

# 임상의사결정지원 시스템 아키텍처 수립 및 적용 사례\*

## (Construction of Clinical Decision Support System Architecture and Case Study)

김정아<sup>¶</sup>      조인숙<sup>§</sup>

(Jeong Ah Kim) (InSook Cho)

**요 약** 의료분야에서의 품질 관리의 중요성이 강화되면서 임상의사결정지원 시스템의 중요성이 높아지고 있다. 이로 인하여, 대형병원 뿐만 아니라 규모가 작은 지역 병원에서도 임상의사결정지원 시스템을 구축하는 것이 필요한 실정이다. 그러나 임상의사결정지원 시스템 구축은 지식저작과 정보시스템 구축, 기존 시스템과의 연동 등 다양한 활동이 필요한 작업으로 비용과 복잡도가 크다. 본 논문에서는 재사용 및 상호운용성을 보장할 수 있는 의료 지식과 재사용할 수 있는 임상의사결정 지원 시스템 구축에 필요한 아키텍처를 구현한다. 3개 병원에 제안한 아키텍처 적용 결과를 바탕으로 아키텍처의 실무 적용 가능성 및 유용성을 검증한다.

**키워드** 임상 의사 결정 지원 시스템, 아키텍처, 임상 지식, Quality control in medical is getting very important issue so that the importance of CDS(Clinical Decision Support) System has been increased. Local clinics as well as big hospitals are required to implement the CDS System. But the cost and complexity of CDS system implementation is so high since many different activities including knowledge authoring, software development, and integrating the legacy system are necessary. In this paper, we suggest the CDS system architecture to be sharable and interoperable and evaluate the availability and efficiency of this architecture.

Key words : Clinical Decision Support System, Architecture, Clinical Knowledge

### 1. 서 론

의료정보기술이 의료분야에 가장 크고 강력하게 기여하는 것은 바로 사람의 한계를 뛰어 넘는 업무를 대신 수행해주는 일일 것이다[1]. 임상 의사결정 지원(Clinical Decision Support: 이하 CDS) 시스템은

환자안전과 의료진을 동시에 보호할 수 있는 안전 장치에 해당하는데, 이러한 안전장치가 필요한 이유는 우리의 임상 실무가 항상 모범적인 것만은 아니기 때문이다. 미국을 비롯한 선진국의 CDSS 관련 연구개발은 약 50년의 역사를 갖고 있다. 대부분 연구개발이 의료정보 학계를 중심으로 룰(rule) 기반에서부터 다양한 인공지능과 기계학습(Machine Learning)법 적용이 폭 넓게 소개된 것에 비해 임상 실무와 산업계는 룰 기반 시스템을 선호하고 있다[2]. 국내에서도 데이터 마이닝 접근법에 의한 CDS 서비스 구현을 시도 하고 있지만, 실제 임상에 적용한 경우는 없다.

보건복지부에서 2005~2010까지 EHR핵심공통 기술 연구개발사업단(이하 EHR사업단)을 통해

\* Copyright©2004 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지: 소프트웨어 및 응용 제 권 제호(2011.)

<sup>¶</sup> 종신회원 : 관동대학교 컴퓨터교육과(clara@kd.ac.kr)

<sup>§</sup> 비 회 원 : 인하대학교 간호학과(insook.cho@inha.ac.kr)

범 국가적 차원에서 의료정보 시스템이 갖춰야 할 표준을 의료 관점과 정보 관점에서 정의하는 연구를 수행해 있다.

EHR 사업단 연구 목표중 하나가 CDS 서비스 분야에서 모든 병원이 재사용할 수 있는 지식을 정의할 수 있는 방법과 이들 지식을 실행할 수 있는 환경을 기존 정보 시스템과 독립적으로 구현하고, 기존 시스템과 쉽게 통합할 수 있는 방법을 제시하는 것이다. 본 논문에서는 EHR 사업단의 CDS 서비스 연구 결과 중 일부인 CDS 아키텍처를 정의하고 구현 결과를 제시하며 적용 가능성을 검증한다.

2장에서는 새로운 CDS 시스템 아키텍처 설계의 필요성과 범국가적 차원에서 CDS 아키텍처의 요구사항을 정의한다. 3장에서는 EHR 사업단에서 이루어진 연구 결과인 CDS 시스템 아키텍처를 정의하고 4장에서는 대형 병원 3개 기관에 아키텍처를 구현한 결과를 분석하여 아키텍처를 평가한다. 5장에서는 연구 결과 활용 방법과 향후 연구 방향을 정리한다.

## 2. 기존 CDSS 시스템 구현의 한계점

### 2.1 CDS 시스템 개요

의료 분야에 있어서 전문의의 임상에서의 진단을 지원하는 시스템을 CDS 시스템이라고 한다. 병원의 의사결정자들에게 적절한 시기에 정보를 제공함으로써 조직의 계획, 운영 및 통제기능을 지원해주는 정보체제이다[7]. 컴퓨터시스템의 도움을 받아 인간이 할 수 없는 적절한 데이터, 정보, 그리고 지식 관리를 통해 이러한 불확실성에 의한 오류를 최소화 하여 보다 정확한 의사결정을 지원하도록 하는 것이 CDSS의 개발과 활용 목적이다. 의학 분야에 있어서 지능적 DSS 또는 전문가 시스템의

개발 목적은 다음과 같다[8] (1) 시스템적이고, 완전하며 분산된 자원으로부터 데이터를 통합함으로써 임상적 진단의 정확성을 향상시킨다. (2) 동일하지 않으며 검증되지 않은 유사성을 갖는 사례의 영향을 받지 않도록 함으로써 임상적 의사 결정의 신뢰성을 향상시킨다. (3) 결정적인 행위의 위험(risk)과 이점(benefits)에 대한 불편함과 시간적인 소모에 균형을 둬으로써 검사와 치료의 비용 효과를 향상시킨다. (4) 일관성이 없고 부정확한 지식을 규명하기 위한 기술의 발전과 연관되어 의학적 지식의 구조에 대한 이해를 증진시킨다. (5) 의학적 교육을 향상 시키고 의학에 대해 이해를 돕기 위한 보다 효율적이고 용이한 시스템을 개발하기 위한 노력으로 임상적 의사 결정의 이해를 증진시킨다.

### 2.2 기존 CDS 시스템 아키텍처

초기에 제안된 CDSS 아키텍처는 매우 단순해서, CDSS는 stand-alone 애플리케이션 방식을 전제된 것으로 수작업으로 사례를 입력하고 설명문으로 된 진단을 제공하는 수준이었다. 2006년 스탠포드 대학 연구 팀이 제안한 SAGE 아키텍처는[3]은 그림 1과 같은 모습으로 VMR(Virtual Medical Record)인터페이스 및 용어 서버와의 연계를 통해 의료 표준을 준수함으로써 재사용 가능한 지식을 실행할 수 있는 지식 엔진 기반 CDS 시스템 아키텍처를 제안하였다. SAGE 아키텍처는 의료 표준을 정의하는 기관인 HL 7이 정의한 VMR을 기준으로 각 병원의 의료정보 시스템이 구축되어 있다는 전제가 있고, 각 병원이 표준 용어를 사용함을 전제로 하였다. 그러나 우리나라의 많은 병원은 자체 의료정보 시스템 데이터 모델을 정의하였고, 자체 용어를 정의하고 있기 때문에, 지식의 상호 운영성을 고려한다면 국내 실정에 적합하지 않는 모델이다.

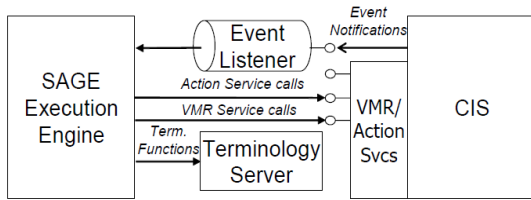


그림 1 SAGE 시스템 아키텍처

그림 2는 Kawamoto가 제안한 아키텍처[4]로, 어플리케이션이 중심이 되는 CDSS 아키텍처이며, 독립된 CDSS어플리케이션에 지식과 처리 서비스를 함께 정의하는 방식이다.

SEBASTIAN 아키텍처는 어플리케이션과 지식을 분리하지 않은 모델로, 지식과 서비스를 분리하여 재사용할 수 없다는 단점과 SAGE 모델이 갖고 있는 상호운영성을 보장할 수 없다는 단점이 존재한다.

### 3. EHR 관점에서의 CDS 시스템 아키텍처 설계

#### 3.1 아키텍처 수립 기본 사항

기존 CDS 시스템 아키텍처를 보완하기 위해 가장 핵심적 개선 요구사항은 4가지이다.

- 1) CDS 실현에 필요한 지식과 CDS 서비스를 구현한 모듈은 반드시 분리하여 각각의 유지보수성과 확장성을 확보해야 한다.
- 2) 전문가들이 저작한 지식과 지식 실행 모듈은 다양한 병원 정보 시스템과 쉽게 통합할 수 있어야 한다.
- 3) 새롭게 CDS 서비스를 추가하기 위한 기존 시스템의 변경 사항은 최소화해야 한다.
- 4) CDS를 구성하는 요소 및 구현 방법은 다양하므로, 이를 수용할 수 있어야 한다.

#### 3.2 CDS 시스템 아키텍처

구현을 위한 핵심 CDSS 아키텍처 수립을 위하여 본 연구에서는 4가지 아키텍처 원칙을 수립하였다[1,6].

첫째, 통합 아키텍처를 수립한다. 둘째, 컴포넌트 기반 Adaptable 아키텍처를 수립한다. 셋째, 구현 가능한 아키텍처로 수립한다. 넷째, 맞춤형 아키텍처로 수립한다.

EHR 관점에서는 동일한 지식을 서로 다른 병원 환경에서 활용할 수 있어야 하며, 기존 애플리케이션과의 유연한 통합을 보장할 수 있어야 한다. 이를 위해서 애플리케이션 아키텍처는 크게 5개의 빌딩 블록 (① 저작환경 모듈 ② 지식엔진 모듈 ③ 데이터 인터페이스 어댑터 모듈 ④ 지식 저장소 모듈 ⑤ 서비스 인터페이스 모듈)으로 정의하였다.

- (1) 저작환경모듈 : 료과 프로세스를 기반으로 하여 지식을 표현하고, 저작한 지식을 컴퓨터가 이해할 수 있는 형태의 지식으로 변환하는 환경이다. 지식 저작 과정의 핵심인 검증을 지원할 수 있는 테스트 케이스 생성기 및 테스트 자동 실행기를 지원하여 지식저작 및 검증의 생산성 향상을 가능하게 한다.
- (2) 지식엔진 모듈: 임상지식을 실행하는 엔진으로 료 엔진과 프로세스 엔진으로 구성하였다. 이로써 프로세스 기반 지식 및 if-then으로 이루어진 기본 료 모듈을 처리할 수 있다[10]. 이를 통해 어플리케이션에 포함해둔 지식을 별도로 분리할 수 있는 기반을 제공하였다.
- (3) 데이터 인터페이스 어댑터: EHR관점에서 지식은 어느 특정 병원에 종속적인 지식으로 저장할 수 없다, 그러므로 지식은 HL7 표준으로 추진되고 있는 VMR(Virtual Medical Record)를 확장한 eVMR(extendible VMR)을 기반으로 저작하여 범용성을 유지해야 한다. 범용성 유지를 위해 필요한 컴포넌트들로 데이터 인터페이스 어댑터를 구현하였다.
- (4) 지식 저장소 모듈: EHR관점에서 지식은 다양한 의료기관 간에 공유할 수 있는 자산이다. 다양한 유형의 지식을 메타 데이터화 하여 저장할 수 있는 컴포넌트, 지식 검증 피드백을 주고 받을

수 있는 협업 환경, 지식의 변경 및 형상 관리 모듈 구현을 통해 지식을 자산으로 공유할 수 있는 환경을 구현하였다.

- (5) 서비스 인터페이스 모듈: 본 논문에서는 2가지 방식을 제안한다. 연계형 방식은 CDSS 서비스에 필요한 화면을 별도로 구현하되, 기존 병원정보 시스템에서는 새로운 서비스를 호출하는 변경하는 경우이다. 통합형 방식은 기존 병원정보 시스템의 화면 변경 및 CDSS 서비스 호출 및 결과 처리를 위한 모듈 수정이 필요한 방식이다.

## 4. 적용 사례

### 4.1 배포 프로세스

논문에서 제안한 아키텍처와 구현한 공통 컴포넌트를 병원에 적용한 배포아키텍처는 그림 4와 같다. 전자의무기록 영역에는 환자 정보를 제공할 수 있는 기존 정보시스템과 정보 시스템으로부터 데이터를 획득할 수 있는 데이터 인터페이스 어댑터 모듈을 탑재한다. 어플리케이션 영역에 새롭게 개발한 CDSS 서비스 모듈을 탑재하고, 지식 엔진과 연계할 수 있는 모듈을 탑재한다. 엔진 영역에는 CDSS 서비스를 위한 지식 실행 엔진과 실행 중에 발생하는 추가 환자 정보 획득을 위한 데이터 인터페이스 어댑터 모듈을 탑재한다.

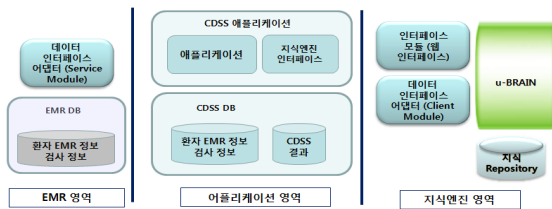


그림 2 배포 아키텍처

### 4.2 검증 결과

본 논문에서 제안한 아키텍처를 검증하고 CDS 운영의 효과를 검증하기 위하여 아주대의료원의 약물상호작용 CDSS, 분당서울대병원의 진단검사 CDSS 임상 평가를 수행하였다[1,9]. 표 1에서 보는 바와 같이 지식엔진은 상당히 안정적인 값을 보여주고 있었으며, 데이터 인터페이스 어댑터 부분이 대부분의 시간을 점유하고 있는 것으로 나타났다. 연구 결과로 만든 아키텍처를 검증하기 위해서는 성능 및 상호운영성을 평가해야 한다. 상호운영성 평가를 위해서는 동일한 아키텍처를 기반으로 하나의 지식을 다수 병원에서 서비스할 수 있어야 한다. 그러나 병원이 갖는 특수성으로 인하여 상호 운영성 검증을 위해서는 일차 성능 검증이 필요하다. 우선 성능 검증을 하기 위하여 운영 환경과 동일한 구성으로 테스트 환경으로 구축하여 검증을 수행하였다.

표 1 성능 평가 결과

(단위: ms)

임상 환경	# of trans.	평균 시스템 반응시간 (SD)		
		전체 응답시간	지식실행 시간	EMR정보 획득시간
외래	122,949	197.36 (251.46)	56.00 (29.02)	141.36 (249.17)
입원	149,282	626.22 (853.22)	48.02 (17.14)	578.20 (853.80)
응급실	51,214	193.98 (247.97)	53.10 (25.16)	140.89 (251.32)

시스템적 활용과 기대효과로는, 다음과 같은 임상적 평가를 받고 있다.

- (1) 서로 다른 의료기관에 적용할 수 있다.
- (2) 독립된 서비스 형태로 여러 의료기관을 대상으로 서비스가 가능하다.
- (3) CDSS 서비스를 확장하는 데 있어 추가 시스템 개발 없이 지식확장, 대상자 확장, 사용자 확장이 유연하다.

- (4) 진단검사의 경우는 지식 처리를 위해 기존 전자 의무기록 시스템으로부터 넘겨받는 데이터의 양이 방대하다는 특성을 갖고 있어, 데이터 인터페이스 어댑터를 통해 지식 실행 중에 필요한 지식을 실시간으로 얻어오는 아키텍처의 실효성을 검증할 수 있었다. 또한 이는 DIA를 통해 범용 EMR 구조 기반으로 저작한 지식의 상호운영성을 확보할 수 있음을 의미한다.
- (5) 지식의 추가, 삭제, 기준 값 변경 등이 저작환경에서 용이하다.

## 5. 결론

오늘날 지식기반 CDS 서비스는 사람의 생각을 모사하는 전문가시스템 접근에서 시작되었다. 잘 개발되고 검증된 지식기반이나 모델이 실무에 널리 이해되고, 수용되기 위해서는 CDS 시스템 표준화와 표준기반의 지식을 실행 가능한 형태로 해석하고 변환할 수 있는 도구 지원이 필요하다.

CDS 시스템 표준화는 다수 의료기관간의 지식 공유와 재사용을 촉진함으로써 지식 확보와 배포의 생산성을 높일 수 있고, 의료기관별 다른 플랫폼과 애플리케이션 환경으로 인해 현지화에 소요되는 추가 노력을 많이 감소시킬 수 있다.

본 연구에서는 국가 차원의 표준 의료정보시스템 구축 사업의 일부로 CDS 시스템 아키텍처를 제안하였다. 핵심 컴포넌트를 구현하여 실제 병원 실무에 적용할 수 있도록 하였다.

3개 대형 병원에 서로 다른 지식을 기반으로 한 CDS 서비스 구현에 적용함으로써 그 실무 적용성을 검증하였다. 본 논문이 제안한 CDS 시스템 아키텍처는 지식기반과 비지식기반 CDS 시스템을 구현하는데 모두 참조 가능한 모델이다.

## 참고 문헌

- [1] 조인숙, 김정아 외, " 임상의사결정지원시스템 개발 요약 보고서", 2010, 엘스비어코리아.
- [2] M. J. Ball, "Back to the future: what have we failed to learn? How does the future look?," in Medinfo2010. CapeTown: International Medical Informatics Association, 2010.
- [3] Samson Tu, Julie Glasgow, "SAGE Guideline Model Technical Specification," 2006
- [4] Kawamoto, K., & Lobach, DF. Design, implementation, use, and preliminary evaluation of SEBASTIAN, a standards-based Web service for clinical decision support. AMIA Annu Symp Proc. 2005;380-4.
- [5] JeongAh Kim, In sook Cho, Yoon Kim, CDSS (Clinical Decision Support System) Architecture in Korea, proceedings of conference ICHIT,2008
- [6] 김정아, 조인숙, 김윤, "EHR 관점에서 수립한 CDSS 아키텍처와 적용 사례," 소프트웨어 공학회 하계학술발표대회, 2009
- [7] Jerome A. Osheroff, et al, Clinical Decision Support Impelementer’s Workbook, , HIMSS, 2004
- [8] Kensaku Kawamoto, Service Functional Model Specification - Decision Support Service (DSS), 2006
- [9] Lee, J. H., Kim, J. A., Cho, I. S., Kim, Y., Integration of Workflow and Rule Engines for Clinical Decision Support Services, MedInfo 2010 Conference, Cafetown
- [10] JeongAh Kim, et al., Translation Protégé Knowledge for Executing Clinical Guidelines, proceedings of conference Protege 2009

**저자소개**

<관심분야> 임상 의사결정지원, 의료정보 표준,  
임상 데이터 마이닝, 간호정보, 전자  
의무기록

**김 정 아**

1994년 중앙대학교 컴퓨터공학과 (공학박사)  
1996~현재 관동대학교 컴퓨터 교육과 교수  
2000~현재 컴퓨터교육학회 이사  
2008~현재 보안공학회 이사

<관심분야> 아키텍처 기반 소프트웨어 설계, 제품  
라인 공학 방법론, 프로세스 개선,  
임상 의사결정지원,

**조 인 숙**

2002 서울대학교 간호대학 (간호학 박사)  
2003 서울대학교 간호대학 간호과학연구소  
선임연구원  
2004 University of Utah 박사후 과정  
(임상정보학)  
2005~현재 인하대학교 의과대학 간호학과 교수  
2005~2010 EHR핵심공통기술연구개발사업단  
CDSS 세부과제 책임  
2011~현재 대한의료정보학회 이사

# ASN.1을 이용한 가변 길이 메시지 표현 방법<sup>†</sup>

(An approach to define variable length messages using ASN.1)

백하은\*      강성원\*\*      김진규\*\*\*      김정민\*\*\*\*  
(Haeun Baek) (Sungwon Kang) (Jingyu Kim) (Jungmin Kim)

권구형<sup>†</sup>      김상수<sup>‡</sup>  
(Koohyung Kwon) (Sangsoo Kim)

**요 약** Variable Message Format(VMF)는 제한된 대역폭을 가지는 통신 환경에서 전송 정보를 효율적으로 교환하기 위하여 개발된 통신 메시지 표준으로, 지시자의 사용을 통해 단지 요구된 정보만을 전송할 수 있도록 메시지 길이 및 구조를 가변적으로 정의할 수 있도록 설계되었다. 그러나 이러한 가변성은 메시지를 효과적으로 분석하고 그 의미를 추출하는 것을 어렵게 할 뿐만 아니라, 메시지의 추가나 변경 시에 메시지 처리와 관련된 소프트웨어를 매번 수정하고, 이를 사용하는 모든 기기에 다시 배포해야 하는 번거로움이 있다. 본 논문에서는 국제 표준의 정형 기법인 ASN.1을 이용하여 VMF 메시지를 표현하는 체계적인 방법을 제안한다. 비트열 형태의 VMF 메시지를 구조적인 ASN.1 자료 구조로 표현함으로써 프로토콜 설계자는 고 수준에서 메시지의 구조와 그 값을 다룰 수 있으며, ASN.1을 지원하는 다양한 도구들을 사용할 수 있다. 제안하는 방법은 VMF 메시지 세그먼트들에 대해 템플릿을 정의하고, 이를 조합하여 다양한 VMF 메시지를 표현한다.

**키워드** 가변 길이 메시지, VMF, 추상 구문 표기법, ASN.1

**Abstract** Variable Message Format(VMF) is a communications protocol that allows computers to exchange tactical military information over low-data-rate bearers. VMF is designed to define message length and structure variably in order to allow the user to transmit only the relevant information by using indicators. However, flexibility of messages makes it difficult to analyze messages structure and figure out meanings. Furthermore, whenever messages are added or modified, message processing software should be updated and distributed to systems. In this paper, we propose a systematic approach to defining VMF messages that uses the international standard notation ASN.1. By describing VMF messages in ASN.1, protocol architects can deal with VMF messages accurately and conveniently, and use various ASN.1 tool. This paper develops ASN.1 description templates for VMF message segments, and presents an approach to describing VMF messages in ASN.1 by combining the templates.

**Key words** Variable length message, Variable Message Format, Abstract Syntax Notation One

<sup>†</sup> 본 연구는 국방과학연구소에서 지원하는 ‘실시간 전송정보 처리를 위한 최적화 방안 연구’ 위탁연구과제로 수행되었습니다.

\* 학생회원 : 한국과학기술원 전산학과, haeun@kaist.ac.kr

\*\* 종신회원 : 한국과학기술원 전산학과 교수,  
sungwon.kang@kaist.ac.kr

\*\*\* 학생회원 : 한국과학기술원 전산학과, jinlooks@kaist.ac.kr

\*\*\*\* 학생회원 : 한국과학기술원 전산학과, jmin11@kaist.ac.kr

<sup>†</sup> 비 회원 : 국방과학연구소, koohyung@add.re.kr

<sup>‡</sup> 비 회원 : 국방과학연구소, plus@add.re.kr

## 1. 서론

Variable Message Format(VMF)는 제한된 대역폭을 가지는 전송 통신 환경에서 실시간/근실시간으로 전송 정보를 교환하기 위해 제안된 비트 단위의 가변길이 메시지에 대한 디지털 정보 교환 표준이다[1]. VMF 메시지는 비트 열로 표현되며, 메시지 내의 인접한 데이터 필드들은 공백 문자 없이 연결된다. 또한 지시자의 값에 따라 뒤따르는 필드의 존재 및 반복 여부가 결정되도록 함으로써, 정보 요청 시 단지 요구된 의미 있는 정보만을 전송할 수 있도록 하는 유연성을 제공한다.

그러나 이러한 특징은 지시자들을 순서대로 해석해 나가지 않고서는 VMF 메시지의 길이와 구조의 예상을 불가능하게 할 뿐만 아니라, 메시지를 효과적으로 분석하고 그 의미를 추출하는 것을 어렵게 한다. 또한 신규 메시지의 추가나 기존 메시지의 변경이 이루어질 때에는, 이들을 정확하고 신속하게 지상 전송 통신 환경에 반영하기 위해 파서 소프트웨어를 비롯한 전송 정보 처리 소프트웨어를 매번 수정하고, 이를 사용하는 모든 기기에 다시 배포해야 하는 어려움이 있다.

ASN.1은 추상 구문 표기법(Abstract Syntax Notation One)의 약어로, 국제 전기 통신 연합 전기 통신 표준화 부문(ITU-T)에서 정의한 네트워크 상에서 전송되는 데이터를 기술하기 위한 정형기법이다[4]. ASN.1은 데이터를 정수 타입(Integer), 비트열 타입(Bit stream) 등의 데이터 타입 및 데이터 값에 따라 정의하고, 다양한 데이터 타입을 조합시켜 복잡한 데이터 구조를 기술할 수 있도록 한다. ASN.1은 여러 분야에서

활용되고 있으며, 자동화 도구들이 많이 개발되어 있다[4-8].

VMF 메시지를 ASN.1으로 표현하게 되면 비트열 형태의 VMF 메시지를 구조적인 데이터 구조로 기술할 수 있으며, 이에 따라 고 수준에서 VMF 메시지의 구조와 그 값을 정확하고 편리하게 다룰 수 있다. 또한, 메시지의 추가나 변경 시에 소프트웨어를 새로 만들거나 수정할 필요 없이 해당 메시지 명세만 ASN.1으로 표현하여 주기만 하면 TTCN-3 기반의 테스트 도구와 같이 ASN.1을 지원하는 다양한 도구들을 사용할 수 있다.

본 논문에서는 VMF 메시지 명세를 국제 표준의 정형기법인 ASN.1을 이용하여 표현하는 체계적인 방법을 제안한다. 이를 위해, 먼저 VMF 메시지의 가변성을 이루는 메시지 명세 세그먼트들에 대해 ASN.1으로 표현하는 템플릿을 정의한다. 이어서 앞서 정의한 템플릿을 조합하여 하나의 VMF 메시지 명세를 ASN.1으로 표현하는 방법을 제안한다. 메시지를 세그먼트 단위로 분할하여 각각에 대해 템플릿을 적용함으로써, VMF 메시지의 구조나 길이에 영향을 받지 않고 다양한 VMF 메시지 각각에 대해서 ASN.1 표현을 정의할 필요 없이 템플릿의 조합으로 쉽게 표현할 수 있다.

본 논문의 구성은 다음과 같다: 2장에서는 VMF 메시지의 구조와 ASN.1 표기법에 대해 논의한다. 3장에서는 VMF 메시지를 ASN.1으로 표현하는 방법을 제안한다. 4장에서는 3장에서 제안하는 방법을 이용하여 특정 메시지 명세 예제를 ASN.1으로 표현하고, 마지막으로 5장에서는 결론을 도출한다.



## 2. 배경 연구

이 장에서는 VMF 메시지를 ASN.1으로 표현 하는데 있어 주목할 만한 VMF 메시지 구조와 VMF 메시지 명세 표현에 이용되는 ASN.1 문법에 대해 정리한다.

### 2.1 VMF 메시지의 구조

VMF 메시지는 필드들의 나열로, 정보필드와 지시자라 불리는 문법필드의 조합으로 구성된다. 여기서 정보 필드는 일반 자료 정보를 담고 있는 필드로, 해당 필드 의미와 기능에 따라 각기 다른 길이의 비트 열로 구성된다. 문법 필드는 정보 필드의 존재 혹은 반복 여부를 나타내는 필드로 1 비트의 길이를 가진다. 문법 필드는 자료 요소의 존재 여부를 나타내는 존재 지시자(Presence Indicator)와 자료 요소의 반복 여부를 나타내는 반복 지시자(Recurrence Indicator)로 나뉘며, 지시자의 종류에는 FPI(Field Presence Indicator), GPI (Group Presence Indicator), FRI (Field Recurrence Indicator), 그리고 GRI (Group Recurrence Indicator) 가 있다[1, 2, 3].

VMF 메시지의 가변성은 지시자와 지시자의 값에 따른 정보 필드의 존재 및 반복 여부에 의해 구현된다. 지시자의 값에 따라 뒤따르는 필드가 포함되지 않을 수도 있고, 여러 번 반복되어 포함될 수도 있다. 그림 1은 각 지시자로부터 정의되는 VMF 메시지의 가변성 명세 메커니즘을 설명한다. 그림 1에서 ‘[]’기호는 선택적(optional) 필드를 나타낸다. ‘val(I)’는 지시자 I의 값을 의미하며, ‘0’ 또는 ‘1’의 값을 가질 수 있다. ‘(F)+’는 필드 F가 하나 또는 그 이상 반복됨을 나타내며, ‘(F)\*’는 필드 F가 0회 또는 그 이상 반복됨을 나타낸다[2].

#### (A) Field Presence Indicator (FPI)

FPI 다음에 연속되는 하나의 자료요소의 존재 여부를 정의한다.

사용: FPI + [DATA\_FIELD]

→ If val(FPI) = 1 then DATA\_FIELD must exist.

→ If val(FPI) = 0 then DATA\_FIELD must not exist.

#### (B) Group Presence Indicator (GPI)

GPI 다음에 여러 개의 필드로 구성된 하나의 그룹의 존재

여부를 정의한다. 사용: GPI + GROUP\*

→ If val(GPI) = 1 then GROUP\* must exist.

→ If val(GPI) = 0 then GROUP\* must not exist.

#### (C) Field Recurrence Indicator (FRI)

FRI 다음에 하나의 자료요소가 반복됨을 정의한다.

사용: (FRI + [DATA\_FIELD])+

→ If val(FRI) = 1 then DATA\_FIELD must exist and ‘FRI + [DATA\_FIELD]’ must be repeated.

→ If val(FRI) = 0, then its associated DATA\_FIELD must not exist and ‘FRI + [DATA\_FIELD]’ must not be repeated any longer.

#### (D) Group Recurrence Indicator (GRI)

GRI 다음에 하나의 반복 그룹 구조가 반복됨을 정의한다.

사용: (GRI + GROUP\*)+

→ If val(GRI) = 1 then GR\* must exist and ‘(GRI + GROUP\*)’ must be repeated.

→ If val(GRI) = 0, then GR\* must not exist and ‘(GRI + GROUP\*)’ must not be repeated any longer.

(where GROUP = DATA\_FIELD | FPI + [DATA\_FIELD] | GPI + GROUP\* | (FRI + [DATA\_FIELD])+ | (GRI + GROUP\*)+)

그림 1. VMF 메시지의 가변성 명세 메커니즘

본 논문에서는 VMF 메시지의 가변성을 구현 하는 지시자를 기준으로 VMF 메시지 명세 세그먼트를 정의하고, 주어진 VMF 메시지 명세를 구성하는 각 세그먼트들에 대해 ASN.1 표현 템플릿을 정의 및 적용함으로써 ASN.1 표현을 완성한다.

### 2.2 ASN.1 표기법

ASN.1 표기법은 정수나 비트열과 같은 기본적인 데이터 타입에서부터 집합(Set)이나 순서열(Sequence)과 같은 복합적인 데이터 타입에 이르기 까지 다양한 기본 데이터 타입을 정의함은 물론, 이들 기본 데이터 타입을 조합하여 새로운 데이터

타입을 정의할 수 있도록 한다. 따라서 프로토콜 설계자는 ASN.1을 이용함으로써 통신 매체를 통해 교환되는 비트나 바이트와 같은 형태의 정보를 구조적인 데이터 형태로 기술할 수 있으며, 고 수준에서 정보의 구조와 값을 다룰 수 있다.

컴퓨터 간에 교환되는 정보는 정보의 종류 및 형태를 나타내는 타입과 정보의 내용이자 실체인 값으로 구성된다. ASN.1은 정보 데이터를 타입과 값으로 정의하며, 각각에 대한 표기법을 제공한다. 새로운 데이터 타입을 정의하는 것을 타입 배정, 데이터 타입의 객체에 값을 지정하는 것을 값 배정이라고 한다[8].

예를 들어 '나이 25세'란 정보를 ASN.1으로 표현한다면, 다음과 같은 타입 배정과 값 배정을 생각할 수 있다.

```
타입 배정문) Age ::= Integer
값 배정문) myAge Age ::= 25
```

위의 타입 배정문은 나이를 표시하기 위한 새로운 데이터 타입 Age 에 먼저 정의되어 있는 데이터 타입인 Integer를 배정하는 것을 나타낸다. 이로써 나이를 표현하기 위한 데이터 타입으로 Age를 참조하는 것이 가능해진다. 이어지는 값 배정문은 myAge라는 이름의 변수에 Age라는 데이터 타입의 값 25를 배정하는 것을 나타내며, 이로써 나이 값 25를 myAge로 참조하는 것이 가능하게 된다.

본 논문에서는 ASN.1의 SEQUENCE 타입과 CHOICE 타입을 조합하여 새로운 데이터 타입을 정의함으로써 VMF 메시지를 표현한다.

SEQUENCE 타입: 0개 이상의 ASN.1 데이터 타입들의 나열을 나타내는데 사용되며, 컴포넌트들의 나열 순서가 의미를 가진다[4].

CHOICE 타입: 하나 이상의 데이터 타입들의 공유형으로, 여러 개의 컴포넌트들 중 하나를 선택

할 수 있도록 한 데이터 타입이다. 후보 컴포넌트들 중 하나를 선택해내기 위해서는 컴포넌트들의 데이터 타입들이 상호 식별 가능해야 한다. 이들 중 같은 데이터 타입을 가지는 후보 컴포넌트가 있는 경우에는 명시적으로 태그를 붙여 구분할 수 있도록 한다[4].

ASN.1은 특정 장치, 데이터 표현방식, 프로그래밍 언어, 하드웨어 플랫폼에 종속되지 않고 데이터를 정의할 수 있다. ASN.1을 지원하는 도구들은 거의 모든 운영체제에서 동작가능하며, JAVA, C, C#에서부터 COBOL에 이르기까지 다양한 프로그래밍 언어로의 변환을 지원한다. ASN.1은 여러 분야에서 활용되고 있으며, 이를 위한 자동화 도구들이 많이 개발되어 있다[4-8].

### 3. ASN.1을 이용한 VMF 메시지의 가변성 표현 방법

이 장에서는 ASN.1을 이용하여 VMF 메시지를 표현하는 체계적인 방법을 제안한다. VMF 메시지는 메시지 내의 자료 항목들의 구성을 정의한 메시지 양식 표준인 메시지 명세와 실제 전송 정보 데이터를 담고 있는 비트열 형태의 VMF 메시지 인스턴스(instance)로 구분할 수 있다. 이에 따라, VMF 메시지 명세는 ASN.1의 타입 배정으로, VMF 메시지 인스턴스는 ASN.1의 값 배정으로 각각 대응시켜 표현할 수 있다. VMF 메시지 인스턴스를 ASN.1으로 표현하는 것은 해당 메시지 인스턴스에 대한 메시지 명세의 ASN.1 타입 배정이 주어진다면, 이에 대해 값을 넣어주어 값 배정문으로 작성하는 것으로 자명하다.

따라서, 이 장에서는 VMF 메시지 명세를 ASN.1의 타입 배정문으로 표현하는 방법을 제안한다. 이를 위해, 먼저 VMF 메시지 명세 세그먼트

들에 대해 ASN.1으로 표현하는 템플릿을 정의한다. 그리고 특정 VMF 메시지 명세가 주어지면, 앞서 정의한 템플릿을 이용하여 ASN.1으로 표현하는 방법을 제시한다.

### 3.1 VMF 메시지 명세 세그먼트에 대한 ASN.1 표현 템플릿 정의

제 2장에서 언급한 바와 같이, VMF 메시지의 가변성은 네 개의 지시자에 의해 구현된다: FPI,

표 1. VMF 메시지 명세 세그먼트에 대한 ASN.1 표현 템플릿

VMF 메시지 명세 세그먼트					ASN.1 표현 템플릿
<b>(A) 세그먼트-FPI</b>					<pre> SEG-FPI ::= CHOICE {   noPresence          [0]FPI-N,   presence            [1]FPI-P }  FPI-N ::= SEQUENCE {   fpi          BIT STRING('0'B) }  FPI-P ::= SEQUENCE {   fpi          BIT STRING('1'B),   idx1-1      BIT STRING(SIZE(n)) }                     </pre>
인덱스	자료항목	비트수	그룹코드	반복코드	
1.	FPI	1			
1.1	data	n			
<b>(B) 세그먼트-GPI</b>					<pre> SEG-GPI ::= CHOICE {   noPresence          [0]GPI-N,   presence            [1]GPI-P }  GPI-N ::= SEQUENCE {   gpi          BIT STRING('0'B) }  GPI-P ::= SEQUENCE {   gpi          BIT STRING('1'B),   Gn          SEG-Gn }  SEG-Gn ::= SEQUENCE {   idx1-1      BIT STRING(SIZE(n1)),   idx1-2      BIT STRING(SIZE(n2)) }                     </pre>
인덱스	자료항목	비트수	그룹코드	반복코드	
1.	GPI	1			
1.1	data1	n1	Gn		
1.2	data2	n2	Gn		
<b>(C) 세그먼트-FRI</b>					<pre> SEG-FRI ::= CHOICE {   noRecurrence        [0]FRI-N,   recurrence           [1]FRI-P }  FRI-N ::= SEQUENCE {   fri          BIT STRING('0'B),   idx1-1      BIT STRING(SIZE(n)) }  FRI-P ::= SEQUENCE {   fpi          BIT STRING('1'B),   idx1-1      BIT STRING(SIZE(n)),   seg-fri     SEG-FRI }                     </pre>
인덱스	자료항목	비트수	그룹코드	반복코드	
1.	FRI	1			
1.1	data	n			
<b>(D) 세그먼트-GRI</b>					<pre> SEG-GRI ::= CHOICE {   noRecurrence        [0]GRI-N,   recurrence           [1]GRI-P }  GRI-N ::= SEQUENCE {   gri          BIT STRING('0'B),   Rn          SEG-Rn }  GRI-P ::= SEQUENCE {   gri          BIT STRING('1'B),   Rn          SEG-Rn,   seg-gri     SEG-GRI }  SEG-Rn ::= SEQUENCE {   idx1-1      BIT STRING(SIZE(n1)),   idx1-2      BIT STRING(SIZE(n2)) }                     </pre>
인덱스	자료항목	비트수	그룹코드	반복코드	
1.	GRI	1			
1.1	data1	n1		Rn	
1.2	data2	n2		Rn	

GPI, FRI, 그리고 GRI. 이에 따라, VMF 메시지를 각 지시자를 기준으로 지시자와 지시자에 의해 그 존재 및 반복 여부를 제한 받는 정보 필드들을 하나의 세그먼트 단위로 구분 지어 분할해낼 수 있다. 표 1은 각 VMF 메시지 명세 세그먼트에 대한 ASN.1 표현 템플릿을 정의한 것이다.

VMF 메시지 명세로부터 VMF 메시지 인스턴스를 생성하게 되면 지시자의 값에 따라 뒤따르는 필드들의 구성이 달라진다. 이러한 가변성은 ASN.1의 CHOICE 타입을 이용하여 표현할 수 있다. 표 1에서 정의한 ASN.1 표현 템플릿에서는 태그 값에 의해 각 세그먼트들의 첫 번째 필드인 지시자의 데이터 값이 '0'인 경우에는 'noPresence' 또는 'noRecurrence' 컴포넌트가, '1'인 경우에는 'presence' 또는 'recurrence' 컴포넌트가 선택된다. 각 컴포넌트들은 필드들의 나열로 SEQUENCE 타입으로 나타낸다.

지시자 GPI와 GRI에 의해 그 존재 및 반복 여부를 제한 받는 하위 필드들의 그룹은 표 1의 (B)세그먼트-GPI의 SEG-Gn과 같이 그룹 코드 또는 반복 코드 이름으로 새로운 데이터 타입을 선언하고, 여기에 SEQUENCE 타입을 배정하여 정의한다.

반복 지시자 FRI와 GRI에 대한 세그먼트의 경우, 지시자의 값이 '1'인 경우에 전체 구조가 재귀적으로 반복된다. 따라서 표 1의 (C)세그먼트-FRI의 seg-fri와 같이 'presence' 또는 'recurrence' 컴포넌트의 마지막에 세그먼트의 데이터 타입을 가지는 컴포넌트를 두어 그 구조가 재귀적으로 반복되어 나타날 수 있도록 정의한다.

정보 필드들은 ASN.1 이 제공하는 기본 데이터 타입인 BIT STRING 타입으로 나타내며, 'SIZE' 키워드를 이용하여 각 필드의 비트 수를 정의한다. 표 1에서 굵게 표시된 구문은 주어진 메시지 명세의 구성에 따라 다르게 정의되는 내용들이다.

### 3.2 특정 VMF 메시지 명세를 ASN.1으로 표현하는 방법

ASN.1으로 표현하고자 하는 VMF 메시지 명세가 주어지면, 3.1절에서 정의한 표 1의 템플릿들을 조합하여 하나의 ASN.1 모듈로 표현할 수 있다. 하나의 VMF 메시지 명세를 ASN.1으로 표현하는 방법은 다음의 세 단계로 이루어진다.

- 단계 1) VMF 메시지 명세의 계층 구조 분석
- 단계 2) VMF 메시지 명세를 메시지 명세 세그먼트 단위로 분할
- 단계 3) ASN.1 표현 템플릿의 적용

#### 단계 1) VMF 메시지 명세의 계층 구조 분석

VMF 메시지는 필드들의 나열로 필드들 간에 계층 구조를 가지며, 따라서 이를 트리 구조로 나타낼 수 있다. 지시자는 연속되는 하위 필드의 존재 혹은 반복 여부를 나타내는 필드로, 지시자에 의해 존재 혹은 반복 여부를 제한 받는 하위 필드들을 자식(Child) 노드로 가지게 된다. 각 하위 필드 역시 트리 구조로 나타낼 수 있다.

GPI는 여러 개의 필드로 구성된 하나의 그룹이 존재하는지 여부를 나타내는 지시자로, GPI에 의해 그 존재 여부를 제한 받는 하위 필드들은 GPI와 같은 그룹 코드를 가진다. FRI와 GRI는 반복 지시자로 하나 이상의 필드들의 반복 여부를 나타낸다. FRI와 GRI의해 반복 여부를 제한 받는 하위 필드들은 반복 코드로 구분된다. FPI 지시자의 경우, FPI 필드 다음에 연속되는 하나의 필드가 FPI의 하위 필드이다. 동일한 부모(Parent) 노드를 가지는 필드들은 트리 구조 상에서 같은 레벨에 있다고 할 수 있으며, 메시지 그 자체는 레벨 0이다. 모든 정보 필드들은 잎 노드(Leaf Node)가 된다.

### 단계 2) VMF 메시지 명세를 메시지 명세 세그먼트 단위로 분할

3.1절에서 정의한 ASN.1 표현 템플릿을 적용하기 위해 VMF 메시지 명세를 메시지 명세에 포함된 지시자들을 기준으로 메시지 명세 세그먼트 단위로 분할한다. 하나의 세그먼트는 부모 노드인 지시자와 지시자의 자식 노드들(지시자보다 레벨이 1만큼 큰 필드들)인 정보 필드들로 구성된다.

부모 노드인 지시자의 레벨을 각 세그먼트의 레벨로 정의한다. 정보 필드들은 모두 앞 노드가 되므로 부모 노드가 될 수 있는 것은 지시자와 루트 노드(Root Node)인 메시지 자체이다. 이 때, 루트 노드인 메시지 그 자체는 세그먼트를 구성하는 필드로 포함시키지 않고, 자식 노드들만 포함하도록 한다.

### 단계 3) ASN.1 표현 템플릿의 적용

단계 2에서 얻은 세그먼트들에 대해서, 3.1절에서 정의한 ASN.1 표현 템플릿을 적용한다. 이 때, 상위 레벨의 세그먼트들부터 차례로 반복적으로(Recursively) 패턴을 적용해 나간다.

레벨이 0인 세그먼트는 부모 노드로 메시지 그 자체를 가지며, 특정 지시자에 의해 구분된 세그먼트가 아닌 필드들의 단순한 나열이다. 따라서 메시지의 최상위 레벨에 해당하는 이 세그먼트는 ASN.1의 SEQUENCE 타입으로 정의한다.

세그먼트의 자식 노드가 또 다시 트리 구조를 가지는 경우, 즉 자식 노드가 지시자인 경우, 해당 지시자의 ASN.1 타입을 새롭게 선언한다. 그리고 새롭게 선언된 타입에 대해서는, 패턴들을 반복적으로 적용해 나가는 중에, 해당 자식 노드를 부모 노드로 가지는 세그먼트를 표현할 때 정의하게 된다.

## 4. VMF 메시지 명세의 ASN.1 표현 예제

이 장에서는 3장에서 제시한 방법을 이용하여 VMF 메시지 명세를 ASN.1 으로 표현하는 예제를 수행한다. 그림 2와 같은 VMF 메시지 명세를 생각해 볼 수 있다.

인덱스	자료항목	비트수	그룹코드	반복코드
1.	GPI	1	G1	
1.1	Year	7	G1	
1.2	Month	4	G1	
1.3	Day	5	G1	
2.	GPI	1	G2	
2.1	GRI	1	G2	R1
2.2	Type	6	G2	R1
2.3	FPI	1	G2	R1
2.3.1	Identification	4	G2	R1
2.4	Status	4	G2	R1
2.5	FRI	1	G2	R1
2.5.1	FRI	1	G2	R1/R2
2.5.2	Target number	28	G2	R1/R2
3.	FPI	1		
3.1	Comment	1400		

그림 2. VMF 메시지 명세 예제

### 단계 1) VMF 메시지 명세의 계층 구조 분석

메시지 자체는 레벨이 0인 루트 노드가 된다. 첫 번째 필드인 1. GPI는 레벨 1이 되고, 1. GPI와 같은 그룹 코드 G1을 가지는 세 개의 하위 필드(1.1 Year, 1.2 Month, 1.3 Day)는 1. GPI의 자식 노드가 된다. 같은 방식으로 2. GPI는 레벨 1의 노드가 되고, 2. GPI의 자식 노드로 2. GPI에 의해 그 존재 여부를 제한 받는 그룹 코드가 G2인 하위 필드들을 생각할 수 있다. 이 때, 이 하위 필드들 모두가 반복 코드 R1을 가지며, 2.1 GRI에 의해 그 반복 여부를 제한받는다. 이렇게 여러 지시자에 의해 그 존재 및 반복 여부를 제한 받는 필드들은 자신의 가장 인접한 지시자를 부모 노드로 가진다.

즉, 지시자에 뒤따르는 하위 필드들 중에 지시자가 존재하면(이를 하위 지시자라고 하자), 하위 지시자의 영향을 받는 필드들은 하위 지시자의 자식노드가 되고, 하위 지시자만이 지시자의 자식노드가 된다. 이에 따라, 2. GPI의 자식 노드는 2.1 GRI가 되고, 나머지 하위 필드들은 2.1 GRI의 자식노드가 된다. 이하 필드들에 대해서도 마찬가지로 반복적으로 진행한다. 그림 2의 메시지 명세에 대한 계층 구조를 분석한 결과는 표 2와 같다.

표 2. 단계 1-VMF 메시지 예제의 계층 구조

	Level. 1	Level. 2	Level. 3	Level. 4	Level. 5		
message	1. GPI	1.1 Year					
		1.2 Month					
		1.3 Day					
	2. GPI	2.1 GRI	2.2 Type	2.3 FPI	2.3.1 Identification		
						2.4 Status	
			2.5 FPI	2.5.1 FRI	2.5.2 Target number		
	3. FPI	3.1 Comment					

**단계 2) VMF 메시지 명세를 메시지 명세 세그먼트 단위로 분할**

표 2로부터 레벨이 1만큼 차이 나는 부모 자식 관계의 필드들을 하나의 세그먼트 단위로 분할해내고, 부모 노드인 지시자의 레벨을 세그먼트의 레벨로 정의한다.

메시지 그 자체는 레벨이 0이므로, 가장 상위 레벨을 가지는 세그먼트는 표 3의 SEG1로 메시지 그 자체와 레벨 1인 노드 1.GPI, 2.GPI, 3.FPI로 구성된다. 다음으로, 레벨 1에 있는 세 개의 지시자들을 기준으로 각각 세 개의 세그먼트를 분할해 낼 수 있다. 1. GPI를 부모 노드로 가지는 세그

먼트는 1. GPI와 하위 필드 1.1 Year, 1.2 Month, 1.3 Day로 구성되고, 세그먼트 레벨은 1이 된다. 이는 표 3의 SEG2에 해당한다.

나머지 노드들에 대해서도 각 레벨에 있는 지시자를 기준으로 분할을 순차적으로 진행한 결과는 표 3과 같으며, 그림 2의 메시지 명세 예제에 대해서 총 8개의 세그먼트들을 분할해내었다.

표 3. 단계 2-VMF 메시지 예제의 메시지 명세 세그먼트

세그먼트	레벨	인덱스	자료항목	비트수	그룹 코드	반복 코드
SEG1	0	1.	GPI	1	G1	
		2.	GPI	1	G2	
		3.	FPI	1		
SEG2	1	1.	GPI	1	G1	
		1.1	Year	7	G1	
		1.2	Month	4	G1	
		1.3	Day	5	G1	
SEG3	1	2.	GPI	1	G2	
		2.1	GRI	1	G2	R1
SEG4	1	3	FPI	1		
		3.1	Comment	1400		
SEG5	2	2.1	GRI	1	G2	R1
		2.2	Type	6	G2	R1
		2.3	FPI	1	G2	R1
		2.4	Status	4	G2	R1
		2.5	FPI	1	G2	R1
SEG6	3	2.3	FPI	1	G2	R1
		2.3.1	Identification	4	G2	R1
SEG7	3	2.5	FPI	1	G2	R1
		2.5.1	FRI	1	G2	R1/R2
SEG8	4	2.5.1	FRI	1	G2	R1/R2
		2.5.2	Target number	28	G2	R1/R2

**단계 3) ASN.1 표현 템플릿의 적용**

표 3에서 얻은 8개의 세그먼트들에 대해서, 상위 레벨의 세그먼트들부터 차례로 3.1절에서 정의한 ASN.1 표현 템플릿을 적용한다. 우선 레벨이 0인 메시지 그 자체에 대한 세그먼트 SEG1은

SEQUENCE 타입으로 정의한다. 나머지 세그먼트들은 각 세그먼트의 첫 번째 필드인 지시자를 기준으로 표 1의 템플릿을 선택하여 적용한다.

이 때, SEG1의 첫 번째 필드인 1. GPI는 지시자로 하위 노드들을 가진다. 따라서, 1. GPI에 해당되는 컴포넌트의 타입을 새로운 이름을 가지는 타입으로 선언하고, 이에 대해서는 1. GPI를 부모 노드로 가지는 SEG2를 표현할 때 타입 배정을 한다. 즉, 표 4의 세그먼트 SEG1에 대한 ASN.1 표현에서 1. GPI에 대한 컴포넌트 `idx1`을 `SEG-IDX1-GPI` 라는 새로운 타입으로 선언한다. 그리고 1. GPI를 부모 노드로 가지는 세그먼트 SEG2를 표현할 때, SEG2의 타입을 `SEG-IDX1-GPI` 타입으로 정의하고 표 1의 세그먼트-GPI 템플릿을 적용한다.

나머지 각 세그먼트에 대해서도 템플릿을 적용하여 ASN.1으로 표현해나간다. 최종적으로, 각 세그먼트에 대한 ASN.1 표현을 연결하면 주어진 VMF 메시지 명세에 대한 ASN.1 표현이 완성된다. 그림 2의 VMF 명세를 ASN.1 으로 표현한 결과는 부록 I 에 나타나있다.

## 5. 결론

본 논문은 VMF 메시지를 국제 표준의 정형 기법인 ASN.1을 이용하여 표현하는 체계적인 방법을 제안하였다. VMF 메시지 명세를 ASN.1으로 정의함으로써, VMF 메시지를 비트열 형태로 부터 구조적인 자료 구조로 기술할 수 있으며 VMF 메시지의 구조와 그 값을 고 수준에서 정확하고 편리하게 다룰 수 있다. 또한, 신규 메시지의 추가나 기존 메시지의 변경 시에, 소프트웨어를 새로 만들거나 수정할 필요 없이 해당 메시지 명세만

ASN.1으로 표현하여 주기만 하면 ASN.1을 지원하는 다양한 도구들을 사용할 수 있어, 보다 정확하고 신속하게 지상 전술 통신 환경에 반영할 수 있다.

본 논문에서 제안하는 방법은 이를 위해 먼저 메시지의 가변성을 이루는 메시지 명세 세그먼트들에 대해 ASN.1으로 표현하는 템플릿을 정의하였다. 이어서 앞서 정의한 템플릿을 조합하여 하나의 VMF 메시지 명세를 ASN.1으로 표현하는 방법을 제시하였다. 제안하는 방법은 메시지를 세그먼트 단위로 분할하여 각각에 대해 템플릿을 적용함으로써, VMF 메시지의 구조나 길이에 영향을 받지 않고 다양한 VMF 메시지를 템플릿의 조합으로 쉽게 표현할 수 있다.

그러나 제안하는 방법은 ASN.1을 이용하여 VMF 메시지의 구문 정보만을 표현하는데 그치고 있다. 향후 연구에서는 ASN.1 또는 ASN.1과 연동하여 사용 가능한 정형 표현 기법을 이용하여 메시지 처리 규칙과 같은 VMF 메시지의 시맨틱 정보를 표현하는 방안을 연구할 필요가 있다.

## 참 고 문 헌

- [1] DoD (Department of Defense), MIL-STD-6017, Variable Message Format(VMF), 2002.
- [2] Kyungmin Go, Sungwon Kang, Myungchul Kim, and Jihyun Lee, "A Systematic Test Cases Generation Approach for Testing Message Length Variability", Fourth International Conference on Software Testing, Verification and Validation (ICST 2011), March 21-25, 2011, Berlin, Germany, 2011.

- [3] Jihyun Lee, Sungwon Kang, Myungchul Kim, Changsup Keum, and Kyungmin Go, "Test Generation for VMF Tactical Data Link Messages - Coping with Message Length Variability and Semantic Message Rules", The IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE 2011), Nov 29 - Dec 2, 2011 Hiroshima, Japan, 2011.
- [4] ITU-T, X.680: Information technology - Abstract Syntax Notation One(ASN.1): Specification of basic notation, 2002.
- [5] ITU International Telecommunication Union, <http://www.itu.int/ITU-T/asn1/>
- [6] Oliver Dubuisson, ASN.1 Communication between Heterogeneous Systems, OSS Nokalva, 2000.
- [7] John Larmouth, ASN.1 Complete, Open Systems Solutions, 1999.
- [8] 하수철, "ASN.1 추상구문표기법 닷원", 대전대학교, 정보통신부 정보통신 연구개발 사업 연구 결과, 2000.

## 저자 소개



백 하 은

2011년 한동대학교 졸업(학사).

2011년~현재 KAIST 전산학과 석사과정 재학 중



강 성 원

1982년 서울대학교 졸업(학사).

1989년 Univ. of Iowa Computer Science 졸업 (석사).

1992년 Univ. of Iowa Computer Science 졸업 (박사).

1993년~2001년 Korea Telecom Principle Researcher.

2001년~2005년 Assistant Professor, ICU.

2005년~2009년 2월 Associate Professor, ICU.

2009년 3월~현재 Associate Professor, KAIST.

2001년~현재 Adjunct Faculty of MSE Program, Carnegie-Mellon University, USA.





**김진규**

2006년 대구대학교 졸업(학사).  
2006년~현재 KAIST 전산학과 박사과정 재학 중



**김상수**

2003년 경북대학교 졸업(석사).  
2003년~현재 국방과학연구소 근무 중



**김정민**

2007년 숙명여자대학교 졸업(학사).  
2011년~현재 KAIST 전산학과 석사과정 재학 중



**권구형**

2003년 고려대학교 졸업(석사).  
2006년~현재 국방과학연구소 근무 중

부록 I. 단계3-VMF 메시지 예제의 ASN.1 표현

메시지 명세 예제					ASN.1 표현
<b>SEG1</b>					EX-MSG ::= SEQUENCE { idx1 SEG-IDX1-GPI, idx2 SEG-IDX2-GPI, idx3 SEG-IDX3-FPI }
인덱스	자료항목	비트수	그룹코드	반복 코드	
1.	GPI	1	G1		
2.	GPI	1	G2		
3.	FPI	1			
<b>SEG2</b>					SEG-IDX1-GPI ::= CHOICE { noPresence [0]IDX1-GPI-N, presence [1]IDX1-GPI-P } IDX1-GPI-N ::= SEQUENCE { gpi BIT STRING('0'B) } IDX1-GPI-P ::= SEQUENCE { gpi BIT STRING('1'B), g1 SEG-G1 } SEG-G1 ::= SEQUENCE { idx1-1 BIT STRING(SIZE(7)), idx1-2 BIT STRING(SIZE(4)), idx1-3 BIT STRING(SIZE(5)) }
인덱스	자료항목	비트수	그룹코드	반복 코드	
1.	GPI	1	G1		
1.1	Year	7	G1		
1.2	Month	4	G1		
1.3	Day	5	G1		
<b>SEG3</b>					SEG-IDX2-GPI ::= CHOICE { noPresence [0]IDX2-GPI-N, presence [1]IDX2-GPI-P } IDX2-GPI-N ::= SEQUENCE { gpi BIT STRING('0'B) } IDX2-GPI-P ::= SEQUENCE { gpi BIT STRING('1'B), g2SEG-G2 } SEG-G2 ::= SEQUENCE { idx2-1-gri SEG-IDX2-1-GRI }
인덱스	자료항목	비트수	그룹코드	반복코드	
2.	GPI	1	G2		
2.1	GRI	1	G2	R1	
<b>SEG4</b>					SEG-IDX3-FPI ::= CHOICE { noPresence [0]IDX3-FPI-N, presence [1]IDX3-FPI-P } IDX3-FPI-N ::= SEQUENCE { fpi BIT STRING('0'B) } IDX3-FPI-P ::= SEQUENCE { fpi BIT STRING('1'B), idx3-1 BIT STRING(SIZE(1400)) }
인덱스	자료항목	비트수	그룹코드	반복코드	
3	FPI	1			
3.1	Comment	1400			
<b>SEG5</b>					SEG-IDX2-1-GRI ::= CHOICE { noRecurrence [0]IDX2-1-GRI-N, recurrence [1]IDX2-1-GRI-P } IDX2-1-GRI-N ::= SEQUENCE { gri BIT STRING('0'B), r1 SEG-R1 } IDX2-1-GRI-P ::= SEQUENCE { gri BIT STRING('1'B), r1 SEG-R1, idx2-1-gri SEG-IDX2-1-GRI } SEG-R1 ::= SEQUENCE { idx2-2 BIT STRING(SIZE(6)), idx2-3-fpi SEG-IDX2-3-FPI, idx2-4 BIT STRING(SIZE(4)), idx2-5-fpi SEG-IDX2-5-FPI }
인덱스	자료항목	비트수	그룹코드	반복코드	
2.1	GRI	1	G2	R1	
2.2	Type	6	G2	R1	
2.3	FPI	1	G2	R1	
2.4	Status	4	G2	R1	
2.5	FPI	1	G2	R1	

메시지 명세 예제					ASN.1 표현
<b>SEG6</b>					SEG-IDX2-3-FPI ::= CHOICE { noPresence [0]IDX2-3-FPI-N, presence [1]IDX2-3-FPI-P } IDX2-3-FPI-N ::= SEQUENCE { fpi BIT STRING('0'B) } IDX2-3-FPI-P ::= SEQUENCE { fpi BIT STRING('1'B), idx2-3-1 BIT STRING(SIZE(4)) }
인덱스	자료항목	비트수	그룹코드	반복코드	
2.3	FPI	1	G2	R1	
2.3.1	Identification	4	G2	R1	
<b>SEG7</b>					SEG-IDX2-5-FPI ::= CHOICE { noPresence [0]IDX2-5-FPI-N, presence [1]IDX2-5-FPI-P } IDX2-5-FPI-N ::= SEQUENCE { fpi BIT STRING('0'B) } IDX2-5-FPI-P ::= SEQUENCE { fpi BIT STRING('1'B), idx2-5-1-fri SEG-IDX2-5-1-FRI }
인덱스	자료항목	비트수	그룹코드	반복코드	
2.5	FPI	1	G2	R1	
2.5.1	FRI	1	G2	R1 / R2	
<b>SEG8</b>					SEG-IDX2-5-1-FRI ::= CHOICE { noRecurrence [0]IDX2-5-1-FRI-N, recurrence [1]IDX2-5-1-FRI-P } IDX2-5-1-FRI-N ::= SEQUENCE { fri BIT STRING('0'B), idx2-5-2 BIT STRING(SIZE(28)) } IDX2-5-1-FRI-P ::= SEQUENCE { fri BIT STRING('1'B), idx2-5-2 BIT STRING(SIZE(28)), idx2-5-1-fri SEG-IDX2-1-4-1-FRI }
인덱스	자료항목	비트수	그룹코드	반복코드	
2.5.1	FRI	1	G2	R1 / R2	
2.5.2	Target number	28	G2	R1 / R2	





## 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](mailto:sypark@sogang.ac.kr) (박수용교수), [bjlee@uos.ac.kr](mailto:bjlee@uos.ac.kr) (이병정교수)

홈페이지 : <http://www.sigse-kiss.or.kr/>

### ◆ 회비 안내

회원구분	<ul style="list-style-type: none"> <li>• 학생회원 : IT 분야 학과 또는 관심 있는 학생</li> <li>• 정회원, 종신회원 : IT 분야 종사자</li> </ul>
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



# 개인회원용 입회원서



회원구분	학생회원 ( ) 정회원( ) 종신회원( )			
성명	한글		생년월일	
	영문			
연락처	직장전화		휴대전화	
	e-mail			
주소	직장명/ 부서		직급	
	직장주소	(우)		
학력	학사	년 월 - 년 월	대학교	과
	석사	년 월 - 년 월	대학원	과
	박사	년 월 - 년 월	대학원	과
관심분야				

본인은 한국정보과학회 소프트웨어공학소사이어티의 취지에 찬성하여 회원으로 가입하고자 이에 입회원서를 제출합니다.

년 월 일

신청인: (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 : 강성원 교수 (KAIST, 042-350-3512, sungwon.kang@kaist.ac.kr)
- 편집이사 : 윤희진 교수 (협성대학교, 031-299-0841, hgyoon@uhs.ac.kr)
- 편집이사 : 이관우 교수 (한성대학교, 02-760-5864, kwlee@hansung.ac.kr)
- 편집이사 : 채홍석 교수 (부산대학교, 051-510-3517, hscha@pusan.ac.kr)
- 편집이사 : 김문주 교수 (KAIST, 042-350-3543, moonzoo@cs.kaist.ac.kr)
- 편집위원 : 김정아 교수 (관동대학교, 033-649-7801, clara@kwandong.ac.kr)
- 편집위원 : 김현수 교수 (충남대학교, 042-821-6657, hskim401@cnu.ac.kr)
- 편집위원 : 이우진 교수 (경북대학교, 053-950-6378, woojin@mail.knu.ac.kr)
- 편집위원 : 박수진 교수 (서강대학교, psjdream@sogang.ac.kr)
- 편집위원 : 이지현 교수 (대전대학교, jihyun30@dju.ac.kr)
- 편집위원 : 최종무 교수 (단국대학교, choijm@dankook.ac.kr)
- 편집위원 : 김태호 박사 (ETRI, taehokim@etri.re.kr)



## 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](mailto: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

구분	성명	소속기관명	E-Mail	
회장	한혁수	상명대학교	hshan@smu.ac.kr	
부회장 (기획)	권기현	경기대학교	khkwon@kyonggi.ac.kr	
부회장 (편집)	강성원	KAIST	sungwon.kang@kaist.ac.kr	
부회장 (학술)	홍장익	충북대학교	jehong@chungbuk.ac.kr	
부회장 (조직)	이병걸	서울여자대학교	byongl@swu.ac.kr	
부회장 (협력)	전진옥	비트컴퓨터	jojeon@bit.co.kr	
운영위원회	총무이사	이정원	아주대학교	jungwony@ajou.ac.kr
		유준범	건국대학교	jbyoo@konkuk.ac.kr
	기획이사	이병정	서울시립대학교	bjlee@uos.ac.kr
		서주영	아주대학교	jyseo@ajou.ac.kr
	조직이사	백종문	KAIST	jbaik@kaist.ac.kr
		김영철	홍익대학교	bob@hongik.ac.kr
	학술이사	인호	고려대학교	hoh_in@korea.ac.kr
		고인영	KAIST	iko@kaist.ac.kr
	편집이사	윤회진	협성대학교	hjyoon@uhs.ac.kr
		이관우	한성대학교	kwlee@hansung.ac.kr
		채홍석	부산대학교	hschae@pusan.ac.kr
		김문주	KAIST	moonzoo@cs.kaist.ac.kr
	홍보이사	조은숙	서일대학교	escho@seoil.ac.kr
		이찬근	중앙대학교	cglee@cau.ac.kr
		박용범	단국대학교	ybpark@dankook.ac.kr
	협력이사	이세영	NIPA	sarahlee230@gmail.com
감사	차성덕	고려대학교	scha@korea.ac.kr	
	이상은	NIPA	selee@nipa.kr	
자문위원회	강교철	포항공과대학교	kck@postech.ac.kr	
	권용래	KAIST	kwon@cs.kaist.ac.kr	
	성기수	KISTI 고문	Kss13@truefriend.com	
	배두환	KAIST	bae@se.kaist.ac.kr	
	신규상	ETRI	gsshin@etri.re.kr	
	양승민	송실대학교	smyang@ssu.ac.kr	
	우치수	서울대학교	wuchisu@selab.snu.ac.kr	
	이경환	중앙대학교	kwlee@object.cau.ac.kr	
	이단형	KAIST	danlee@cs.kaist.ac.kr	
	정기원	송실대학교	chong@comp.ssu.ac.kr	
	박수용	서강대학교	sypark@sogang.ac.kr	
	황선명	대전대학교	sunhwang@dju.kr	
	전진옥	비트컴퓨터	jojeon@bit.co.kr	
	김수동	송실대학교	sdkim777@gmail.com	
	이궁해	항공대학교	khlee@kau.ac.kr	
최병주	이화여자대학교	bjchoi@ewha.ac.kr		

구분	성명	소속기관명	E-Mail
이사	김정아	관동대학교	clara@kwandong.ac.kr
	남영광	연세대학교	yknam@yonsei.ac.kr
	박수진	서강대학교	psjdream@sogang.ac.kr
	염근혁	부산대학교	yeom@pusan.ac.kr
	오재원	카톨릭대학교	jwoh@catholic.ac.kr
	윤희병	국방대학원	hbyoon37@hanmail.net
	이우진	경북대학교	woojin@knu.ac.kr
	이은석	성균관대학교	leees@skku.edu
	이석원	아주대학교	leesw@ajou.ac.kr
	이은주	경북대학교	ejlee@knu.ac.kr
	전태웅	고려대학교	jeon@korea.ac.kr
	정인상	한성대학교	insang@hansung.ac.kr
	최승훈	덕성여자대학교	csh@duksung.ac.kr
	최종무	단국대학교	choijm@dankook.ac.kr
	최호진	KAIST	hojinc@kaist.ac.kr
	현창문	탐라대학교	cmhyun@tnu.ac.kr
	계승교	삼성SDS	seankae@samsung.com
	권경룡	국방기술품질원	ka-ja17@hanmail.net
	권원일	STA컨설팅	wonil@softwaretesting.co.kr
	김상기	현대자동차	sangkikim@hyundai-motor.com
	김진태	SEEG	jtkim@swexpertgroup.com
	민경오	LG전자	davidmin@lge.com
	민상윤	솔루션링크	sang@sol-link.com
	박복남	핸디피엠지	pbnknb@hitel.net
	박찬규	국방SW산학연합회	milspark@hotmail.com
	배현섭	슈어소프트테크	hsbae@suresofttech.com
	손세창	인천공항공사	scsohn@airport.kr
	손진규	삼성탈레스	Jinkyu.son@samsung.com
	신석규	SW시험인증센터	skshin@tta.or.kr
	양상욱	KAI	sangyang@koreaaero.com
	유영수	현대엠앤소프트	ysyoo@hyundai-mnsoft.com
	윤태권	한국SW기술진흥협회	tkyune@empal.com
	윤형진	케피코	Hyoungjin.yoon@kefico.co.kr
	이근	삼성전자	gskeun.lee@samsung.com
	이성남	방위사업청	dapalee@korea.kr
	이우복	삼성전자	woobok.yi@samsung.com
	이장수	한국원자력연구원	jslee@kaeri.re.kr
	장주수	모아소프트	jsjang@moasoftware.co.kr
	정연대	N3SOFT	ydchung@n3soft.co.kr
	조병인	국방과학연구소	chobyun@dreamwiz.com
	조상윤	다한테크	sycho@dahan.co.kr



## 2011-2012 소프트웨어공학소사이어티 논문지 편집위원회

편집위원장 강성원 교수(KAIST)

편집위원 김문주 교수(KAIST)

김정아 교수(관동대학교)

김현수 교수(충남대학교)

박수진 교수(서강대학교)

윤회진 교수(협성대학교)

이관우 교수(한성대학교)

이우진 교수(경북대학교)

이지현 교수(대전대학교)

채흥석 교수(부산대학교)

최종무 교수(단국대학교)

김태호 박사(ETRI)



## 소프트웨어공학소사이어티 논문지 제25권 제2호 (통권 94호)

발행일 || 2012년 6월 30일

발행인 || 한혁수

편집인 || 강성원

발행처 || 사단법인 한국정보과학회 소프트웨어공학소사이어티

연락처 || 서울특별시 종로구 홍지문 2길 20, 상명대학교 소프트웨어 대학관 G511

전화 : 02-2287-5033, 팩스 : 02-2287-0049

홈페이지 : <http://www.sigse-kiss.or.kr/>

인쇄처 || (주)참기획 (전화 : 042-861-6380, 팩스 : 042-861-6381)

Copyright© 2012 한국정보과학회 소프트웨어공학소사이어티(비매품)