

국방 CBD 방법론의 상호운용성 향상을 위한 체크리스트 연구

류 동 국[†] · 김 영 철^{**}

요 약

국방 정보 시스템은 임무 중심적인 기능을 수행하므로 시스템간의 상호운용성의 확보가 절실히 요구된다. 시스템간의 상호운용성을 증진하고 CBD 방법론의 장점을 활용하기 위하여 군에서는 일종의 컴포넌트 뱅크 개념인 공통운용환경(COE)의 개발을 계획하고 있다. 공통운용환경의 개발을 위해 설계된 ADDMe는 기존의 CBD 방법론들을 비교, 분석, 종합하여 국방 환경에 적합하게 고안된 새로운 국방 CBD 방법론이다. 본 논문은 국방 CBD 방법론인 ADDMe를 소개하고 ADDMe를 적용하여 개발된 컴포넌트 기반 시스템의 상호운용성을 향상하기 위한 체크리스트를 제시한다. 상호운용 체크리스트는 ADDMe의 세부 개발 단계별로 적용되어 국방 컴포넌트 기반 시스템의 상호운용성을 향상시킨다.

키워드 : 컴포넌트 기반 개발 방법론, 상호운용성, 체크리스트

A Study on Checklist for Improving Interoperability of Advanced Defense Component Development Methodology(ADDMe)

Dong-Kuk Ryu[†] · R. Young-Chul Kim^{**}

ABSTRACT

Defence Information System has surely required the guarantee of interoperability between systems as performing task oriented functions. To improve interoperability between them and use the benefits of component based development, defense officers are planning to develop COE like a kind of component bank concept. Defense component development methodology(ADDMe) within COE is a new approach of component based methodology for Defense environment after comparing, analyzing, and synthesizing with traditional CBDs. In this paper we introduce advanced defense component development methodology(ADDMe) and present checklist for improving interoperability between Defence component based systems which are developed with ADDMe. With our proposed interoperability's checklist, we can improve interoperability of defense component based systems through applying per each phase of ADDMe.

Key Words : CBD(Component Based Development), Interoperability, Checklist

1. 서 론

정보통신 기술의 발달로 많은 정보 시스템이 개발되고 있다. 정보 시스템간의 유기적인 상호 연동이 요구됨에 따라 시스템간의 상호운용성의 확보는 시스템 개발에 있어서 매우 중요한 부분이 되고 있다[1, 2]. 특히 국방, 금융, 행정 등이 절적이고 임무중심적인 시스템간의 상호 연동이 요구되는 분야에서는 시스템의 개발에 있어서 상호운용성은 매우 중요한 고려 요소이다[3].

상호운용성을 향상하기 위한 한 방법으로 근래에 들어 기 개발된 컴포넌트를 재사용하여 시스템을 개발하는 CBD(Component Based Development) 방법으로 많은 시스템이

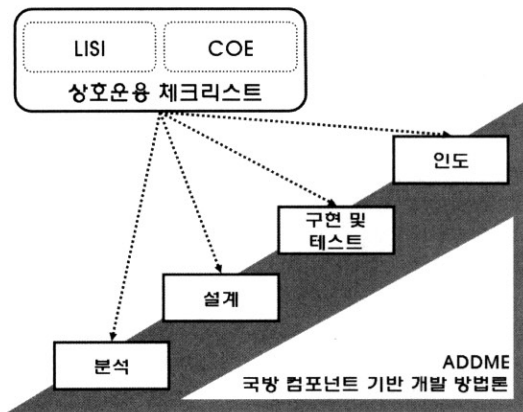
개발되고 있다. CBD 방법은 이미 개발된 컴포넌트를 재사용하여 시스템을 개발하므로 상호운용성을 향상하고, 개발 비용을 절감하고, 개발기간을 단축하고, 부품화된 컴포넌트를 쉽게 교체하여 성능향상을 하는 등의 많은 장점을 가지고 있다. 이러한 CBD 방법의 장점을 활용하기 위하여 군에서는 공통운용환경(COE : Common Operating Environment)의 개발을 계획하고 있다[4]. 그러나 시스템의 상호운용성 검증은 개발이 완료된 후인 시험단계(통합시험)에서 실시하므로 개발과정에서 시스템의 상호운용성이 확보되지 못한 경우, 다시 분석, 설계, 개발 단계를 반복하여 수행하는 경우가 자주 발생하고 있다[3]. 따라서 개발과정에서 각 단계별로 필요한 상호운용 고려 사항을 도출하여 개발 단계별로 점검하여야만 쉽게 시스템의 상호운용성을 확보할 수 있다.

CBD 방법으로 개발되는 시스템간의 상호운용성을 확보하기 위해서는 CBD 방법론의 각 개발 단계와 활동에서 상

[†] 정 회 원 : 국방과학연구소 선임연구원

^{**} 정 회 원 : 홍익대학교 컴퓨터정보통신공학과 교수

논문접수 : 2005년 4월 27일, 심사완료 : 2005년 7월 18일



(그림 1) 상호운용 체크리스트

호운용성 확보에 필요한 관련된 사항을 확인하는 구체적인 체크리스트가 필요하다. 그러나 카탈리시스, RUP(Rational Unified Process), Select Perspective 그리고 마르미-III와 같은 국내외의 최신 CBD 방법론에서도 외부 시스템과의 상호운용성 확보를 위한 구체적이고 체계적인 고려가 부족하다[6-10].

본 연구에서는 컴포넌트 기반 시스템의 개발 과정에서 발생하는 시스템간 상호운용성 확보의 문제점을 해결하기 위하여 CBD 방법론의 각 개발 단계별로 필요한 상호운용 고려요소를 체크리스트로 개발하였다[11, 15]. 상호운용 체크리스트는 CBD 방법론에서 정의하는 각 단계별로 상호운용성 확보를 위해 점검해야할 요소들을 포함하고 있다. 본 체크리스트는 CBD 방법론의 개발 과정에 적용하여 컴포넌트 기반 시스템간의 상호운용성을 향상하는 것을 목적으로 하고 있다.

상호운용 체크리스트의 점검항목을 도출하기 위하여 본 연구에서는 (그림 1)과 같이 상호운용 평가 모델인 LISI (Levels of Information Systems Interoperability)의 능력모델, 구현 옵션 그리고 공통운용환경의 순응성 체크리스트를 활용하였다. CBD 방법론은 국방 CBD 방법론인 ADDMe (Advanced Defense Component Development Methodology)를 사용하였다[12]. ADDMe 방법론은 기존 CBD 방법론을 종합 발전시킨 최신의 CBD 방법론으로 국방 컴포넌트 개발을 위하여 새롭게 개발되어 표준 방법론으로 사용되고 있다.

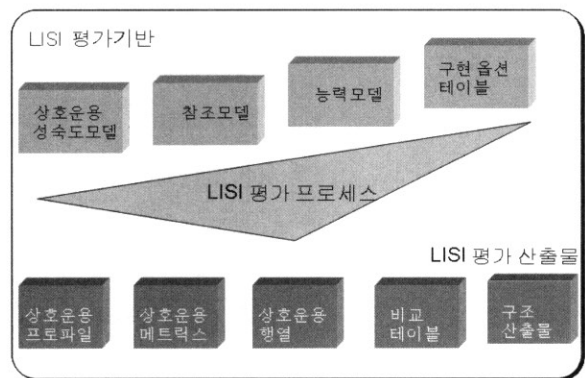
본 논문의 구성은 다음과 같다. 제 2장 관련연구에서는 상호운용 체크리스트 작성에 참조한 LISI와 공통운용환경을 설명한다. 제 3장에서는 ADDMe 방법론을 소개하고 상호운용성 관점에서 분석하였다. 제 4장에서는 ADDMe의 개발 단계별로 적용가능한 상호운용 체크리스트를 설명한다. 제 5장에서는 상호운용성 측정 사례를 살펴보고, 마지막으로 제 6장에서는 본 연구의 결론 및 향후과제를 논하였다.

2. 관련 연구

본 장에서는 상호운용 체크리스트 도출에 관련된 LISI와 공통운용환경을 설명한다.

2.1 LISI

이질적인 시스템간의 상호운용성 향상에 관심이 많은 조직은 미 국방부이다. 미군은 상호운용성을 “체계, 부대, 군이 공통의 임무를 수행하기 위하여 다른 체계, 부대 혹은 군과 정보 혹은 서비스를 교환 이용하는 능력”으로 정의하고 시스템간의 상호운용성 향상에 많은 예산을 투자하고 있다[18]. LISI는 이러한 노력의 일환으로 연구된 상호운용성 평가 모델이다. 1998년에 CMM(Capability Maturity Model)을 개발한 카네기 멜론대학의 SEI 연구소에서 정보 시스템의 상호운용성을 평가하기 위한 모델로 개발되었다[13, 14]. LISI는 성숙도 모델에서의 수준 개념을 이용하여 정보시스템의 상호운용 능력을 평가한다.



(그림 2) LISI 구성요소

2.1.1 LISI 구성요소

(그림 2)는 LISI의 구성요소를 나타낸다. LISI는 상호운용성을 평가하는 기반환경이라 할 수 있는 LISI 평가 기반과 이를 활용하여 평가하는 평가 프로세스 그리고 평가 결과에 해당하는 LISI 평가 산출물로 구성된다. LISI에 의한 상호운용 평가는 먼저 구조화된 상호운용 질의서를 이용하여 평가 대상 시스템의 정보를 수집한다. 질의서를 통하여 수집된 정보는 LISI 평가 기반에서 정의된 상호운용성숙도 모델, 참조 모델, 능력 모델, 구현 옵션 테이블을 이용하여 LISI 평가 프로세스를 통하여 상호운용 프로파일, 상호운용 매트릭스, 상호운용 행렬, 비교 테이블, 구조 산출물 등의 LISI 평가 산출물을 생성하게 된다. 구성요소에 대한 자세한 설명은 다음과 같다.

- 상호운용 성숙도 모델 : 시스템의 상호운용 요구 능력을 (그림 3)과 같이 6개의 수준으로 표시한 성숙도 모델이다. 요구되는 상호운용 능력을 정보 교환, 상호 협력, 자료와 응용 관계, 컴퓨팅 환경에 따라 격리, 불완전, 연결, 기능적, 도메인, 전군적으로 상호운용 수준을 구분한다. 각 수준의 세부적인 요구 능력은 다음과 같다.
- 수준 0(격리) : 타 시스템과의 상호운용이 이루어지지 않고 독립적으로 운용되는 수준
- 수준 1(불완전) : 상호운용이 이루어지고 있으나 사람이 디스켓 등을 이용하여 수동적으로 이루어지는 수준

특징 수준	정보교환	상호협력	자료 & 응용관계	컴퓨팅 환경
5 전군적	전군적 공유자료	협력가능 공간	공유된 자료 공유된 응용	
4 도메인	공유된 데이터베이스	진보된	공유된 자료 개별적 응용	
3 기능적	이종의 자료	복잡한	개별적 자료 개별적 응용	
2 연결	동종의 자료	기본	개별적 자료 개별적 응용	
1 불완전	수동적 교환	인간	개별적 자료 개별적 응용	
0 격리	없음	없음	개별적 자료 개별적 응용	

(그림 3) 상호운용 성숙도 모델

- 수준 2(연결) : 동종의 자료를 이용하여 컴퓨터간에 1:1 방식으로 이루어지는 상호운용 수준
- 수준 3(기능적) : 개별적인 응용 프로그램으로 이종의 자료에 대하여 상호운용이 가능한 수준
- 수준 4(도메인) : 도메인 내부에서 개별적인 응용 프로그램으로 공유된 데이터베이스를 이용하여 상호운용이 가능한 수준
- 수준 5(전군적) : 전군의 도메인간의 자료공유와 협력의 가상공간을 제공하는 상호운용 수준
- 참조 모델 : LISI 참조 모델은 (그림 4)와 같다. 상호운용 성숙도 모델을 절차(Procedure), 응용(Application), 기반 구조(Infrastructure), 데이터(Data)로 구분하여 상호운용 수준을 정의한다. 참조 모델을 이용하여 분야별로 시스템의 상호운용성을 평가할 수 있다.
- 능력 모델 : LISI 참조 모델은 일반적이고 단순화된 비교로 상호운용성 수준 및 특성에 관련된 세부사항은 제공하지 않는다. 따라서 각 수준별 세부적인 특성을 PAID (Procedure, Application, Infrastructure, Data) 항목으로

특징 수준	절차	응용체계	기반구조	데이터
5-전군적	전사적 수준	가상협력 지원	다차원 광역 망	전사적 모델
4-도메인	도메인 수준	그룹협력 지원	광역 망	도메인 모델
3-기능적	프로그램 수준	기본업무 지원	근거리 망	프로그램 모델
2-연결	지역적 수준	단순 상호작용 지원	물리적 연결(P2P)	지역적 모델
1-불완전	수동적 수준	없음	이동 가능 매체	개별자료
0-격리	상호운용성 없음			

(그림 4) LISI 참조 모델

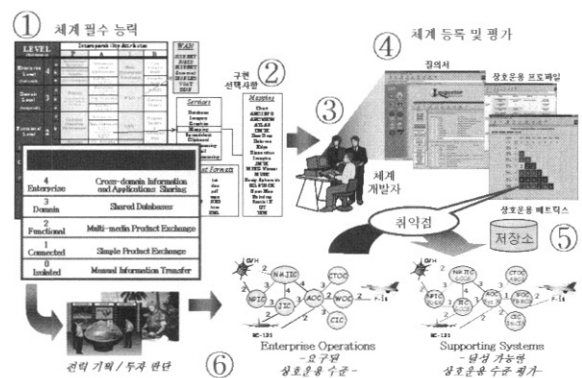
표현하기 위한 세부적인 모델이 필요한데 이러한 확장을 LISI 능력 모델이라 한다.

- 구현 옵션 테이블 : LISI 능력 모델의 세부적인 기능을 구현하기 위하여 가용한 구현 선택사항(표준, 제품)을 포함한다.

2.1.2 LISI 평가 프로세스

LISI의 상호운용 평가 프로세스는 (그림 5)와 같다. 그림에서 1단계와 2단계는 상호운용성 평가를 위한 지침을 제정하여 표준 및 평가 모델을 만드는 단계이고, 일반적인 상호운용성 평가는 3단계에서 6단계까지를 포함한다. 평가 단계별 내용은 다음과 같다.

- ① 상호운용에 대한 정의, 표준, 지침을 제정하여 상호운용 평가 환경을 정립한다.
- ② LISI 성숙도 모델, 참조 모델, 구현 옵션을 개발하여 상호운용성 평가를 가능하게 한다.
- ③ 상호운용 질의서에 개발 시스템에 대한 구현 정보(구현 옵션)를 입력한다.
- ④ 질의서에 입력된 정보를 바탕으로 평가 대상 시스템의 상호운용 능력에 관련된 정보를 종합한 상호운용 프로파일을 작성한다.
- ⑤ 상호운용 프로파일을 바탕으로 상호운용성을 평가하고 평가 결과를 저장한다.
- ⑥ 더욱 향상된 상호운용 수준을 확보하기 위하여 시스템의 구현 옵션을 조정하고 시스템 개발에 반영한다.



(그림 5) 상호운용성 평가 프로세스

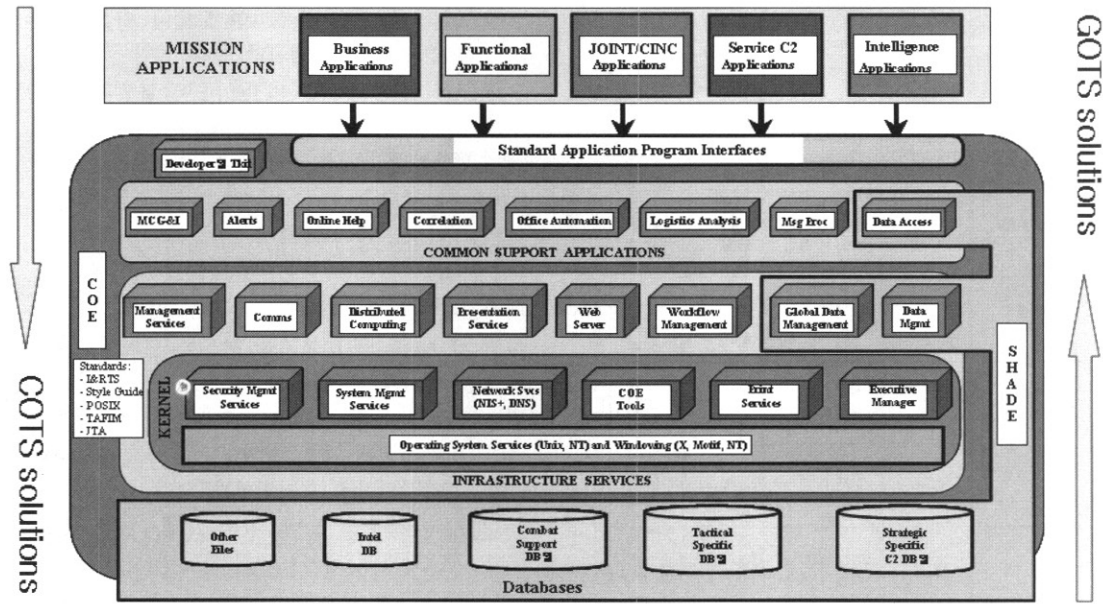
2.1.3 상호운용 체크리스트 도출

본 연구에서는 상호운용 체크리스트의 세부적인 내용을 LISI의 체크리스트를 분석하여 작성하였다. LISI 체크리스트는 상호운용에 관련된 많은 사항들을 PAID로 분류하여 정의한다. 본 연구에서는 LISI 질의서를 ADDMe의 각 단계에 맞게 다시 분류하고 활용하여 체크리스트로 작성하였다.

2.2 공통운용환경 (COE : Common Operating Environment)

2.2.1 공통운용환경 개요

공통운용환경은 미 국방성에서 현재 운용중인 일종의 컴



(그림 6) 공통운영환경 세부 구성 요소

포넌트 백이다. 미군은 여러 정보 시스템의 개발과정에서 동일한 기능이 반복적으로 개발되는 현상을 발견하고 개발 비용 절감, 개발의 적시성 그리고 상호운용성의 증진을 위하여 공통운영환경을 개발하여 운영하고 있다. 공통운영환경의 구성은 (그림 6)과 같다.

그림의 하단 부분에 있는 기반 구조 서비스(Infrastructure Services)는 응용 시스템의 운영환경에 해당하며 운영체제, 데이터베이스 등 상용제품을 선택하는 방식으로 표준화하여 이종 상용제품간의 상호운용성 문제를 해결하고 있다. 그리고 그림의 상단에 있는 공통 지원 응용(Common Support Applications)는 공통적으로 사용되는 소프트웨어 모듈들이 존재한다. 이 모듈들은 표준화된 공통기능의 컴포넌트로 한번 개발되면 후속 체계 개발에 재사용 된다. 그리고 시스템의 개발과정에서 공통 모듈이 도출될 경우에는 공통운영환경에 등록하여 관리된다. 우리군도 현재 공통운영환경의 개발을 고려하고 있고 시스템 개발에 있어서 CBD 방법론을 이용한 개발을 장려하고 있으며 공통운영환경 컴포넌트 개발의 방법론은 ADDMe를 사용한다[5].

2.2.2 공통운영환경 순응성 체크리스트

상호운용 체크리스트를 작성하는데 있어서 공통운영환경의 순응성 체크리스트(COE Compliance Checklist)를 활용하였다. 공통운영환경 순응성 체크리스트는 <표 1>과 같이 공통운영환경 컴포넌트를 체크리스트로 평가하여 공통운영환경의 규격에 부합하는 정도를 수준으로 평가한다.

공통운영환경은 표준화된 개발 및 운영환경을 제공하므로 체크리스트의 많은 사항들이 상호운용에 관련된다. 본 연구에서는 공통운영환경 순응성 체크리스트를 분석하여 ADDMe에 적용 가능한 부분을 도출하고 ADDMe의 각 단계별로 분류하여 상호운용 체크리스트 개발에 활용하였다.

<표 1> 공통운영환경 순응성 수준

수준	수준별 요구사항
1	Standard OS, 네트워크, 데이터베이스 표준에 관한 기초적인 사항 요구
2	Network 네트워크 환경에서 보안사항 요구
3	Platform 플랫폼 측면에서 상호운용성 요구
4	Bootstrap 자동화된 관리 도구를 통한 패키지 설치 및 제거
5	Minimal 기초적인 DII 적합성 요구
6	Intermediate 중간 수준의 DII 적합성 요구
7	Interoperable DII 적합성을 만족하면서 상호운용 가능
8	Full DII 적합성 모두 만족 요구

*DII : Defense Information Infrastructure

3. 국방 컴포넌트 개발 방법론 (ADDMe)

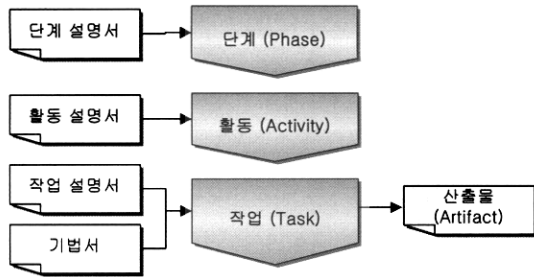
본 장에서는 국방 컴포넌트 개발 방법론인 ADDMe를 소개하고 ADDMe의 구성 요소 및 개발 절차를 설명한다.

3.1 개요

ADDMe는 국방 CBD 방법론으로 국방 분야에서 사용되는 컴포넌트를 획득하기 위한 표준 방법론으로 2004년에 제정되었다[12]. 산업 표준인 UML(Unified Modeling Language) 기반의 CBD 방법론에 기초하여, 분석, 설계, 구현 과정의 산출물을 표준화하였다. ADDMe는 카탈리시스, RUP, Select Perspective 그리고 마르미-III 등 기존의 CBD 방법론들의 장점을 종합하여 국방 환경에 적합한 방법론으로 설계되었다.

3.2 구성요소 및 관련 문서 구조

ADDMe는 (그림 7)과 같이 단계(Phase), 활동(Activity), 작업(Task)으로 구성된다. 단계, 활동, 작업의 세부사항은



(그림 7) 관련 문서 구조

단계 설명서, 활동 설명서, 작업설명서 및 기법서에서 구체적으로 정의된다. 또한 각 작업의 결과는 산출물로 생성된다. 각 문서에 대한 설명은 다음과 같다.

- 단계 설명서 : 해당 단계에 대한 개요 설명과 활동 흐름도를 나타낸다. 단계에 속한 활동에 대한 설명과 활동 간의 관계 정보가 포함된다.
- 활동 설명서 : 해당 활동에 대한 개요 설명과 작업 흐름도를 나타낸다. 활동에 속한 작업에 대한 설명과 활동을 수행하기 위한 입력력 목록을 보여준다.
- 작업 설명서 : 해당 작업에 대한 설명과 작업을 수행하기 위한 세부 절차의 흐름도 및 설명을 나타낸다.
- 산출물 : 작업의 결과로 생성된다. 작성자 및 작성 이력, 산출물을 작성하는데 필요한 입력물 관련 기법이 표기된다. 산출물별로 정해진 양식에 따라 작성한다.
- 기법서 : 산출물 작성에 참고하는 문서로 특별한 기법에 대한 설명이다. 예를들어 UML 다이어그램을 작성하는 방법을 설명한 UML 모델링 기법이나 데이터베이스 설계 시 컴포넌트와의 연동을 위한 기법 등을 설명한다.

ADDMe는 총 4단계, 12활동, 37작업 그리고 41종의 산출물로 구성된다[12].

3.3 개발절차

ADDMe의 개발 절차는 분석, 설계, 구현 및 테스트, 인도

총 4 단계로 이루어져 있다. 각 단계는 세부적인 활동, 작업, 산출물로 구성된다. <표 2>는 ADDMe 분석 단계의 세부적인 구성을 나타낸 것이다. 각 단계는 세부적인 활동으로 구성되며 활동별로 식별자 번호를 부여하였다. 분석단계의 식별자 번호는 1R이며 활동, 작업별로 세부적인 번호가 부여되어 식별된다. <표 2>와 같이 단계, 활동, 작업 그리고 산출물은 구조적 포함관계를 형성하고 개발의 단계별로 산출물을 작성한다. 각 산출물은 정형화된 일정한 양식이 정의되어 있어 개발자나 관리자는 양식에 따라 문서를 작성한다.

3.4 상호운용 체크리스트 도출

ADDMe의 산출물 문서는 일정한 양식이 정해져 있다. 따라서 문서 내부의 각 항목들은 문서의 목적에 맞는 의미를 포함한다. 본 연구에서는 총 41종의 ADDMe 산출물들을 상호운용 중요도에 따라 분석하고 산출물 문서 내부에 존재하는 상호운용성 향상에 필요한 고려 요소를 분석하여 체크리스트로 작성하였다.

4. 상호운용 체크리스트

4.1 ADDMe 산출물 문서 분석

상호운용성 체크리스트 식별에 앞서 ADDMe에서 정의한 총 41종의 산출물을 상호운용 관점에서 분석하였다. <표 3>에서와 같이 각 문서에 대하여 상호운용 중요도를 상, 중, 하로 구분하였다. 상호운용 중요도가 하인 문서는 문서가 시스템의 상호운용성과 특별한 관련성을 포함하고 있지 않은 문서이다. 그리고 산출물 문서 내부에 존재하는 상호운용 관련 항목 갯수의 평균인 5개를 기준으로 5개 이상의 상호운용 관련 항목을 포함하는 문서는 중요도를 상으로 평가하였고 5개 미만인 문서는 중으로 평가하였다.

4.2 ADDMe 개발 단계 활동별 상호운용 요소 식별

본 절에서는 ADDMe의 각 개발 단계별로 도출한 상호운

<표 2> ADDMe 개발절차 (분석단계)

단계	활동	작업	산출물	
분석 1R	요구사항 정의 1R1	상위요구사항 정의	1R11	상위요구사항 정의서
		도메인 모델링	1R12	도메인 명세서 용어집
		현행 시스템 분석	1R13	현행시스템 분석서
		비즈니스 모델링	1R14	비즈니스 정의서
		요구사항 명세	1R15	요구사항 명세서
	아키텍처 정의 1R2	소프트웨어 아키텍처 정의	1R21	소프트웨어 아키텍처 정의서
		시스템 아키텍처 정의	1R22	시스템 아키텍처 정의서
		표준 지침 수립	1R23	표준 지침서
	요구사항 분석 1R3	유스케이스 모델링	1R31	유스케이스 명세서
		사용자 인터페이스 프로토타이핑	1R32	사용자 인터페이스 정의서
		클래스 모델링	1R33	클래스 명세서
		테스트케이스 정의	1R34	테스트케이스 정의서

〈표 3〉 상호운용 관련 산출물

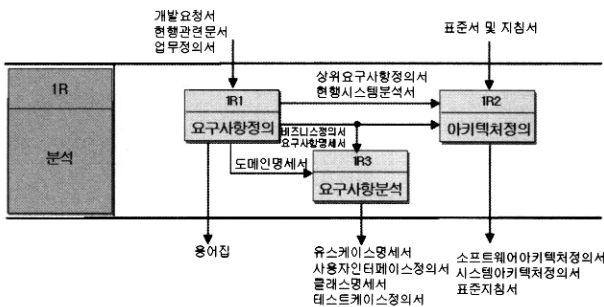
단계	산출물	상호운용 관련요소	중요도
분석 1R	상위 요구사항 정의서/요구사항 명세서	-외부 체계와의 상호운용 요구 사항 정의	상
	비즈니스 정의서	-외부 조직과의 상호운용 비즈니스 모델 기술	상
	소프트웨어 아키텍처 정의서	-외부 체계와의 상호운용 아키텍처 구조 기술	중
	표준 지침서	-상호운용을 위한 표준 정의	중
	유스 케이스 명세서	-외부 체계와의 상호운용에 대한 유스 케이스 명세	중
	클래스 명세서	-상호운용 클래스 식별	중
설계 2D	컴포넌트 획득 방법 식별서	-획득 방식에 따른 상호운용 절차 기술	중
	인터페이스 상호작용 명세서	-상호운용 인터페이스 식별	상
	컴포넌트 명세서	-상호운용 컴포넌트 식별	중
	컴포넌트 설계서	-상호운용 컴포넌트 설계	중
	컴포넌트 구현 설계서	-상호운용 컴포넌트 구현 설계	중
구현 및 테스트 3T	테스트 계획서	-상호운용 요구사항 테스트 계획	중
	통합 테스트 설계서	-상호운용을 위한 통합 테스트	중

용 체크리스트를 설명한다. 각 단계별 체크리스트의 일부를 <부록 1>에 기술하였다.

4.2.1 분석 단계 (1R)

ADDMe의 분석 단계에서는 (그림 8)과 같이 요구사항 정의, 아키텍처 정의 및 요구사항 분석 활동을 수행한다. 요구사항 정의를 통해 프로젝트의 목적 달성에 필요한 도메인의 범위를 정하고 비즈니스 업무에 대한 이해와 개발 시스템의 기능적 요소를 식별하여 요구사항을 정한다. 아키텍처 정의를 통해 개발 시스템을 구조화하는 아키텍처를 정의하며, 요구사항 분석 활동을 통하여 정의된 요구사항을 시스템 개발 관점에서 구체화하고, 상세화하여 시스템에 대한 보다 정확한 이해를 도모한다.

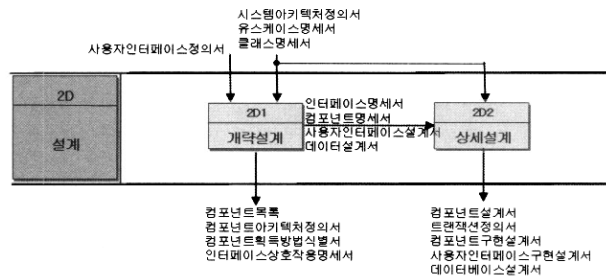
이 단계의 상호운용 체크리스트에서는 요구 사항 분석 활동에서 상호운용 유스케이스, 타 시스템과의 상호운용 요구, 클래스간의 상호운용 등을 확인한다. 그리고 아키텍처 정의 활동에서 상호운용을 위한 타 시스템과의 상호운용 구조를 점검한다.



(그림 8) 분석 단계 활동 흐름

4.2.2 설계 단계 (2D)

설계 단계에서는 (그림 9)와 같이 개략 설계와 상세 설계 활동을 수행한다. 개략설계 활동에서는 컴포넌트를 식별하



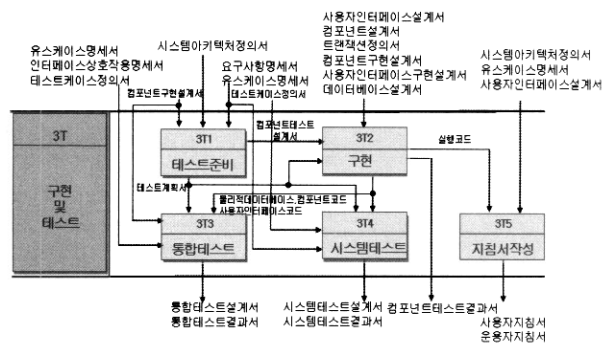
(그림 9) 설계 단계 활동 흐름

여 컴포넌트 명세를 작성하고, 사용자 인터페이스 설계 및 데이터 모델을 작성한다. 상세 설계 활동을 통하여 컴포넌트의 내부를 설계하고 구현 플랫폼과 연동되도록 상세하게 설계한다.

이 단계의 상호운용 체크리스트에서는 상호운용 요구 사항을 구현하기 위한 설계를 점검한다. 외부 컴포넌트와의 구조, 인터페이스, 데이터베이스 등을 점검한다.

4.2.3 구현 및 테스트 단계 (3T)

구현 및 테스트 단계는 (그림 10)과 같이 테스트 준비, 구현, 통합 테스트, 시험 테스트 및 지침서 작성 활동을 수행



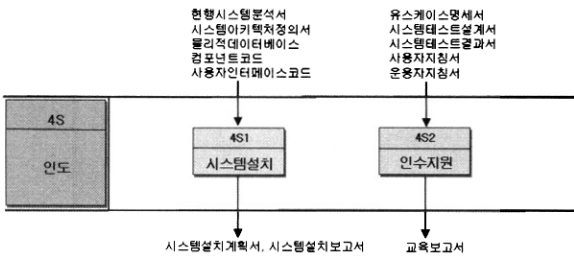
(그림 10) 구현 및 테스트 단계 활동 흐름

한다. 테스트 준비 활동을 통해 테스트 계획을 작성하고 컴포넌트에 대한 설계를 한다. 구현 활동을 통해 컴포넌트와 시스템을 구현하고, 사용자 인터페이스를 구현하여 서버 컴포넌트와 연동한다. 구현이 완료되면 통합 테스트와 시스템 테스트를 수행하고 지침서를 작성한다.

이 단계의 상호운용 체크리스트에서는 설계 문서를 바탕으로 상호운용 요구 사항의 구현 여부를 점검한다. 또한 테스트와 구현의 순환 과정을 통하여 상호운용이 확보되었는지 점검한다.

4.2.4 인도 단계 (4S)

인도 단계는 (그림 11)과 같이 개발된 결과물을 실제 시스템이 운영될 사용자 환경에 설치하고 인수 테스트를 지원한다. 시스템 사용자에게 시스템의 운영 방법과 사용 방법에 대한 교육을 실시하여 사용자가 새로운 시스템에 쉽게 적응할 수 있다. 이 단계의 상호운용 체크리스트에서는 시스템간에 원활하게 상호운용 되는지를 점검한다.



(그림 11) 인도 단계 활동 흐름

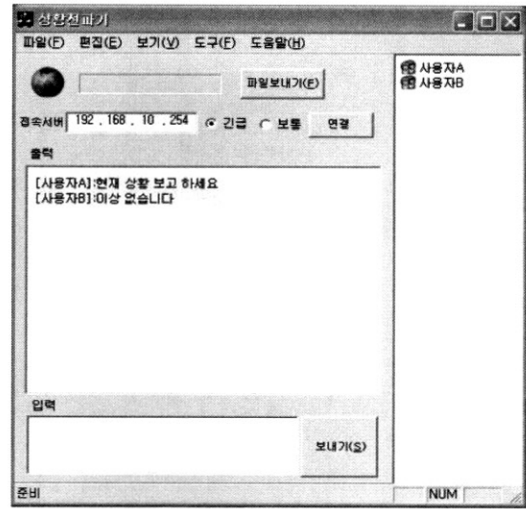
5. 적용사례 및 상호운용 수준 측정 결과

5.1 적용사례

본 연구에서 개발한 상호운용 체크리스트는 ADDMe 방법론으로 컴포넌트 기반 시스템을 개발하는 과정에서 시스템의 상호운용성 향상을 위하여 점검하여야 할 항목들을 제시하는 기능을 한다. 따라서 개발자가 ADDMe 방법론으로 개발하는 과정에 상호운용 체크리스트를 활용할 경우 최종적으로 개발되는 시스템의 상호운용 수준은 사용자의 요구 수준을 만족하여야 한다. 본 장에서는 이러한 상호운용 체크리스트의 기능을 실험하였다. 상호운용 수준 평가는 국방과학연구소에서 개발한 국방 정보체계 상호운용 시험 및 평가 시스템(SITES : Systems Interoperability Test and Evaluation System)을 이용하였다[17].

5.1.1 상황 전파 메신저 컴포넌트

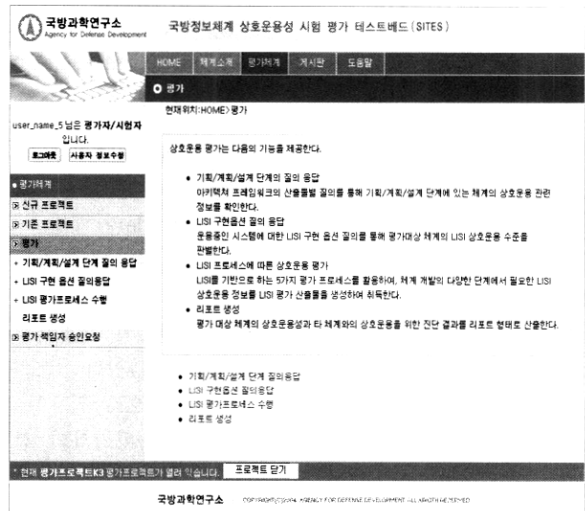
상황 전파 메신저 컴포넌트는 상호운용 체크리스트를 검증하기위하여 ADDMe 방법론을 적용하여 실험적으로 개발한 가상의 컴포넌트이다. 상황 전파 메신저 컴포넌트는 (그림 12)와 같이 같은 임무를 수행하는 사람들간의 정보를 교환하는 기능을 수행한다.



(그림 12) 상황 전파 메신저 컴포넌트

5.1.2 SITES

SITES는 (그림 13)과 같이 국방과학연구소에서 국방 정보 체계의 상호운용성 평가 및 시험을 위하여 개발한 시스템이다[17]. SITES는 개발문서들을 LISI 모델에 적용하여 상호운용 수준을 평가하는 기능과 실질적인 상호운용성을 검증을 위한 코드기반 시험 기능으로 구성되어 있다. 본 논문에서는 SITES의 상호운용 수준 평가 기능을 이용하여 상황 도시 컴포넌트의 상호운용 수준을 평가하였다.



(그림 13) SITES 초기화면

5.2 상호운용 수준 평가 결과

SITES를 이용한 상황 전파 메신저 컴포넌트의 상호운용 수준 평가결과는 (그림 14)와 같다. 상황 전파 메신저 컴포넌트에 대한 LISI 상호운용 수준 평가 결과 요구 수준인 수준 4를 만족하였다. 평가 결과를 PAID 세부 항목으로 살펴보면 절차 4a, 응용 4b, 기반구조 4b, 데이터 4a로 평가되었다. LISI 수준 평가에서는 가장 낮은 평가 항목이 전체 상호

수준(환경)		상호운용 애크리뷰트				
		P	A	I	D	
전군(enterprise) 대화식 조작, 공유 데이터 및 응용	5	b				
		a				
도메인(domain) 공유 데이터, 분리된 응용	4	b	각군차원 상호운용 절차	자료공유를 지원하는 응용 매체포맷	TCP/IP	
		a			개별 DB	
기능(function) 최소한의 공통 기능, 분리된 데이터 및 응용	3	c	국방데이터공유환경 업무편람	단순 데이터베이스를 사용한 데이터 관리	TCP IP	
		b			HTML	
		a	Win32	MIME		
연결 (connected) 전자적 연결, 분리된 데이터 및 응용	2	c	DITA 적용	SMTP		
		b		TELNET	자동 만족	
		a	Security	단순메시지		GIF
고립(isolated) 비 연결	1	b	Kerberos		ISO 9660	
		a	군사보안업무 시행 규칙	N/A	자동 만족	FAT
상호운용성 없음	0	a	상호운용성 없음			

(그림 14) 평가 결과

운용 수준을 대표하므로 상황 전파 메신저 컴포넌트의 잠재적 상호운용 수준이 4a로 측정된다[13].

6. 결론

CBD 방법론으로 개발된 컴포넌트 기반 시스템의 상호운용성을 확보하기 위해서 본 논문에서는 컴포넌트 기반 시스템의 개발 단계별로 상호운용성을 증진하는 체크리스트를 작성하였다. 컴포넌트 기반 시스템간의 상호운용성을 확보하기 위해서는 각 단계별로 상호운용 고려요소를 식별하고 체크리스트화하여 점검하여야만 최종적으로 개발되는 컴포넌트 기반 시스템의 상호운용성을 확보할 수 있다. 본 논문에서는 국방 컴포넌트 개발 방법론인 ADDMe의 개발 단계별로 점검할 상호운용 체크리스트를 LISI 질의서와 공통운용환경 순응성 체크리스트 등을 활용하여 작성하였다. 작성된 체크리스트는 상황 전파 메신저 컴포넌트의 개발과정에 적용하여 상호운용성이 증진됨을 확인할 수 있었다.

향후연구로는 모든 CBD 방법론에 일반적으로 적용할 수 있도록 체크리스트의 보완이 필요하며 LISI 평가와의 접목을 통한 상호운용성 수준 향상 방안의 연구 또한 필요하다.

참고 문헌

[1] Wegner, P., "Interoperability", ACM Computing Surveys, Vol.28, No.1, March, 1996.
 [2] Garlen, D., Rober, A. and John, O., "Architectural Mismatch or Why it's hard to build systems out of existing parts", Proceedings of the 17th International Conference on Software Engineering IEEE, 1995.
 [3] "DoD Architecture Framework version 2.1", DoD

Architecture Framework Working Group, 2000.
 [4] "Defense Information Infrastructure Common Operating Environment(DII COE) : Integration and Runtime Specification 4.0 (I&RTS)", US DoD, 1999.
 [5] "공통운용환경(COE) 업무편람", 국방부, 2003.
 [6] George, T. H. and William, T. C., "Component based Software Engineering", Addison-Wesley, 2001.
 [7] D. D'sousz, and A. Wills, "Objects, Components, and Frameworks with UML: The Catalysis Approach", Addison-Wesley, 1998.
 [8] I. Jacobson, G. Booch, and J. Rumbaugh, "The Unified Software Development Process", Addison-Wesley, 1999.
 [9] "Select Perspective" White Paper, Princeton Softech, 2000, <http://www.princetonsofttech.com/index.asp>.
 [10] "컴포넌트 기반 개발 방법론 마르미-III 버전 4.0", 한국전자통신연구원, 2003.
 [11] Bill, B., "A Survey of Software Inspection Checklists", ACM SIGSOFT, Software Engineering Notes Vol.24, No.1, p.82, 1999.
 [12] "국방 컴포넌트 기반 방법론 지침 (ADDMe)", 국방과학연구소, 2004.
 [13] "국방정보체계 상호운용성 수준(LISI) 업무편람", 국방부, 2002.
 [14] "Level of Information System Interoperability(LISI)", C4ISR Architecture Working Group, 1998.
 [15] 김영철, "컴포넌트 기반 체계 상호운용 적합성 평가 및 인증 기술 연구", 국방과학연구소 위탁과제, 홍익대학교, 2004.
 [16] 류동국, 김우열, 김영철, "국방 정보 시스템간의 상호운용을 위한 LISI 기반 체크리스트 연구", KCSE 2005, 한국 소프트웨어공학 학술대회, 2005.
 [17] 류동국 외 3명, "국방 자동화 정보체계 상호운용성 시험 및

평가 시스템”, 한국 정보과학회지 2005년 7월호 (국방 시스템 통합 및 상호운용), 2005.

[18] “Department of Defense Dictionary of Military and Associate Terms”, Joint Publication, 2001.

류 동 국



e-mail : dkryu@lycos.co.kr
 1994년 중앙대학교 컴퓨터공학과(학사)
 1996년 중앙대학교 컴퓨터공학과
 (공학석사)
 1996년~1999년 국방정보체계 연구소 연구원
 1999년~현재 국방과학연구소 선임연구원

관심분야 : 상호운용성, 소프트웨어 성숙도 모델, 임베디드 소프트웨어 모델링, 임베디드 소프트웨어 개발 방법론

김 영 철



e-mail : bob@hongik.ac.kr
 2000년 Illinois Institute of Technology
 (박사)
 2000년~2001년 LG 산전 중앙연구소
 Embedded System 부장
 2001년~현재 홍익대학교 컴퓨터정보통신
 공학과 교수

관심분야 : 소프트웨어 성숙도 모델, Use Case 방법론 및 도구 개발, 정보 구조, 임베디드 소프트웨어 개발 방법론

<부록 1> ADDMe 단계별 상호운용 체크리스트 예제

1. 분석 단계

단계	활동	작업	산출물	질의	결과
분석 IR	요구사항정의서 (IR1)	상위요구사항정의 (IR11)	상위요구사항 정의서	상위 수준의 기능적 요구사항과 비 기능적 요구사항에 타 시스템과의 상호운용이 요구되는가?	
		현행시스템분석 (IR13)	현행시스템분석서	타 시스템과의 상호운용 부분에 대한 데이터베이스 연동은 기술되었는가?	
		비즈니스모델링 (IR14)	비즈니스정의서	외부 비즈니스 흐름에 대한 모델링은 기술되었는가?	
		요구사항 명세(IR15)	요구사항명세서	타 시스템과의 상호운용 수준은 기술되었는가?	
	아키텍처 정의(IR2)	소프트웨어 아키텍처 정의 (IR21)	소프트웨어 아키텍처 정의서	타 시스템과의 연동을 위한 현행 시스템의 소프트웨어 연동 운용 환경 구조는 정의되었는가?	
		표준지침수립 (IR23)	표준지침서	국방표준(DITA)을 준수하였는가?	
	요구사항분석 (IR3)	유스케이스모델링 (IR31)	유스케이스명세서	외부 시스템과의 상호연동이 요구사항에 따라 명세되었는가?	
		사용자 인터페이스 프로토타이핑 (IR32)	사용자 인터페이스 정의서	외부 체계와의 인터페이스가 정의되었는가?	

2. 설계 단계

단계	활동	작업	산출물	질의	결과
설계 2D	개략설계(2D1)	컴포넌트 식별 (2R11)	컴포넌트 목록	타 시스템과 연동하는 컴포넌트의 인터페이스는 식별되었는가?	
		컴포넌트 획득방법 식별(2D12)	컴포넌트 획득 방법 식별서	상호운용 가능한 상용 컴포넌트를 사용하는가?	
		인터페이스 상호작용 정의 (2D13)	인터페이스 상호작용 명세서	타 시스템과의 상호작용을 표준 모델링 언어로 명세되었는가?	
		컴포넌트 명세(2D14)	인터페이스 명세서	타 시스템과 상호운용하는 인터페이스가 식별되었는가?	
	상세설계(2D2)	사용자 인터페이스 구현 설계 (2D23)	사용자 인터페이스 구현 설계서	타 시스템과의 인터페이스는 설계되었는가?	
		데이터베이스 설계(2D24)	데이터베이스 설계서	타 시스템과의 상호운용을 위한 데이터는 식별되었는가?	

3. 구현 및 테스트 단계

단계	활동	작업	산출물	질의	결과
구현 및 테스트 3T	구현 (3T2)	데이터베이스 구축(3T21)	물리적 데이터베이스	타 시스템과의 데이터베이스 연동은 가능한가?	
		컴포넌트 구현 및 테스트(3T22)	컴포넌트 테스트 결과서	타 시스템과의 컴포넌트 수준에서 상호운용은 시험은 하였는가?	
	시스템 테스트 (3T4)	시스템 테스트 설계(3T31)	시스템 테스트 설계서	타 시스템과 상호운용 수준은 만족되는가?	
				타 시스템과의 기능적 상호운용은 가능한가?	

4. 인도 단계

단계	활동	작업	산출물	질의	결과
인도 4S	시스템 설치(4S1)	시스템 설치 계획(4S11)	시스템 설치 계획서	설치 후 타 시스템과의 상호운용성 기능 수행을 확인하는가?	