

소프트웨어 자산관리를 위한 패키지소프트웨어 점검서비스 구현

차 태 원[†] · 안 재 경^{††}

요 약

국제적인 규제와 정부차원의 지원과 관리가 이뤄지고 있음에도 불구하고 일반기업 등에서는 여전히 소프트웨어자산에 대한 인식과 관리 소홀의 문제점을 안고 있다. 이는 관리해야 할 소프트웨어자산의 범위가 모호할 뿐 아니라 소프트웨어의 특성상 복사 및 설치가 쉽고 눈에 보이지 않아 관리 자체가 어렵기 때문이다. 즉 기업의 소프트웨어관리자는 구매부터 조달, 배포, 유지, 처분까지의 전체적인 소프트웨어 라이프사이클의 각 단계별로 수시로 변화되는 정보를 정확히 확인하고 관리하기가 쉽지 않다는 문제점이 있다. 따라서 본 논문에서는 이를 해결하기 위해서 개발된 점검용 시스템인 Inspector를 활용할 것을 제안한다. 제안된 점검서비스에는 패키지 소프트웨어의 특성을 고려한 검색기법이 구현되었고, 소프트웨어의 권리관리 정보를 표준화하였으며, 레지스트리 정보를 활용하였다. Inspector를 사용한 결과 PC 1대당 점검시간이 획기적으로 단축되었다. 이러한 효과 이외에도 소프트웨어를 효과적으로 관리함으로써 소프트웨어 구입비용을 절감할 수 있도록 지원하여 경영합리화에 이바지하는 것으로 나타났다.

키워드 : 소프트웨어 자산관리, 소프트웨어 불법복제, 소프트웨어 저작권

Realization of Package Software Inspection Service for Software Asset Management

Tea Won Cha[†] · Jaekyoung Ahn^{††}

ABSTRACT

Even support-and-management efforts by government are enforced with the growing international restrictions for complying with global standard, not a few private enterprises are still suffering software management problems from lack of understanding the software asset management and carelessness of the software management. This is due to the ambiguous scope of software asset management as well as characteristics of software property such as invisibility, easiness of copy and installation. In other words, it is not easy for software managers to comply all the information throughout the software life cycle; from purchasing, procurement, distribution, preservation to disposition. Therefore, Inspector which has been developed for resolving these problems are proposed in this study. The inspection service consists of search algorithms adapted for software characteristics, standardization of software right management information, and practical use of registry information. Inspection time for a PC by using inspector make the inspection time much shorter. It is also shown that effective software management enables the enterprise to curtail software acquisition cost and to help rationalizing its business management.

Keywords : Software Asset Management, Software Piracy, Software Copyright

1. 서 론

소프트웨어산업은 지식정보사회의 기반산업으로서 타 산업의 생산성 및 효율성을 증대시키고 경쟁력을 획기적으로 개선시키는 전략산업으로서 우리나라가 지속적으로 육성·발전시켜야 할 분야이다. 그러나 소프트웨어 불법복제로 인하여 관련 산업발전이 심각하게 저해되고 있으며, 통상문제가 점점 심화되고 있다. 이는 소프트웨어 특성상 복사와 설치

가 손쉽고 눈에 보이지 않아 소프트웨어를 관리하는 담당자가 이를 일일이 확인을 하고 저작권 침해여부를 파악 하는데 많은 어려움이 있으며, 소프트웨어를 하드웨어와 같은 자산으로 인식하는 경우가 미흡한 것이 불법복제의 주요 원인으로 파악되고 있다.

일부 기업의 CEO들은 정부의 지속적인 소프트웨어불법복제 단속으로 정품소프트웨어사용에 많은 노력을 기울이고 있으나 관리체계가 미흡하여 인터넷 등의 경로를 통해 손쉽게 입수한 소프트웨어를 무분별하게 설치하고 방치함으로써 CEO 및 관리자의 의지와는 상관없이 소프트웨어저작권을 침해하고 있다. 또한 정보화·전문화된 업무여건 변화로 사무용 소프트웨어의 활용 비중이 증가함에 따라 IT자산비용

[†] 정 회 원 : 서울산업대학교 산업정보시스템전공 박사과정
^{††} 정 회 원 : 서울산업대학교 산업정보시스템공학과 교수
논문접수: 2008년 7월 14일
수 정 일: 2008년 9월 25일
심사완료: 2008년 12월 13일

이 기존 하드웨어에서 소프트웨어 중심으로 옮겨가는 추세이며, 새로이 중요자산으로 부각되는 소프트웨어의 효율적인 관리능력 보유여부가 전체기업 관리능력의 중요한 사항으로 부각되고 있다.

특히 최근 타결된 한·미 FTA 협상에서는 소프트웨어지 재산권과 관련하여 공공기관에 자발적인 소프트웨어관리 규정 및 책임을 의무화하도록 하였으며[9] 2006년 5월에는 ISO(국제표준화 기구)에서 ISO/IEC 17990-1로 규정된 소프트웨어자산관리 기준을 정식으로 발표하였다. 이는 소프트웨어 자산관리에 대한 관심과 중요성이 점차 커지고 있다는 것을 의미하며, 국제표준에 맞추어 소프트웨어자산 관리를 해야 할 시점이 되었음을 알리고 있다.

하지만, 국제적인 규제와 정부차원의 지원 및 관리가 이루어지고 있음에도 불구하고 여전히 소프트웨어자산에 대한 인식과 관리 소홀의 문제점을 안고 있다. 이는 관리해야 할 소프트웨어자산 범위의 모호함과 소프트웨어의 특성상 복사 및 설치가 쉽고 눈에 보이지 않아 관리 자체가 어렵기 때문이다.

따라서 본 논문에서는 상업용 패키지소프트웨어를 효과적으로 검색할 수 있는 알고리즘을 연구하여 이를 구현함으로써 조직 내 PC에 설치된 소프트웨어 자산을 효율적으로 관리할 수 있는 방안을 제시하고, 그 효과를 조사하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 문헌조사로써 소프트웨어 자산관리의 효과 및 표준화 등에 대한 기존문헌을 살펴보고 3장에서는 기업 내 소프트웨어 관리절차 및 문제점을 고찰한 후 소프트웨어자산을 식별할 수 있는 검색기법을 제시한다. 4장에서는 제시된 검색기법을 기반으로 패키지 소프트웨어 점검서비스를 구현하고 이를 통한 사용효과 등을 제시한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해서 언급한다.

2. 관련 문헌 및 표준화동향

2.1 소프트웨어 자산관리효과 관련

기업의 IT예산의 규모가 증가하면서 IT투자대비 평가와 관련된 연구와 IT 자산관리의 효과에 대한 보고서가 등장하기 시작하였는데 대부분의 학술적인 연구는 IT투자에 대한 전체적인 차원에서의 유·무형 효과 평가 및 방법론과 관련된 연구가 대부분이며, 소프트웨어자산관리의 측면에서는 학술연구 보다는 미국의 다국적 소프트웨어 저작권사의 의뢰를 받아 IDC등 컨설팅업체가 주도가 되어 진행되고 있다.

IDC는 소프트웨어 비용은 IT예산의 25%를 차지하며 효과적인 라이프사이클 자산관리를 통해 25%의 비용을 절감할 수 있다는 결론을 이끌어 내었고, 메타그룹 역시 효과적인 라이프사이클 자산 관리를 통해 25%의 소프트웨어 비용을 절감할 수 있다는 연구결과를 발표한 바 있다.[5] 가트너 그룹은 IT자원들의 라이프사이클 통합 관리로 IT자원의 수명이 업계 평균 15%~20%까지 연장이 가능하고, 자산 추적과 활용 적합성, 편리한 자원 감사로 라이선스 리스크가 80~90% 감소할 것이라는 결과를 발표하였다. 또한 관리되지

〈표 1〉 2008년 BSA 주요국의 소프트웨어 불법복제율

	2001	2002	2003	2004	2005	2006	2007
한국	48%	50%	48%	46%	46%	45%	43%
일본	37%	35%	29%	28%	28%	25%	25%
중국	92%	92%	92%	90%	86%	82%	82%
미국	25%	23%	22%	21%	21%	21%	20%
세계평균	37%	40%	39%	36%	35%	35%	35%

않는 윈도 XP 데스크톱 PC의 3년 총 소유경비가 5천309달러이며, 같은 데스크톱이 잘 관리됐을 때 3천 335달러가 된다고 발표한 바 있다.[11] 공통적으로 소프트웨어 TCO 절감의 핵심 사항은 효과적인 소프트웨어자산관리 체제를 마련하는데 있다고 주장하고 있다. 한편 “다수의 기업에 문제를 남기는 소프트웨어 라이선스 관리”보고서[1]에 따르면 기업의 소프트웨어 자산 관리 실무에 대한 인식은 확산 중이지만 소프트웨어 라이선스 위반에 따른 리스크를 관리하는데 있어서는 적절한 활동이 미흡한 것으로 나타나 효과적인 관리를 위해 소프트웨어 사용정보 모니터링 Tool의 필요성을 제기하고 있다.

이러한 결과들을 종합하면 소프트웨어자산관리는 재무적 관점, 위기관리 관점, 업무프로세스 관점으로 효과를 측정하고 있으며 재무적 관점은 소프트웨어비용 절감과 유지보수 비용의 절감으로부터 TCO절감 및 ROI극대화를 이끌어내는 것이다. 즉 소프트웨어를 자산으로 인식하고 체계적으로 관리하게 되면 필요한 소프트웨어만을 계획적으로 구입하여 불필요한 비용손실을 막을 수 있으며, 보다 적극적인 소프트웨어 자산관리는 벤더와의 관계에서 Negotiating Power를 증진 시킬 수 있고 다양한 서비스와 지원을 이끌어 낼 수 있다. 또한 허가 받지 않은 프로그램들로 인한 바이러스와 보안상 문제, 그리고 이를 통해 발생하는 네트워크와 시스템 및 인력 손실을 대비 할 수 있다. 결과적으로 소요비용뿐 아니라 소프트웨어의 교육, 유지, 분배, 관리 등 눈에 보이지 않는 비용을 최대한 절감하여 경영 효율을 높일 수 있다. <표 1>는 올해 IDC에서 발표한[6] 주요국의 연도별 소프트웨어 불법복제율로써 우리나라의 경우 2007년 소프트웨어 불법복제율이 세계평균인 35%보다 높은 43%로서 소프트웨어 불법복제를 근절하기 위해 정부차원의 단속활동 강화 등 강력한 지적재산권 보호정책을 수행해 오고 있다. 따라서 기업이 정부의 이러한 단속활동 등에 대처하기 위한 위기관리 관점에서 불법소프트웨어사용을 사전에 방지함으로써 불법소프트웨어사용으로 인한 법적인 위협과 이와 관련된 저작권자의 합의금 지불의 위기를 막아내어 기업 이미지를 지켜내야 할 것이다.

2.2 소프트웨어 자산관리 문제점 및 IT자산관리 Tool현황

SW자산관리(Software Asset Managemenet)란 SW를 관리하기 위한 정책을 수립하는 것부터 계약조건을 연구하고 구매하는 것과 사용 현황을 조사 하는 것, 적절하게 배포하고 관련자원을 지원하는 것의 전 과정을 포함하는 개념[3]으

〈표 2〉 소프트웨어 자산관리 주요 문제점

구분	SW자산관리 주요 문제점
SW특성 요인	- HW와 달리 Version Upgrade는 있으나 폐기되는 경우가 거의 없음 - 설치된 SW는 시간이 경과할 수록 실체를 파악하기가 어려움 - 구매 → 입고 → 자산등록 → 설치 → 활용 → 증설 → 폐기(혹은 재사용)등의 생명주기 과정에서 수시로 변화하는 정보를 실제 시스템(혹은 장부)의 기록들과 정확 일치 시키기가 어려움
기업 내부적 요인	-자산관리와 실사 시 발생하는 비용에 대한 민감성 -거짓정보 입력 가능 및 잦은 인수 합병
기업 외부적 요인	-기술트렌드 및 라이선스 계약 방식이 수시로 변경 될 때 -SW제공 업체별로 SW재사용 및 유지보수 제약 조건이 상이

〈표 3〉 국내 주요 IT 자산관리 Tool현황[9][10][11]

구분	TCO!stream	AutoDeploy	NetClient
개발사	(주)티씨오솔루션	(주)아이젝스	(주)닥터소프트
주요기능	<ul style="list-style-type: none"> · 보안관리 · 자산관리 · 불법소프트웨어관리 · 소프트웨어 배포 · 원격제어 · 소프트웨어 미터링 · 보고서 작성 및 통계분석 	<ul style="list-style-type: none"> · 자산관리 · 소프트웨어 배포 · 원격제어 및 감시 · IP/PC관리 · 패치 및 보안관리 · 라이선스관리 	<ul style="list-style-type: none"> · 라이선스 관리 · 보안관리 · 백신관리 · 정책관리 · 원격/장애관리 · 배포관리 · 보고서/통계 · 이력/구성 관리

로서 효과적인 소프트웨어 자산관리를 위해서는 소프트웨어의 구매부터 조달, 배포, 유지, 처분까지의 전체적인 소프트웨어 라이프사이클의 각 단계별로 수시로 변화되는 정보를 실제 시스템 기록과 정확히 일치 시켜야한다는 문제점이 있으며 특히 <표 2>에 제시한 바와 같이 소프트웨어의 특성 요인 및 기업의 내·외부적인 요인 등으로 관리의 어려움이 증대되고 있다.

이를 위해 기업에서는 새로운 IT시스템 구축에 대한 노력과 함께 이미 구축된 소프트웨어를 포함한 IT자산을 어떻게 효율적으로 사용하고 관리할 것인지에 관한 자산관리 Tool 도입에 관한 관심이 증대되고 있다.<표 3>은 국내 주요 IT자산관리 Tool현황으로 대부분 소프트웨어의 효율적 배포와 더불어 보안관리, 원격제어 및 감시 기능을 주된 기능으로 하고 있으며, 소프트웨어 자산관리를 위한 검색의 경우 3개 제품 모두 관리대상 소프트웨어를 사전에 입력 한 후 검색 및 필터링 기법을 이용하여 관리할 수 있도록 하는 기능을 제공하고 있으나, 설치되어 사용 중인 소프트웨어의 사용허락조건(상업용 프로그램인지 여부 등)을 관리자가 쉽게 파악하기에는 어려움이 있으며, 또한 검색 기법 및 소프트웨어의 권리관리정보의 표준화의 부재로 불법소프트웨어 사용여부를 정확히 파악하기에는 한계가 있다.

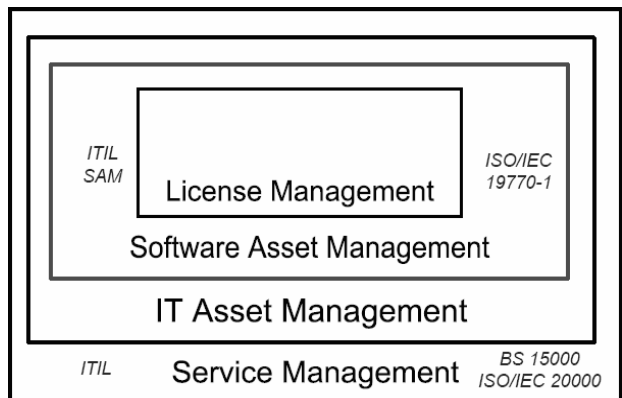
2.3 소프트웨어 자산관리 표준화 동향

ISO/IEC 19770-1 (Information technology Software asset management Part 1: Process)은 2006년 5월에 ISO/IEC JTC1/SC7 WG21에서 제정한 소프트웨어자산관리관련 국제 표준으로 소프트웨어를 자산으로 인식하고 라이선스와 사용 절차를 체계화하기 위해 소프트웨어자산관리 절차를 다루고,

그 결과를 명시하고 있으며 IT서비스관리를 효과적으로 지원하고 조직이 일정한 표준에 따라 소프트웨어를 관리 하고 있다는 것을 증명할 수 있는 기준을 제시하고 있다.[3]

여기서 소프트웨어관리(SAM : Software Asset Management) 원칙은 소프트웨어와 관련된 매체, 설치, 라이선스, 라이선스 증명, 지식재산권에까지 모두 적용되며 결국 소프트웨어자산에 대한 TCO를 절감하고 소프트웨어라이선스 준수에 들어가는 투자비용 대비 효과를 극대화하는 것으로 요약할 수 있다. 또한 이 표준은 (그림 1)과 같이 IT 자산 관리의 일부분에 해당하며, IT 품질을 다루는 ISO/IEC 20000과 IT Service 관리를 다루는 BS 15000 및 ITIL의 일부분에서 발전된 연구이다.

ISO/IEC 19770 Part 1의 주요내용은 기업 등에서 소프트웨어 자산관리를 실천하기에 앞서 기본적으로 필요한 ‘프로세스’에 대한 정의와 설명으로 구성되어 있으며 프로세스는



(그림 1) ISO/IEC 19770-1의 범위

조직관리 프로세스	핵심SAM 프로세스	주요프로세스 인터페이스
<ul style="list-style-type: none"> SAM 통제환경 <ul style="list-style-type: none"> -SAM 기업지배 프로세스 -SAM 역할과 책임 -SAM 정책, 프로세스, 절차 -SAM 역량 SAM 계획 및 구현 프로세스 <ul style="list-style-type: none"> -SAM 계획 -SAM 구현 -SAM 모니터링과 검토 -SAM의 지속적인 개선 	<ul style="list-style-type: none"> SAM 인벤토리 프로세스 <ul style="list-style-type: none"> -SW자산의 확인 -SW자산 인벤토리 -SW자산통제 SAM 검증 및 기준준수 프로세스 <ul style="list-style-type: none"> -SW 자산 기록검증 -SW라이선스 기준준수 -SW자산 보안 기준준수 -SAM 적합성 검증 SAM 운영관리 프로세스 및 인터페이스 <ul style="list-style-type: none"> -SAM 관계 및 계약관리 -SAM 재무관리 -SAM 서비스 레벨관리 -SAM 보안관리 	<ul style="list-style-type: none"> SAM 라이프사이클 프로세스 인터페이스 <ul style="list-style-type: none"> -변경관리 프로세스 -취득 프로세스 -SW개발 프로세스 -SW배포 관리 프로세스 -SW폐기 프로세스 -인스턴트 관리 프로세스 -문제 관리 프로세스 -폐기 프로세스

(그림 2) SAM 프로세스의 주요내용

(그림 2)와 같이 SAM을 위한 조직 측면의 관리 프로세스, 핵심 SAM 프로세스, SAM을 위한 기본 프로세스 인터페이스 등의 3 부분으로 구성되고, 전체적으로는 27개의 세부 프로세스로 이루어져 있다. 한편 ISO 19770-part2 태그 부분도 산업계의 최종 검토를 거쳐 08년도에 발표될 예정이다.

Part 2는 애플리케이션 ID, 번호, 버전을 등 모든 소프트웨어 권리관리 정보를 담을 수 있는 국제표준 체계이다. 산업계의 의견을 검토하는 내용을 담은 ISO 19770-Part2의

전문가 회의 자료는 태그제정의 목적을 <표 4>과 같이 정리하고 있다.

정리하면 19770-Part1은 소프트웨어를 자산으로 인식하고 전사적인 소프트웨어관리 절차(Process)를 규정하였다면, ISO/IEC 19770-Part2는 소프트웨어자산 각각에 대한 정확한 정보규명과 이를 통한 정확한 인벤토리 관리를 위한 수단을 제공하는데 핵심이 있다.

따라서 ISO/IEC 19770-Part 2가 공식적인 소프트웨어분야의 권리관리 표준으로서 정착이 되면 기업 내에서 사용되고 있는 소프트웨어자산을 식별하고 관리하는데 보다 손쉬운 Tool 제공이 가능할 것이며 이에 따른 소프트웨어자산관리 솔루션 개발업체들 사이의 경쟁이 본격화 될 것으로 전망된다.

3. 소프트웨어 관리 및 검색 기법

3.1 사용 허락에 의한 소프트웨어분류

소프트웨어는 다양하게 분류할 수 있으나 본 논문에서는 <표 6>와 같이 기업에 설치된 소프트웨어 중 관리 대상 차원에서 저작권자에 의해 허용된 사용범위에 따라 상용 소프트웨어

<표 4> 19770-Part2 제정 목적

태그 제정 목적	내용
인벤토리 관리	각각의 소프트웨어에 특정ID를 부여함으로써 사내소프트웨어배치, 재무회계, 라이선스 일치를 용이하게 함
보완 및 안정성 확충	소프트웨어별 ID에 의해서 인증되지 않은 소프트웨어를 폐기하고 소프트웨어사용권 등 소프트웨어데이터 저장 인프라 구축과 데이터에 대한 접근통제 기술로 소프트웨어정보 및 사용권 관리
정보처리기능을 가진 서비스 제공	각각의 소프트웨어가 특정 ID를 가지고 있기 때문에 소프트웨어의 특성에 따라 패치, 업그레이드 등 맞춤 서비스 제공 가능
소프트웨어관리 행정의 효율화	각각의 소프트웨어자산에 대한 보안ID를 기반으로 몇 개의 소프트웨어를 어느 정도의 기간 동안 사용할 것인지 등의 계획 수립이 가능하여 소프트웨어에 대한 근본적인 관리가 가능

<표 5> ISO 19770-Part2에 논의되고 있는 주요 소프트웨어관리 정보[4]

정보군	정보내용
애플리케이션 이션	애플리케이션ID, 제품 군, 제품 조, 제품 에디션, 애플리케이션 번호
버전	버전이름, 버전번호
제조사	서비스 팩, 빌드 번호, 제품SKU, 제조날짜, 회사명, 언어, 시리얼넘버, 제품출처
소프트웨어 사용	링크지원, 주석, 키워드, 공개 키 토큰, 프로세서 아키텍처
저작권	저작권자, 법적인 저작권, 라이선스 권리ID
소프트웨어설치	최초설치ID, 설치시간, 설치소스, 조작변경출처, 조작변경유형, 조작변경일

<표 6> 소프트웨어 분류

구 분	개 념
상용 소프트웨어(정품)	돈을 주고 구입함으로써 사용자로서 정당한 권리를 획득한 후 사용해야 하는 대부분의 프로그램들을 의미.
프리웨어(Free Ware)	사용자의 일정한 제약 조건하에(예를 들어 개인목적이나 가정 내에서만 사용, 배포금지 등) 무료로 사용할 수 있는 프로그램을 말하며, 예를 들어, Adobe Acrobat Reader, MS Internet Explorer 등이 이에 해당.
셰어웨어(Share Ware)	정품을 구입하기 전에 미리 사용해 볼 수 있는 프로그램으로 일부 기능이나 사용기간, 사용대상이 제한되어 있으므로 정당한 사용조건을 파악한 후 사용해야 한다.
번들(Bundle)	별도로 판매되는 제품들을 묶어 하나의 패키지로 만들어진 형태로, 번들소프트웨어 단독 판매는 불법이며 사용 권한 등에서도 제한이 있으므로 확인 후 사용해야 한다.

트웨어, 프리웨어, 웨어웨어, 번들용으로 분류한다. 여기서 기업 내에서 관리해야 소프트웨어로는 프리웨어를 제외한 모든 소프트웨어가 관리대상임을 알 수 있다.

3.2 기업 소프트웨어 관리 절차 및 문제점

다음 (그림 3)은 기업 내에서 기본적인 소프트웨어 관리를 위해 설계된 절차 모델이다.

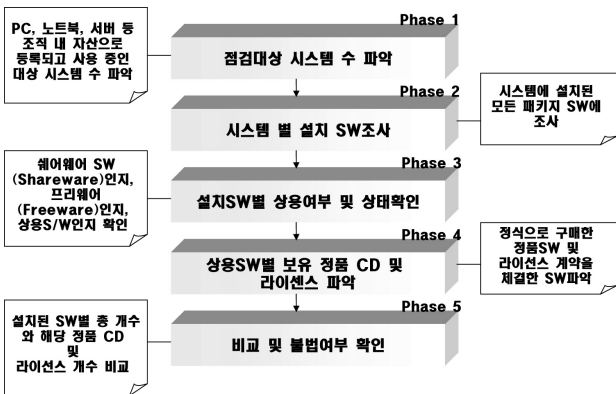
- Phase 1 : 점검대상 시스템 수 파악
- Phase 2 : 시스템별 설치 소프트웨어조사
- Phase 3 : 설치 소프트웨어별 상용여부 및 상태 확인
- Phase 4 : 상용소프트웨어별 보유 정품CD 및 라이선스 파악
- Phase 5 : 비교 및 불벌여부 확인

일반적으로 소프트웨어관리는 조직의 내부에서 소유하고 있는 모든 컴퓨터에 설치되어 있는 소프트웨어를 조사함으로써 시작된다. 이어서 소프트웨어에 대한 라이선스를 보유하고 있는지를 확인해야 한다.

조직 내 패키지 소프트웨어관리 절차 (그림 3)의 Phase 2 단계에서 조직의 점검대상 시스템에 설치된 소프트웨어를 신속하고 정확하게 검색하기 위해서는 PC에 설치된 각각에 소프트웨어들 중 상용소프트웨어에 대한 저작권 정보가 사전에 정의 되어있어야 하지만 Windows 기반 OS에 제공되는 소프트웨어설치 정보에는 해당 소프트웨어가 상용소프트웨어인지 프리웨어인지 소프트웨어관리자가 한눈에 알 수 있는 정보가 없어 효율적인 소프트웨어관리에 많은 문제점을 안고 있다. 즉 점검대상 시스템 수가 증가할수록 수작업에 의한 점검시간이 증가하고 이에 따른 종합적인 관리가 어렵게 된다는 것이다.

3.3 관리대상 소프트웨어 정보 및 검색 알고리즘

상용소프트웨어에 관한 소프트웨어관리정보를(“이하 소프트웨어 Check list이라 함”)사전에 정의한 후 이를 해당 시스템에 설치된 소프트웨어와 매칭 시키면서 검색하면 신속한 점검이 이루어지며 효율적인 소프트웨어관리가 가능해



(그림 3) 소프트웨어관리를 위한 기본 절차 모델

진다. 소프트웨어관리 정보 등록은 국내에 유통되고 있는 소프트웨어 중 관리 대상 소프트웨어를 선정하여 (그림 4)와 같이 이를 개발한 회사의 저작자 및 개발회사, 소프트웨어 명칭, 소프트웨어 파일명(Filename), 파일크기 등을 DB화한다.

(그림 5)는 등록된 소프트웨어 관리정보를 검색한 예를 보여 주고 있다.

소프트웨어 Check List에는 소프트웨어관리자 입장에서 관리해야할 대상소프트웨어에 관한 저작권정보 즉 저작권사, 프로그램명, 버전 등 다양한 메타 정보값이 포함된다.

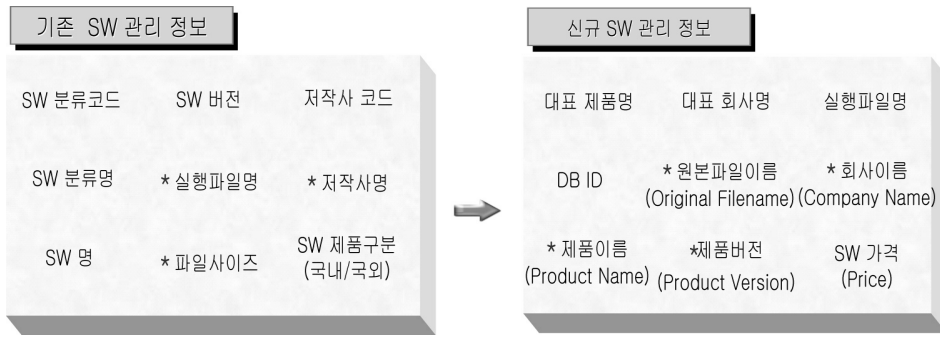
그러나 (그림 4)의 경우 소프트웨어관리정보 입력시 일부 상용 소프트웨어의(Alzip 등) 경우 프로그램 패치가 수시로 일어나, 이를 매번 조사하여 메타 정보값을 Update하기에는 한계가 있고, 바이러스 등에 의한 파일 사이즈의 변동으로 똑같은 소프트웨어임에도 불구하고 여러 개의 소프트웨어 관리정보를 입력하고 관리해야하는 문제점이 발생하였다. 즉 (그림 8)의 기존검색의 경우 PC에 설치된 모든 파일을 검색하여 기존 SW관리정보에 등록된 상용소프트웨어 추출 기준 중에 하나인 실행파일명과 파일사이즈를 비교한 결과와(result1) 레지스트리를 검색하여 PC에 설치된 소프트웨어가 Check List에 등록된 파일명과 사이즈가 다르지만 Registry에는 존재할 경우의 결과(result2)를 활용하여 설치된 모든 관리 대상 소프트웨어를 검색할 경우에는 소프트웨어 Check List중 파일이름 및 파일사이즈가 정확히 매칭이 되지 않으면 정확한 검색이 도출되지 않았다. 따라서 이러한 문제점을 개선하기 위해 (그림 7)의 예시와 같이 저작권사 등이 소프트웨어를 배포할 당시 제공한 소프트웨어의 속



(그림 4) 소프트웨어 Check List 구조

순번	구분	저작권사/프로그램명	파일 이름	파일크기	설치개수
0	2	[Microsoft Corporation]Windows 98 Welcome41...	WELCOME.EXE	262524	1
1	1	[Microsoft]Windows98	imekr9u.ime	27192	1
2	2	[Microsoft Corporation]Windows 98 Welcome41...	WELCOME.EXE	278528	1
3	1	[Microsoft]windows98(SE,AME)	imekr9u.ime	81920	1
4	1	[Microsoft]MS-Excel2000	excel.exe	7151681	1
5	1	[Microsoft]MS-PowerPoint2000	poweront.exe	4255428	1
6	1	[Microsoft]MS-Word2000	word.exe	8441907	1
7	2	[Symantec Corporation]Norton Utilities Integrator...	NORTON.EXE	1900232	1
8	2	[Ahnlab, Inc.]V9Pro5.0.0.0	v9pro2.exe	79586	1
9	1	[EGTecon]ALZIP9.0	alzip.exe	393744	1
10	2	[FINAL DATA INC.]FinalData MFC Application1...	FinalData.exe	2445312	1
11	1	[한글과컴퓨터]한글97	hwpw.exe	1604096	1
12	1	[문서작성용류체]01047197	web.exe	571904	1
13	2	[Sybase Inc.]Sybase Inc. Product File	plbys30.dll	1236922	1
14	1	[DM-computer]Ultra-Edit7.0	uedit32.exe	1138888	1
15	2	[Oracle]LOGIN1.00	LOGIN.EXE	16384	1
16	1	[Microsoft]Windows-2000-Professional	welcome.exe	366954	2
17	2	[Microsoft Corporation]NT Tape Backup Utility5...	ntbackup.exe	1155344	2
18	2	[Microsoft Corporation]Microsoft 한글 입력 시스...	imekr9u.ime	92432	1
19	2	[Microsoft Corporation]NT Kernel & System5.00...	ntoskrnl.exe	1641552	1
20	2	[FinePrint Software, LLC]FinePrint 2004.80	fpdip4.exe	384544	1

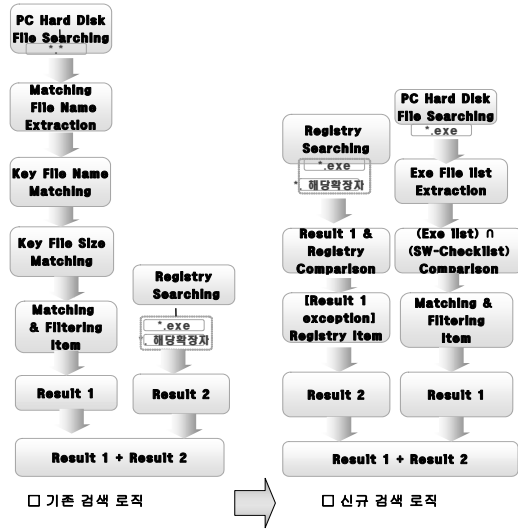
(그림 5) 소프트웨어 Check List 검색



(그림 6) 개선된 소프트웨어 Check List 구조



(그림 7) ALZip.exe 속성 정보 값 예시



(그림 8) 개선된 검색 로직

<표 7> 검색 방법 비교

기존 검색 방법		신규 검색 방법	
Condition 1	PC에 설치된 SW가 Check List에 등록된 파일명과 사이즈가 동일한 경우	Condition 1	PC에 설치된 SW가 SW Check List에 등록된 SW 속성정보와 동일한 경우 (기존 Condition 1, 2 모두 포함)
Condition 2	PC에 설치된 SW가 Check List에 등록된 파일명은 동일하나, 사이즈가 다른 경우	Condition 2	PC에 설치된 SW가 SW Check List에 등록된 SW 속성정보와 다르지만, Registry에는 존재하는 경우로 SW의 속성정보를 추출하는 경우 (Condition1 이외의 검색조건)
Condition 3	PC에 설치된 SW가 Check List에 등록된 파일명과 사이즈가 다르지만, Registry에는 존재하는 경우로 파일명을 추출하는 경우		

성 정보 값 일부를 활용하여 (그림 6)과 같이 소프트웨어 관리정보 구조를 개선하였다.

(그림 6)에서 보논바와 같이 Check List에 등록된 소프트웨어 관리 정보에는 저작권자들이 해당 소프트웨어를 릴리즈 할 당시의 속성 값이 포함되어 있어 기존 파일명 및 파일사이즈 비교 검색을 통한 문제점을 상당부분 해소할 수 있다. 즉 이러한 속성 값을 이용하여 소프트웨어관리정보 DB를 구축 한 후 (그림 7)과 같이 검색 대상 PC내에 설치된 .exe파일의 속성 값을 비교한 결과(Result 1)와 PC에 설치된 소프트웨어가 Check List에 등록된 소프트웨어 속성정

보와 다르지만, Registry에는 존재하는 경우(Result 2)로 구분하여 검색하면 기존 검색 로직에 따른 파일 사이즈 변동 등의 문제점을 해결할 수 있다. <표 7>은 기존 검색 및 신규 검색 방법을 비교한 결과이다.

4. 패키지 소프트웨어 점검서비스 구현

4.1 검색 메소드 구현

(그림 8)의 개선된 검색로직을 구현한 결과는 (그림 9)과 같다.

```

CMainFrame::OnInspectorStart()
    ← 검색시작 버튼 클릭(메시지 처리)
CMainFrame::OnInspectStartKey()
    ← 검색시작
    pDlg.DoModal() ← 사용자 정보 다이얼로그 띄움
    CInspectorDoc::InspectStartThread()
        ← 검색 Thread 띄움
    ----- 검색 스레드 -----
    CSearchThread::Run() ← 스레드 시작
    CSearchThread::KeyFileDataReadSet() ← Check list 로드
    CSearchThread::FindFileRecursive(char *path)
        ← 모든 파일 검사, 재귀함수

CSearchThread::KeyDataCheck(CString fname)

CSearchThread::ArrayFind(CString str) ← Check list와 비교
(

CSearchThread::GetRegSoftWareInfo(char *fname)
    ← 키파일에 없을경우 레지스트리 검색 )

CSearchThread::WriteLocalPcCheckList() ← 파일에 쓰기

CInspectorDoc::SelectTreeViewFolder(INSPECT_SOFTWARE_INFO *fd)
    ← 검색된 소프트웨어정보 필터링(중복 소프트웨어 flag=4로 처리)
(

SearchThread::FindFileRecursive(char *path) )
    ← 하위폴더가 있을경우 자기자신 호출 )
    ::SendMessage(pFrame->m_hWnd, WM_COMMAND, ID_INSPECT_STOP, 0)

    ← 메인프레임에 검색 종료 메시지 전달
    ----- 검색 스레드 -----
CMainFrame::OnInspectStop()
    ← 검색 종료 메시지 처리
CMainFrame::OnInspectStopKey()
    ← 검색 스레드
    사용자로부터 "검색시작" 이벤트를 받으면, 사용자에게 사용자 PC의 정
    보를 요구하게 된다. 사용자로부터 PC의 정보를 입력받으면 본격적으로 해
    당 PC에 설치되어 있는 software(해당 소프트웨어의 실행 file or dll
    file)를 scanning하게 된다.
    
```

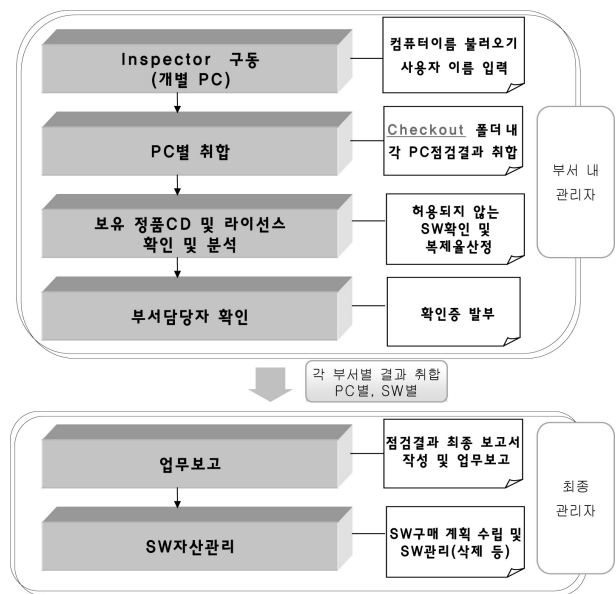
(그림 9) PC내 소프트웨어 검색을 위한 메소드

4.2 소프트웨어 점검용 Tool(Inspector)

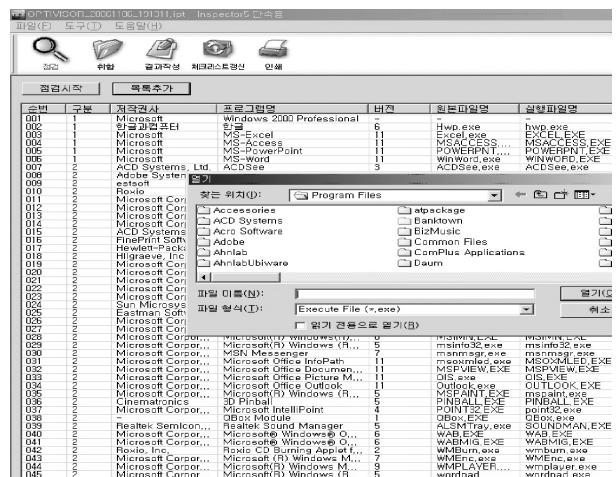
(그림 6)과 (그림 8)의 신규 검색 로직을 적용하여 개발된 Inspector는 개별 PC에 대한 소프트웨어설치 정보를 수집 하면서 소프트웨어 Check List와 비교 한 후 레지스트리 정보를 추가적으로 비교하여 점검대상 PC에 설치되어 있는 소프트웨어에 대하여 점검대상 기관이 보유하고 있는 소프트웨어관리 대장과의 일치여부를 대조·확인한 후 소프트웨어관리자가 신속하게 판단하여 효과적으로 소프트웨어를 관리 할 수 있도록 설계되었다. (그림 10)은 소프트웨어관리자가 Inspector를 이용하여 소프트웨어를 관리하는 절차를 나타내고 있다.

(그림 11)은 조사대상 PC에 설치된 소프트웨어를 검색한 결과로서 그림에서 구분 란의 1은 PC에 설치된 소프트웨어가 소프트웨어 Check List에 등록된 소프트웨어 속성정보와 동일한 경우이며 구분 2는 PC에 설치된 소프트웨어가 소프트웨어 Check List에 등록된 소프트웨어 속성정보와 다르지만, Registry에는 존재하는 경우를 나타낸다.

(그림 12)은 각 PC별로 검색을 완료한 후 각각의 검색결과를 취합하는 화면이며 취합된 결과는 (그림 13)와 같이 소프트웨어사용 현황을 출력할 수 있다.



(그림 10) Inspector를 이용한 소프트웨어관리 절차



(그림 11) Inspector 검색 결과 출력 화면



(그림 12) 각 PC별 검색결과 취합

순번	제작사명	프로그램명(버전)	설치수량	비고
1	Microsoft	Windows 2000 Professional (-)	1	3
2	한글과컴퓨터	한글 (6)	1	3
3	Microsoft	MS-Excel (11)	1	3
4	Microsoft	MS-Access (11)	1	3
5	Microsoft	MS-PowerPoint (11)	1	3
6	Microsoft	MS-Word (11)	1	3
7	AhnLab, Inc.	Smart Update Utility (5)	1	3
총 1대 PC 7종 SW 7개				

(그림 13) 소프트웨어사용현황 출력

4.3 사용효과 및 배포건수

소프트웨어관리자가 (그림 3)의 Phase 2단계에서 다수의 PC에 설치된 소프트웨어에 대하여 수작업에 의해 사용현황을 파악할 경우에는 많은 시간이 소요된다. 그러나 점검용 Tool을 이용하면 실험결과 점검대상 PC가 많아질수록 점검 시간 및 이에 소요되는 인력을 상당부분 절감할 수 있는 것으로 나타났다.

즉 수작업으로 주요 PC사양이 Pentium 4 CPU에 500MB RAM을 보유한 PC 100대를 대상으로 1인이 점검하여 결과를 도출하기 까지 27시간 이상이 소요되는 것으로 나타났으나 Inspector를 이용할 경우에는 <표 8>과 같이 43분이 소요되어 상대적으로 1일에 3인 이상의 투입인원을 절감한 효과와 이에 따른 경제적 효과로서 2007년 소프트웨어기술자의 등급별 노임단가를 적용하여 초급기술자 기준으로 438,854원의 절감 효과가 있는 것으로 나타났다.

현재 Inspector는 일반 기업체의 소프트웨어 관리자들이 직접 이용할 수 있도록 인터넷을 통한 무료 다운로드 서비스[8]를 실시하고 있으며, 07년도 점검용SW의 총 배포건수

<표 10> 사용 만족도 조사결과

응답 항목	응답자수	비율
매우 만족	529명	47%
대체적으로 만족	342명	30%
보통	132명	12%
불만족	131명	12%
합 계	1,134 명	

(기간 : '07. 5. 1 ~ '07. 8. 31)

<표 9>과 같이 총 3만 3천여 건으로서 정부차원에서 실시한 공공기관 패키지SW 사용실태 자체 점검 시행에 따라 6~8월간 배포건수가 전체건수의 43%를 차지하고 있다.

또한 <표 10>와 같이 Inspector사용한 경험이 있는 사용자를 대상으로 07.5~07.8월 까지 인터넷 Cyber-Poll을 이용한 사용 만족도 조사를 실시한 결과 80% 가까이 사내 소프트웨어 관리에 도움이 된 것으로 조사되어 점검용 소프트웨어가 기업에서 소프트웨어를 효과적으로 관리함으로써 구입비용의 절감 및 경영합리화에 많은 이바지를 함과 동시에 정품 소프트웨어 사용문화 정착에 중요한 용도로 활용되고 있는 것으로 나타났다.

5. 결론 및 향후 연구방향

정보통신기술의 발달은 인터넷 사용 인구의 폭발적인 증가와 더불어 괄목할만한 성장을 이루었으나, 다른 한편으로는 인터넷과 정보통신 기술의 발전에 따른 소프트웨어 복제의 용이성 증대로 소프트웨어를 관리하는 담당자 입장에서는 사내에 있는 모든 PC에 설치된 소프트웨어를 일일이 확인하고 불법여부를 파악 하는데 많은 어려움을 겪고 있으며 관리

<표 8> PC 100대당 점검 소요시간 비교

구분	소요시간 및 비용	
	수작업	Inspector 이용시
PC 1대당 평균 점검시간	15분	20초
100대 점검시간	15분(1PC) × 100대 = 25시간	20초(1PC) × 100대 = 33분
점검결과 취합시간	60분	30초
보유CD및 라이선스 비교 확인 시간	10분	10분
점검결과 레포팅 시간	15분	즉시
PC 100대 기준 총 소요 시간	27시간 10분	43분
1인 투입(8시간 기준/1일)	3.3인/1일	0.08인/1일
인건비('07년 초급기술자 노임 단가 기준)	3.3인 × 136,290원 = 449,757원/1일	0.08인 × 136,290원 = 10,903원/1일

<표 9> '07년 월별 Inspector 배포 건수

(단위 : 건)

구분	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
홈페이지	467	471	1,153	1,301	1,949	6,108	4,606	7,048	1,711	2,088	1,139	1,154
공유 계시판	51	436	414	104	328	1,150	518	526	344	211	178	117
합계	518	907	1,567	1,405	2,277	7,258	5,124	7,574	2,055	2,299	1,317	1,272

해야할 PC가 증가 할 수 록 소프트웨어관리자의 어려움도 증대하고 있다. 이를 해결하기 위해 소프트웨어자산관리를 위한 모니터링 Tool로써 Inspector를 개발하여 점검서비스를 구현하였고, 정확한 소프트웨어 현황 정보를 추출하기 위한 검색알고리즘 적용에 따른 사용효과 등을 제시하였다.

소프트웨어자산관리의 기본 활동은 소프트웨어권리관리 정보의 표준화를 통해 소프트웨어 Check List에 등록되어야 할 상용 소프트웨어 및 관리대상 정보를 효과적으로 획득하고 이를 기반으로 각 PC에 설치된 소프트웨어 정보를 분석하여 조직 내에 정식으로 보유하고 있는 소프트웨어 정보와의 일치 여부를 식별하는 것으로 정의할 수 있다. 그러나 전술한 바와 같이 현재 Windows기반에서 제공하고 있는 소프트웨어권리관리 정보로는 상용 소프트웨어 여부를 사용자가 효과적으로 파악할 수 없어 효과적인 소프트웨어자산관리의 한계가 있었으나 Inspector 서비스 구현 사례를 통해 이러한 문제점을 상당부분 해소할 수 있음을 알 수 있다.

또한 관련연구 사례와 같이 전사적인 관점에서 성공적인 소프트웨어 자산관리를 위해서는 소프트웨어 자산을 식별하고 자산관리를 위한 정책(policy)이 선행 되는 것이 중요하다. 이를 통해 IT서비스 중단 위험방지 및 소프트웨어라이선스의 효율적 재배치를 통한 비용절감과 시장의 수요에 대응하여 빠르고 신뢰성 있게 새로운 시스템과 기능을 배치하는 등의 효과와 더불어 소프트웨어저작권 침해를 예방할 수 있는 효과를 가져 올 수 있다.

또한 향후 개방, 공유, 참여의 정신을 모티브로 한 Web 2.0환경에서는 패키지 위주의 소프트웨어 유통방식이 이제는 웹 기반의 SaaS(Software as a Service)서비스 방식으로 전환되어 소프트웨어의 소유에서 접속으로 사용방식의 변화가 가속화 될 것이다.

즉, Web 2.0을 위시한 새로운 IT패러다임이 사용자와 더불어 소프트웨어산업구조와 사업모델 전반에 커다란 영향을 주고 IT패러다임의 변화와 함께 새로운 소프트웨어 비즈니스 모델을 만들어 경제적 가치를 창출해내는 플랫폼으로의 웹이 뿌리내리고 있어 기존의 패키지 위주의 소프트웨어 유통방식은 이러한 패러다임의 변화에 대처하기에는 한계가 있어 이제는 웹 기반의 서비스 방식으로 전환됨으로써 SOA와 Web Service기술을 기반으로 한 온 디맨드(On Demand) 소프트웨어 유통 방식인 SaaS(Software as a Service) 시장이 점차 정착될 것으로 전망되며 궁극적으로 이러한 SaaS를 기반의 소프트웨어 사용방식이 보편화 되면 소프트웨어 자산관리 및 소프트웨어 저작권 침해의 문제점 등이 상당부분 해소될 수 있을 것이다. 따라서 이의 활성화를 위해 정부차원의 다각적인 지원 및 관련 연구가 필요한 시점이라 하겠다.

참 고 문 헌

[1] Gartner Research “Software License Compliance Remains a Problem for Many Companies,” Gartner March, 2005.
 [2] 한국소프트웨어저작권협회 “SAM국제 동향 연구”, July, 2007.
 [3] ISO/IEC, “19770-1 Information technology Software asset

management Part 1: Process,” 2006.
 [4] ISO/IEC, “19770-2 Software Asset Management Tag: A discussion Document for industry consultation,” December, 2005.
 [5] 김현숙, “소프트웨어자산관리의 필요성과 국제동향”, 한국 소프트웨어저작권협회, p.10-25, 2006.
 [6] BSA, “Fourth Annual BSA and IDC Global software 2008 Piracy Study,” BSA, 2008.
 [7] 컴퓨터프로그램보호위원회 홈페이지, <http://www.pdmc.or.kr/affairs/appraisal/intro.jsp>.
 [8] 인스펙터 배포용 홈페이지, <http://www.itsam.or.kr>.
 [9] (주)티씨오솔루션 홈페이지, http://www.tcosol.com/product/product_stream01.html.
 [10] (주)닥터소프트 홈페이지, <http://www.doctorsoft.co.kr/>.
 [11] (주)아이엑스 홈페이지, <http://www.izex.co.kr/>.
 [12] 김혜창 “2007 소프트웨어 IPReport”, 컴퓨터프로그램보호위원회, p.51-57, 2007.
 [13] 컴퓨터프로그램보호위원회 “2008 소프트웨어저작권 등 기술 및 정책 동향,” 2008.
 [14] Gartner Research “DeskTop PC TCO Update,” Gartner, 2003.
 [15] 차태원 외 3인, “네트워크기반 소프트웨어점검용 시스템 효과적 구축에 관한연구”, 컴퓨터프로그램보호위원회, 2005.
 [16] 차태원 외 2인, “공공기관 소프트웨어관리 가이드라인,” 컴퓨터프로그램보호위원회, 2005.
 [17] <http://www.microsoft.com/resources/sam/default.xml>.
 [18] <http://www.asset-management-software.org/>.
 [19] Alexa Bona Jane B. Disbrow “Gartner Survey Shows IncreasesinSoftware License Audits” Gartner December, 2006.
 [20] Frank DeSalvo Jane B. Disbrow “Gartner Predicts Increase in Vendor Software Audits,” Gartner, January 2008.
 [21] John Bace Carol Rozwell Joseph Feiman Bill Kirwin “Understanding the Costs of Compliance” Gartner, July 2006.



차 태 원

e-mail : twcha@socop.or.kr

1996년 관동대학교 전자계산공학과(학사)

1999년 한국외국어대학교 정보관리학과

(석사)

2004년 정보처리기술사

1997년~현 재 컴퓨터프로그램보호위원회

기술정책팀 팀장

2006년~현 재 서울산업대학교 IT정책전문대학원

산업정보시스템전공 박사과정

2008년 (사)한국정보통신기술사협회 이사

관심분야: 소프트웨어 공학, DRM, 소프트웨어 육성정책



안 재 경

e-mail : jkhan@snut.ac.kr

1985년 서울대학교 산업공학(학사)

1987년 서울대학교 산업공학(석사)

1991년 The Univ. Iowa, U.S.A 산업공학
(박사)

1991년~현 재 서울산업대학교 산업정보시스템공학과 교수

1997년 일본 와세다 대학교 교환교수

2000년 방송위원회 디지털방송추진위원회 위원

2001년 미국 Georgia Tech. 교환교수

2003년~2005년 대한산업공학회 재무이사

2005년~현 재 대한산업공학회 감사

2006년~현 재 통신위성·우주산업 연구회 편집이사

관심분야: 방송·통신경영, IT경제성 분석