

소프트웨어 제품라인 아키텍처 모델에서의 가변성 표현 방법 비교 연구[†]

(Expressing Variability in Software Product Line Architecture Models:
A Comparative Study)

이혜선[‡] 조성배[§] 강교철[¶]
(Hyesun Lee) (Sungbae Cho) (Kyo Chul Kang)

요약 소프트웨어 제품라인 공학은 제품라인의 공통적인 부분과 차이점을 핵심 자산으로 관리하여 품질을 높이고, 핵심 자산을 계획된 제품들에 재사용함으로써 소프트웨어의 생산성을 높이는데 효과적인 방법이다. 제품라인 아키텍처에는 제품에 따라서 포함되거나 포함되지 않는 가변적인 부분이 있기 때문에, 기존 단일 소프트웨어를 대상으로 한 아키텍처 모델과는 달리, 제품라인 아키텍처는 제품 간의 차이점인 가변성을 나타낼 수 있어야 한다. 기존에 여러 연구자가 제품라인 아키텍처 모델에 가변성을 표현하는 방법을 제안하였지만, 이들이 제시한 표현 방법들의 강약점을 분석하고 차이점을 비교하는 연구가 부족하였다. 따라서 본 논문에서는 현재까지 제안된 제품라인 아키텍처 모델에서의 가변성 표현 방법을 분석하고 비교하여, 적합한 표현 방법을 선택하는데 가이드가 되고자 한다.

키워드 테스트 소프트웨어 제품라인 공학, 가변성 구현, 소프트웨어 제품라인 아키텍처

Abstract Software product line engineering is a software reuse paradigm that helps organizations improve software productivity and quality by developing software products from reusable core assets. For the satisfaction of common and variable requirements among products in the product line, the core assets must be configurable according to the selection of variable features. Therefore, unlike software architecture model of a single product, product line architecture model must embed and express variabilities among the products. Many researches have proposed methods of embedding and expressing variabilities in the product line architecture models, but there are few comparative studies on the proposed methods. In this paper we discuss strong points and weak points of the proposed methods and compare expressiveness of the methods, which helps select a proper method.

Key words Software product line engineering, variability, software product line architecture

[‡] 본 연구는 정보통신산업진흥원의 SW공학 요소기술 연구 개발사업 및 한국연구재단을 통해 교육과학기술부의 세계 수준의 연구중심대학육성사업(WCU, R31-10100)으로부터 지원받아 수행되었습니다.

[‡] 학생회원 : 포항공과대학교 컴퓨터공학과
compial@postech.ac.kr

[§] 비 회 원 : 포항공과대학교 정보전자융합공학부
craxy2010@postech.ac.kr

[¶] 종신회원 : 포항공과대학교 정보전자융합공학부 교수
kck@postech.ac.kr

논문접수 : 2011년 09월 03일

심사완료 : 2011년 09월 23일

1. 서 론

소프트웨어 제품라인 공학(Software Product Line Engineering)[1]은 어떤 도메인(Domain)에서 시장 요구를 만족하고 공통된 기능을 공유하며 핵심 자산(Common Assets)으로부터 개발되는 유사한 제품군을 개발하는 새로운 재사용

패러다임이다. 소프트웨어 제품라인 방법론을 통하여 소프트웨어를 개발하면 제품라인의 공통적인 부분과 차이점 부분을 핵심 자산으로 관리하여 품질을 높이고 핵심 자산을 계획된 제품들에 재사용함으로써 재사용성을 보장 받을 수 있다[2].

소프트웨어 제품라인의 핵심 자산(요구사항, 아키텍처, 코드 등을 포함)으로부터 계획된 제품들을 개발하기 위해서는, 제품라인의 제품들 사이에 공통점(Commonality)과 차이점(Difference)이 분석되고, 분석된 차이점이 핵심 자산에 가변성(Variability)으로써 구현이 되어야 한다. 예를 들어, 어떤 휘처(Feature; 제품라인에서 제품군의 공통점과 차이점 분석 단위)가 제품에 따라서 포함이 되거나 포함이 되지 않는 휘처라면, 핵심 자산 중 이 휘처를 구현하는 아키텍처 컴포넌트 및 자산 코드가 휘처의 선택에 따라서 제품에 포함되거나 포함되지 않도록 할 수 있어야 한다. 이러한 가변성을 분석하고 관리하는 것이 성공적인 제품라인 공학을 위해 필수적이다.

제품라인 아키텍처에는 제품에 따라서 포함되거나 포함되지 않는 가변적인 부분이 있기 때문에, 기존 단일 소프트웨어를 대상으로 한 아키텍처 모델과는 달리 아키텍처 모델에 제품라인의 가변성을 나타낼 수 있어야 한다. 이를 위해 여러 연구자가 제품라인 아키텍처 모델에 가변성을 표현하는 방법을 제안하였다 [3-4, 6-12]. 하지만 이들이 제시한 제품라인 아키텍처 모델에서의 가변성 표현 방법들의 차이점 및 강약점을 비교하는 연구가 부족하였다.

따라서 본 연구에서 현재까지 제안된, 제품라인 아키텍처 모델에서의 가변성 표현 방법을 분석하고 강약점을 비교하여, 적합한 표현 방법을 선택하는데 가이드가 되고자 한다. 본 논문에서 엘리베이터 컨트롤 시스템 제품라인이 표현 방법을 적용할 예제로 사용되었다. 본 논문은 아키텍처 모델 상의 가변성 표현 방법 비교에만 초점을

맞추며, 컴포넌트 명세 방법 및 컴포넌트 명세에서의 가변성 표현은 본 논문의 범위에 벗어난다.

본 논문의 구성은 다음과 같다. 2장에서는 예제로 사용될 엘리베이터 컨트롤 시스템 제품라인과 해당 제품라인 아키텍처 모델에 대해 간략히 설명하고자 한다. 3장에서는 본격적으로 예제와 함께 각 아키텍처 모델에서의 가변성 표현 방법을 소개하고 강약점을 분석할 것이다. 4장에서는 분석한 내용을 바탕으로 아키텍처 모델에서의 가변성 표현 방법들을 비교할 것이다. 마지막으로 5장에서는 토론과 함께 논문을 마무리하고 향후 연구를 소개하고자 한다.

2. 사례 연구 : 엘리베이터 컨트롤 시스템 제품라인

본 장에서는 본 논문의 예제로 사용될 엘리베이터 컨트롤 시스템(Elevator Control System: ECS) 제품라인을 소개하고자 한다. ECS는 실시간 임베디드 시스템(Realtime Embedded System)의 일종으로써, 사용자를 건물의 층 사이에 편리하고 빠르게 이동시키는 것이 목적이다. 본 논문에서 대상으로 하는 ECS 제품라인은 고층 빌딩 엘리베이터, 종합 병원 엘리베이터, 아파트 엘리베이터를 포함하며, 제품라인의 공통점 및 차이점 휘처를 나타내는 휘처 모델은 그림 1과 같다.

ECS 제품라인의 휘처 모델(그림 1)에서 Driving Service, Call Handling, Indication Handling, Door Handling 등은 모든 제품에 공통적으로 들어가는 필수(Mandatory) 휘처이다. Driving Service 중 VIP(특정한 층에만 엘리베이터가 멈추도록 엘리베이터를 운행하는 서비스)와 Emergency (지진, 화재 등의 위급 상황에 안전을 고려하여 엘리베이터를 운행하는 서비스)는 일부 제품에만

들어가는 선택적(Optional) 휘처이다. Motor의 타입을 나타내는 Hydraulic과 Electric은 둘 중 하나만 선택될 수 있는 택일(Alternative) 휘처이다.

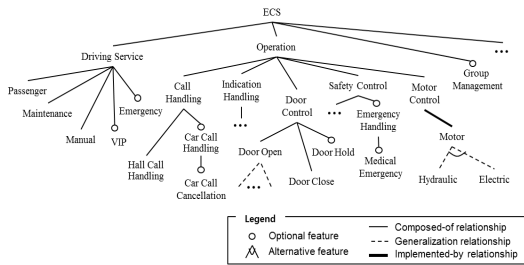


그림 1. ECS 제품라인의 휘처 모델

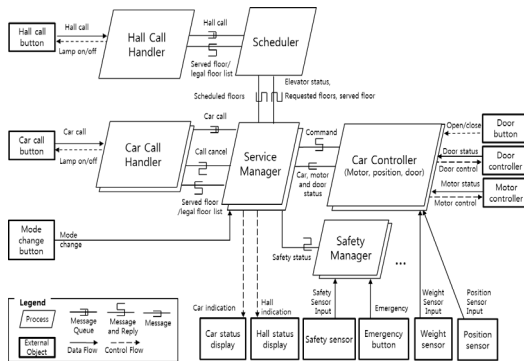


그림 2. ECS 제품라인의 프로세스 아키텍처 모델

ECS 제품라인의 프로세스 아키텍처 모델은 그림 2와 같다. 프로세스 아키텍처 모델에서 프로세스는 동시에 수행될 수 있는 실행 단위를 나타낸다. ECS는 건물의 홀에서부터 오는 Call을 처리하는 Hall Call Handler 프로세스, 엘리베이터의 캐이지로부터 오는 Call을 처리하는 Car Call Handler 프로세스, 엘리베이터가 이동할 층을 스케줄링하는 Scheduler 프로세스, 각 엘리베이터의 상태를 관리하고 Driving Service를 관리하는 Service Manager 프로세스, 엘리베이터 캐이지를 관리하는 Car Controller 프로세스, 그리고 안정성을 관리하는 Safety Manager 프로세스로 구성된다.

그림 2에서 프로세스 아키텍처 모델의 가변성은 현재 표현이 되지 않았지만, 프로세스 및 커넥터 중 휘처의 선택에 따라서 제품에 포함되거나 포함되지 않아야 할 부분이 있다. Car Call Handler 프로세스는 Car Call Handling 휘처가 선택될 때에만 제품에 포함되는 프로세스이고, Safety Manager 프로세스는 Emergency Handling 휘처가 선택될 때에만 제품에 포함되는 프로세스이다. Emergency button 외부 객체는 Hospital Emergency 휘처가 선택될 때에만 제품에 포함되는 외부 객체이다.

다음 장에서는 ECS 제품라인의 프로세스 아키텍처 모델을 예제로 하여, 각 가변성 표현 방법을 분석하고 강약점을 파악할 것이다.

3. 소프트웨어 제품라인 아키텍처 모델에서의 가변성 표현 방법

3.1. 독립적 가변성 모델을 이용한 표현 방법

Thiel et al.[3]과 Pohl et al.[4]은 아키텍처 모델과 별개로 독립적인 가변성 모델(Thiel et al.은 독립적인 가변성 모델을 Variability Model이라고 명칭하고, Pohl et al.은 독립적인 가변성 모델을 Orthogonal Variability Model이라고 명칭함)을 관리하여 표현하였다. 가변성 모델에서는 가변점(Variation Point)과 가변점에 결합될 수 있는 가변부(Variant)를 명세하고, 가변점과 가변부간의 요구(Require) 관계와 배척(Exclude) 관계를 명세 한다. 본 절에서는 그 중 Thiel et al.의 방법을 중점적으로 살펴보고자 한다.

Thiel et al.은 제품라인의 다양한 관점의 아키텍처 모델(개념적 아키텍처, 프로세스 아키텍처, Deployment 아키텍처, Physical 아키텍처)에서의 가변성을 나타내는 방법을 제안하였다(모듈 아키텍처는 대상으로 하지 않았지만 동일한 표현 방법으로 확장이 가능하다). 각 아키텍처 모델의

아키텍처 컴포넌트가 가변부가 될 수 있다고 보았으며, 각 아키텍처 모델마다 가변부와 가변점과의 관계를 독립적인 가변성 모델로 나타내었다. 그리고 각 아키텍처 모델의 독립적인 가변성 모델 사이의 관계 또한 표현하였다.

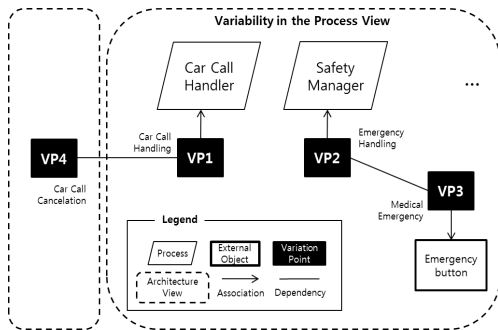


그림 3. Thiel et al.의 표현 방법으로 나타낸 ECS 제품라인의 프로세스 아키텍처 모델의 가변성 모델

그림 3은 ECS 제품라인의 프로세스 아키텍처의 가변성을 Thiel et al.의 방법으로 나타낸 그림이다. 제품라인 프로세스 아키텍처의 가변성 모델에서 “VP”는 가변점을 의미하며, 휘처 모델의 가변 휘처와 맵핑된다. 독립적인 가변성 모델을 통해서 Car Call Handler 프로세스는 Car Call Handling 휘처가 선택될 때 제품에 포함이 되고, Safety Manager 프로세스는 Emergency Handling 휘처가 선택될 때 제품에 포함이 된다는 것을 알 수 있다.

Thiel et al.의 표현 방법의 장점은 가변성을 독립적인 모델로 관리함으로써 제품라인 아키텍처 모델에 가변성을 함께 표시하는 것보다 복잡도가 감소한다는 것이다. 그리고 다양한 관점의 아키텍처 모델을 고려하였으며, 서로 다른 관점의 아키텍처 모델에 대응되는 가변성 모델 사이의 관계를 명시하였기 때문에 서로 다른 관점의 아키텍처 모델에 구현된 가변성 사이의 관계를 파악할 수 있다.

하지만 독립적인 가변성 모델을 관리하기 때문에 아키텍처 모델만으로는 가변성을 파악할 수

없다는 약점이 있다. 또한, 아키텍처의 컴포넌트만을 가변 대상으로 하며 커넥터는 가변 대상으로 하지 않는다. 따라서 커넥터에 가변성이 있는 경우(예를 들면, 프로세스 모델에서 두 프로세스의 연결 방법이 제품마다 다를 수 있는데, 어떤 제품은 메시지 큐(Message Queue) 방법[5]을 이용하지만 어떤 제품은 메시지 리플라이(Message Reply) 방법[5]을 이용할 수 있다.) 이를 표현할 수 없다. 또한, 아키텍처 컴포넌트 내부에 가변성이 있는 경우(Internal Variability; 예를 들어 어떤 휘처의 선택에 따라서 컴포넌트 내부의 구현이 달라질 경우)는 가변성 표현 대상으로 고려하지 않는다.

3.2. Gomaa의 표현 방법

Gomaa[6, 7]는 UML의 스테레오타입(Stereotype)을 확장하여 다양한 UML 모델에 가변성을 표현하였다. 본 절에서는 Gomaa의 표현 방법 중 UML 아키텍처 모델(컴포넌트 모델, 패키지 모델, 클래스 모델)에 가변성을 표현한 방법에 초점을 맞추었다.

Gomaa는 휘처 모델과 유즈케이스(Usecase)를 바탕으로 컴포넌트의 기능 및 컴포넌트가 구현하는 휘처에 따라 컴포넌트 종류를 다양하게 구별하였다. 제품에 항상 포함이 되는 필수적인 컴포넌트는 “kernel”을, 제품에 포함이 되거나 포함이 되지 않을 수 있는 선택적인 컴포넌트는 “optional”을, 컴포넌트 내부가 변할 수 있는 변화를 내포하고 있는 컴포넌트는 “variant”를 컴포넌트의 스테레오타입에 명시하도록 하였다. 가변점이 되는 컴포넌트는 “vp”로 컴포넌트의 스테레오타입에 명세하였는데, 그 중 추상(Abstract) 컴포넌트는 “abstract-vp”로, 매개변수화(Parameterization)를 사용하는 컴포넌트는 “param-vp” 표현하였다. (또한 “control”, “coordinator”, “data collection”, “external output component” 등 컴포넌트의 역할을 구분하여 이들 컴포넌트가 프로세스로 어떻게 동작하는지를 표현할 수 있도록 하였다.)

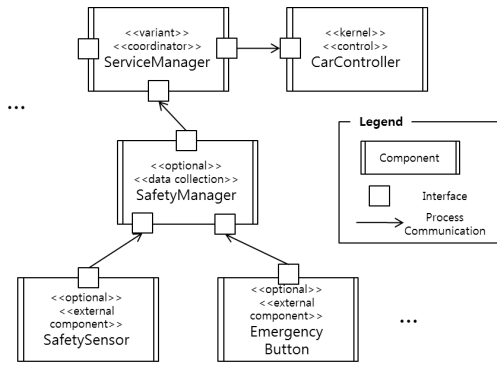


그림 4. Gomaa의 표현 방법으로 가변성을 나타낸 ECS 제품라인의 프로세스 아키텍처 모델

그림 4는 Gomaa의 표현 방법으로 ECS 제품라인의 프로세스 아키텍처 모델의 가변성을 나타낸 그림이다(그림 4의 UML 컴포넌트 모델에서 각 컴포넌트는 프로세스를 의미한다). “<<kernel>>”로 표시된 CarController 컴포넌트는 모든 제품에 포함이 되는 컴포넌트이며, “<<optional>>”로 표시된 SafetyManager 컴포넌트는 일부 제품에 포함이 되는 컴포넌트이고, “<<variant>>”로 표시된 ServiceManager는 제품에 따라서 내부 구현이 바뀌는 컴포넌트임을 알 수 있다.

Gomaa의 표현 방법의 장점은 UML 기본 표현 방법을 확장하여 사용하였기 때문에 모든 UML 모델(컴포넌트 아키텍처, 패키지 아키텍처 등)에 대해 적용 가능하며 기존 UML 모델링 도구를 사용할 수 있다는 점이다. 또한, “<<variant>>”를 사용하여 컴포넌트 내부의 가변성도 표현이 가능하다.

하지만 UML의 스테레오타입에 컴포넌트의 종류만 나타내고 가변 컴포넌트에 대응되는 휘처는 표현하지 않기 때문에, 아키텍처 모델만 보고 각 가변점/가변부와 관련이 있는 휘처를 파악하기 어려운 점이 있다. 예를 들어, 그림 4에서 SafetyManager 컴포넌트가 제품에 따라서 포함될 수 있는 컴포넌트임을 알 수 있지만 어떤

휘처가 선택됐을 때 제품에 포함이 되는지는 알 수 없다. 또한, 아키텍처 컴포넌트만을 가변성의 대상으로 보고, 컴포넌트의 커넥션은 가변성의 대상으로 보지 않는다(클래스 모델에서도 클래스 단위가 아닌, 속성(Attribute)과 함수(Method) 단위에 대해서는 가변성의 언급이 없다). 하지만 컴포넌트의 커넥션, 클래스 모델의 속성과 함수에도 UML 스테레오타입을 확장한 가변성 표현 방법을 사용하면 가변성을 나타낼 수 있을 것이다.

3.3. Moon et al.의 표현 방법

Moon et al.[8]은 제품라인의 아키텍처 모델의 메타 모델(Meta Model)을 그림 5와 같이 정의하였는데, 제품라인 아키텍처 컴포넌트와 인터페이스, 함수(Operation)에 가변성을 나타낼 수 있는 CV_property 타입의 “vp” 속성을 추가하였다. CV_property 타입은 “common” 또는 “optional”로 정의되어 해당 개념의 가변성을 표현할 수 있다.

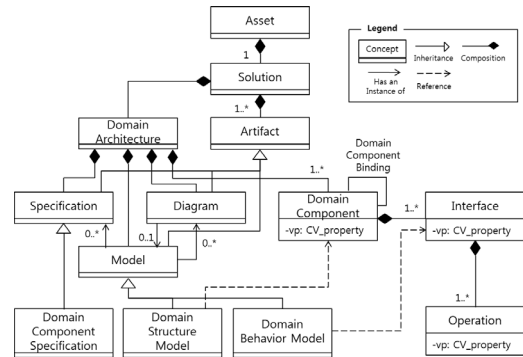


그림 5. Moon et al.의 표현 방법의 제품라인 아키텍처 모델의 메타 모델

Moon et al.은 그림 5에 정의된 제품라인 아키텍처 모델의 메타 모델을 바탕으로, 3.2절에서 소개한 Gomaa[6, 7]의 표현 방법처럼 UML의 스테레오타입을 확장하여 하여 컴포넌트, 인터페이스, 함수의 가변성을 표현하였다.

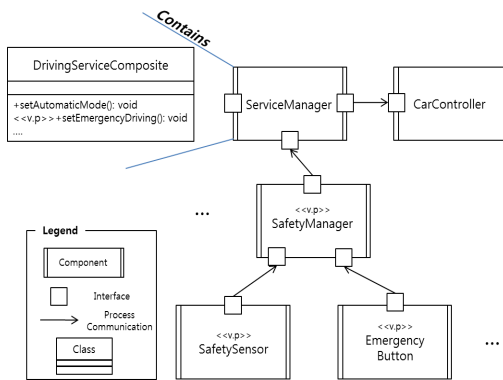


그림 6. Moon et al.의 표현 방법으로 가변성을 나타낸 ECS 제품라인의 프로세스 아키텍처 모델

그림 6은 Moon et al.의 표현 방법으로 ECS 제품라인의 프로세스 아키텍처 모델의 가변성을 나타낸 것이다. 프로세스를 나타내는 각 컴포넌트에서 “<<v.p>>”는 제품에 따라서 포함되는 아키텍처 컴포넌트를 의미한다. 그림 6에서 SafetyManager 컴포넌트, SafetySensor 컴포넌트, EmergencyButton 컴포넌트가 가변적인 컴포넌트임을 알 수 있다. 또한, ServiceManager 컴포넌트에 들어있는 DrivingServiceComposite 클래스에서 setEmergencyDriving() 함수가 가변적인 것을 알 수 있다.

Moon et al.의 표현 방법은 Gomaa처럼 UML의 스테레오타입을 확장한 것은 동일하지만, Gomaa가 다양한 스테레오타입 표기법을 정의한 것과 달리 “<<v.p>>”라는 한 가지 스테레오타입을 사용하여 표기법을 단순화하였다. 따라서 Moon et al.의 표현 방법을 적용한 아키텍처 모델이 Gomaa의 표현 방법을 적용한 아키텍처 모델보다 간결해 보이지만, 그만큼 표현력이 부족하다. Moon et al.의 표현 방법은 Gomaa의 표현 방법과 달리 컴포넌트 내부의 가변성을 표현할 수 없고, Alternative 가변성을 표현할 수 없다(Gomaa의 표현 방법에서 “<<optional>>”은 선택적 가변성을, “<<vp>>”는 택일적 가변성을 표현할 수 있지만,

Moon et al.의 표현 방법에서 “<<v.p>>”는 선택적 가변성만 표현할 수 있다). 또한 Gomaa의 표현 방법과 마찬가지로 UML의 스테레오타입에 가변 컴포넌트에 관련이 있는 휘처를 함께 표현하지 않기 때문에, 아키텍처 모델만 보고 각 가변성에 대응되는 휘처를 파악하기 어렵다. Moon et al.의 표현 방법은 다른 표현 방법들과 달리 제품라인 컴포넌트 바인딩(Domain Component Binding)에 CV_property를 정의하여, 가변적인 컴포넌트 사이의 직접적인 또는 간접적인 컴포넌트 바인딩을 고려한 것이 장점이다.

3.4. Czarnecki et al.의 표현 방법

Czarnecki et al.[9]도 Gomaa[6, 7]나 Moon et al.[8]처럼 UML 모델의 스테레오 타입을 확장하여 제품라인 아키텍처의 가변성을 표현하였다. 하지만 Gomaa나 Moon et al.의 표현 방법과는 달리, 스테레오 타입에 가변성의 타입이 아닌 휘처의 이름을 표시하였다.

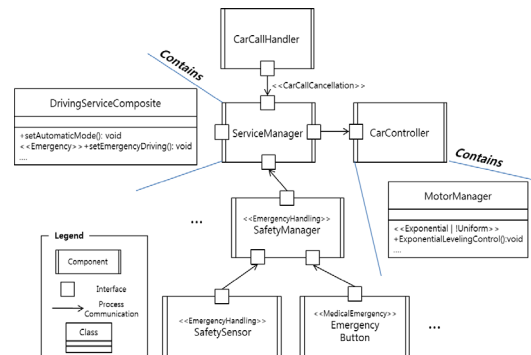


그림 7. Czarnecki et al.의 표현 방법으로 가변성을 나타낸 ECS 제품라인의 프로세스 아키텍처 모델

그림 7은 Czarnecki et al.의 표현 방법으로 ECS 제품라인의 프로세스 아키텍처 모델의 가변성을 나타낸 것이다. 프로세스를 나타내는 각 컴포넌트/커넥터 중 가변 컴포넌트/커넥터는 맵핑되는 가변 휘처 이름을 스테레오타입에 정의하고 있다. 그림 7에서 SafetyManager 컴포넌트는 Emergency Handling 휘처가 선택되었을 때

제품에 포함이 되고, EmergencyButton 컴포넌트는 Medical Emergency 휘처가 선택되었을 때 제품에 포함이 되는 것을 알 수 있다. 또한 UML 모델의 모든 요소(Element)와 관계(Relationship)에 스테레오 타입으로 관련된 휘처의 이름을 표시할 수 있도록 하였기 때문에 컴포넌트 뿐 아니라 클래스와 속성, 함수 단위까지 가변성을 표현할 수 있다. ServiceManager 컴포넌트 안에 들어있는 DrivingServiceComposite 클래스의 setEmergencyDriving() 함수는 Emergency 휘처가 선택되었을 때 제품에 포함이 된다. Czarnecki et al.의 표현 방법은 다른 표현 방법과 다르게 스테레오 타입에 휘처의 이름으로 논리 연산(Logical Operation)을 표현할 수 있다는 특징이 있다. 그림 7에서 CarController 컴포넌트 안에 들어있는 MotorManager 클래스의 Exponential LevelingControl() 함수는 “<<Exponential !Uniform>>”을 통하여 Exponential 휘처가 선택되거나 또는 Uniform 휘처가 선택되지 않을 때 제품에 포함이 된다는 것을 알 수 있다.

Czarnecki et al.의 표현 방법의 장점은 가변적인 아키텍처 컴포넌트/커넥터에 가변 휘처를 표현하기 때문에 제품라인 아키텍처 모델만 보고서도 아키텍처의 어느 가변 부분이 어느 가변 휘처와 연관이 되어 있는지를 알 수 있다는 점이다. 또한 휘처 사이의 논리 연산을 이용하여, 하나 이상의 가변 휘처의 선택 관계에 의해 제품에 포함 여부가 결정되는 아키텍처 컴포넌트/커넥터를 표현할 수 있다는 것이 강점이다. 그리고 아키텍처 모델에 표현된 논리 연산을 바탕으로 휘처 설정(Configuration)에 따라 아키텍처 모델의 생성을 검증하는 규칙을 함께 제안하고 있다.

하지만 아키텍처 컴포넌트/커넥터가 택일적 휘처와 맵핑되는 경우, 택일 휘처 사이의 관계가 아키텍처 모델에 나타나지 않거나 복잡하게 나타난다(택일 휘처 사이의 관계를 휘처 사이의 논리 연산으로 나타낼 경우, 택일 휘처의 수가

늘어날수록 가능한 조합 경우가 늘어나서 논리 연산이 복잡해진다). 또한, Gomaa의 표현 방법과 달리 컴포넌트 내부의 가변성을 나타낼 수 없다.

3.5. Yoon의 표현 방법

Yoon[10]은 UML을 확장하지 않고 그림 8과 같이 독자적인 아키텍처 모델을 정의 하였다. 가변성의 대상은 다양한 아키텍처 모델의 컴포넌트와 커넥터였는데, 그림 8에서처럼 “선택 또는 택일”과 “컴포넌트 또는 커넥터”의 조합에 따라서 도형의 색과 모양을 다르게 나타내어 이를 구분하였다.

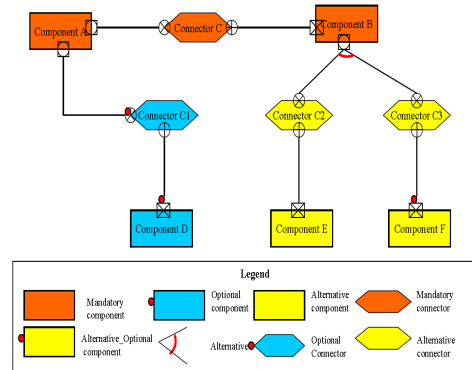


그림 8. Yoon의 표현 방법

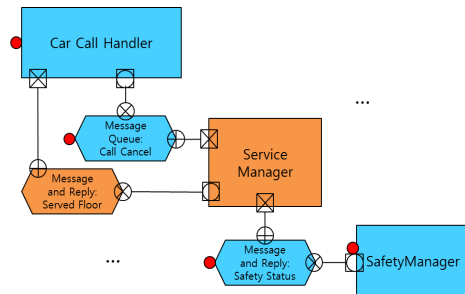


그림 9. Yoon의 표현 방법으로 가변성을 나타낸 ECS 제품라인의 프로세스 아키텍처 모델

그림 9는 Yoon의 표현 방법을 사용하여 ECS 제품라인의 프로세스 아키텍처의 가변성을 나타낸 것이다. 그림에서 Service Manager 컴포넌트와 Message and Reply: Served Floor 커넥터는

제품라인의 모든 제품에 포함되는 필수 컴포넌트/커넥터이고, Car Call Handler 컴포넌트와 Message Queue: Call Cancel 커넥터는 제품에 따라서 선택적으로 포함되는 컴포넌트/커넥터임을 알 수 있다.

Yoon의 방법의 장점은 커넥터의 가변성을 명시적으로 표현한다는 점과, 컴포넌트/커넥터가 필수적인지, 선택적인 가변성인지, 필수적인 가변성인지의 여부를 한 눈에 알아보기 쉽게 색으로써 명확하게 구분한다는 점이다. 하지만 이 방법은 색이 단색으로 표현될 수밖에 없는 경우에는 사용할 수 없고, 보편적이지 않은 아키텍처 모델 표현 방법이며, 이를 지원하는 도구가 없다는 약점이 있다. 또한, 컴포넌트 내부의 가변성을 표현할 수 없고, 가변 컴포넌트/커넥터에 대응되는 휘처가 무엇인지를 아키텍처 모델만 보고서 알 수 없다. 그리고 하나 이상의 커넥터가 택일적으로 선택되는 관계임을 나타낼 수 있으나, 하나 이상의 컴포넌트가 택일적으로 선택되는 관계는 나타내기 어렵다(즉, 어느 컴포넌트가 택일적으로 선택되는 컴포넌트인지는 표현할 수 있지만, 어떤 컴포넌트와 택일적인 관계인지는 표현할 수 없다).

3.6. Jang의 표현 방법

Jang[11]은 UML의 모델 중 컴포넌트 모델만을 확장하여 제품라인 아키텍처 모델을 표현하였다. Jang은 UML의 컴포넌트 모델 중 컴포넌트, 인터페이스, 그리고 컴포넌트 내부 구현을 가변성의 대상으로 하였다.

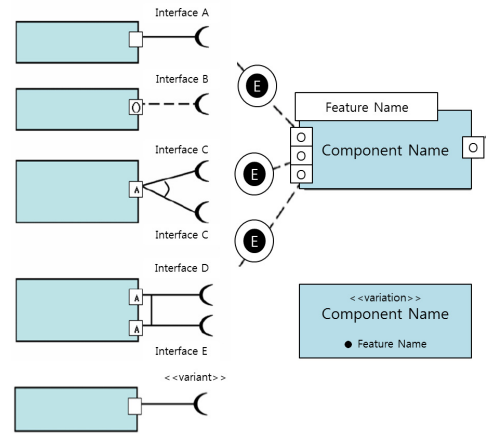


그림 10. Jang의 표현 방법

그림 10은 Jang의 표현 방법을 나타낸 그림이다. 컴포넌트 자체가 제품에 포함되거나 포함되지 않는 가변성은 컴포넌트 위에 해당 컴포넌트와 연관된 가변 휘처의 이름을 적어서 표현한다. 인터페이스가 바뀌는 가변성은 “<<variant>>”로 표현하고, 인터페이스가 선택적 또는 택일적으로 제품에 포함되는 여부를 “O” 또는 “A”로 나타낸다. 제품에 따라서 컴포넌트 내부가 변하는 경우는 컴포넌트에 “<<variation>>” 스테레오타입을 추가하고, “● 대응되는 가변 휘처 이름”을 컴포넌트 하단에 명시한다.

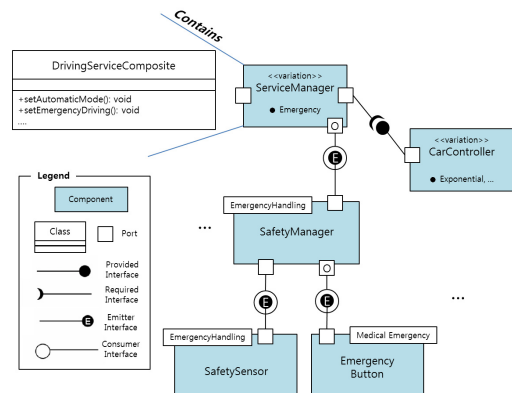


그림 11. Jang의 표현 방법으로 가변성을 나타낸 ECS 제품라인의 프로세스 아키텍처 모델

그림 11은 Jang의 표현 방법을 사용하여 ECS 제품라인의 프로세스 아키텍처의 가변성을 나타낸 것이다. 그림에서 SafetyManager 컴포넌트는 Emergency Handling 휘처에 대응되고, Emergency Button 컴포넌트는 Medical Emergency 휘처에 대응되는 것을 알 수 있다. 그리고 ServiceManager 컴포넌트는 내부의 구현이 Emergency 휘처의 선택에 따라 변하는 것을 알 수 있다(ServiceManager 컴포넌트 내부에 있는 DrivingServiceComposite 클래스의 setEmergencyDriving() 함수는 Emergency 휘처가 선택된 경우에만 제품에 포함된다).

Jang의 표현 방법은 Yoon의 방법과 마찬가지로 커넥터의 선택적/택일적 가변성을 명시적으로 표현한다는 것이 강점이다. 또한 컴포넌트 또는 컴포넌트 내부의 가변성에 대응되는 가변 휘처가 무엇인지 나타낼 수 있으며, 컴포넌트 내부의 가변성도 표현이 가능하다.

하지만 인터페이스의 가변성에 대응되는 가변 휘처를 나타내지 못한다. 그리고 컴포넌트 또는 컴포넌트 내부의 가변성에 대응되는 가변 휘처가 선택 휘처인지 택일 휘처인지 알기가 어렵다. 또한, UML의 표현 방법을 확장하였지만, UML에서 제공하지 않는 표현 방법(예를 들어 컴포넌트 위에 네모 박스로 휘처 이름을 나타내는 것)을 사용하였기 때문에 일반적인 UML 도구를 사용할 수 없다.

3.7. Lee et al.의 표현 방법

Lee et al.[12]은 Gomaa[6, 7], Moon et al. [8], Czarnecki et al.[9]의 표현 방법처럼 UML 스테레오타입을 확장하여 제품라인 아키텍처 모델의 컴포넌트/커넥터의 가변성을 표현하였다. Lee et al.의 표현 방법의 특징은 가변 휘처에 대응되는 컴포넌트/커넥터의 스테레오타입에 휘처의 타입(선택 휘처 또는 택일 휘처), 대응 방식(컴포넌트/커넥터의 전체에 대응 또는 내부 구현에 대응)과 이름을 함께 나타낸다는 점이다.

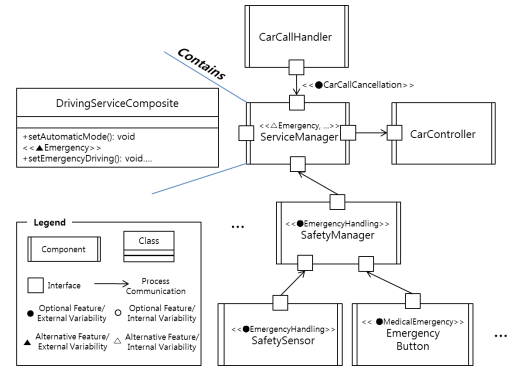


그림 12. Lee et al.의 표현 방법으로 가변성을 나타낸 ECS 제품라인의 프로세스 아키텍처 모델

그림 12는 Lee et al.의 표현 방법으로 ECS 제품라인의 프로세스 아키텍처 모델의 가변성을 나타낸 것이다. Lee et al.의 표현 방법은 휘처가 선택 휘처일 경우는 원을, 가변 휘처일 경우는 삼각형을 사용하는 휘처의 이름 앞에 나타낸다. 그리고 휘처가 컴포넌트/커넥터 전체에 대응되는 경우에는 검은색으로 채워진 도형을, 컴포넌트/커넥터 일부에 대응되는 경우에는 흰색으로 채워진 도형을 사용한다. 그림 12에서 SafetyManager 컴포넌트는 Emergency Handling 휘처에 대응되며, “<<●EmergencyHandling>>”을 통하여 Emergency Handling 휘처가 선택 휘처(도형이 원이기 때문)이고, SafetyManager 컴포넌트 전체에 대응된다는 것(도형이 검은색으로 채워졌기 때문)을 알 수 있다. ServiceManager 컴포넌트는 Emergency 휘처에 대응되며, “<<△Emergency>>”를 통하여 Emergency 휘처가 택일 휘처이고(도형이 삼각형이기 때문), ServiceManager 컴포넌트의 내부 구현에 대응된다(도형이 흰색으로 채워졌기 때문)는 것을 알 수 있다. ServiceManager 컴포넌트 안에 들어있는 DrivingServiceComposite 클래스의 setEmergencyDriving() 함수는 Emergency 휘처가 선택되었을 때 제품에 포함이 된다.

Lee et al.의 표현 방법의 강점은 제품라인 아키텍처 모델만 보고도 각 가변 컴포넌트/커넥터에

대응되는 휘처의 이름과 종류, 대응 방식을 파악할 수 있다는 것이다. 또한 휘처의 종류와 대응 방식을 나타내는 도형의 모양과 색을 사용하여 나타내어 표현 방법이 간결하면서도 쉽게 파악하기가 쉽다.

하지만 Lee et al.의 표현 방법은 표현할 수 있는 정보가 많은 만큼 가변 휘처의 수가 많아지면 아키텍처 모델이 복잡해진다는 약점이 있다. 컴포넌트 내부에 대응되는 휘처를 모두 표시해야 하기 때문에, 추상화가 높은 아키텍처 컴포넌트인 경우 해당 컴포넌트에 대응되는 모든 정제된(Refined) 컴포넌트의 가변성을 내부 가변성으로 표시하면 아키텍처 모델이 복잡해질 수 있다.

4. 소프트웨어 제품라인 아키텍처 모델에서의 가변성 표현 방법 비교

지금까지 살펴본 제품라인 아키텍처 모델에서의 가변성 표현 방법을 요약하여 표로 나타내면 표 1과 같다. 여덟 가지 항목(아키텍처 모델 자체에 가변성 표현, UML 모델 사용/확장, 휘처 이름 표현, 휘처 타입 표현, 내부가변성 표현, 도구 지원, 휘처 선택 관계 표현, 표현할 수 있는 가변성 범위)에 대하여 각 표현 방법을 비교하였다.

독립적인 가변성 모델을 관리하는 Thiel et al.의 표현 방법을 제외하고는 모두 아키텍처 모델 자체에 가변성을 표현할 수 있었다. Thiel et al.과 Yoon의 표현 방법은 독자적인 아키텍처 모델을 사용하였다. UML의 스테레오타입을 확장한 표현 방법들(Gomaa, Moon et al., Czarnecki et al., Lee et al.의 표현 방법)은 기존 UML 도구를 사용하여 제품라인 아키텍처에 가변성을 표현할 수 있었다. Czarnecki et al., Jang, Lee et al.의 표현 방법은 가변 컴포넌트/커넥터에 대응되는 휘처의 이름을 표현할 수 있었고, Gomaa, Yoon,

Jang, Lee et al.의 표현 방법은 가변 컴포넌트/커넥터에 대응되는 휘처의 타입을 표현할 수 있었다. Gomaa, Jang, Lee et al.의 표현 방법은 컴포넌트/커넥터의 내부 가변성을 표현할 수 있었고, Czarnecki et al.의 표현 방법은 휘처 선택 관계를 논리 연산으로 표현하여 이를 가변 컴포넌트/커넥터와 대응시킬 수 있었다. UML의 스테레오 타입을 확장한 Gomaa, Moon et al., Czarnecki et al., Lee et al.의 표현 방법은 UML 모든 단위에 가변성을 표현할 수 있었고, Thiel et al.은 각 아키텍처의 컴포넌트에, Yoon은 각 아키텍처의 컴포넌트/커넥터에, Jang은 컴포넌트 아키텍처의 컴포넌트/인터페이스에 가변성을 표현할 수 있었다.

5. 결론 및 향후 연구

본 논문에서는 현재까지 제안된 제품라인 아키텍처 모델에서의 가변성 표현 방법을 분석하고 엘리베이터 컨트롤 시스템 제품라인 예제에 적용하여 강약점을 파악하였다. 또한 분석 결과를 정리하여 각 표현 방법의 특징을 비교하였다. 비교 결과 각 표현 방법마다 표현할 수 있는 정보가 달랐다. 표현할 수 있는 정보의 범위가 작으면 아키텍처 모델이 간결했지만 아키텍처 모델을 통해 파악할 수 있는 정보가 적었고, 범위가 넓으면 아키텍처 모델이 복잡해졌다. 이를 해결하기 위해 어떤 표현 방법은 색깔이나 도형을 사용하여 많은 정보를 표현하면서 복잡성을 줄이려고 노력하였다. 많은 표현 방법이 UML 모델을 확장하였으며, UML 모델을 확장한 경우 독자적인 아키텍처 모델을 제시한 경우보다 활용도가 높고 도구 지원이 가능하다는 것이 강점이었다.

각 제품라인 아키텍처 모델에서의 가변성 표현 방법마다 강약점과 표현 범위가 다르기 때문에

표 1 제품라인 아키텍처 모델에서의 가변성 표현 방법 비교표

	독립적 가변성 모델 이용 [3, 4]	Gomaa [6, 7]	Moon et al. [8]	Czarnecki et al.[9]	Yoon[10]	Jang[11]	Lee et al.[12]
아키텍처 모델 자체에 가변성 표현	X	O	O	O	O	O	O
UML 모델 사용/확장	X	O	O	O	X	O	O
휘처 이름 표현	X	X	X	O	X	O	O
휘처 타입 (선택/택일) 표현	X	O	X	X	O	O	O
내부 가변성 표현	X	O	X	X	X	O	O
도구 지원	X	O	O	O	X	X	O
휘처 선택 관계 표현	X	X	X	O	X	X	X
표현할 수 있는 가변성 단위	각 아키텍처의 컴포넌트	UML 모든 단위	UML 모든 단위	UML 모든 단위	각 아키텍처의 컴포넌트/ 커넥터	컴포넌트 아키텍처의 컴포넌트/ 인터페이스	UML 모든 단위

사용하려는 목적에 적합한 표현 방법을 선택하는 것이 중요할 것이다. 하지만 휘처 기반의 제품라인에서는 휘처와 제품라인 아키텍처 자산 사이의 쉬운 추적성을 위해 휘처 이름 표현과 휘처 타입 표현이 필요할 것으로 생각된다. 또한 가변성이 있을 수 있는 컴포넌트/커넥터 전체/내부의 가변성을 모두 표현할 수 있는 표현 방법이 적합할 것이다. 또한 표현 방법을 실제로 적용하고 사용하기 위해 도구(UML 도구 또는 그 외 독자적인 도구) 지원이 가능한 표현 방법을 택하는 것이 필요하다.

본 논문에서는 제품라인 아키텍처 모델에서의 가변성 표현 방법의 표현력에 초점을 맞추어 표현 방법을 비교하였지만, 제품라인 아키텍처 모델에 가변성을 언제 바인딩(Binding)할 것인가에 대한 결정 또한 중요하다. 가변성을 제품라인 아키텍처 모델에 일찍 바인딩하여 가변성을 아키텍처 모델 안에서 관리할 경우, 가변성 휘처가 바뀔 때마다 아키텍처 모델이 계속 바뀌어야 하고

표현된 가변성의 일관성을 검증하여야 한다. 제품라인 아키텍처 모델과 가변성을 따로 관리하고 가변성을 아키텍처 모델에 늦게 바인딩 할수록, 제품라인 아키텍처는 가변성의 변화에 유연해진다. 향후에는 제품라인 아키텍처 모델에 가변성을 늦게 바인딩하여 유연한 제품라인 아키텍처를 갖는 방법을 연구할 것이다.

참 고 문 헌

- [1] V. Sugumaran, S. Park, and K.C. Kang, "SPECIAL ISSUE: Software Product Line Engineering," Communications of the ACM(CACM), Vol. 49, Issue. 12, pp.28-32, December 2006.
- [2] F. Van der Linden, K. Schmid, and E. Rommes, "Software product lines in action: the best industrial practice in product line Engineering," Springer-Verlag New York Inc., 2007.

- [3] S. Thiel and A. Hein, "Systematic Integration of Variability into Product Line Architecture Design," 2nd International Software Product Line Conference (SPLC 2002), LNCS 2379, pp. 130-153, 2002.
- [4] K. Pohl, G. Böckle, and F. van der Linden, "Software Product Line Engineering: Foundations, Principles, and Techniques," Springer-Verlag New York Inc., 2005.
- [5] H. Gomaa, "A Software Design Method for Real-time Systems," Communications of the ACM, Volume 27, Issue 9, Sept. 1984.
- [6] H. Gomaa, "Designing Software Product Lines with UML, From Use Cases to Pattern Based Software Architectures," Addison-Wesley, 2004.
- [7] H. Gomaa, "Designing Software Product Lines with UML," Addison Wesley, 2005.
- [8] M. Moon, H.S. Chae, and K. Yeom, "A Metamodel Approach to Architecture Variability in a Product Line," International Conference on Software Reuse (ICSR 2006), LNCS 4039, pp. 115-126, 2006.
- [9] K. Czarnecki and K. Pietroszek, "Verifying Feature-Based Model Templates Against Well-Formedness OCL Constraints," 5th International Conference on Generative Programming and Component Engineering (GPCE 2006), October 22-26, 2006, Portland, Oregon, USA.
- [10] C. Yoon, "Verification of Syntactic Consistency between Feature Model and Product line Architecture Model (휘쳐 모델과 프로덕트 라인 아키텍처 모델 간의 문법적 일관성 검증)," Master Thesis, 2003.
- [11] Y. Jang, "Component architecture specification method for software product line engineering (소프트웨어 프로덕트 라인 공학에서의 컴포넌트 아키텍처 명세 방법)," Master Thesis, 2007.
- [12] H. Lee, H. Choi, K.C. Kang, D. Kim, and Z. Lee, "Experience Report on Using a Domain Model-Based Extractive Approach to Software Product Line Asset Development," In: Proceedings of the 11th International Conference on Software Reuse (ICSR 2009), 2009, pp. 149-161.

저 자 소 개



이 혜 선

2009년 포항공과대학교 컴퓨터공학과 졸업(학사).
2009년~현재 포항공과대학교 컴퓨터공학과 대학원
통합과정.

<관심분야> 소프트웨어 재사용, 소프트웨어 제품
라인 공학, 추출식 제품라인 접근,
휘쳐 기반 소프트웨어 개발 방법



조 성 배

2010년 동국대학교 컴퓨터공학과 졸업(학사)

2010년~현재 포항공과대학교 정보전자융합공학부
석사과정.

<관심분야> 소프트웨어 재사용, 소프트웨어 제품
라인 공학, 추출식 제품라인 접근



강 교 철

1973년 고려대학교 통계학과 졸업(학사).
1974년 콜로라도대학교 산업공학 졸업(석사).
1982년 미시간대학교 소프트웨어공학 졸업(박사).
1982년~1984년 미시간대학교 Visiting Professor.
1984년~1985년 (미국)벨 통신 연구소 Technical Staff.
1985년~1987년 AT&T 벨 연구소 Technical Staff.
1987년~1992년 카네기멜론대학교 Project Leader.
1992년~현재 포항공과대학교 교수.
2000년~2001년 카네기멜론대학교 Visiting Scientist.

<관심분야> 소프트웨어공학, 실시간 내장형
시스템, 소프트웨어 재사용, 소프트
웨어 제품라인 공학.

SBA 효과도 분석을 위한 국방 모의 실험 사례분석[†]

(A Case Study on Military Modeling and Simulation for SBA's Effectiveness Estimation)

최달님[‡]

김형종[§]

(Dal-Nim Choi)

(Hyung-Jong Kim)

요 약 국가 방위를 위한 무기체계의 획득은 다른 분야와 비교할 때 상대적으로 비용이 과다히 소모되고, 획득자원의 시험과정 등이 매우 위험하며 획득결과의 품질이 매우 중요하다. 이러한 무기체계 획득의 특성으로 인해서 모의실험이 필수적으로 이루어지고 있으며, 이를 시뮬레이션 기반 획득(SBA, Simulation Based Acquisition) 이라고 한다. 그런데, 모의실험을 위한 모델개발 및 이의 실행분석이 갖는 비용이 무시할 수 없는 수준이기 때문에 모의실험이 갖는 효과도에 대한 분석에 대한 연구필요성이 제기되어 왔다. 본 연구에서는 이러한 효과도 분석에 요구되는 4가지의 지표를 제시하고, 이 지표들이 기존 연구에 어떻게 적용될 수 있는지에 대한 사례분석을 수행하였다. 비록 기존 연구들이 이러한 지표를 고려하지 않았지만, 본 논문에서 명시하고 있는 효과도에 대한 고찰을 통해 모의실험의 국방 획득 과정에서의 필요성을 확인 할 수 있다.

키워드 무기체계 획득, SBA(Simulation Based Acquisition) 효과도, 모델링 및 시뮬레이션(M&S)

Abstract The weapon system acquisition for national defense is too costly, risky and requires high quality result. Because of these characteristics of the weapon system acquisition, the modeling and simulation is a prerequisite process for enhancing the effectiveness and efficiency of weapon system acquisition. We call the process as SBA (Simulation Based Acquisition). However the modeling and simulation entails costs of model development, execution and analysis. Thus, the SBA's effectiveness analysis is needed. Especially, we developed 4 types of index which represent the effectiveness thoroughly and we applied them to various weapon systems' acquisition process. This work presents the necessity of SBA by showing the application of suggested effectiveness index in various defense weapon system acquisition cases.

Key words Software reuse, Software Reuse-related Knowledge, Web 2.0-based Software Engineering

1. 서 론

SBA는 국방 분야에서의 무기체계 획득과 관련하여 소요제기부터 체계운용까지 전 수명주기에 걸쳐 M&S 기술을 적용하는 새로운 획득 패러다임이다. SBA는 시간 및 자원을 절약하고, 무기체계 획득 관련 위험을 감소시키며, 최적의 군사 유틸

리티 품질을 획득하는 것이 목적이다[1]. 즉, SBA 도입을 통하여 군의 획득 업무를 보다 신속하고(faster) 저렴하며(cheaper) 뛰어나게(better) 처리하고자 하며[2], 이러한 목적은 미래의 군 획득업무 발전에 큰 영향을 미칠 것으로 전망된다. 미 국방부를 비롯한 여러 국가에서는 국방 획득 업무의 발전을 위한 SBA 연구를 활발히 진행 중이며, 최근 국내에서도 국방 분야의 SBA 도입 관련 연구에 주목하여 국방 무기체계 획득 발전을 위해 노력하고 있다. 그러나 SBA 도입의 효과를 증명하기 위해 구체적으로 제안된 SBA 효과도 입증 방안은 현재 존재하지 않아, 도입 효과도에 대한 과학적 입증이 불가능한 실정이다. 또한

[†] 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.

[‡] 비 회 원 : 서울여자대학교 컴퓨터학과
dnchoi@swu.ac.kr

[§] 비 회 원 : 서울여자대학교 정보보호학과
hkim@swu.ac.kr

논문접수 : 2011년 08월 17일

심사완료 : 2011년 09월 16일

SBA 적용 사례 간 비교 분석의 경우, 사례 간 유사성 도출이 어려워 상호 비교를 통한 SBA 도입 효과도 분석은 힘든 것이 사실이다. 따라서 객관적이며 과학적으로 SBA 효과도를 입증하기 위한 연구가 진행되어야 하며, 본 논문에서는 효과도 입증 방안 중 하나로서 객관적 지표에 의한 정량적인 효과도 분석 방법을 제안하고자 한다.

객관적 기준에 근거하지 않은 SBA 효과도는 도입 효과에 대한 신뢰도를 감소시킨다. SBA의 기대 효과 중 하나인 비용 절감의 경우, SBA 도입을 통해 항공모함의 설계비용을 최초 요구 비용보다 5만 불 절감하였다고 가정하였을 때, 단순히 표면적으로 절감된 5만 불은 비용 절감 효과를 증명하기 위한 근거 자료로서의 가치를 지니지 못한다. 절감 비용 5만 불은 항공모함의 최초 설계 요구비용에 따라 비용절감 효과도가 달라지기 때문이다. 항공모함의 최초 설계 요구 비용이 각각 10만 불과 1000만 불인 경우, 절감 비용 5만 불이 갖는 비용절감 효과도는 동일한 가치를 지니지 않으며, 5만 불이 비용 절감 효과로서 갖는 신뢰도는 상당히 떨어진다. 따라서 기준에 근거한 정보 입력을 바탕으로 환경에 제약을 받지 않는 과학적이고 체계적인 SBA 효과도 입증 방안 연구가 필요하다.

본 논문에서는 국방 분야의 SBA 도입 목적 및 SBA 실적용 사례를 조사하고, 이를 바탕으로 SBA 효과도 입증 방안 제안에 앞서 효과도 입증 위한 정량적 지표를 정의한다. 또한 간단한 효과도 입증 사례를 구성하여 제안된 지표 및 효과도 입증 방법을 적용한다. 정의된 지표는 SBA 효과도의 객관적이며 과학적인 입증을 뒷받침하는 근거가 될 뿐 아니라, SBA 효과도가 갖는 신뢰도를 향상시키는 근거로 활용 가능할 것이다.

2장에서는 SBA의 도입 목적 및 적용 사례에 대한 조사를 수행한다. 3장에서는 2장을 바탕으로 SBA 효과도를 과학적으로 입증하기 위한 입증

지표 및 입증 방법을 제안하고, SBA 적용 사례를 구성하여 제안된 SBA 효과도 입증 지표를 통한 SBA 효과도 입증 사례를 제안한다. 4장에서는 결론 및 향후 연구 발전을 전망한다.

2. 관련연구

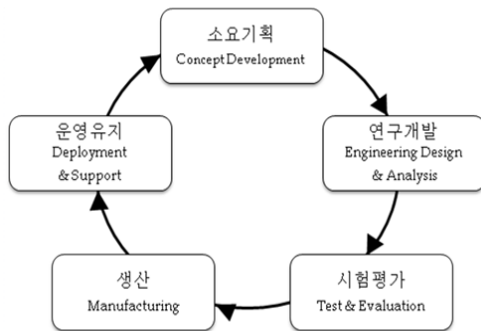
SBA는 국방자원의 효과적인 획득 관리를 목표로 자원의 획득비용 절감, 획득기간 단축, 시행착오 최소화를 목표로 하며, 이를 위해 M&S를 기반으로 무기체계 프로세스 전반에 걸쳐 가상 환경에서의 무기체계 프로세스 설계 운영 및 검증을 수행한다. M&S는 SBA 핵심 기술로서, 현실 세계의 복잡한 구조를 추상화하여 간단한 모형으로 표현하는 모델링 방법과 가상 환경에서의 모의실험을 통하여 해당 구조의 특성을 파악하는 시뮬레이션 방법이 결합된 개념이다[5]. SBA 도입은 최적의 군사 유틸리티 품질 획득에 상당한 영향을 미칠 것으로 기대된다. 그러나 객관적 기준에 근거하지 않은 SBA 효과도는 타당한 신뢰도를 갖지 못하며, SBA 기대 효과를 뒷받침하기 위한 효과도 입증 방안조차 존재하지 않아, 도입 효과도에 대한 과학적 입증이 불가능한 실정이다. 이러한 문제점을 해결하기 위하여 효과도 입증을 위한 방안으로서, 정량적 지표에 의한 효과도 분석 방안을 제안하고자 하며, 방안 제안에 앞서 효과도 분석의 객관적인 기준으로 작용 될 정량적 지표 정의가 선행되어야 한다.

2.1에서는 지표 정의에 앞서 SBA 관련 연구 분석을 통해 SBA 획득절차에 따른 지표의 속성을 분석한다. 미군의 경우, 효과적인 SBA 운영을 위하여 JMASS(Joint Modeling and Simulation System) 환경을 개발하여 활용 중이나, 국내 국방 분야에서의 SBA 활용은 아직 초기 연구 단계이다. 따라서 2.2에서는 국외 SBA 적용 사례를 조사

하여 SBA 도입을 통해 발생한 실제 효과를 분석한다.

2.1 SBA 획득절차에 따른 지표 분석

미 국방부는 SBA 도입의 당위성을 설명하기 위하여 1996년 M&S 및 SBA 도입 관련 보고서를 제시한 바 있다[3].



<그림 1> SBA의 획득체계 절차

SBA 획득절차는 <그림 1>과 같이 총 5단계로 분류되며, 각 단계는 소요 기획부터 운용유지까지 순차적으로 발생한다.

소요 기획이란 소요 요청 및 소요 제기, 소요 결정 단계를 포괄하여 국방 획득체계에 요구되는 조건을 기획 정립하는 단계이다. 연구개발 단계는 소요 기획을 통해 필요성이 제기된 국방 무기 체계를 생산 획득하며, 시험 평가 단계를 통해 요구 성능에 대한 기술적 도달 정도를 평가한다[4]. SBA 도입에 대한 효과 분석은 각 단계별 특징을 고려하여 진행되어야 한다.

[3]에서는 군의 획득 업무를 보다 신속하고(faster) 저렴하며(cheaper) 뛰어나게(better) 처리하고자 하는 SBA 목적에 근거하여 시간(Time), 비용(Cost), 품질(Quality)을 기준으로 5개의 획득 절차 분석을 제안하고 있다.

2.2 SBA 적용 사례

SBA 도입에 따른 적용 사례를 분석한 결과 [3][6], 총 30건의 SBA 적용 사례가 조사되었다.

비용 절감 사례는 SBA 적용 사례 중 가장 많은 비중을 차지하였으며, 총 17건으로 다음과 같다.

- 미 DARPA(방위고등연구계획국): 레이더 경고 시스템 설계 단계의 M/M을 기존의 96에서 46으로 감소
- 미 TARDEC(국방부 산하 전차연구개발센터): BFV (Bradley Fighting Vehicle) 엔진 교체 분석 기술을 통해 M&S를 도입하지 않았을 경우보다 M/M을 절반으로 감소
- 미 TARDEC: 탱크 프로토타입 설계에 요구되는 소요 인력을 기존의 55명에서 14명으로 감소
- 보잉사(The Boeing Company): 보잉 767 지주 설계 과정에서 이전보다 약 30,000 공수(Man-Hours)감소
- 미 해군: 강습 상륙함 LPD-17의 설계비용(약 600만 달러) 절약
- 미 공군 AEDC(아놀드 기술개발센터): 각종 미사일의 특성 조사를 위한 M&S 기반 테스트는 미사일 당 70만 불이 소요되는 실 시험을 1만 불 이하의 비용으로 진행함
- 미국 중거리 공대공 미사일 AMRAAM(Advanced Medium Range Air-to-Air Missile): 실 환경에서의 성능 시험 시 미사일 당 약 3백만 불의 고비용이 요구되나, M&S를 통해 동일 비용으로 약 16,000회의 미사일 소프트웨어 및 하드웨어 시험완료.
- 미 공군 AEDC: 1970년 Lockheed Aircraft Corp 사가 개발한 민간수송기 L-1011의 로켓부스터 분리 예측 과정에서 M&S를 통하여 75만 불 절약
- 미 해군: 실 환경에서 1회 발사 당 약 6만 달러가 소요되는 어뢰시험이 M&S 기반의 동일한 비용 조건에서 약 200회의 시험이 가능하다고 보고
- 미 공군: APG-63 레이더 시험 사업을 통하여 약 35%의 비용 절감 성과 달성
- 미국 NAWCAD: 미 해군의 공격기 EA-6B에 탑재되는 ALQ-99 수신기와 ALQ-149 통신장비의 가상 시험을 통해 개발 비용 감소

- 미 공군: F-16 사업은 1990년부터 AFFTC(the Air Force Flight Test Center)의 M&S를 통해 비행 시험을 실시하여 약 4천만 불의 비용 절감
- 미 공군: M&S 기반의 F-15 전투기의 레이더 시험을 통해 약 30%의 비용 절감
- 미국 Comanche 사업: CATIA(the Computer Aided Three-Dimensional Interactive Simulation)를 통해 요구 인력을 19M/Y 에서 1M/M로 절감
- 미 전투공격기 JSF(Joint Strike Fighter) 관련 사업: 수명주기 비용의 3%인 50억 불을 절약
- 보잉사: 보잉 777 생산 과정에 M&S를 도입하여 요구 인력을 40명에서 2명으로 감소.
- 미군: AMRAAM 개발을 위하여 650만 불을 투자하여 M&S 기술을 활용한 결과, 25,000만 불을 절

조사된 기간 단축 사례는 총 6건으로 다음과 같다.

- 미 NAVSEA(해군무기체계사령부): 1994년 M&S 기술을 도입하여 선박 설계과정에서 약 27일이 소요되던 내항성(Seakeeping) 분석을 약 3일로 단축
- 미 NAVSEA: RCS(레이더 단면적) 분석 및 선박 설계를 위해 57일을 소요하였으나, 1990년 M&S 기술을 적용하여 동일한 분석을 약 16일 만에 완료
- 뉴포트 조선소(Newport News Shipbuilding): M&S 기술을 적용하여 약 40%의 설계 시간 단축 및 25%의 세공 시간 단축
- 보잉사: 보잉 767 지주 설계 소요시간을 17% 단축
- 미 국 NAWCAD(Naval Air Warfare Center Aircraft Division): 공격기 EA-6B에 탑재되는 ALQ-99 수신기와 QALQ-149 통신장비의 가상 시험을 통해 개발 시간을 단축
- 미 TARDEC: 탱크 프로토타입 설계 기간을 3년에서 16개월로 단축

기타 적용 사례는 성능 향상 및 정보 수집 비용 향상 등을 포함하며, 총 7건으로 다음과 같다.

- 보잉사: 보잉 767 지주 설계 변경 확률을 65% 감소
- 미 해군: 강습 상륙함 LPD-17은 M&S 기술을 통하여 수선 상부의 갑판 무게를 약 100톤 감소
- 미 공군: APG-63 레이더 시험 사업은 M&S를 적용하여 약 300%의 데이터 수집 달성
- 미국 Comanche 사업: CATIA(the Computer Aided Three-Dimensional Interactive Simulation)를 통해 이전의 35%에 비해 더 높은 95%의 성과 비율 달성
- 보잉사: 보잉 777 생산 과정에 M&S를 적용하여 기존의 보잉 747과 비교하여 스크랩을 30% 감소. 또한 재작업 횟수의 경우, 보잉 747이 30%인 반면 보잉 777은 3%로 감소
- 미 공군: F-15 전투기의 레이더 시험을 통해 약 300%의 정보 수집 비율 달성
- 미 해군: Seawolf급 잠수함의 약 95,000개였던 표준 부품 목록을 원자력 잠수함 NSSN 개발을 통해 16,000개 항목으로 감소

3. SBA 효과도 입증 지표 도출

지표를 통한 효과도 입증 방식은 SBA 목적에 부합하는 객관적인 평가 기준을 제시함으로써, SBA 효과도에 대한 신뢰도를 향상시킨다. SBA 효과도 입증은 객관적 지표에 의한 정량적 시간 및 비용 절감 효과뿐만 아니라, 획득 과정을 통해 발생 가능한 성능 향상 등의 다양한 요소를 고려할 필요가 있다. 2.1에서 연구된 내용을 바탕으로 SBA 도입 목적 및 기존 관련 연구, SBA 적용 사례 연구 관점에서 지표를 도출 할 때, SBA 효과도를 입증하기 위한 지표는 다음과 같이 네 가지로 도출된다.

- 비용(Cost)
- 시간(Time)
- 품질(Quality)
- 위험도(Risk)

위의 네 가지 요소는 SBA 효과도를 입증하기 위하여 반드시 고려되어야 하는 지표이다.

비용 지표는 SBA 목적 중 하나인 최소비용 도출을 위해 평가되어야 하는 기준 지표이며, 실제 적용 사례 중 가장 많은 비중을 차지하는 SBA 효과도 입증 지표이다. 비용 관점에서의 SBA 효과도 입증은 무기체계 획득을 위한 최적의 비용 도출을 통해 SBA의 비용 절감 효과를 평가한다.

비용 지표와 마찬가지로 시간 지표는 SBA 목적 중 하나인 소요시간 절감을 위해 반드시 평가되어야 할 입증 지표이며, 실제 적용 사례에 대해 비용 지표 다음으로 많은 비중을 차지하였다. 특히 무기체계 획득을 위해 최단 시간을 소요하는 것은 곧 동일한 시간 내 다량의 무기체계 획득에 대한 높은 가능성을 의미한다. 특히 시간 지표는 전시 대비 국방력 강화 및 전시 상황에서의 신속한 대처를 위해 반드시 고려되어야 할 요소이다.

무기체계 획득 과정에서의 가장 큰 문제점은 품질평가 및 무기체계 성능 시험을 위한 환경, 소요 인력의 확보 어려움, 예측 불가능한 위험 요소의 발생 가능성이다. 품질 지표는 획득 목적에 맞는 높은 품질의 무기체계를 획득하기 위해 반드시 고려되어야 하며, 위험 지표는 예측 불가능한 위험 요소를 사전에 탐지 및 제거하여 무기체계 획득의 효율성을 높이기 위해 고려되어야 한다.

SBA 효과도 입증 지표는 세분화 될 수 있으며, 추가적인 지표 도출이 가능하다. 또한 네 가지 지표가 결합된 SBA 효과도는 한 가지 지표를 바탕으로 도출된 SBA 효과도보다 더 높은 신뢰도를 가질 것으로 예상된다. 각 지표에 대한 효과도 입증 방법은 다음과 같다.

3.1 비용 지표의 효과도 입증 방법

비용 측면에서의 SBA 도입 효과도는 같은 품질의 결과물을 얻는데 사용된 SBA 도입 기반 획득 비용($Cost_{acq_p_sba}$)과 SBA를 도입하지 않고

획득한 비용($Cost_{acq_n_sba}$) 간의 차를 통하여 평가되며, SBA 기반 획득 비용은 SBA를 수행하는데 소요되는 비용의 총 합($Cost_{sba}$)과 SBA를 통해 예상된 실제 획득 비용($Cost_{acq}$)의 합으로 도출된다. 또한 SBA를 도입하지 않고 획득한 비용은 과거 획득 비용 정보를 기반으로 도출한다. 획득 비용은 인력, 물리적 도구 등 다양한 요소의 요구비용에 대한 총 합으로 도출 가능하며, SBA는 M&S를 통하여 각 요소에 대한 최적의 요구비용을 도출한다. 정량적인 수치 도출에 앞서 SBA 도입의 효과 여부를 판단할 수 있는 기준이 제시되어야 하며, 비용 관점에서의 SBA 도입 효과도는 <표 2>의 기준에 의하여 판단된다.

<표 2> 비용 측면에서의 SBA 효과도

SBA 효과도	예상 획득 비용 간의 차
Positive	$Cost_{acq_n_sba} - Cost_{acq_p_sba} > 0$
Negative	$Cost_{acq_n_sba} - Cost_{acq_p_sba} < 0$

<표 2>에 따라, 비용 관점에서의 SBA 도입 효과도는 SBA를 도입하지 않고 획득한 비용이 SBA 기반 획득 비용보다 클 경우에 효과적으로 나타나며, 이는 SBA를 통한 무기체계의 모의 성능시험이 실제 획득 과정을 위한 소요비용을 절감 할 것으로 예상되기 때문이다.

예를 들어, 이론적으로 37°의 각도로 발사한 경우에만 10km 간격의 목표물을 명중 가능한 미사일의 성능을 실제로 측정을 위해 시험을 실시할 때, <표 3>에 제시한 기준 비용에 의해 다음과 같이 비용에 대한 두 사례의 SBA 도입 필요성이 평가 된다.

<표 3> 미사일 성능 시험을 위한 기준 비용 예시

$Cost_{acq_n_sba}$	$Cost_{sba}$	$Cost_{acq}$
\$ 400만	\$ 70만	\$ 220만

사례 1) SBA를 통해 1회의 실제 획득 과정만으로 명중을 위한 각도(37°) 측정에 성공한 반면, SBA를 도입하지 않고 1회의 실제 획득 과정을 통해 최고의 성능 측정에 성공한 경우

사례 2) SBA를 통해 4회의 실제 획득 과정만으로 명중을 위한 각도(37°) 측정에 성공한 반면, SBA를 도입하지 않고 2회의 실제 획득 과정을 통해 최고의 성능 측정에 성공한 경우

사례 1의 경우, SBA 기반 획득 비용은 SBA 도입 비용 70만 불과 1회의 실제 획득 과정 비용 220만 불에 의해 총 290만 불의 소요비용이 측정된 반면, SBA를 도입하지 않고 단 1회의 실제 획득 과정을 통해 성능 측정에 성공한 경우 비용은 400만 불로 측정되었다. 사례 1의 결과는 <표 2>에 따라 SBA 도입이 비용 절감에 효과적일 것으로 판단된다. 사례 2는 SBA 기반 획득 비용의 경우 SBA 도입 비용 70만 불과 4회의 실제 획득 과정 비용 880만 불에 의해 총 950만 불이 도출된 반면, SBA를 도입하지 않고 단 2회의 실제 획득 과정을 통해 성능 측정에 성공한 경우의 비용은 800만 불로 도출되었다. 사례 2의 결과는 <표 2>에 따라 SBA 도입에 대한 비용절감 효과도가 매우 낮을 것으로 예상된다.

3.2 시간 지표의 효과도 입증 방법

소요시간 관점에서의 SBA 도입 효과도는 비용 지표와 마찬가지로 같은 품질의 결과물을 얻는데 사용된 SBA도입 기반 획득 시간($Time_{acq_p_sba}$)과 SBA를 도입하지 않고 획득한 시간($Time_{acq_n_sba}$) 간의 차를 통하여 평가되며, SBA 기반 획득 시간은 SBA를 수행하는데 소요되는 시간의 총 합($Time_{sba}$)과 SBA를 통해 예상된 실제 획득 시간($Time_{acq}$)의 합으로 도출된다. 또한 SBA를 도입하지 않고 요구된 소요시간은 과거 획득에 소요된

시간 정보를 기반으로 도출한다. 획득 시간은 설계, 개발, 시험 등 다양한 요소를 통해 소요되는 시간의 총 합으로 도출 가능하다. 정량적인 수치 도출에 앞서 SBA 도입의 효과 여부를 판단할 수 있는 기준이 제시되어야 하며, 시간 관점에서의 SBA 도입 효과도는 <표 4>의 기준에 의하여 판단된다.

<표 4> 시간 측면에서의 SBA 효과도

SBA 효과도	예상 획득 시간 간의 차
Positive	$Time_{acq_n_sba} - Time_{acq_p_sba} > 0$
Negative	$Time_{acq_n_sba} - Time_{acq_p_sba} < 0$

<표 4>에 따라, 시간 관점에서의 SBA 도입 효과도는 SBA를 도입하지 않고 도출된 소요시간이 SBA 기반 소요시간보다 클 경우에 효과적으로 나타나며, 이는 SBA를 통한 무기체계의 모의 성능시험이 실제 획득 과정을 위한 소요시간을 감소시킬 것으로 예상되기 때문이다.

예를 들어, 이론적으로 37° 의 각도로 발사한 경우에만 10km 간격의 목표물을 명중 가능한 미사일의 성능을 실제로 측정을 위해 시험을 실시할 때, <표 5>에 제시한 기준 비용에 의해 다음과 같이 비용에 대한 두 사례의 SBA 도입 필요성이 평가된다.

<표 5> 미사일 성능 시험을 위한 기준 시간 예시

$Time_{acq_n_sba}$	$Time_{sba}$	$Time_{acq}$
35시간	2시간	19시간

사례 1) SBA를 통해 1회의 실제 획득 과정만으로 명중을 위한 각도(37°) 측정에 성공한 반면, SBA를 도입하지 않고 1회의 실제 획득 과정을 통해 최고의 성능 측정에 성공한 경우

사례 2) SBA를 통해 4회의 실제 획득 과정만으로 명중을 위한 각도(37°) 측정에 성공한 반면, SBA를 도입하지 않고 2회의 실제

획득 과정을 통해 최고의 성능 측정에 성공한 경우

사례 1의 경우, SBA 기반 획득 시간은 SBA 도입에 소요된 2시간과 1회의 실제 획득 과정 소요시간인 19시간에 의해 총 21시간의 소요시간이 도출된 반면, SBA를 도입하지 않고 단 1회의 실제 획득 과정을 통해 성능 측정에 성공한 경우의 소요시간은 35시간으로 예상되었다. 사례 1의 결과는 <표 4>에 따라 SBA 도입이 소요시간 절감에 효과적일 것으로 판단된다. 사례 2는 SBA 기반 획득 비용의 경우 SBA 도입에 소요된 2시간과 4회의 실제 획득 과정에 소요된 76시간에 의해 총 78시간이 도출된 반면, SBA를 도입하지 않고 단 2회의 실제 획득 과정을 통해 성능 측정에 성공한 경우의 소요시간은 70시간으로 예상되었다. 사례 2의 결과는 <표 4>에 따라 소요시간 절감에 대한 SBA 도입 효과도가 매우 낮을 것으로 예상된다.

3.4 품질 지표의 효과도 입증 방법

높은 품질의 무기체계 획득을 위한 SBA 도입 효과도는 동일한 기준에 근거하여 SBA 기반 획득 품질($Quality_{acq_p_sba}$)과 SBA를 도입하지 않은 경우의 획득 품질($Quality_{acq_n_sba}$) 간 비교를 통해 평가한다. 이 때, SBA 효과도를 입증하기 위한 기준은 무기체계 획득 목적에 따라 다르게 정의한다. SBA 기반 획득은 SBA를 통해 얻어진 정보를 활용하여 기준을 평가하며, SBA를 도입하지 않은 경우의 획득은 과거 획득 과정을 통해 도출된 정보를 기반으로 도출한다.

예를 들어, 아래와 같이 두 가지의 사례에 대한 SBA 도입 효과도를 분석하고자 할 때, 품질 관점에서의 효과도 입증에 위한 입증 기준은 각각의 무기체계 획득 목적에 따라 다르게 정의된다.

- 사례 1) 2kg 이하의 무게를 가진 소총 개발
- 사례 2) 5000km의 사정거리를 가진 미사일 개발

소총 개발을 목적으로 하는 사례 1은 무기체계 획득 목적에 근거하여 SBA 효과도 입증 기준 요소로서 소총 무게, 탄알비행속도, 예상 평균 사용 연수가 정의된다. 또한 미사일 개발을 목적으로 하는 사례 2는 SBA 효과도 입증 기준 요소로서 발사 중량, 사거리, 예상 평균 사용 연수가 정의된다. 위의 기준에 근거하여 사례 1과 사례 2에 대한 품질 관점에서의 SBA 효과도 분석 결과는 각각 <표 6>, <표 7>과 같다.

<표 6> 사례 1에 대한 품질 지표 분석 결과

기준	$Quality_{acq_p_sba}$	$Quality_{acq_n_sba}$
소총 무게	1.2kg	1.3kg
탄알비행속도	1200m/s	1600m/s
평균 사용연수	5년	10년

사례 1의 경우, SBA 미 도입 시 획득된 소총 품질은 SBA 기반 획득 품질과 비교하여 세 개의 기준 요소 중 탄알비행속도, 평균 사용 연수의 두 가지 요소가 뛰어나, SBA 도입 효과가 매우 낮을 것으로 추정된다.

<표 7> 사례 2에 대한 품질 지표 분석 결과

기준	$Quality_{acq_p_sba}$	$Quality_{acq_n_sba}$
발사 중량	20,000kg	20,000kg
사거리	3200km	1600km
평균 사용연수	10년	5년

반면 사례 2의 경우, SBA 기반 미사일 획득 품질은 SBA 미 도입 시 획득된 품질과 비교하여 세 개의 기준 요소 중 사거리, 평균 사용연수가 높은 것으로 나타나 SBA 도입이 매우 효과적일 것으로 예상된다.

3.5 위험 지표의 효과도 입증 방법

위험도 관점에서의 SBA 도입 효과도는 동일한 무기체계 개발 과정에 대해 SBA 기반의 탐지된

위험 요소 총 합($Risk_{acq_p_sba}$)과 SBA를 도입하지 않은 경우에 탐지된 위험 요소의 총 합($Risk_{acq_n_sba}$) 간 차를 통하여 평가된다. 위험 요소는 무기체계 획득 과정에서 발생 가능한 사건을 의미하며, 사전 탐지된 위험 요소가 많을수록 실 획득과정에서의 위험요소가 감소된다고 할 수 있다.

예를 들어, 5000km의 사정거리를 가진 미사일일을 개발하고자 했을 때, 획득 과정에서 발생 가능한 위험요소의 탐지 결과는 <표 8>과 같다.

<표 8 >미사일 획득에서 발생 가능한 위험요소

기준	탐지결과	세부 탐지내용
$Risk_{acq_p_sba}$	5건	예상 사거리 내 건물 존재 미 작동 버튼 화학물질 방출 기능 부품A의 결함 표적 위치 오차
$Risk_{acq_n_sba}$	2건	예상 사거리 내 건물 존재 미 작동 버튼

<표 8>과 같이 SBA 기반 미사일 획득 시 발생 가능한 위험 요소가 5건 탐지되고, SBA 미 도입 획득 시 위험 요소는 2건이 탐지되었을 때, SBA 기반 미사일 획득에 대한 위험 요소 제거 확률이 더 높으므로 SBA 도입이 매우 효과적일 것으로 예상된다.

4. 결 론

국방 분야에서의 SBA 도입은 비용 및 시간 절감 및 품질 향상, 위험 요소 제거 등의 효과를 양산할 것으로 기대되나, 이러한 기대를 뒷받침하기 위한 구체적이며 과학적인 입증 방안이 존재하지 않아 SBA 도입 효과도 입증에 상당한 어려움이 발생하고 있다.

본 논문에서는 이러한 문제점을 해결하기 위하여 지표에 근거한 SBA 효과도 분석 방안의

필요성을 언급하고, 효과도 방안 제시에 앞서 효과도 분석의 객관적인 기준으로 작용 될 평가 지표를 정의하였다. 지표 정의를 위한 선행연구로서 국방 분야에서의 SBA 도입 목적 및 총 30건의 SBA 실적용 사례를 조사하고, 이를 바탕으로 SBA 효과도 입증을 위한 비용, 시간, 품질, 위험도 네 가지의 객관적 지표를 정의하였다. 무기체계 획득을 위한 최적의 비용 도출, 시간 절감 효과, 무기체계 획득 목적 기준의 성능 평가, 예측 불가능한 위험 요소 탐지 등은 각각의 지표를 통해 SBA 효과도를 도출하기 위해 제안되었다. 또한 지표 별 효과도 입증 방안을 제시하고, 간단한 효과도 입증 사례를 구성하여 제안된 지표 및 효과도 입증 방법을 적용해보았다. 본 논문에서 정의된 지표는 SBA 효과도를 입증하기 위한 과학적 근거로 활용 될 뿐만 아니라, SBA 효과도의 신뢰도를 향상시키는 밑바탕이 될 것이다.

향후 네 가지 지표에 대한 각각의 세부 연구를 통하여 더 높은 신뢰도를 가진 SBA 효과도를 도출 가능 할 것으로 예상된다. SBA 효과도 입증 지표는 세분화 될 수 있으며, 추가적인 지표 도출이 가능하다. 또한 비용, 시간 지표 도출을 위해 요구되는 SBA 획득 비용 및 시간에 대한 다양한 도출 방안과 SBA 효과도 분석을 위한 최적의 입증 모델 개발 연구가 진행 될 것이다.

참 고 문 헌

- [1] Patricia Sanders, "Simulation Based Acquisition: An Effective, Affordable Mechanism for Fielding Complex Technologies", OUSD(A&T), 3. 1997
- [2] V. R. Johnson, F. McKeon, R. Szanto, "SIMULATION BASED ACQUISITION : A NEW APPROACH", Report of the Military Research Fellows DSMC 1997-1998, 12. 1998

- [3] Patricia Sanders, "Study on the Effectiveness of Modeling and Simulation in the Weapon System Acquisition Process", DoD Tech, 10. 1996
- [4] 방위사업청, "방위사업청 정보화 운영 규정", 방위사업청 훈령 제 107호, 8. 2009
- [5] Bernard P.Zeigler, Herbert Praehofer, T.G.Kim, "Theory of Modeling and Simulation" 2nd Edition, Academic Press, 2000
- [6] 최상영, 변재정, "SBA(Simulation Based Acquisition) 개념과 발전 전망", 정보과학회지, 제 26권 제 11호, pp.6-12, 11. 2008
- [7] Anu Maria, "INTRODUCTION TO MODELING AND SIMULATION", Proceedings of the 1997 Winter Simulation Conference



김 형 종

1996년 2월 성균관대학교 정보공학과 공학사
 1998년 2월 성균관대학교 정보공학과 공학석사
 2001년 2월 성균관대학교 전기전자 및 컴퓨터공학과 공학박사
 2001년~2007년 한국정보보호진흥원 수석연구원
 2004년~2006년 미국 카네기멜론대학 CyLab Visiting Scholar
 2007년 3월~현재 서울여자대학교 정보보호학과 조교수

<관심분야> 인터넷전화 보안, 취약점 분석 및 모델링, 이산사건 시뮬레이션 방법론

저 자 소 개



최 달 님

2011년 2월 서울여자대학교 컴퓨터학과 공학사
 2011년 3월~현재 서울여자대학교 대학원 컴퓨터학과 석사과정

<관심분야> 개인정보보호, 시뮬레이션 및 모델링

재사용 서비스의 등록/검색을 위한 확장된 UDDI 시스템[†]

(Extended UDDI System for Registering and Discovering the Reusable Services)

신수혜[†] 백선재⁺⁺ 박준석⁺⁺⁺ 문미경^{*} 엄근혁^{**}
(Soohye Shin) (Sunjae Baek) (Joonseok Park) (Mikyeong Moon) (Keunhyuk Yeom)

요약 웹 서비스(Web Service)는 SOAP, WSDL, UDDI 등의 표준화된 XML 메시지를 통해 네트워크 상에서 상이한 시스템간의 상호작용을 가능하게 하는 소프트웨어 시스템이다. 특히, UDDI는 서비스 제공자에 의한 서비스 등록과 서비스 요청자의 서비스 검색을 지원하는 레지스트리이다. 기존의 UDDI연구는 서비스의 단순 검색과 등록에 관한 연구로, 서비스 검색 향상에 관한 UDDI 연구나 컴포넌트 단위의 재사용성 향상을 위한 확장된 UDDI설계에 관한 연구를 제시하고 있다. 본 논문에서는 기존 UDDI의 서비스 등록과 검색 기능뿐만 아니라, 재사용을 위한 서비스 모델과 이를 위한 새로운 UDDI 자료구조와 API를 제안하며, 재사용을 위한 서비스 등록과 검색 기능을 제공하는 확장된UDDI를 설계 및 구현한다. 제시된 UDDI 시스템을 통해 서비스 개발자는 이미 개발된 서비스를 사용하여 서비스 애플리케이션을 개발함으로써 개발 비용 및 시간을 줄일 수 있으며, 검증된 서비스를 재사용함으로써 품질도 보장할 수 있을 것으로 기대된다.

키워드 UDDI, 웹 서비스, 재사용을 위한 서비스, 서비스 지향 아키텍처

Abstract Web Service which consists of SOAP, WSDL and UDDI is a software system that enables interactions of heterogeneous systems on networks with standardized XML message. Especially, UDDI is a registry for supporting the registration of a service by service publisher and discovery of a service by service requester. The preliminary studies are just about publishing and searching services. It presents the researches about UDDI study for improvement of searching a service and extended UDDI design for improvement of reusability with service components aspects. In this paper, we suggest not only features about publishing and searching services of existing UDDI system but also novel UDDI data structures and API for a reusable service model. In addition, we design and implement an extended UDDI system for publishing and finding the reusable services. Therefore, by using proposed UDDI systems, service developers reduce development costs and time through developing a service application reusing the already implemented services. In addition, it can expect to ensure the quality of services by reusing the proven services.

Key words UDDI, Web Service, Reusable Service, Service Oriented Architecture

[†] 본 연구는 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단(한국연구재단-2010-2010032800)의 연구비 지원으로 수행하였습니다.

⁺ 비회원 : 부산대학교 컴퓨터공학과
ssh5304@pusan.ac.kr

⁺⁺ 비회원 : 부산대학교 컴퓨터공학과
sjbaek@pusan.ac.kr

⁺⁺⁺ 비회원 : 부산대학교 컴퓨터공학과
pjs50@pusan.ac.kr

^{*} 비회원 : 동서대학교 컴퓨터정보공학부
mkmoon@dongseo.ac.kr

^{**} 회원 : 부산대학교 컴퓨터공학과
yeom@pusan.ac.kr

논문접수 : 2011년 09월 02일

심사완료 : 2011년 09월 16일

1. 서론

웹 서비스는 SOAP[1](Simple Object Access Protocol), WSDL[2](Web Services Description Language), UDDI [3](Universal Description Discovery and Integration) 등의 표준화된 XML 메시지를 통해 네트워크 상에서 상이한 시스템간의 상호작용을 가능하게 하는 소프트웨어 시스템을 말한다[4]. 이러한 웹 서비스는 다수의 비즈니스에서 자신들의 비즈니스 로직에 대한 메타정보를 단순성과 상호운용성을

향상시키기 위해 XML 기반으로 네트워크 메시지를 작성하여, 비즈니스 간 협업과 통합을 위한 환경을 제공한다.

특히, UDDI는 웹 서비스에 대한 디렉토리 서비스를 지원하기 위해 개발된 표준 레지스트리(Registry)로, 서비스를 등록하고 검색하기 위한 메커니즘을 제공한다[5]. 이는 비즈니스 개체에 대한 정보와 해당 비즈니스 개체가 제공하는 서비스의 등록, 검색 기능을 지원한다.

최근 들어, 소프트웨어의 재사용은 소스코드의 재사용에서 소프트웨어 설계 등의 아키텍처 산출물의 재사용으로 초점이 옮겨져 왔다. 이는 소프트웨어 개발 시 소스코드의 재사용을 위해서는 아키텍처 산출물의 재사용이 선행되어야 하기 때문이다[6].

이러한 재사용성 향상을 위한 목적으로 기존에 제시된 UDDI로는, 컴포넌트에 대한 인터페이스를 기술한 컴포넌트 데이터 복을 UDDI 기술모델(tModel)에 등록하여 컴포넌트 저장소에서 적합한 컴포넌트를 선택하여 컴포넌트의 재사용을 향상시키기 위한 연구[7]가 있다. 그러나, 이 연구에서 제시하는 데이터 복은 저장소의 컴포넌트에 대한 정보만 기술함으로써 개발자에게 단순히 소스코드 단계에서의 재사용성 향상을 제공할 뿐이다.

본 논문에서는 아키텍처 수준에서의 재사용을 위한 서비스 모델을 제안하고, 이러한 재사용을 위한 서비스의 등록 및 검색 기능의 제공을 위해 기존 UDDI의 4가지 자료구조(Business Entity, Business Service, Binding Template, tModel) 이외 새로운 자료구조인 vModel과 API를 제안한다. 또한, 재사용을 위한 서비스의 등록과 검색, 결정 기능을 제공하는 확장된 UDDI를 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구로 웹 서비스, UDDI에 대해 설명한다. 3절에서는 기본 개념으로 재사용을 위한 서비스 모델과 자료구조를 설명하고, 4절에서는 서비스 재사용 향상을

위한 UDDI 시스템 설계에 대해 설명한다. 5절에서는 4절의 설계를 바탕으로 구현한 UDDI 시스템을 활용한 사례 연구 및 기존 연구와의 비교 평가를 제시한다. 마지막으로 6절에서는 결론에 대해 설명한다.

2. 관련 연구

2.1 웹 서비스

최근 서비스 지향 아키텍처(Service Oriented Architecture)가 각광 받으면서 기반 기술이 되는 웹 서비스가 주목 받고 있다. 웹 서비스란 표준화된 XML 메시지를 통해 네트워크 상에서 접근 가능한 연산들의 집합을 기술하는 인터페이스이다[5].

웹 서비스는 그림 1과 같이 서비스 제공자(Service Provider), 서비스 요청자(Service Requester), 서비스 중개자(Service Broker)로 구성된다.

서비스 제공자는 서비스를 구현하여 제공하는 주체이며, 서비스 요청자는 서비스 중개자에서 해당하는 서비스를 검색하여 서비스 제공자의 서비스를 요청하는 주체를 말한다. 서비스 중개자는 서비스 등록 및 검색을 통해 서비스 제공자가 제공하는 서비스를 서비스 요청자에게 연결하는 중개 역할을 담당한다[4]. 이러한 웹 서비스는 다음과 같은 기술을 사용한다.

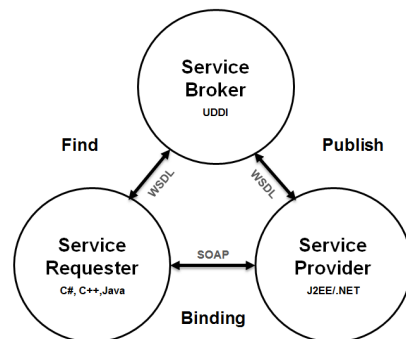


그림 1 웹 서비스 구조

- SOAP: 웹 서비스의 요청(Request) 및 응답(Response)에서 사용되어지는 메시지 전송 프로토콜
- WSDL: 웹 서비스 기술 언어
- UDDI: 웹 서비스를 등록하고 검색하기 위한 레지스트리
- 웹 서비스는 교환 메시지를 포함한 모든 정보를 XML로 표현함에 따라, 기존 웹 환경에서 제약 없이 구현할 수 있으며, 상이한 시스템간의 상호운용성을 지원하는 이점을 가지고 있다.

2.2 UDDI

웹 서비스의 핵심 기술인 UDDI는 비즈니스 기업과 그들이 제공하는 서비스들에 대한 정보를 구조화된 방법으로 저장하는 레지스트리를 말한다. UDDI는 웹 서비스의 서비스 중개자 역할을 담당하고, 서비스 제공자로부터 서비스를 등록(Publish)할 수 있으며 서비스 요청자로부터 서비스를 검색(Find)할 수 있는 인터페이스를 제공한다. 다음 요소는 UDDI의 핵심 자료구조이다.

- Business Entity: 서비스를 제공하는 비즈니스 개체에 대한 정보
- Business Service: 제공하는 서비스에 대한 상세 정보
- Binding Template: 서비스와 기술정보 모델의 연결점
- tModel: UDDI 객체의 가장 작은 기술 정보 단위 모델로써 서비스 이용의 기술 표준을 표현

UDDI는 핵심 자료구조 이외에 서비스 제공자와 서비스 요청자 사이에 UDDI를 사용하기 위한 UDDI API를 제공한다. UDDI API는 크게 서비스 제공자가 UDDI에 서비스를 등록하기 위한 Publish API와 서비스 요청자가 UDDI에 등록된 서비스를 검색하기 위한 Inquiry API로 구성된다.

기존에 제시된 UDDI로는 IBM, MS, Apache[8]에서 개방형 UDDI 레지스트리를 제시하였고, 기업, 산업체를 중심으로 사설 UDDI 레지스트리를 구축하였으며, 한국전산원을 중심으로 공용 UDDI

레지스트리가 제시되고 있다[9]. 국내·외 개발된 UDDI 레지스트리는 단일 키워드 검색 등 제한된 검색 방식에 의존하여 효율적인 정보 검색 환경을 지원하지 못하며, 등록된 서비스의 신뢰성 및 품질을 보장할 수 없다. 또한, 서비스의 재사용을 고려하지 않고 설계되었기 때문에 서비스 개발자가 UDDI에 등록된 서비스의 재사용을 기대할 수 없다.

또한, 재사용성 향상을 위해 재사용할 컴포넌트에 대한 인터페이스를 기술한 데이터 복을 제안한 UDDI 연구가 있다. 이 연구는 컴포넌트 데이터 복을 UDDI 기술모델(tModel)에 등록함으로써 서비스 개발자가 찾고자 하는 서비스를 컴포넌트 저장소로부터 선택하여 서비스의 재사용을 향상시키고자 제안하였다[7]. 그러나, 이 연구에서 제안하는 컴포넌트 데이터 복은 저장소의 컴포넌트에 대한 인터페이스 정보만 기술함으로써 개발자에게 단순히 소스코드 단계에서의 서비스 재사용 향상을 기대할 뿐이다.

본 논문에서는 재사용을 위한 서비스를 위해 기존 UDDI의 tModel과 유사한 새로운 자료구조와 이를 제어할 수 있는 추가적인 UDDI API를 제시하고, 이들을 포함하는 확장된 UDDI를 제안한다.

3. 기본 개념

3.1 재사용을 위한 서비스

재사용을 위한 서비스는 유사한 시스템의 집합인 도메인에서 공통적 요소(공통성)와 선택하여 변할 수 있는 요소(가변성)를 개발자가 식별하여 새로운 서비스를 생성하는 재사용 가능한 서비스를 의미한다[10]. 이 서비스는 그림 2와 같이 인터페이스, 오퍼레이션 타입, 메시지 타입으로 구성되며, 각 요소별 공통적/선택적 요소를 가진다.

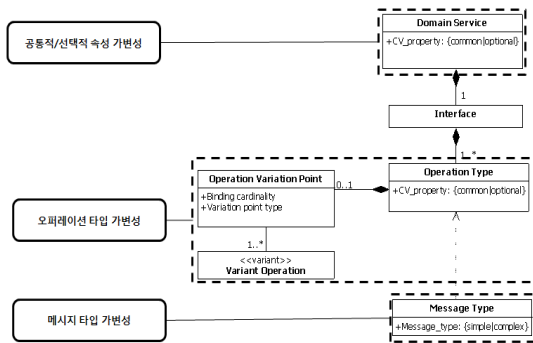


그림 2 재사용을 위한 서비스 메타모델[10]

재사용을 위한 서비스에서 오퍼레이션 타입 가변성은 다수의 실행 가능한 공통적 또는 선택적 오퍼레이션으로 구체화되는 가변을 말한다. 메시지 타입 가변성은 재사용을 위한 서비스에서 입력 및 출력으로 사용되는 메시지의 공통적 또는 선택적 메시지의 가변을 표현한다.

본 논문에서는 그림 2의 재사용을 위한 서비스 메타모델을 실체화하기 위하여, 서비스 기술 언어인 WSDL과 유사하게 재사용을 위한 서비스 기술 모델인 가변성을 표현한 모델(vWSDL)을 정의한다. vWSDL은 재사용을 위한 서비스의 인터페이스를 나타내는 vService와 오퍼레이션 타입 가변성을 표현하는 vOperation, 메시지 타입 가변성을 나타내는 vMessage로 구성된다. 이에 관한 클래스 다이어그램은 그림 3과 같다.

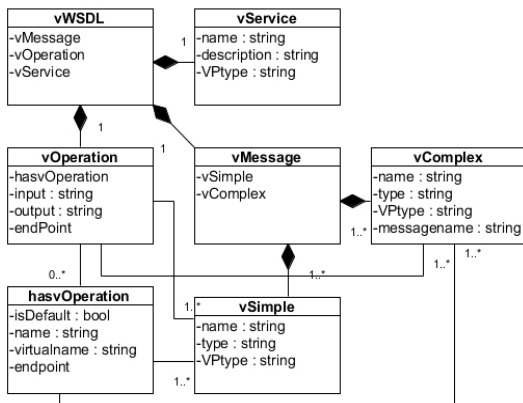


그림 3 vWSDL 클래스 다이어그램

vService는 vWSDL의 이름과 설명, 서비스의 가변 속성 타입을 포함한다. vOperation은 개발자에 의해 선택될 수 있는 오퍼레이션의 가변 속성 타입을 나타내는 hasvOperation과 각 오퍼레이션별 입력, 출력 메시지에 대한 명세를 포함한다. vMessage는 오퍼레이션의 메시지 이름과 메시지의 매개변수를 Simple타입과 Complex타입으로 분류하여 각 타입별 이름과 메시지의 가변 속성 타입을 포함한다.

3.2 UDDI 확장 자료구조

UDDI자료구조는 그림 4와 같이 기존 UDDI의 자료구조인 Business Entity, Business Service, Binding Template, tModel 이외 재사용을 위한 서비스 정보를 기술하기 위하여, tModel과 유사한 새로운 자료구조인 vModel을 제안한다.

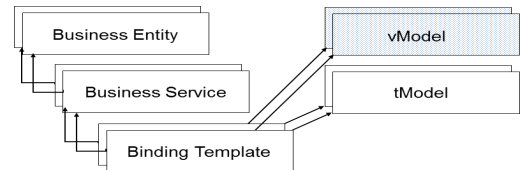


그림 4 확장된 UDDI 시스템의 핵심 자료구조

vModel은 그림 5와 같이 vModel Key와 이름, 설명을 포함한다. 그리고 재사용을 위한 서비스인 vWSDL과 오퍼레이션에 대한 설명을 포함하는 vModel OperationInfo, 카테고리 및 식별자를 포함하는 Category Bag, Identifier Bag을 포함한다.

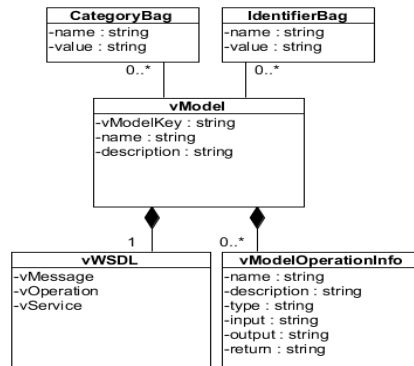


그림 5 vModel 클래스 다이어그램

vModel은 tModel과 동일하게 서비스 등록자가 vModel을 확장된 UDDI에 등록하고, 서비스 개발자가 원하는 vModel을 검색하여 vWSDL의 가변 속성 타입을 결정한 후 WSDL을 생성하도록 한다.

4. 재사용 서비스 등록/검색을 위한 확장된 UDDI 시스템

4.1 시스템 구성

본 논문에서는 기존 UDDI의 서비스 등록과 검색 기능을 지원하며, 재사용을 위한 서비스의 등록과 검색 기능을 위해 UDDI의 새로운 자료구조와 API를 포함한 확장된 UDDI 시스템을 제안한다.

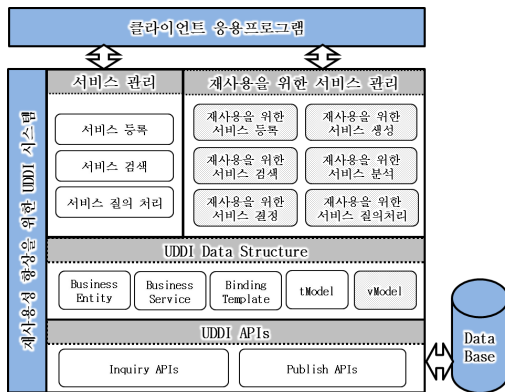


그림 6 확장된 UDDI 시스템 아키텍처

그림 6은 본 논문에서 제안하는 재사용 서비스의 등록/검색을 위한 확장된 UDDI 시스템의 아키텍처이며, 시스템 내부는 서비스 관리 계층과 재사용을 위한 서비스 관리 계층, 그리고 UDDI 자료구조와 UDDI API로 구성된다.

서비스 관리 계층은 기존 UDDI의 일반적인 서비스의 등록과 검색, API와 자료구조로의 메시지 질의 처리 부분을 담당한다.

재사용을 위한 서비스 관리 계층은 본 논문에서 제안하는 재사용을 위한 서비스의 등록과 검색, 그리고 서비스 개발자에 의한 서비스의 가변성

결정을 담당한다. 또한 재사용을 위한 서비스 등록 시 포함되어야 하는 vWSDL의 생성 및 분석과 기타 메시지 질의 처리를 포함한다.

UDDI 자료구조는 기존 UDDI 자료구조인 Business Entity, Business Service, Binding Template, tModel과 3절에서 제안한 재사용을 위한 서비스의 등록 및 검색을 위해 새로운 자료구조인 vModel을 포함한다.

UDDI API는 UDDI 3.0버전의 API를 이용하여 UDDI API를 설계하였고, vModel 자료구조를 제어하기 위한 API는 표 1과 같이, vModel을 등록하기 위한 Publish API와 UDDI에 등록된 vModel을 검색하기 위한 Inquiry API로 구성된다.

표 1 확장된 UDDI 시스템의 API

	Inquiry	Publish
Business Entity	findBusiness(Business Key):Business Entity	saveBusiness(Business Key):Business Entity
	getBusinessDetail(Business Key):Business Detail	deleteBusiness(Business Key):Business Entity
Business Service	findService(Service Key):Business Service	saveService(Business Key):Business Service
	getServiceDetail(Service Key):Service Detail	deleteService(Business Key):Business Service
Binding Template	findBinding(Binding Key):Binding Detail	saveBinding(Binding Key):Binding Template
	getBindingDetail(Binding Key):Binding Detail	deleteBinding(Binding Key):Binding Template
tModel	findtModel(tModel Key):tModel	savetModel(tModel): tModel
	gettModelDetail(tModel Key): tModel Detail	deletetModel(tModel): tModel
vModel	findvModel(vModel Key):vModel	savevModel(vModel): vModel
	getvModelDetail(vModel Key):vModel Detail	deletevModel(vModel): vModel

4.2 시스템 설계

그림 7은 본 논문에서 제안하는 재사용 서비스 등록/검색을 위한 UDDI 시스템을 나타낸 클래스 다이어그램으로, UDDI 자료구조와 API를 중심으로 나타내었다.

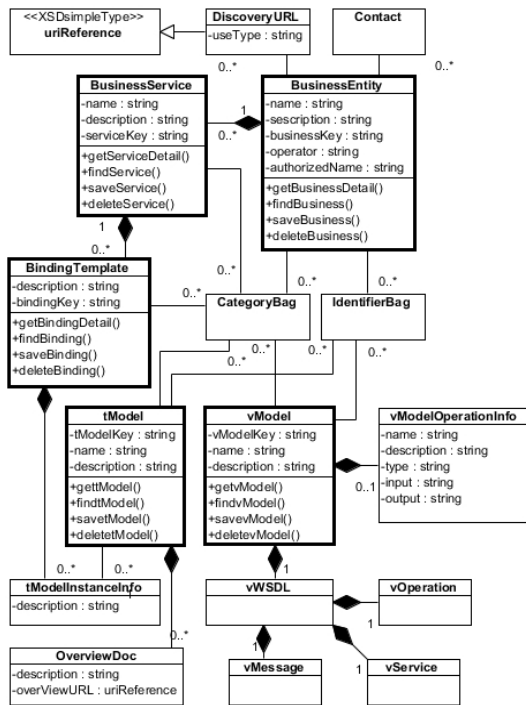


그림 7 확장된 UDDI 시스템의 클래스 다이어그램

비즈니스 개체에 대한 정보를 나타내는 Business Entity는 고유 Business Key를 보유하며 비즈니스의 연관자에 대한 연락처 정보를 포함하는 Contact와 Category별 검색을 위한 Category Bag을 가진다. 서비스에 대한 상세 정보를 나타내는 Business Service는 고유 Service Key를 보유하며 Category별 검색을 위한 Category Bag을 가진다. 서비스의 기술 정보 단위 모델인 tModel은 고유 tModel Key를 보유하며, 실행 가능한 서비스의 WSDL을 포함하는 OverviewURL을 포함한다. Business Service와 tModel의 연결점인 Binding Template은 특정 Business Service에 포함되는 tModel정보를 가지는 tModel InstanceInfo를 가진다.

또한, 재사용을 위한 서비스인 vWSDL를 위한 vModel은 고유 vModel Key와 오퍼레이션에 관한 정보를 포함한 vModel OperationInfo를 포함한다.

본 논문에서 제안하는 재사용성 향상을 위한 UDDI시스템의 배치 다이어그램은 그림 8과 같다.

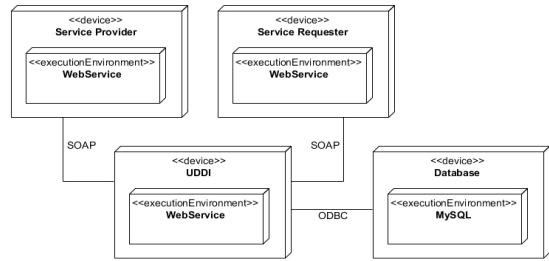


그림 8 확장된 UDDI 시스템의 배치 다이어그램

본 논문에서 제안한 확장된 UDDI 시스템은 웹 서비스로 구현하여 서비스 제공자나 서비스 요청자와 SOAP통신을 통해 UDDI 시스템을 사용할 수 있다. 이에 관한 제안 시스템의 개발 환경은 다음과 같다.

표 2 확장된 UDDI 시스템 개발 환경

	요소
클라이언트 개발언어	C#
웹 서비스	.Net C#
데이터베이스	MySQL 5.1
웹 서버	IIS 6
빌드 대상 프레임워크	.Net Framework 3.5

5. UDDI 시스템 구현 및 사례 연구와 비교평가

본 논문에서는 UDDI 시스템 사례연구로 출입 통제 시스템을 예로 들어 설명한다. 출입통제 시스템은 인가된 사람만 출입되도록 하는 시스템으로, 출입문에 RFID 카드로 신분을 점검하여 출입을 허가 및 거부하도록 한다. 그림 9는 출입 통제 시스템의 조화에 관한 UDDI 시스템의 서비스 등록과 검색 화면이다.

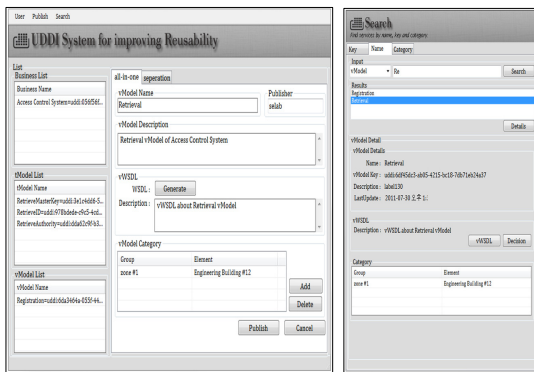


그림 9 재사용을 위한 서비스 조회 등록과 검색 화면

출입통제 시스템의 조회 서비스를 vModel로 등록하기 위해 vModel의 이름과 서비스 등록자의 이름, 설명, 카테고리를 입력한다. 또한, 조회에 관한 재사용을 위한 vWSDL을 생성하기 위하여 3.1절에서 정의한 명세에 따라 vMessage, vOperation, vService를 입력하며, vOperation은 시스템에 공통적인 Common 오퍼레이션과 가변적인 Alternative 오퍼레이션으로 구분하여 등록한다. Master Key와 ID조회는 시스템에 공통인 Common 오퍼레이션으로, 권한 조회는 가변적인 Alternative 오퍼레이션으로 입력하여 vWSDL을 생성한 후 vModel을 UDDI시스템에 등록한다. Alternative 오퍼레이션은 vModel 검색 후 가변을 제거하기 위한 결정 단계에서 개발자의 선택에 의해 서비스에 포함될지 결정된다.

UDDI의 모든 자료구조는 Key와 이름, 카테고리에 의해 검색이 가능하다. 그림 9는 출입통제 시스템의 조회에 관한 vModel의 이름에 의한 검색 화면으로, 등록된 vModel의 Detail과 vWSDL을 검색할 수 있다. vModel Detail에는 vModel Key와 vModel 설명, 등록 날짜, 카테고리 등이 포함된다.

그림 10은 출입통제 시스템의 조회 vWSDL의 가변적인 오퍼레이션들을 선택하여 가변을 제거한 서비스를 생성하게 한다.

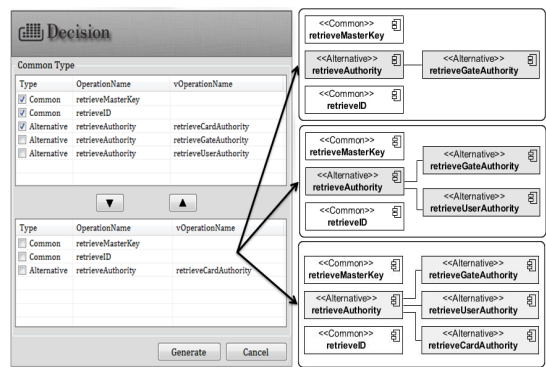


그림 10 재사용을 위한 서비스 가변성 결정 화면

Master Key와 ID조회는 Common 오퍼레이션으로 서비스의 공통적인 오퍼레이션이므로 동일하게 선택되며, 권한 조회 Alternative 오퍼레이션은 개발자의 선택에 의해 결정된다.

개발자에 의해 오퍼레이션 선택이 완료되면 선택된 오퍼레이션을 조합하여 새로운 WSDL이 생성된다. 생성된 WSDL은 vWSDL로부터 가변적인 요소가 제거되어 개발자가 원하는 서비스만을 제공하는 WSDL을 나타낸다. 즉, 개발자가 권한 조회의 출입문 권한과 사용자 권한 오퍼레이션을 선택하면, 두 오퍼레이션과 Common 오퍼레이션인 Master Key, ID 조회 오퍼레이션을 조합한 WSDL을 생성한다. 그러나 개발자가 권한 조회의 출입문 권한 오퍼레이션만 선택하면, 출입문 권한 조회 오퍼레이션과 Master Key, ID 오퍼레이션을 조합하여 WSDL을 생성할 수 있다. 이에 관한 내역은 표 3에서 설명한다.

기존에 제시된 UDDI 시스템과 본 논문에서 제안하는 서비스 재사용 향상을 위한 UDDI 시스템의 비교는 표 4와 같다. 공용 UDDI의 경우 기존의 UDDI 시스템의 기능을 충분히 제공하지만, 재사용에 대한 고려는 미흡하다. 개방형 UDDI시스템인 경우 기존의 UDDI시스템 기능과 재사용을 위한 기능 모두 미흡하다.

표 3 재사용을 위한 서비스의 선택으로 개발되는 서비스 시스템 유형

오퍼레이션		*오퍼레이션명(입력메시지명):출력메시지명	
		- retrieveMasterKey(CardInfo):CardInfo - 출입 인가를 위해 출입자의 Master Key 확인을 위한 오퍼레이션 - retrieveID(CardInfo):UserInfo - 출입 인가를 위해 출입자의 ID 확인을 위한 오퍼레이션 - retrieveAuthority: 출입 인가를 위해 출입자의 권한 확인을 위한 오퍼레이션 - retrieveGateAuthority(CardInfo):GateInfo - 출입자의 출입문 권한 확인을 위한 오퍼레이션 - retrieveUserAuthority(CardInfo):UserInfo - 출입자의 사용자 권한 확인을 위한 오퍼레이션 - retrieveCardAuthority(CardInfo):CardInfo - 출입자의 카드 권한 확인을 위한 오퍼레이션	
	A 시스템	B 시스템	C 시스템
시스템 선택 오퍼레이션	- retrieveMasterKey - retrieveID - retrieveAuthority - retrieveGateAuthority	- retrieveMasterKey - retrieveID - retrieveAuthority - retrieveGateAuthority - retrieveUserAuthority	- retrieveMasterKey - retrieveID - retrieveAuthority - retrieveGateAuthority - retrieveUserAuthority - retrieveCardAuthority
시스템 설명	출입 요청 시, Master Key와 ID를 출입문 권한에 따라 출입을 허가하는 시스템	출입 요청 시, Master Key와 ID를 출입문, 사용자 권한에 따라 출입을 허가하는 시스템	출입 요청 시, Master Key와 ID를 출입문, 사용자, 카드 권한에 따라 출입을 허가하는 시스템

표 4 UDDI 시스템 비교

평가항목		UDDI 시스템		
		공용 UDDI[9]	국외 개방형 UDDI[8]	재사용성 향상 UDDI
특징		서비스 공유를 활성화하기 위한 공용 UDDI	Apache에서 java 언어로 제작한 개방형 UDDI	재사용을 위한 서비스의 등록 및 검색 환경을 지원하기 위한 UDDI
관점 분석	서비스 등록자	(강) 네트워크를 통해 편리한 서비스 등록 지원	(약) 서비스 등록을 위한 환경 구축의 어려움	(강) 클라이언트 어플리케이션을 통해 편리한 서비스 등록 지원
	서비스 요청자	(강) 서비스 이름, Key, 카테고리 이외 검색한정자를 통해 정확한 서비스 검색 지원	(약) 서비스의 Key, 이름을 통한 검색 지원	(중) 서비스의 이름, Key, 카테고리를 통한 검색 지원
	서비스 개발자	(약) 단순 검색을 통한 서비스 조회만 가능	(약) 단순 검색을 통한 서비스 조회만 가능	(강) 재사용을 위한 서비스 검색을 통해 개발자가 필요로 하는 서비스 생성 가능
재사용 개념		미반영	미반영	(강) 재사용을 위한 서비스의 등록 및 검색으로 서비스 개발자로부터 필요로 하는 서비스를 개발할 수 있도록 지원

그러나 본 논문에서 제안하는 서비스 재사용 향상을 위한 UDDI는 기존의 UDDI시스템의 기능을 충분히 제공하면서, 나아가 재사용을 위한 기능까지 제공한다.

본 논문에 제안하는 재사용을 위한 서비스를 이용하여 개발자가 요구하는 시스템에 맞는 오퍼레이션을 선택하여 서비스를 제공함으로써

서비스의 재사용을 향상할 수 있을 것으로 기대한다. 또한 확장된 UDDI시스템을 통하여 서비스 제공자, 요청자, 개발자로부터 재사용성 향상을 위한 환경을 제공함으로써 서비스 개발의 비용 및 시간, 품질 보장을 기대할 수 있다.

6. 결 론

본 논문에서는 웹 서비스 중 핵심요소인 UDDI에서 재사용을 위한 서비스 개발을 지원하기 위하여, 재사용을 위한 서비스 모델을 설계하였고, 서비스의 재사용 향상을 위한 UDDI 시스템을 제안하였다. 이를 위하여, 새로운 UDDI 자료 구조와 API를 제안하였으며, 확장된 UDDI 시스템을 설계하고 구현하였다.

제안된 시스템을 통하여 재사용을 위한 서비스의 생성 및 검색에 대한 체계적인 서비스의 재사용이 기대된다. 향후 연구로는 제안된 UDDI 시스템 서비스의 안정성과 품질 보장을 위해 테스트 기능 및 모니터링 기능을 추가하여 UDDI 시스템을 확장할 계획이다.

참 고 문 헌

- [1] SOAP : “<http://www.w3.org/TR/soap12-part1>”
- [2] WSDL : “<http://www.w3.org/TR/wsdl>”
- [3] UDDI : “http://www.uddi.org/pubs/uddi_v3.htm”
- [4] Web Service : “<http://www.w3.org/2002/ws>”
- [5] 정보통신산업진흥원, “UDDI등을 이용한 공공 부문 전자거래 구현 방안 연구”, 정보통신 연구진흥원 학술기사, NCA IV-RER-01026, 2002
- [6] 이순복, 김진우, 송치양, 김영갑, 권주흠, 이태웅, 김현석, 백두권, “소프트웨어 제품 계열 공학의 온톨로지 기반 휘처 공통성 및 가변성 분석 기법”, 정보과학회논문지, Vol34, No.3, pp.196-211, 2007
- [7] 이동근, 최은만, “UDDI 레지스트리와 웹 기반의 컴포넌트 저장소의 통합에 관한 연구”, 한국 정보과학회 가을 학술발표논문집, Vol31, No2, pp.520-522, 2004

[8] JUDDI : “<http://juddi.apache.org/index.html>”

[9] 국가 웹 서비스 등록 저장소 :

“<http://www.uddi.nia.or.kr>”

[10] 박준석, 문미경, 염근혁, “서비스 지향 애플리케이션을 위한 가변성을 가진 서비스 실체화 방법”, 정보과학회논문지, Vol38, No.2, pp.77-85, 2011

저 자 소 개



신 수 혜

2011년 부산대학교 정보컴퓨터공학부(학사)
2011년~현재 부산대학교 컴퓨터공학과 석사과정,

<관심분야> 소프트웨어 재사용, 소프트웨어 아키텍처, 사이버네틱스 등



백 선 재

2011년 동서대학교 컴퓨터정보공학부(학사)
2011년~현재 부산대학교 컴퓨터공학과 석사과정,

<관심분야> 소프트웨어 재사용, 소프트웨어 아키텍처, 상황인식 미들웨어 등

**박 준 석**

1999년 부경대학교 컴퓨터공학과(학사)
 2002년 부산대학교 컴퓨터공학과(석사)
 2002년~현재 부산대학교 컴퓨터공학과 박사과정.

〈관심분야〉 소프트웨어 재사용, 프로덕트 라인
 공학, 서비스 지향 아키텍처, 소프트
 웨어 아키텍처 등

**문 미 경**

1990년 이화여자대학교 전자계산학과(학사)
 1992년 이화여자대학교 전자계산학과(석사)
 2005년 부산대학교 컴퓨터공학과(박사)
 2005년 3월~2005년 8월 부산대학교 차세대 물류
 IT기술사업단 박사후
 연구원
 2005년 9월~2006년 8월 부산대학교 컴퓨터 및
 정보통신 연구소 기금교수
 2006년 9월~2008년 2월 부산대학교 정보컴퓨터
 공학부 연구교수
 2008년 3월~현재 동서대학교 컴퓨터정보공학부
 조교수.

〈관심분야〉 소프트웨어 프로덕트 라인 공학,
 적응형 소프트웨어 개발, RFID
 미들웨어 개발 및 RFID 기반

애플리케이션 개발 등

**염 근 혁**

1985년 서울대학교 계산통계학과(학사)
 1992년 Univ. of Florida 컴퓨터공학과(석사)
 1995년 Univ. of Florida 컴퓨터공학과(박사)
 1985년~1988년 금성반도체 컴퓨터 연구실 연구원
 1988년~1990년 금성사 정보기기연구소 주임연구원
 1995년~1996년 삼성SDS 정보기술연구소 책임연구원
 1996년~현재 부산대학교 정보컴퓨터공학부 교수.

〈관심분야〉 소프트웨어 재사용, 프로덕트 라인
 공학, 소프트웨어 아키텍처, 센서
 네트워크 기반 상황인식 미들웨어,
 적응형 소프트웨어 개발 등

에너지 관리 시스템(EMS)과 이기종 시스템간 CIM 기반의 통합 인터페이스[†]

(An Integrated Interface based on CIM between Energy Management System
and Heterogeneous Systems)

강동현[‡]이용익[‡]박종호[‡]신용학[‡]

(Kang Dong Hyun) (Yong Ik Lee) (Jong Ho Park) (Yong-Hack, Shin)

요약 스마트 그리드 시대가 도래됨에 따라서, 세계적으로 탄소량을 줄이는 그린 에너지와 기존의 전력망에 IT기술을 결합한 친환경 지능형 전력망에 대한 관심이 높아지고 있다. 지능형 전력망을 효율적으로 사용하기 위해서는 EMS, MOS, WIS와 같은 이기종 시스템들이 서로 연계되어야 한다. 그러나 IEC에서는 EMS에 대한 인터페이스만 정의하고 있어 MOS, WIS와 같은 이기종 시스템과 연계를 위해서는 별도의 인터페이스 개발이 필요하다. 특히, 별도의 인터페이스는 각 시스템에 의존적으로 개발될 수 있기 때문에 기존 시스템과 새로운 시스템을 연계하기 위해서 추가 소프트웨어의 설계 및 개발이 필요할 수 있다. 본 논문에서는 이기종 시스템간의 상호 연계를 위한 CIM 기반 통합 인터페이스를 제안한다. 제안된 인터페이스는 이기종 시스템마다 존재하는 메시지를 하나로 통합할 수 있으며 새로운 시스템의 추가 및 기존 시스템의 삭제에서 발생하는 비용을 줄일 수 있다.

키워드 스마트 그리드, 공통 인터페이스 모델, 에너지 관리 시스템, 시장 운영 시스템, 기상 정보 시스템, 인터페이스

Abstract With the emergence of the smart grid era, the interest in green energy which reduce carbon emissions and environment-friendly smart grid which combines the advantages of power and IT technology is increasing around the world. To use the smart grid effectively heterogeneous systems such as EMS, MOS and WIS must be linked together. However, additional interface should be developed to link EMS and heterogeneous systems since IEC has only defines the interface for EMS. Especially, to link legacy system and new system, some software might need to be designed and developed because separate interfaces for connection can be developed depending on each system and environment. In this paper, we propose an integrated interface based on CIM for interconnection between systems. A proposed interface can integrate messages of heterogeneous system and will be able to reduce costs that could be incurred by the removal of the existing system and the addition of the new system.

Key words Smart Grid, Common Interface Model, Energy Management System, Market Operating System, Weather Information System, Interface

[†] 본 논문은 지식경제부에서 시행한 전력산업원천기술개발사업 (과제번호 : 2011T100200350 및 20101010400501)으로 수행되었습니다. 관계자분들께 감사드립니다.

[‡] 비회원 : LS산전 융합기술연구소

dhkangb@lsis.biz, yileel@lsis.biz

jonghop@lsis.biz, yhshin@lsis.biz

논문접수 : 2011년 08월 19일

심사완료 : 2011년 09월 16일

1. 서론

스마트 그리드 시대가 도래됨에 따라서, 세계적으로 탄소량을 줄이는 그린 에너지와 기존의 전력망에 IT기술을 결합한 효율적인 친환경 지능형 전력망에 대한 관심이 높아지고 있다. 특히,

전력망과 IT기술을 융합시키기 위해서는 기존 전력 계통 데이터에 대한 모델링이 필요하다. 이에, 국제 규격인 IEC에서는 EMS(Energy Management System)정보를 제공하기 위한 CIM(Common Interface Model)을 정의하였다.

지능형 전력망을 효율적으로 사용하기 위해서는 EMS, MOS, WIS와 같은 이기종 시스템들이 서로 연계되어야 한다. 그러나 IEC에서는 EMS에 대한 인터페이스만 정의하고 있어 MOS, WIS와 같은 이기종 시스템과 연계를 위해서는 별도의 인터페이스 개발이 필요하다. 특히, 별도의 인터페이스는 각 시스템에 의존적으로 개발될 수 있기 때문에 기존 시스템과 새로운 시스템을 연계하기 위해서 추가 소프트웨어의 설계 및 개발이 필요할 수 있다.

이에, 본 논문에서는 이기종 시스템간의 상호 연계를 위한 CIM 기반 통합 인터페이스를 제안한다. 제안된 인터페이스를 통해 각각의 시스템마다 상이하게 존재하는 메시지를 하나의 통신 인터페이스로 통합할 수 있다. 또한, 새로운 이기종 시스템 추가 또는 기존 시스템의 교체, 삭제의 경우, 시스템간의 새로운 통신 인터페이스 설정에 대한 비용을 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 CIM에 대해서 알아보고 EMS, MOS, WIS시스템의 특징과 사용 목적에 대해서 각각 설명한다. 3장은 본 논문에서 제안하는 EMS와 이기종 시스템간 CIM기반의 표준 인터페이스에 대해서 설명한다. 마지막으로 4장을 통해 결론을 맺는다.

2. 관련 연구

2.1 Common Interface Model

CIM은 CCAPI에서 만들었으며 EMS 정보 모델에 전형적으로 포함된 전력 회사 내부의 모든 주요

객체들을 표현하는 추상모델이다. 또한, CIM은 공개 클래스들(Public Classes), 객체들을 위한 속성, 모델간의 관계(Relationships)를 포함하고 있다[1].

CIM은 IEC61970, IEC61968으로 구성된다. IEC61970파트는 전력 계통에서의 발전 송전을 위한 모델링을 제공하며 IEC61968은 배전을 위한 모델링을 제공한다. 특히, IEC61968은 전력회사, 어플리케이션 기반 조직의 시스템 인터페이스를 UML을 통해 정의한다. 그림 1은 IEC에서 정의한 CIM의 최상위 패키지 레벨을 보여준다[1].

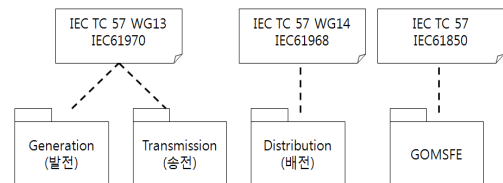


그림 1. IEC 61968의 배전관리 시스템을 위한 인터페이스

CIM에서 보여주는 객체들의 본질은 추상적이고 다양한 어플리케이션에서 사용될 수 있기 때문에 EMS의 어플리케이션 프로그램에 국한되지 않는다. 특히, 공통적인 전력계통 모델이 필요한 도메인 분야에서 특정한 적용 분야에 관계없이 시스템과 응용 프로그램 사이의 호환성을 가능하게 한다[2].

2.2 EMS, MOS, WIS

EMS는 가정 또는 빌딩과 같은 건물 내에서 사용하는 에너지의 상황을 모니터링하고 효율적으로 전력을 제어하기 위한 시스템으로써, 전력망의 최상위에 위치하여 전체적인 전력의 수급과 계통의 운용을 관장하고 있다[3].

전기의 특성상 발전된 전기의 저장이 불가능하기 때문에 발전과 동시에 소비가 반드시 이루어져야만 한다. MOS는 발전과 소비 측의 전기 생산과 수요에 대한 입찰, 자료검증, 시장가격 결정, 공시, 전력수요예측 등의 기능을 가진 양방향 전력시장 시스템이다[4][5].

WIS는 시간 별 및 주간 예보, 과거 기상 자료 및 기상 레이다 정보 등을 웹 또는 XML형태로 제공해주는 한국의 기상청 정보 시스템이다.

3. 제안 모델

3.1 CIM 기반의 시스템 구성도

본 절에서는 효율적인 지능형 전력망을 구성하기 위해 필요한 EMS, MOS, WIS간의 CIM 프로파일(Profile) 구성에 대해서 소개한다.

EMS는 서로 다른 환경과 목적을 가지는 MOS 시스템과 WIS 시스템간의 정보 교환을 목적으로 CIM 프로파일을 사용한다. 또한, EMS는 교환된 CIM 프로파일의 객체 특성에 따라 교환된 정보를 Oracle Database에 저장한다. EMS의 CIM 프로파일은 IEC 61970에서 표준 모델을 사용하며 MOS와 WIS의 CIM 프로파일은 CIM Model을 정의하는 Enterprise Architecture를 사용하여 새롭게 정의한다. 그림 2는 IEC 61970에서 정의한 EMS와 이기종 시스템과의 CIM 프로파일 기반의 표준 인터페이스 구성도를 보여준다.

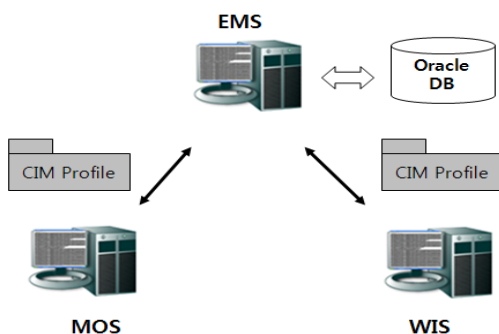


그림 2 CIM 프로파일 기반의 시스템 구성

EMS를 위한 CIM 프로파일은 IEC 61970 - 301에 정의되어 있으며 다른 시스템간의 통합이 용이한 구조를 가지고 있다. 특히, 시멘틱스(Semantics)를 이용하여 CIM Model을 정의함으로써 시스템이

내부적으로 사용하는 정보와 상관없이 데이터에 대한 접근과 정보의 교환이 가능하다. 또한, CIM Model은 객체 지향 모델링 기법을 이용하여 정의하며 CIM의 각 패키지는 패키지에 포함된 모든 클래스와 관계를 표현하기 위해 하나 이상의 클래스 다이어그램을 포함하고 있다. 그림 3은 IEC 61970 - 301에 정의된 CIM Model의 패키지를 보여준다[6].

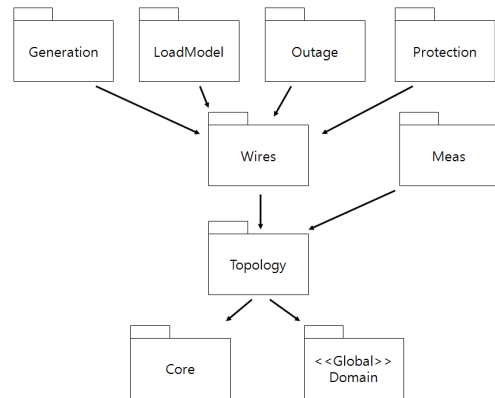


그림 3 CIM 61970-301 패키지 다이어그램

IEC에서 정의하고 있지 않은 MOS와 WIS에 대한 CIM 프로파일의 정의하기 위해서 MOS 시스템과 WIS 시스템에 대한 객체들의 속성 및 관계를 파악해야만 한다. 또한, EMS 운영에 필요한 정보를 MOS와 WIS에서 각각 추출하여 관련된 인터페이스만을 CIM 프로파일로 작성한다.

3.2 WIS을 위한 CIM 프로파일

본 절에서는 EMS와 WIS간의 통신을 위해 필요한 CIM 프로파일의 작성 방법에 대해서 설명한다.

WeatherForecast CIM 프로파일은 EMS와 WIS 시스템간의 기상 정보 교환을 목적으로 작성되었으며 시간별 예보와 주간 예보 정보를 가지고 있다. 작성된 WeatherForecast는 EMS가 기상 정보를 필요한 시점에 WIS에 요청함으로써, 상호간 정보의

교환이 이루어진다. 또한, EMS는 WeatherForecast의 정보를 필요에 따라 Oracle Database에 저장 관리할 수 있다. 기상 예보 정보를 교환하기 위한 WeatherForecast 프로파일은 표 1과 같다.

표 1 WeatherForecast 프로파일

이름	타입	최소	최대	필수
DailyForecast	String	1	1	Y
WeeklyForecast	String	1	1	Y

WeatherForecast 클래스는 DailyForecast와 WeeklyForecast로 구성되며 모두 String 타입의 객체이다. 또한, 최소와 최대를 1건으로 명시함으로써, EMS가 기상 정보를 요청할 때 최소 1건 최대 1건의 정보를 전달해야 한다. 즉, DailyForecast와 WeeklyForecast는 EMS와 WIS간의 WeatherForecast의 정보 교환 시 필수 항목이다. 그림 4은 표 2을 통해 정의한 WeatherForecast의 클래스를 CIM Model 프로그램인 Enterprise Architect 을 이용하여 작성한 CIM 프로파일을 보여준다.

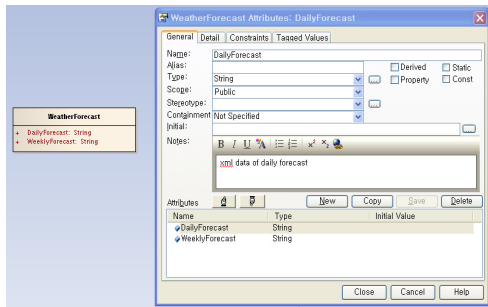


그림 4 WeatherForecast 클래스

Enterprise Architect을 이용하여 새로운 CIM 프로파일을 정의할 때, 각 객체의 속성 및 상속 관계를 명확히 정의해야 한다.

WeatherForecast 클래스가 완성되면 클래스를 기반으로 스키마 파일을 생성한다. 생성된 스키마 파일은 EMS와 WIS간 통신을 위해 사용된다.

4. 결 론

전력망의 기술이 발달함에 따라 기존의 전력망에 IT기술을 결합한 지능형 전력망에 대한 관심이 높아지고 있다. 그러나 전력망에 대한 규격을 가진 IEC에서는 발전, 송전, 배전 정보를 위한 인터페이스만을 CIM에서 정의하고 상이한 시스템간의 연계에 대한 통신 인터페이스는 정의하고 있지 않다. 즉, MOS, WIS와 같은 시스템간의 통신을 위한 인터페이스는 정의하고 있지 않다.

본 논문에서는 효율적인 지능형 전력망을 구성하기 위해 EMS, MOS, WIS간의 통신 인터페이스를 CIM 기반으로 작성하는 방향을 제시하였다. 그 결과 시스템별 상이하게 존재하는 이 기종간의 메시지를 하나의 통신 인터페이스로 통합 할 수 있었다. 특히, 통합된 인터페이스는 향후 기존 시스템의 통신 인터페이스 재설정 및 삭제 그리고 새로운 시스템과의 통신 인터페이스 설정에 용이할 것이다.

참 고 문 헌

- [1] IEC International Standard, "IEC61968 Part 1: Interface architecture and general requirements", 2003.
- [2] 이진호, 손진만, 남영우, 이용익, 박종호, 김필석, 김병섭, 신용학, "IEC 61970 기반의 EMS 데이터 베이스 구축에 관한 연구", 대한전기학회 전력기술 부문회 추계학술대회, 2006.
- [3] 윤상훈, 조운성, "한국형 EMS 시스템용 실시간 조류계산 프로그램 개발", 전기학회논문지 제59권 제 2호, pp. 225-483, 2010.
- [4] 전기위원회웹진, http://www.leadernews.co.kr/korec/webzine03/source/main_mail.asp?wm=200312&seq=19&th=1&per=1

- [5] 박봉용, 김명웅, 안재승, 김민배, “계통운영 시스템(EMS)과 시장운영시스템(MOS,CBP)간의 연계 운영” 대한전기학회 제 40회 하계학술대회, 2009.
- [6] IEC International Standard, “Energy management system application program interface (EMS-API) –Part 301 : Common Information Model(CIM) base”, 2003.

저 자 소 개



강 동 현

2007년 한국산업기술대학교 컴퓨터공학과(학사).
 2010년 성균관대학교 전자전기컴퓨터공학과
 (공학석사).
 2011년~현재 LS산전 융합기술연구소 Smart
 Platform 연구팀 연구원.

〈관심분야〉 스마트그리드, CIM, AMI 등



이 용 익

2000년 중앙대학교 제어계측공학과(학사, 공학석사).
 2000년~2002년 의료기 회사 연구원.

2002년~현재 LS산전 융합기술연구소 Smart Platform
 연구팀 책임 연구원.

〈관심분야〉 CIM, IEC61970, 61968, 61850, AMI 등



박 중 호

1993년 고려대학교 전산학과(학사, 공학석사).
 1993년~현재 LS산전 융합기술연구소 Smart Platform
 연구팀 수석 연구원.

〈관심분야〉 스마트그리드 운영 시스템(EMS,
 디지털 변전소, SCADA) 등



신 용 학

1984년 한양대학교 수학과(학사).
 1986년 한양대학교 전자계산학과(석사).
 1988년~현재 LS산전 융합기술연구소 Smart Grid
 연구단 연구단장.

〈관심분야〉 스마트그리드 운영 시스템(EMS,
 디지털 변전소, SCADA), CIM 등



회원 가입 안내 및 회비 납부 요령



한국정보과학회 소프트웨어공학소사이어티는 회원 여러분에게 유익한 정보를 제공해 드리기 위하여 보다 충실한 내용의 논문지 발간 배포, 그리고 국제·국내 학술발표회 및 초청강연회와 단기강좌 등의 여러 가지 사업들을 추진하고 있습니다.

소프트웨어공학 소사이어티의 가입을 통해 정보 및 기술 교류, 그리고 인적 네트워크의 구성에 참여하시기를 기대합니다. 회원 가입을 위하여 아래의 회비 안내를 참고하시어 회비를 납부하시고, 다음 쪽의 입회원서를 작성하시어 아래 소프트웨어공학소사이어티 주소로 보내주시거나 팩스 또는 이메일을 통해 보내어 주시기 바랍니다.

한국정보과학회 소프트웨어공학소사이어티 연락처는 아래와 같습니다.

◆ 소프트웨어공학소사이어티

주 소 : (우) 121-742 서울시 마포구 신수동 1번지, 서강대학교 신과학관 202호
한국정보과학회 소프트웨어공학소사이어티 박수용

전 화 : (02) 705-8928

팩 스 : (02) 704-8273

전자우편 : sypark@sogang.ac.kr (박수용교수), bjlee@uos.ac.kr (이병정교수)

홈페이지 : <http://www.sigse-kiss.or.kr/>

◆ 회비 안내

회원구분	• 학생회원 : IT 분야 학과 또는 관심 있는 학생 • 정회원, 종신회원 : IT 분야 종사자
가입비	학생회원, 정회원 : 20,000원, 종신회원 : 200,000원
년회비	학생회원, 정회원 : 20,000원

- 회비납부 방법

- (1) 무통장입금 또는 계좌이체 후 입회원서 발송
: 계좌 번호 : 제일은행 150-20-358028 (이병정)
- (2) 소사이어티 주관 학술행사 개최시, 행사장 당일 가입 및 납부 가능



본인은 한국정보과학회 소프트웨어공학소사이어티의 취지에 찬성하여 회원으로 가입하고자 이에 입회원서를 제출합니다.

신 청 인: (인)

한국정보과학회 소프트웨어공학 소사이어티 회장 귀하



논문지 논문 모집 (Call for Papers)



한국정보과학회 소프트웨어공학 소사이어티에서는 매년 4회에 걸쳐 ‘소프트웨어공학 소사이어티 논문지’를 발간하고 있습니다. 이 논문지에는 소프트웨어공학 전반에 걸친 연구논문과 산업계 논문을 게재해 오고 있습니다. 다음과 같은 소프트웨어공학 주제에 관련된 논문을 모집하고 있으니 학계와 산업계의 여러분의 적극적인 논문투고를 바랍니다.

◆ 논문 주제

- 소프트웨어 설계 및 아키텍처
- 소프트웨어 재사용 및 프로덕트라인
- 요구공학
- 소프트웨어 품질 및 테스트
- 관리 및 프로세스
- 소프트웨어 정형 기법
- 서비스기반 소프트웨어 개발
- 임베디드, 모바일, 웹기반 소프트웨어 개발
- 기타 소프트웨어 응용 (국방, 자동차, 조선 등의 분야)

◆ 논문심사

- 투고된 논문은 편집위원회에서 심사 선정하며, 필요 시 외부 심사위원을 위촉하여 심사를 합니다. 제출된 논문은 반환하지 않습니다.
- 심사료 및 게재료: 없음

◆ 논문 제출

- 소프트웨어공학 소사이어티의 논문지 투고 양식(<http://www.sigse-kiss.or.kr/>)을 사용하며, 논문의 분량은 10장으로 제한합니다.
- 논문지 투고규정에 따라 작성된 심사용 논문파일은 온라인투고시스템을 통하여 투고하시기 바랍니다.

◆ 문의처 (편집위원회)

- 편집이사 : 고인영 교수 (KAIST, 042-350-3547, iko@kaist.ac.kr)
- 편집이사 : 윤희진 교수 (협성대학교, 031-299-0841, hjyoon@uhs.ac.kr)
- 편집위원 : 강성원 교수 (KAIST, 042-350-3512, sungwon.kang@kaist.ac.kr)
- 편집위원 : 김문주 교수 (KAIST, 042-350-3543, moonzoo@cs.kaist.ac.kr)
- 편집위원 : 김정아 교수 (관동대학교, 033-649-7801, clara@kd.ac.kr)
- 편집위원 : 이우진 교수 (경북대학교, 053-950-6378, woojin@knu.ac.kr)
- 편집위원 : 이찬근 교수 (중앙대학교, 02-820-5829, cglee@cau.ac.kr)
- 편집위원 : 이 근 박사 (삼성전자, 031-277-7323, gskeun@gmail.com)
- 편집위원 : 이세영 박사 (정보통신산업진흥원, 02-2132-1323, sarahlee230@gmail.com)



투 고 요 령



1. 소프트웨어공학소사이어티 논문지에 실리는 원고는 주제 논문, 일반 논문, 산업체 기고 등으로 구분하며 다음과 같은 분야에 대하여 모집한다.
 - 가. 소프트웨어공학 및 그 응용분야에 대한 연구결과
 - 나. 강좌 및 관련 교육사항 소개 (목적, 과정, 일정, 대상, 특징)
 - 다. 소프트웨어 도구 및 방법론 소개 (가격, 특징, 종류, 적용사례)
 - 라. 소프트웨어 산업에 대한 학계, 업계의 주요 관심사
 - 마. 기타 관련 사항
2. 투고자는 원칙적으로 본 소사이어티의 회원으로 한다. 다만 공동 또는 초청 기고자는 예외로 한다.
3. 논문은 원칙적으로 한글로 작성한다.
4. 원고는 한글(hwp), 워드(MS Word), PDF 형식 중 하나를 택하여 A4용지에 작성하며, 그림과 표를 포함하여 10쪽 이내로 한다.
5. 논문 내용에 직접 관련이 있는 문헌에 대해서는 이들 문헌에 관련이 있는 본문 중에 참고 문헌 번호를 쓰고 그 문헌을 참고문헌 난에 인용 순서대로 기술한다. 참고문헌은 학술지의 경우 저자, 제목, 학술지명, 권, 호, 쪽수, 발행 연도의 순서로, 단행본은 저자, 서명, 쪽수, 발행처, 발행 연도의 순서로 기술한다.

[1]Cole, R., "Parallel Merge Sort", SIAM Journal of Computing, vol.17, No.4, pp.770-785, 1988.

[2]김수형, 강명호, 조형재, 송주석. "안전하고 효율적인 침입자 역추적 시스템", 정보과학회논문지, 제25권, 제10호, pp.1123-1131, 1998
6. 논문은 소프트웨어공학 소사이어티(<http://www.sigse-kiss.or.kr/>)의 온라인 투고 시스템을 통해 제출한다.
7. 논문투고신청서를 반드시 작성하여 이메일(hjyoon@uhs.ac.kr)로 제출한다.
7. 원고 접수는 수시로 하며, 접수일은 온라인 접수일로 한다.
8. 기타 자세한 사항은 한국정보과학회 논문지 투고 요령을 따른다.



한국정보과학회 소프트웨어공학소사이어티 임원명단



직책		성명	소속
회 장		박 수 용	서강대학교
부 회 장		권 기 현	경기대학교
부 회 장		민 경 호	LG전자
부 회 장		전 진 옥	비트컴퓨터
부 회 장		최 병 주	이화여자대학교
부 회 장		한 혁 수	상명대학교
감 사		이 금 해	항공대학교
		차 성 덕	고려대학교
자 문 위 원 회		강 교 철	포항대학교
		권 용 래	KAIST
		배 두 환	KAIST
		성 기 수	KISTI 고문
		신 규 상	ETRI
		양 승 민	숭실대학교
		우 치 수	서울대학교
		이 경 환	중앙대학교
		이 단 형	KAIST
		정 기 원	숭실대학교
		황 선 명	대전대학교
협 력 기 관 위 원 회		권 경 룡	국방품질기술원
		박 찬 규	국방SW산학연합회
		신 석 규	SW시험인증센터
		유 법 민	지식경제부
		윤 태 권	SW기술진흥협회
		이 상 은	한국소프트웨어진흥원
운 영 위 원 회	운 영 위 원 장	권 기 현	경기대학교
	총 무 이 사	이 병 정	서울시립대학교
	기 획 이 사	백 종 문	KAIST
	조 직 이 사	이 관 우	한성대학교
	학 술 이 사	홍 장 의	충북대학교
	편 집 이 사	고 인 영	KAIST
	편 집 이 사	윤 회 진	협성대학교
	협 력 이 사	인 호	고려대학교
기 술 위 원 회	홍 보 이 사	이 정 원	아주대학교
	기 술 위 원 장	최 병 주	이화여자대학교
	요 구 공 학 분 과 장	박 수 진	서강대학교
	아 키 텍 처 분 과 장	조 은 숙	서일대학
	재 사 용 분 과 장	이 관 우	한성대학교
	정 형 화 분 과 장	김 문 주	KAIST
	테 스 팅 분 과 장	김 영 철	홍익대학교
	프 로 세 스 분 과 장	이 세 영	NIPA

	직책	성명	소속
산 학 위 원 회	고 신 회 성 분 과 장	배 현 섭	슈어소프트테크
	재 공 학 분 과 장	김 진 태	SEEG
	실 시 간 분 과 장	이 찬 근	중앙대학교
	산 학 위 원 회 장	전 진 옥	비트컴퓨터
	자 동 차 분 과 위 원 장	유 영 수	오토에버시스템(주)
	국 방 분 과 위 원 장	윤 희 병	국방대학원
	금 융 분 과 위 원 장	김 정 아	관동대학교
	전 자 분 과 위 원 장	이 근	삼성전자
	의 료 분 과 위 원 장	최 호 진	KAIST
	전 력 분 과 위 원 장	이 장 수	한국원자력 연구소, MMIS팀 책임연구원
	이 사	강 성 원	KAIST
	이 사	계 승 교	삼성SDS
	이 사	권 원 일	STA컨설팅
	이 사	남 영 광	연세대학교
	이 사	김 문 주	KAIST
	이 사	김 상 기	현대자동차
	이 사	김 영 철	홍익대학교
	이 사	김 정 아	관동대학교
	이 사	민 상 윤	솔루션링크
	이 사	박 복 남	핸디피엠지
	이 사	백 종 문	KAIST
	이 사	손 세 창	인천공항공사
	이 사	손 진 규	삼성탈레스
	이 사	양 상 옥	KAI
	이 사	염 근 혁	부산대학교
	이 사	오 재 원	카톨릭대학교
	이 사	유 준 범	건국대학교
	이 사	윤 형 진	케피코
	이 사	윤 희 병	국방대학원
	이 사	이 근	삼성전자
	이 사	이 병 결	서울여자대학교
	이 사	이 성 남	방위사업청
	이 사	이 우 복	삼성전자
	이 사	이 우 진	경북대학교
	이 사	이 은 주	경북대학교
	이 사	이 정 원	아주대학교
	이 사	이 해 서	솔루션링크
	이 사	장 주 수	모아소프트
	이 사	정 연 대	N3SOFT
	이 사	정 인 상	한성대학교
	이 사	조 병 인	국방과학연구소
	이 사	조 상 윤	다한테크
	이 사	최 승 훈	덕성여자대학교
	이 사	최 종 무	단국대학교
	이 사	현 창 문	탐라대학교



2010-2011 소프트웨어공학소사이어티 논문지 편집위원회



편집위원장 윤회진 교수(협성대학교)

편 집 위 원 고인영 교수(KAIST)

강성원 교수(KAIST)

김문주 교수(KAIST)

김정아 교수(관동대학교)

이우진 교수(경북대학교)

이찬근 교수(중앙대학교)

이 근 박사(삼성전자)

이세영 박사(정보통신산업진흥원)



소프트웨어공학소사이어티 논문지 제24권 제3호 (통권 91호)



발 행 일 ॥ 2011년 9월 30일

발 행 인 ॥ 박수용

편 집 인 ॥ 윤회진

발 행 처 ॥ 사단법인 한국정보과학회 소프트웨어공학소사이어티

연 락 처 ॥ 서울특별시 마포구 신수동 1번지, 서강대학교 신과학관 202호

전 화 : 02-705-8928, 팩 스 : 02-704-8273

홈페이지 : <http://www.sigse-kiss.or.kr/>

인 쇄 처 ॥ (주)참기획 (전화 : 042-861-6380, 팩스 : 042-861-6381)

Copyright© 2011 한국정보과학회 소프트웨어공학소사이어티(비매품)