

국방시스템에서의 공개SW 활용 방안

A Research for Using Open Source Software
at the National Defense System

2006. 12

한국소프트웨어진흥원

제 출 문

한국소프트웨어진흥원장 귀하

본 연구보고서를 “국방시스템에서의 공개SW 활용 방안“의
최종 연구개발 결과보고서로 제출합니다.

2006. 12

연구 수행 기관 : 한국과학기술원 충무IT연구센터

연구 책임자 : 김진형 교수(충무IT연구센터 소장)

참여 연구원 : 이승호 교수(충무IT연구센터)

: 엄의석 교수(충무IT연구센터)

: 안태남 교수(충무IT연구센터)

: 장상구 연구원(충무IT연구센터)

요 약 문

1. 제목

국방시스템에서의 공개SW 활용 방안

2. 우리나라 공개SW산업의 현황과 국방정보화

- SW 산업은 우리나라의 주요 산업에서 차지하는 산업별 SW개발 원가 비중이 33.5%나 되는 산업의 인프라로서, 유비쿼터스 시대를 대비한 미래의 성장 동력이다.
- 그러나 국내 SW산업은 04년 기준 세계 시장에서의 한국 점유율이 패키지SW 1.7%, 디지털콘텐츠 2.7%, IT 서비스 3.3%로서 매우 열악한 상황이다.
- 정통부는 국내 SW 산업의 육성을 위해 분야별 중점과제와 세부계획의 추진을 통해 2010년 국내 SW산업 생산 53조원, 수출 50억불을 달성하도록 추진하고 있는데, 공개SW 산업은 이러한 계획에서 매우 중요한 역할을 하고있다.

가. 공개SW 산업의 육성을 위한 정책방향

- 국내 공개SW 산업의 현황은
 - 서버분야는 리눅스의 점진적인 향상으로 2010년에 27.6%를 차지할 것으로 전망되고,
 - 데스크탑은 2004년부터 데스크탑 리눅스 기술개발 및 시범사업을 실시 중에 있으며,
 - 임베디드SW는 통신산업을 기점으로 군사/항공/의료 등 전 산업으로 확대되어가고 있다.

▪공개SW산업 육성을 위한 정책 방향

- 공개SW를 활용한 SW 기술강국 진입의 비전 달성을 위하여 SW산업의 종속 탈피, 국가 주요 정보시스템의 안전성·호환성 확보, 기술혁신, 인력양성의 정책을 추진 중이고,
- 이러한 정책 결과로 신교육행정정보시스템(NEIS), 인터넷 뱅킹 등의 구축사례를 통해 해외에서도 주목 받는 공개SW 도입사례를 발굴하였고, 공개 SW 기술지원센터를 설립·운영함으로써 공개 SW 도입기관들의 유지보수 불안감 해소에 큰 기여를 하였으며, 향후의 정책방향은 서버에서 데스크탑 및 임베디드 SW분야로 확대되고 있다.

나. 국방시스템의 정보화

▪국방시스템의 정보화 방향

- 세계 정보화의 방향은 다수의 전투체계가 동시적으로 정보를 공유하는 네트워크로 조직됨으로서 전투력 발휘 효과를 크게 증대시키는 네트워크중심전(NCW)의 방향으로 발전 중이고
- 국방부는 NCW 구현과 및 u-korea 정책과 연계한 u-디펜스의 정책을 개발하고 추진 중으로
- u-디펜스는 군 전술이동통신에 와이브로를 적용하고 군용 네트워크 로봇과 USN 무인경계시스템 공동 개발 등의 협력을 통해 IT산업을 육성하기 위한 것이다.

다. 국방시스템에서 공개SW활용에 대한 연구의 필요성

▪연구의 필요성

- 국방부는 국방시스템에 공개SW의 활용을 확산하는 정책에 적극 참여하고 있음
- 2005년에는 공군본부의 국방정보체계 통합 틀 개발 및 시범체계 구축 사업을 추진하였고, 2006년에는 공개SW 도입·운영 가이드라인을 작성하여 공개SW활성화 정책에 참여하고 있다.
- 최근 미국방성에서는 NCW에 대비한 국방력 향상 및 관련 산업의 육

성을 위한 방안으로서 공개SW의 활용을 SW기술 및 산업발전의 주요 정책방향으로 고려하여 추진하고 있는 실정으로서 우리나라에 시사하는 바가 크고 국방시스템에서 공개SW활용에 대한 체계적 연구의 필요성이 제기되었으며, 이에 따라 본 연구가 추진되었다.

3. 미 국방성의 공개SW 활용현황 및 정책 연구 사례

- 미 국방성의 공개SW 관련 정책 연구 사례
 - 미국방성은 공개SW와 관련하여 2개의 중요한 정책적 연구를 수행하였는데,
 - 첫째는, 2002년 Mitre를 통해 수행한 미 국방성의 공개SW활용에 관한 연구이고
 - 둘째는, 미 국방성의 지원하에 2006년에 수행한 개방형기술개발의 적용에 관한 정책연구이다.
 - 2002년에 수행된 연구는 향후 미 국방분야에서의 공개SW 활용정책에 관한 논의의 시작점으로서의 의미를 가지는 반면 최근의 연구는 공개SW를 포함한 개방형기술의 활용을 증대시키기 위하여 수행한 좀 더 발전되고 구체화된 정책연구라고 할 수 있다.

가. 공개SW의 활용에 관한 연구

- 연구 수행 방법
 - 연구의 수행 기관은 비영리 전문 연구기관인 Mitre로서
 - 공개SW의 활용현황은 광범위한 서베이를 통해 조사한 후 이를 분석 요약하였다.
- 연구 결과
 - 115 종의 공개SW가 251개의 응용사례에서 매우 다양하게 활용되고 있었는데,
 - 보안, 인트라넷, SW 개발, 연구개발의 4분야에서 상당한 수준으로 적용되고 있다는 것이다.

나. 개방형기술개발 정책 연구

▪ 개방형기술개발의 개념

- 미 국방성은 국가의 안전보장 및 산업기반의 진흥의 두 가지의 정책 목표를 달성해야 하는데
- 개방형 표준 및 Interface, 공개SW활용, 협업적 문화, 기술적 유연성의 4가지를 수용하는 개방형기술개발 정책은 이러한 목표달성에 매우 중요한 역할을 할 것으로 기대하고 있다.

▪ 개방형기술개발의 적용 로드맵

- 개방형기술개발을 통해 공개SW의 지식베이스를 구축 관리하고 Interface, Modularity와 재사용의 내부문화를 제고하여 NCW에서 요구하는 무기체계의 획득을 위해
- 로드맵을 3단계로 구분하여 설정하고 적용대상 분야와 함께 현재의 획득 및 개발관련 행태를 공개SW 및 기술서비스에 적합하도록 변화 유도하도록 설정하였다.

다. 시사점 및 교훈

- 미국방성에서의 공개SW관련 연구를 통한 시사점 및 교훈은 다음과 같은 7가지로서
- 첫째, 미국방성에서는 이미 2002년에도 공개SW가 다양하고 수준높게 활용되고 있다는 점
- 둘째, 비용보다 필요성에 의해 공개SW를 활용하나 경쟁에 의해 가격이 조정된다는 점
- 셋째, 공개SW가 국방시스템의 보안에서 우수하다는 점
- 넷째, 미 국방성의 정책목표가 국방력 향상 및 관련 산업의 육성이 동시에 고려된다는 점
- 다섯째, 개방형기술개발은 공개SW개발모델 등이 결합된 문화로서 대상도 광범위하다는 점
- 여섯째, 개방형기술개발의 적용을 위해서는 국방획득제도의 변경이 필요하다라는 점

- 일곱째, 개방형기술개발의 성공적 적용을 위해서는 관련된 조직, 예산이 필요하다는 점이다.

4. 우리나라 국방시스템에서의 공개SW 활용현황

가. 국방시스템 관련 기존의 연구현황

- 본 연구와 직접적으로 연관된 기존의 정책연구 과제는 2가지로서
 - 첫째는, 2006년 1월에 정통부에서 발표한 공개sw 가이드이고
 - 둘째는, 국방부에서 2006년 3월에 발표한 공개sw 도입·운영 가이드로서,
 - 공개SW가이드는 공개SW 도입 시 필요사항을 작성한 가이드인 반면, 공개SW 도입·운영 가이드라인은 서버의 공개SW 전환 시 단계별 추진에 대한 내용 및 신뢰성이 높은 공개SW 21종을 추천하였다.

나. 국방전력발전규정과 SW도입절차

- 국방획득 규정과 상호 운용성 및 공개SW
 - SW를 포함한 국방시스템 관련 모든 제품의 획득은 국방획득관리규정을 통해 이루어지는데, 획득과정의 보수적인 절차, 장기적인 소요시간, 상호운용성 등의 문제가 제기되고 있었다.
 - 최근 방위사업청의 신설과 함께 이러한 문제 극복을 위하여 구 국방획득관리규정을 대체하는국방전력발전업무규정을 제정하여 시행 중이다.
 - 이 규정에서는 무기체계와 비무기체계를 새롭게 정의하였고, NCW의 구현에 필수적인 상호운용성 확보를 위한 조항들을 포함하였다.

다. 국방시스템관련 연구를 위한 공개SW 체계분류

- 국방시스템관련 공개SW 체계분류
 - 국방시스템은 일반적 정보시스템과 달리 무기체계, 비무기체계 또는 전장관리, 자원관리 등 일반적 환경과는 다른 요소가 많이 있다.
 - 이에 따라 우리나라 국방의 현실적 환경을 고려하여 무기체계에서의

무기체계용 시스템과 무기체계용 임베디드 시스템으로 구분하고, 비무기체계에서의 자원관리와 기반체제로 구분한 다음 각각에 필요한 해당 SW를 설정하고 이에 맞는 관련정책을 제안하였다.

라. 우리나라 국방시스템에서의 공개SW 활용현황

- 무기체계에서의 공개SW 활용현황
 - 지휘통제(C4I)시스템의 전장정보SW의 개발은 상용SW위주로 개발되어 있다.
 - 기동무기, 함정무기 등에서 사용하는 실시간 임베디드리눅스의 경우에도 현재로는 거의 없는 실정이고, 공개SW의 적용에 대한 비공식 설문 조사를 통해 긍정적/부정적 시각이 공존한다는 것을 파악하였다.
- 비무기체계의 자원관리체계에서는 수백종의 단위업무시스템이 개발되어 운영되고 있는데
 - 구축된 형태별로 분류하면 웹서버형, 클라이언트-서버형, 스탠드얼론형으로 구분되고
 - 상당수의 시스템들에 리눅스를 포함한 공개SW가 활용되고 있으나
 - 대부분 특정 HW에 종속된 SW로 수직계열화되고, 서버 OS에 따라 올라가는 제품군이 동일 계열로 스택화됨으로 HW(OS) 종속화 및 비용 증감 요인으로 작용하고 있으며, 일반 상용SW 환경에 익숙한 운영자들이 새로운 환경(공개SW)의 적용에 소극적인 경향이 있는 상황이었다.
- 비무기체계의 기반체계내 정보통신시스템에서는 공개SW는 아니지만 우리 독자기술로 개발되어 세계표준으로 인정된 와이브로기술을 적용하고자 추진하고 있다.

5. 국방시스템에서 공개SW활용을 위한 정책제안

가. 국방정보화의 발전방향과 우리의 여건

- 국방정보화의 발전 방향
 - 국방부는 국방정보화 추진 현상 진단 및 문제점 분석을 통한 국방 정보

화 핵심과제의 도출을 위하여 국방정보화 발전연구단을 운영하고, 국방정보화의 동향과 현상분석을 통한 발전방향을 도출하였다.

- 그 결과 현재의 정보화 방법으로는 업무개혁이 어렵고, 우리의 정보화 현황은 전형적인 굴뚝형 정보체계, 정보화 수요 대비 통신 인프라 절대 부족, 정보체계와 통신 인프라 간 심한 불균형, 신기술 도입 지연 등 많은 문제점들이 있는 것으로 나타났으며 이를 해결하기 위하여 4개의 개혁목표와 함께 세부적으로 추진해야할 과제를 도출하였다.

나. 우리나라와 미국방성의 공개SW관련 현황 비교분석

■ 우리나라와 미국방성의 공개SW 관련 현황을 비교분석한 결과

- 무기체계에서 미국방성은 수많은 공개SW를 활용하여 개발 중이고 향후에도 공개SW를 개발하는 방식을 전체무기로 확대하기 위하여 우선 첨단제품에 대해 적용코자 추진 중이나 우리나라의 경우에는 공개SW 활용 사례가 거의 없고 공개SW의 활용에 긍정적인 시각과 부정적인 시각이 공존하는 것으로 나타났다.
- 비무기체계에서 미국방성은 기반구조 분야, SW개발분야, 보안분야, 연구분야에서 상당한 수준으로 활용되고 있는데 비해 우리나라는 서버 중심으로 리눅스, DB, WAS 등의 공개SW가 다수 사용 중이지만 사례 부족으로 공개SW도입에 소극적인 경향이 있으며, 전술이동통신망에 와이브로 기술을 적용하고자 추진 중이다.

다. 국방시스템에서 공개SW 활용을 위한 정책제안

- 국방시스템에서의 공개SW 활용을 위한 적용대상 시스템은 무기체계의 군사정보, 모의훈련 등의 시스템과 비무기체계 자원관리 분야의 예산회계, 기획인사, 군수조달, 전자행정 등의 시스템으로 분석되고 있다. 이들에 대한 SW의 활용현황과 공개SW의 특성을 비교 분석한 결과 네 가지 측면에서 공개SW의 활용에 대한 필요성이 제기되고 있다.

- 첫째는 자체개발환경 및 상호운용성 확보로서, 공개SW를 활용하면 기술개발과 관련된 장점으로 사용자 측면에서는 소스코드의 수정 및 확

장, 업체의 의존탈피, 유지관리의 영속성, 호환성 확보, 정보시스템 통합의 용이성의 장점이 있고, 국가산업적 측면에서는 원천기술확보를 통한 SW산업의 육성에 기여를 할 수 있는 장점이 있다.

- 둘째는 보안체계 강화로서, 미국방성의 연구결과에서도 공개SW는 보안에 우수한 것으로 나타났고, 기간별 윈도우즈와 리눅스의 취약점 발표 비교도, 세계적인 기관들로 부터의 발표자료에 의하면 공개SW는 보안에 많은 기여를 하는 것으로 나타났다.
- 셋째는 예산절감으로서, 공개SW는 라이선스 비용이 무료이기 때문에 총소요비용에서 우수한 것으로 나타나고 있다.
- 넷째는 공개SW모범사례 확보로서, 그동안 정부의 노력으로 많은 구축 사례를 확보하고 있지만 어느 공공기관보다도 안전과 신뢰성이 요구되는 국방시스템에서의 모범적 구축사례는 공개SW의 확산에 많은 기여를 할 것이다.

라. 국방시스템에서 개방형기술개발의 활용을 위한 정책제안

- 개방형기술개발의 활용을 위한 적용대상 시스템은 지휘통제, 감시정찰, 기동무기, 함정무기, 항공무기 등의 무기체계 분야와 비무기체계에서 기반체계에 해당하는 정보통신체계, 컴퓨터체계, 정보보호체계, 상호운용성 기반체계에 해당하는 시스템으로 분석되고 있다.
- 개방형기술개발의 활용의 주요 내용은 앞서 미 국방성의 개방형기술개발의 정책연구에서 나타난 것처럼 개방형 표준 및 인터페이스, 공개SW 활용 및 설계, 협업적 문화와 온라인 지원도구, 기술적 유연성의 4가지를 활용함으로써 NCW에 필요한 무기시스템 및 비무기시스템들을 경제적으로 획득하게 하여 국가의 안전보장을 제공케 하고 또한 국가적으로 중요한 역할을 하고 있는 국가의 산업기반을 진흥하는데 기여하는 것을 목적으로 하고 있다. 2006년 12월 국방개혁법이 제정됨에 따라 앞으로 상시 병력이 68만 명에서 50만 명으로 감축되어도 실제 우리군의 전투력은 증강하기위한 목표의 달성을 위해서는 개방형기술개발의 활용이 매우 중요한 역할을 할 것이다.

6. 결론 및 시사점

가. 정부에서 해야 할 일

- 미국은 SW산업에서 세계 제일임에도 불구하고 국방력과 SW산업의 육성을 위해 공개SW의 활용을 장려하는 정책을 추진하였고, 최근에는 더욱 발전된 개방형기술개발의 정책을 추진하는 것처럼 정부에서는 우리나라에서도 공개SW의 활용을 장려하는 정책을 발굴하고 지속적인 추진을 할 필요가 있다.

나. 대학 및 연구소에서 해야 할 일

- 연구결과 대학 및 연구소에서의 공개SW 활성화 수준이 국방에서의 공개SW 활성화 수준과 비례관계를 가진 것으로 판단되기 때문에 대학과 연구소에서는 리눅스를 포함한 다양한 공개SW가 쉽고 다양하게 활용되도록 하는 정책을 추진 할 필요가 있다.

다. 산업체에서 해야 할 일

- 우리나라에서는 아직까지 공개SW에 대한 시장이 부족하여 공개SW산업의 발전에 어려움이 따르며 투자가 어렵지만 사용자의 입장에서 선호하는 SW가 지원되지 않으면 시장이 커지지 못하므로 사용자가 선호하는 SW의 개발에 산업체가 적극 동참할 필요가 있다.

- 목 차 -

1. 서론.....	1
가. 우리나라 SW산업의 발전전략.....	1
나. 공개SW 산업의 육성을 위한 정책방향.....	4
다. 국방시스템의 정보화.....	8
라. 본 연구의 목적.....	11
2. 공개SW의 특성 및 국내외 동향.....	16
가. 공개SW의 특성.....	16
나. 각 국의 정책 동향.....	18
다. 기술 및 커뮤니티 동향.....	27
라. 국내외 기업 및 표준화 동향.....	32
3. 미 국방성의 공개SW 활용현황 및 정책 연구 사례.....	37
가. 연구 배경	37
나. 공개SW의 활용에 관한 연구	38
다. 개방형기술개발 정책 연구	45
라. 시사점 및 교훈.....	52
4. 우리나라 국방전력발전 규정과 공개SW 활용현황.....	62
가. 국방시스템관련 기존의 연구현황.....	62
나. 국방전력발전 규정과 SW도입 절차.....	67
다. 국방시스템관련 연구를 위한 공개SW체계 분류.....	73
라. 우리나라 국방시스템에서의 공개SW 활용현황.....	77

5. 국방시스템에서의 공개SW활용을 위한 정책제안.....	93
가. 국방정보화의 동향과 현상분석을 통한 발전 방향.....	93
나. 우리나라와 미국방성의 공개SW 관련 현황 비교 분석	100
다. 국방시스템에서의 공개SW 활용을 위한 정책제안	101
라. 국방시스템에서의 개방형기술개발의 활용을 위한 정책제안.....	116
6. 결론 및 건의사항	123
가. 연구결과 요약.....	123
나. 결론 및 시사점.....	126
7. 참고 문헌	127

별첨목록

1. 공개SW라이선스 별 비교표.....	128
2. 미 국방성의 공개SW활용 현황 표	131

< 표 목 차 >

<표 1> Korea Server Forecast by O/S Update, 2006-2010	4
<표 2> 공개SW와 상용SW의 차이점 비교	16
<표 3> 리눅스 개발 커뮤니티 현황	30
<표 4> 미국방성에서 사용하고 있는 공개SW 리스트	33
<표 5> AS&C의 조직 및 기능에 대한 정리표	50
<표 6> 보안 관련분야에서 활용되고 있는 공개SW 리스트	57
<표 7> 비공개SW와 공개SW 비교표	65
<표 8> 정통부와 국방부의 추천 공개SW 목록 비교표	66
<표 9> 기존 및 새로운 규정에서의 무기체계 및 비무기체계 분류표	69
<표 10> 체계분류별 세부 정보체계	74
<표 11> 일반적인 공개SW 분류표	75
<표 12> 국방시스템에서의 체계 분류표	76
<표 13> 무기체계에서의 년도별 SW 점유율 비교표	78
<표 14> 육군 무기체계에서 임베디드SW 관련 현황	83
<표 15> 해군 무기체계에서 임베디드SW 관련 현황	87
<표 16> 공군 무기체계에서 임베디드SW 관련 현황	88
<표 17> 차세대형 국방통합체계 구축을 위한 핵심과제 내용	90
<표 18> 독자적 NCW 구현과 전장정보화를 위한 핵심과제 내용	93
<표 19> 제도개선을 위한 핵심과제 내용	99
<표 20> 미국방성과 우리나라 국방시스템의 공개SW 현황 비교표	100
<표 21> 국방시스템에서 공개SW활용시 기술개발과 관련된 장점	105
<표 22> 2006년 공개SW 시범과제 내용	111
<표 23> 윈도우 대비 리눅스 기능 대체표	114
<표 24> 윈도우와 리눅스의 데스크탑에서의 활용성 비교표	115

< 그 립 목 차 >

그림 1. 산업별 SW 개발 원가 비중	1
그림 2. SW에 의한 융복합화 모습	1
그림 3. 주요 IT 산업별 세계시장에서의 점유율	2
그림 4. SW산업 육성을 위한 핵심 정책 방향	3
그림 5. 리눅스 활용 분야	5
그림 6. 공개SW 추진 비전	6
그림 7. 공개SW 추진 경과	7
그림 8. 공개SW 추진 방향	7
그림 9. NCW의 개념도	8
그림 10. 공개SW 기술 hyper cycle	8
그림 11. 공개SW 사용 금지에 대한 영향도	41
그림 12. 요구사항 중심에 의한 기존의 개발 방법	47
그림 13. 목적 중심에 의한 협업적 개발 방법	47
그림 14. 개방형기술개발 로드맵	49
그림 15. 표준자원 모습	53
그림 16. 혁신에 의한 결합	53
그림 17. free market에 의한 다양한 분야에서의 혁신모습	64
그림 18. 무기체계의 획득절차에 대한 개략도	66
그림 19. 미국방성의 국방력향상을 위한 체계도	61
그림 20. 무기체계 및 비무기체계의 획득 세부 절차도	70
그림 21. 우리나라 국방정보화 체계도	73
그림 22. 전장정보시스템과 C4I의 관계도	78
그림 23. 전장정보시스템의 개념도	79
그림 24. 전투지휘시스템의 사례도	81
그림 25. 우리나라 C4I 체계의 구축 목적 및 수단	82

그림 26. 군단 ~ 연대급 부대의 C4I 배치 사례도	8
그림 27. 임베디드 시스템의 개념도	8
그림 28. 웹서버형의 구성도	9
그림 29. 클라이언트-서버형의 구성도	9
그림 30. 육군정보시스템의 보편적 구성 형태도	9
그림 31. 현재의 정보화 방법에 의한 악순환 개념도	9
그림 32. 국방통합정보지원센터를 통한 NCW 구현의 개념도	98
그림 33. 전장관리와 자원관리 체계간의 연동 제한 모습	102
그림 34. 건축분야와 정보화분야의 EA개념 비교도	103
그림 35. 안전 및 신뢰성을 위한 표준 모델의 구성도	106
그림 36. 상호운용성이 확보된 체계의 개념도	106
그림 37. 윈도우 XP와 리눅스의 초기 취약점 발표 횟수 비교표	107
그림 38. NEIS 시스템의 사업 목표	112
그림 39. 국방개혁법 제정안 주용 내용	117
그림 40. 미국방성의 무기체계 개발 방식	117
그림 41. 블랙박스 형태의 요구사항과 시스템 구현의 관계	118
그림 42. 지휘관의 요구사항과 생산성의 관계 비교도	118
그림 43. 전술이동통신망의 구성도	119
그림 44. 와이브로 개념도	120
그림 45. 전술데이터링크의 개념도	121

1. 서론

가. 우리나라 SW산업의 발전 전략

1) SW 산업의 중요성

- SW 산업은 단순한 하나의 산업이 아니라, 모든 산업의 경쟁력을 높여주는 인프라로서 텔레콤, OA, 가전, 자동차 등의 우리나라의 산업의 근간을 이루는 주요 산업에서 차지하는 산업별 SW개발 원가 비중이 33.5%이다.

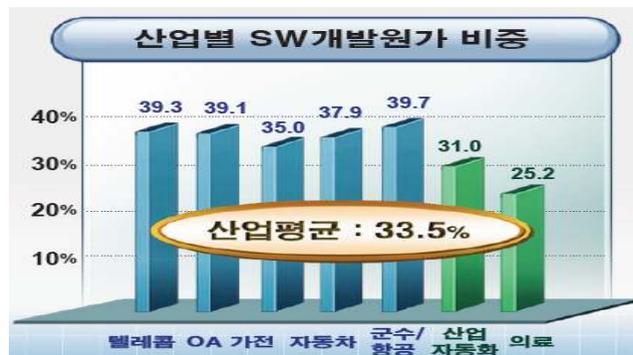


그림 1. 산업별 SW 개발 원가 비중

- 특히 유비쿼터스 시대가 열리면서, SW와 네트워크를 중심으로 가정, 국방, 의료, 교육 등 사회 전체적으로 SW에 의한 융복합화가 가속되고 있는 추세이기 때문에 SW산업의 발전이 없이 IT산업의 발전은 지속될 수가 없고, IT산업의 발전이 없이는 경제발전이 어렵다고 볼 수 있다.

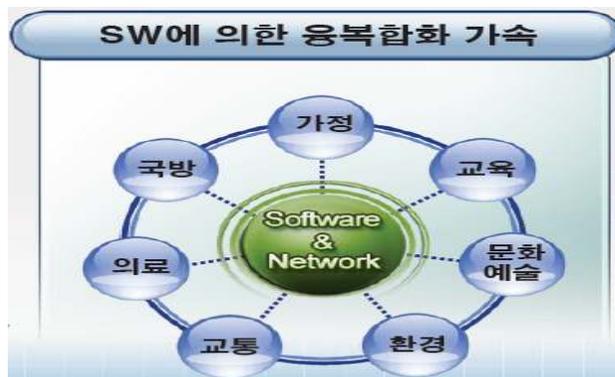


그림 2. SW에 의한 융복합화 모습

2) 국내 SW 산업의 현황

- 그러나, 국내 SW산업의 현황을 보면, 업체들의 영세성과 과당경쟁, 취약한 수익구조, 요소기술의 부재, 핵심 기술의 해외 의존 등 악순환의 구조에서 벗어나지 못하고 있으며, 이에 따라 04년 기준 IT산업관련 주요 항목들에 대해 세계 시장에서의 한국 점유율을 비교해 보면, 하드웨어에 산업분야는 메모리 47.1%, LCD 41.4%, 휴대폰 27.8% 등인데 반해 SW산업분야는 패키지SW 1.7%, 디지털콘텐츠 2.7%, IT 서비스 3.3%로서 열악한 상황임을 알 수 있다.



그림 3. 주요 IT산업별 세계 시장에서의 점유율

- 90년대 후반 벤처붐을 통해 급성장한 국내 SW산업은 그동안 산업 기반이 조성된 시기였고, 이제는 새로운 성장단계로 진입하는 단계이다.
- 2005년 국내 SW시장규모는 61억불(Spending 기준)로 전 세계 SW 시장에서 0.9%의 점유율(IDC, Blackbook, 2005.4Q)을 차지하고 있다.
- 2000년 후반 코스닥 하락과 세계적인 IT경기침체에 따라, 2002년 이후 심각한 경기침체를 경험하였고 SW가치인식, SW기업의 수익성 문제 등 구조적 문제에 봉착하였으나, 지금은 SW산업의 새로운 성장기로 도약해야하는 단계로서, 유비쿼터스라는 기술패러다임 변화와 공개 SW를 중심으로 한 시장경쟁구조 변화가 가지고 온 새로운 사업기회, 이미 구축된 산업기반, 적극적인 SW기업, SW인력, 정부라는 인적자원 등 도약을 위한 조건이 모두 마련되어 있다.

3) 국내 SW 산업 발전을 위한 핵심 정책 방향

- 정통부는 위와 같은 국내 SW 산업의 현황을 인식하고, 분야별 4대 중점과제와 이에 따른 세부계획을 발표하고 이를 통해 2010년 국내 SW산업 생산 53조원, 수출 50억불을 달성하도록 추진하고 있다.
- 먼저 IT서비스 기업의 전문화는 국가차원에서 대형 공공 지식정보 프로젝트를 발굴 및 추진하고 아울러 전자정부 및 IT839 성과물인 와이브로, DMB 등의 구축경험을 지식서비스화해 전략적 해외진출을 추진할 계획이며, IT서비스 기업의 기술 및 서비스 역량을 제고하기 위해 공공SW사업 발주 시 업종 전문화기업에 대한 우대방안을 강구하는 한편, 중소기업의 국제품질인증 획득을 지원해 나갈 예정이다.
- 임베디드SW 고급인력 양성을 위해 IT839에 적용되는 대형 임베디드SW 개발과제를 발굴 및 추진하고, 글로벌기업과 국내 대학, 연구소와의 합동연구를 지원해 아키텍트급 고급인력을 양성할 계획이다.
- 패키지SW의 중견기업 육성에서는 SW 원천기술 부족해소를 위해 리눅스 등 공개SW의 보급과 확산에 주력할 방침으로서, 기존 서버위주의 보급사업을 일반 PC로 확대하여 공공분야 사무용 PC의 리눅스 설치를 추진하고 공개SW 시범도시 및 시범대학 등을 통해 각급 기관의 참여를 적극 유도해 나갈 예정이다.
- 디지털컨텐츠 세계 일류기업 양성에서는 디지털 한류의 선봉이며 세계적인 경쟁력을 보유하고 있는 온라인 게임의 세계시장 지배력을 지속적으로 확보해 나가기 위해, 언어나 플랫폼 변환 등 현지화 지원을 강화하는 한편, 국내 우수기술과 해외자본과의 연계를 통하여 중국 등 외국시장 진출을 가속화 해 나갈 계획이다.



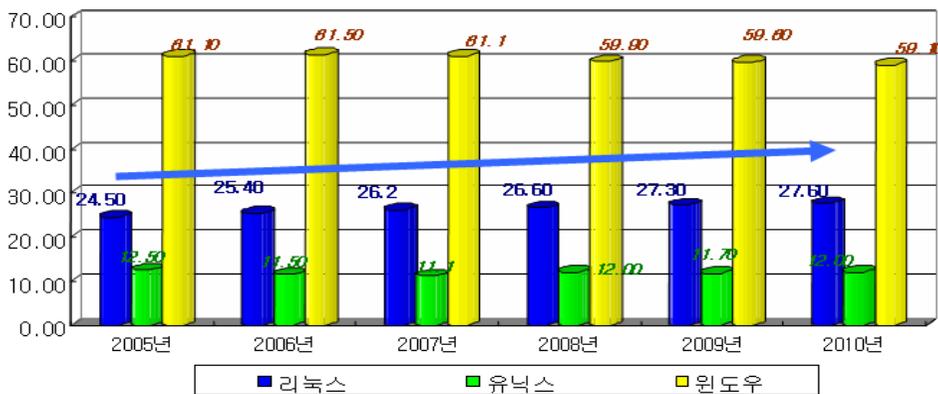
그림 4. SW산업 육성을 위한 핵심 정책 방향

나. 공개SW산업의 육성을 위한 정책 방향

1) 공개SW 산업의 현황

- 서버시장은 리눅스의 점진적인 향상으로 인해 2010년에는 27.6%를 점할 것으로 전망되며, 반면 유닉스 및 윈도우는 점차 감소 추세
- 세계 서버시장 또한 2010년에는 리눅스가 26.1%를 차지하며, 반면 유닉스는 6.5%로 급격한 하향세를 보일 것으로 전망

<표1> Korea Server Forecast by O/S Update, 2006-2010 (shipment, %)



※ 출처: IDC 2006

- 서버 분야에서는 중소형 서버에서 중대형 서버로 리눅스가 시장을 확대
 - 4way 서버에서 8, 16way로 그 성능이 비약적으로 발전, 64bit 컴퓨팅이 상용화
 - 향후 지속적인 서버분야의 리눅스의 약진은 계속 될 것이며, 그 방향은 중소형서버에서 대중소 서버 전체를 아우르는 형태로 진행
- 리눅스의 적용 범위는 커널 2.6이 발표되면서 기능 및 성능이 향상되어 기업 시장으로 활용 분야 확산 증으로, 아래 그림에서 영역 1, 2는 재 시장에서 활발하게 적용되고 있는 분야이고 영역 3은 차후 도입이 예상되는 분야이다.

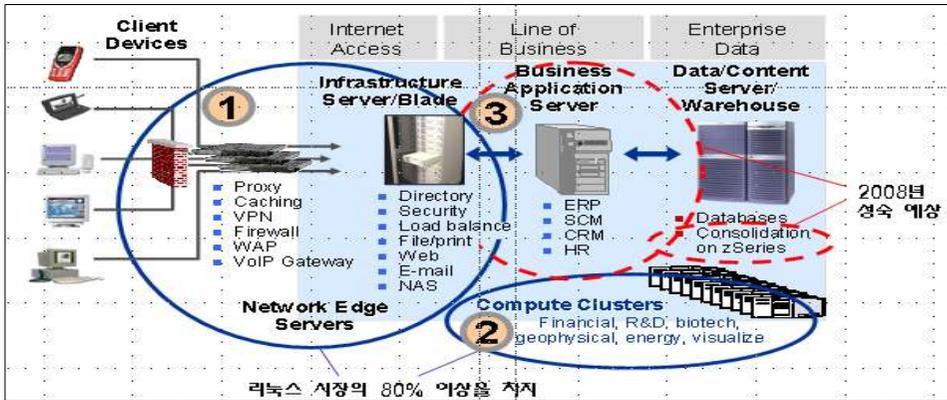


그림 5. 리눅스 활용 분야

- 리눅스 데스크탑의 경우 SW산업 육성, 예산절감 등 이유로 각 국은 리눅스 PC 보급을 적극 지원 중에 있으며 국내는 04년부터 데스크탑 리눅스 기술개발 및 시범사업을 실시 중에 있음
 - 영국 경찰청, 독일 뮌헨시 등은 데스크탑의 OS를 리눅스로 전환(05.7)
 - 중국 장수성은 학교에 14만대의 리눅스 데스크탑 설치 결정(05.10)
 - 미국 인디애나주는 30만 고등학생 대상 리눅스 데스크탑 보급(05.8)
 - 산연 공동연구로 국내 표준(Booyo)를 개발하여 기술이전상용화(05.7)
 - 한국SW진흥원 업무용PC(200대)와 우체국 인터넷플라자PC(220대) 전환
 - 04년 기준 국내 공공기관의 리눅스 데스크탑 사용률은 0.69%(한국전산원, 05.1)
- 임베디드 SW 활용분야는 유선통신, 무선통신, 정보가전, 자동차제어, 산업 자동화, 군사/항공우주, 의료장비 등 전 산업으로 확대 추세임

2) 공개SW 산업의 육성을 위한 주요 정책 방향

- 정통부에서는 SW산업의 증속 및 독과점 구조 탈피, 국가 주요 정보 시스템의 안전성·호환성 확보, 기술혁신, 인력양성을 통한 국내 SW 산업 경쟁력 강화를 위하여 공개SW 활성화 추진 확정
- 공개SW 추진 Vision

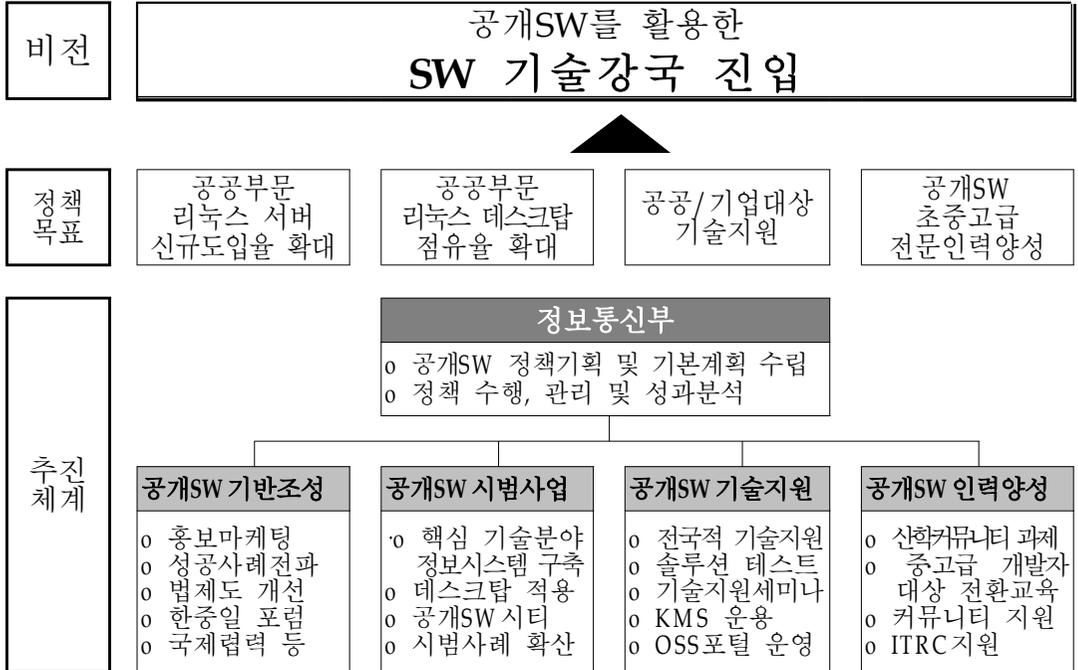


그림 6. 공개SW 추진 비전

- 정책 성과 : 공개 SW 활용도를 실질적으로 제고할 수 있는 정책의 추진
- 신교육행정정보시스템(NEIS), 인터넷 뱅킹 구축사례 등을 통해 해외에서도 주목 받는 공개SW 도입사례를 발굴
- 2006년 정부정보화사업 37개에 공개SW 도입을 확정하는 등 정부부처의 적극적인 참여를 유도하여 공공시장을 활성화
- 또한, “공개 SW 기술지원센터”를 설립·운영함으로써 공개 SW 도입기관들의 유지보수 불안감 해소에 기여

■ 추진 경과

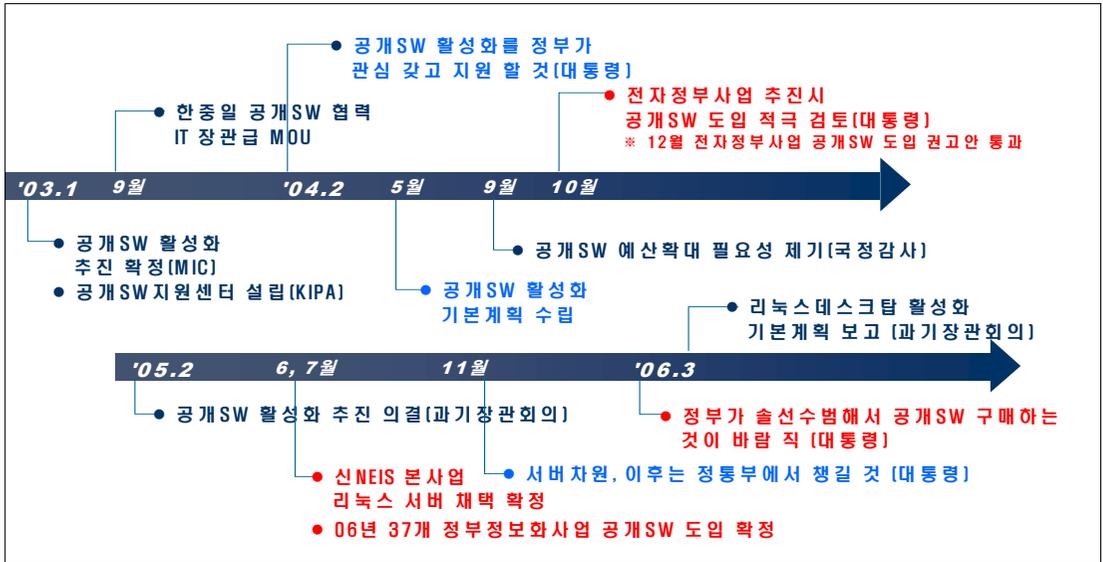


그림 7. 공개SW 추진 경과

■ 공개SW 적용 추진 방향



그림 8. 공개SW 추진 방향

다. 국방시스템의 정보화

1) NCW 구현을 위한 유비쿼터스 국방

- 우리나라 국방시스템의 정보화 방향은 NCW(네트워크중심전:Network Centric Warfare)을 위한 유비쿼터스 국방 추진으로서, 이는 미래의 전쟁환경이 지리적 공간과 가상공간 및 전자적 공간 등으로 영역이 확장되고, 정보전, 도시전 등과 같이 비대칭 및 분산 작전으로 달라지므로, 종전의 무기나 플랫폼 단위로 사용하던 종래의 개념에서 다수의 전투체계가 동시적으로 정보를 공유하는 네트워크로 조직됨으로서 전투력 발휘 효과를 크게 증대시키는 새로운 방법이다. 이러한 개념은 국가에서 추진하는 u-korea의 정책과도 부합되며 국방시스템을 미래 전쟁환경에 맞게 개선함과 동시에 국방산업을 IT산업과 연계할 수 있는 좋은 정책이라고 할 수 있다.



그림 9. NCW의 개념도

- 그러나 NCW 구현에 가장 앞서고 있는 미국에서는 NCW 구현을 위한 기간단축과 비용절감을 가져오는 여러 가지 제도와 절차를 획득 전략에 반영하고 있고, 일련의 전투수행개념과 획득을 위한 소요기획도 상향식에서 하향식으로 변경하고 있는 실정이고, 우리는 NCW 추진의 당위성은 공감하고 있으나 추진 실적은 미미한 가운데 구현전략이 구체화되는 과정이기 때문에 선진국의 제도 및 절차를 심도 깊게 연구한 후 우리나라에 적합한 정책을 모색해야 할 것으로 보인다.

2) 국방산업을 통한 IT 산업 육성 정책

- 정통부의 IT839의 정책은 미래 사회에 전개될 IT와 그 정책 패러다임이 변하면서 유비쿼터스 사회 조성을 목표로 하고 있고, 유비쿼터스 사회 조성을 위한 필수 과제로 u-시티, RFID/USN, 텔레매틱스, 홈네트워크, u-디펜스, u-페이먼트, u-헬스 등 7개 선도과제가 선정되어 정책을 추진하고 있다.
- 이 중 가장 기본적이면서 u-서비스의 근간이 될 분야는 바로 u-시티로서, 현재 각 지방자치단체에 의해 계획/추진되고 있고, 우선적으로 수도권과 신도시 위주로 해당 지역 특성에 맞춘 u-시티 구축이 한창 진행 중이며, 지역 특성에 맞는 u-시티 구축 중 화성 동탄 및 과주 운정 등 신도시에는 첨단형 u-아파트 같은 주거형 서비스로, 인천 송도 및 상암 DMC 등은 R&D 등 산업/비즈니스형 서비스로, 부산, 광주, 제주 등 행정중심 복합도시는 종합 u-시티로 개발되고 있다.
- 국방시스템에서 추진하고 있는 u-디펜스의 경우, 국방부와 국방정보화 협력계획을 확정해 군 전술이동통신에 와이브로를 적용하고 군용 네트워크 로봇과 USN 무인경계시스템 공동 개발 등을 진행 중에 있으며, 2010년까지 유비쿼터스 시범부대를 창설할 계획이다.
- 2005년 육군 탄약관리 시범 사업과 공군 F-15K 자산관리 시범사업
- 2006년의 탄약관리 체계 확산 사업, 해병대 출입통제체계 시범사업 등 추진
- 보급·수송·정비 등 3개 군수 업무를 중심으로 한 u-군수의 추진 등
- 국방부와 정통부의 이러한 협력을 통해 국방부가 우리나라 성장 동력인 IT839의 테스트베드 역할에 적극 나섬으로써 IT산업의 경쟁력 강화는 물론 미래 네트워크 전쟁에 대한 방비에도 상당한 도움이 될 뿐만 아니라 국방 분야에서 IT839 수요가 창출된 후 IT839 기반의 국방기술 개발로 민간 산업 분야의 신규 수요도 창출될 것으로 예측되고 있다.

3) 국방시스템에서의 공개SW 활용에 대한 현황 및 문제점

가) 국방시스템에서의 공개SW 활용 정책

- 국방부는 정통부와의 협력을 통해 국방산업에 공개SW의 활용을 확산하는 정책에 적극 참여하고 있고, 이러한 사례로서 2005년도에는 정통부의 공개SW시범사업에 참여하여, 공군본부의 국방정보체계 통합 틀 개발 및 시범체계 구축 사업을 추진하였다.
- 2006년 3월에는 ‘공개SW 도입·운영 가이드라인을 작성하여 각 군 연대급 부대에까지 배포하고 가이드라인으로 애플리케이션 SW는 정통부에서 국산 SW 육성을 위해 추진하는 GS제도를 지원하기 위해 GS인증제품을 도입하라는 내용을 각급 연대급까지 배포함으로써 국방정보종합처리체계(MIMS), 국방동원시스템, 국방 이포털구축사업 등에 GS인증제품이 사용되도록 하였다.
- 또한 2006년도 공개SW의 시범사업에 참여한 국방부의 사업은 육군 교육사 전투지휘훈련단의 공개SW 기반 한국형 위게임(WarGame) 시스템 구축사업으로, 그동안 핵심 기반 요소 중 국외제품 비중이 절대적이었던 군내 위게임 시스템을 한국형 리눅스 등 국산 공개SW로 전환하는 것이다. 전투지휘훈련단은 우선 위게임 모델을 32비트의 국산 공개SW로 전환하고, 이후 이를 64비트로 업그레이드할 계획이다. 또 시스템 장애로 인한 모의훈련 중단을 막기 위해 주요 시스템을 클러스터화 하는 한편, 원격 모의훈련 지원을 위해 보안성도 강화할 계획이다. 전투지휘훈련단은 또 향후 모든 위게임 모델에 한국형 공개SW 적용과 국산화를 추진키로 했다.
- 국방부의 공개SW 전용 교육체계 구축 사업은 전역 이후 국내 전산 분야를 담당할 군 정보화 담당 장교, 부사관 등을 대상으로 공개SW 인식교육을 통해 향후 리눅스 기반의 공개SW 사용 확대를 유도할 수 있다는 점에서 의미가 큰 것으로 받아들여지고 있다. 국방부는 정보화 관련 하드웨어, 공개SW의 이론 및 실습교육이 가능한 전용 컴퓨터 교육장을 육·해·공군 정보통신학교에 각 1씩씩 구축할 계획이다.

라. 본 연구의 목적

1) 연구의 필요성

- 국내 공개SW 관련 현황
 - 앞에 기술한 바와 같이 SW산업은 주요산업에서 차지하는 비중이 원가 비중이 33.5%로서 유비쿼터스 시대를 대비한 미래의 성장동력임에도 불구하고 2004년 기준 국내SW산업은 세계 시장에서 한국의 점유율이 패키지SW 1.7%, 디지털콘텐츠 2.7%, IT 서비스 3.3%로서 매우 열악한 상황이다.
 - 정통부는 위와 같은 국내 SW 산업의 현황을 인식하고, 분야별 4대 중점과제와 이에 따른 세부계획을 발표하고 이를 통해 2010년 국내 SW산업 생산 53조원, 수출 50억불을 달성하도록 추진하고 있으며, 공개SW는 이러한 목표달성을 위한 주요과제 중의 하나로서 원천기술의 확보에서 유리한 공개SW의 보급과 확산에 주력하는 정책을 추진하고 있다.
- 미 국방성에서의 공개SW 관련 추진 현황
 - 해외에서도 상용SW에 의한 독과점에 따른 피해를 해소하기 위해 공개SW의 활성화를 위한 정책을 추진하고 있는데, 독일, 프랑스, 영국 등의 유럽에서는 기술개발을 통한 기반조성에 주력하고 있는 반면에 남미, 일본, 중국 등에서는 수요창출에 많은 노력을 기울이고 있다.
 - 특히 미국의 국방성에서는 기술개발, 인프라 및 보안의 측면에서 공개SW의 중요성을 인식하고 2002년 MITRE를 통해 미 국방성에서의 공개SW 활용현황에 대한 연구를 통해 국방시스템에서의 공개SW 활용현황을 정확히 파악하고 분석한 후에 정책의 방향을 설정하였고, 2006년에는 공개SW의 개발 모델을 포함한 포괄적 개념의 개방형기술개발에 대한 연구를 수행하고 미래의 NCW에 필요한 무기체계를 효율적으로 개발하여 국방력 향상 및 관련산업 육성을 위한 정책을 추진하고 있다.

■ 연구의 필요성

- 미 국방성에서의 공개SW에 관련된 연구는 우리나라에 시사하는 바가 매우 많은데, 왜냐하면 우리나라 국방정보화의 방향은 미국과 같이 미래를 대비한 NCW이고 국방부와 정통부가 협력하여 첨단국방정보화를 위한 u-디펜스의 정책을 추진하고 있으며 공개SW의 원천기술은 이러한 정책에서 매우 중요한 요소이기 때문이며, 이에 따라 미국방성의 공개SW관련 정책에 대한 세부내용 분석과 국내의 현황 분석 및 정책에 대한 연구가 필요하다.

2) 연구의 목적

- 본 연구의 목적은 우리나라 국방시스템에서의 공개SW활성화를 위한 정책의 제안으로서 아래와 같은 2가지의 연구가 포함되어 있다.
 - 첫째는 미국방성에서 공개SW와 관련하여 연구한 2002년의 MITRE의 연구 내용 및 2006년의 개방형기술개발 로드맵에 대한 연구 내용의 세부적 분석이다.
 - 둘째는 우리나라 국방시스템에서 공개SW와 관련된 현황 분석을 통한 공개SW 활성화 정책의 제안으로서 미 국방성에서는 현황분석을 위해 서베이를 수행하였으나 본 연구에서는 일정 및 예산에 따라 각 분야 전문가들의 초청 및 의견수렴을 통하여 현황을 파악하고 미국방성과 우리나라의 현황에 대한 비교분석을 통해 우리나라 국방시스템에 적합한 공개SW 활성화 정책을 제안하고자 한다.

3) 연구의 개요

■ 연구 개요

- 과제 명 : 국방시스템에서 공개SW 활용을 위한 정책 연구
- 연구 기간 : 2006년 7월 28일 ~ 2006년 12월 5일
- 미 국방성의 사례 연구 분석
 - 미국 국방성에서 2002년 MITRE를 통해 수행한 공개SW활용에 대한 정책연구 보고서의 심층적 연구
 - 미국 국방성에서 2006년 각계전문가들과 공동으로 수행한 개방형 기술개발 정책연구보서서의 심층적 연구
- 우리나라 국방시스템에서의 공개SW 활용에 대한 정책 제안
 - 전문가 초청 및 의견 수렴을 통한 우리나라 국방시스템에서의 공개 SW 활용 현황 분석
 - 미 국방성과 우리나라 국방시스템과의 차이점에 대한 연구
 - 미국과 우리나라 국방시스템간의 상호간 차이점을 비교한 후 우리나라 국방시스템에서의 공개SW활용에 대한 정책을 제안함

■ 세부 사업내용

- 우리나라 국방시스템에서의 공개SW 활용을 위한 정책을 제시하기 위한 연구는 다양하게 나타날 수 있으나 본 과제에서는 아래와 같이 2가지 주제와 4가지의 하위 과제의 연구를 통해서 수행하는 것으로 구체적인 연구의 내용과 범위를 제안하고자 한다.

▪ 1주제 : 미 국방성의 사례 연구 분석

1. 미 국방성의 공개SW 관련 정책 연구 조사 분석
<p><연구 내용></p> <ul style="list-style-type: none"> ○ 2002년 MITRE의 연구 내용을 분석하여 미국방성에서의 공개SW 활용 현황을 조사분석한다. ○ 2006년 미국방성에서 각계 전문가들과 공동연구한 개방형기술개발 정책연구보고서의 내용을 조사 분석한다.
<p><연구 범위></p> <ul style="list-style-type: none"> ○ 미 국방성의 공개SW 활용 현황 조사는 MITRE 사에서 작성한 보고서로 한정하고, 개방형기술개발 정책에 대한 내용분석은 미 국방성에서 발표한 개방형기술개발 도드맵으로 한정하고, 기타 관련된 내용은 인터넷을 통한 자료를 통해 조사한다.
2. 미 국방성에 제안한 정책 내용의 심층 분석
<p><연구 내용></p> <ul style="list-style-type: none"> ○ 미 국방성의 공개SW 활용현황 및 정책제안의 내용을 연구보고서의 분석을 통해 정확히 조사 분석한다. ○ 미 국방성의 개방형기술개발에 대한 정책의 내용을 연구보고서의 분석을 통해 정확히 조사 분석한다.
<p><연구 범위></p> <ul style="list-style-type: none"> ○ 미 국방성에서의 공개SW 활용 현황 연구 <ul style="list-style-type: none"> - 활용 현황, 정책 대안 별 비교 및 정책 제안 내용 ○ 미 국방성에서의 개방형기술개발에 대한 정책 연구 <ul style="list-style-type: none"> - 개방형 기술개발의 내용, 정책 제안 내용 및 추진 로드맵

○ 2주제 : 우리나라 국방시스템에서의 공개SW 활용에 대한 정책 제안

1. 우리나라 국방시스템과 미 국방성의 공개SW 관련 차이점 비교
<p>< 연구 내용 ></p> <ul style="list-style-type: none"> ○ 우리나라 국방 시스템에서의 공개SW 활용 현황을 전문가 초청 및 의견 수렴을 통해 분석한다. ○ 분석된 자료를 통해서 우리나라 국방시스템과 미국 국방성의 차이점을 비교한다.
<p>< 연구 범위 ></p> <ul style="list-style-type: none"> ○ 미 국방성은 공개SW의 활용에 대한 현황조사를 위해서 survey를 실시하였고, 우리나라와 미국의 차이점을 정확히 비교하기 위해서는 우리나라도 survey가 필요하나 이 연구과제에서는 현실적 여건을 고려하여 전문가 초청 및 의견 수렴을 통해서 차이점을 비교한다.
2. 우리나라 국방시스템에서의 공개SW 활용에 대한 정책 제안
<p>< 연구 내용 ></p> <ul style="list-style-type: none"> ○ 우리나라의 국방시스템에서 고려해야 될 정책을 제안한다.
<p>< 연구 범위 ></p> <ul style="list-style-type: none"> ○ 우리나라는 공개SW산업의 활성화를 위해 지금까지 많은 정책을 추진하여 왔고, 이를 통해 어느 정도 공개SW 활성화를 위한 성공적인 결과가 나타났으나, 미국 방성과 같이 우리나라 국방시스템에서 공개SW가 적극적으로 활성화 되면 공개 SW산업의 활성화에 큰 기여를 할 것으로 기대된다.

2. 공개SW의 특성 및 국내외 동향

가. 공개SW의 특성

■ 공개SW의 정의

- 공개SW란 소스코드를 공개한 상태로 실행프로그램을 제공하는 소프트웨어로서, 소스코드를 누구나 자유롭게 사용·제작·재배포할 수 있도록 허용한 소프트웨어이다. 공개SW는 누구나 소스코드를 읽을 수 있고 사용자가 능력이 있다면 각종 버그의 수정은 물론이고 그것을 제작하여 기능을 추가할 수 있으며, 누구나 소프트웨어 개발에 참여할 수 있다. 따라서 공개SW는 소스코드에 접근할 수 있는 권리, 프로그램을 복제하여 배포할 수 있는 권리, 프로그램을 개선할 수 있는 권리를 개발자에게 보장한다. 이러한 공개SW는 소스코드에 대한 접근성이 보장되므로 시스템간의 호환성을 확보할 수 있을 뿐 아니라 사용자의 요구에 부합하는 일관성과 함께 일치성을 보장받을 수 있다.

■ 공개SW의 특성

- OSI에서 정의하는 공개소프트웨어 조건은 자유배포, 소스코드 공개, 2차적 저작물, 소스코드 수정제한, 개인이나 단체에 대한 차별 금지, 사용분야에 대한 제한 금지 등 10개 항목이 있다.
- 공개SW와 상용SW의 차이점은 다음 표와 같다.

<표2> 공개SW와 상용SW의 차이점 비교

구분	공개SW	상용SW
소스코드 접근성	가능	불가
도입 비용	유료 또는 무료	유료
개발 및 공급자	다수 개발자(기업) 다수 공급자	단일 개발기업 다수 공급자
수익모델	서비스 중심	라이선스료 중심

※ 출처 : 공개SW 가이드

- 대표적 공개SW인 리눅스의 경우 리눅스 토발즈는 초기 버전인 10만 여 라인 정도의 리눅스 커널을 인터넷을 통해 공개한 후, 짧은 기간에 100만 여 라인 정도의 리눅스 커널로 발전시켰다. 이점이 공개 SW의 가장 큰 강점이자 무한한 발전 가능성을 내포한 소프트웨어 개발 모델이다.
- 공개(open source)라는 단어는 1998년 1월 팔로알토의 넷스케이프사에서 네비게이터의 소스 코드의 공개를 발표하면서 언급된 것을 시초로, 현재 8만 여 개의 공개SW 라이선스 하에 개발되어 있으며, 10만 이상의 공개SW 프로젝트가 활발하게 진행되고 있다. 산업의 고도화 및 고부가가치화가 가속화되는 현 상황에서 차세대 성장 산업으로 손꼽히는 소프트웨어 산업의 중요성이 날로 부각되고 있고, 현재 사용중인 대부분의 기반 소프트웨어들이 외산 소프트웨어이기 때문에, 공개SW 산업을 기반으로 한 소프트웨어 개발의 촉진 및 활성화가 요구되는 시점이다.
- 우리나라처럼 신흥 소프트웨어 개발국이 원천기술 확보를 통해 대선진국 기술격차를 해소할 수 있는 대안으로 공개SW가 부상하고 있다. 특히 21세기 지식기반경제에서는 고부가가치를 창출하는 소프트웨어 산업이 국가경쟁력의 핵심 산업이며, 제조업, 서비스업, 정보통신산업 등 주요 기반산업의 경쟁력을 향상시키는 원천이기 때문에 그 중요성이 날로 부각되고 있다.
- 그러나, 세계 소프트웨어시장은 MS, IBM, Oracle 등 소수의 다국적 기업에 의한 시장지배, 특히 수직계열화에 의한 독과점화로 인하여 후발기업의 시장진입이 갈수록 어려워지고 있다. 이러한 수직계열화에 의한 독과점화는 국가 주요 정보시스템의 설계가 특정 벤더에 종속되어 독자적인 보안정책 수립, 유지보수·확장, 시스템 호환성 확보에도 걸림돌로 작용하는 문제를 동반하게 된다. 대표적인 공개SW인 리눅스 기반의 시장은 초기 시장형성 단계에 있기 때문에 초기시장을 선점하는 기업 및 국가가 경쟁력을 확보할 수 있는 대안으로 대두되고 있다.

- 공개SW와 비교되는 SW로는 프리웨어, 셰어웨어, 애드웨어 등이 있다. 이들 소프트웨어와 공개SW의 유사점은 무료로 사용할 수 있다는 점이고 가장 큰 차이점은 이들 소프트웨어는 소스코드가 공개되지 않는다는 점이다.
 - 프리웨어는 라이선스 비용없이 무료로 배포되는 소프트웨어로 최종 사용자가 대금을 지불
 - 셰어웨어는 대개 판매를 목적으로 제작되지만 일단 사용해 보고 마음에 들면 구입할 수 있도록 하는 프로그램임
 - 애드웨어는 프로그램을 기능이나 날짜상의 제한 없이 무료로 사용하는 대신 해당 소프트웨어로 작업하는 동안 광고 창을 통해 계속적으로 새로운 광고가 노출되도록 한 새로운 마케팅 기법으로 등장한 소프트웨어임

나. 각국의 정책 동향

1) 국외 동향

- 공개SW 활성화 정책은 1990년대 말부터 동북아시아, 유럽, 중남미를 중심으로 활발히 전개되어 왔다. 공개SW 지원 정책은 공개SW 관련 연구 개발 확대 등 공급을 촉진하는 기술 공급 정책과 공개 소프트웨어 사용을 장려하는 수요 촉진 정책으로 나눌 수 있다. 주로 유럽 지역은 개발 촉진 정책을 강화하고 있으며 동북아, 중남미 국가들은 수요 촉진 정책을 상대적으로 강화하고 있다. 대부분의 국가들이 기술 공급 정책보다는 수요촉진 정책을 중심으로 정책을 추진하고 있다.
- 국가별 공개SW 정책 동향
 - 중국
 - 중국은 국가가 투자하는 프로젝트에서 SW를 조달하는 경우 리눅스 관련 제품의 채택을 권고하고 있는데, 정보산업부는 리눅스를 중점사업 분야로 선정하여 산하기관인 CCID(China Center for Information Industry Development)NET Investment를 통해 리눅스

스 기업에 직접투자를 한다.

- 중국 정보산업부는 리눅스를 중점사업 분야로 선정하여 정보보안, 기업용, 오피스 등을 개발하고 있으며 중국 내 대학, 기업들이 연합해 리눅스기반의 독자 운영체계를 개발 중이다.
- 2001년 3월 정보산업부 산하 CCIDNET Investment가 홍기리눅스에 투자하면서 리눅스 정책이 구체적으로 실현되기 시작되었으나 아직 중국 OS중 리눅스 사용율은 2% 미만이다. (CCID, 2002)
- 2001년 발표된 18호 문건의 제 25 조에서 국가가 투자하는 주요 프로젝트는 국내 기업에 우선적으로 위탁하고 동등 성능일 경우 가격이 낮으면 중국산 S/W(공개 S/W는 아님)를 구매하는 원칙을 정하고 정부 조달시 리눅스 관련 제품 채택을 권고하고 있다.
- 현재 공공기관, 북경시정부 등에 리눅스 도입이 활발히 진행되고 있으며, 베이징과학기술위원회 부국장은 “중국은 리눅스가 소프트웨어 산업을 발전시킬 수 있는 가장 중요한 기회, 중국정부는 컴퓨터 구입시 리눅스를 우선적으로 고려할 것”이라고 밝히고 있다.

▪일본

- 경제기술산업성 공개소프트웨어기술개발 프로젝트 추진하고 있으며, 2003년 신규사업으로 오픈소스 S/W 활용기반 정비사업(20억엔), 디지털정보기기 상호운영기반 프로젝트(25억엔)가 수행되고 있다.
- 전자정부 구현시 공개 S/W 활용가능성에 대한 타당성을 검토하고 있다.

▪미국

- 미국의 공개SW 기반조성 정책은 경제, 보안, 특정기업의 독점방지의 목적으로 오픈 소스 활성화 지원하고 있으며, 국가보안국(NSA)이 소스가 공개된 컴퓨터 운영체제인 리눅스의 운영을 확대하기 위해 리눅스 지원 추진하고 있다.
- GNU, FSF(Free Software Foundation) 등 민간 차원에서 공개 S/W관련 다양한 정책이 활발히 추진되고 있다. 또한 미국정부가

“자유 S/W의 사용 권장”을 2001년 8월 29일 공시하였다.

- 미국의 공개SW 수요 창출 정책은 국방성, NSA, 농무부·에너지부·공군·연방항공청 등에 리눅스 보급이 활발하다.

■EU

- EU의 프레임워크 IST(Information Society Technologies)프로그램에서 오픈소스 S/W 기술개발 프로젝트를 시행하였으며, IDA(Interchange of Data Between Administrations) 프로그램에서는 회원국간 전자정보 교환과 호환성 확보를 위한 수단으로 공개 S/W 사용을 위한 선행연구, 재사용·공유를 위한 타당성 검토 작업을 하고 있다.

■독일

- 내부부의 KBSt (Koordinierungs und Beratungsstelle für Informationstechnik)을 설치하여 공공부분에서 오픈소스 S/W를 구현하고 활용하는 활동을 지원해주는 역할 및 공공부분의 오픈소스 S/W 사용 경험 및 이전과 자원센터로서의 역할 수행한다.
- 민간부분의 오픈소스 S/W 활용 촉진 및 중개 기구 설립 필요성 증대로 경제기술부에서 민간오픈소스 S/W 개발 및 활용 지원 실시하고 있으며, 오픈소스 S/W의 개발자와 사용자, 서비스 제공자 및 제조업체를 연계시키는 역할을 수행하는 BerliOS 프로젝트를 추진하고 있으며 이를 통해 중소기업의 오픈소스 S/W 활용 촉진 지원한다.
- 독일 재무부는 저비용 고효율을 가져올 수 있는 Linux를 인터넷 환경에 사용할 수 있도록 15,000사용자에게 보급을 추진하고 있다.

■프랑스

- 프랑스 사회당 의원 3명은 공개 S/W를 공공업무현장(관청)에서 사용하도록 하는 법안을 제출하였으며 ‘98년 수상 산하에 전자정부 구현시 필요한 부처간 업무조정 및 기술지원업무를 수행하기 위해 ATICA(Agency for Information and Communication Technologies in Administration) 설립하였다.

- ATICA는 공공부문에서 오픈소스 S/W 활용촉진과 경험공유를 위한 정보수집•분류 및 홍보컨설팅, 전문가 네트워킹 센터, 공공부문 오픈소스 활용경험 이전 등 활동 수행하고 있다.
- 350여 개의 초·중등학교에서는 SLIS라는 리눅스 서버를 사용 중이다. PAGSI(Preparing Frances's entry into the information society) 자료(2002)에 따르면 전자정부 구축 및 인터넷확대를 위해 공개 S/W 활용에 대한 내용이 포함되었다.

■영국

- 전자정부 등 향후 IT개발에 있어서 공개 표준을 지원하는 호환성을 지닌 제품만을 사용하고 정부의 SI사업의 경우 소스코드에 대한 완전한 권리 확보 의지를 표명하였다.
- 경제성을 근간으로(best value for money) 한 S/W 조달기준 마련·시행(금액대비 최고의 가치를 가진 솔루션을 구매, 개개의 IT 공급업체에 대한 의존에서 탈피, 시스템 개발, 향상 및 통합에 있어서 보다 많은 융통성을 제공하는 공개 S/W 사용을 적극 장려 등)하고 있다.
- 영국의 조달청(OGC)과 지방의 IT 관련 공무원 단체인 정보기술관리협회(SOCITM) 등이 MS와의 라이선스 갱신에 어려움을 겪자 Windows를 리눅스로 대체를 검토하고 있다.

■남미

- 공개소프트웨어의 대명사인 리눅스를 강력히 지원하고 있는 IBM이 남미 최대 시장인 브라질에 두개의 리눅스 센터를 설치한다고 C넷이 보도했다. 이들 두 리눅스 관련 시설은 브라질 상 파울루주의 주도(州都)인 상 파울루에 개설된다.
- 이번 개설은 지난 10월 IBM과 브라질 정부간에 맺은 브라질 공공분야에서의 리눅스 기술 확산을 보다 진전시키기 위해 계획됐다.
- 두 시설 중 하나는 리눅스 통합 센터(LIC)인데 이는 IBM이 자사의 소프트웨어(DB2, 티볼리 등)와 리눅스 간에 연계하는 실험을 하는 곳이다. 브라질 이외에도 이미 IBM은 중국,싱가포르, 일본, 독일 같

은 나라에서 LIC 시설을 설치, 운영하고 있다(기사발췌 : 전자신문 <방은주기자 ejbang@etnews.co.kr>).

- LIC 외에 또다른 시설은 브라질 프로그래머들을 위한 리눅스기술 센터인데 IBM은 이곳에서 브라질 프로그래머들을 훈련, 리눅스 핵심인 커널을 개발토록 할 예정이다.
- S/W 구매비용의 절약, 국가의 안전보장, 기술종속성 극복 및 S/W 산업육성을 위해 공개 S/W 활성화 정책 적극 추진하고 있다. 공공기관에서 공개 S/W의 구입을 강제하는 법안이 브라질의 자치단체에서 통과된 이후 여러 국가에서 법안이 제출되고 있으며, 행정부, 입법부, 사법부 각 기관 및 자치단체, 정부투자기관 등의 시스템 및 컴퓨팅 장비에 공개 S/W를 사용해야 한다.
- 이는 특정 목적을 수행하는 공개 S/W가 없는 경우 예외적으로 독점 S/W의 사용을 인정하나, 이 경우에도 한시적이다. 현재 페루, 아르헨티나, 콜롬비아에서 비슷한 법안이 상정되어 있으며, 베네주엘라에서 기획개발 장관이 관련 정책 발표하였다.

2) 국내 동향

- 우리나라 공개SW 확산의 장애요인은 수요 측면에 있어서는 시장 진입 장벽, 성공 사례 부족, 호환성 결여 등을 꼽고 있으며, 공급 측면에 있어서는 기술 지원 체계의 미흡, 전문 인력의 부족 등을 들 수 있으나, 이러한 제약 요인은 일부는 사실이나 대부분은 수요자의 잘못된 인식과 편견에서 비롯된 것이라 할 수 있다. 이러한 제약 요인을 해소하기 위해 다음과 같은 주요 정책을 추진하고 있다.
- 첫째, 공개 SW 확산의 가장 큰 현실적 장애 요인은 특정기술과 제품을 명시하는 시장 진입장벽이라 할 수 있다. 공개 SW 시장진입 장벽은 특히 공공기관에서 정보시스템 구축을 위한 사업 발주시 제안요청서에 명시되게 되는데, 고의 또는 과실로 인하여 유닉스 등 특정 운영체제를 명시함으로써 리눅스로 대표되는 공개 SW 진입을 가로막고 있으며, 나아가 국산 하드웨어에 대해서도 진입장벽이 되고 있다. 이러한 진입장벽을 해소하기 위해 지난 2004년 12월 정부 혁신지방분권 위원회는 ‘전자정부사업 공개소프트웨어 도입 권고안’을 통과시킨 바 있고, 정보통신부는 다양한 기술 가이드를 통해 불공정 경쟁환경을 개선하고자 노력하고 있다.
- 둘째, 공개 SW가 성공 사례가 부족하고, 리눅스기반의 우수한 상용 소프트웨어가 부족하다는 인식이다. 이러한 인식은 대표적인 공개 SW에 대한 편견과 오해라 할 수 있다. 그 이유는 웹 서버, 파일 서버 등에서 이미 기술적 안정성을 입증 받았으며, 중대형 서버 시장에서도 도입이 되고있고 IHV, ISV들도 주요 응용 솔루션을 개발 제공하고 있기 때문이다. 이러한 편견과 오해를 해소하기 위해 다양한 공개 소프트웨어 도입 성공사례를 발굴 및 전파하고 있으며, 공공부문을 중심으로 한 ‘공개소프트웨어기반 정보시스템 구축 시범사업’을 통하여 기술적 안정성을 입증하고 있다. 또한, 다양한 공개소프트웨어기반 솔루션을 알리기 위해 ‘공개기반 기업과 제품정보 가이드’를 발간하고, DB화하여 정보를 제공하고 있다.

- 셋째, 공개SW는 전문기업 및 전문인력이 부족하여 제대로 된 기술 지원을 받을 수 없다는 인식이다. 이는 공개 SW 수요 부족에 따른 인력공급 및 진입 기업이 부족하기 때문이다. 따라서 정부는 리눅스를 중심으로 전문인력을 양성하고, 한국소프트웨어진흥원 내에 ‘공개 소프트웨어 기술지원센터’를 설치하여 리눅스를 도입한 공공기관을 중심으로 온-오프라인 기술지원을 수행하는 동시에, 리눅스기반의 애플리케이션에 대한 안정성 테스트를 수행하고 있다.
- 넷째, 공개 SW는 다양한 기업 및 개발자 커뮤니티를 통하여 개발되고 있기 때문에 호환성이 부족하다는 인식이 있다. 특히 다양한 배포판의 난립으로 인한 호환성 결여를 해소하기 위해 ETRI와 민간기업 컨소시엄을 통해 표준 리눅스 스펙인 ‘부요(Booyo)’를 개발하고 있다.
- 국내 표준 리눅스 규격, 부요
 - 부요는 표준 규격서인 동시에 규격 구현물인 표준판이다. 사용상의 호환성 및 안정성을 확보하고 국외 배포판에 대등한 수준의 운영체제를 개발하는 것을 목표로 하는 부요를 개발함으로써, 다양한 분야에서 널리 공개 소프트웨어가 활용될 수 있는 신뢰감을 마련하고자 한다. 국내뿐만 아니라, 국외 여러 나라에서도 공개 SW를 활용하여 자국의 운영체제를 확보함으로써 자국의 소프트웨어 산업을 육성하고 주요 시스템의 보안과 관련 기반 기술을 확보하고자 추진중에 있다. 그 예로 중국의 RPLinux, 필리핀의 Bayanihan, 태국의 TLE Linux, 말레이시아의 Mimos Linux, 인도의 CDAC Linux, 스페인의 bnuLixEX 등이다.
 - 부요라는 단어는 날아오르는 펭귄을 상징한다. 우리나라 민요 가운데 까투리 사냥에서 중간에 까투리를 풀 샛에서 나와 날아오르도록 놀라게 하는 소리가 있다. 남쪽 지방에서는 ‘휘이여’ 북쪽 지방에서는 ‘우여’ 그런데 중부 지방에서는 ‘부요’라고 외친다. 즉 부요라는 이름은 리눅스의 희망과 도전 정신을 나타내는 펭귄을 날아오르게 하자는 의성어이다. 부요를 통해 펭귄을 날아오르게 해 리눅스 산업

도 크게 일어날 것이라는 희망을 담고 있다. 그래서 부요의 로고도 썩권이 날아오르는 모습이다. 또한 부요란 말은 한자로 풍요롭다는 의미인 ‘富饒’와 발음이 같아 소프트웨어 산업을 부유하고 넉넉하게 하겠다는 의도도 담겨 있다.

- 부요 플랫폼은 부요 데스크톱과 부요 서버로 구성된다. 부요 기술개발은 예전의 국내 배포판 제품처럼 국외 제품을 기반으로 한글화, GUI 변경만의 수준이 아니라, 명실 공히 국내 표준 운영체제로 대표할 수 있게 안정성, 호환성 결여 문제 극복뿐만 아니라 자체적으로 커널 및 미들웨어 부분의 핵심기술 개발에도 초점을 두고 있다. 이렇게 개발되는 핵심 기술들은 유명 공개 프로젝트 사이트를 통해서 공개된다. 부요가 국내 표준 운영체제라고 국제표준을 부합하지 않는 것은 아니다. 2005년 12월 부요 데스크톱과 서버 규격 1.0은 한국정보통신기술협회의 표준 승인을 거쳤고, 현재 규격 1.0의 시험 절차서를 마련하고 있다. 이와 병행하여 규격 2.0도 제정중에 있다. 부요 표준판을 기반으로 하는 상용 배포판도 출시되고 있다. 계속해서 출시될 국내 상용 배포판들은 사용자 환경뿐만 아니라, 응용 소프트웨어 수행 환경에서도 완전한 호환성, 안정성을 보장할 수 있도록 부요 규격을 따르기를 희망한다. 그러면 그동안 국내 사용자가 느끼고 있는 공개 소프트웨어에 대한 불안감을 해소할 수 있는 계기가 마련되고, 국내 공개 소프트웨어 시장을 보다 활성화할 수 있기 때문이다.
- 부요 데스크톱은 사무, 제조, 교육 환경과 같은 제한된 응용만을 사용하는 환경에서의 편의성, 기능개선을 우선적으로 목표로 한다. 그 다음 게임 등과 같은 일반 사용자가 편리하게 사용할 수 있는 데스크톱 환경까지 꾸준히 개선하고 발전시킬 계획이다. 현재의 부요 데스크톱은 익숙하게 사용되고 있는 데스크톱 환경보다는 부족한 점이 많다. 그러나 리눅스뿐만 아니라, 데스크톱을 구성하는 주요 공개 SW 기술이 지속적으로 개선되고 발전되고 있기 때문에 훌륭한 데스크톱 환경으로 빠른 시간 내에 부상될 것으로 생각된다. 부요 데스크톱은 다음 사항들을 고려하여 개발되고 있다.

- . LSB 등 산업 표준 규격을 기반으로 하는 개방형 구조 제공
- . 기존 대중적인 데스크톱 운영체제 대비 부족 기능 해결
- . 국내 환경을 위해서 표준 한글 사용 환경과 한글처리 기능을 제공
- . 설치가 용이하고, 원격 업데이트 지원체제 제공
- . 데스크톱 환경 개선과 주변 장치 지원
- . GNOME 기반의 사용자 친화적인 데스크톱 환경 제공
- . 서버중심의 리눅스를 데스크톱용으로 가볍고 빠른 환경 제공
- . 오피스, 인터넷 बैं킹 등 사무업무용 환경 제공 (그림 3)에서 볼 수 있듯이, 사용자에게 친화적인 환경을 제공하기 위해서 복잡하고 정리가 안된 메뉴구조를 개선하고, 아이콘 이름도 사용자가 어떤 응용인지를 쉽게 할 수 있도록 하였으며, 테마 및 용어통일 등 전체적으로 통일성을 제공하였다.
- 부요 서버는 국제 산업 표준을 만족하고 고성능, 고가용성, 고신뢰성을 보장하면서 엔트리급, 엔터프라이즈급 서버 환경을 안정적으로 지원하는 것을 목표로 다음 사항을 고려하여 개발되고 있다.
 - . LSB, FHS 등 국제 산업 표준 만족
 - . 리눅스 기능 강화를 위한 CGL, DCL 규격 만족
 - . 풍부한 서버 개발 도구 제공
 - . 고부하 지속환경 및 서버 성능에서 국외 제품과 동일 기능 제공
 - . 공개SW기반의 보안, 시스템관리, 서버 가상화, 웹 등 솔루션 제공
- 부요 플랫폼은 이미 존재하는 공개 소프트웨어들 중에 소스가 공개되어 있고 많은 개발자가 참여하여 진행중인 소프트웨어를 선정하여 개발한다. 그러나 부요 규격에 적합한 기능의 소프트웨어가 없는 경우는 직접 개발하여 국제 커뮤니티에 공개한다.

다. 기술 및 커뮤니티 동향

1) 기술 로드맵

- 아래 그림은 2005년 가트너에서 보고된 공개SW 기술 hyper cycle 자료로 두 가지 측면에서 공개 SW 기술 발전 방향을 분석할 수 있다.
- 첫번째로 시스템 플랫폼 관점이다. 서버 플랫폼의 경우는 지난 몇 년 동안은 1-4 프로세서 환경인 엔트리급(소형) 서버 환경에서 발전해 왔지만, 점차적으로 성능, 기능 개선, 보안 기능 강화로 64비트 프로세서, 8 프로세서 이상의 중요업무를 처리하는 엔터프라이즈급(중대형) 서버로 확대되고 있다. 데스크톱 플랫폼의 경우는 제한적인 업무 사용자 환경에서 다양한 계층의 일반사용자가 편히 사용할 수 있는 환경으로 기술이 발전할 것이다.
- 두번째 소프트웨어 계층적인 관점으로 리눅스 커널과 같은 최하위 계층의 기술 발전에서 미들웨어의 공개 SW화를 거쳐 응용 소프트웨어까지 공개 SW 기술로 발전할 것이라고 분석하고 있다. 또한 공개 SW 기술은 개발자와 개발자들 간에 소스를 무료로 공개하며 서로 간의 요구 사항을 반영하면서 기술이 성숙해진다. 즉, 기술 개발이 시작(technology trigger)하여 기술이 시장에 적용되기 전(trough of disillumination)까지는 개발자 커뮤니티를 통해서 기술이 개발된다는 것이다. 이 개발 과정 동안 기업은 기술개발에 필요한 비용을 투입하는 것은 아니지만, 여러 공개 SW를 가지고 제품화를 하는 단계인 통합, 시험 및 안정화 과정에 소요되는 개발 비용을 투입하여 상용화(plateau of productivity)를 한다. 따라서 공개 SW기반 제품은 초기 개발 때부터 개발비가 투입된 상용 소프트웨어보다 다소 저렴한 비용으로 구입할 수 있다는 것이지 무료가 아닌 것이다.

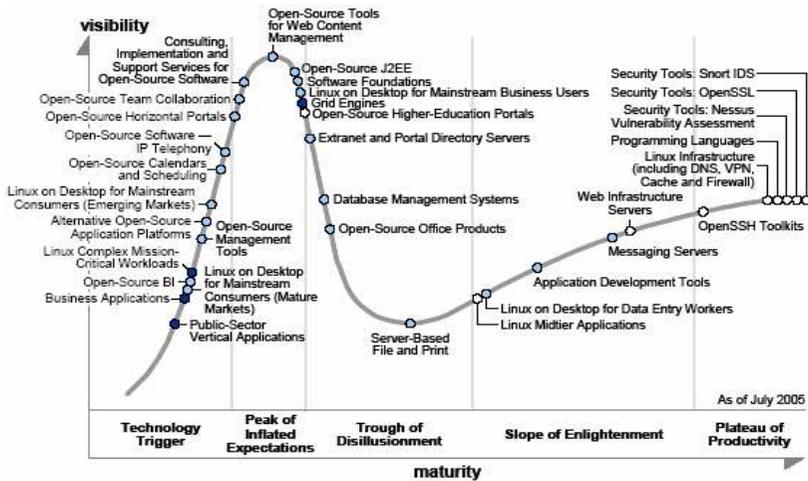


그림 10. 공개SW 기술 hyper cycle

2) 커뮤니티 동향

- 앞서 언급한 공개SW 기술들은 개발자 커뮤니티를 통해서 발전하고 성숙되고 있다. 물론 최근에는 국외 대형 기업의 공개SW에 대한 관심으로 기술 개발의 투자를 점차 늘리고는 있지만 공개SW의 기술 성장의 중심축은 자발적으로 구성된 커뮤니티이다. 이러한 커뮤니티는 북미와 북유럽에 많이 발달되어 있다. 한국을 포함한 아시아 지역은 부족한 부분이 많다. 개발자가 자발적으로 커뮤니티에 참여하여 많은 시간을 기여할 수 있는 것은 공개 소프트웨어가 가진 철학을 공유하고 타인에 대한 기여를 통한 자기 만족, 즉 소스 코드의 개발에 참가함으로써 얻어지는 명성과 기술적인 가치의 공유를 통해 자기 발전 등을 얻을 수 있기 때문이다.
- IBM에서는 공개SW를 소스 코드가 공개되어 있는 소프트웨어라고 정의하는 것보다 공개 협력(public collaboration)이라고 정의한다. 즉 상용소프트웨어의 개발 방식인 팀 체제인 계층적인 형태의 개발 모델이 아니라 네트워크 형태의 커뮤니티 방식으로 기술이 성숙해지는 것이다. 이 과정에 학력, 성별, 소속에 관계없이 개발 능력을 가지고 있는 어느 누구나 개발에 참여하여 동료 검토를 거치면서 견고하고 안전한 소프트웨어로 발전한다는 것이다.

- 국내의 공개SW 커뮤니티 현주소는 프로그램을 만들고, 개량하고, 매뉴얼 등의 문서를 쓰고 번역하는 활동과 오프라인 모임과 세미나 등을 운영하는 것이다. 즉 국내의 커뮤니티 특성은 아직까지 질의응답을 중심으로 운영되고 있어, 활동이 국내에서만 머무르고 국제적인 활동에 동참하거나 기여하는 부분이 매우 미진하다는 것이다. 또한 국내의 공개 소프트웨어 생산율이 이미 존재하거나 개발중인 소프트웨어에 대비 1%도 되지 못하는 점이다. 따라서 공개 소프트웨어를 잘 활용하는 정책도 중요하지만, 보다 다양하고 성숙한 기술의 발전을 위해 기술 개발에 기여할 수 있는 국내 개발자 커뮤니티의 육성도 시급하다.
- 국외의 경우는 대표적인 커뮤니티로 현재 130만명의 개발자와 11만 개 이상의 개발 프로젝트를 운용중인 sourceforge.net을 들 수 있다. 그뿐만 아니라, 아파치, 파이썬, KDE 등 수많은 주요 커뮤니티가 운영되고 있다. 여기서는 국외의 다양한 커뮤니티에서 리눅스에 관련 되는 커뮤니티의 동향을 살펴본다.
- 리눅스 개발 커뮤니티는 북미와 북유럽을 중심으로 300여 개의 커뮤니티가 존재한다. 그 중에서도 100여 개가 활발히 진행중에 있다. <표 3>은 현재 활발히 운영중인 리눅스 개발 커뮤니티를 정리한 것이다. 국내에 이미 잘 알려져 있는 젠투(gentoo), 데비안(debian), 페도라(fedora) 커뮤니티에는 수천명의 참여자가 있는 반면에, 뎀피스(mepis), 뎀스몰(damn small) 같은 커뮤니티는 운영자 혼자만 존재하는 경우도 있다. 이런 경우에도 커뮤니티 인기도가 상위에 존재하는 이유는 다른 커뮤니티에 비해차별적인 요인이 있기 때문이다. 예를 들면, 세상에서 가장 사용하기 쉬운 데스크톱 리눅스를 만든다든지, 개발 주기를 짧게 하여 수시로 새로운 버전을 제공한다든지, 특정 플랫폼이나 특정 응용을 잘 지원한다든지 등이다. 또한 규모가 작은 커뮤니티의 경우는 독자적인 운영 방식을 택하고 있지만, 대다수의 커뮤니티에서는 투표 등의 의사 결정 조직과 체계를 갖추어 운영을 하고 있다. 개발자를 위한 버그관리 및 소스 코드 관리 시스템,

개발자간의 의사 소통 방식도 구축되어 있다. 개발뿐만 아니라 사용자를 위한 공간도 아주 중요하다. 다양한 시험과 사용에 따른 버그 보고 및 요구사항을 제공하는 중요한 주체이기 때문이다. 따라서 소프트웨어의 다운로드 및 업데이트 사이트, 질의응답 및 공지 등에 필요한 공간을 마련하고 있다. 이처럼 공개SW 기술 발전에 있어 커뮤니티의 중요성은 아무리 강조해도 지나치지 않다. 커뮤니티를 통한 많은 참여자로 안정적인 소프트웨어를 생산할 수 있으며, 많은 기능들은 분담하여 개발함으로써 단시간에 다양하고 성숙한 기능을 생산할 수 있다는 것이다. 더불어 커뮤니티 활동은 훌륭한 인력 양성의 장이 될 수 있으며, 기술 확보로 창업의 기회를 마련할 수도 있다. 끝으로 향후의 공개 SW의 비즈니스 모델에 있어 커뮤니티가 없는 제품은 사라질 것이고 존재하더라도 그 제품의 인지도는 떨어질 것이다.

<표 3> 리눅스 개발 커뮤니티 현황

	ubuntu	mandriva	opensuse	mepis	damn small	debian	slackware	gentoo	freebsd	vector	xandros	puppy	arch	Yellow dog
소스 공개	○	×	○	×	×	○	○	○	○	×	×	×	○	○
기반 시스템	Debian	from the scratch	RPM	Debian	Debian	from the scratch	from the scratch	from the scratch	BSD Unix	Slackware	Corel	from the scratch	from the scratch	Fedora
의사 결정	기술 위원회	NA	노벨기업	1인	2인	투표	NA	관리자 협의	관리자 결정	관리자	회사내 관련조직	재단	Trusted User 결정	내부적
개발 커뮤니티	○	○	○	×	×	○	○	○	○	×	×	×	○	×
펀드 수단	회사	회사	회사	기부	기부	기부	회사	기부	기부	기부	판매	×	×	회사
규모	NA	NA	NA	1명	2명	1500명 이상	4명	330여명	350여명	7~8명	NA	19명	18명	NA
데스크톱	○	○	○	○	○	○	구분없음	○	○	○	○	○	○	○
서버	○	○	○	×	×	○		○	○	×	×	×	○	○
기본 데스크톱 환경	Gnome	Any	KDE	KDE	Flux Box	Gnome	Any	Any	Gnome	KDE, gnome	KDE	Gnome	Enlightenment	Any
특성	사용편이	유럽형 리눅스	세계2위 리눅스	쉬운 데스크톱	Lived	개발 시스템 체계화	Simple/stable	Flexibility가 높음	진행 프로젝트가 다수	SOHO	MS APP 호환	Lived	Lightweight and simple	ppc 전용
국가	Isle of Man	프랑스	독일	미국	미국	×	미국	미국	미국	캐나다	캐나다	호주	캐나다	미국

※ 출처 : 전자통신동향분석, 2006

3) 한중일 공개SW 활성화 포럼 동향

- 2004년부터 한국, 중국, 일본 동북아시아 3국은 공개SW 기술, 특히 리눅스 기술의 이용 활성화를 통해 사유 독점 소프트웨어 솔루션에서 자립할 수 있도록 기술을 공동 개발하고 인력 및 정보를 교환하는 목적으로 동북아 한·중·일 OSS 활성화 포럼을 결성하였다. 포럼의 공식적인 명칭은 NEA CJK OSS Promotion Forum이다. 포럼의 첫 회의는 2004년 3월 북경 회의를 시작으로 같은 해 7월에는 일본 회의, 12월은 서울 회의로 진행되었다. 그리고 2006년 4월에는 4차 회의를 중국 천진에서 개최하였다. 포럼 회의에서는 3국 IT 국장급 회의도 함께 개최된다.
- 2003년 3월 1차 중국 북경 회의에서는 다음과 같은 10가지의 실천 항목이 합의되어, 각국은 내부적으로 활동을 이행하고 포럼에서 그 결과를 보고하고 있다.
 - . 이용의 우호적인 활성화에 노력
 - . 리눅스 문자셋 표준화에 노력
 - . 리눅스기반 데스크톱, 오피스 개발에 노력
 - . 임베디드 협력에 노력
 - . 전자정부에서 리눅스 활성화에 노력
 - . 중소기업 지원에 노력
 - . 하드웨어 및 소프트웨어 벤더 지원에 노력
 - . 동북아 아시아 세계 커뮤니티에 기여
 - . 인력 양성에 노력
 - . 정책연구 및 정보 교환
- 2003년 7월 2차 일본 회의에서는 기술개발 및 평가 워킹그룹, 인력양성 워킹그룹, 표준화 워킹그룹 등 3개의 워킹그룹 결정을 합의하여, 그 해 12월 서울 회의에서 워킹그룹 첫 회의를 가졌다. 그 이후, 워킹그룹별로 회의를 정기적으로 가져, 기술개발 및 평가 워킹그룹에서는 3국 개발자가 참여하는 커뮤니티를 구성하여 공통 기반기술 개발을 준비중에 있다. 인력양성 워킹그룹에서도 한·중·일 3국간

통용되는 공개SW 자격증 제도를 마련하고 있다. 표준화 워킹그룹에서는 한·중·일 문자에 대한 입력 방식의 표준화로 각국의 요구사항을 수렴하여 기능 명세서를 완성하였으며, 현재 구현을 진행하고 있다. 이번의 입력 방식 표준화가 완료되면 국제 표준화 단체에 안건으로 상정할 계획을 가지고 있다. 5차 포럼 회의와 워킹그룹 회의는 2006년 11월 일본에서 개최되었다.

라. 국내의 기업 및 표준화 동향

1) 국외 기업 동향

- 세계적으로 유통되고 있는 배포판 제품의 양대산맥적인 레드햇과 수세 리눅스는 리눅스 커널 2.6을 기반으로 데스크톱, 엔트리급, 엔터프라이즈급 플랫폼을 지원한다. 특히 최근 하드웨어 벤더와 고객이 많은 관심을 가지고 있는 운영체제 가상화에 대한 기능들도 추가되어 개발되고 있다. 특히 수세 리눅스는 노벨에 인수되면서 기존 유럽 리눅스 시장뿐만 아니라 북미 시장에도 진출하고 있다. 레드햇이 2003년 11월부터 운영한 페도라 프로젝트와 유사하게 노벨도 오픈 수세 프로젝트를 구성하여 많은 개발자와 사용자가 참여하길 희망하고 있다. 페도라 프로젝트는 현재 페도라 코어 5까지의 결과물을 개발하고 있는데 비해 오픈 수세는 개발 초기 단계라고 할 수 있다. 그러나 최근 레드햇은 페도라 프로젝트에 개발비를 투입하지 않을 것이라고 선언하였다.
- 현재 페도라 프로젝트는 운영되고 있지만 점차 개발 속도와 결과물 산출에 있어 차질도 예상된다. 임베디드 리눅스 분야의 대표적인 기업인 몬타비스타는 리눅스 커널을 응용한 다양한 패키지 기능을 보유하고 있어, 리눅스를 통신 시스템에 적용하거나 고 가용성 기능을 요하는 응용 분야에도 적용할 수 있도록 기술을 보유하고 있다. 썬 마이크로시스템즈는 솔라리스 제품의 기술적인 우수성에도 시장 경쟁력에서 뒤지는 현상 때문에 오픈 솔라리스 정책을 폈다. 이 정책으로 썬 운영체제 소스 코드를 공개함으로써 시장에서 활용도를 높

이고자 하고 있다. 라이선스 정책도 특히 사용권까지 허용하는 CDDL 라이선스를 제시하고 있다.

- IBM은 LTC 센터를 통해 리눅스를 포함한 공개소프트웨어가 빠르게 성숙하도록 지원하고 있다. 매년 300여 명의 개발자가 내부 회사의 업무를 수행하지 않고 공개 소프트웨어 프로젝트에 참여하도록 하고 있다. 이들이 참여하는 프로젝트는 서버 및 데스크톱 리눅스 분야의 다양한 프로젝트에 참여하고 있으며, IBM의 제품이 공개 소프트웨어와 잘 동작 되도록 요구사항을 전달하거나 개발을 선도하는 역할을 한다. HP도 자사 플랫폼이 리눅스를 지원하도록 기술 지원 센터를 운영하고 있다.
- 일본은 대기업 중심으로 공개 SW에 대한 활동을 전개하고 있다. 히타치, NTT, 후지쓰 등이 그 예다. NTT 경우는 공개 SW 기술을 서버의 솔루션 스택으로 구축하는 작업을 진행하고 있으며, OSDL의 DCL 활동을 적극적으로 하고 있다. 또한 리눅스 커널의 견고성을 위해 특히 파일 시스템 분야의 다양한 시험을 통해 발견되는 버그와 패치 파일을 리눅스 커널의 메인라인(mainline)에 입력하고 있다. 히타치도 서버 분야의 관련 공개 SW 기술 발전을 위해서 100여 명 규모의 조직을 운영하고 있다. 특히 서버의 성능 벤치마킹에 초점을 두고, 도구 개발과 벤치마킹 데이터를 분석하여 공개 SW의 문제점을 제시하고 있다.
- 중국은 홍기(Red Flag), 공창(Co-Creat), 선화(Xinhwa)의 배포판 중심으로 진행되고 있다. 홍기는 일본의 미라클 리눅스와 한국의 한글과컴퓨터 3개 기업이 레드햇 제품을 기반으로 아시아눅스 제품을 개발하고 있다. 공창은 윈도 기능에 필적할 수 있는 데스크 톱 리눅스 개발과 오픈 오피스를 기반으로 한 자국의 오피스 제품을 지원하고 있다.

2) 국내 기업 동향

- 국내의 배포판 기업으로는 아이젯 리눅스, 한글과컴퓨터, 리눅스원, 슈퍼유저코리아가 있다. RPM기반의 제품들을 제공하고 있다. 한글과컴퓨터는 아시아눅스 프로젝트에 참여하고 있으며, 나머지 기업들은 국내 부요 프로젝트에 참여하고 있고, 향후 참여할 예정이다. 그렇지만 부요 규격을 만족하는 제품을 개발함으로써 사용상의 호환성 문제는 해소되리라 예상한다. 삼성전자는 자사 브랜드의 리눅스 서버와 워크스테이션을 생산하고 있다. 포스데이타는 리눅스기반의 다양한 응용 솔루션을 개발하고 있으며, 엘지엔시스도 서버 및 통신 분야에 리눅스 시스템을 적극 판매하고 있다. SI 업체들도 공개 소프트웨어기반의 응용 시스템을 구축하고 있으며, SKC&C 및 국내 포털 업체들도 자사 내부 시스템용으로 리눅스 배포판을 개발하고 있다. 큐브리드 등 일부 기업들은 데이터베이스 등의 미들웨어 응용 기술을 공개하는 정책으로 사업을 진행하고 있다. 국외 기업과 달리, 국내 기업의 커뮤니티 참여도는 전무하다. 공개 SW와 관련되는 사업은 진행하고 있지만 그에 따른 수익성이 낮고 기업의 기부 문화가 정착되지 않았기 때문이라고 분석된다.

3) 표준화 동향

- LSB 규격
 - 1998년 5월경 LSB 프로젝트는 리눅스 토발즈, 상용 배포판 업체, 리눅스 인터내셔널, 그리고 Free-BSD의 지원 아래 세상에 나오게 되었다. 현재 LSB 프로젝트는 공개 소프트웨어 기술의 활성화를 지원하는 국제 비영리 컨소시엄인 FSG의 워킹그룹으로 진행되고 있으며, 2001년 6월 최초 규격의 발표를 시작으로 현재 3.0까지 발표하였다. LSB의 목적은 리눅스 배포판 간에 바이너리 응용의 호환성을 증대하고, 하드웨어와 소프트웨어의 이질성으로 발생하는 소프트웨어 개발과 이식의 복잡성을 감소시키기 위해서이다.
 - LSB에서는 운영체제의 시스템 호출 인터페이스, 시스템 라이브러리,

패키지 포맷 등을 규정하고 있고, 대부분의 리눅스 배포판들의 기준으로 사용되고 있다. 하지만 바이너리 응용의 호환성을 유지한다면, 리눅스 커널이 사용되지 않아도 상관없다. LSB 인증은 FSG의 인증 프로그램을 거쳐야 하며, 제품이 적용되는 인증 시험을 통과했을 때 제품과 관련되는 LSB 마크를 사용하도록 라이선스를 얻게 된다.

■ FSB 규격

▪FHS는 소프트웨어로 하여금 설치된 프로그램과 디렉토리의 위치를 예견할 수 있게 하고, 사용자로 하여금 설치된 프로그램과 디렉토리의 위치를 예견할 수 있게 한다. 1993년 8월부터 시작되어 대부분의 리눅스 배포판은 이 규격을 따르고 있으며, 현재 최신 규격 버전은 2.3이다.

■ CGL, DCL 규격

▪CGL, DCL 규격은 OSDL의 산업 요구기능 규격이다. OSDL 기관은 다양한 IT 시스템에 리눅스 기술을 적용시키는 역할을 하는 비영리 기관이다. 2000년에 설립되어 IBM, 인텔, HP, 노키아, ETRI 등 75개 회원사로 구성되어 있다. 리눅스 커널 개발자와 메인테이너인 리누스 토발즈와 앤드류 몰튼 등 45명의 직원으로 구성되어 있다.

▪CGL, DCL 규격뿐만 아니라 OSDL은 DTL, MLI 요구기능 규격도 개발하고 있다. 요구기능 규격 제정은 고객과 벤더의 요구사항과 use case로부터 시작된다. 제안된 요구사항과 use case는 관련되는 내부 워킹그룹에 전달되어, 이미 존재하는 규격과의 중복성, 필요성을 검토한 후 규격으로 추가한다. 다른 규격서와 달리, OSDL은 규격의 구현물을 공개 프로젝트로 진행하기 때문에 PoC에 대한 웹 사이트 정보도 함께 제공한다. 따라서 시간이 지나면 OSDL 규격 기능에 대한 구현물은 공개 프로젝트로 완성되며, 벤더들은 그 결과를 제품화로 활용할 수가 있다. 하지만 OSDL은 표준화의 주체는 아니다. 산업체의 요구사항에 따른 문제(gap) 분석, 규격화 및 구현을 한다. 즉 OSDL은 기존의 표준들, LSB, POSIX 등을 준수하고 있다.

▪CGL 규격은 리눅스를 통신 장비와 같은 실시간성, 고가용성, 고신뢰

- 성을 요구하는 시스템에 적용하기 위한 규격이다. 2002년부터 규격 개발을 시작하여 현재 규격 3.2 버전을 완료하였다. 규격 내용으로는 가용성, 서비스성, 표준, 성능, 하드웨어 지원, 보안, 클러스터링 등 7개 분야이며, 140여 개의 요구 기능 규격으로 구성되어 있다. 2005년부터는 CGL 등록(registration)으로 유통되고 있는 리눅스 배포판의 CGL 규격 만족 정도를 시험하고 있다. 이것은 통신 사업자가 리눅스 제품을 선택하는 기준으로 활용되도록 하고 있다. DCL 규격은 리눅스를 데이터베이스 시스템과 같은 중요업무 시스템에 활용하기 위한 규격이다. 2002년부터 시작하여 현재 규격 2.1 버전을 완료하였다. 엄밀히 말하면 DCL 경우는 규격이 아니라 capability로 정의하고 있다.
- DTL은 데스크톱 리눅스 분야이다. 전 세계적으로 3% 정도의 점유율인 리눅스를 더욱더 활성화하고자 한다. 2004년부터 시작되었으나 활동이 미진 하였고, 2005년부터 활발한 활동을 하고 있다. MLI은 모바일 시스템에 대한 규격이다. 2005년부터 시작되어 규격 작업이 진행중에 있다.
 - 기타 규격들
 - 공개 SW 기술 관련 기타 규격들로는 시스템 자원 관리의 규격인 DMTF의 CIM과 WBEM[9] 표준, 서비스 가용성에 대한 규격인 AIS, 문서 포맷의 표준인 ODF 등이 있다.

3. 미 국방성의 공개SW관련 정책연구 사례

가. 연구 배경

- 미 국방성은 공개SW와 관련하여 2개의 중요한 정책적 연구를 수행하였는데, 하나는 2002년 Mitre를 통해 수행한 미 국방성의 공개SW 활용에 관한 연구이고 또 하나는 미 국방성의 지원하에 2006년 6월에 수행한 개방형기술개발의 적용에 관한 정책연구이다.
- 이 중에서 2002년에 수행된 Mitre의 공개SW활용에 관한 연구는 미국 국방시스템에서 공개SW와 관련된 최초의 정책보고서로서 향후 미 국방분야에서의 공개SW 활용정책에 관한 논의의 시작점으로서의 의미를 가진다.
- Mitre의 공개SW활용에 관한 연구 보고서의 목표를 요약하면 다음의 세가지로 압축된다.
 - 국방분야에서 사용되고 있는 공개SW는 무엇이고, 공개SW를 활용하여 사용되고 있는 응용사례에 대한 조사분석
 - 공개SW 활용 시 보안과 신뢰성 측면에서의 문제점
 - 상용SW에 대한 공개SW의 라이선스 문제점
- 반면 최근 2006년 6월에 수행한 개방형기술개발의 적용에 관한 정책연구는 미 국방성 산하 기관인 AS&C에서 수행하는 첨단시스템 및 개념과 관련된 프로젝트에 대해 공개SW를 포함한 개방형기술개발의 활용을 증대시키기 위하여 수행한 좀 더 발전되고 구체화된 정책연구로서, 공개SW의 개발모델, 개방형 표준 및 인터페이스의 수용, 기술적 유연성 및 협업적 개발방법을 결합한 개방형기술개발을 적용하여 NCW에 대비한 무기체계를 혁신적이면서도 저비용으로 획득하기 위한 정책이다.

나. 공개SW의 활용에 관한 연구

1) 연구결과 요약

- Mitre의 연구를 실제로 수행한 기관은 미국 DISA로서 2002년 10월에 연구가 완료되었고, 몇 가지 주목할 만한 특징은 다음과 같다.
- 연구는 기존의 공개SW활용에 대한 서베이를 중심으로 함
- 연구는 공개SW의 활용을 제고하기 위한 정책제안을 포함
- 서베이 결과는 국방분야에서의 공개SW 활용현황을 광범위하게 조사대상으로 하고 있고 상당한 수준의 광범위한 서베이를 수행한 결과를 분석, 요약하고 있어서 우리나라 공개SW 활용정책의 연구에 시사하는 바가 많으며, 공개SW 활용 현황의 요약은 아래와 같다.

< 표 4 > 미국방성에서 사용되고 있는 공개SW의 리스트

알파벳	공개SW명
A	ACE, ACE ORB (TAO), ACID, AMANDA, Apache, Autoconf, Automake
B	bash, Bastille, BIND
C	C++ Boost, Colt, Condor, COPS, Crack, CVS, C, CVW, Cygwin
D	DDD, DjVuLibre
E	EADSIM, Emacs, eTrust, Expect
F	FreeBSD
G	GateD, gawk (awk), GCC, GDB, Ghostscript, GNAT, GnuPG, gnuplot, grep
H	h2n, HOSTS
I	ImageMagick
J	JADE, Jakarta, Jaxen, JBoss, JDOM, Jikes, jSIP
K	Kaffe
L	LaTeX, Linux, Linux (Red Hat), Linux firewalls, Lsof
M	m4, Majordomo, make, Maxima, MIMESweeper, MRTG, MTR, MySQL
N	Nessus, NetBSD, NetSaint, nload, Nmap, ntop, NTP
O	Octave, OpenBSD, OpenMap, OpenOffice, OpenSSH, OpenSSL
P	Perl, Perl CGI scripts, PerLDAP, PHP, PingScan, Procmal
Q	Qmail
R	RealSecure, RRDtool, RTLinux, RWhois, RXVT
S	Samba, SARA, SATAN, Saxon, SCA, sed, SELinux, Sendmail SNARE, Snort, Squid
T	Tcl/Tk, TCP Wrappers, Tomcat, Top, Tripwire
V	VisAD, VOCAL, VTK
W	Webmin, WebTAS, Weka, WU-FTPd
X	Xalan, Xerces, XFree86, XGobi, Xpatch
Z	zlib, Zope

- 위의 표에 있는 115 종의 공개SW 응용프로그램이 현재 사용중
- 251개의 활용사례가 조사
 - 다양한 분야에 활용중
 - 많은 활용사례가 아직 미 조사되거나 발견된 것으로 판단
 - Linux 만이 아니라 다양한 응용과 사례가 조사

- 특히 주목할만한 사실은 공개SW의 활용이 광범위하게 이미 진행되고 있고 일반의 생각과는 다르게 상당한 수준으로 활용되는 사례가 다양한 분야에서 발견되고 있으며, 유의할만한 새로운 발견은 다음과 같다.
 - 공개SW는 보안분야에 강점을 보유
 - 인트라넷 분야에서 공개SW의 사용은 필수
 - SW개발분야에서의 공개SW 활용은 상당한 수준
 - 연구개발분야에서도 공개SW를 활용(Rapid innovation)
 - 비용 측면 만을 고려하여 공개SW를 사용하는 경우는 매우 드물지만 공개SW는 경쟁을 촉진시켜 비용통제의 역할을 함
 - 공개SW와 상용 SW를 연계하여 활용가능

2) 주요 조사 내용

- 국방분야에서의 공개 SW는 다음의 네 분야에서 강세를 보이는 것으로 조사되었다
 - 기반구조 지원 (Infrastructure Support)
 - 네트워크 지원(특히 인터넷관련 기술)
 - 소프트웨어 개발
 - C, C++, Ada, Perl, Python, 디버깅, source code 관리
 - 보안
 - 고신뢰도 운영체제, 감리, 긴급대응(rapid response)
 - 연구
 - 수학적 도구, 저-비용 슈퍼컴퓨팅, 디스플레이 도구 등
- 이 조사결과는 보안이나 운영체제(Linux) 분야를 공개SW의 강세분야로 생각하던 일반적인 견해와 많은 차이를 보이는 것으로 공개 SW 활용 정책의 중요성과 파급효과를 다시 생각하게 해주는 매우 중요한 조사결과이다.
- 또한 NCW 기반의 무기체계에서의 SW의 중요성과 의존도를 고려해볼 때 공개SW에 대한 정책적 연구와 활용방안에 대한 중장기적 Roadmap은 더 이상 늦출 수 없는 과제이기도 하다.

2) 국방분야 공개SW 정책의 접근방법

- Mitre 연구에서는 미 국방성에서 공개SW 정책의 도출을 위하여 아래 요약한대로 5가지 접근방법을 가정하여 그 결과를 분석하였다.
 - 접근방법 1: 모든 공개SW의 국방성에서의 사용을 금지
 - 접근방법 2: 관련 정책의 부재를 잠정적으로 허용(Limbo Status)
 - 접근방법 3: 공개SW의 사용을 선택적으로 허용
 - 접근방법 4: 전략 분야에 집중적으로 활용(보안, 인프라, R&D)
 - 접근방법 5: 공개SW 제품의 사용을 선도

- 정책 접근 방법 1: 모든 공개SW의 국방성에서의 사용을 금지
 - 이 정책은 모든 공개SW의 국방성에서의 사용을 금지하는 방안으로 그 영향이 상당히 부정적으로 분석되었다. 공개SW의 사용금지의 영향을 요약하면 아래와 같다.

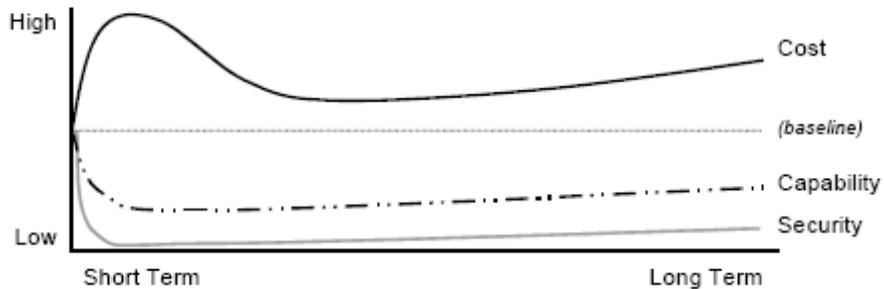


그림 11. 공개SW의 사용 금지에 대한 영향도

- 단기적 비용, 보안 및 능력에 미치는 영향이 심각할 것으로 예측
 - 비용측면에서 부정적인 영향이 심대
 - 국방분야에서의 정보관련 능력을 장단기 모두 저해
- 정책 접근 방법 2: 잠정적 정책의 부재
 - 정책 접근방법 2는 잠정적 정책의 부재(Limbo Status)를 상당기간 유지하는 방안으로서 현재의 상태와 유사하다고 할 수 있다.
 - 현재 공개SW의 사용은 미 국방성에서 허용도 불용도 아닌 상태.
 - 정책부재 상태는 프로젝트, 프로그램, 개발에서의 공개SW의 사용 여부에 대한 결정을 불가하게 만들
 - 개발자들은 공개SW 제품의 활용시 효과를 인지하고 있으나 공개 SW 사용불가의 결정이 내려졌을 때의 잠재적 위험때문에 상급자나 지휘관에게 공개SW 관련 정보를 보고 또는 공유하기를 주저하는 경향이 있음
 - 현재의 정책부재 상태가 국방분야 공개SW에 관한 de facto 정책이라고도 할 수 있음.

- 정책 접근 방법 3: 공개SW의 선택적 허용
 - 이 정책은 공개SW의 선택적 허용을 정책방안으로 현재의 공개SW 현황에 근거를 두고 입안된 방안으로서의 장단점을 보유하고 있다.
 - 시장에서 입증된 공개SW 제품은 국방분야 전반에서 선택적으로 사용이 허용됨.
 - 이 접근방법은 즉각적이고 심대한 긍정적인 효과를 가져옴
 - 이러한 공개SW 프로그램들은 국방분야에서 이미 다수의 사용자들을 가지고 있고 광범위하게 사용되고 있다는 이점을 가지고 있다.
 - 선택적 허용정책은 기존의 사용자 기반으로 인하여 이러한 양질의 제품의 사용을 단기간에 확산시키는 긍정적인 효과를 초래함
 - 주요 문제점: 잠재력 있는 공개SW를 간과할 우려가 있음

- 정책 접근 방법 4: 전략 분야에서 공개SW의 적극적인 사용의 허용
 - 이 정책은 전략적 분야에서 공개SW의 적극적인 사용을 허용하는 방안으로 찬반이 극명하게 엇갈리는 정책이다.
 - 전략 분야에서 공개SW 제품의 허용: 인프라, 보안, 연구개발 등.
 - 전략적 분야에서의 공개SW 제품의 사용지침을 채택하고 인가된 제품정보를 전략적 분야에서 광범위하게 활용하는 정책

- 정책 접근 방법 5: 공개SW의 활용을 선도하고 권장
 - 이 정책은 공개SW의 활용을 선도하고 권장하는 방안으로 미 국방성에서도 단기적인 실현가능성에 회의를 가지는 방안이다.
 - 공개SW 제품의 사용을 권장/강제하는 접근방법
 - 이러한 선도정책은 양질의 공개SW 제품이 강세를 보이고 있는 전략적인 분야에서는 가능하고 현실적인 접근방법.
 - 공개SW 제품이 강세를 보이고 있는 분야에서는 현재 시스템이나 SW의 이점을 극대화할 수 있는 형태로 그 활용의 행태가 고도화될 수 있을 것으로 예상
 - 데스크탑은 현재로서는 상용 독점 SW의 영역으로 남아있을 것임 .

3) 결론 및 제안

가) 전반적 공개SW 사용 정책의 개발

- 미 국방성은 전반적으로 공개SW 활용 정책을 개발하여 양질의 공개SW 제품이 광범위하고도 효과적으로 사용되도록 하고 공개SW와 상용제품과의 효율적인 연계를 장려하는 것이 필요하다(예: UNIX 상에서의 Windows 서비스)
- 전략적 분야별 정책은 아래의 각 분야별 사용에 효과적으로 대처할 수 있도록 입안하는 것이 중요하다.
- 기반구조와 개발분야: 정책적 고려가 GRAS 제품의 효율적인 사용에 초점을 맞출 필요가 있다.
 - 예: Apache, Linux, GCC 등은 광범위하게 사용되나 사용정책의 미흡으로 모호한 상태에 있음 .
- 보안분야: 잘 정의된 보안 세부분야에서의 GPL 활용의 권장지침은 사이버 위협에 효과적이고 신속하게 대응할 수 있는 Autonomous 해법을 제공할 것이다.
- 연구분야: 기반연구에서의 공개SW 사용의 장려를 통하여 관련 지식을 공유하고 상업적 혁신을 지원할 수 있는 방향으로의 정책이 필요

나) 공개SW 사용의 다양성 이점

- 공개SW는 비용측면에서 상용제품에 비하여 대단히 저렴하다는 장점을 보유하고 있고 기능성과 사용자 지지도에서도 상당한 수준에 도달해있다. 국방분야 시스템의 획득과 아키텍처측면에서 제품의 다양성의 확보라는 두 핵심목표를 충족시킬 수 있는 대안으로 부상하고 있다.
- 획득 다양성 (Acquisition diversity)은 단일 SW 제품에 전적으로 의존적일 경우 비용과 보안측면에서의 위험을 감소시키는데 기여하는 반면에 아키텍처 다양성 (Architectural diversity) 은 기존에 광범위하게 사용되는 제품의 특정 기능이나 약점에 대한 자동화된 침투에 기반한 재앙적인 사이버 위협의 위험을 감소시키는데 기여한다.

■ 결론적으로

- 공개SW제품의 금지나 제한은 국방분야의 보안이나 국방력(Defense Capabilities)에 부정적인 영향을 끼칠 것으로 판단.
- 공개SW 제품과 상용 SW를 같이 운용할 수 없을 것이라는 우려와 공개SW 제품의 모호한 상태로 인한 불확실성은 공개SW제품의 최적 사용을 저해하고 있다.

■ MITRE 의 제안

- 미 국방성은 다음의 3 단계 정책의 개발을 통하여 공개SW 사용을 최적화 시킬 필요가 있다:
 - "안전인정"(GRAS) 공개SW 인증제도의 구현하여 품질과 유용성을 제고한다.
 - 전략적 분야 (Generic, Infrastructure, Development, Security, & Research)에서의 공개SW 활용정책을 개발.
 - SW 제품의 다양성 확보를 위하여 공개SW의 사용을 권장.
- "안전인정" (GRAS: Generally Recognized As Safe)은 일종의 자격증제도로서 공개SW GRAS 자격은 SW의 공식적 인정체계를 구축하여 민간분야에서 상용으로 지원되고, 광범위하게 사용되며, 안전성과 신뢰성측면에서 입증된 기록을 보유하는 체도를 구축하고자 하는 것이다. 현재 유사한 체계로는 보안분야 CERT 체계를 들 수 있다.
- 최초 단계 고려대상은 기존사용 중인 115개 응용 SW(기반구조나 개발분야에서 광범위하게 활용되는 양질의 SW, 즉 Linux, BSD, Samba, Apache, Perl 등)를 포함하고 기존에 조사되지 않았으나 그 활용이 광범위하게 인정되는 새로운 SW군 (예: Python 등)을 포함하고 있다.

다. 개방형 기술개발 정책 연구

1) 개방형 기술개발의 적용에 대한 필요성

- 국방성은 다음과 같은 서로 상충되는 두 가지의 목표를 추구한다.
 - 국가의 안전보장을 제공
 - 국가의 산업기반을 진흥하고 지원
- 국방분야에서의 디지털 기술의 확산은 디지털제품(특히 SW)의 관리의 중요성이 더욱 높은 우선순위를 가지게 됨을 의미한다. 다음의 이슈들은 공개SW를 관리하는데 새롭게 고려해야 될 사실들을 포함하고 있다.
 - SW 및 서비스의 획득에 필요한 BM(Business Model)은 기존의 BM과 매우 상이함
 - 시장은 새롭고 혁신적인 해법과 능력을 위한 경쟁체제로 변화
 - 국방성의 공개SW 도입과 관리는 관련 산업이 경쟁력과 유연성을 갖추도록 동기부여와 자극을 제공
 - 국방성의 공개SW 도입 및 관리에 관한 정책과 제도적 장치가 필요

2) 국가안보와 공개SW

- 미국은 공개SW와 개방형기술개발(OTD)정책이 국가안보에 가지는 중요성이 매우 높고 중장기적인 중요성은 더욱 높아질 것이라고 판단하고 있으며, 공개SW가 국가안보에 미치는 영향과 중요성은 다음과 같다.
 - IT 산업의 유연성을 증진시켜 급변하는 사용자의 요구에 신속하고 효과적으로 대응할 수 있게 함
 - 디지털 경제에 더 적합한 산업의 경쟁을 촉진시켜 산업기반을 강화
 - 중요성이 증대되고 있는 IT 분야의 국제경쟁력측면에서의 자국 산업의 위상을 인지하고 경쟁우위를 선점하고 유지가 가능함
 - 중국은 공개SW 분야에서 우위를 점하고자 국가적인 노력을 경주
 - 국방분야에서 실제 사용되고 있는 공개SW를 인지하여 중요한 기반구조와 보안을 강화할 수 있는 능력을 배양

- 기술산업분야의 급격한 변화와 적대적 행위에 대응할 수 있는 신속 대응체계를 준비

3) 개방형 기술개발(OTD)

- 국방성은 공개SW의 지식베이스(KB: Knowledge Base)를 구축 관리하고 Interface, Modularity와 재사용(Reuse)의 내부문화를 제고하기 위한 전략을 도출 구현하여야 한다. 이를 위하여 개방형기술개발은 다음의 분야에서의 기술의 발전을 반영할 수 있어야 한다
 - 개방형 표준 및 Interface
 - 공개SW 개발모델
 - 협업적 문화 (Collaborative Culture) 와 온라인 지원도구
 - 기술적 유연성(Agility)
- 표준 Interface를 사용하고 목표기반의 협업적 개발방법론이 개방형 기술의 개발에 효과적이다.
- 공개SW 개발 모델은 협업적 개발방법론에 의한 새로운 기술혁신 모델의 가능성을 열어준다. 아래 그림과 같이 소스코드를 공개하여 누구라도 소스코드를 가지고 버그를 수정하고 자신의 목적에 맞게 코드를 수정사용 할 수 있는 기본조건을 제공하여, 기업의 R&D 비용과 절감시키고, 기업 성장을 위한 새로운 비즈니스 기회를 제공하기 때문이다.
- 기존의 획득방법에 의한 방식에서는 아래 그림과 같이 요구사항 중심의 개발방법으로서 요구사항 정의를 시작점으로 하여 자원배분에 따른 스케줄링 및 구현 방식으로 과제는 단계를 거치면서 소과제로 나누어지고, 개발자는 자신의 소과제만을 개발하기 때문에 구현과정에서 약간의 변화가 있게된다.

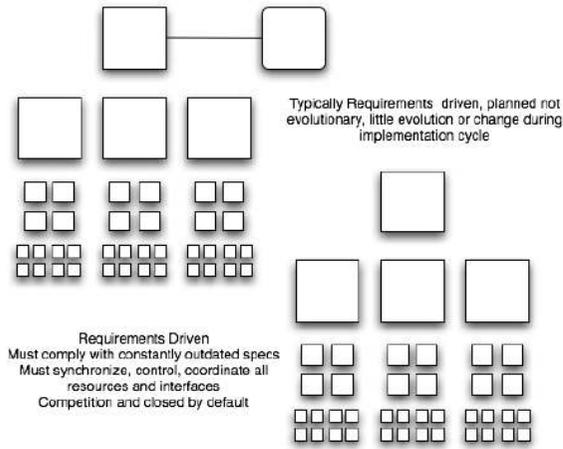


그림 12. 요구사항 중심에 의한 기존의 개발 방법

- 그러나, 공개SW 개발 모델을 적용한 협업적 개발방법론은 목적중심의 개발방법으로서, 다수의 개발자가 병행적으로 프로젝트에 참여하고 있으며, 조직화된 동료집단의 커뮤니케이션에 의해 프로젝트가 질서화 하고 중앙으로 집중화 시키는 방식에 의해서 작동하고, 이러한 병행적 개발 방식은 모듈라 아키텍처에 의해 각 부분이 협업적 분업에 의해 동시에 개발되며, 이 개발방법론을 통해 변화에 대한 속도, 요구사항 대비 목적중심의 결과 비교, 그룹웨어 중심의 IT 문화 등 많은 기술혁신을 기대할 수 있다.

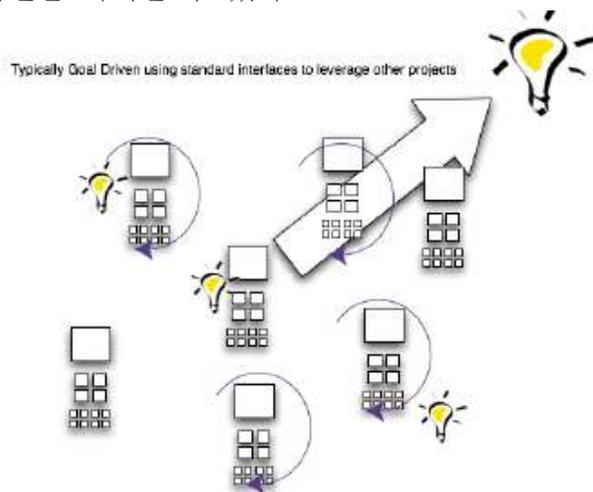


그림 13. 목적 중심에 의한 협업적 개발 방법

4) 개방형기술개발의 핵심 이슈와 목표

- 개방형기술개발의 핵심 이슈들은 다음과 같다.
 - 국방성이 국방분야 SW 개발의 관리를 더 효과적으로 할 수 있는 방법은 무엇인가?
 - 어떻게 개방형기술개발의 BP(Business Process)에 기반한 장점들이 SW 개발의 혁신적 개선과 유지를 가능하게 할 것인가?
 - 개방형기술개발의 장점을 극대화하기 위하여 국방획득분야 정책과 제도를 어떻게 변화시켜야 하는가?
 - 국방성이 기존의 외부 공개SW 자원을 활용할 수 있는 방안은 무엇인가?
- 이의 성공적 구현을 위한 개방형기술개발의 목표와 추진체계는 다음과 같이 요약될 수 있다.
 - 목표: 개방형기술개발의 적정구현을 위한 현행 기술획득절차의 혁신적 변경
 - 현행 시스템의 성공적 혁신을 위하여 다음의 보상체계가 필요
 - 외부 시스템의 활용을 장려
 - 충분한 효과를 위하여 필요한 과학적 위험관리 도입
 - 개발주기관련 비용과 장점을 다른 프로젝트에 이전하기 위한 요인 추출 및 이전
 - 요구사항결정, 정책, 절차, 및 평가 등의 분야에서 점진적 변화가 개방형기술개발의 구현을 위하여 효과적이며, 아래 그림은 개방형기술개발의 실현을 위한 로드맵으로 국방성의 산하기관인 AS&C에서 여러 가지 프로젝트를 수행하면서 적용토록 계획되어 있다.

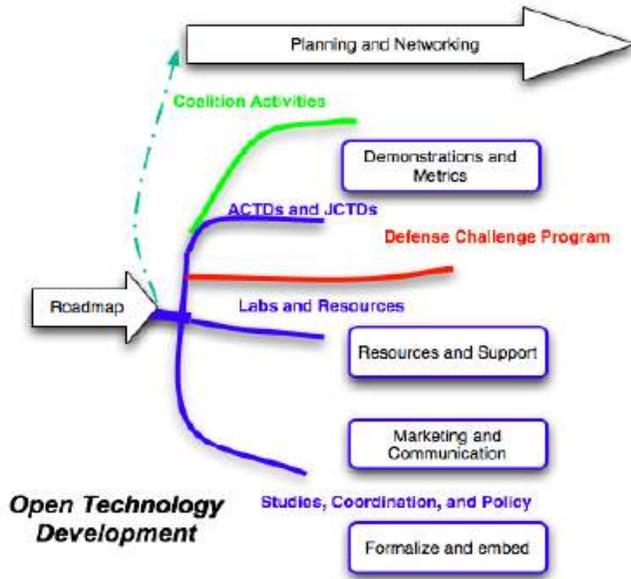


그림 14. 개방형 기술개발 로드맵

5) 개방형기술개발의 단계별 구현 및 적용분야

- 개방형기술개발의 성공적 구현을 위하여 필요한 단계 및 대상분야는 다음과 같다.
- 개방형기술개발의 구현단계
 - 단기: AS&C 프로젝트에 시험적용
 - 중기: 미래 (FY07) AS&C 프로젝트에 개방형기술개발의 요구사항과 절차를 확산 적용
 - 장기: 기타 분야로 확산

- AS&C(Advanced Systems and Concepts) 개요
 - AS&C의 역할 : 국방성의 산하 기관으로서 NCW에 필요한 최선의 운용개념 및 기술의 발견, 적용 및 전환 등을 수행하는 기관임.
 - AS&C의 조직 및 기능

<표 5> AS&C의 조직 및 기능에 대한 정리 표

부서	기능
JCTD Joint Capabilities Technology Demonstrations	주요무기체계의 개선을 위해 개발자로부터 사용자에게 성숙된 기술을 활용케하고 새로운 기술과 개념을 도입하고 적용시키는 부서
CTO(비교시험국) Comparative Testing Office	혁신적이면서도 비용이 저렴한 기술을 자국내에서 또는 해외에서 도입한 후 시험하는 부서
OTT(기술이전국) Office Technology Transition	연구개발에서 무기체계에서의 전환까지를 담당하는 부서로 실제적으로 아래와 같은 프로그램을 수행함 - 기술전수, 독립연구 및 개발, 중소기업혁신연구 - 북미기술산업기반국, 성숙된 기술 프로젝트 - 방위생산 Act Title III, 기술전환주도
JCOS(합동연합운용지원국) Joint and Coalition Operations Support	해외를 포함한 합동연합작전에서 지휘관의 요구사항에 대응하는 부서
PR&I(프로그램자원통합국) Program Resources and Integration	예산을 관리하는 부서

- 개방형기술개발의 적용 대상 분야
 - 공개SW 기반구조와 기술을 관리
 - 공개SW의 협업기술을 관련 소규모 대상에 적용
 - 현재의 획득 및 개발관련 행태를 공개SW 및 기술서비스에 적합하도록 변화를 유도. 적용 대상분야 중 관련 프로그램과 기타 사용가능한 자원들은 국방성 산하의 공개SW의 적용대상으로 현 시점에서 확산 가능한 분야이다.

6) 개방형기술개발의 성공적 구현을 위한 제안

- 개방형기술개발의 성공적 구현을 위하여 점진적이고 단계적인 확산 전략이 필요하다. 공개SW의 구현을 위한 정책, 절차, 요구사항, 및 현행 규정 등을 수정보완하여 공개SW, OIS, 협업기술방법론 등을 기본으로 활용할 수 있도록 점진적인 변화를 추구하여야 한다. 그 적용 대상으로는 AS&C 프로젝트에 우선적으로 시험적용하고 향후 기타 기관과 사업에 확산하는 방안이 현실적이다.
- 주요 제안은 아래와 같이 압축될 수 있다.
 - 개방형기술개발 추진 팀을 구성운영
 - 개방형기술개발을 촉진시키기 위한 외부활동을 정규화
 - AS&C 프로젝트에 우선적용
 - FY07 사업승인을 위하여 필요한 새로운 변화를 강화할 수 있는 평가체계, 정책 및 절차를 구현
 - 외부 관련기관과 사업에 비전을 홍보하고 전파
 - AS&C내에 개방형기술개발 자문위원회를 설치운영

라. 시사점 및 교훈

- 2002년에 수행된 미 국방성에서의 공개SW활용에 관한 정책연구에 대한 MITRE의 보고서 분석을 통한 시사점 및 교훈은 다음과 같은 3가지로 요약된다.
 - 미 국방성에서는 이미 2002년에도 공개SW가 다양하고 수준높게 활용되고 있다는 점
 - 비용보다 필요성에 의해 공개SW를 활용하나 경쟁에 의해 가격이 조정된다는 점
 - 공개SW가 국방시스템의 보안에서 우수하다는 점
- 또한 2006년에 수행된 미 국방성에서의 개방형기술개발의 정책연구에 대한 보고서 분석을 통한 시사점 및 교훈은 다음과 같은 4가지로 요약된다.
 - 미 국방성의 정책에 대한 기본방향은 국방력 향상 및 관련 산업의 육성이 동시에 고려된다는 점
 - 개방형기술개발이란 공개SW개발모델, 개방형 표준 및 인터페이스, 기술의 유연성 및 협업적 개발방식의 4가지가 결합된 문화로서 대상도 SW를 포함한 첨단 IT제품이라는 점
 - 개방형기술개발의 적용을 위해서는 국방획득제도의 변경이 필요하다는 점
 - 개방형기술개발의 성공적 적용을 위해서는 관련된 조직, 예산이 필요하다는 점

1) 공개SW활용에 관한 정책 연구

가) 미 국방성에서의 공개SW 활용 현황

- 미 국방성에서의 공개SW 활용 현황
 - 2002년도에 서베이를 통해 분석한 미 국방성의 공개SW 활용 현황을 보면 앞에 기술한 바와 같이 115종의 공개SW가 251개의 응용분야에서 매우 다양하게 활용되고 있는 것으로 리눅스만이 아니라
 - 매우 다양한 응용사례와 사용자가 있고
 - 인터넷에서 공개SW의 사용을 금지하면 네트워크가 정지될 정도이고
 - SW개발 및 연구개발 분야에서 많은 공개SW가 유용하게 활용되고 있으며
 - 상용SW와 공개SW와 상호 연동되어 운용되고 있다는 것이다.
 - 2002년도 MITRE에서 이 연구를 수행한 Terry Bollinger가 GOSCON 2005에서 발표한 자료에 의하면 1970~80년대에는 아래 왼쪽의 표준 자원들이 회사 또는 조직에서 오른쪽과 같은 그림의 혁신을 이루는데, 이러한 혁신은 invisible hand로 제한 되었지만,

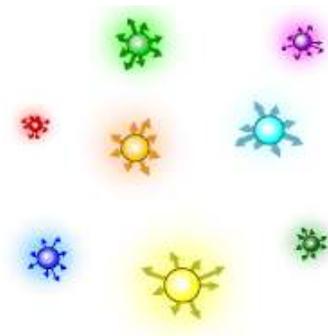


그림 15. 표준자원 모습

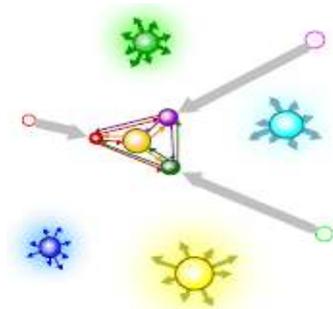


그림 16. 혁신에 의한 결합

- 1990년대부터의 free market에 의한 활동으로 invisible hand가 자류로워지면서 아래와 같은 다양한 분야에서 혁신이 일어나고 미 국방성에도 이러한 적용이 이루어진 것으로 분석하였음.

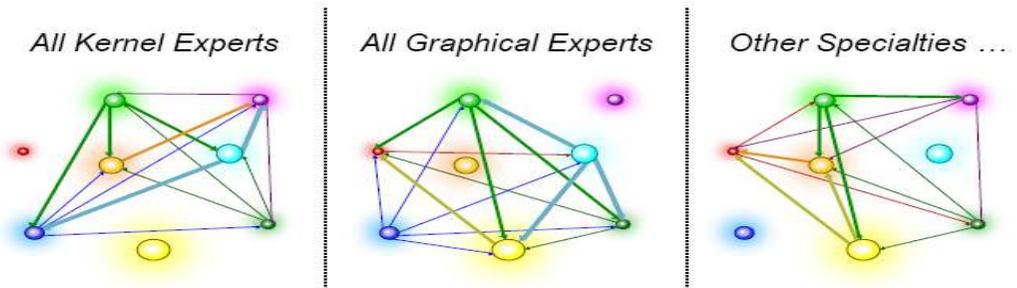


그림 17. free market에 의한 다양한 분야에서의 혁신 모습

- 또한 아래와 같은 2가지의 경우에 공개SW가 업무에 가장 많은 기여를 하는 것으로 분석되었다.
 - 첫째는, 어떠한 방법으로 공유하던 간에 공개SW가 인프라의 역할을 하는 경우이고
 - 둘째는, 기존의 방식으로는 비즈니스에 핵심이되는 혁신이 이루어지지 않는 경우이다.
- 미 국방성의 공개SW활용에 관한 연구를 직접 수행한 Terry Bollinger의 이러한 분석은 우리나라에서 공개SW를 활성화하기 위한 정책과 관련하여 많은 시사점을 주고 있는데, 위의 내용을 이용하여 공개SW를 활성화하기 위한 전제조건은 무엇인가? 라는 의미에서 해석할 경우 공개SW의 소스를 공유하여 업무를 혁신하기 위한 필요성이 존재하고, 공개SW가 많은 관련자들 간에 인프라로서 공유할 수 있는 환경이 중요하다는 것을 의미하며 이미 앞에서 기술한 것처럼 미국방성에서는 이러한 조건이 갖추어져있는 것으로 분석되었다. 그러나 우리나라의 경우에는 서버 중심의 리눅스 또는 MySQL, Apache 등의 일부 공개SW만이 활용되고 있어 공개SW가 기본적 인프라로 공유할 수 있는 환경이 미비하고, 또한 공개SW를 활용하여 업무를 혁신시켜야만 되는 필요성이 부족한 것으로 보여지기 때문에 이러한 조건을 개선할 정책적 지원이 필요하다.

나) 공개SW 활용과 경쟁을 통한 예산 절감

- 비용 측면의 공개SW 활용 현황
 - 공개SW는 상용SW에 비교하여 라이선스 비용이 발생하지 않기 때문에 비용측면에서 우수하다는 것은 주지의 사실이다. 그런데 미 국방성의 공개SW 활용에 관한 정책연구 결과 공개SW를 선택하는 중요 요소는 비용보다 필요성인데, 그러나 공개SW를 사용하면서 가격에 대한 조정이 가능하고 이로 인해 결과적으로 전체적인 예산이 절감되었다고 분석된 것이다.
 - 일반적으로 가격은 수요와 공급의 관계에 의해 형성되는데, SW의 공급에서도 파는 측이 가격을 지배하는 공급독점이 이루어지면 독점자인 개발업체가 가격을 조정하지만 공개SW를 활용하여 SW의 수요자이면서도 일부분에 대한 SW의 공급을 하게되면 이를 통해 독점공급에서 경쟁의 상태로 바뀌게 되면서 수요자가 가격을 조정할 수 있게되는 것으로 보여지고, 미 국방성에서는 라이선스의 비용절감보다 경쟁을 통한 가격조정이 전체적인 예산절감에 더 큰 영향을 준 것으로 분석된다.

다) 공개SW에 대한 보안 분야에서의 우수성

- 공개SW와 상용SW의 보안에 대한 논란
 - 공개SW인 리눅스가 상용SW인 윈도우 보다 보안에서 우수하다는 의견
 - 레드햇은 2005년 8월 SANS 연구소의 보고서에서 밝혀진 20개의 인터넷 보안 취약점들 가운데 단 2개의 가벼운 취약점만이 리눅스 사용에 영향을 주었고, 이 또한 RHN이 제공하고 있는 패치로 해결 가능 때문에 리눅스가 윈도우보다 보안에서 우수하다고 주장.
 - 1989년 설립된 SANS(SysAdmin, Audit, Network, Security)는 정보 보안 트레이닝과 인증 분야에서 큰 영향력을 발휘하는 기관으로서, 분기별로 리눅스, 유닉스 및 윈도우, 매킨토시와 같은 다양한 플랫폼상의 보안 취약점을 분석하는 보고서를 발표하고 있으며, 보

안 전문가들과 네트워크 사용자들에게서 가장 신뢰할 수 있는 보고서라는 평을 받고 있다.

■윈도즈가 리눅스보다 보안에서 우수하다는 의견

- 리눅스가 데스크 탑 컴퓨터에서 윈도즈보다 더 많이 사용되지 않기 때문에 타깃이 되는 경우 적고, 패치수가 적은 것이지 단순히 패치수에 따라 보안의 우수성을 비교하는 것은 논리적이지 못하다.
- 조사기관인 포레스터가 2004년 4월 발표한 자료에 의하면 리눅스가 윈도우보다 더 안전하다고 믿고 있지만 이는 단순하게 볼 수 없는 매우 복잡한 문제로서, 보안의 결함은 시간의 흐름에 따른 수명과 각각의 처리 단계와 관계가 있는 것으로, 단순히 빠른 패치출시와 패치의 적용보다 업체가 자사의 제품의 알려진 결함에 대해 수정을 얼마나 빨리 처리하는가와 결함이 발견되었다면 타 제품대비 얼마나 심각한가?이고, 이를 계산하기 위해 포레스터는 위험노출기간과 결함에 대한 위험도(ICAT)를 적용하여 비교한 결과, 윈도즈는 위험노출기간이 평균 25일로 가장 빠르지만 높은 위험도에서 윈도즈는 67%인 반면 리눅스는 56%의 관정을 받음

■ 공개SW를 활용하고 있는 미국방성에서의 보안

- 위에 기술한 것처럼 공개SW의 대표적인 리눅스와 상용SW의 대표적인 윈도즈의 보안에 대한 비교는 논란이 많은 것이 사실이다.
- 그러나 미 국방성에서의 공개SW활용에 대한 연구 결과를 보면 인터넷에서 공개SW를 금지하면 네트워크가 정지되고, 아래 표와 같이 보안관련 분야에서 활용하고 있는 총 44개의 공개SW 중에서 소스를 가장 많이 공개하는 GNU라이센스가 오히려 공개와 반대되는 개념의 보안분야에서 가장 많이 활용되고 있는 사례를 볼때 우리나라 국방시스템에서의 보안 강화를 위해서는 공개SW의 활용이 필수적일 뿐만 아니라, 리눅스 중에도 리눅스, 리눅스(레드햇), SELinux 등 다양한 종류의 버전이 업무에 따라 다르게 적용되고 있는 것을 알 수 있다.

<표 6> 보안 관련분야에서 활용되고 있는 공개SW 리스트

GPL (26) [59%]	ACID	Bastille	COPS	GCC	GnuPG
	HOSTS	Linux	Linux (Red Hat)		MRTG
	MTR	MySQL	Nessus	NetSaint	nload
	Nmap	ntop	Perl	Perl CGI scripts	PerLDAP
	RRDtool	SELinux	SNARE	Snort	Squid
	Tripwire				
BSD (5) [11%]	FreeBSD	OpenBSD	OpenSSH		
	Tcl/Tk	Webmin			
Closed from open: eTrust , MIMESweeper , RealSecure (3) [7%]	eTrust	MIMESweeper			
	RealSecure				
SATAN (2) [4.5%]	SARA	SATAN			
Artistic (1) [2%]	Crack				
Community: CIS (1) [2%]	CIS Benchmarks				
Lsof (1) [2%]	Lsof				
OpenSSL (1) [2%]	OpenSSL				
PHP (1) [2%]	PHP				
Public Domain (Expect) (1) [2%]	Expect				
Qmail (1) [2%]	Qmail				
TCP Wrappers (1) [2%]	TCP Wrappers				

2) 개방형 기술개발에 관한 정책 연구

가) 국가 안보 및 산업 육성

- 국가 안보와 산업 육성의 관계
 - 세계 어느 나라를 막론하고 주권국가는 국민의 자산과 생명을 스스로 보호할 수 있는 조치를 취하는 국가 안보를 국가의 첫 번째 의무로 하고 있고, 이를 위해 국방력 향상을 위한 방위산업을 육성하고 있다.
 - 방위산업은 기술적 특성상 전자, 기계, 항공, 조선, 자동차, 화학공업, 정보산업 등 거의 모든 산업과 밀접하게 연관되어 있어, 방위산업육성과 동시에 관련산업을 육성하는 것이 바람직하나, 많은 경우 목표가 상반되는 트레이드 오프의 관계를 가져 동시에 육성하기 어려운 경우가 있다.
- 미 국방성의 정책 및 우리나라 u-디펜스에서 고려해야 할 사항
 - 미 국방성의 경우에는 국가안보와 관련산업의 동시 육성이라는 큰 틀을 그대로 유지하고 있다. 특히 NCW를 위한 첨단 무기체계는 대부분 복합무기체계로서 고도 정밀화되면서 SW개발에 소요되는 비용이 지속적으로 증가되고 있기 때문에 획득비용이 막대하여 국가재정과 국방예산에 미치는 영향이 대단히 크며, 이러한 무기체계 획득 기회를 어떻게 활용하여 관련 산업을 동시에 발전시키느냐에 대해 정책적으로 매우 중요하게 고려하고 있다.
 - 우리나라도 참여정부들어 미래 사회를 대비한 유비쿼터스 사회 조성이 매우 중요하다고 인식하고, 정통부를 중심으로 u-시티, RFID/USN, 텔레매틱스, 홈네트워크, u-디펜스, u-페이먼트, u-헬스 등 7개 선도과제를 선정해 관련부처와 협동하여 사업을 추진 중인데, 미국방성의 개방형기술개발 정책에서 고려하고 있는 국가안보 및 SW산업의 동시 육성 정책은 현재 정통부와 국방부가 서로 협력하여 진행하고있는 u-디펜스의 추진에 많은 참고가 될 것이다.

나) 개방형 기술개발의 개념 및 대상 범위

■ 개방형 기술개발의 개념

- 개방형 기술개발에 대한 개념은 앞에 기술한 것처럼 공개SW의 리눅스, 또는 MySQL 등과 같은 어떤 특정한 SW제품이나 CDMA 또는 TDMA 등과 같은 어떤 특정한 기술을 의미하는 것이 아니고, 공개SW를 개발하는 모델을 중심으로 하면서, 여기에 개방형 표준 및 인터페이스, 새로운 기술의 유연한 적용 그리고 이러한 3가지 사항들이 서로 협업적으로 결합되면서 결과적으로 혁신을 가져오게 하는 큰 개념의 개발 방식을 의미함
- 공개SW의 개발모델의 장점은 소스를 보다 많은 개발자가 공유하는데 큰 장점이 있고,
- 개방형 표준 및 인터페이스의 장점은 인터넷의 TCP/IP, 개방형 네트워크인 OSI 등을 필두로 해서 J2ME/MIDP, IMS 등의 사례와 같이 현재 세계적 기술추세가 개방형의 방향으로 진행되고 있고,
- 새로운 기술의 적용에서는 과거 15년동안 무선통신의 방식이 1세대 AMPS, 2세대 TDMA/CDMA, 3세대 W-CDMA, 4세대 OFDM으로 발전된 것처럼 이러한 새로운 기술에 유연하게 적응해야하고,
- 협업적 개발방법은 지금까지의 지휘관이 요구하는 요구사항 중심의 시스템 개발 방법을 목적지향적인 개발방식으로 바꾸고, 기술한 3가지 사항이 결합되면서 개발하는 과정 중에 기술혁신이 이루어지는 것을 의미한다.

■ 개방형 기술개발의 대상 및 범위

- 미국방성에서 고려하고 있는 개방형기술개발의 대상 및 범위는 공개 SW에 한정된 것이 아니라 공개SW가 핵심이기는 하지만 NCW에 필요한 모든 IT 시스템을 대상으로 하고 있는 것이다.
- 2005년말 기준 주요산업에서 차지하는 전반적인 SW의 원가비중이 평균 33.5%이지만, NCW에서 핵심 전투무기 중의 하나인 F-22의 경우 SW가 무려 1,500만 라인이면서 원가비중이 약 80%라는 것을 고려한 것으로 분석된다.

다) 개방형 기술개발과 국방획득 제도와의 연관성

■ 국방획득제도

- 아래 그림은 우리나라에서 추진되고 있는 무기체계의 획득 절차를 개략적으로 보여주는 것으로, 미 국방성도 이러한 절차를 이용하여 무기를 획득하고 있지만 이 체계는 기존의 무기체계에 적합하였으나 NCW에 필요한 첨단 무기체계의 획득의 경우, 특히 개방형 기술개발이 적용되어야 할 SW가 핵심인 무기체계에서는 새로운 획득절차가 필요한 것으로 분석되었다.

그림 18. 무기체계의 획득 절차에 대한 개략도

■ 개방형기술개발 적용을 위한 국방획득제도 변경의 필요성

- 위의 표에 나타나고 있는 바와 같이 새로운 무기체계의 획득에는 개념연구, 탐색개발, 체계개발의 3단계로 구분되어 각각의 절차에 따라 엄격하면서도 보수적인 규정에 의거 진행되고 있기 때문에 개방형기술개발에서 수용하고 있는 개방형 표준, 새로운 기술의 적용 및 협업적 개발 방법론이 실제적으로는 적용되기가 불가능한 실정이다.
- 미 국방성에서도 이러한 사항을 인식하여 미 국방성 산하 기관으로 첨단개념 및 시스템에 관련된 업무를 수행하는 AS&C를 통해 여러 가지 프로젝트를 수행하면서 획득절차를 검토하고 있다.

라) 개방형 기술개발을 위한 지원 체계

- 미 국방성의 국방력 향상을 위한 체계
 - 아래 그림은 국방력향상을 위한 체계를 보여주고 있다.

그림 19. 미국방성의 국방력향상을 위한 체계도

- 이러한 국방력 향상을 위한 조직이 AS&C로서 아래와 같은 부서에서 역할이 구분되어 추진되고 있으므로 우리나라에서 개방형기술개발을 적용할 경우 참고가 될 것이다.
 - JCTD : 개발자로부터 사용자에게 성숙된 기술을 활용케하고 새로운 기술과 개념을 도입하고 적용시키는 부서
 - CTO : 혁신적이면서도 비용이 저렴한 기술을 자국내에서 또는 해외에서 도입한 후 시험하는 부서
 - OTT : 연구개발에서 무기체계에서의 전환까지를 담당하는 부서로 실제적으로 많은 프로젝트를 수행함
 - JCOS : 합동연합작전에서 지휘관의 요구사항에 대응하는 부서
 - PR&I : 예산을 관리하는 부서

4. 우리나라 국방전력발전 규정과 공개SW활용현황

가. 국방시스템관련 기존의 연구현황

1) 기존의 연구현황 개요

- 소프트웨어 진흥원은 그동안 우리나라 공개SW 산업의 활성화를 위하여 많은 노력을 하였고, 이를 위해 많은 정책연구 과제를 수행하고, 정책에 반영함으로써 공개SW의 활성화에 큰 기여를 하였으며, 본 연구와 직접적으로 연관된 기존의 정책연구 과제는 다음과 같다.
 - 공개sw 가이드(정통부, '06. 1)
 - 공개sw 도입·운영 가이드(국방부, '06. 3)
- 이중 공개SW가이드는 공개소프트웨어 대한 이해를 높이고 공개소프트웨어에 대한 우려와 현실을 객관적으로 분석하여 공개소프트웨어 도입 시 필요한 검토사항 및 위험요소 분석, 관리방안 등을 수립하는데 도움을 주기 위한 목적으로 작성한 가이드로서 '06년 1월에 정통부에서 발표를 하였고, 이 가이드에 포함된 주요내용은 다음과 같으며, 부록으로서 신뢰성이 높은 공개SW 36종을 추천하였다.
 - 공개sw의 개요
 - 공개sw에 대한 오해와 진실
 - 공개SW도입 유형 및 절차
 - 위험분석 및 관리
 - 법률문제
- 반면, 공개SW 도입·운영 가이드라인은 국방부의 주관 하에 작성된 것으로 서버 시스템 운영체제(OS)의 공개SW 전환을 권장하고, 우선 웹 기반 업무 시스템과 홈페이지 구축 시 리눅스와 모질라 사용자의 접근이 가능하도록 개발할 것 등의 내용과 함께 웹 서버는 물론, 애플리케이션 서버, 데이터베이스관리시스템(DBMS) 서버 등에도 순차적으로 리눅스를 도입토록 권장하는 내용이 포함되어 있으며, 신뢰성이 높은 공개SW 21종을 추천하였다.

2) 기존 연구를 통한 공개SW와 비공개SW의 차이점

- 공개SW의 현실적 활용가치와 기능성 등을 확인하기 위하여 기존 연구를 통해 분석한 비공개SW와 공개SW의 차이점은 다음과 같다.
- 공개SW는 소스코드의 공개, 개작, 재배포가 자유로운 소프트웨어를 의미하는 것이며, 비공개SW는 이러한 권리가 제한된 소프트웨어를 의미한다. 공개SW를 상용소프트웨어와 반대되는 개념의 소프트웨어로 이해하는 경우가 있으나, 이는 공개SW를 무료 소프트웨어로 잘못 이해하고 있기 때문에 생긴 오해이다. 공개SW도 유료 즉 상용으로 배포할 수 있다.
- 첫째, 가장 대표적인 차이점으로서 비용면에서의 차이점이다.
 - 비공개SW는 라이선스 비용이 발생하는 반면 공개소프트웨어는 라이선스 비용은 발생하지 않으나, 기술지원이나 서비스에 대한 수수료는 발생한다. 또한 비공개SW는 시스템에 대한 초기 적용비용이 높지만 공개SW는 초기 적용비용이 매우 낮다는 점이다.
- 둘째, 성능면에서의 차이점이다.
 - 비공개SW의 경우 전문가가 직접 설치 및 최적화 설정을 하기 때문에 규모가 큰 시스템 환경에서 널리 사용되어 왔다. 공개SW의 경우 최적화가 이루어지지 않은 경우가 많으나, 공개SW의 경우에도 전문가를 통해 최적화하여 설치한다면 비공개SW의 경우와 같이 규모가 큰 시스템 환경에서 사용될 수 있을 정도의 성능을 보여준다. 또한 버전이 올라가면서 점차적으로 우수한 성능을 보여주기도 한다. 대표적인 공개SW 운영체제인 리눅스의 경우 커널 2.6.X 이상에서는 중대형 시스템에서 안정적 성능을 나타내고 있으며 이러한 추세는 앞으로 더욱 활발해 질 것으로 예상된다.
- 셋째, 보안면에서의 차이점이다.
 - 비공개SW는 프로토콜의 호환이 어려워 인증체계가 취약하며 폐쇄적인 운영으로 인해 공개되지 않는 약점이 존재할 수 있다. 반면 공개SW는 개발 시부터 공개되어 이미 많은 취약점이 개선 및 해결되었으며 안정된 상태에서 운영되고 있다.

- 넷째, 기술적인 면에서의 차이점이다.
 - 비공개SW는 프로젝트의 연속성 및 재사용성이 낮은 반면 공개SW는 소스코드의 공개로 인하여 언제든지 재사용이 가능하기 때문에 프로젝트의 연속성이 보장되며 유지보수 및 업그레이드 비용이 낮다.
- 다섯째, 확장성에 대한 차이점이다.
 - 비공개SW는 높은 적용비용과 제한된 시스템 운영환경으로 인하여 확장성이 완벽하게 보장되는 것은 아니다. 하지만 공개SW는 소스가 공개되어 있기 때문에 시스템의 확장성이 확실히 보장된다는 장점이 있다.
- 여섯째, 공급권에 대한 차이점이다.
 - 비공개SW의 생산업체는 하나이므로 독점 및 단일공급 된다. 단, 유통회사는 여러 개 존재할 수 있지만, 원 공급 업체는 하나이다. 반면 공개SW는 동일한 솔루션에 대하여 다수 업체들로 부터 공급이 가능하기 때문에 사용자의 공급업체 선택권이 넓은 편이다.
- 일곱째, 저작권에 대한 차이점이다.
 - 비공개SW는 일반 라이선스를 채택하고 있는 반면 공개SW는 거의 대부분 GPL 라이선스, BSD 라이선스, MPL 라이선스 등을 원하는 대로 적용할 수 있다.
- 여덟째, 경쟁력에 대한 차이점이다.
 - 비공개SW는 핵심 소프트웨어에 대한 개발능력 및 저작권을 이미 보유하고 있는 소프트웨어 선진국에 유리한 소프트웨어이다. 반면 공개SW는 소프트웨어의 핵심기술 공유할 수 있는 기회가 있기 때문에 신흥 소프트웨어 개발국에 유리하다.

■ 비공개SW와 공개SW의 차이점을 요약하면 다음 표와 같다.

<표 7> 비공개SW와 공개SW 비교표

구 분	비공개소프트웨어	공개소프트웨어
비용분석	<ul style="list-style-type: none"> - 적용비용이 높음 - 유지비용및 시스템 개선비용이 높음 - 소수의 관리자에 대한 관리비용 높음 - 라이선스에 대한 비용발생 	<ul style="list-style-type: none"> - 적용비용이 낮음 - 유지비용이 낮고 기능 확장에 대한 추가비용이 낮음 - 관리비용이 낮음 - 서비스에 대한 비용발생
성능분석	<ul style="list-style-type: none"> - 규모가 큰 시스템 환경에서 비교적 높은 성능 - 고가의 장비로 인한 고성능 - 전체적으로 공개SW와 비슷한 성능 	<ul style="list-style-type: none"> - 다양한 환경에 최적화된 설정으로 높은 성능 - 높은 안정성 및 비용효율이 높음
보안성	<ul style="list-style-type: none"> - 폐쇄적인 운영으로 인한 공개되지 않은 시스템 취약점 보유 - 최근에 다수의 취약점 발견으로 보안 위협 노출 - 프로토콜 호환이 어려워 인증체계 취약 	<ul style="list-style-type: none"> - 개발시부터 공개되어 이미 많은 취약점이 해결된 안전화 상태 - 공개키 기반의 인증 매카니즘 구현을 위한 통합패키지존재 - 다양한 암호화 알고리즘 및 키 관리에 대한 기능 제공
기술성	<ul style="list-style-type: none"> - 재사용성 없음 	<ul style="list-style-type: none"> - 재사용성 높음 - 유지보수,업그레이드용이 - 독점폐해방지
저작권	<ul style="list-style-type: none"> - 일반라이선스 - 독과점에 의한 가격결정우려 	<ul style="list-style-type: none"> - GPL라이선스 - BSD라이선스등
확장성	<ul style="list-style-type: none"> - 서버의 가용성 측면에서 클러스터링 비효율성 - 소프트웨어간의 호환성이 보장되나 높은 적용비용과 제한된 시스템 운영환경 	<ul style="list-style-type: none"> - 효율적인 클러스터링 구현가능 - 소프트웨어간의 적용비용이 거의 들지 않고 낮은 수준에서의 기능 추가 가능
경쟁력	<ul style="list-style-type: none"> - 소프트웨어선진국에 유리 	<ul style="list-style-type: none"> - 공동개발상식에 따른 기술공유 효과 우수(교육효과 우수) - 신흥 소프트웨어 개발국에 적합
공급권	<ul style="list-style-type: none"> - 개발업체로 부터 단일공급 	<ul style="list-style-type: none"> - 다수 업체로 부터 공급가능

3) 기존 연구를 통해 추천한 공개SW의 목록 비교

- 아래 표는 정통부에서 추천한 36종의 공개SW와 국방부에서 추천한 공개SW 21종을 부문별로 비교하여 나타낸 것이다.
- 정통부와 국방부에서 추천한 공개SW 목록을 비교하면 중요한 부문에 해당하는 배포판, 웹서버, office 및 브라우저 등은 동일한 공개SW를 추천하고 있는데, 국방부에서는 국방업무의 특수성을 고려하여 데스크 탑에서 꼭 필요한 유틸리티 만을 추천하였으며, 웹서버의 개발에 필요한 Tomcat, JBoss 및 JSP의 공개SW를 추가하였다.

<표 8> 정통부와 국방부의 추천 공개SW 목록 비교표

구분		정통부 추천(36종)	국방부 추천(21종)
배포판		Linux	Linux
서버	web	Apache	Apache
	DBMS	MySQL	MySQL
	DNS	X	X
	FTP	Proftpd	Proftpd
	Mail	Sendmail	Sendmail
	SSH	Openssh	Openssh
	Proxy	Tinyproxy	X
	File	Samba	X
미들웨어	WAS	X	Tomcat, JBoss
Desktop	office	Open office	Open office
	브라우저	Mozilla Firefox	Mozilla Firefox
	유틸리티	Ximin Evolution, Downloading Client, Gnozip, X CD Roast, Xpdf, Xchat, Gnutella, Nautilus, GNUPG, Gaim, Xmms, Mplayer, GQview 등	Gnome, gftp, Gnozip, GNUPG, Nautilus, MRTG
	PC보안툴	Firewall-Easy	Firewall-Easy
개발	언어	PHP	PHP, JSP
	개발툴	KDevelop	KDevelop
		Kylix	Kylix
library	wxBasic	X	

나. 국방전력발전 규정과 상호운용성

1) 국방전력발전 규정의 발전 개요

- 우리나라 국방시스템의 획득은 국방획득관리규정(국방부 훈령 제733호, 2003년 5월 13일 개정)에 의해 이루어지는데, 이 규정은 아래와 같은 5가지의 획득관리 원칙과 1999년 1월 제정한 이래로 7번의 개정을 통해 수정되어왔다.

■국방획득관리 원칙

- 성 능 보 장 : 작전운용성능을 충족시킬 수 있는 장비획득
- 적기 전력화 : 요구되는 시기에 전력화 가능
- 국산화 촉진 : 국방과학기술에 의해 ‘자주국방 달성’이 가능토록 연구개발 및 국내생산 우선 추진
- 경제적 획득 : 성능이 보장될 수 있는 장비·물자를 경제적으로 획득하여 투자효율 극대화
- 운영유지 보장 : 수명주기간 효율적인 운영유지 보장

■국방획득관리 규정 개정 현황

- 제 정 1999. 1. 2. 국방부훈령 제610호
- 개 정 1999. 4. 3. 국방부훈령 제621호
- 개 정 1999. 6. 1. 국방부훈령 제631호
- 전문개정 1999.12.31. 국방부훈령 제651호
- 전문개정 2000.12.30. 국방부훈령 제676호
- 전문개정 2002. 1.10. 국방부훈령 제699호
- 전문개정 2003. 2. 1. 국방부훈령 제727호
- 개 정 2003. 5.13 . 국방부훈령 제733호

- 그러나 최근 방위사업청의 신설과 국방 환경변화 등 국방획득제도를 획기적으로 개선해야할 필요성이 제기됨에 따라 국방부에서는 국방 전력증강업무체계 전반을 재정립하여 구(舊) 국방획득관리규정을 대체하는 『국방전력발전업무규정』을 제정하여 “국방부훈령 793호”로 발령하게 되었음.

- 이 규정의 성격과 범위는 방위사업법 및 동법 시행령·규칙에 맞게 국방전력발전업무절차를 규정하고 지침을 제공하도록 하였으며, 무기체계 및 비무기체계의 소요기획, 획득관리, 운영유지관리, 정보체계관리, 시설사업관리, 분석평가 등을 포괄함.
- 이 규정은 본문 10장 398조, 부칙 3조로 구성되었으며, 주요 내용은 아래와 같다.
 - 방위력개선분야 중기계획은 방사청이 작성하여 국방부 검토결과를 받아 장관에게 보고하고, 국방부가 이를 위원회에 상정하여 의결토록 함.
 - 중기계획 초년도 착수사업 중 1,000억원 이상사업은 장관보고, 5,000억원 이상사업 및 국가정책사업은 대통령보고
 - 국방부는 예산편성지침을 하달하고, 지침 부합 여부 등 검토 후 장관보고토록 함.
 - 방사청이 창정비요소 등 전력화지원요소 확보방침 및 계획(안)을 작성하고 소요군의 의견을 반영하여 확정
 - 방사청은 시설 및 정보통신공사를 소요군의 공사집행기관을 통하여 집행할 수 있도록 절차 마련
 - 전력화 후 운영유지과정에서 나타나는 제반 문제점에 대해 국방부가 진단·조치토록 절차 마련
 - 무기체계 국산화는 방사청이, 운영중인 군수품의 부품 국산화는 국방부 군수관리관실이 수행토록 절차 마련
 - 무기체계를 포함하는 모든 정보화사업은 상호운용성 확보 등을 위해 “국방아키텍처”내에서 추진토록 의무화
 - 본 규정을 제정함에 따라 방위사업법 및 시행령/시행규칙에서 다루지 못한 국방부/합참/각군/방사청 및 기관 간에 업무분장 및 세부 업무절차/기준이 정립되었고, 특히 방사청 개청에도 불구하고 소요기획, 획득관리 및 운영유지업무 간에 일관성과 연계성을 유지하는데 크게 기여할 것으로 예상됨.

2) 개정된 국방전력발전 규정의 주요 내용

- 새로운 규정에서는 시대의 변화에 따라 무기체계와 비무기체계를 새롭게 정의하였다.
 - 무기체계는 유도무기, 항공기, 함정 등 전장에서 전투력 발휘하기 위한 무기와 이를 운영하는데 필요한 장비·부품·시설·소프트웨어 등 제반요소를 통합한 것을 말한다.
 - 비무기체계는 무기체계 외의 장비·부품·시설·소프트웨어, 그 밖의 물품 등 제반요소를 말한다.
-
- 무기체계의 분류기준은 대분류, 중분류 및 소분류로 구분하며 무기체계 대분류는 지휘통제·통신무기체계, 감시·정찰무기체계, 기동무기체계, 함정무기체계, 항공무기체계, 화력무기체계, 방호무기체계, 기타 무기체계로 분류하고, 비무기체계는 일반군수품, 자동화정보체계, 교육훈련용 장비 및 물자, 기타 일반시설로 분류한다.

<표 9> 기존 및 새로운 규정에서의 무기체계 및 비무기체계 분류표

구분	내용		비고
국방획득 관리규정 (‘03. 5. 13)	무기체계	- 지휘 및 통제, 기동, 화력 항공기, 유도·방공 - 통신·전자, 정보, 전자전, 화생방, 기타 무기체계	폐기
	비무기체계	- 주요 비무기체계, 자동화 정보체계, 기타 비무기체계	
국방전력 발전규정 (‘06. 6. 29)	무기체계	- 유도무기, 항공기, 함정 - 무기체계를 운영하는 필요한 장비, 부품, 시설, SW 등	
	비무기체계	- 무기체계 외의 장비, 부품, 시설, SW 등	

- 무기체계와 비무기체계의 획득절차는 아래 그림과 같이 방사청/소요군에 의한 소요결정 및 사업추진방법을 결정한 후 해당하는 획득절차에 의거 추진되도록 규정되어있다.

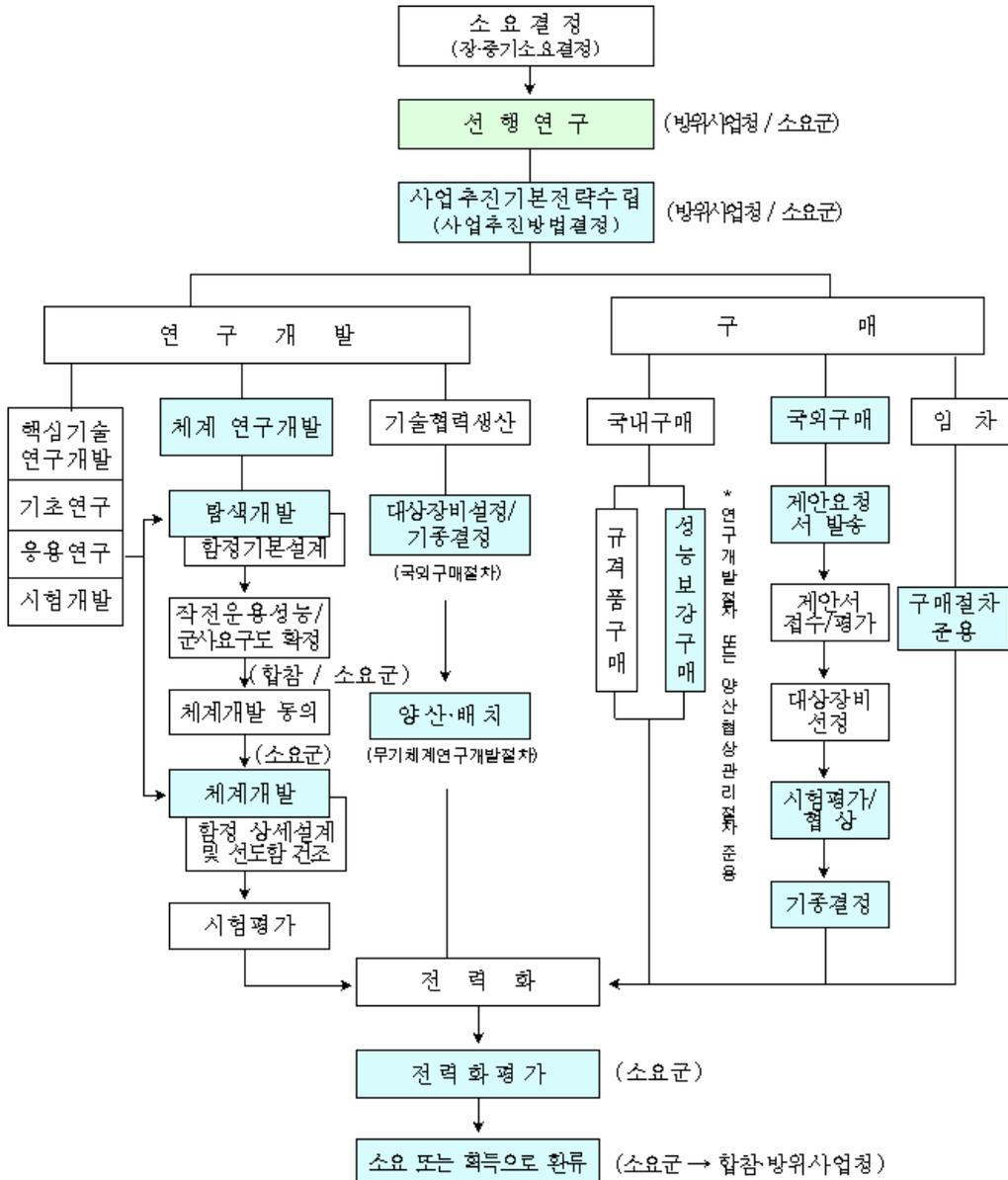


그림 20. 무기체계 및 비무기체계의 획득 세부 절차도

3) 상호운용성 확보를 위하여 개정된 주요 내용

- 새로운 규정에서는 미래지향적인 NCW의 전장환경 하에서 국방정보 체계의 상호운용성의 확보가 무엇보다도 중요하다는 것을 인식하여 상호운용성 확보를 위하여 아래와 같은 조항을 포함하였다.
- 제 284조 (상호운용성 적용범위 및 원칙)
 - 정보화기획관실은 국방정보체계간의 상호운용성 보장에 필요한 제반 사항을 관장하며, 무기체계와 관련된 상호운용성 요구사항이 변경될 시에는 반드시 합참 등 관계부서의 검토의견을 반영한다.
 - 체계개발간 상호운용성 및 표준화 관련 절차와 기반환경은 체계개발 기관에 동일하게 적용하며, 정보화기획관실에서 총괄 구축하여 공동으로 활용한다.
 - 체계개발기관은 체계간 상호운용성 보장에 필요한 표준기술 및 핵심 기술 연구소요 발생시 정보화기획관실로 소요 제기하여 정보화정책서에 반영시켜야 하고, 공통 기능 재사용성을 높이기 위하여 개발 전 과정에 걸쳐 상호운용성 보장 활동을 지속적으로 수행하여야 한다.
- 제 285조 (국방정보체계 표준기술 준수)
 - 체계개발시 상호운용성 보장을 위하여 국방정보체계기술구조를 국방 표준기술로 정하며 체계개발기관은 이를 준수한다. 다만, 체계 특성상 국방 표준기술 이외의 기술이 필요할 경우 국방 표준기술로 소요를 제기하여 정보화기획관실로부터 사전 승인을 받아야 한다.
 - 정보화기획관실은 정보기술 발전추세를 고려하여 국방 표준기술을 최신 정보들로 관리하고 국방정보체계 구축 사업 관련기관에 제공하여야 한다.
- 제 286조 (공통운용환경 적용)
 - 정보화기획관실은 국방정보체계의 상호운용성 달성을 위하여 재사용 가능한 공통운용환경을 구축한다.
 - 체계개발시 공통운용환경의 주요기능을 적용하며 신규 소요 발생시는 상용제품을 우선 사용하고 없으면 자체 개발하되 국방부에 등록하여 공동 활용토록 한다.

- 공통운영환경 구조는 개방형 표준을 기반으로 도메인별로 구축하되 특정 하드웨어에 독립적이며 특정 프로그램을 선호하지 않는다.
- 제 287조 (데이터공유환경 적용)
 - 정보화기획관실은 상호운영성 달성을 위하여 재사용 가능한 핵심 데이터를 식별하여 데이터간 호환성이 확보되도록 표준화를 통한 데이터공유환경을 구축한다.
 - 체계개발기관은 국방 데이터공유환경에서 제공하는 표준 데이터를 적용하고 신규 소요 발생시는 표준 데이터를 자체 제정하되 새로 제정한 표준 데이터는 국방부에 등록하여 공동 활용토록 한다.
- 제 288조 (국방 아키텍처 프레임워크 적용)
 - 정보화기획관실은 아키텍처 기반으로 국방정보체계를 개발할 수 있도록 국방 아키텍처 프레임워크 정의와 아키텍처 정보관리저장소를 구축하여 관리한다.
 - 체계개발 시 국방 아키텍처 프레임워크를 적용하여 소요제기, 개념연구, 체계개발(시스템/서비스시스템설계정의서) 단계별로 아키텍처를 필수 및 보조 산출물로 구분하여 설계한다.
 - 체계개발 시 설계한 아키텍처 정보는 정보관리저장소에 수록하여 정보화기획관실로 제출한다.
 - 체계개발 시 아키텍처 설계는 아래 항목 중 1개 이상 해당되는 사업에 적용하는 것을 원칙으로 하되, 필요시 정보화기획관실 사전 승인 후 아키텍처 설계 작업을 생략할 수 있다.
 - 전군차원(2개 군 또는 2개 작전사급 이상 제대)에서 운용되는 체계
 - 전군차원에서 운용되는 체계가 2개 이상 연동되는 체계
 - 사업예산 규모가 100억원 이상되는 체계
 - 국방정보체계 획득절차에 의거 사업기간이 1년 이상되는 체계
- 제 289조 (상호운영성 수준평가 및 능력시험 체계 적용)
 - 정보화기획관실은 체계개발시 상호운영성 시험평가?인증기관을 통하여 상호운영성 수준 달성 여부를 평가한다.
 - 체계개발기관은 개념연구, 체계분석 및 설계, 체계개발 단계에서 상

호운용성 수준을 평가하여 정보화기획관실로 보고한다. 이때 평가체계는 국방부 상호운용성 수준평가 및 능력시험 체계를 적용한다.

- 제 290조 (기타) 체계개발 시 체계간 상호운용성 달성을 위하여 제4관의 내용이행에 필요한 세부사항은 국방정보체계 상호운용성 및 표준화 관리지침을 따른다.

다. 국방시스템 관련 연구를 위한 공개SW의 체계 분류

1) 국방정보화 체계

- 우리나라의 국방정보화 체계 구성은 아래 그림과 같이 정보화의 인프라에 해당하는 기반체계와 응용에 따라 구분되는 전장관리 체계와 자원관리체계로 구분이 된다.

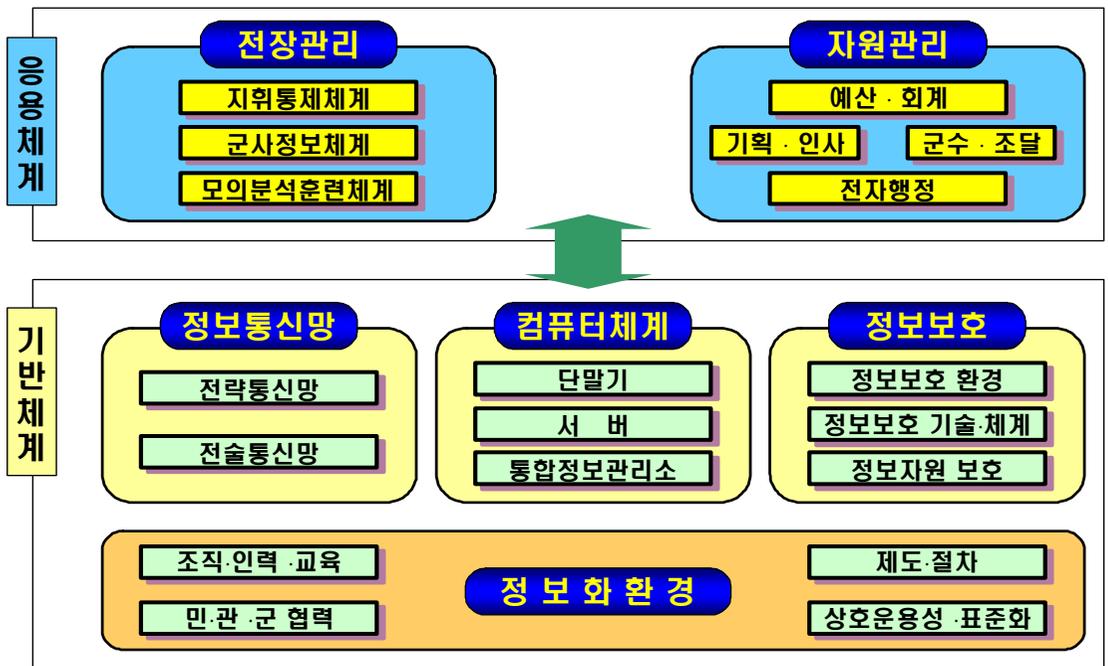


그림 21. 우리나라의 국방정보화 체계도

- 또한 체계 분류별로 담당하는 세부 정보체계는 다음 표와 같다.

<표 10> 체계 분류별 세부 정보체계

2) 일반 정보시스템에서의 공개SW 체계 분류

- 시대의 흐름에 따라 정보시스템의 체계구분도 달라지지만, 일반적인 정보시스템에서는 하드웨어의 구성형태에 따라 서버, 데스크탑, 모바일로 구분하고, 하드웨어를 동작시키는 SW는 응용형태에 따라 시스템SW, 유틸리티, 응용SW로 분류할 수 있으며, 이러한 체계로 분류한 상태에서 관련된 공개SW를 구분하면 다음 표와같이 된다.

<표 11> 일반적인 공개SW 분류 표

구분		관련 공개SW
서버	* 웹서버, 응용서버, DB서버 * 웹서버, 클라이언트/서버, 스탠드올원	시스템 SW - 리눅스 - Apache - MySQL - Proftpd
		유틸리티 - Tomcat - JBoss - Sendmail 등
		응용SW - 각종 응용프로그램
데스크 탑	PC, 노트북	시스템 SW - 리눅스 배포판 - Open office - Mozilla Firefox
		유틸리티 - Gnome - GIMP 등
		응용SW - 각종 응용프로그램
모바일	임베디드, 리얼타임	시스템SW - 임베디드리눅스 - 리얼타임 임베디드리눅스
		유틸리티 - 임베디드용 각종 유틸리티
		응용SW - 각종 응용프로그램

3) 국방시스템의 체계 분류

- 그러나 국방시스템은 일반적 정보시스템과 달리 무기체계, 비무기체계 또는 전장관리, 자원관리 등 일반적 환경과는 다른 요소가 많이 있기 때문에 정책의 선택을 위한 국방시스템의 체계분류가 필요하다고, 판단되어 국방의 현실적 환경을 고려하여 아래와 같은 체계의 분류를 제안하고자 한다.

<표 12> 국방시스템에서의 체계 분류 표

구분		하드웨어 분류	관련 SW 및 정책	
			SW	정책
무기 체계	지휘통제 감시정찰 기동 무기 함정 무기 항공 무기 등	무기체계용 시스템	전장정보 SW	개방형 기술개발
		무기체계용 임베디드시스템	무기체계용 임베디드 SW	
	군사 정보 모의 훈련 등	서버	서버용 SW	공개SW 활성화
비무기 체계	자원관리	데스크탑	데스크탑용 SW	
	기반체계	정보통신체계	비무기체계용 시스템	비무기체계용 SW
		컴퓨터체계		
		정보보호체계 상호운용성 기반체계	비무기체계용 임베디드시스템	비무기체계용 임베디드 SW

라. 우리나라 국방시스템에서의 공개SW 활용현황

1) 공개SW활용 현황 개요

- 현재 우리나라 국방시스템의 무기체계용 시스템에서 공개SW를 활용하고 있는 사례는 거의 없는 실정으로 파악되고 있다.
- 지휘통제(C4I)시스템의 전장정보SW의 개발은 단위 기능별로 여러업체가 협동으로 개발되었기 때문에 상용SW로 분류가 되는데, 다만 개발을 위한 관련 유틸리티로서 공개SW를 활용했을 가능성은 있다.
- 기동무기, 함정무기 등에서 사용하는 실시간 임베디드리눅스의 경우에도 이 분야의 전문가인 국방대학교의 윤희병교수 조사한 자료에 의하면 현재로는 거의 없는 실정이나 비공식적인 설문조사에 의하면
 - 공개SW의 적용은 바람직하고, 정책이 정해지면 군 특성상 단기간에 활성화 될 것으로 판단된다는 긍정적 의견도 있는 반면에
 - 상용에 비해 HW 환경이 특수하여 제약적 부분 많이 상존하고, 공개SW의 적용시 소스코드 수정 및 이를 위한 공개SW 분석이 필요하므로 많은 인력과 시간 투입 불가피하며, 또한 수정SW에 대한 신뢰성 및 책임부담 가중으로 사용이 곤란하다는 의견도 있었음
- 비무기체계의 자원관리체계에서는 수백종의 단위업무시스템이 개발되어 운영되고 있는데
 - 구축된 형태별로 분류하면 웹서버형, 클라이언트-서버형, 스탠드얼론형으로 구분되고
 - 상당수의 시스템들에 리눅스를 포함한 공개SW가 활용되고 있으나
 - 대부분 특정 HW(OS)에 종속된 SW 선택으로 수직계열화되고, 서버 OS에 따라 올라가는 제품군(DB, WAS, WEB)이 동일 계열로 스택화됨으로 HW(OS) 종속화 및 비용증감 요인으로 작용하고 있으며, 일반 상용SW 환경에 익숙한 운영자들이 새로운 환경(공개SW)의 적용에 소극적인 경향이 있는 상황임
- 비무기체계의 기반체계내 정보통신시스템에서는 공개SW는 아니지만 우리 독자기술로 개발되어 세계표준으로 인정된 와이브로기술을 적용하고자 추진 중임

2) 무기체계에서의 현황

가) 무기체계용 시스템의 전반적 현황

- 무기체계용 시스템에는 앞에서 분류한 바와 같이 지휘통제, 감시정찰, 기동, 함정 등의 무기와 관련된 시스템을 의미하는데, 현대전에서는 단순한 병력이나 화력의 규모보다는 전투에 직간접으로 참여하는 각급 부대 및 전투원이 공유하는 정보에 의해 승패가 갈리는 상황이므로 각각의 무기들과 기반체계들이 지휘통제시스템을 중심으로 네트워크화 되어가고있고, 이러한 것을 전장정보시스템이라고 한다.
- 전장 정보시스템은 실시간 지휘통제 정보를 단위부대의 전투, 전투지원 및 전투 근무 지원부대 지휘자 및 병사까지 제공하는 디지털 전투지휘 정보체계이다. 전장 전역에 걸친 전투 지휘 정보를 실시간으로 제공하며 상황인식 및 지휘통제 기능을 전술적 최하위 단위까지 제공한다. 이 시스템은 전투 명령 및 정보를 전장에서 물 흐르듯이 상급 부대 지휘관으로부터 하급 병사에게로 전달하며 전술C4I시스템과의 정보교환을 통하여 지휘통제 명령 전달, 센서 시스템으로부터의 수집 정보 획득, 수집된 정보의 상위 보고 등의 역할을 전술 인터넷 및 상위 네트워크를 통하여 수행한다. 아래 그림은 여단 급 부대의 구성 시 필요한 전장정보 시스템과 전체 지휘통제시스템(C4ISR)시스템과의 역할과 통합 관계를 나타낸다

그림 22. 전장정보시스템과 C4I의 관계도

- 전장정보시스템은 지휘관으로 하여금 간단한 조작으로 빠른 상황 판단과 명령 전달을 가능케 하여 지휘관의 임무 수행을 향상시키고, 다음의 구성요소로 되어 있다.
 - 휴대용 컴퓨터 또는 내장형 시스템 하드웨어(컴퓨팅 장비)
 - 단위부대를 위한 해당 전장정보 소프트웨어
 - 위치정보 및 이동경로 추적을 할 수 있는 장비(Global Positioning System)
 - 전술인터넷에 연결 가능한 장비 및 인터페이스 또는 위성통신이 가능한 장비
 - 전투시 아군 식별장치 (Battlefield Combat Identification System)

그림 23. 전장정보시스템의 개념도

나) 무기체계용 SW의 현황

- 무기체계용 SW에는 무기체계시스템 중 지휘통제 시스템에서 필요한 전장정보SW, 감시정찰, 기동, 함정 무기 등에 필요한 SW들이 있지만, 여기서는 전장정보SW를 의미한다.
- 전장정보SW는 네트워크 설정과 같은 컴퓨터의 기본 기능 관리, 위치 보고 및 아군 위치 도식과 같은 정보 전달 기능, 지휘 통제를 위한 명령 하달 및 보고, 정보의 확산 등의 전술 정보교환 기능, 작전 준비 및 계획 수립을 돕는 의사 결정 보조 기능, 이동 및 군수 보급을 담당하는 군수 기능 등의 기능을 수행할 수 있는 소프트웨어 모듈의 집합으로서, 아래와 같이 5가지의 기능을 수행하여야 한다.
- 기본기능 (Digital Basics)
 - 네트워크 구성 및 관리기능, 장비 유지 및 기록관리, 사용자 확인 및 보안 기능
- 전장정보 전달기능 (Battlefield Visualization)
 - 전장공통상황 도식기능, 맞춤형 상황인식/도식, 장애물 및 지형 도식, C2 메세지 전달
- 임무기획/준비(Mission Planning & Preparation)
 - 지형분석을 위한 LOS도구 제공, 지역방위계획수립을 위한 LOS도구 제공
 - 화력 통제 및 계획 도구, 군수공급 체계 지원 기능, COP 및 상황 판단을 위한 정보 준비
- 전술정보 교환(Information Exchange)
 - 전술정보 수령 및 관리, 전술정보의 전송 및 확산, 중요 명령의 수령 확인
- 이동성 및 기동성 향상 기능(Mobility & Maneuver)
 - 이동 시 신속성과 정확성 증대, 아군의 공격력 및 생존율 향상을 위한 기동성 제공
 - 이동 시 아군간의 충돌 방지

- 전장정보SW의 구조는 하드웨어 플랫폼 위에 아래 그림과 같이 3개의 계층으로 구성된 SW 플랫폼과 이를 바탕으로 수행되는 응용 소프트웨어로 구성할 수 있다. 아래그림에서의 하드웨어 플랫폼은 무기 체계의 내장형 시스템 또는 병사가 소지하는 휴대용 컴퓨터를 의미한다. 소프트웨어는 국방 공통운영환경 개념 (DII COE concept)에 준하여 3개의 계층으로 구성되고 임무 중심의 응용프로그램이 이를 바탕으로 수행된다. 임무 중심 응용프로그램의 예로는 전투지휘 시스템(battle command), 전투 지원통제 시스템(combat services support control system), 야전포병 전술데이터 시스템(advanced field artillery tactical data system) 등이 있다. 이 응용 프로그램은 그 용도에 따른 개별적 기능이 있으며 각 사용자의 요구에 맞게 개발된 시스템이다. 여기서는 전투지휘 시스템을 예로 살펴 본다.

그림 24. 전투지휘 시스템의 사례도

- 국방 공통운영환경은 커널 (DII COE Kernel), 기반구조 서비스 (Common Infra-structure Services), 공통 지원 응용(Common Support Applications) 계층으로 구성된다. 공통지원응용 계층은 각 단계에서 사용되는 데이터의 공통된 시각(view)을 통하여 상호 운용성을 유지하도록 하는 계층이다. 이 계층은 특정 임무 영역을 위해 공통적으로 지원되는 기능들을 제공하는 계층으로 메시지 서비스, HCI, 군수, 작전계획 수립 및 준비, 상황인식과 같은 기능을 제공하는 모듈로 구성된다. 기반구조서비스 계층은 네트워크를 통한 데이터의 송신과 서버로부터 데이터 수신을 제공하기 위한 계층으로 각 응용 프로그램과는 독립적이며 전 국방 소프트웨어에서 공통적인 기능을 갖는다. 본 계층에서는 정보관리 서비스와 통신서비스를 제공한다. 커널 계층은 국방 공통운영 환경을 제공하고 모든 워크스테이션이나 내장형 시스템 또는 휴대용 단말기에 있는 운영 시스템이다. 이 계층은 상위 계층들이 하드웨어 플랫폼에 독립적으로 운용될 수 있도록 하는 기능을 제공하며, 기능적으로 보안 서비스, 네트워크 관리 서비스, 시스템 관리리 서비스를 제공한다.
- 우리나라의 지휘통신시스템은 지휘·통제·통신·컴퓨터의 유기적인 통합으로 실시간에 정보를 공유함으로써, 효율적인 감시 - 결심 · 작전수행(타격)을 보장하는 지휘통제체계를 위하여 2016년까지 군단 급이하 전술제대에 “先見, 先決, 先打”의 전투수행을 보장하는 전술 C4I체계 구축목표로 추진하였다.

그림 25. 우리나라 C4I체계의 구축 목적 및 수단

- 추진경과는 다음과 같다.
 - 1차 개념연구 / 국방부 승인 : '96. 4 / '97. 8
 - 개념 재연구(IMF로 사업순연) : '99. 7 ~ '00. 2
 - 국방부 사업추진 지침 : '00. 6
 - 체계개발은 진화적 개발
 - 단계화 사업추진
 - 핵심기능 위주로 응용SW 단순화
 - ROC / 체계규격서 수정 : '00. 9
 - 사업 착수 / 추진 (단계화) : '00. 12 ~ '05. 4
 - 개발 종료 / 전력화 추진 : '05. 7.26
- 다음 그림은 군단 ~ 연대급부대의 지휘통제본부별 구축 배치 사례를 보여주고 있다.

그림 26. 군단~연대급 부대의 C4I 배치 사례도

다) 무기체계용 임베디드 SW의 현황

- 임베디드 시스템이란 어떤 제품이나 솔루션에 추가로 탑재되어 그 제품 안에서 특정한 작업을 수행하도록 하는 솔루션을 말한다. 예를 들어 주된 용도가 전화인 휴대폰에 텔레비전 기능이 들어가 있다면, 텔레비전 기능(시스템)이 바로 임베디드시스템이다. 곧, 본 시스템에 끼워넣은 시스템이라는 뜻이다. 첨단 기능이 들어 있는 컴퓨터, 가전 제품, 공장자동화 시스템, 엘리베이터, 휴대폰 등 현대의 각종 전자·정보·통신 기기는 대부분 임베디드시스템을 갖추고 있다. 대개의 경우 그 자체로 작동할 수도 있지만, 다른 제품과 결합해 부수적인 기능을 수행할 때에 한해 임베디드시스템이라고 한다. 컴퓨터의 경우에는 전용 동작을 수행하거나 특정 임베디드 소프트웨어 응용 프로그램과 함께 사용되도록 디자인된 특정 컴퓨터 시스템 또는 컴퓨팅 장치를 일컫는다. 일반적으로 범용 컴퓨터 또는 장치를 상업적으로 대체해 사용할 수는 없기 때문에, 특정한 임베디드 라이선스 제품에서 허용되는 사무자동화 및 개인 컴퓨터 기능은 사용이 제한된다. 그러나 임베디드 리눅스와 같은 프로그램은 일반 퍼스널컴퓨터와도 연동이 가능하다. 컴퓨터 외에 휴대용 개인정보 단말기(PDA)에 들어 있는 음성솔루션, 텔레비전·전기밥솥·냉장고·자동차 등에 내장되어 있는 웹기능 등도 모두 임베디드시스템이다.

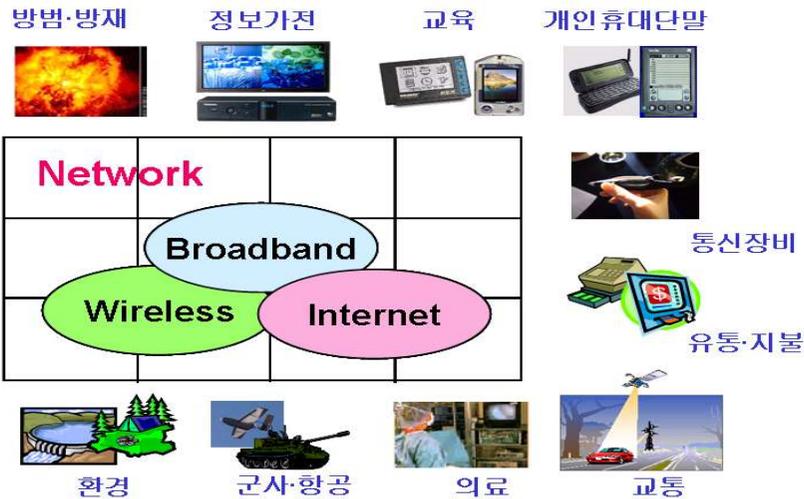


그림 27. 임베디드 시스템의 개념도

- 임베디드SW는 임베디드 시스템을 동작시키도록 하는 핵심SW로서, Anytime, Anywhere, Anynetwork, Anydevice, Anyservice를 지향하는 유비쿼터스 사회에서 인간 생활을 둘러싼 모든 환경과 사물 속에 특정 용도와 기능 중심의 컴퓨터를 센서, 칩, 배지, 마이크로 머신 로봇 등과 같은 포스트 PC 형태로 내장 시키고 이들을 유무선 네트워크로 연결함으로써 보이지 않은 컴퓨터를 통해 사람,장소,사물의 시시각각 변하는 상황정보를 활용할 수 있는 미래 정보화의 핵심으로 부상하고 있다.
- 국방에서의 임베디드 SW는 각종 무기/비무기체계에 내장되어 해당 장비의 임무에 전용으로 제공되는 소프트웨어로 정의되고 있고, 현대 무기 체계 내에서 소프트웨어(SW)가 차지하는 비중은 지속적으로 증가하고 있다. 특히 SW는 무기체계의 성능과 질을 결정하는 핵심 요소로 부각되고 있는데, 아래표와 같이 임베디드 SW 비중이 1960년 대 전투기인 F-4는 8%에 불과했으나 차세대 전투기인 F-22의 경우 전체 기능 중 80%를 차지하고 있을 정도이다.

<표 13> 무기체계에서의 년도별 SW 점유율 비교표

무기체계	연도	점유율(%)	무기체계	연도	SW 라인수
F-4	1960	8	A-7	1960	1만
A-7	1964	10	F-111	1964	5만
F-111	1970	20	AWACS	1970	60만
F-15	1975	35	F-16	1975	100만
F-16	1982	45	F-15E	1982	500만
B-2	1990	65	B-2	1990	1000만
F-22	2000	80	F-22	2000	1500만



미 육군
2천라인

※ 출처 : 국방대, 윤희병 교수

- 미래네트워크 중심전(NCW)에서 임베디드 SW 역할은 전쟁 승패를 결정짓는 핵심요소로서, 무인기·정찰기 등과 같은 무기체계에 탑재된 임베디드 시스템에서 그 시스템의 자원을 관리하고 요구 기능을 효율적으로 수행하는 것이 임베디드 운영체제다. 따라서 임베디드 SW는 무기체계 내에서 가장 핵심적인 역할을 맡고 있다. 다양한 유형의 임베디드 시스템은 그 특성과 목적에 따라 서로 다른 종류의 운영체제를 요구하지만 기본적으로 갖춰야 할 요소가 있다. 우선 실시간 처리를 지원해야 한다. 실시간 처리가 지원되지 않을 경우 비행제어 시스템·항법 시스템 등에서 큰 사고를 야기할 수 있다. 둘째, 항공기와 미사일 제어에 사용된 임베디드 SW의 오작동은 심각한 결과를 초래할 수 있기 때문에 고도의 신뢰성을 요한다. 셋째, 오작동 시 막대한 피해가 예상되는 무기체계에서는 시스템 동작 응답 시간이나 그 영향이 예측 가능해야 한다. 이밖에 조립성(configurability)·적응성(adaptability) 등의 특징을 갖고 있다.
- 육군 무기체계에서 임베디드 SW와 관련한 현황은 아래 표와 같다.

<표 14> 육군 무기체계에서 임베디드 SW 관련 현황

구분	K1A1전차, K-55자주포	차기 전차	BTCS A1
개발언어	C, Assembly	C/C++, Assembly	C/C++ PLM80/86
개발방법론	구조적	구조적	구조적, 객체지향
개발도구	C Editor/Compiler, Assembler, 에뮬레이터	Visual Studio	Visual C++ .NET, Tornado 2.2, Intel PL/M-86, MetaWare High C/C++
사용OS	QNX, VxWorks, 자체개발	VxWorks	VxWorks 5.5
관리도구	-	와이즈맨 (ADD 개발)	Clearcase, Clearquest, Rational Rose

- 해군 무기체계에서 임베디드 SW와 관련한 사용현황은 다음과 같다.

<표 15> 해군 무기체계에서 임베디드 SW 관련 현황

구분	KDX- I & II	LPX	KDX-III	PKX	함대함 유도탄	청상어
개발 언어	Ada83	Ada95, C/C++, Java	C/C++	C/C++	C++	C, Assembly
개발 방법론	구조적, Waterfall, BAe사 주기	MIL-STD-498, 구조적, 객체지향	MIL-STD-498, 객체지향	MIL-STD-498, Waterfall, Prototyping	객체지향	구조적
개발 도구	BAe Sys' Toolset, TeleGen2, Intel SBC	ROSE, Visual Age, Java Twist HCI Toolkit	Tornado IDE, Visual Compiler	SA2001, Visual C++ for Tornado	Visual Studio 6.0	TI C, Diab C, MPASM, Delphi
사용 OS	TeleAdaExec	VxWorks	VxWorks	VxWorks	-	-
관리 도구	ETM, BAe SLIC, OR/CR	DOORS, SCMS	DOORS, Clearcase, Clearquest	DOORS, Clearcase	Clearcase, Clearquest, Rational Rose	Clearcase, Clearquest, Rational Rose

- 공군 무기체계에서 임베디드 SW와 관련한 사용현황은 다음과 같다.

<표 16> 공군 무기체계에서 임베디드 SW 관련 현황

구분	KF-16	T-50	ALQ-200
개발언어	Jovial, Assembly	C/C++	C/C++
개발방법론	구조적	구조적 (incremental)	구조적
개발도구	-	Tornado	Metaware Compiler/Linker
사용OS	-	VxWorks	-
관리도구	VAX기반 CMS (미활용)	PVCS	Clearcase, Clearquest, Rational Rose

- 위와 같이 육해공군의 임베디드 SW와 관련한 사용현황을 정리하면 다음과 같다.
 - 육·해·공군 모두 개발방법론에 적합한 언어인 Assembly, Ada, C/C++ 사용
 - 육·해·공군은 구조적 개발방법론, 해군은 일부 객체지향 개발방법론 적용 중
 - 육·해·공군 모두 개발/관리도구 및 OS는 대부분 외국제품 사용

3) 비무기체계에서의 공개SW활용 현황

가) 비무기체계에서의 정보시스템 전반적 현황

- 국방시스템 중에서 육군 비무기체계에서의 정보시스템에 대한 전반적 현황에 대한 전반적현황은 다음과 같다.
- 특정 HW(OS)에 종속된 SW 선택으로 수직계열화
 - 서버 OS에 따라 올라가는 제품군(DB, WAS, WEB)이 동일 계열로 스택화됨으로 HW(OS) 종속화 및 비용증감 요인으로 작용
- HW 도입시 연동되는 상용SW로 구성
 - 군 용도에 보다 적합한 상용SW 선택보다는 HW에 의존적인 상용 SW 구매현상 발생
- 공개SW 보다 상용SW 선호
 - 일반 상용SW 환경에 익숙한 운영자들이 새로운 환경(공개SW)의 적응에 소극적인 경향
 - 공개SW에 대한 적용사례, 경험 부족에 따른 막연한 불안감 존재
- 이러한 현상은 국방 분야 정보시스템이 일반적으로 대형이며 안정성이 요구되고 유지보수 및 기술지원을 중요시 함에 따라 국내SW 시장의 종속화, lock-in 현상을 그대로 답습하고 있으며, 장기적으로 군의 자체 SW개발 역량 및 보안성 강화 등 육군 자체역량 강화와 독립적인 시스템 운영에 많은 제약점이 존재하는 것으로 판단됨

나) 서버용 SW의 현황

- 일반적인 정보시스템은 구축형태에 따라, 웹서버형, 클라이언트/서버형, Stand Alone로 구분할 수 있는 것처럼 비무기체계에서의 국방 정보시스템도 이와 같이 구분할 수있고 각각에 대한 특성은 다음과 같다.
- 웹서버형(웹환경에서 개발된 응용체계)
 - 애플리케이션 사이에서 플랫폼, 프로그래밍 언어와는 독립적인 상호 연결
 - 모듈 형태별 상호 운용 방식으로 효율성 증대
 - 어플리케이션과의 느슨한 연결 방식(Loosely Coupled)
 - 느슨한 연결방식으로 유지보수 개발시간과 비용 절감

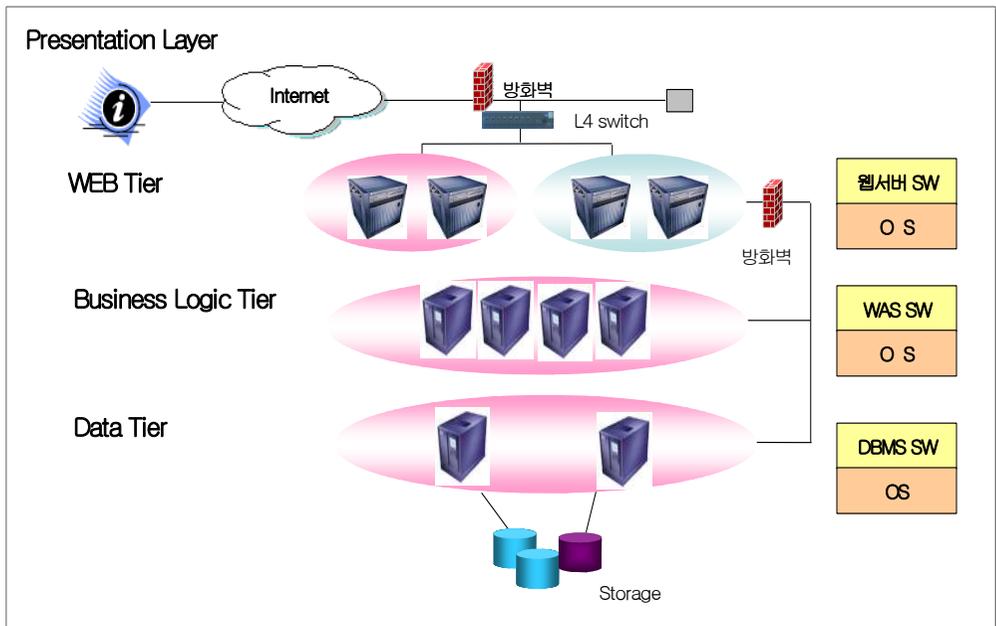


그림 28. 웹서버형의 구성도

- 클라이언트/서버형(클라이언트, 서버 환경에서 개발된 응용체계)
 - 클라이언트-서버간 전용 프로그램 사용
 - 업무시스템 개발 시 클라이언트-서버 각각 개발해야 하는 단점 발생
 - 클라이언트 프로그램 변경 시 배포된 모든 프로그램 변경
 - 타이트한 인터페이스 연결
 - C/S 형태에서 Web 환경으로 전환되고 있는 추세임



그림 29. 클라이언트-서버형의 구성도

- Stand Alone형
 - 소규모 업무와 독립적인 운용업무에 적합
 - 타이트한 연결방식으로 운영유지보수 어려움 발생

- 또한 육군 정보시스템을 유형별로 분류하면 실제로는 구축형태별로 다양한 유형으로 분류되지만 가장 보편적 구성형태는 다음과 같다.
- OS로는 Unix 계열인 Unix, AIX, HP-UX, Solarix, Windows 등이 사용
- DBMS로는 Oracle, Informix, MySQL, MSSQL, Access 등이 사용
- WAS로는 Tomcat, Weblogic, Oracle-AS 등이 사용
- Web server로는 Apache, IIS 등이 사용

그림 30. 육군 정보시스템의 보편적 구성 형태도

5. 국방시스템에서의 공개SW활용을 위한 정책제안

가. 국방정보화의 동향과 현상 분석을 통한 발전방향

1) 국방정보화 발전연구단

- 2020년을 대비한 차세대 정보화의 발전을 위해서는 패러다임의 대전환 필요하다는 인식하에 국방부는 국방정보화 추진 현상 진단 및 문제점 분석을 통한 국방 정보화 핵심과제의 도출을 위하여 아래와 같이 국방정보화 발전연구단을 운영하여, 국방정보화의 동향과 현상 분석을 통한 발전방향을 도출하였다

■추진경과

- '06.2.27 . 6.16 국방정보화발전연구단 운영
- 주요 관계관 보고 및 의견수렴 :
 - 차관(6/1), 자원관리본부장(6/1),
 - 혁신기획본부장(5/31, 6/7), 합참의장(6/12)
 - 장관보고 : 6/20, 6/21

2) 패러다임의 변화

- 범정부 차원의 대규모 정보화 추진 가속화
 - 첨단 IT 산업육성
 - IT서비스 + 인프라 + 제조업을 연계하여 신 성장 동력 창출
 - 세계최고 수준의 전자정부 구현
 - 인터넷을 통한 민원 서비스 개선, 공공기관 일하는 방식 혁신
 - 유비쿼터스 환경에서 삶의 질 혁신
 - 학생은 e-Learning확산으로 캠퍼스에서 전 세계 도서관 이용
 - 주부는 홈 오토메이션을 통해 모든 가전기기를 원격통제
 - 기업은 전 세계를 무대로 24시간 무중단 글로벌 비즈니스 수행
- 주변국의 국방 정보화 동향
 - 미국은 IT를 기반으로 군사변환
 - NCW 추진 선도, 단위 부대를 디지털 고속통신망으로 연결
 - 전세계정보격자망(GIG) 기반 위에 합동 지휘통제체계 구축
 - 일본은 세계 1위의 민수분야 기술 생산력을 바탕으로 정보화 추진

- '07년초 까지 총 4기의 첩보위성 발사
- 중국은 군용기술 우선의 과학정책으로 군사대국화 지향
 - NCW와 동시대응 개념에 중점을 두고 정보화
 - 점혈전략(點穴戰略)과 연계하여 추진
- 차세대정보화를 위한 도전대상
 - 차세대 형의 국방업무 & 차세대 형의 전장관리
 - One Server 와 Back Up 장치가 지원하는 통합체계
 - 대폭 격상된 2020 통신 Infra
 - 독자적 NCW를 개척하면서 선진국 NCW 기법 수용

2) 현상분석

- 현재의 정보화 방법으로는 업무개혁하기 어렵다. 왜냐하면
 - 정보화를 통해 업무개혁 하려면 두 가지 과제 공히 중요
 - (1) 새로운 업무방법채택 -> (2) 정보체계구축
 - 전자(1)는 업무적 정보화, 후자(2)는 기술적 정보화
 - 결국, 업무적 정보화 수준 만큼 국방정보화 목표달성
 - A. 업무정보화 단계에서 시간, 예산, 특히 지휘관심 투자 부족
 - B. 설익은(덜 익은 밥) 설계로 기술정보화 단계 진입
 - C. 추진과정에서 발견된 Idea나 문제점 대부분 사장

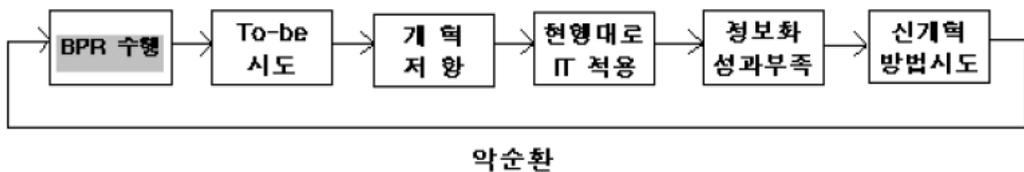


그림 31. 현재의 정보화 방법에 의한 악순환 개념도

- 전형적인 굴뚝형(Stovepipe) 정보체계
 - 중앙통제 없이 기능별로 정보화 추진
 - 부분적으로 구현된 다수의 분리된 정보체계 산재
 - 전체 체계로서의 목표와 성과를 기대하기 어려움
 - 향후에도 전장관리체계와 자원관리 체계 분리 구축 우려

- 정보화 수요 대비 통신 인프라 절대 부족
 - 정보체계와 통신 인프라 간 심한 불균형
 - 통신 인프라 능력 고려 없이 정보체계 구축
 - 평시 주둔지 전송로도 부족
 - 전시 이동통신 능력과 정보체계 소요 미달
 - 전시 환경고려 시 이동통신 용량 및 기반환경 모두 부족한 실정
 - 군사 요구조건 과다한 적용으로 통신 인프라 구축을 위한 예산증가 및 사업 장기간 소요
- 신기술 도입 지연
 - 정보체계 속성상 진화적 획득이 필수적이거나,
 - 실제로 예산제도와 연계성 부족으로 적시적 성능개선이 어려움
 - ※ 사례 : 일괄개발 방식으로 구축된 물자, 탄약, 시설 정보체계의 경우 사용성과를 내기 전에 재개발과 다름없는 성능개선 사업 추진으로 예산 낭비
 - 군사규격 과다적용 및 획득체계 경직성으로 발전된 민간 기술의 적시 적용 지체
 - IT 기술은 급속히 발전하나 군의 유입기반 부족
 - 미래 NCW를 대비한 군의 전문 인력도 부족

3) 국방정보화의 발전방향

- 국방정보화의 추진원칙
 - 국방 정보화를 통해 이룩할 체계는 전투우선, 예산중심, 관리지향 체계가 되어야 한다
- 국방정보화의 목표체계
 - 국방 통합 정보지원 서비스센터와 유비쿼터스 환경 하에서 활동하는 모든 Actor(기동대대, 전투원, 업무담당자, 지원요소)가 상호 연계되어 운영되는 단일 체계이다.
 - 따라서 국방부, 합참, 각군의 모든 정보체계는 국방정보화 목표체계의 부분체계가 된다.

가) 개혁목표 1. 차세대형 국방통합정보체계 구축

<표 17> 차세대형 국방통합정보체계 구축을 위한 핵심과제 내용

핵심과제	과제 세부 내용
<p>기존의 굴뚝형 정보체계를 일부 성능개선 및 재구축하여 하나의 체계로 통합(Merging)</p>	<ul style="list-style-type: none"> ▪업무적 정보화(개혁)와 기술적 정보화 연계 추진 ▪업무적 정보화는 TOP DOWN식, 정보체계 구축은 기능/군별 분권화 ▪국방 EA 기법을 적용 ▪국방 전체에 공통 적용할 수 있는 표준 정립 시행 ▪서비스 유형별 통합 수준 설정 추진 ▪새로운 환경에 부합하지 않는 구 프로세스 개선 및 삭제 ▪장기적 지속사업으로 추진할 수 있도록 계획반영
<p>국방 EA 구축 추진 - 국방 전체 조직관점의 업무체계 설계(Architecture), 정보화 추진관리</p>	<ul style="list-style-type: none"> ▪국방 EA 추진 TF를 구성하여 관리 ▪신규 정보체계 구축 시 EA 적용 의무화 ▪정보화 투자기준 설정 및 종합발전계획 수립과 연계 ▪국방 성과관리 시스템 및 기능연계 모델을 통해 구축한 기존 산출물과 개념연구결과 활용 ▪체계 개발 사업관리의 통제수단으로 활용 ▪범정부 EA 도입 제도와 방침을 준용
<p>국방 통합 정보지원 서비스센터"구축</p>	<ul style="list-style-type: none"> ▪기존 메가센터 구축 사업의 추진방법과 일정을 조정하여, ▪국방차원 One Set System 목표 하에 서비스 및 운영개념 재정립 ▪요소별 통합 우선순위 설정 및 단계별 이행 ▪『국방통합 정보지원 서비스 센터』추진준비단 설치 및 운영

나) 개혁목표 2. 정보통신 인프라 혁신적 확충

- 핵심과제 : 2020 년을 목표로 정보통신소요 종합, 미래 네트워크 설계 및 전력화 반영
- 정보체계 수요를 고려한 통신 기반체계 재설계
 - 일차적으로 각종 정보체계간 필수 유통 데이터 범위 및 종류 정의
 - 정보 유형별 트래픽 제어를 통해 인프라 불균형 최소화
 - 자원관리, 전장관리 및 기타 분야의 수요를 종합적으로 검토
- 국방 BTL 사업 : 현 고정 지휘소 중심의 통신 인프라 조기 구축
- 민간 통신 서비스 및 국가 인프라 활용과 연계, 전력화 반영
 - ALL IP/Mobile IP 기반 데이터 전송환경
 - 광대역 전달망 + Ad-hoc 전술 이동망 + 센서망이 결합된 『전술 광대역 통합망』
 - SDR (Software Defined Radio) 기술 적용한 개방형 단일 플랫폼
 - 지휘소의 무선 LAN + Telematics 환경으로 경량화 및 기동 간 지휘통제 보장
 - NCW 구현 고려, IT/네트워크 군 전문가 양성 및 연구기능 활성화
- 국방부/합참 추진 팀을 구성, 정책과제로 추진

다) 개혁목표 3. 독자적 NCW 구현과 전장 정보화



* Ad-hoc 네트워크 : 별도의 유선 기반시설 없이 이동형 단말기들로 자체 네트워크를 구성할 수 있는 무선통신망

그림 32. 국방통합정보지원센터를 통한 NCW 구현의 개념도

<표 18> 독자적 NCW 구현과 전장 정보화를 위한 핵심과제 내용

핵심과제	과제세부내용
NCW 개념 구현을 위한 실험부대 - 가칭 u-J 여단 (실험여단) 창설 운영	실험부대 u-J여단 개념 u (Ubiquitous) : 기술 → 첨단기술로 무장 J (Joint) : 전술 → NCW 개념하의 육.해.공 합동/합동전술 채택 여단 (Brigade) : 실험부대 규모 NCW의 핵심 기반 = 통신체계 + {uJ 지휘통제체계 + uJ 군사정보(情報)체계 +uJ 군수 지원체계} 독자적 NCW 개념 구현을 위한 정보화 차원 기반 조성 합참 또는 육군 지휘 하에 추진체 구성(가칭 NCW 추진발전단) 썩크탱크(Think Tank) 설치: 기술과 전술을 연계할 Technology & Tactics Center 실험부대 『u-J 여단』 창설 및 운영 별도의 법.제도, 인력, 예산 등의 지원체제 구축
2020 목표로, 전략/전술 군사정보 체계대혁신 (IT 분야 우선 반영)	첨단 IT 환경 하에 정보의 수집-분석-처리-분배-활용 체제 전반에 대하여 마스터플랜 작성 국방통합정보체계의 일부로 MIMS 등 각급 정보체계 구축 추진 전략/전술 정보유통에 요구되는 통신 소요는 전력화하여 반영 정보본부 중심으로 사업추진 팀 구성

다) 개혁목표 4. 제도개선

<표 19> 제도개선을 위한 핵심과제 내용

핵심과제	과제세부내용
<p>IT 환경에 부합하도록 법.제도 개선 추진</p>	<p>신기술과 새로운 전술개념 실험부대에 예산사용 절차 법제화 예산과 연계하여 정보체계의 진화적 구축 및 관리 제도화 획득체계의 개선과 연계하여 정보화 조직의 임무기능 및 역할 정립 정보화 추진상의 기술적 통제체제 법적 규정 마련</p>
<p>신기술의 유기적 유입기반 구비</p>	<p>기술 수요 및 공급 부서와 밀접하게 연결 산 . 학 . 연 전문 기관 및 협회 조직 활용 IT/네트워크 분야 군 전문가 양성 군 정보화 전문 인력의 공동체 구성 및 연구 활동 조직화 『u-J여단』을 테스트 베드로 첨단장비 전투실험 추진</p>

나. 우리나라와 미국방성의 공개SW 관련 현황 비교분석

- 우리나라와 미국방성의 공개SW 관련 현황 비교분석의 내용은 아래 표와 같다.

<표 20> 미국방성과 우리나라 국방시스템의 공개SW 현황 비교표

구분		미국방성	우리나라 국방시스템
무기 체계	지휘통제 시스템	<ul style="list-style-type: none"> ▪미 파악 	<ul style="list-style-type: none"> ▪공개SW활용사례가 거의 없고 ▪공개SW의 활용에 긍정적인 시각과 부정적인 시각이 공존함
	기동무기	<ul style="list-style-type: none"> ▪수많은 공개SW를 활용하여 개발 중이며, 향후에는 공개SW를 개발하는 방식을 전체무기로 확대하기위하여 우선 첨단제품에 대해 적용코자 추진 중 	
비무기 체계	자원관리	<ul style="list-style-type: none"> ▪기반구조 지원 <ul style="list-style-type: none"> - 네트워크 지원 ▪소프트웨어 개발 <ul style="list-style-type: none"> - C, C++, Ada, Perl, Python 등 ▪보안 <ul style="list-style-type: none"> - 고신뢰도 운영체제 , 긴급대응 등 ▪연구 <ul style="list-style-type: none"> - 수학적도구, 저-비용 슈퍼컴퓨팅 등 	<ul style="list-style-type: none"> ▪서버 중심으로 리눅스, DB, WAS 등의 공개SW가 다수 사용 중 ▪사례부족으로 공개SW 도입에 소극적인 경향이 있음
	기반시스템	<ul style="list-style-type: none"> ▪상동 	<ul style="list-style-type: none"> ▪전술이동통신망에 와이브로 기술을 적용키 위하여 추진 중임

다. 국방시스템에서의 공개SW활용을 위한 정책제안

1) 제안 개요

- 국방시스템의 체계분류에서 기술한 것처럼 국방시스템에서의 공개 SW 활용을 위한 적용대상 시스템은 무기체계의 군사정보, 모의훈련 등의 시스템과 비무기체계 자원관리 분야의 예산회계, 기획인사, 군 수조달, 전자행정 등의 시스템으로 분석되고 있다.
- 이들 적용대상 시스템들에 대한 SW의 활용현황과 공개SW의 특성을 비교 분석한 결과 다음과 같은 네 가지 측면에서 공개SW의 활용에 대한 필요성이 제기되고 있고 국방부에서 목표로 하고 있는 공개 SW를 활용한 유비쿼터스 기반의 정보화 강군 구현을 위한 정책적 지원이 필요한 것으로 나타나고 있다.
- 첫째는 자체개발환경 및 상호운용성 확보로서, 공개SW를 활용하면 기술개발과 관련된 장점으로 사용자 측면에서는 소스코드의 수정 및 확장, 업체의 의존탈피, 유지관리의 영속성, 호환성 확보, 정보시스템 통합의 용이성의 장점이 있고, 국가산업적 측면에서는 원천기술확보를 통한 SW산업의 육성에 기여를 할 수 있는 장점이 있다.
- 둘째는 보안체계 강화로서, 미국방성의 연구결과에서도 공개SW는 보안에 우수한 것으로 나타났고, 기간별 윈도우즈와 리눅스의 취약점 발표 비교도, 세계적인 기관들로 부터의 발표자료에 의하면 공개SW는 보안에 많은 기여를 하는 것으로 나타났다.
- 셋째는 예산절감으로서, 공개SW는 라이선스 비용이 무료이기 때문에 총소요비용에서 우수한 것으로 나타나고 있다.
- 넷째는 공개SW모범사례 확보로서, 그동안 정부의 노력으로 많은 구축사례를 확보하고 있지만 어느 공공기관보다도 안전과 신뢰성이 요구되는 국방시스템에서의 모범적 구축사례는 공개SW의 확산에 많은 기여를 할 것이다.

2) 자체개발환경 및 상호 운용성 확보

- 국방정보화 발전연구단이 분석한 우리나라 국방정보시스템의 현상분석과 국방시스템의 SW활용 현황에 기술된 것처럼 우리나라의 국방정보시스템은 전형적인 굴뚝형(Stovepipe) 정보체계로서 중앙통제 없이 기능별로 정보화가 추진되어, 부분적으로 구현된 다수의 개별정보체계가 산재됨으로 인해 전체 체계로서의 목표와 성과를 기대하기 어려운 실정이며, 향후에도 전장관리체계와 자원관리 체계 분리 구축이 우려되고 있는 실정이다.

그림 33. 전장관리와 자원관리체계간의 연동제한 모습

- 국방부에서는 이러한 문제의 극복을 위해 국방 엔터프라이즈아키텍처(EA)의 도입을 추진하고 있다. EA란 일종의 IT 설계도로 조직의 비전과 목표를 설정하고 이를 실현하기 위해 IT를 어떻게 활용 할 것인지를 규정한 일종의 표준 가이드라인으로서 아래 그림과 같이 건축분야에서 도시에 대한 설계를 하고 건물을 구축하는 것처럼 정보화분야에서도 기업 환경이 복잡해지고 규모가 커지면서 정보기술(IT) 시스템을 효율적으로 사용 하는 방법이 필요함에 따라 조직전체의 설계를 한 후 정보시스템을 구축하는 방식이다.

그림 34. 건축분야와 정보화분야의 EA개념 비교도

- 국방정보시스템에서 가장 큰 규모를 가진 육군의 정보시스템에 대한 전반적 현황에서도 나타난 바와 같이 수백 개의 단위 정보시스템들이 구축형태에 따라 웹서버시스템, 클라이언트-서버시스템, 스탠드얼론시스템으로 구분되고 단위시스템의 개발 시에 개발을 주도하는 업체에서 하드웨어를 선택하면서 관련된 OS가 선택되고 해당되는 SW들이 수직 계열화되는 방식으로 선택됨으로 인해 단위 시스템들 간에 업무의 연계성은 물론이고 상호운용성이 확보되지 못하고 있는 실정이며, 또한 외산 상용SW 의존성에 따른 자체SW 개발 능력 배양에 많은 어려움이 있는 실정이다.

- 공공시스템에 대한 일반적인 단위시스템의 개발은 아래와 7단계의 과정을 거치면서 개발이 이루어지는데, 단위사업에 대한 계획서의 작성에서 계약체결까지는 관련부서에서 담당하지만 실제의 사업수행은 계약을 체결한 업체의 전문가들에 의해 수행되고, 이들에 의해 HW에 의존한 상용SW가 선택되고 이에 맞는 응용업무프로그램이 개발되기 때문에 자체개발환경이 확보되지 못한 실정이다.
- 사업계획서 작성 : 과제 발굴, 예산편성, 사업 타당성, 적정성 검토, 사업 추진계획서
- 제안요청서 작성 : 제안요청서 작성(객관성 및 타당성 검토가 철저히 요구되는 단계)
- 사업공고 및 제안서 접수 : 입찰공고, 제안 설명회, 사업제안서 접수
- 제안서평가 및 선정 : 제안서평가(평가위원회), 원가분석, 기술협상 및 가격협상
- 계약체결 : 최종계약체결
- 사업수행 : 사업수행, 진도보고(착수, 주간, 중간보고 등), 완료보고
- 결과보고 : 최종 결과보고서 평가(검수)
- 그러나 국방시스템에서 단위정보시스템의 개발 시에 공개SW를 활용하게 되면 소스코드가 공개되는 공개SW의 특성으로 인해 공급자 위주의 상용SW와 달리 아래 표와 같이 수요자 측면에서는 소스코드의 수정 및 확장, 업체의 의존탈피, 유지관리의 영속성, 호환성확보, 정보시스템 통합의 용이성의 장점이 있음은 물론이고 국가 산업적 측면에서 기술개발능력의 배양에서 많은 장점을 가지고 있기 때문에 자체개발환경의 확보에 많은 기여를 할 것이다.

<표 21> 국방시스템에서 공개SW 활용시 기술개발과 관련된 장점

구분		국방시스템에서 공개SW활용 시 기술개발과 관련된 장점
수요자 측면	소스코드의 수정 및 확장	공개소프트웨어는 소스코드가 공개되어 있기 때문에 현재 시스템에 적용되어 있는 소프트웨어의 수정과 추가 확장을 위한 변경작업이 얼마든지 가능하다.
	업체의 의존 탈피	독점소프트웨어의 경우에는 선택의 여지없이 사용해야 만하는 불합리성이 존재할 수 있지만 공개소프트웨어는 공급업체와 제품이 다수이기 때문에 이와 같은 구속은 존재하지 않는다.
	유지관리의 연속성	원공급자가 폐업하여 사라지거나 지원을 중단하더라도 공개된 소스를 보유하고 있으므로 기관은 해당코드를 새로운 공급자에게 양도함으로써 정보시스템 유지관리의 지속성(연속성)을 보장받는다.
	호환성 확보	공개소프트웨어는 다양한 형태의 소스코드를 배포하고 있으므로 상대적으로 많은 플랫폼에 이식가능하며 또한 이와 같은 플랫폼 독립성(호환성)은 수요자에게 하드웨어 선택의 폭을 넓혀준다.
	정보시스템 통합의 용이성	정부 및 공공부문의 정보시스템 구축 시에 핵심모듈을 공개소프트웨어로 구축할 경우 공공부문의 개방표준화가 용이하게 되어 정보통합 예산을 절감하고 호환성 및 이식성을 확보하여 공공부문 전체의 정보시스템 통합을 촉진시킬 수 있다.
국가 산업적 측면	SW개발 원천기술을 확보함으로써 지식기반경제 핵심 산업인 SW산업의 경쟁력을 강화시키고, 최신기술의 공개, 개발자 커뮤니티 등을 통한 기술교육에 의하여 국내 SW제품의 선진제품과의 기술격차를 줄일 수 있으며, 빠른 신제품 개발 등 시장에 즉시 대응이 용이하며 개발의욕 고취를 통한 양질의 핵심SW 인력양성이 가능하다.	
확보 가능한 기술개발 능력	<ol style="list-style-type: none"> ① 운영체제, 웹서버, DBMS, 사용 언어 등에 대한 기반기술 ② 공용 framework 개발 및 업무 서비스 요구 분석 및 설계 기술 ③ 분산 DB 구축 설계 및 관리 기술 ④ 그룹웨어의 확장 및 공유 수준 지정에 의한 access 범위 설정 기술 ⑤ 유관 기관 간 정보 DB 표준화 기술 	

- 특히 상기 표에서 나타난 호환성 확보와 정보시스템 통합의 용이성은 국방부가 추구하는 국방 엔터프라이즈아키텍처에 구현에 있어 핵심역할을 할 수 있을 것이나 이를 위해서는 안전성, 보안성 및 신뢰성이 확보된 표준화 된 모델의 개발이 선행되어야 할 것이며, 아래에 제안된 그림은 하나의 예를 보여주고 있다.

그림 35. 안전 및 신뢰성을 위한 표준 모델의 구성도

- 국방시스템에 적용 가능한 표준화된 공개SW의 모델을 개발하고 이를 통해 자체개발환경을 확보한 후 국방부에서 추진하는 국방엔터프라이즈아키텍처에 따라 기능별 자원관리 정보체계를 통합하고, 전장관리체계와 연동, 전·평시 지속적인 자원정보 제공하고, 전자행정 업무를 한곳에서 처리할 수 있는 포털구축단위시스템을 개발하게 되면 국방부가 추구하는 상호운용성이 확보될 것이다.

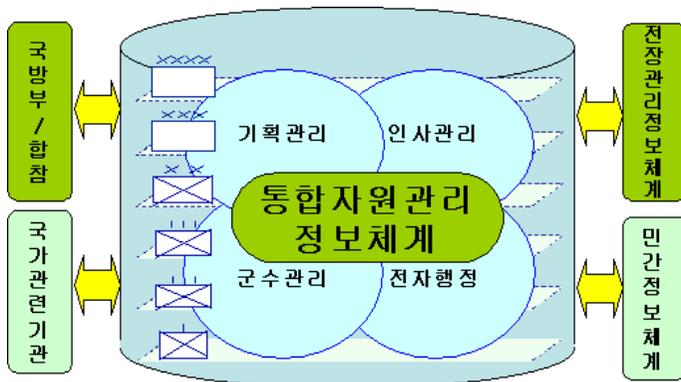


그림 36. 상호운용성이 확보된 체계의 개념도

3) 보안체계강화

- MITRE가 미 국방성에서의 공개SW 활용현황을 광범위하게 조사대상으로 하면서 상당한 수준의 광범위한 survey를 통해 수행한 연구 결과에서도 나타난바와 같이 공개SW는 보안에서 아주 우수한 강점을 가진 것으로 분석되었고, 기타 여러 자료에서도 많은 전문가들이 공개된 소스코드를 공동으로 보고 있기 때문에 상용 독점SW에 비해 상대적으로 빨리 취약점을 발견하고 해결함으로써 보안에서 우수한 것으로 평가하고 있으며, 이러한 점에서 우리나라 국방시스템에서의 보안체계의 강화를 위해서 공개SW의 활용이 필요한 것으로 나타나고 있다.
- 아래 그림은 대표적 상용SW인 윈도우 XP와 대표적 공개SW인 리눅스의 기간별 취약점 발표 회수를 그래프로써 보여주는 것으로 왼쪽 윈도우 XP의 경우에는 기간과 독립적으로 취약점 발표 회수가 일정하게 나타나는데 비해 오른쪽 리눅스의 경우에는 초기에 취약점 발표회수가 많으나 빠르게 안정이 되면서 시간이 지날수록 안정화가 되는 것으로 나타나고 있다.

그림 37. 윈도우 XP와 리눅스의 초기 취약점 발표 횟수 비교도

- 또한 2005년 7월 SANS가 발표한 보고서에 따르면 20개의 인터넷 보안 취약점들 가운데 단 2개의 가벼운 취약점만이 리눅스 플랫폼 사용에 영향을 주는 것으로 나타났고, 세계적 리서치 기관인 가트너는 제로데이 어택(Zero-day Attack)시대라는 점을 기준으로 볼 때 리눅스의 보안성이 높은 것으로 보고하고 있다.
- 일반적으로 보안성은 기업 및 기관에서 운영하는 IT인프라의 안정성 및 신뢰성과 직결되는 문제로서, 사내 인프라의 보안수준이 낮다면 기업이나 기관은 비즈니스 연속성(Business Continuity)을 보장할 수 없을 뿐만 아니라 예기치 못한 장애로 막대한 피해까지 입을 수 있다. 이 같은 관점에서 볼 때 리눅스는 공개소프트웨어만이 제공하는 보안 메커니즘을 통해 안정적인 서비스를 보장한다고 볼 수 있다.
- 리눅스의 보안성이 얼마나 뛰어난지에 대한 대표적인 사례로는 미국 국가안전보장국(NSA;National Security Agency)에서 학계, 업계의 보안 전문가들의 참여를 통해 추진하고 있는 SELinux(Security-Enhanced Linux) 프로젝트가 있다. SELinux는 운영체제 관련 가장 진보된 보안 프로젝트 중 하나로 평가받고 있으며 이 같은 프로젝트에 리눅스가 선택된 배경은 공개소프트웨어의 주요 특징 중 하나인 투명성때문이다. 소스를 제공하지 않는 비공개 소프트웨어의 경우 보안 결점이나 버그들이 곳곳에 있어도 이를 공급자가 인지하고 개선안을 마련하기 전까지 알 수 없으나 공개소프트웨어는 소스 코드 분석을 직접 해볼 수 있어 사용자의 능동적인 대처가 가능하며, 공개소프트웨어 지원업체 및 관련 커뮤니티를 통해 보안결점이나 버그들에 대한 개선이 빠르게 일어나 각종 보안 침해 사고를 최소화 할 수 있다. 즉 소스가 폐쇄된 독점소프트웨어의 경우 보안상의 문제 발생 시 해당 업체의 대응을 기다리는 방법밖에 없지만 공개소프트웨어의 경우 보안 취약성과 같은 결함 발견 시 여러 업체 및 공동체의 공동대응으로 짧은 시간 내에 패치 또는 업그레이드가 이루어지기 때문에 보안성이 높다고 할 수 있다.

- SELinux는 리눅스 보안 모듈 구조체를 이용하여 리눅스 커널에 의무 접근 제어를 구현하여 보안을 강화하고 있다.
- 표준 리눅스 보안은 자유재량 접근 제어 모델로서, 이 모델에서는 파일과 자원에 대한 결정권은 오직 해당 객체의 사용자에게 있고 소유권에 따라 이뤄진다. 이에 따라 각 사용자와 그 사용자에 의해 실행된 프로그램은 자기에게 할당된 객체에 대해 전적으로 자유 재량권을 갖는데, 이러한 상황에서는, 악의 있는 일반 혹은 루트 사용자가 실행시킨 결함이 있는 소프트웨어를 통해 주어진 객체로 원하는 어떠한 일을 해도 막아낼 방법이 없으며 보안 정책을 시스템 전체에 걸쳐 시행되도록 할 방법이 없다.
- 반면 의무접근제어 모델은 보안 정책을 모든 프로세스나 객체에 대하여 관리차원으로 규정함으로써 모든 프로세스와 객체를 제어할 수 있고, 단지 인증된 사용자가 아니라 이용 가능한 모든 보안 관련 정보에 근거하여 이뤄진다.
- 최근의 자료에 의하면 국산 리눅스 버전의 차세대 보안 OS인 ‘Secuve TOS for SEEN’이 개발된 것으로 발표되었는데, 이러한 공개SW는 충분한 시험평가를 거치고 우리나라 국방시스템에 맞도록 검토수정한 후에 활용하게 되면 국방시스템의 보안체계강화에 큰 역할을 할 것이다.

4) 예산절감

- 상용 독점SW는 일반적으로 해당 라이선스에 의해 판매 되고, 이에 따라 시스템을 증설하기 위하여 서버의 숫자가 증가한다면 독점SW 솔루션 도입비용은 계속 상승하게 되나 공개SW는 라이선스의 비용이 무료이기 때문에 많은 예산절감이 가능하므로 우리나라 국방시스템에서의 예산절감을 위해서도 공개SW의 적극적 활용이 필요하다.

- 그러나 공개SW를 도입하면 운영비용이 너무 높다는 인식도 있는데, 이는 잘못된 오해로서 일반적으로 시스템 구축에 필요한 HW, SW 등 초기의 도입비용 뿐만 아니라 유지관리비용, 교육비용 등 정보시스템 구축과 운영에 필요한 모든 비용을 포괄하는 비용인 TCO(총소유비용)에 의하여 비교하는 것이 원칙으로서 IDC가 2002년에 발표한 인텔기반의 리눅스와 RISC기반의 유닉스 서버 간 TCO분석에 따르면 모든 면에서 공개SW의 TCO가 더 낮게 나타났다고 보고하고 있으며, 미국의 시장조사업체인 로버트프랜시스 그룹이 웹서버를 운영하는 전 세계 2천여 개 업체들을 대상으로 조사한 결과 3년간 리눅스기반의 웹서버를 운영 했을 때 윈도우를 사용 했을 때 보다 39%정도 TCO가 낮은 것으로 나타났으며, 이 조사에 따르면 하루 10만 건의 히트수를 처리하는 것을 기준으로 볼 때 3년간 웹서버 운영에 솔라리스 기반시스템은 56만2천불, 윈도우기반은 19만1천불, 리눅스는 7만4천불의 TCO가 소요되는 것으로 나타났다.
- 공개SW의 대표적인 장점인 라이선스 구매비용이 없다는 것은 공개 SW를 도입하여 적용하려고 하는 기업이나 공공기관에게는 비공개 소프트웨어에 비하여 절대적인 비용절감을 가져다주게 된다. 또한 공개SW는 설치시간, 유지보수비용, 교육비용 등에서 비공개 소프트웨어 보다 비용이 많이 들거나 설치가 복잡하다고 일괄적으로 말할 수 없고, 기술지원 서비스 전문업체로부터 다양한 서비스를 받을 수 있으며 필요한 경우 민간 커뮤니티로부터 무상으로 기술지원을 받을 수도 있다. 교육비용은 어떤 SW나 처음 접하면 반드시 필요한 것이며 공개SW라고 해서 더 많이 들거나 작게 드는 비용이 아니다.
- 미 국방성의 연구결과에서도 공개SW는 독점 상용SW와 경쟁을 촉진시켜 비용통제의 역할을 함으로서 예산이 절감된다는 것으로 분석되고 있고, 2005년 7월 기획예산처에서 정보화예산혁신을 위해 추진했던 2006년 정보화사업 중 공개SW 도입 사업 비용효과분석 결과 초기 도입비용이 평균 약 20% 절감되었다고 발표하였다.

5) 공개SW모범사례

- 공개SW는 기술적, 경제적 측면에서 많은 장점을 보유하고 있지만 구축사례 부족 등으로 각 기관들이 도입에 소극적이어서 이를 해결하기 위한 방안의 하나로 정통부는 현재 공개SW 시범사업을 진행하고 있으며, '2005년 공개SW 시범사업'의 대상기관으로는 통일부, 공군본부 등 11개 기관이 선정되었고, 2006년의 대상기관으로는 아래 표와 같이 국방부, 광주시, 조달청, 공정거래위원회, 강원전문대학, 금오공과대학교, 전국지방의료원연합회, 육군 교육사령부 전투지휘훈련단이 선정되어 사업이 추진되고 있다.

<표 22> 2006년 공개SW 시범사업 과제 내용

분야	기관명	추진과제
공개SW시범도시	광주시	SW 주권을 위한 OSS-광주 기반혁신사업
공개SW시범대학	강원전문대학	공개SW 시범적용대학구축
리눅스 데스크탑 도입/전환	국방부	공개SW 전용교육장 구축
	조달청	지능형상품정보 클라이언트/서버시스템 구축
	금오공대	리눅스 전용 교육 관리시스템 개발
서버전환	전국지방의료원연합회	공개SW 확대보급을 위한 보안강화 시스템 구축
	육군교육사 전투지휘훈련단	공개SW 기반 한국형 War Game 시스템 구축
	공정거래위원회	공개SW 기반 소비자 종합홈페이지 구축

- 특히 아래와 같은 사업목표를 위하여 2005년 8월에 공개SW인 리눅스를 활용하여 구축한 교육부의 신교육행정정보시스템(NEIS) 사업은 단독 데이터베이스(DB)서버, 웹서버 등 총 2천375대의 서버 운영체제(OS)로서 단일 정보시스템으로는 세계 최대 규모이고, 공개SW를 도입코자하는 기관에게는 모범적 구축사례의 하나로 평가되고 있으나 공개SW의 확산을 위해서는 더 많은 모범적 구축사례가 필요한 실정이다.



그림 38. NEIS 시스템의 사업목표

- 국방시스템에서는 2005년 12월 SW산업 발전전략 보고회 시 국산 SW 및 공개SW 육성을 위해 국방부 등의 기관에서 적극적으로 협조할 것을 대통령이 언급함에 따라 2006년 5월 정부혁신 지방분권 위원회, 한국SW진흥원, ETRI와 공동으로 공개SW 정부정책 및 운영기술 세미나를 실시하고, 2006년 7월에는 야전부대 공개SW 현 운영실태 파악과 공개SW 적용기관 방문 및 우수사례 분석을 하였으며, 2006년 8월부터 10월까지 3개월 동안 육군본부, 정통부, 한국SW진흥원, KAIST가 협동하여 육군 공개SW 도입 종합실행계획 작성을 위한 TFT을 운영하였으며 그 결과 공공기관 중에서도 가장 안정성과 신뢰성 측면에서 엄격한 요구사항이 적용되는 국방시스템에서의 구축사례는 타 기관에 많은 모범사례가 될 것으로 파악되었고, 아래 사항은 공개SW 활성화를 위한 국방부의 추진 경과이다.

- 공개SW 활성화를 위한 국방부의 추진 경과
 - 공개SW 시범사업 수행('05 ~ '06)
 - '05 : 국방 정보체계 통합 틀 개발 및 시범체계 구축(공군)
 - '06 : 각 군 정통교 공개SW 전용 교육장 구축(국방부),
공개SW 기반 한국형 War Game 시스템 구축(육군 교육사)
 - IT관련 위원회 구성 및 운영
 - 국방부·정통부간 MOU 체결('05. 2) : 협력 위원회 연 2회 개최
 - 국방부 산하법인 「국방SW 산학연 협회」 발족('05. 12)
 - 『공개 SW 도입운영 가이드 라인』 발간 및 배포 : '06. 3
- 특히 데스크탑 분야의 공개SW 활용에 대한 공공기관의 모범적 구축사례는 절실한 실정으로서, 정통부가 2010년까지 목표로 하고 있는 리눅스 서버 48%, 데스크탑 15%의 달성을 위해서는 국방시스템에서의 데스크탑에 대한 공개SW의 적용이 필요하고, 특히 국방시스템에서도 많은 소요를 가지고 있는 육군에서의 공개SW의 적용에 대한 검토가 요구된다.
- 아래의 사항은 하나의 사례로서 1개 연대급 부대를 기준하여 윈도우 XP를 사용하는 경우와 리눅스를 사용하는 경우에 대한 전체비용인 TCO의 비교를 보여주는 것으로 1개 연대급에서 약 6,500,000원의 비용절감이 이루어지는 것으로 계산되고 있다.
 - 연대급 기준 : PC 50대, 업무 활용: 워드작업(70%), 인터넷(20%), 기타(10%)
 - 윈도즈의 TCO : 대당 28만원(윈도즈 8만, 오피스 15만, 한글 5만)×50=14,000,000원
 - 리눅스의 TCO : 대당 15만원(리눅스, 한글, Thinkfree Office)×50=7,500,000원

- 다음 사항은 업무환경의 적합성에 대한 검토사항을 보여주고 있다.
 - 윈도즈 대비 리눅스 기능 대체

<표 23> 윈도즈 대비 리눅스 기능 대체 표

구분	윈도즈 프로그램	공개SW 및 리눅스 프로그램	비고
운영체제	마이크로소프트 윈도즈	리눅스	
워드프로세서	훈글	리눅스용 훈글	호환율:100%
오피스	마이크로소프트 오피스	MS Office file과 호환가능한 리눅스용 오피스 프로그램	호환율:95%
PDF Reader	Adobe Acrobat Reader	리눅스용 PDF Viewer	
메일 클라이언트	아웃룩익스프레스	리눅스용 메일 클라이언트	
웹브라우저	인터넷 익스플로러	모질라	
메신저	MSN 메신저	리눅스용 메신저 프로그램	
파일 압축	알집(압축프로그램)	리눅스용 압축 프로그램	GUI 기반
파일 매니저	윈도즈 탐색기	리눅스용 파일탐색기	
파일 공유	파일 매니저	리눅스용 파일공유 프로그램	GUI 기반

▪데스크탑의 활용성 비교

<표 24> 윈도우와 리눅스의 데스크탑에서의 활용성 비교 표

구분	리눅스	윈도우	비교
파일시스템 호환성	1	5	서버와 내용 상동
멀티운영체제	1	3	리눅스에서는 윈도우의 멀티 부팅을 지원하나 윈도우에서는 지원 미약.
시스템 안정성	1	4	윈도우는 문제발생 시 시스템을 재부팅해야 하나 리눅스에서는 재부팅 없이 해당 프로그램을 종료한 후 바로 사용 가능
CPU/메모리 사용	1	3	리눅스에서는 비효율적인 메모리 사용 요소는 바로 제거할 수 있으나 윈도우의 경우 시스템을 재부팅했을 때 제거되는 경우가 많음
APP간의 연동성	4	1	윈도우의 경우 응용프로그램간의 상호 연동이 뛰어나
응용프로그램 지원성	4	1	윈도우의 응용프로그램은 여러 회사에서 개발하나 리눅스의 경우 대부분이 오픈소스로 진행되어 여러 분야의 응용프로그램이 부족함
하드웨어 지원성	4	1	윈도우의 하드웨어 디바이스 드라이버의 제작은 하드웨어 제조사에서 수행하나 리눅스의 디바이스는 공개프로젝트 형태로 진행되어 개발이 늦음
가격성	1	5	리눅스의 가격은 윈도우에 비해 훨씬 저렴
웹페이지 접근성	4	1	홈페이지 제작사에서 대부분의 홈페이지를 IE에 최적화하여 네스케이프 등 HTML표준을 따르는 프로그램에서 구동이 불가능
기타	4	2	아직 리눅스의 데스크탑 환경은 윈도우로 보면 win95정도의 수준임

라. 국방시스템에서 개방형기술개발의 활용을 위한 정책제안

1) 제안 개요

- 국방시스템의 체계분류에서 기술한 것처럼 국방시스템에서의 개방형 기술개발의 활용을 위한 적용대상 시스템은 지휘통제, 감시정찰, 기동무기, 함정무기, 항공무기 등의 무기체계 분야와 비무기체계에서 기반체계에 해당하는 정보통신체계, 컴퓨터체계, 정보보호체계, 상호운용성 기반체계에 해당하는 시스템으로 분석되고 있다.
- 이들 중 무기체계에서는 하드웨어적으로 무기체계용 시스템으로 분류한 지휘통제시스템의 SW에 해당하는 C4I의 전장정보SW가 주요 대상이 되고, 장갑차, 탱크, 함정 등의 무기가 디지털화 되기 위해 필요한 무기체계용 임베디드SW가 주요 대상이 되고, 비무기체계에서의 기반체계에는 정보통신체계의 전술이동통신망과 전술데이터통신 네트워크의 비무기체계용 SW가 주요대상이 된다.
- 개방형기술개발의 활용의 주요 내용은 앞서 미 국방성의 개방형기술개발의 정책연구에서 나타난 것처럼 개방형 표준 및 인터페이스, 공개SW 활용 및 설계, 협업적 문화와 온라인 지원도구, 기술적 유연성의 4가지를 활용함으로써 NCW에 필요한 무기시스템 및 비무기시스템, 특히 이들 시스템들이 동작되게 해주는 SW를 경제적으로 획득하게 하여 국가의 안전보장을 제공케 하고 또한 국가적으로 중요한 역할을 하고 있는 국가의 산업기반을 진흥하는데 기여하는 것을 목적으로 하고 있다.
- 2006년 12월 참여정부가 의욕적으로 추진해온 국방개혁법이 제정됨에 따라 앞으로 아래 표와 같이 상시병력이 68만 명에서 50만 명으로 감축되어도 실제 우리군의 전투력은 증강하기위한 목표의 달성을 위해서는 개방형기술개발의 활용이 매우 중요한 역할을 할 것이다.

국방개혁법 제정안 주요내용

구 분	주요내용(2020년 목표)
상시병력규모	68만명에서 50만명으로 감축 (육군 37만1000명, 해군 6만4000명, 공군 6만5000명)
육군군단	10개에서 6개로 통폐합(전방 2개군단, 후방 2개군단 감축)
군수뇌부 및 산하기관장 인사청문회	합동참모회의 의장만 인사청문회 실시 육·해·공군 참모총장 및 방위사업청장 제외
신설 주력 부대	육군: 유도탄 사령부(북의 장사정포 대비) 해군: 기동전단(한반도 전 해역 감시·타격) 공군: 전방위 타격대(한반도 전역 감시·타격)
국방개혁 예산	2006년부터 2020년까지 621조원(방위력 개선비는 272조원)

〈자료: 국방부〉

그림 39. 국방개혁법 제정안 주요 내용

2) 무기체계의 개방형기술개발 활용

- 무기체계에서의 SW의 역할에 대한 중요성은 년도 별 점유율과 SW 라인수를 비교해보면 더욱 분명해지는데, 1960년대의 전투기인 F-4의 경우 점유율이 8%였으나, 2000년대의 F-22에서는 점유율이 80%에 해당되어 최신 전투기는 사실상 SW 제품이라고 해도 과언이 아닐 정도이고, F-22에서 필요로 하는 SW의 라인 수는 무려 1500만 라인 정도로 커지다 보니 개발방법에 있어 획기적 변화가 필요하다.
- 우리나라에서의 무기체계에 대한 개발방식도 유사하지만 미 국방성의 자료에 의하면 미국에서는 아래 그림과 같이 지휘관의 요청을 시작으로 하여, 운용개념 기술, 요구사항 검증, 자원조달, 설계 및 개발, 시험, 제조 및 통합, 구현의 과정을 거치게 되는데, 현재는 SW도 이러한 획득과정을 거치도록 되어 있다.

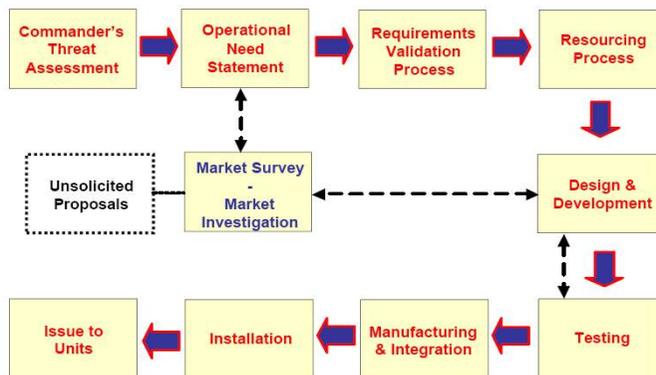


그림 40. 미국방성의 무기체계 개발 방식

- 이러한 획득과정에서, 특히 F-22와 같은 전투기에서는 수많은 SW개발자가 연관이 되어 시스템을 개발하게 되는데, 요구사항의 결정과 설계 후에 시스템으로 만들어지는 것이 하나의 기적이 아니고는 만들 수가 없게 되는 것이다.

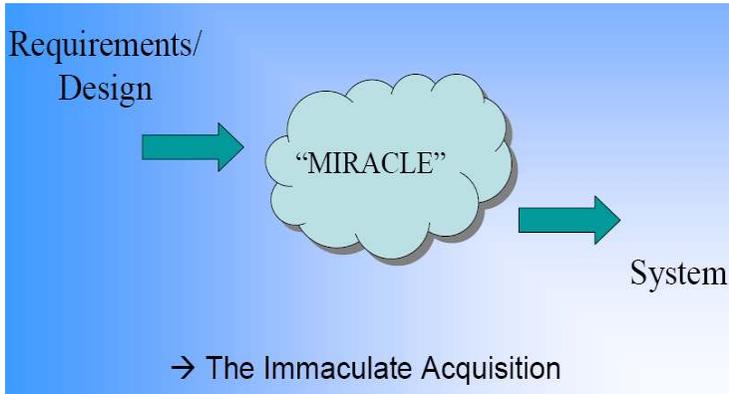


그림 41. 블랙박스 형태의 요구사항과 시스템 구현의 관계

- 아래의 그림은 미 국방성에서 개발한 UAH 장갑차의 년도 별 SW개발에 대한 생산성과 요구사항간의 비교를 나타낸 것으로 2003년 5월의 생산성은 15에서 17개월 후인 2004년 10월에는 450으로 30배 증가하였는데, 요구사항은 235에서 8105로 약 34배 증가되어 아무리 생산성을 증가시켜도 시간에 따라 빠르게 변화하는 요구사항에 대처하기가 매우 어렵다는 것을 보여주는 사례인 것이다.



그림 42. 지휘관의 요구사항과 생산성의 관계 비교도

- 지금까지 우리나라는 미군 국방시스템들과의 합동작전의 필요성에 따라 중요 무기체계를 미국으로부터 수입하거나 미국에서 개발된 무기체계를 모델로 하여 우리나라에 맞도록 설계변경을 한 후 국산화 개발을 추진하는 방식이 주류를 이루었으나 2012년으로 예상하고 있는 전시작전권의 환수에 대비한 자주국방의 전력화를 위해서는 선진화된 한국형 무기체계의 개발이 매우 중요하고, 개방형기술개발정책에 대한 심도 깊은 연구를 통해 우리나라에 맞는 개방형기술개발의 정책을 마련하고 추진해야 될 것이다.

3) 비무기체계 기반시스템의 개방형기술개발 활용

가) 전술이동통신망에서의 개방형기술개발

- 비무기체계의 기반시스템에서 전술이동통신망은 아래 그림과 같이 군지휘소(군단-사단-연대-대대 등)간은 물론이고 전투원들 간에도 Ad-hoc 네트워크를 구성케 해주는 이동통신망을 근간으로 한 기반시스템이다.



그림 43. 전술이동통신망의 구성도

- 국방부는 우리나라가 세계적으로 기술을 선도하고 있는 국산 휴대무선인터넷 기술인 와이브로 기술을 정통부와 협력하여 전술이동통신망 분야에 적용하기 위하여 검토를 추진하였는데, 아래와 같은 특성을 가진 와이브로는 고속으로 이동 중에도 높은 속도의 데이터 전송보장이 가능하고, 보안성이 뛰어난 최첨단 이동통신기술이다.
- 와이브로 개념도



그림 44. 와이브로 개념도

- 와이브로 개념 : 와이브로란 Wireless Broadband Internet의 약자로서, 넓은 지역에까지 도달하는 무선 인터넷이라는 의미이고, 외국에서는 이를 모바일 와이맥스라고 한다. 현재 우리가 사용하고 있는 네스팟 같은 무선 인터넷 서비스를 흔히 3세대 통신기술이라고 하는데 비해 와이브로는 3.5G 또는 4세대라고 하고, 가장 큰 차이는 이동성으로서 와이브로는 시속 100km 이상 고속으로 이동 중인 차량에서도 현재의 유선 인터넷 속도 이상으로 무선 인터넷 서비스를 즐길 수 있도록 하기 때문이다.
- 와이브로의 중요성 : 우리나라가 세계최초로 상용화한 CDMA 기술은 우리의 기술이 세계에서 인정받는 선진국이 되었음에도 불구하고 미국 퀄컴이 원천기술을 보유하고 있어 국내 휴대전화업체는 매년 기술 사용료를 지불해야 되고, 지난 10년간 퀄컴에 지불한 로열티만 해도 3조원이 넘는 실정이나, 국내에서 개발한 와이브로 기술을 사용하면 미국 회사에 비용을 지불할 필요도 없을 뿐만 아니라 우리가

기술 사용료를 받게 되는 것으로서 정통부와 삼성전자가 주축이 돼 개발한 와이브로는 2005년 IEEE에서 모바일 와이맥스 표준기술로 인정받은 기술이다. 우리나라는 2006년부터 SK텔레콤과 KT가 서울 일부지역에서 와이브로 서비스를 시작하여 기술개발국가로서 서비스 시장을 선도하고 있고, 나아가 삼성전자가 미국 스프린트와 전략적 제휴를 함으로써 와이브로 상용화도 속도가 붙고 있는 실정이다.

- 와이브로를 우리나라의 전술이동통신망에 맞추어 개발하고 전력화하여 우리나라 국방시스템에 적용할 경우 상용화된 국산기술을 활용함에 따라 군의 다양한 전술적 요구사항을 충분히 반영하여 군 요구에 대한 개발 기간단축이 가능하며, 군 작전환경에 적합하게 개발될 이동성 및 보안성에 대한 관련 기술은 민간에 이전하여 새로운 시장창출 및 수출도 가능할 것으로 보여 군 전력증강은 물론 국가경쟁력 제고에도 기여할 수 있을 것으로 국방부는 전망하고 있는데, 바로 이렇게 우리의 독자기술로 상용화된 시스템을 적절하게 적용하는 것도 개방형기술개발정책의 하나의 대안으로 고려할 필요가 있을 것이다.

나) 전술데이터링크에서의 개방형기술개발

- 전술데이터링크란 아래 그림과 같이 육·해·공군 및 합동/연합 작전을 지원하는 전술자료를 실시간 분배하는 신뢰성과 보안성 있는 통신체계로서 NCW의 핵심통신체계 기반시스템의 하나이다.

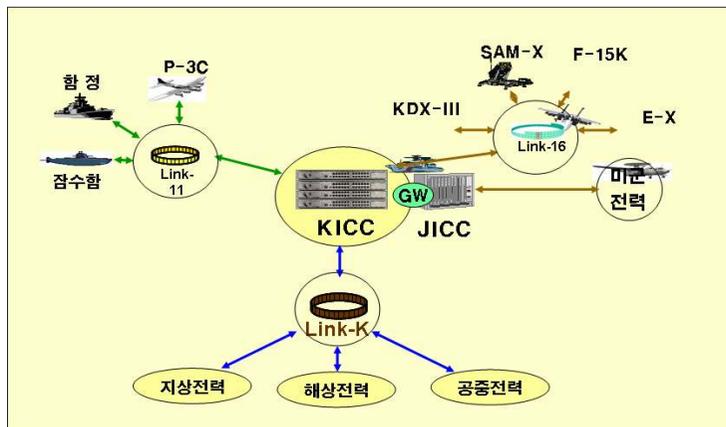


그림 45. 전술데이터링크의 개념도

- 현재 가장 대표적인 표준체제로는 링크(LINK)-16의 미 국방시스템으로서 링크-16은 무선 네트워크와 수동관제에서 사용하던 절차를 자동화한 프로토콜, 위치정보시스템(GPS)을 이용한 레이더 센서자료의 위치 정확성 향상, 주파수 호핑을 통한 통신방해에 대한 대응, 보안 솔루션이 모두 결합된 데이터링크 시스템으로, 정찰위성, 조기경보기, 관제레이더, 정찰 무인 항공기 등의 센서자료를 가공, 전투기와 지대공유도탄뿐만 아니라 전투기끼리 정보를 제공하는 체제를 갖춰 레이더로 표적을 추적하지 않더라도 데이터링크에 연동된 타 아군기의 자료를 이용해 미사일을 발사할 수 있는 기술수준에 도달해 있는 상태로, NCW의 핵심기반이나 국내 독자기술의 확보가 필요한 실정이다.
- 전술이동통신망은 우리나라의 독자기술인 와이브로 기술을 기반으로 하여 개발할 수 있는 기반기술의 지원이 있는 실정이나 전술데이터링크의 경우에는 핵심 통신부분인 HW보다 핵심 SW의 기반기술이 더욱 강한 시스템이기 때문에 기술기반이 부족한 시스템의 SW개발에 어떠한 개방형기술개발정책을 활용하는 것이 효과적인가에 대한 심도 깊은 연구가 필요할 것이다.

6. 결론 및 시사점

가. 연구 결과 요약

- 3장 미국방성의 공개SW 활용현황 및 정책연구에 대한 보고서의 세부적인 연구 분석을 통해서 아래와 같은 7가지의 정책적인 시사점을 도출하였다.
 - ① 미 국방성에서는 2002년에 115종의 공개SW가 다양하고 수준높게 활용되고 있었다는 점
 - ② 비용보다 필요성에 의해 공개SW를 활용하나 경쟁에 의해 가격이 조정된다는 점
 - ③ 공개SW가 국방시스템의 보안에서 우수하다는 점
 - ④ 미 국방성의 정책 방향은 국방력 향상 및 관련 산업의 육성이 동시에 고려된다는 점
 - ⑤ 개방형기술개발이란 공개SW개발모델, 개방형 표준 및 인터페이스, 기술의 유연성 및 협업적 개발방식의 4가지가 결합된 문화로서 대상도 SW를 포함한 첨단 IT제품이라는 점
 - ⑥ 개방형기술개발의 적용을 위해서는 국방획득제도의 변경이 필요하다는 점
 - ⑦ 개방형기술개발의 성공적 적용을 위해서는 관련된 조직, 예산이 필요하다는 점
- 또한 4장 우리나라 국방전력발전 규정과 공개SW활용현황에대한 연구를 통해서는 아래와 같은 결론을 도출하였다.
 - ① 국방전력발전 규정은 NCW에 대비한 방사청의 신설 및 획득과정의 절차, 획득시간, 상호운용성 등을 고려하여 개정되었으나, 첨단 IT제품의 획득에서는 여전히 문제가 존재.
 - ② 무기체계에서의 공개SW활용은 전문가들에 의한 조사 결과 현재로는 거의 없는 실정이나 비공식 설문조사 결과 공개SW의 적용은 바람직하고, 정책이 정해지면 군 특성상 단기간에 활성화 될 것이라는 긍정적 의견도 있는 반면 환경이 특수하여 공개SW 사용이 곤란하다는 부정적 의견도 있었음

③ 비무기체계의 자원관리체계에서는 수백종의 단위업무시스템이 개발되어 운영되고 있고, 구축된 형태는 웹서버형, 클라이언트-서버형, 스탠드얼론형으로서, 상당수의 시스템들에 리눅스를 포함한 공개 SW가 활용되고 있으나 대부분 특정 HW(OS)에 종속된 SW 선택으로 수직계열화되고, 서버 OS에 따라 올라가는 제품군(DB, WAS, WEB)이 동일 계열로 스택화됨으로 HW(OS) 종속화 및 비용증감요인으로 작용하고 있으며, 일반 상용SW 환경에 익숙한 운영자들이 새로운 환경(공개SW)의 적응에 소극적인 경향이 있는 상황임

④ 비무기체계의 기반체계내 정보통신시스템에서는 공개SW는 아니지만 우리독자기술로 개발되어 세계표준으로 인정된 와이브로기술을 적용하고자 추진 중임

■ 이에 따라 5장의 국방시스템에서의 공개SW 활용을 위한 정책제안에서는 공개SW의 활용과 개방형기술개발의 활용으로 구분하여 아래와 같이 정책적인 제안을 하였다.

▪ 공개SW의 활용을 위한 정책 제안

① 국방시스템에서 공개SW의 활용을 위한 적용 대상 시스템은

- 무기체계에서는 군사정보, 모의훈련 등의 시스템이고
- 비무기체계에서는 자원관리의 예산회계, 기획인사, 군수조달, 전자행정 등의 시스템임

② 적용 대상 시스템들에 대해서는 아래와 같은 4가지 측면에서 공개 SW를 적극적으로 활용할 필요성이 제기된다.

- 첫째, 자체개발 및 상호운용성 확보
- 둘째, 보안체계의 강화
- 셋째, 예산절감
- 넷째, 공개SW의 모범사례 확보

▪ 개방형기술개발의 활용을 위한 정책 제안

① 개방형기술개발의 활용을 위한 적용 대상 시스템은

- 무기체계에서는 지휘통제, 감시정찰, 기동무기, 함정무기, 항공무기 등의 시스템이고

- 비무기체계에서는 정보통신 시스템, 컴퓨터 시스템 등 기반체계에 해당하는 시스템임
- ② 적용 대상 시스템들에 대해서는 미 국방성이 국가 안보 및 관련산업 육성이라는 두가지의 목적을 위해 추진하는 것처럼 우리나라도 같은 목적으로 적극적으로 활용할 필요성이 제기된다.

나. 결론 및 시사점

- 미 국방성의 정책에 대한 연구 결과 미국은 SW산업에서 세계 제일임에도 불구하고 국방력과 SW산업의 육성을 위해 공개SW의 활용을 장려하는 정책을 추진하였고, 최근에는 더욱 발전된 개념의 개방형기술개발 정책을 추진하고 있는 반면 우리나라에서의 공개SW 활용 현황은 서버 중심의 리눅스 및 관련 SW로 제한되어 있어 미래의 NCW를 대비하여 공개SW와 개방형기술개발의 적극적인 활용의 필요성이 제기되었으며, 이를 위하여 정부, 대학 및 연구소, 산업체에서는 각자의 역할을 해야 될 것이고, 미 국방성의 사례와 같이 전체적으로 긴밀하게 협력하는 체계가 구축되면 우리나라 SW산업의 경쟁력은 물론이고 국방력을 획기적으로 향상시키는 국가발전의 초석이 될 것으로 확신한다.
- 정부에서 해야 할 일
 - 공개SW의 활성화를 위해서는 현재 육군의 현황분석을 통해 나타난 것처럼 공개SW가 확산되고 있지만 아직까지는 사례부족으로 공개SW도입에 소극적인 경향이 있는 것을 고려하여, 국방부는 정통부의 공개SW 시범사업과 연계하여 육군의 정보시스템에 다양하게 적용할 수 있는 모범적 공개SW 시스템을 개발하는 과제인 가칭 OAIS(Open sw based Army Information System) 프로젝트를 추진하고 결과를 단계적으로 확산해야 될 것으로 판단된다.
 - 개방형기술개발을 위해서는 대상 시스템들이 대부분 무기체계이고 무기에서 필요하는 임베디드 SW 및 첨단 기술에서 많은 활용이 필요하며, 아직은 개방형기술개발의 개념이 정립되어가는 단계이니 만

컴 국방부에서는 관련부서에서 우리나라의 실정에 맞는 개방형기술 개발 로드맵을 수립하는 과제를 수행할 필요가 있는 것으로 판단된다.

▪대학 및 연구소에서 해야 할 일

- 미 국방성의 연구 결과에 의하면 대학 및 연구소에서의 공개SW 활성화 수준과 실제 국방시스템에서의 공개SW 활성화 수준과는 상당히 비례하는 관계가 있는 것으로 판단되며, 이에 따라 대학 및 연구소에서는 구성원들이 리눅스를 포함하여 다양한 공개SW가 쉽게 사용되도록 환경이 구비되어야 할 것으로 판단된다.

▪산업체에서 해야 할 일

- 산업체 입장에서는 시장이 부족하여 사용자가 필요한 공개SW를 개발하는데 어려움이 있지만 반면 사용자가 필요한 SW의 공급이 없이는 결코 시장이 커지지 못한다는 것을 인식하고, 사용자가 필요한 SW 개발에 적극 동참해야 될 것으로 판단된다.

7. 참고 문헌

- [1] Mitre Report: Use of Free & Open Source Software(FOSS) in the US DoD, Version 1.2.04, Jan 2,2003, Report # MP 02 W00000101
- [2] Herz, JC, Lucas, Mark, & Scott, John, Open Technology Development: Roadmap Plan v3.1 (Final), Apr 2006.
- [3] Federal Government Policy on the use of Open Source Software, OMB Memorandum M-04-16, subject: Software Acquisition, <http://www.whitehouse.gov/memoranda/fy04/m04-16.html>
- [4] DoD CIO Memo, Open Source software (OSS) in the DoD, May 28, 2003.
- [5] Weber, Steven, The Success of Open Source, Havard University Press, 2004.
- [6] 공개sw 가이드(정통부, '06. 1)
- [7] 공개sw 도입·운영 가이드(국방부, '06. 3)
- [8] 국방전력발전업무규정(국방부훈령 793호, '06. 6)
- [9] C4ISR 연구 및 인력양성체제 연구보고서, 2006. 1. 한국과학기술원, ChungmuIT연구센터
- [10] 발표자료: 육군정보화 발전방향, 2006. 9. 육군본부, 김삼남 장군
- [11] 발표자료: 공개SW 무기체계 활용방안, 2006. 10. 국방대학교, 윤희병 교수
- [12] 발표자료: 데스크탑분야에 대한 공개SW 활용방안, 2006. 12. 고려대학교, 문종섭 교수

부록 1. 공개SW 라이선스 별 비교표

- As of mid 2002, nearly three dozen software licenses qualified as being open source, and thus FOSS, according to the defining criteria¹ of the Open Source Initiative.² In practice, however, only a small number of these licenses are widely used. Furthermore, less frequently used licenses are often based on or closely similar to more commonly used licenses. Table 1 summarizes a number of differences between four of the most important FOSS licenses. Additionally, the table includes the related concept of public domain software and an example of a proprietary software license that is notable for precluding the use of FOSS software.

Property	License:	GPL	LGPL	BSD & MIT	Apache	Public Domain	Microsoft MIT ⁴ EULA
a. Can be stored on disk with other license types		✓	✓	✓	✓	✓	(bans FOSS) ⁵
b. Can be executed in parallel with other license types		✓	✓	✓	✓	✓	(bans FOSS) ⁵
c. Can be executed on top of other license types		✓	✓	✓	✓	✓	(bans FOSS) ⁵
d. Can be executed underneath other license types		✓ ¹	✓	✓	✓	✓	(bans FOSS) ⁵
e. Source can be integrated with other license types			✓	✓	✓	✓	(bans FOSS) ⁵
f. User decides if and when to publish derived code		✓ ²	✓	✓	✓	✓	✓
g. Software can be sold for a profit		✓	✓	✓	✓	✓	✓
h. Binary code can be replicated by users as desired		✓	✓	✓	✓	✓	
i. Binary code can be redistributed as desired		✓ ³	✓	✓	✓	✓	
j. Binary code can be used as desired by users		✓	✓	✓	✓	✓	
k. New users always receive source code of derived works		✓	✓ ⁴				
l. New users receive full source modification rights for derived works		✓	✓ ⁴				
m. New users receive full redistribution rights for derived works		✓	✓ ⁴				
n. Binary code can be released without source code				✓	✓	✓	✓
o. Derived code can have a different type of license			?			✓	
p. Original source can be incorporated into closed source products						✓	

¹ Provided that both programs are fully and independently usable in other unrelated contexts.

² Provided that the binary code has not been previously released to the public.

³ Provided that source code is always redistributed along with the binary code.

⁴ The proprietary Microsoft MIT EULA is not related to the similarly named MIT (X/MIT) license.

⁵ Specifically bans use of: GPL, LGPL, Artistic, Perl, Mozilla, Netscape, Sun Community, and Sun Industry Standards.

⁶ The rights granted by LGPL do not necessarily extend to the applications linked into an LGPL library.

⁷ The LGPL does permit re-licensing under GPL as a special case, but not re-licensing under any other license type.

License Acronyms:

GPL – GNU General Public License
 LGPL – GNU Lesser General Public License
 BSD – Berkeley Software Distribution
 MPL – Mozilla Public License

(Microsoft) MIT – Mobile Internet Toolkit
 (X/MIT) MIT – Massachusetts Institute of Technology
 EULA – End-User License Agreement
 FOSS – Free and Open-Source Software

GPL 라이선스와 보안

- GPL 라이선스(GNU General Public License) 국방분야 공개SW에서 가장 광범위하게 사용되는 라이선스이나 GPL 라이선스는 민감한 소스코드를 노출시킬 위험 요인으로 잘못 인식되고 있으나, 실제조사에 의하면 GPL 라이선스는 보안 그룹과 응용을 위한 아래의 다양한 기능들을 보유하고 있다.
- GPL 사용자권한은 그룹이 새로운 보안위협에 대한 긴급한 조치대응능력을 개발하는데 기여
- 일반적인 오해와 달리, GPL은 binary 코드를 제공할 때까지 대응되는 source 코드 변경을 제공하지 않을 권한을 사용자에게 부여한다. 이 권한은 새로운 위협모드가 적절하게 분석되어 대응방안이 준비될 때까지 핵심적인 변경을 "포장상태"로 유지할 수 있도록 허용한다.
 - GPL 의 활용은 bug 치료에 관한 정보를 공유하게 함으로써 네트워크 전반의 사이버 위협의 기회를 감소시키는데 중요한 역할을 한다
 - GPL 은 GPL 기반을 공유하는 사용자에게 핵심적인 방어적 변경을 신속하고 효과적으로 확산시키는 경로를 제공해준다.
 - GPL 과 Proprietary 와의 다양한 병행 사용이 가능하다.
- 분산 병용 (Distribution Mixing) - GPL 과 다른 software 를 같이 저장/전송 (예: GPL software 와 상용 SW 를 동일한 하드디스크에 저장)
- 운용 병용 (Execution Mixing) - GPL 과 다른 software 를 동일한 컴퓨터나 네트워크상에서 동시에 운용하는 방법 (예: GPL 과 proprietary 응용을 동일한 PC에서 동시에 운용).
- 응용 병용 (Application Mixing) - GPL SW 가 다른 SW 나 SW가 제공하는 서비스(일반적인 OS 서비스이거나 non-GPL 로 선언된)에 의존적일 경우 (예: proprietary 운영체제인 Microsoft Windows Services for UNIX 3.0.위에서 운용되는 GPL 응용)

- 서비스 병용 (Service Mixing) - GPL Sw 가 GPL SW 의 상세설계에 의존적이 아닌 일반적 서비스(Generic Service)를 제공할 경우 (예: Unix pipes)
- 주: GPL source 에서 개발된 모든 SW 는 GPL 라이선스를 가져야 한다.

부록 2. 미 국방성의 공개SW활용 현황 표

- Application Descriptions describes the applications, provides references for finding them on the Internet, and lists the 251 identified instances of DoD use of the applications, including email contact points for each instance. The applications are organized alphabetically by FOSS application. The application descriptions include links back to the main index table to make rapid browsing easier. The information in this table was last updated on August 7, 2002.

Application	Description	License	References
ACE (1 example)	ACE = ADAPTIVE Communication Environment. ACE is a toolkit for creating software to perform common cross-platform network communication tasks. ACE helps create software for demultiplexing, event handler dispatching, signal handling, service initialization, interprocess communication, message routing, dynamic reconfiguration of distributed services, shared memory management, concurrent execution, and process synchronization. The TAO real-time CORBA ORB is a major component of ACE.	ACE/TAO	http://www.cs.wustl.edu/~schmidt/ACE-overview.html http://www.riverace.com/ http://www.theaceorb.com/eroduc/aboutace.html
ACE ORB (TAO) (3 examples)	TAO is a standards-based (CORBA) "Object Request Broker" (ORB) that allows programs located on many networked computers to work together securely and in real-time.	ACE/TAO	http://www.cs.wustl.edu/~schmidt/TAO.html http://www.theaceorb.com/
ACID (1 example)	ACID = Analysis Console for Intrusion Databases. ACID is a PHP-based analysis engine used to search and process databases of security events generated by various intrusion detection systems, firewalls, and network monitoring tools.	GPL	http://www.cert.org/kb/acid/
AMANDA (1 example)	AMANDA = Advanced Maryland Automatic Network Disk Archiver. AMANDA allows a single master backup server to back up large sets of workstations running multiple versions of Unix. AMANDA can also use SAMBA to back up Microsoft Windows 95/NT systems.	BSD	http://www.amanda.org/
Apache (22 examples)	A web server is the software that presents web pages to Internet users. Apache is easily the most popular and widely used web server (open or closed source) on the Internet. It is popular for its reliability, security, range of features, and low cost.	Apache	http://www.apache.org/
Autoconf (1 example)	Autoconf adapts software source code to many kinds of Unix-like systems without manual user intervention.	GPL	http://www.gnu.org/software/autconf/
Automake (1 example)	For software development, Automake generates Makefiles that are compliant with GNU coding standards.	GPL	http://sources.redhat.com/automake/
bash (2 examples)	The default command line interface for Linux. It is used both to create scripts (high level programs), and to interact directly with the operating system.	GPL	http://www.gnu.org/software/bash/bash.html

Application	Description	License	References
Bastille (1 example)	The Bastille Hardening System is a package of adjunct software that can be used to "harden" the Linux operating system. The goal of Bastille is to provide the greatest possible security while keeping the system easy to use. Bastille currently supports the Red Hat and Mandrake Linux distributions, and in late 2002 to early 2003 is also expected to support the Debian, SuSE, and TurboLinux distributions of Linux. Support for the proprietary HP-UX operating system is also planned.	GPL	http://www.bastille-linux.org/ http://sourceforge.net/projects/bastille-linux/
BIND (3 examples)	BIND = Berkeley Internet Name Domain. It is BIND that allows easy-to-use URL text names (e.g., place.com) to be used to identify web sites, instead of the long numeric addresses that the Internet itself uses. Nearly all systems and commercial software that connect to the Internet use BIND.	ISC	http://www.isc.org/products/BIND/
C++ Boost (1 example)	C++ Boost is a web site that provides a broad range of free, portable, high-quality, peer-reviewed C++ source libraries. The site emphasizes compatibility with the C++ Standard Library, and holds many candidates for eventual inclusion in that library.	C++ Boost	http://www.boost.org/
CIS Benchmarks (1 example)	CIS = Center for Internet Security. The CIS Benchmarks are a set of documents that specify in detail how to configure common operating systems for maximum security. An associated collection of freeware Scoring Tools provide automated checks of how closely a given system comes to meeting the Benchmark specifications. The Benchmarks documents are developed and maintained using an easily-to-join community-style (limited FOSS) development process. However, the associated scoring tools are zero-cost freeware, not FOSS, and are provided in binary form only (no source code). (See the HOSTS tool for a FOSS analog to the CIS Scoring Tools.)	Community (CIS)	http://www.cisecurity.org/
Colt (1 example)	Colt is a free collection of high-quality scientific and mathematical software written in Java. It includes software for efficient data structures, data analysis, linear algebra, multi-dimensional arrays, histoprogramming, Monte Carlo simulation, and parallel and concurrent programming. Colt serves as a constantly evolving repository for some of the best concepts and designs for such software.	Colt	http://blde-hoschek.home.cern.ch/~hoschek/colt/
Condor (1 example)	Condor is a computing environment that allows scientists and engineers to harness the capacity of large collections of distributed Unix systems (workstations and PCs running Linux or BSD) to solve processing-intensive problems. Future versions may also work with Windows.	GPL	http://www.cs.wisc.edu/condor/
COPS (1 example)	COPS = Computer Oracle and Password System. COPS analyzes Unix-like systems for weaknesses.	GPL	http://www.tripwire.org/ganda/faq.php
Crack (3 examples)	Crack is used by network admins to verify the quality of user passwords by attempting to break or "crack" those passwords.	Artistic	http://www.uszone.org/diemch/analysis/sec-inctools.html#5.2

Application	Description	License	References
CVS (3 examples)	CVS = Concurrent Versions System. CVS is a popular system for helping software development projects keep track of the history and any multiple versions of the source code they develop. CVS can be used in a wide range of project sizes.	GPL	http://www.cvshome.org/
CVW (3 examples)	Originally developed by MITRE and later released as FOSS. Used by over 5000 people for collaboration. Currently being phased out at NSA in favor of the commercial InfoWorkSpace product from ezenia!, as per directions by Congress and the DoD. The InfoWorkSpace product includes elements of the CVW design.	MITRE	http://cvw.sourceforge.net/
Cygwin (2 examples)	A surprisingly complete Linux-like emulation of Unix and the Unix tool set for use on Windows systems. Cygwin provides access to useful Unix tools and capabilities without requiring users to restart their systems or go to another computer.	GPL	http://www.cygwin.com/
DDD (1 example)	GNU DDD is a graphical front-end for debugging tools. It is noted for its ability to display debugging data in a convenient chart format.	GPL	http://www.gnu.org/software/ddd/
DjVuLibre (1 example)	DjVuLibre provides efficient distribution and display of images in a variety of compressed formats.	GPL	http://djvu.sourceforge.net/
EADSIM (1 example)	EADSIM = Extended Air Defense Simulation. Combat developers, materiel developers, and operational commanders use EADSIM simulations to assess the effectiveness of Theater Missile Defense (TMD) and air defense systems against a full spectrum of extended air defense threats. It is provided without charge under a restricted community (versus FOSS) license.	Community (EADSIM)	http://www.eadsim.com/EADSIMRochure.html
Emacs (6 examples)	A full-functioned and popular editing tool that is especially useful for creating software. There are multiple "families" of Emacs, such as GNU Emacs and X-Emacs, for use in different environments. Emacs also supports language-specific extensions that are widely used for development in languages such as Java and C.	GPL	http://www.gnu.org/software/emacs/emacs.html
eTrust (1 example)	eTrust is a closed source network security evaluation and monitoring tool with FOSS (OpenSSL toolkit) origins.	Closed from open (eTrust)	http://www3.ca.com/Solutions/Solution.asp?ID=271
Expect (1 example)	Expect is a system admin and user tool for automating and testing interactive Unix applications such as telnet, ftp, passwd, fsck, and rlogin. It can be used to greatly simplify and automate tasks that would be prohibitively time consuming and costly if done interactively by people.	Public Domain (Expect)	http://expect.nist.gov/
FreeBSD (3 examples)	The Unix-like operating BSD operating systems are FOSS competitors to Linux, and are notable for having generally higher levels of reliability and security. OpenBSD, NetBSD, and FreeBSD are best known. FreeBSD is notable for being highly efficient when used on PC (Pentium) computers.	BSD	http://www.freebsd.org/ http://www.daemonnews.org/199904/editorial.html http://www.openbsd.org/ http://www.netbsd.org/
GateD (1 example)	GateD provides network routing services, a routing database, and support for a variety of routing protocols.	Closed from open (GateD)	http://www.nexthop.com/products/gated.shtml (current closed version) http://www.merit.edu/internet/net-research/irp/mitre/doc/gated_doc/main.html (earlier FOSS version)

Application	Description	License	References
gawk (awk) (2 examples)	gawk = GNU awk (Aho, Weinberger, Kernighan - the authors of awk). Gawk is the GNU version of the awk file transformation language. Awk is an interpreted C-like language with strong pattern matching and capabilities, making it useful for writing quick programs to make minor transformations on files. For larger or more frequently used file transformations, Perl is usually a better choice than gawk, since the more recent Perl provides similar capabilities plus a number of advanced features.	GPL	http://www.gnu.org/software/gawk/gawk.html
GCC (9 examples)	GCC = GNU Compiler Collection (formerly GNU C Compiler). GCC is a suite of compilers that includes C, C++, Objective C, Chill, Fortran, Java, and (in the next release) GNAT Ada. The original GNU C compiler dominates the C software development market.	GPL	http://gcc.gnu.org/
GDB (1 example)	GDB = GNU Project Debugger. GDB, the GNU Project debugger, allows you to see what is going on inside another program while it executes, or what another program was doing at the moment it crashed. GDB can be used to start your program with any options you want, stop your program when specified conditions occur, examine the state of your program after stopping it, and change your program temporarily to examine the effects of possible fixes.	GPL	http://sources.redhat.com/gdb/
Ghostscript (2 examples)	Ghostscript, along with its associated graphical interface tools Ghostview and GSview, provides viewing of postscript and PDF documents.	AFPL	http://www.cs.wisc.edu/~ghost/
GNAT (9 examples)	GNAT is a FOSS implementation of Ada 95. Commercial versions of GNAT (GNAT Pro Ada 95) and support are provided by Ada Core Technologies, and fully FOSS (GPL) versions are also available. A GNAT Ada front-end will also be added to GCC in GCC 3.1.	GPL	http://www.gnat.com/ ftp://cs.nyu.edu/pub/gnat/ http://gcc.gnu.org/
GnuPG (1 example)	GnuPG stands for GNU Privacy Guard and is GNU's tool for secure communication and data storage. GnuPG is a complete and free replacement for PGP. It can be used to encrypt data and to create digital signatures, and it includes an advanced key management facility. Because it does not use the patented IDEA algorithm, it can be used without any restrictions. GnuPG is a RFC2440 (OpenPGP) compliant application, providing compatibility with PGP from NAI Inc.	GPL	http://www.gnupg.org/
gnuplot (3 examples)	gnuplot is a command-driven interactive function plotting program that can be used to plot functions and data points in two or three dimensions and many different formats. It is free, but not GPL, despite its name.	Gnuplot	http://www.gnuplot.info/
grep (1 example)	GNU grep can be used to search text files or text streams for lines that match simple or complex patterns.	GPL	http://www.gnu.org/software/grep/grep.html
h2n (1 example)	h2n converts a table of host computers into a form usable as Internet (DNS) names.	GPL	http://www.crihan.fr/system/linux/maint/bind/old/h2n-man.html
HOSTS (1 example)	HOSTS = Host-Oriented Security Test Suite. Provides greater consistency and repeatability in security testing of Unix and Unix-like operating systems by automating many aspects of the testing process.	GPL	http://www.openchannelfoundation.org/projects/HOSTS

Application	Description	License	References
ImageMagick (1 example)	ImageMagick provides display and conversion of images in about 70 major formats.	ImageMagick	http://www.imagemagick.org/
JADE (1 example)	JADE = Java Agent Development framework. JADE provides Java middleware for creating "multi-agent" software that on multiple networked machines. JADE implements FIPA agent communication standards.	LGPL	http://sharon.oselt.it/projects/jade/
Jakarta (1 example)	Jakarta is a web site that provides FOSS Java solutions for a wide range of applications and problems.	Apache	http://jakarta.apache.org/
Jaxen (1 example)	Jaxen = Java XPath Engine. Jaxen is a Java and XML development tool that interprets XPath expressions for multiple XML models, including DOM, dom4j, EXML, and JDOM.	Apache	http://jaxen.org/
JBoss (1 example)	JBoss is a J2EE-compliant web application server that provides middleware capabilities (EJB and JMS), database connectivity (JDBC), transactions (JTA/JTS), presentation (servlets and Java Server Pages), and directory services (JNDI). In March 2002, Sun Microsystems expressed stronger support for getting the popular JBoss package Java certified.	LGPL	http://jboss.org/
JDOM (1 example)	JDOM provides a fast, easy-to-read way to represent XML documents in Java. (JDOM is a name, not an acronym.)	GPL	http://www.jdom.org/
Jikes (1 example)	Jikes is a FOSS compiler for Java.	JPL	http://oss.software.ibm.com/developmentworks/opensource/jikes/
jSIP (1 example)	jSIP = Java Session Initiation Protocol. The jSIP library provides text-based collaboration by users, including Instant Messaging.	GPL	http://jsip.sourceforge.net/
Kaffe (1 example)	Kaffe is a FOSS implementation of the Java Virtual Machine (JVM), which is the software that interprets Java software. (Sun JVMs are free but not FOSS.)	GPL	http://www.kaffe.org/
LaTeX (2 examples)	LaTeX (pronounced "lay-tek") is a high-quality typesetting system, with features designed for the production of technical and scientific documentation. It is the de facto standard for writing and publishing scientific documents.	LaTeX	http://www.latex-project.org/
Linux (19 examples)	Linux is a popular Unix-like FOSS operating system. It contains hundreds of individual tools, and has more commercial and applications support than any other FOSS operating system.	GPL	http://www.linux.org/
Linux (Red Hat) (3 examples)	Red Hat is the most popular commercial source for the Linux operating system.	GPL	http://www.redhat.com/
Linux firewalls (1 example)	Linux provides a variety of tools for creating firewalls.	GPL	http://www.linuxjournal.com/article.php?sid=1212 http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html
Lsof (1 example)	Lsof = List Open Files. Lsof lists any currently open files or process communications.	Lsof	http://freshmeat.net/projects/lsof/
m4 (1 example)	GNU m4 is a "macro expander" that can be used to create large sets of source code (such as web pages) with a shared format or visual look and feel.	GPL	http://www.gnu.org/software/m4/
Majordomo (1 example)	Majordomo automates management of Internet mailing lists. Once a list is set up, nearly all operations can be performed remotely by email. A graphical user interface called MajorCool is also available.	GPL	http://www.greatcircle.com/majordomo/

Application	Description	License	References
make (2 examples)	The GNU make utility automatically determines which pieces of a large program need to be recompiled, and issues the commands to recompile them.	GPL	http://www.gnu.org/manual/make-3.79.1/html_node/make_toc.html
Maxima (1 example)	Maxima is a Common Lisp implementation of MIT's Macsyma system for computer based algebra.	GPL	http://www.ma.utexas.edu/users/wfs/maxima.html
MIMEsweeper (1 example)	MIMEsweeper is a closed source product with FOSS origins. It looks for suspicious patterns in the actual content of emails and communications to help identify suspicious activities.	Closed from open (MIMEsweeper)	http://www.mimesweeper.com/default.asp
MRTG (3 examples)	MRTG = Multi Router Traffic Grapher. MRTG provides monitoring of traffic load on network links, and shows the live status graphically using HTML images that can be viewed over the Internet.	GPL	http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg.html
MTR (1 example)	MTR determines whether a network computer is available, and the overall quality of the link to it.	GPL	http://www.bitwizard.nl/mtr/
MySQL (4 example)	MySQL is the world's most popular FOSS database. It is fast, full-functioned, and precise enough to be used in both heavy load and mission critical applications.	GPL	http://www.mysql.com/
Nessus (3 examples)	An easy-to-use, full-functioned, and up-to-date remote security scanner.	GPL	http://www.nessus.org/
NetBSD (1 example)	The Unix-like operating BSD operating systems are FOSS competitors to Linux, and are notable for having generally higher levels of reliability and security. OpenBSD, NetBSD, and FreeBSD are best known. NetBSD is notable for being highly portable across a wide range of computer platforms.	BSD	http://www.netbsd.org/ http://www.daemonnews.org/199904/editorial.html http://www.openbsd.org/ http://www.freebsd.org/
NetSaint (1 example)	NetSaint monitors network Linux hosts services and can alert administrators of problems via email when a problem arises.	GPL	http://www.netsaint.org/
nload (1 example)	nload monitors and graphically displays real-time network traffic and usage.	GPL	http://www.roland-nieel.de/nload/index_en.html
Nmap (2 examples)	Nmap scans networks and maps out their configurations.	GPL	http://www.insecure.org/nmap/
ntop (2 examples)	ntop is a Unix tool that shows the heaviest users of network resources in ranked order, making it easy to see hot spots or anomalous usage.	GPL	http://www.ntop.org/ntop.html
NTP (1 example)	NTP = Network Time Protocol. NTP software provides the ability to synchronize in network computer clocks precisely.	GPL	http://www.eecis.udel.edu/~ntp/ http://www.eecis.udel.edu/~mills/ntp.htm http://www.eecis.udel.edu/~ntp/ntp_spool/html/ntpd.htm
Octave (1 example)	GNU Octave is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. It may also be used as a batch-oriented language.	GPL	http://www.octave.org/

Application	Description	License	References
OpenBSD (1 example)	The Unix-like operating BSD operating systems are FOSS competitors to Linux, and are notable for having generally higher levels of reliability and security. OpenBSD, NetBSD, and FreeBSD are best known. OpenBSD is notable for its high security, support for encryption, and an exceptionally rigorous self-auditing process. OpenBSD has been particularly successful at avoiding the kinds of default security holes commonly encountered when installing most operating systems.	BSD	http://www.openbsd.org/ http://www.daemonnews.org/1999-04/editorial.html http://www.freebsd.org/ http://www.netbsd.org/
OpenMap (2 examples)	OpenMap is JavaBeans-based programmer's toolkit that allows Java application to access map data from older databases and formats.	OpenMAP	http://openmap.kbn.com/
OpenOffice (1 example)	OpenOffice is a suite of business office support programs comparable to Microsoft Office, but based on the open and easily exchanged XML format. OpenOffice began as a free but closed-source system called StarOffice, which was bought by Sun Microsystems, who eventually made it fully FOSS. As of early 2002, OpenOffice was still undergoing the transition from closed to FOSS.	GPL	http://www.openoffice.org/
OpenSSH (1 example)	OpenSSH = Open Secured Shell. OpenSSH provides secure (encrypted) access to remote network computers.	BSD	http://www.openssh.com/
OpenSSL (1 example)	OpenSSL is a FOSS implementation of the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols for secure communications over the Internet. It includes a full-strength, general-purpose library of cryptography software.	OpenSSL	http://www.openssl.org/
Perl (25 examples)	Perl = Practical Extraction and Reporting Language. A popular, functionally rich Internet language that is used in a wide range of applications that include extracting data from text, reformatting documents, and integrating software components.	GPL	http://www.perl.org/
Perl CGI scripts (1 example)	Perl scripts provide numerous functions to support web sites, including various types of search.	GPL	http://www.scriptsearch.com/Perl/Scripts_and_Programs/ http://awsd.com/scripts/
PerLDAP (2 examples)	The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing online directory services. PerLDAP is a Perl implementation of it.	GPL	http://www.perldap.org/
PHP (4 examples)	PHP = PHP Hypertext Preprocessor. PHP allows web pages to interact with users (e.g., to accept and display form data).	PHP	http://www.php.net/
PingScan (1 example)	PingScan scans networks to find all accessible systems.	GPL	http://www.linux.org/apps/Appleid_1996.html
Procmal (1 example)	Procmal supports lists and automated pre-processing of email, such as sorting, selecting, and re-routing emails based on various criteria and conditions.	GPL	http://www.procmal.org/
Qmail (1 example)	Qmail is a FOSS replacement for Sendmail, the program that transfers emails between computers on the Internet. Qmail has improved security, reliability, and performance features.	Qmail	http://qmail.groff.com/top.html

Application	Description	License	References
R (1 example)	R is a language and environment for statistical computing and graphics. It provides a wide variety of statistical and graphical techniques such as linear and nonlinear modeling, statistical tests, time series analysis, classification, and clustering. It is also known as GNU S, a reference both to its use of the GPL and its similarity to the S statistical language.	GPL	http://www.r-project.org/
RealSecure (1 example)	RealSecure is a closed source intrusion detection product with FOSS origins.	Closed from open (RealSecure)	http://www.iss.net/products_services/enterprise_protection/rsnetwork/index.php
RRDtool (2 examples)	RRDtool = Round Robin Database tool. RRDtool provides efficient collection of network usage data over extended periods of time.	GPL	http://people.ee.ethz.ch/~roetiker/webtools/rrdtool/
RTLinux (1 example)	RTLinux allows Linux and BSD operating systems to respond reliably to time-critical applications such as embedded device control, instrumentation, and certain types of communications.	RTLinux	http://www.fsmlabs.com/communitv/
RWhois (1 example)	RWhois provides improved administrative identification of users on a network.	GPL	http://www.rwhois.net/
RXVT (1 example)	A smaller, more compact way to interact via command lines with programs.	GPL	http://www.rxvt.org/ http://www.math.fu-berlin.de/~quackes/rxvt/#intro
Samba (3 examples)	Samba is a popular tool that allows Linux and BSD (OpenBSD, NetBSD, and FreeBSD) operating systems to provide invisibly the same file and printer services as Windows servers. Since the Linux and BSD operating systems were generally more stable than early Windows NT servers, administrators often invisibly converted Windows servers to Linux-plus-Samba to improve network reliability.	GPL	http://us1.samba.org/samba/samba.html
SARA (1 example)	SARA = Security Auditor's Research Assistant. SARA is a third generation Unix security analysis tool that is based on the SATAN model.	SATAN	http://www.arc.com/sara/
SATAN (2 examples)	SATAN = Security Administrator Tool for Analyzing Networks. It is a first generation Unix security analysis tool that collects data on networked hosts.	SATAN	http://www.wazone.org/demch/analysis/sec-inchtools.html#5.1
Saxon (1 example)	SAXON provides tools for processing XML. (XML is the successor to the HTML used in most Internet web pages.) It includes an XML standards-compliant XSLT processor, plus a number of useful extensions, such as a Java library that provides XSL-like processing.	MPL	http://saxon.sourceforge.net/
SCA (1 example)	SCA = Software Communications Architecture. This web site provides standards for writing code for software-defined radio applications. This is an early effort to promote sharing by providing an overall standards framework through which contributed software can work and interoperate.	Community (SCA)	http://www.jtrs.saaft.army.mil/docs/documents/sca.html
sed (2 examples)	sed = Stream Editor. GNU sed can be used to extract or transform text in very large files, or in incoming or outgoing streams of text data of indefinite length. Perl and awk (gawk) both provide more functionality, but for simple filtering and conversions, sed is both fast and easy to use.	GPL	http://www.gnu.org/software/sed/sed.html http://www.dreamwv.com/sed-info/sed-faq.html

Application	Description	License	References
SELinux (2 examples)	SELinux = Security Enhanced Linux, a set of Linux enhancements developed specifically by NSA to make Linux usable in a broader range of government and industry applications. (Note: In contrast, NSA Signal Intelligence prohibits use of Linux.)	GPL	http://www.nsa.gov/selinux/
Sendmail (1 example)	Sendmail takes care of the actual transfer of email messages between Internet computers. Sendmail is the most widely used such program on the Internet. Qmail provides a more security-focused FOSS alternative.	Sendmail	http://www.sendmail.org/
SNARE (1 example)	SNARE = System INtrusion Analysis and Reporting Environment). An auditing and intrusion detection module that can be attached directly to the Linux kernel.	GPL	http://www.intersectalliance.com/projects/Snare/
Snort (3 examples)	Snort is a multi-platform, lightweight, rule-based tool for detecting hostile intrusions into a network. It works well on small networks, and can be deployed quickly to help fill in network security holes when new attacks emerge.	GPL	http://www.snort.org/ http://www.snort.org/docs/lisaaop/er.txt
Squid (1 example)	Squid improves web performance for Unix and Unix-like systems by invisibly providing local copies (caching) of frequently used files and information from remote parts of the web. It supports full-featured proxying (that is, invisible replacement of requests for files from remote sites with copies of the same information previously stored locally) and caching for most of the major web protocols and formats, including HTTP, FTP, and web site names (URLs), and also proxying for SSL.	GPL	http://www.squid-cache.org/
Tcl/Tk (1 example)	Tcl is a scripting language for controlling computer devices, and Tk is a library for creating graphical interfaces to those parts.	BSD	http://www.tcl.tk/
TCP Wrappers (2 examples)	Provides monitoring and filtering of incoming requests for network services, including sysstat, finger, ftp, telnet, rlogin, rsh, exec, ftp, and talk.	TCP Wrappers	ftp://ftp.porcucine.org/pub/security/tcp_wrappers_7.6.BLURB ftp://ftp.porcucine.org/pub/security/index.html
Tomcat (1 example)	Tomcat is a FOSS implementation of the official "servlet container" for Java Servlets and JavaServer Pages.	Apache	http://jakarta.apache.org/tomcat/index.html
Top (1 example)	Top is a standard Unix (Linux and BSD) tool for determining which processes are consuming the most processing resources.	GPL	http://www.tac.eu.org/cgi-bin/man.cgi?top+1
Tripwire (2 examples)	Tripwire monitors key attributes of files that should not change and provides alerts when they do change.	GPL	http://www.tripwire.org/gandalf/faq.php
VisAD (1 example)	VisAD = Visualization for Algorithm Development. VisAD is a Java library for interactive and collaborative visualization and analysis of numerical data.	LGPL	http://www.ssec.wisc.edu/~billh/visad.html
VOCAL (1 example)	VOCAL = Vovida Open Communication Application Library. VOCAL provides tools and software for building advanced Internet telephony (VoIP) applications.	Vovida	http://www.vovida.org/applications/downloads/vocal/home.html
VTK (1 example)	VTK = Visualization Toolkit. VTK provides 3D computer graphics, image processing, and visualization. It has interfaces to most of the major Internet computer languages, and is used by thousands of researchers and developers around the world.	VTK	http://public.kitware.com/VTK/

Application	Description	License	References
Webmin (1 example)	Webmin makes it possible to do web-based remote or local system administration of Unix and Unix-like systems. Using any browser that supports tables and forms (and Java for the File Manager module), you can setup user accounts, Apache, DNS, file sharing, and other common system admin tasks. The web server part of Webmin is written in Perl, using only standard Perl modules.	BSD	http://www.webmin.com/
WebTAS (1 example)	WebTAS = Web-enabled Timeline Analysis System. WebTAS provides tools for analyzing data and looking for suspicious patterns in the data of both large and small organizations, particularly law enforcement and security agencies. It is free under a limited access (community) license.	Community (WebTAS)	http://www.webtas.com/ http://iswsolutions.com/webtas.shtml
Weka (1 example)	Weka is a collection of machine learning algorithms for solving real-world data mining problems. It is written in Java and runs on almost any platform.	GPL	http://www.cs.waikato.ac.nz/~ml/weka/
WU-FTP (1 example)	WU-FTP provides the ability to transfer files easily ("FTP") between computers on the Internet. It is the most widely used program for providing FTP capabilities.	WU-FTP	http://www.wu-ftpd.org/
Xalan (1 example)	Xalan-Java and Xalan-C++ are XSLT-based tools for converting XML documents into HTML, text, or other XML document types.	Apache	http://xml.apache.org/xalan-j/ http://www.garshol.priv.no/download/xmltools/erod/Xalan-C.html
Xerces (3 examples)	Xerces interprets (parses) XML, which is the successor to HTML. Xerces is available for Java, C++, and Windows.	Apache	http://xml.apache.org/
XFree86 (1 example)	XFree86 is a FOSS version of the X windowing system used in most Unix-like systems, including Linux and the BSD operating systems. It provides easy-to-use, PC-like graphical displays and controls for computer users.	Xfree86	http://www.xfree86.org/
XGobi (1 example)	XGobi is a data visualization system for viewing high-dimensional data. The most recent version is called GGobi. GGobi components include four FOSS licenses: AT&T Open Source License , GPL , BSD , and LGPL .	AT&T Open Source License	http://www.ggobi.org/ ; http://www.research.att.com/areas/stat/ggobi/
Xpatch (1 example)	Xpatch provides tools for predicting the likely radar signatures of both nearby and distant objects.	Community (Xpatch)	http://www.saic.com/products/software/xpatch/
zlib (6 examples)	A library of FOSS compression software with a license that permits use in closed source products.	zlib	http://www.gzip.org/zlib/
Zope (1 example)	Zope is a web application server used to create web-based applications such as intranets and portals.	ZPL	http://www.zope.org/