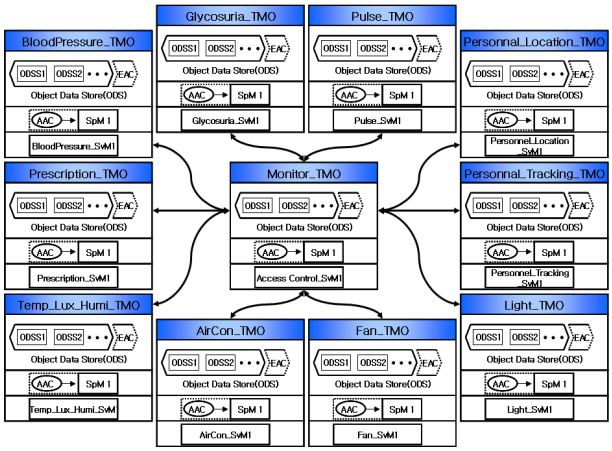


(그림 10) u-헬스케어 응용의 수행 환경

또한 응용을 위해 병원 내 물리 공간에 존재하는 센서 및 정보 가전기기들은 수행 환경 내에서 TMO 스킴으로 구현하여 능동적인 상호동작을 수행하도록 한다. u-헬스케어 응용의 물리영역은 (그림 10)과 같은 수행환경으로 도시된다.

동적 보안 서비스의 수행결과를 확인하기 위한 시나리오는 다음과 같다. 시나리오 1은 제안하는 동적 접근제어 모델이 역할 기반 접근제어를 올바르게 수행하는지 알아보기 위한 것으로, 병실 A에는 환자 <방자>와 <향단>의 담당의사인 <이몽룡>이 위치하고, 병실 B에는 환자 <향단>과 간호사 <성춘향>이, 병실 C에는 환자 <월매>와 <월매>의 담당의사인 <홍길동>이 위치하여 <담당의사>, <의사>, <간호사> 역할에 할당된 권한에 따라 적합한 서비스를 제공받는지 확인한다. 시나리오 2는 상황기반 접근제어 서비스의 수행성을 검증하기 위한 것으로, 병실 A에 위치해 있던 의사 역할에 배정된 <이몽룡>이 병실 B로 이동함에 따라 담당의사 역할에 동적으로 배정되고, 간호사 <성춘향>이 환자 <향단>의 담당의사인 <이몽룡>과 함께 있음으로써 환자에 대한 접근권한이 변경되는지 확인한다. 이를 위해 u-헬스케어 응용 서비스는 병동 내의 다양한 사용자에 대한 위치 추적을 통한 위치 추적 서비스와 환자의 건강 정보 수집을 기반으로 한 건강 정보 제공서비스 그리고 환자의 병실 환경을 최적의 상태로 유지하기 위한 쾌적 환경 지원 서비스로 구분된다. 각 서비스 그룹에 해당하는 분산객체는 TMO 객체들로 하나 이상의 서비스 수행 객체로 구성되며 분산객체그룹 프레임워크에 의해 관리된다. 수행객체들은 구축된 데이터베이스로부터 필요한 정보를 검색하여 모니터링 서비스를 통해 사용자 권한에 맞는 u-헬스케어 환경의 상황정보를 제공한다.

(그림 11)은 u-헬스케어 응용을 위한 그룹과 이들 간의 상호작용을 나타낸다. 모니터링 서비스를 위한 수행 객체는 Monitor_TMO 객체가 있으며, GUI를 통해 사용자 인터페이스를 지원한다. Monitor_TMO 객체는 사용자의 요청에 의해 u-헬스케어 응용 서비스 수행객체에게 정보를 요청하고, 수행객체로부터 제공받은 정보를 GUI를 통해 사용자에게 u-헬스케어 정보를 제공한다. 위치 추적 서비스 그룹에는 u-헬스케어 환경의 근무자와 환자와 같은 사용자의 위치를 제공하는 Personnal_Location_TMO 객체와 주기적인 위치 정보를 수집하여 사용자의 이동경로를 파악하는 Personnal_Tracking_

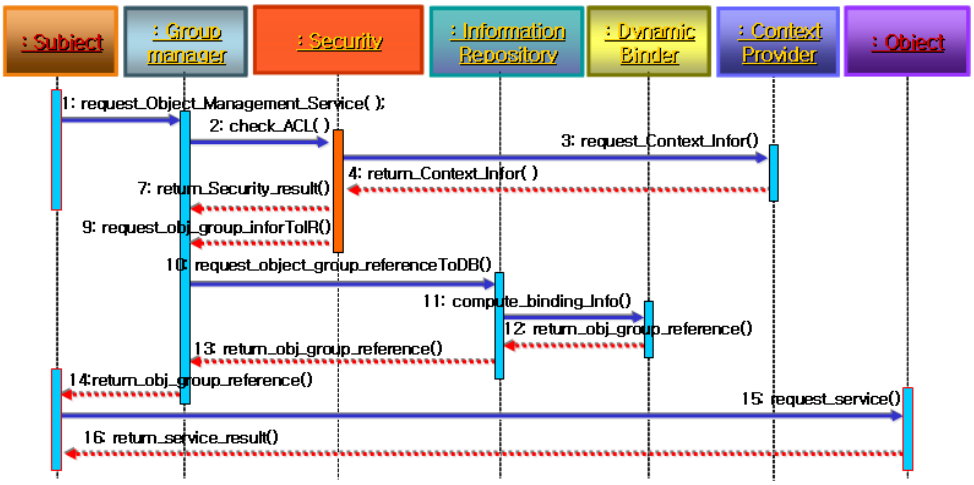


(그림 11) u-헬스케어 응용을 위한 TMO 구성요소

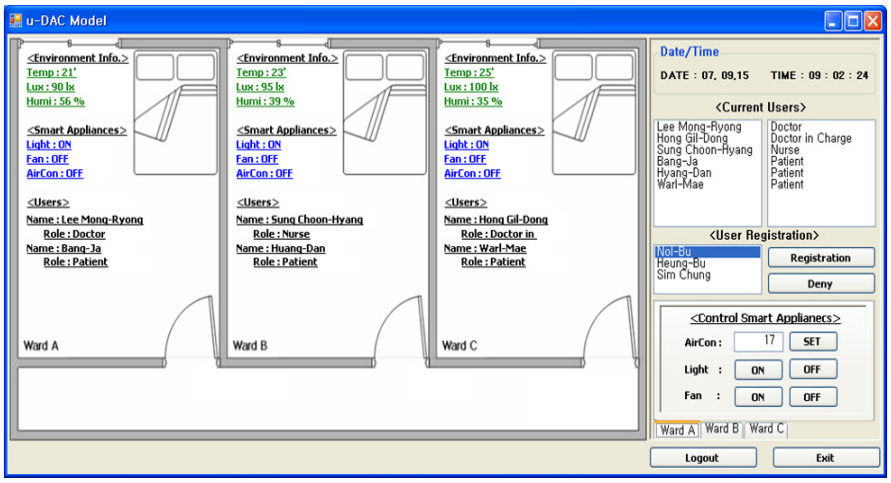
TMO 객체가 있다. 건강정보 서비스 그룹에는 환자의 건강 정보를 제공하는 객체들로 환자의 혈압(BloodPressure_TMO), 혈당(Glycosuria_TMO), 맥박(Pulse_TMO) 정보를 제공하는 객체와 진단정보(Prescription_TMO) 서비스를 제공하는 객체가 있으며, 쾌적 환경 지원 서비스 그룹에는 병실 내 설치되어 있는 전등(Light_TMO), 에어컨(AirCon_TMO), 선풍기(Fan _TMO)의 상태정보를 갖고 있으며, 각 기기를 제어하는 객체들과 병실 환경 정보를 제공하는 온도/조도/습도(Temp_Lux_Humi_TMO) 객체가 있다.

4.2 동적 보안서비스 수행 과정

사용자의 모든 접근 요청은 분산객체그룹 프레임워크의 그룹 관리자 객체를 통한다. 그룹 관리자 객체는 분산객체들의 전반적인 관리를 통해 보안 도메인을 제공하며, 클라이언트 객체와 응용서비스를 지원하는 분산객체들 간의 바인딩을 지원하기위한 인터페이스 역할을 수행한다. 그룹관리자객체는 주체 요소인 클라이언트 객체가 서비스 요청 시 제공한 주체 요소 정보와 객체 요소인 서비스 객체의 이름을 보안객체에게 전달하고, 보안객체는 상황정보 제공자 객체로부터 상황정보를 제공받은 뒤 접근제어 리스트를 갱신하고 참조하여 3.4절에서 생성한 접근제어 결정 규칙에 따라 주체요소의 접근을 제어한다. 이를 위해 보안 객체는 주체-역할 제어 리스트를 참조하여 주체 요소가 속해 있는 역할을 검색하고 역할-객체 연산 제어 리스트를 참조하여 접근제어 결정 규칙을 찾아서 주체 요소의 권한을 제어한다. 보안 객체로부터 주체 요소의 접근 권한이 성공적으로 인증되면, 그룹관리자객체는 정보저장소객체에게 서비스를 수행할 서버객체인 객체 요소의 레퍼런스를 요청한다. 서버객체가 중복되어있지 않다면 정보저장소객체는 그룹관리자객체에게 해당 서버객체의 레퍼런스를 반환하고 중복으로 등록되었을 경우는 동적바인더객체의 선정 전략에 따라 결정된 객체의 레퍼런스를 반환한다. (그림 12)는 u-헬스케어 환경에서 서비스를 제공하는 주체 요소와 자원을 관리하는 객체 요소 그리고 DOGF 컴포넌트들의 상호작용 과정을 보인다.



(그림 12) DOGF 기반으로 한 헬스케어 서비스간의 상호작용



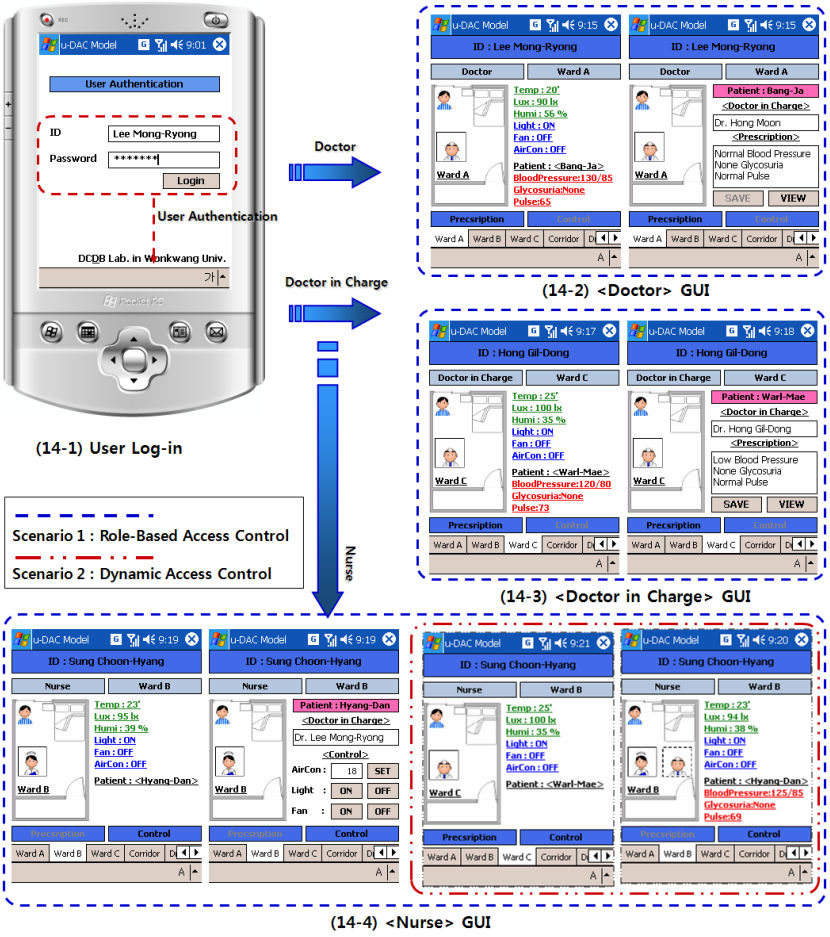
(그림 13) 서버 측 관리자 GUI

(그림 12)에서 보이는 바와 같이 주체 요소는 시스템 자원과 연관된 객체 요소와의 상호작용을 위해 그룹관리자객체의 request_Object_Management_Service()를 통해서 객체 요소의 레퍼런스를 DOGF의 그룹관리자객체에게 요청한다. 그룹관리자객체는 객체 요소에 대한 주체 요소의 접근권한 여부를 검사하기 위해 check_ACL()을 통해 보안객체에게 접근권한 검사를 요청한다. 보안객체는 접근권한을 요청받으면 주체 요소와 객체요소와 연관된 접근제어 결정 규칙을 검사하고 필요한 상황정보들을 request_Context_Infor()를 통하여 상황 정보 제공자 객체에게 요구한다. 상황정보 제공자 객체는 데이터베이스로부터 요청받은 정보를 검색하여 return_Context_Infor()를 통해 반환한다. 보안 객체는 전달받은 상황정보를 입력으로 하여 접근제어 결정규칙에 따라 접근 권한이 주어진다. 접근권한이 올바른 경우 return_Security_result()를 통해 결과를 통보하고, return_obj_group_infoToIR()을 통하여 그룹 관리자 객체에게 접근허가정보를 전달된다. 또한 그룹관리자객체는 정보저장소 객체에게 request_obj_group_referenceToDB()를 통해서 객체 요소의 레퍼런스를 요청한다. 요청한 객체 요소를 수행하는 객체가 하나만 존재할 경우 return_obj_group_reference()를 통해 해당 센서그룹의 레퍼런스를 반환한다. 그러나 요청한 객체 요소의 수행객체가 중복되어 존재할 경우 정보저장소객체는 compute_binding_Info()에 의해 동적바인더객체에게 동적바인딩서비스를 요청한다. 동적바인더객체는 적정 객체그룹 선정을 위한 알고리즘에 의해 수행 객체의 레퍼런스를 return_obj_group_reference()를 통해 반환하게 된다. 반환된 수행 객체의 레퍼런스는 그룹관리자객체의 return_obj_group_reference()를 통해 주체 요소에게 전달된다. 주체 요소는 수행객체와 상호작용을 통해서 u-헬스케어 서비스를 수행한다.

4.3 u-헬스케어 응용에서의 동적 보안 서비스 수행결과

본 절에서는 동적 접근제어 모델이 적용된 보안객체의 동적 보안 서비스 수행 능력을 검증하기 위해, 분산객체그룹 프레임워크 상에 구현한 응용을 통해 수행 결과를 확인한다.

(그림 13)은 서버 측의 관리자 GUI를 보인 것으로, 지정된 사용자가 아이디/패스워드 기반의 사용자 인증 방식을 통해 접근할 수 있으며, 사용자 관리와 상황정보 모니터링



(그림 14) 클라이언트 응용 프로그램의 GUI

서비스를 제공한다. 관리자 역할은 위치에 상관없이 u-병동 내의 모든 정보에 대해 읽기 권한을 갖고 있으며, 또한 가전 및 기기를 제어할 수 있다. 모니터링 정보는 각 병실 내에서 수집된 환경 정보, 환자의 일반적인 건강 정보 그리고 가전 및 기기의 상태 정보와 각 병실 내에 사용자들의 위치 정보를 포함한다.

(그림 14-1)는 클라이언트 응용 프로그램의 GUI로 사용자 인증과정을 통해 사용자-역할 상황과 환경 상황 정보를 기준으로 사용자-역할 할당 연산을 수행한다. 그 결과 (그림 14-2)의 '의사' 역할 GUI, (그림 14-3)의 '담당의사' 역할 GUI, (그림 14-4)의 '간호사' 역할 GUI를 보이며, 각 역할에 맞는 접근제어를 통해 서로 다른 수행 결과를 보이고 있다. 14-2는 '이몽룡'이 의사 역할에 배정되어 현재 위치해 있는 병실의 환경정보와 환자 '방자'의 건강정보를 받은 결과이며, '방자'의 담당의사가 아니므로 진료 기록에 대한 쓰기 권한이 없음을 보인다. 이와 달리 14-3은 '홍길동'이 의사역할과 담당의사 역할에 배정되어 환자 '월매'의 진료기록에 대한 쓰기 권한이 인증되어 진료기록을 저장할 수 있다. 14-4는 '성춘향'이 간호사 역할에 배정되어 모든 병실의 환경 정보 제공 서비스와 쾌적 환경 지원 서비스에 대한 읽기 권한은 있으나 현재 위치해 있는 병실 B에 대해서만 쾌적 환경 지원 서비스에 대해 쓰기와 실행 권한이 있음을 보인다. 시나리오 2의 수행결과 '이몽룡'이 병실 A에서 병실 B로 이동함에 따라 '이몽룡'과 '성춘향'의 상황정보가 변경되었다. 이에 따라 병실 B에서 '이몽룡'은 담당의사 역할에 배정되어 담당 환자인 '월매'의 진료기록을 작성할 수 있고, '성춘향'은 담당의사와 함께 있는 상황에 따라 환자의 건강정보를 열람할 수 있는 권한을 획득하여 '월매'의 건강 정보를 읽을 수 있어 GUI상에 건강 정보가 나타난다.

수행 결과 의사, 담당의사, 간호사 역할과 상황에 따라 정보에 대한 접근 제어가 올바르게 수행함을 확인 하였다. 또한 시나리오 2에 따라 상황기반의 동적 보안 서비스를 제공하기 위해 상황 정보를 기준으로 동적 역할 배정과 상황에 따른 객체 요소에 대한 접근권한이 변경됨을 수행결과를 통해 확인 하였다.

5. 결론 및 향후연구

본 논문에서는 유비쿼터스 환경을 지원하는 분산객체그룹 프레임워크의 보안 서비스에 대해 기술하였다. 특히 u-헬스

케어 환경의 보안 요구사항을 충족시키기 위해 동적으로 재구성되는 보안 도메인에 따라 시간과 공간 그리고 사용자의 상황에 따라 재구성되는 보안 도메인을 관리하기 위해 분산 객체그룹 프레임워크를 사용하고, 강건하면서도 유연한 보안 서비스를 제공하기 위해 보안객체에 동적 접근제어 모델을 적용하였다. 보안 객체는 동적 접근제어 모델을 기반으로 유비쿼터스 환경의 주체 요소에 대한 서비스 요청 정보를 그룹관리자객체로부터 전달받고 상황정보 제공자 객체로부터 상황정보에 따라 접근권한을 주어 동적 보안 서비스를 제공한다.

동적 접근제어 모델은 주체 요소, 역할 요소, 객체 요소 외에 상황 정보 요소를 두어 기존의 역할 기반 접근제어 메커니즘에 동적 특성을 부여하였다. 이에 주체 요소의 역할 할당과, 역할 간 권한 상속, 역할과 객체의 접근모드 배정은 상황 정보를 기준으로 접근제어 결정 규칙에 따라 동적으로 해당 연산을 통해 이루어지며, 연산의 결과는 접근제어 리스트에 반영된다. 접근제어 결정 규칙은 러프집합 이론을 적용하여 상황정보를 입력값으로 최소 의사 결정테이블을 생성하고 이를 기반으로 생성하였다.

상황기반의 동적 접근제어 모델은 자원에 접근할 수 있는 권한을 역할에 배정하며, 사용자를 그 역할의 구성원에 소속되도록 함으로써 상황에 따라 접근 권한을 동적으로 제어하여, 악의적인 클라이언트로부터 유비쿼터스 시스템 자원을 보호한다. 동적 보안 서비스의 수행결과를 확인하기 위해 유비쿼터스 응용 가운데 u-병원 응용을 분산객체그룹 프레임워크 상에 구현하고, 역할에 따라 강건한 접근권한 제어와 상황 정보에 기반을 둔 동적 보안 서비스의 수행 능력을 검증하였다.

향후 다양한 유비쿼터스 컴퓨팅 환경에 적용하기 위해 표준화된 상황정보 연구를 진행하고 필드테스트를 통해 유비쿼터스 환경 내의 다양한 지능형 장치와 능동적인 객체 간의 보안 서비스를 제공하는 규칙 기반의 동적 보안에 관한 추가적인 연구를 진행할 예정이다.

참 고 문 헌

[1] Martin Strassner and Thomas Schoch, “Today’s Impact of Ubiquitous Computing on Business Process,” Pervasive 2003 short paper proceedings, pp.62-74, Zurich, May, 2002.

[2] D. F. Ferraiolo, J. A. Cugini, D. Richard Kunj, “Role-Based Access Control(RABC) : Features and Motivations,” Proceedings of the 11th Annual Computer Security Applications Conferences, pp.241-248, 1995.

[3] R. S. Sandhu and E. J. Coyne, “Role-Based Access Control Models,” IEEE Computer, 20(2), pp.38-47, 1996.

[4] R. S. Sandhu, D. Ferrraiolo and R. Kuhn, “The NIST Model for Role-Based Access Control: Towards A Unified Model Approach,” ACM Workshop on Role-Based Access Control, pp.47-63, 2000.

[5] Chang-Sun Shin, Myoung-Suk Kang, Chang-Won Jeong and Su-Chong Joo, “TMO-Based Object Group Framework for Supporting Distributed Object Management and Real-Time Services,” Lecture Notes in Computer Science, Vol.2834, pp.525-535, 2003. 9.

[6] Chang-Sun Shin, Chung-Sub Lee and Su-Chong Joo, “Healthcare Home service System Based on Distributed Object Group Framework,” Lecture Notes in Computer Science, Vol.3983, pp.798 - 807, 8-11, May, 2006.

[7] 김동호, 정창원, 주수종, “분산 객체그룹 프레임워크의 보안 서비스를 지원하는 보안 정책 관리”, 한국인터넷정보학회 학술지, 제8권 2호, pp.149-152, 2007.10.02-03.

[8] 정창원, 김동호, 김명희, 주수종, “u-헬스케어 지원 분산 프레임워크에서 접근 제어 모델을 이용한 동적 보안 서비스”, 한국인터넷정보학회 논문지, 제8권 6호, pp.29-42, 2007.12.

[9] J. F. Barkley, K. Beznosov and J. Uppal, “Supporting Relationships in Access Control Using Role Based Access Control,” pp.55-65, RBAC ’99. Proceedings of the 4th ACM workshop on Role based access control, 1999.

[10] M.J.Moyer and M. Ahamad, “Generalized Role-Based Access Control,” IEEE International Conference on Distributed Computing Systems(ICDCS2001), pp.391-398, 2001.

[11] Gustaf Neumann and Mark Strembeck, “An Approach to Engineer and Enforce Context Constraints in an RBAC Environment,” In 8th ACM Symposium on Access Control Models and Technologies (SACMAT2003), pp.65-79, Como, Italy, June, 2003.

[12] MIT Oxygen Project, http://oxygen.lcs.mit.edu/

[13] UC Berkeley Smart Dust Project, http://www-bsac.eecs.berkeley.edu/archive/users/ warneke-brett/SmartDust/index.html.

[14] CMU : Aura Project : Pervasive invisible computing, http://www-2.cs.cmu.edu/~aura/

[15] 이충섭, 정창원, 주수종, “헬스케어 홈 서비스를 위한 데이터베이스 및 응용 서비스 구현”, 한국인터넷정보학회 논문지 제 8권 1호, pp.57-70, FEB., 2007.

[16] Pawlak, Z., “Rough Set Approach to Knowledge-Based Decision Support,” ICSWUT Reports on Rough Set, March, 1995.

[17] Le Hoai Bac and Nguyen Anh Tuan, “Using Rough Set in Feature Selection and Reduction in Face Recognition Problem,” PAKDD 2005, LNAI 3518, pp.226-233, 2005.

정 창 원

e-mail : mediblue@wku.ac.kr
1993년 원광대학교 컴퓨터공학과(학사)
1998년 원광대학교 전자계산교육과(석사)
2003년 원광대학교 컴퓨터공학과(공학박사)
2004년~2006년 전북대 학술연구교수
2006년~현 재 원광대학교 전기전자 및 정보공학부 박사후 연구원
관심분야 : 분산객체 컴퓨팅, 유비쿼터스 컴퓨팅, 멀티미디어 서비스, LBS

김 동 호

e-mail : bit4me@nate.com
2005년 원광대학교 전기전자 및 정보공학부 (학사)
2008년 원광대학교 컴퓨터공학과(석사)
현 재 (주)PC 닥터 부설 연구소 연구원
관심분야 : 분산객체 컴퓨팅, 객체지향 프로그램, 유비쿼터스 컴퓨팅

주 수 종

e-mail : scjoo@wku.ac.kr
1986년 원광대학교 전자계산공학과(학사)
1988년 중앙대학교 컴퓨터공학과(공학석사)
1992년 중앙대학교 컴퓨터공학과(공학박사)
1993년 미국 Univ. of Massachusetts at Amherst, Post-Doc.
2003년 미국 Univ, of California at Irvine, Visiting Professor.
1990년~현 재 원광대학교 전기전자 및 정보 공학부 교수
2007년~현 재 원광대학교 정보전산원 원장
관심분야 : 분산 실시간 컴퓨팅, 분산객체모델, 시스템 최적화, 멀티미디어 데이터베이스