DOI: 10.3745/KIPSTD.2009.16-D.4.561

모바일 서비스관리를 위한 정책정의언어

안 성 욱[†] · 류 성 열[††]

요 약

모바일 환경에서 유지보수의 효율적인 관리를 위해서는 서비스를 정책으로 관리하기 위한 시스템 구조와 정책정의언어가 필요하다. 본 연구는 IETF의 정책 프레임웍 중에 정책 실행자인 PEP 구조를 정의하고, PEP 구조하에서 수행될 수 있는 정책정의언어를 제안한다. 제안한 정책정의언어는 문헌자료와 모바일 특성을 기반으로 하여 요구사항을 도출하고, 3단계 접근 방법으로 정책정보모델을 설계하여 정책정의언어로 정의하였다. 3단계 접근 방법은 정책이 적용될 범위를 결정하는 정책도메인, 정책 적용 및 제어하는 종류를 구분하는 정책규칙, 정책 구조를 문맥화하는 정책문법으로 구성된다. 제안한 정책정의언어의 효율성을 검증하기 위하여 시나리오를 정책정의언어로 정의하여 정책 도구를 이용하여 검증하였고, 타 정책정의언어들과 비교 분석하여 확장성을 입증하였다.

키워드 : 정책, 정책 정보 모델, 정책 정의 언어, 정책 프레임웍

Policy Definition Language for Service Management in Mobile Environment

Sung Wook Ahn[†] · Yul Sung Rhew[††]

ABSTRACT

In order to manage repair and maintenance efficiently in the mobile environment, the system structure to manage service as a policy and the policy description language are needed. This research defined the structure of PEP, which is the executioner of policy in the IETF policy framework, and proposed the policy description language which can be carried out under the PEP structure. The proposed policy description language derived demand matters based on documentary data and the characteristics of mobile and the policy information model was designed with the three stage approaches and was defined as policy description language. The three stage approaches are made up of the policy domain that decides the scope to which the policy applies, the policy rules which distinguish the kinds of policy application and control, and policy grammar which contextualizes the policy structure. In order to verify the efficiency of the policy description language, scenarios are defined with the policy description language and verified it by using policy tool and proved the expansive nature by comparing and analyzing other policy description language.

Keywords : Policy, Policy Information Model, Policy Definition Language, Policy Framework

1. 서 론

최근의 망 개방 및 개방형 플랫폼으로의 전환에 따라 정책기반의 서비스 관리는 기회의 강화 및 위협의 해소라는 측면에서 동시에 작용하며, 환경의 급속한 변화에 따른 유연한 사업전략을 지원하기 위하여 반드시 필요하다. 다중 채널 콘텐트 관리는 콘텐츠 저작권의 보호, 비즈니스 정책의 배포를 통한 비즈니스 민첩성을 확보해야 한다. 또한, 무선 네트워크를 우회하는 콘텐츠의 대체 유통 채널로 인한 비즈니스 위협에 대응 가능한 콘텐츠의 보호 및 신규 채널 콘텐츠 유통 전략지원 가능한 정책기반의 관리 프레임웍을 구축해야 하다. 콘텐츠의 안전성 보장 및 해당 콘텐츠의 유통을 제어하며, 사업 환경에 따른 업무 정책변경 시 이의 적절한 적용을 통해 정책 변경에 따른 영향을 최소화하고자 한다. 이를 위해서 정책기반의 관리 프레임웍을 제공하기 위해서 우선적으로 정책 정보 모델을 통해서 정책정의언어를 결정하여야 한다.

기존 정책정의언어는 호스트, 네트워크 보안, 어플리케이션 등을 제어하는 정책 정의들이 존재하였다[5][6][7][8]. 이들 연구를 분석한 결과 기존 정책정의언어는 특정 도메인에 의존적인 구조이기 때문에 다양한 도메인에 적용할 수 없거나 허가, 금지 규칙에만 한정적으로 정책을 정의하고 있다. 본 연구에서는 IETF(Internet Engineering Task Force)에서

† 정 회 원 : 숭실대학교 컴퓨터학과 박사과정
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2009년 4월 1일
수 정 일 : 1차 2009년 5월 22일
심사완료 : 2009년 5월 22일

제시하는 PCIM(Policy Core Information Model, 정책중심정보모델)을 기반[1][2]하여 다양한 정책 규칙을 수용할 수 있는 정책정보모델(Policy Information Model, PIM)을 설계하여 다양한 도메인과 여러 정책규칙을 지원하는 정책정의언어를 정의한다. 제안한 정책정의언어는 3단계 접근 방법인 정책이 적용될 범위를 결정하는 정책도메인, 정책 적용 및 제어하는 종류를 구분하는 정책규칙, 정책 구조를 문맥화하는 정책 문법으로 설계한다. 또한, 정책정의언어의 효율성과 확장성을 검증하기 위해서 시나리오를 정책정의언어로 기술하고, 이를 정책 도구로 검증하며 타 정책정의언어와 비교한다.

2. 관련 연구

정책정의언어는 정책정보모델, 정책관리를 위한 프레임웍, 정책이 수행되는 서비스 환경을 모두 고려하기 위해서 관련 문헌들을 중심으로 수행한 연구 결과를 요약한다.

2.1 IETF의 PCIM 모델

IETF PCIM에서 정책은 정책규칙(PolicyRule) 들의 집합을 사용하여 적용되고, 각각의 정책규칙은 정책조건(Policy-Condition)의 집합과 정책행위(Policy-Action)의 집합으로 구성된다. 여러 정책규칙은 정책그룹(PolicyGroup)과 결합된다. 조건(condition)에 대응하여 수행되는 행위(action)를 지정하는 규칙(Rule)로 정의되며, 규칙은 조건이 참일 때 수행되는 행위의 집합으로 정의된다[1][2]. 이 연구에서는 특정 시스템에 적용할 때에는 PCIM 정책모델의 핵심을 기반으로 확장하도록 제시하고 있으나, 특정 도메인에 대한 정책정보모델 제시는 없어, 본 논문에서는 모바일 환경의 정책정보모델 뿐만 아니라, 다른 도메인으로 확장할 수 있는 좀더 구체적인 정책정보모델을 제시하여 정책정의언어에 반영한다.

2.2 타 정책정의언어

PPL(Path-based Policy Language)는 네트워크 환경에서 트래픽 경로를 지정하여 트래픽에 대한 제어를 표현하는 언어이다[5]. 특정 대상에 대한 트래픽 경로, 조건 및 행동을 명시하여 다양하고 통합된 네트워크관련 보안정책을 표현한다. PPL은 네트워크 도메인을 위주로 설계된 정책 언어이므로 허가와 금지 정책만을 지원하며, 복합적인 보안정책의 표현을 지원하지 않는다.

Policy Description Language(PDL)은 Bell-Lab에서 개발한 이벤트 방식의 보안정책 명세 언어이다[6]. 발생한 이벤트에 대해 그에 적합한 행위로 바꾸기 위한 함수로 정책을 정의하는 이벤트 방식인 의무 정책만을 지원하며, 접근 제어 정책을 지원하지 않는다.

LaSCO(Language for Security Constraints on Ojects)는 시스템의 특정 상황에서 만족하여야 하는 제약 사항(constraints)을 기술하는 그래픽 언어이다[7]. LaSCO는 시각적인 모델링 방법으로 쉽게 이해할 수 있다는 장점은 있지만 제약 사항 중심으로 표현이 이루어지므로 의무, 위임, 박탈 규칙을 지원하지 않는다.

ASL(Authorization Specification Language)는 어플리케이션 의존적인 접근 제어를 표현하는 언어이다[8]. ASL은 역할기반의 접근제어만을 지원하지 재사용성을 고려한 그룹핑을 제공하지 않으며 의무, 위임, 박탈 규칙을 지원하지 않는다.

2.3 IETF의 Policy Framework 구조

IETF 정책 프레임웍(Policy Framework)은 4가지 구성요소로 구성된다. 정책을 작성하는 PMT(Policy Management Tool, 정책관리도구), 작성한 정책을 저장하는 PR(Policy Repository, 정책저장소), 저장소에서 정책을 참조하여 정책을 검사하여 전송하는 PDP(Policy Decision Point, 정책결정자), 그리고 받은 정책을 실행하고 정책결정자로 정책 버전을 체크하는 PEP(Policy Enforcement Point, 정책실행자)로 구성된다[2]. 본 연구에서는 IETF에서 제시하는 정책 프레임웍 중에서 PEP 구조를 정의하여 PEP 기반 하에 정책이 수행되는 정책정의언어를 정의한다.

2.4 IETF의 Policy Framework 구조

OMA(Open Mobile Alliance)의 OSE(OMA Service Environment)는 서비스의 신속한 설치, 트래킹, 실행자(enforcer), 구동자(enabler)들과의 관계 등의 서비스 플랫폼을 제시하고 있다[3][4]. 이 연구에서는 단말의 서비스 환경을 제시하고 있으며, 본 논문에서는 제안한 PEP 구조를 OSE 기반으로 설계하여 실행자와 구동자간 정책 처리를 위한 정책정의언어를 제시한다.

3. 정책정의언어의 환경

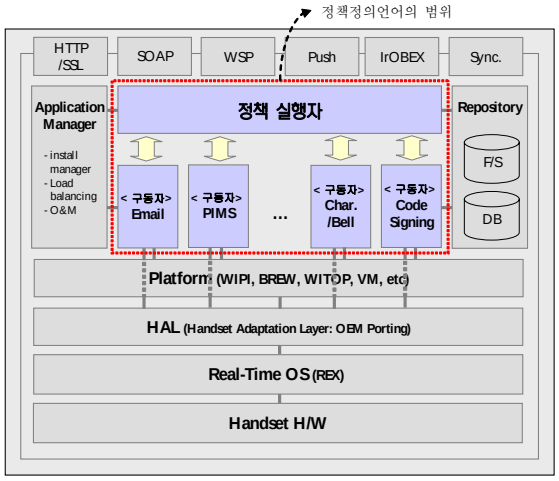
IETF 프레임웍 중에 PEP 구조로 정책을 실행하는 단말의 PEP 구조와 정책정의언어의 요구사항을 도출한다. 정책정의언어는 정책 시나리오를 표현하기 위해서는 정책이 수행되는 단말의 PEP 구조를 명확히 정의하고, 언어적인 특성들을 도출하여야 한다. 이를 기반으로 4장의 정책정의언어 설계에 반영한다.

3.1 단말의 PEP 구조

정책정의언어 설계를 위한 모바일 환경에서의 PEP 구조를 제시하여 정책수행자의 아키텍처로 수행자와 구동자간의 관계를 설명한다. (그림 1)은 단말의 PEP 구조로 정책 실행자와 구동자간의 관계를 보여주며 이 구조가 본 논문에서 제시한 정책정의언어의 범위이다.

3.2 요구사항

정책기반 언어는 현재 접근 제어(Access Control) 및 규칙 기반의 연구성과에 따라 정책정의언어가 나오게 되었지만[5][7][8], 정책에 대한 일반적인 요구사항은 정책의 개념



(그림 1) 단말의 PEP 구조

적인 정의에 따라서 조건과 실행을 만족하는 PCIM에 기반하고 있으며, 이에 문헌 자료와 모바일 환경의 특성을 고려하여 확장 요구사항을 정의한다.

R1. Policy = Condition + Action: 정책은 조건과 실행으로 구성하며, 정책은 그룹화되어야 한다[2].

정책의 기본적인 요구사항으로서 PCIM의 정책 그룹의 개념과는 PEP로 전달하는 부분에서 차이를 보이고 있다. 제안한 정책정의언어에서는 정책이 무선 단말기에 탑재되는 것을 기본 조건으로 하고 있기 때문에 정책 그룹화는 정책의 조합을 넘어서 정책의 변경 시에 최소 단위 변경을 요구한다. 정책 그룹화와 재사용에 대한 요구는 정책을 그룹화하고, 규칙 셋을 컴포넌트로 구성하여 디바이스와 네트워크의 배포 상황에 따라서 정책과 규칙 컴포넌트를 재사용하게 하는 것으로 정의한다.

R2. 분류화(Categorization): 정책의 실행 타입에 따라 분류되어야 하고 카테고리의 추가를 고려하여 구성되어야 한다.

정책은 매우 다양한 모습으로 정의하고 활용할 수 있기 때문에 도메인 내부에서도 정책의 실행 관점으로부터 접근 방법과 그에 따른 분류화가 필요하다. 이런 실행 기반의 분류 기법은 크게 정책의 접근 제어와 이벤트 발생시 적용되는 정책으로 구분되며, 정책을 직관적인 관점에서 정의하고 적용하는데 유용하다.

R3. 파라미터화(Parameterization): 정책의 action를 세밀하게 제어하고 상황에 따라서 변경된 내용을 적용하기 위해서 파라미터되어야 한다.

분산환경에서는 다양한 도메인에서의 정책은 많은 PEP에 적용되는 인프라를 가정하고 있다[10][11]. 따라서 정책이 변경되는 경우 정책에 대한 변경을 최소화할 수 있는 방향으로 정책 정의가 되어야 한다. 이에 정책정의언어는 파라미터에 따라 정책의 반응이 변경 될 수 있는 구조가 요구되고, 파라미터화의 적용 가능한 범위는 네트워크 상태 또는 정책 구동자의 성능 등 여러 가지 요인을 복합적으로 고려하여야 한다. 특히, 모바일 환경에서는 디바이스 확장성을 높이는 부분에도 응용 가능한 부분이다.

R4. 제약조건(Constraint): 정책 자체에 제약 조건을 부여하거나 에러 발생시 대처할 수 있는 정책 제약 조건을 제공하여야 한다[9][10].

PCIM 기반의 정책은 단순한 조건만으로 구성이 되어있어 세밀한 상황 제어 또는 충돌 처리 부분에 있어 구체적인 내용을 담고 있지는 않다. 하지만, 제안한 정책정의언어는 엔터프라이즈 비즈니스의 서비스 환경을 고려하여야 하고 이러한 부분을 정책의 조건에 같이 포함하여 기술하는 것보다는 정책의 제약조건을 분리하여 정의하게 되면, 정책의 제약조건의 독립성이 보장되면서 정책 자체의 개념을 좀더 명확하게 표현할 수 있다.

R5. Metadata: Policy의 속성에 대한 추가와 정보를 담고 활용하기 위한 Meta Model 이 필요하다.

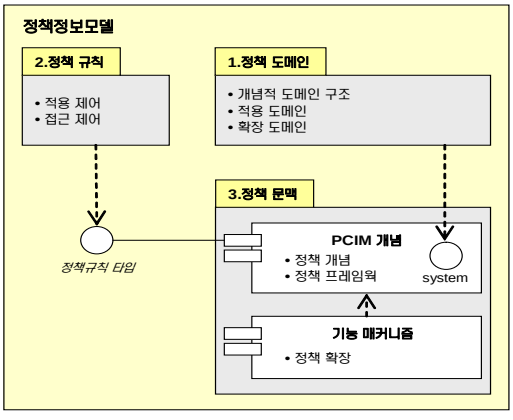
정책을 기술하는 자원들인 정책 수행 주체인 Subject, 정책 적용 대상인 Target, 정책관련 속성인 Resource 등에 대한 정보 데이터 관계를 정책 메타로 정의하여 기술한다.

R6. 위임(Delegation): 분산되어 있는 정책 서버에 정책의 소유자에 대한 속성을 부여하고, 위임을 통해서 정책을 분산관리하고 부여된 정책을 박탈할 수 있어야 한다[7][8].

정책 프레임웍에서 PDP는 정책관리 서버 또는 위임된 정책프레임웍 서버(DRM서버, Code Signing서버, Sync서버 등)이 될 수 있다. 분산환경 구조에서는 정책은 정책관리 서버에서 작성된 정책을 제외한 모든 정책은 위임되어서 관리되어야 한다. 또한, 분산되어 있는 정책의 관계에 정책의 소유자에 대한 속성을 부여하고 관리하기 위해서는 디바이스 간의 정책 위임을 만족해야 한다.

4. 정책정의언어 설계

정책정의언어는 정책정보모델을 정의하여 그를 기반으로 정책정의언어로 표현하여야 한다. 정책정보모델은 3장의 정책시나리오, 단말의 PEP 구조, 정책정의언어의 요구사항을 입력으로 해서 설계한다. 정책정보모델은 PCIM를 기본 구조로 하여 3단계 방법으로 설계한다. 정책을 규정하는데 필요한 범위를 정의하는 정책도메인 (Policy Domain, PD), 사용자 요구사항에 부합 하도록 정책의 종류를 구분하는 정책규칙(Policy Rule, PR), 정책 구조를 문맥화하는 정책문법(Policy Context, PC) 단계이다. (그림 2)는 정책정보모델 구



(그림 2) 정책정보모델의 구성요소간의 관계

성요소간의 관계를 보여주며, 정책도메인은 정책문맥의 시스템으로 매핑하고, 정책규칙은 정책문맥의 정책규칙 타입으로 매핑하여 정책정보모델을 설계하고 그를 정책정의언어로 정의한다.

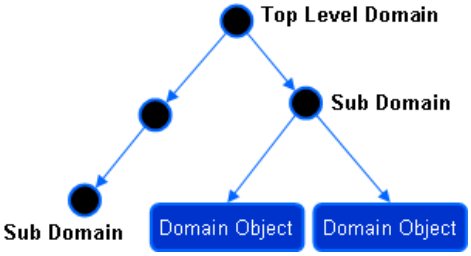
4.1 정책 도메인

정책 도메인은 정책을 생성하고 관리하기 위해 특정 관점에서 관리 대상에 대한 영역을 구분시킨 개념이다[11][12]. 정책 도메인은 기본적으로 계층 구조 형태로 구성되며, 도메인 간의 연관 관계를 고려한 개념에서 적용하고자 하는 관점에 따라 적용 도메인으로 확장 시킬 수 있다. 정책 도메인에 적용 대상으로는 정책 생성시 사용되는 배포 도메인(Deploy Domain)과 오퍼레이션 도메인(Operation Domain)이 있으며, 시스템 관리 목적에 따라 정책 도메인(Policy Domain)과 정책 재사용을 위한 템플릿 도메인으로 확장시킬 수 있다.

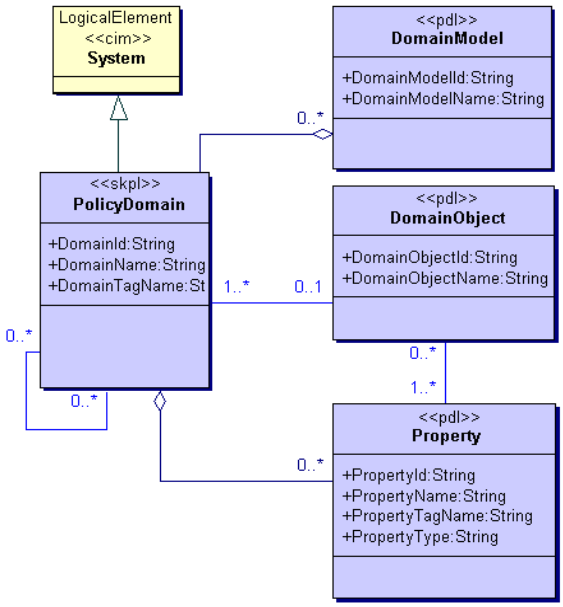
개념 도메인은 추상적인 도메인을 기반으로 구조화된 도메인을 정의할 수 있다. (그림 3)은 Top Level Domain, Sub Domain, Domain Object 의 연관 관계에 대한 상위 구조 형태의 구조를 가지는 추상적인 모델이다. 이를 통해 구조화된 모든 도메인 영역에 적용될 수 있는 클래스들을 정보모델로 정의한다.

(그림 4)는 CIM 의 System 클래스를 기반으로 도메인은 PolicyDomain, DomainModel, DomainObject, Property 클래스로 구성되며, 이를 기반으로 도메인관련 클래스들이 확장되는 관계를 보여준다. 이는 특정 도메인 오브젝트에 대하여 다수의 도메인 형태로 표현 할 수 있으며, 도메인과 도메인, 도메인과 도메인 오브젝트에 대한 복잡한 구조의 연관 관계를 구조화 할 수 있다.

적용 도메인은 개념 도메인의 추상적인 구조를 실제 도메인으로 구체화한 것을 의미한다. 도메인을 구체화하기 위해서는 특정 관점에 대한 접근 방법이 필요하다. 적용 도메인은 배포 도메인과 오퍼레이션 도메인으로 구성한다. 배포 도메인은 정책을 작성하여 정책이 수행되는 기기에 해당되며, 디바이스 속성(디바이스타입, 모델, 제조사 등)별로 관리하며 배포 시 사용된다. 오퍼레이션 도메인은 배포한 기기 내에서 동작하는 수행자(Enforcer)와 구동자의 오퍼레이션이 여기에 해당된다. (그림 5)는 배포도메인에 WIPI 운영체제를 사용하는 단말 모델들과, 오퍼레이션 도메인에 WIPI 운영체제의 각 애플리케이션내의 오퍼레이션들을 보여준다.



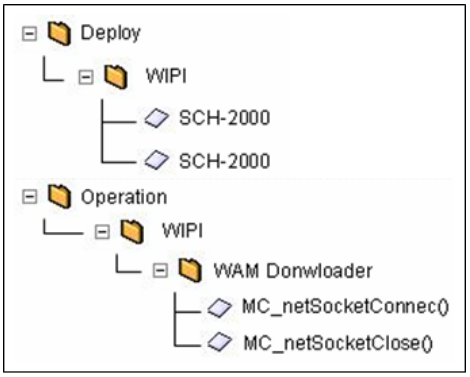
(그림 3) 추상적인 도메인 구조



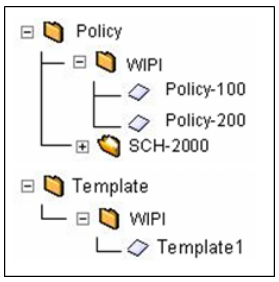
(그림 4) 도메인의 클래스다이어그램

오퍼레이션 도메인은 정책정의언어 작성시 수행자와 구동자의 오퍼레이션이 알맞게 정의되었는지를 컴파일 시에 의미분석(Semantics Analysis) 단계에서 체크하게 된다.

확장 도메인은 적용 도메인 이외의 좀더 확장된 개념의 도메인을 의미한다. 이는 시스템 관리 목적에 따라 확장되어 정의될 수 있으며, 본 논문에서는 정책 구조 도메인(Policy Structure Domain)과 템플릿 도메인(Template Domain)으로 구성한다. 정책 구조 도메인은 정책을 기술하기 위한 구성 요소들에 대한 도메인으로서 규칙, 행위, 이벤트 등에 대하여 정책관점에서 도메인화 하는 것을 의미한다. 또한 정책을 배포관점, 조직 및 사용자 관점, 시나리오 관점으로 도메인화 함으로써 정책에 대한 관리를 쉽게 할 수 있다. 템플릿 도메인은 정책 작성시 기본 구조 및 일반화된 정책을 템플릿화하여 사용하는 것으로 사용자의 편이성을 위해 유용하다. (그림 6)은 정책 구조 도메인에 WIPI 운영체제와 SCH-2000 단말 모델에 적용된 정책들과, 템플릿 도메인에 WIPI 운영체제에 사용되는 Template1의 템플릿들을 보여준다.



(그림 5) 적용도메인 구조



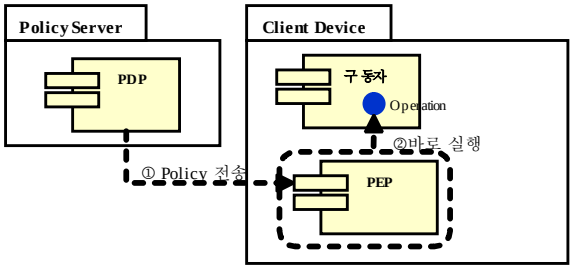
(그림 6) 확장도메인 구조

4.2 정책 규칙

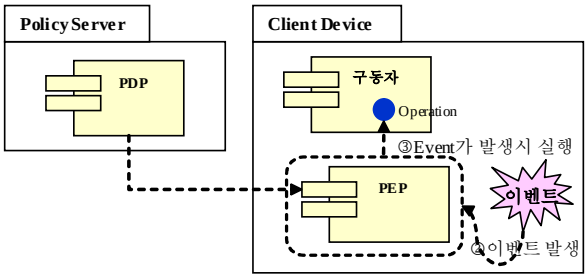
정책 규칙은 정책을 생성할 때 정책 적용 시나리오 별 정책 규칙의 종류를 제공함으로써 정책의 구분을 명확히 하기 위한 개념이다. 시나리오를 기반으로 정책 규칙 모델은 크게 적용 제어(Apply Control), 접근 제어(Access Control) 타입으로 나눌 수 있으며, 정책규칙 타입은 확장되어 추가 가능한 구조이다. 적용 제어는 적용된 정책을 바로 실행할 것인지 또는 특정 이벤트를 기반으로 하여 실행할 것 인지에 따라서 2가지로 나누어지며, 접근 제어는 권한을 주거나 제한하거나 또는 위임을 하거나 위임을 박탈할 것인지에 따라서 4가지로 구분된다.

수행 규칙은 적용제어 방식으로 정책의 대상인 클라이언트 디바이스가 정책을 받았을 경우, 이벤트의 발생이 없이도 정책을 받은 경우 정책의 조건을 체크하여 만족하는 경우 바로 실행을 하게 된다. (그림 7)은 정책관리 프레임웍 구조에서 PDP 에서 결정된 정책이 PEP 로 전송되면 PEP 가 구동자의 오퍼레이션을 호출하여 바로 수행하도록 하는 구조이다. 이는 즉시성을 갖는 기능으로서 서비스 장애, 구성파일 업데이트 등의 경우에 적용할 수 있는 서비스 적용 제어 규칙이다. 정책정의언어에서는 exec 키워드를 사용하여 표현한다.

의무 규칙은 적용제어 방식으로 정책의 대상인 클라이언트 디바이스에 이벤트가 발생했을 때 적용되며 정책의 조건을 만족하는 경우에 실행하게 된다. (그림 8)은 PDP 에서 결정된 정책이 PEP 로 전송되면 PEP 가 이벤트가 발생했을 때 구동자의 오퍼레이션을 수행하도록 하는 구조이다. 이는 단말에서 동작하는 시간, 통화버튼, 전화 걸기 등의 이벤트가 발생하는 경우 서비스를 제어하는데 효과적인 서비스 적용 제어 규칙이다. 정책정의언어에서는 oblg 키워드를 사용하여 표현한다.



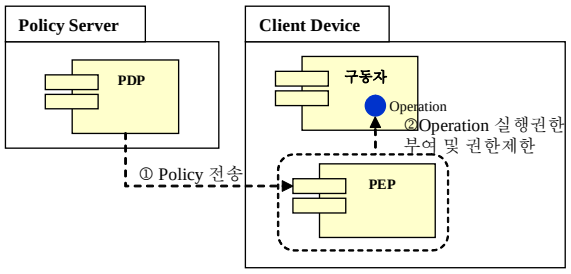
(그림 7) 수행 규칙 프로세스



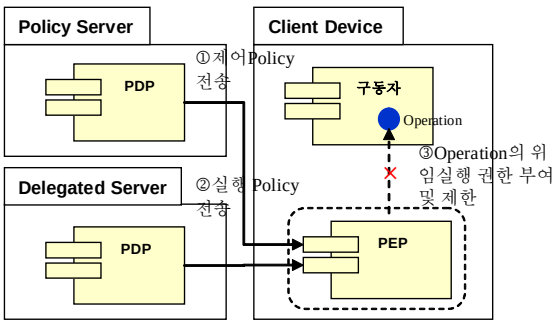
(그림 8) 의무 규칙 프로세스

인가 규칙 및 금지 규칙은 접근제어 방식으로 허가 규칙은 실행 권한을 허가하고자 할 때 정의하고, 금지 규칙은 실행 권한을 금지하고자 할 때 정의한다. (그림 9)는 PDP 에서 결정된 정책이 PEP 로 전송하여 정책을 해석하여 구동자의 오퍼레이션을 인가 또는 금지하는 구조이다. 이는 서비스 장애 시나 특정 기간까지 구동자의 오퍼레이션을 금지하거나 금지된 오퍼레이션을 다시 인가하는데 적용하는 서비스 접근 제어 규칙이다. 정책정의언어에서는 인가 규칙인 경우 perm 키워드를, 금지 규칙인 경우에는 proh 키워드를 사용하여 표현한다.

위임 규칙 및 박탈 규칙은 접근제어 방식으로 위임 규칙은 실행 권한을 위임하고자 할 때 정의하고, 박탈 규칙은 부여된 실행 권한을 취소하고자 할 때 정의한다. (그림 10)은 정책 서버의 PDP 에서 PEP로 위임된 정보를 사전에 전송한 다음, 위임된 정책 서버에서 수행할 정책 정보를 PEP로 전송하면 PEP는 위임된 정보가 있는지 확인한 후 위임된 정보가 있는 경우에 정책을 해석하여 구동자의 오퍼레이션을 인가 또는 금지하는 구조이다. 이는 정책관리시스템이 분산환경으로 이루어진 경우에 정책을 위임하거나 박탈하는



(그림 9) 인가 및 금지 규칙 프로세스



(그림 10) 위임 및 박탈 규칙 프로세스

서비스 접근 제어 규칙이다. 정책정의언어에서는 정책서버에서 위임된 서버정보는 grantee 키워드를 사용하고, 위임 규칙인 경우 delg 키워드를, 박탈 규칙인 경우에는 revc 키워드를 사용하여 표현한다.

4.3 정책 문법

정책 규칙은 정책을 생성할 때 정책 적용 시나리오 별 정책 규칙의 종류를 제공함으로써 정책

정책 문맥은 정책의 내부 구조에 대한 전반적인 내용을 담고 있으며, 정책에 기술한 내용을 처리하는 조건과 실행, 예외처리, 충돌, 메타데이터 등의 정책을 수행하기 위한 기능 컴포넌트를 제공한다. 정책의 기본 개념 및 구조를 표현하기 위한 PCIM 개념과 다양한 시나리오를 정의하기 위한 확장된 문법을 조합하는 개념이다. 정책을 상세하게 기술하기 위해서는 정책의 기본 개념과 프레임웍 외에도 기본 개념의 확장과 추가적인 기능들이 필요하게 된다. 정책 문맥의 접근 방법은 개념을 분석하고, 이를 구조화하여 모델링 한다.

(그림 11)은 중립적인 정보모델인 CIM의 Managed Element, Managed System Element, Logical Element, System 클래스와 정책 모델인 PCIM의 Policy, Rule, Condition, Action 클래스를 기반으로 하여 제안한 PIM의 클래스로 확장하여 정책정보모델을 정의한다. 모델 관점에서는 스테로타입으로 <<CIM>>, <<PCIM>>, 제안한 <<PIM>> 을 조합하여 구성하였고, 설계 관점에서는 정책 도메인, 정책 규칙, 정책 문맥 단계를 통해 정책정보모델이 정의된다. 조건과 실행은 확장하고, 추가 기능으로 이벤트, 메타데이터, 제약사항, 파라미터, 메타데이터를 보여주고 있다.

조건과 실행은 기본적인 정책의 구성으로 정책은 조건과 실행으로 정의한다. 조건은 단일조건과 시간조건으로 구분

〈표 1〉 조건 종류

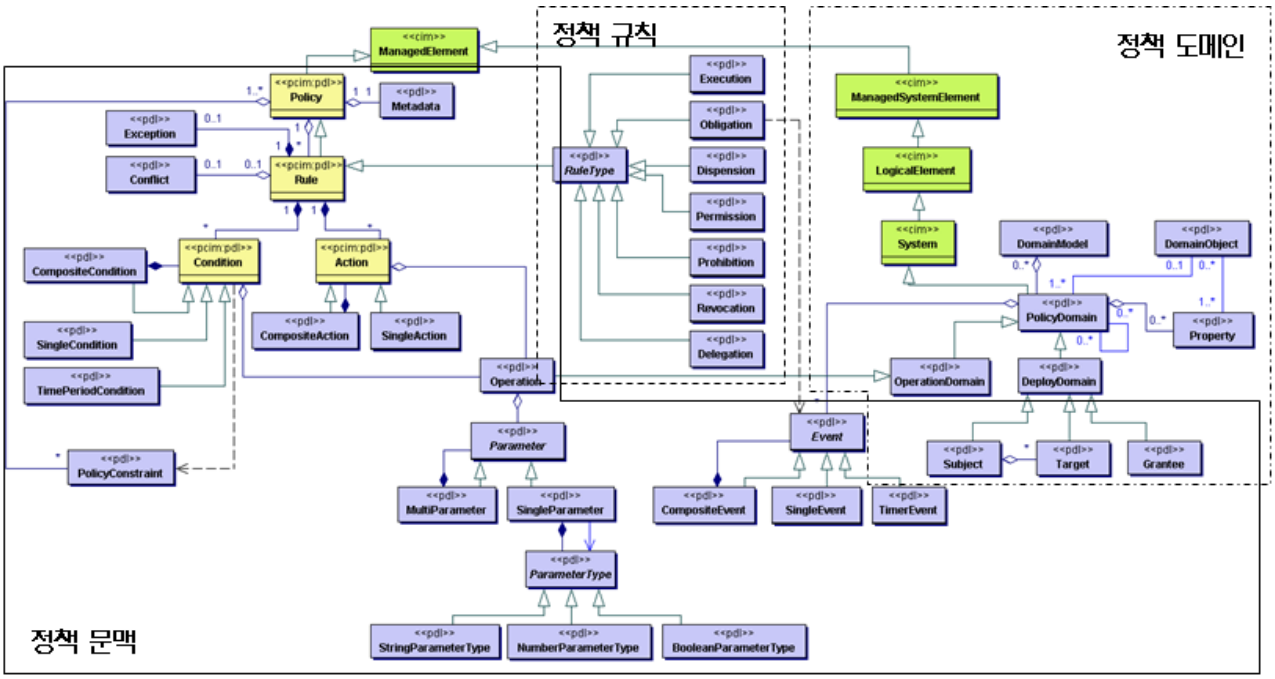
종류	설명 및 예
단일 조건	한번의 실행으로 True 또는 False 값을 반환하는 조건 예) t1.isValid() 예) t1.getValue() > 1
복합 조건	단일 조건을 조건 연산자로 결합한 복합된 조건 예) t1.isValid() && (t1.getValue() > 1) 예) t1.isValid() && time > 2005.7.17
시간 조건	시간의 제약을 필요로 하는 조건 예) time > 2005.7.17 예) time == “1thW”

〈표 2〉 실행 종류

종류	설명 및 예
단일 실행	한번의 실행으로 종료가 되는 실행 예) t1.start() 예)t1.start(“2007.7.17”)
복합 실행	단일 실행을 실행연산자 (, →)로 묶은 복합 적인 실행 예) t1.init() → t1.start() 예) t1.startVacine(), t1.startMemo()

되며, 이 두 가지의 조합으로 이루어진 복합조건으로 구성된다. 단일 조건은 산술 연산자 및 비교 연산자를 통하여 구조화 하고, 시간 조건은 비교 연산자를 통하여 구조화하며, 복합 조건은 단일 조건과 시간 조건을 조건 연산자를 통하여 구조화 된다. <표 1>은 조건의 예시이다. 그리고 행위는 단일실행과 단일실행의 조합으로 이루어진 복합실행으로 구성된다. <표 2>는 실행 종류와 예시를 보여준다.

이벤트(Event)는 의무 규칙일 경우에만 적용되는 매커니즘으로 단일 이벤트와 특정시간 또는 시간간격으로 발생하



(그림 11) 정책정보모델의 클래스 다이어그램

〈표 3〉 이벤트 종류

종류	설명 및 예
단일 이벤트	한번의 이벤트가 발생 예) App2start 예) /Mobile/WIPI/App1Start
시간 이벤트	반복적으로 발생하는 이벤트 또는 특정시간에 발생하는 이벤트일 경우에 Timer를 사용하여 설정 예) Timer.repeat(3,"hour") 예) Timer.start(2005.06.07.13.10.00)

는 시간 이벤트로 구분되며, 이 두 가지의 OR 조건으로 연결되는 복합 이벤트로 구성된다. <표 3>은 이벤트 종류와 예시를 보여준다.

제약사항(Constraint)은 정책과 규칙을 제약하는 메커니즘으로 선언적인 제약 조건으로 그 내용을 기술한다. 제약사항은 상황을 제어하기 위해서 사용하며, 규칙에 독립적인 조건을 선언하여 규칙의 처리 흐름을 제어하게 된다. <표 4>는 제약사항의 종류와 예시를 보여준다.

파라미터(Parameter)는 조건과 실행을 다양하고 세밀하게 표현하는 것을 가능하게 함으로써 정책의 확장성을 부여한다. 구성은 단일 파라미터와 두 가지의 단일 파라미터의 이루어진 멀티 파라미터로 구성된다. 파리미터 타입의 종류는 string, number, boolean, target, grantee, constraint 의 6가지 만으로 제한하여 구성한다. <표 5>는 파라미터 종류와 예시를 보여준다.

배열(Array)는 사용자의 편의성을 위하여 선언 형태로 배열 변수를 선언하고, 실행에서 재사용할 수 있는 형태이다. 배열은 constraint 키워드처럼 array 키워드를 사용하여 선언하고, 실행 안에 파라미터 형태로 사용한다. <표 7>은 array의 종류와 예시를 보여준다.

메타데이터(metadata)는 정책 자체가 가져야 할 데이터를 기술하는데 사용된다. 메타데이터를 작성하고 배포할 때 정책에 대한 추가적인 정보를 기술하는데 활용된다. 메타데이터 기술은 정책 도구에서 정책을 직접 작성하거나 또는 속성 창을 통하여 값을 입력하게 된다. 메타데이터는 정책

〈표 4〉 제약사항 종류

종류	설명 및 예
선언적 조건제약	조건을 제약사항으로 선언하여 규칙에 독립적으로 적용 예) constraint c1 = t1.isValid()

〈표 5〉 파라미터 종류

종류	설명 및 예
단일	한 개의 파라미터로 구성되는 경우 예) t1.sayWord("hello")
멀티	여러 개의 파라미터로 구성되는 경우 예) t1.sayStatement("hello","world")

과 1:1 의 관계를 가지며 다른 객체와는 달리 속성값으로 그 내용이 표현된다.

정책의 에러는 정책의 규칙을 실행하기 전에 규칙간에 발생할 수 있는 충돌(Conflict)과 정책의 실행 중에 발생하는 예외처리(Exception)로 구분된다. 이 두 가지 에러를 분명하게 정의 함으로써 정책의 무결성을 보장하고 에러 처리를 하게 된다. (그림 12)는 정책 규칙간의 정책 충돌이 발생되는 영역을 보여준다. 이를 해결하기 위하여 컴파일 시 발생할 수 있는 정책 내 또는 정책들에 정의된 규칙 타입 간 실행들에 대한 충돌 현상을 제거하여야 한다.

1. 허가 & 금지: 하나의 정책 내에서 금지 규칙을 부여하고, 다른 규칙에 허가를 부여하는 경우 실행에서 발생하는 충돌 현상을 말한다.

2. 금지 & 수행: 정책 내에서 금지 규칙의 실행을 부여하고, 동일 또는 다른 정책에서 수행 규칙에서 실행을 부여하는 경우 발생하는 충돌 현상을 말한다.

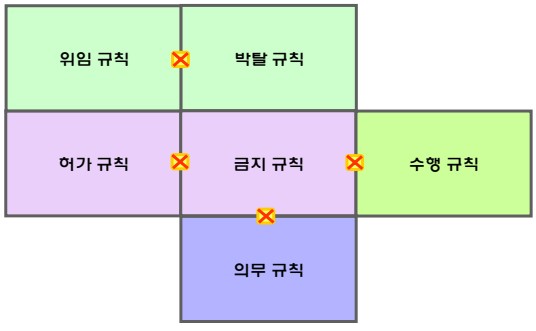
3. 금지 & 의무: 정책 내에서 금지 규칙의 실행을 부여하

〈표 6〉 파라미터 타입 종류

종류	설명 및 예
문자형	string type의 파라미터 형태 예) type ruleType1(string s1) 예) t1.sayWord("hello"+1)
숫자형	숫자 type의 파라미터 형태 예) type ruleType1(number n1) 예) t1.sum(1,2)
불린형	불린 type의 파라미터 형태 예) type ruleType1(boolean s1) 예) t1.isValid(true)

〈표 7〉 배열 종류

종류	설명 및 예
string	String 타입으로 array 변수를 지정 예) array t1= ["12", "34", "56"]
number	Number 타입으로 배열 변수를 지정 예) array t1= [12, 34, 56]
Boolean	Boolean 타입으로 배열 변수를 지정 예) array t1= [true, false]



(그림 12) 정책 규칙간의 충돌 관계

고, 동일 또는 다른 정책에서 의무 규칙에서 실행을 부여하는 경우 발생하는 충돌 현상을 말한다.

4. 위임 & 박탈: 하나의 정책 내에서 위임 규칙을 부여하고, 다른 규칙에 위임을 부여하는 경우 실행에서 발생하는 충돌 현상을 말한다.

5. 검증 및 비교연구

5.1 정책 시나리오 기반의 검증

모바일 환경에서 일어날 수 있는 두 가지 시나리오를 제안한 정책정의언어로 기술한다. <표 8>의 첫 번째 시나리오는 의무규칙으로 전화 걸기 이벤트인 call이 발생되고, 파라미터 값이 sync 이면 t1.place()를 수행하여 전화 걸기를 막는다. 두 번째 사니리오는 수행규칙으로 sync manager가 준비상태인 경우에 t3.getFile()를 호출하면 특정 디렉토리의 구성파일을 업데이트하여 애플리케이션이 동작할 때 변경된 구성파일의 값을 가지고 업무 처리를 하게 된다.

S1. 의무규칙: 고객이 단말 분실로 인하여 전화 걸기를 제어하고자 한다.

S2. 수행규칙: SycManager의 구성파일의 데이터를 업데이트한다.

〈표 8〉 시나리오기반의 정책정의언어

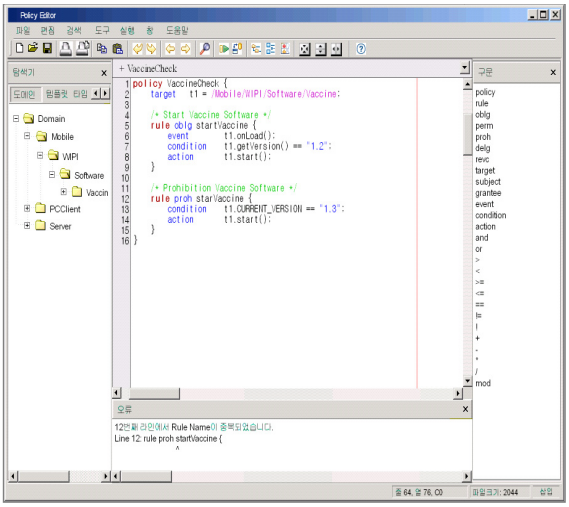
```
policy scenario12345 {
  target t1 = /Mobile/Application/CALL;
  //의무규칙
  rule oblg stopCALL{
    event call ; //전화걸기 이벤트가 발생한다.
    when (LPARAM=="sync") //파라미터 값이 일치
    action t1.place(); //전화걸기를 막는다.
  }
  //수행규칙
  rule exec getFile {
    target t3 = /SycManager/synEnabler;


      condition t3.getReady() == true;
      //config.txt 파일을 업데이트한다.


    action t3.getFile("config",
  "http://127.0.0.1:8080/config.txt"); }
}
```

5.2 정책 도구

정책 도구(Policy Editor)는 정책을 생성 및 변경, 조회, 삭제한다. 정책은 정책저장소에 저장되며, 정책 도구는 정책저장소를 연동하여 정책을 관리하게 된다[2][12][13]. 정책도구는 정책 정의를 위한 편집기능, 문맥 검사, 파싱, 정책결과물 등이 포함된다. (그림 13)은 본 논문에서 제안한 정책정의언어를 작성하도록 개발된 정책도구이다. 정책도구는 SableCC[14]를 이용하여 컴파일러를 구현하고, JEdit [15]오픈 소스를 이용하여 정책 도구를 개발하였다.



(그림 13) 정책 도구

〈표 9〉 타 정책정의언어와의 비교

Policy Definition Language			PPL [5]	PDL [6]	LaSCO [7]	ASL[8]	Our Work
일반적인 특성	정책언어 분석		△	●	△	△	●
	언어의 확장성		–	△	–	–	●
	도메인 및 그룹핑		●	–	△	–	●
정책규칙타입	적용제어	수행 규칙	–	–	–	–	●
		의무 규칙	–	●	–	–	●
	접근제어	허가 규칙	●	–	●	●	●
		금지 규칙	●	–	●	●	●
		위임 규칙	–	–	–	–	●
		박탈 규칙	–	–	–	–	●

(●) Support, (△) Partially Support, (–) Not Support

5.3 타 정책정의언어 비교

일반적인 정책 특징과 본 논문에서 제시한 정책규칙타입에 매핑하여 타 정책정의언어에서 제공하고 있는 특징들을 비교 분석하였다. 본 연구에서 제시한 정책정의언어의 특징은 다양한 도메인에 적용할 수 있는 구조와 수행 규칙처럼 모바일 환경의 수행능력을 고려하여 정책을 즉시 수행할 수 있는 규칙 타입을 적용하여 제시한 부분이 타 연구와 비교되는 점이라 할 수 있다. <표 9>는 정책정의언어의 일반적인 특성과 정책규칙 타입별로 타 정책 정의 언어와 비교 분석한 결과이다.

6. 결론 및 향후 연구과제

최근 모바일 환경에서 소프트웨어 관리의 문제점인 서비스 오류 및 변경에 빠르게 해결하기 위한 새로운 대안으로 정책 관리 프레임웍을 활용하는 여러 연구가 진행되고 있다. 기존의 정책정의언어들은 호스트, 네트워크 기반의 도메인을 대상으로 정의되었기 때문에 모바일 환경에서 정책 관리 프레임웍을 적용하기 위한 정책정보모델을 기반한 정책정의언어 연구는 거의 이루어지고 있지 않고 있다.

따라서, 본 연구에서는 이러한 모바일 환경에서 어플리케이션의 문제점을 해결하여 유지보수 비용을 절감하고 민첩한 서비스를 적용할 수 있는 효율적인 방법으로 이를 구현하는데 중요한 요소인 정책정의언어를 정의하였다. 제안한 정책정의언어는 시나리오를 정책정의언어로 표현하여, 이를 정책 도구로 검증하였고, 타 정책정의언어들과 비교 분석하였다.

본 연구에서 제시한 정책정의언어는 정책 도메인과 정책규칙 영역을 변경하여 모바일 환경뿐만 아니라, 다양한 분산 환경의 이 기종 시스템 도메인에 적용할 수 있다.

향후 연구 과제로는 정책관리시스템 구현 단계에 변경 없이 런타임 시에 기존 서비스를 구성하고 관리하는 매커니즘을 적용할 수 있는 정책 프레임웍 설계에 대한 연구가 수행되어야 한다.

참 고 문 헌

[1] Common Information Model CIM : http://www.dmtf.org/standards/standard_cim.php

[2] IETF Policy Core Information Model (PCIM): http://www.ietf.org/rfc/rfc3060.txt

[3] The Parlay Group, Comparing OMA OSE and Parlay Architectures, March, 2005.

[4] Open Mobile Alliance PEEM : http://www.openmobile.com

[5] Stone, B. LUNDY, et al. 2001. Network Policy Languages: A Survey and a New Approach. IEEE Network: 10-20.

[6] J. LOBO, R. BHATIA, et al. 1999. A Policy Description Language. AAAI, Orlando, Florida.

[7] J. Hoagland et al., Security Policy Specification Using a Graphical Approach, Technical report CSE-98-3, University of California, Davis Department of Computer Science, 1998.

[8] S. Jajodia et al., A Logical Language for Expressing Authorizations, Proceedings of the IEEE Symposium on Security and Privacy 1997, pp.31-42.

[9] G. Russello, C. Dong, N. Dulay, Authorization and Conflict Resolution for Hierarchical Domains, Proceedings of the 8th International Workshop on Policies for Distributed Systems and Networks (POLICY'07), IEEE, 2007.

[10] E. Lupu and M. Sloman, Conflicts in Policy-based Distributed Systems Management, Proceedings of the 5th IEEE/IFIP International Symposium on Integrated Network Management, IEEE, 1999.

[11] K. Verlaenen, B. De Win, W. Joosen, Towards simplified specification of policies in different domains, IEEE, 2007.

[12] N. Damianou, N. Dulay, E. Lupu, M. Sloman, Tools for Domain-based Policy Management of Distributed Systems, IEEE, 2002.

[13] A. Pilz, Policy-Maker a Toolkit for Policy-Based Security Management, INST-CNR, 2007.

[14] E. Gagon, SableCC: An Object-Oriented Compiler Framework, School of Computer Science, McGill University. Montreal, Canada, March, 1998.

[15] jEdit Programmer's Text Editor: http://www.jedit.org

안 성 욱

e-mail : swahn74@ssu.ac.kr

1999년 숭실대학교 컴퓨터학부(학사)

2001년 숭실대학교 컴퓨터학과(공학석사)

2007년~현 재 숭실대학교 컴퓨터학과 박사과정

관심분야 : 소프트웨어 재사용, 서비스 지향 아키텍처, 정책 관리 프레임웍, 오픈소스 소프트웨어

류 성 열

e-mail : syrhew@ssu.ac.kr

1981년~현 재 숭실대학교 컴퓨터학부 교수

1982년~1995년 숭실대학교 전자계산연구소 및 중앙전자계산소 소장

1997년~1998년 George Mason University 객원 교수

1998년~2001년 숭실대학교 정보과학대학원 원장

2004년~현 재 한국품질재단 운영위원회 위원장

2006년~현 재 공정거래위원회 성과관리위원회위원

2008년~현 재 정보통신연구진흥원 비상임이사

관심분야 : 소프트웨어 유지보수, 소프트웨어 재사용, 오픈소스 소프트웨어, 정책 관리 프레임웍