AHP기법을 이용한 안티바이러스 소프트웨어 평가요인분석

김 종 기* · 황 숙 연**
* 부산대학교 경영학부 조교수
** 부산대학교 경영학과 석사과정

Ⅰ. 서론

현재의 컴퓨터 바이러스는 초기의 단순하고 미약한 형태를 벗어나 상당히 다양한 특성을 가지고 있으며 그 피해의 정도도 기업 측면이나 개인적 측면 둘 다에서 재무적 혹은 비재무적인 차원의 심각한 영향을 미친다(Dunham, 2003; Mamaghani, 2002).

이러한 이유로, 정보자산에 심각한 악영향을 미치는 바이러스나 악성코드의 피해를 줄이기 위한 보안 대책이 필요하다. 즉, 기업에서는 이러한 바이러스의 피해를 막기 위해 하나의 방법으로 안티바이러스 소프트웨어(Anti-Virus Software)를 구입하여 설치하는 보안대책을 활용하고 있다.

하지만, 안티바이러스 소프트웨어는 지극히 개인의 경험에 의해 선택되어지며, 최선의 안티바이러스 소프트웨어를 선택하는 것은 많은 이유들로 인해 어려운 의사결정의 문제이다(Dunham, 2003).

안티바이러스 소프트웨어를 선택하는 과정은 안티바이러스 소프트웨어의 기능과 성능 혹은 가격 등의 차원에서 상당히 다양한 특성을 지니고 있다는 점과 사용하는 기업이나 조직 혹은 개인이 요구하는 사항이 차별적인 성격을 가지고 있다는 점에서 상당히 복잡한 차원의 문제라고 언급하고 있다(Dunham, 2003; Mamaghani, 2002).

일반적인 소프트웨어의 평가모형에 관한 연구는 국내외에 다수가 있으며 국외에서는 오래전부터 소프트웨어의 품질을 향상시키기 위한 품질평가에 관한 연구가 이루어져 왔다. 하지만, 이런 표준안이 있음에도, 일반적으로 소프트웨어 제품의 측정과 평가에는 다음과 같은 여러 가지 어려움이 있다. 첫째, 소프트웨어 제품평가기준이 되는 속성들의 다양성과 무형성을 들 수 있다. 둘째, 다수의 평가속성들 간에는 목표달성에 대한 상충성이 존재한다. 다수의 상충되는 목표들이 주어졌을 때, 유일한 최적해는 존재할 수 없으며, 다른 목표를 희생시키고 단일 목표를 개선시키는 절충이 필요하다. 이러한 평가속성의 특징으로 인하여 소프트웨어 제품의 평가와 선정 문제는 구조상 다속성 의사결정(MADM : Multiple Attribute Decision Making)의 성격을 가지게 된다(박호인과 정호원, 1997).

따라서 기업이나 개인 측면에서의 안티바이러스 소프트웨어의 선택과 사용도 대표적인 다속성 의사결정으로 볼 수 있는 것이다. 다속성 의사결정은 하나의 의사결정에 다수의 요인들이 주요한 고려대상이 되는 경우 각 대상에 대한 체계적인 정립을 통해서 효과적이고 합리적인 의사결정을 할 수 있게 하는 주요한 방법론으로 사용되고 있다. 하지만 안티바이러스 소프트웨어는 그 성격상 기존의 소프트웨어와는 다른 특성들로 인해 일반적인 소프트웨어 평가모형을 그대로 적용시키기에는 안티바이러스 소프트웨어만의 특징을 가지고 있다. 따라서 본 연구에서는 안티바이러스 소프트웨어의 평가요인분석을 위한 주요한 기준을 다속성 의사결정에 유용하다고 평가되고 있는 계층적 분석기법(AHP)을 사용하여 측정하고자 하며, 안티바이러스 소프트웨어의 특성에 맞는 평가요인을 분석하고자 한다. 그리고 이러한 설문결과의 분석을 통해서 개별사용자 혹은 조직이나 기업의 합리적인 안티바이러스 소프트웨어 선택의 요인들을 제시함으로써 안티바이러스 소프트웨어의 평가요인에 영향을 미치는 주요한 요인들을 규명하는 것을 본 연구의 목적으로 한다.

Ⅱ. 이론적 고찰

1. 컴퓨터 바이러스에 대한 연구

1.1. 컴퓨터 바이러스의 정의

초기의 바이러스라는 명칭은 다른 악의적인 코드와 명확한 구분이 되는 경우가 상당히 많았으나, 최근의 안티바이러스 소프트웨어나 그 개발자의 애플리케이션 혹은 사용자의 인지 측면에서 볼 때, 다른 악의적인 코드를 통합해서 지칭하는 개념으로 바이러스라는 용어를 사용하고 있는 것이

다. 이것을 통틀어 악성코드라는 용어로 사용하고 있다.

이것은 웜(Worm)이나 트로이 목마(Trojan Horse), 논리 폭탄(Logic bomb), 트랩도어(Trap door) 같은 개념을 포괄적으로 바이러스로 지칭하기도 한다는 것이다. 따라서 일반적으로 사용되는 바이러스의 실질적인 정의는 다른 사람에게 심리적, 실질적인 피해를 입히는 컴퓨터 프로그램 또는 실행 가능한 부분으로 제작자(사) 실수로 포함된 버그는 제외되나 광범위한 피해가 예상되는 경우는 포함하게 된다.

1.2. 컴퓨터 바이러스의 종류

본 연구에서의 컴퓨터 바이러스는 웜이나 트로이 목마, 논리폭탄, 혹스, 조크, 인터넷 웜 같은 것들을 지칭한다(Sherif & Gilliam, 2003).

트로이 목마(Trojan Horses)는 실행 가능한 소프트웨어 프로그램에 행위자가 의도하는 결과를 수행하는 특정한 명령어를 불법적으로 삽입한 은폐된 프로그램을 지칭한다(한국정보보호센터 편, 2000; Sherif & Gilliam, 2003).

웜(Worm)은 컴퓨터 내의 다른 시스템에는 직접적인 영향을 미치지 않고 기억장소 내에서 자기 자신을 계속적으로 증식하는 프로그램이다(한국정보보호센터 편, 2000). 그 중에서 현재 바이러스 사고의 대부분을 차지하고 있는 것이 인터넷 웜(Internet Worm)이다.

인터넷 웜(Internet Worm)이란 원격지에서 불특정 시스템의 취약성을 이용하여 자신을 복제한 후 다른 시스템으로 전파하는 프로그램이다. 일반적으로 메모리 내에 자가복제를 하는 프로그램을 의미했으나, 최근에는 컴퓨터 상에서 네트워크를 통하여 자가복제를 하는 프로그램을 말하며, 자가복제, 빠른 전파, 서비스 거부현상 등과 같은 특징을 가지고 있다.

1988년 널리 퍼져 피해를 준, 모리스웜(Morris Worm)을 시작으로 많은 웜들이 만들어져 큰 피해를 주고 있다.

논리폭탄(Logic Bombs)은 정상적인 프로그램에 은폐된 루틴을 삽입하여 불법 행위의 대상이 되는 컴퓨터에서 루틴이 정하는 논리적인 조건이 이루어지면 루틴에 삽입된 명령을 실행하도록 하는 것이다(한국정보보호센터 편, 2000; Sherif & Gilliam, 2003).

혹스는 주로 이메일을 통하여 다른 사람에게 거짓 정보 즉 루머를 유포하는 것으로 사용자에게 심리적인 위협이나 불안감을 조장한다.

조크는 심리적인 위협이나 불안감을 조장하는 프로그램으로써, 물질적인 피해는 없으나 백신에서 진단/삭제한다. 지금까지 살펴본 여러 가지 악성코드들 중 대표적인 악성코드의 일종인 컴퓨터 바이러스, 트로이목마, 웜에 대한 비교를 통해 코드간의 차이를 다음의 [표 2-1] 같이 볼 수 있다.

[표 2-1] 악성코드의 비교

	자기복제	감염대상	형태	복구방법
컴퓨터 바이러스	O	O	기생/결침	치료
트로이목마	×	×	독립	삭제
웜	O	×	독립	삭제

자료원 : 정관진과 이희조(2003), pp.3 인용

1.3. 컴퓨터 바이러스의 영향

1.3.1. 컴퓨터 바이러스 발생원인 추세

컴퓨터 바이러스의 감염 경로 중 가장 빈번한 경우는 소프트웨어의 불법 복사에 의한 감염이다. 또한 네트워크를 통해서 다운로드 받은 소프트웨어에 의하여 바이러스가 감염되기도 한다. 최근에는 인터넷을 이용하는 인구가 빠른 속도로 증가하고 있기 때문에 네트워크를 관리하는 관리자 뿐만 아니라 컴퓨터를 사용하는 모든 사용자들도 컴퓨터 바이러스에 대한 주의를 기울여야 한다.

[표 2-2] 바이러스의 발생원인

바이러스 발생원인	1996	1997	1998	1999	2000	2001	2002
E-mail	9%	26%	32%	56%	87%	83%	86%
인터넷 다운로드	10%	16%	9%	11%	1%	13%	11%
웹 브라우징	0%	5%	2%	3%	0%	7%	4%
알 수 없음	15%	7%	5%	9%	2%	1%	1%
다른 진로	0%	5%	1%	1%	1%	2%	3%
소프트웨어를 통해	0%	3%	3%	0%	1%	2%	0%
디스켓 : 다른 원인	71%	84%	64%	27%	7%	1%	0%

자료원 : ICSA Labs(2003), pp.25 인용

1.3.2. 컴퓨터 바이러스 피해 규모

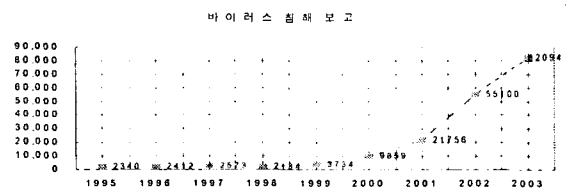
최근의 컴퓨터 바이러스는 웜과 백도어가 결합된 형태나 네트워크 전파와 같이 복합적인 형태를 가진 바이러스가 많이 나타나고 있으며, 일반적으로 웜이나 트로이목마 제작에 고급언어를 사용하여 바이러스 보다 제작이 쉬운 것도 확산이유중의 하나이다.

즉, 현재의 컴퓨터 바이러스는 컴퓨터를 사용하는 사람의 숫자가 증가 할수록, 네트워크의 수도 증가하여 바이러스의 확산 속도도 같이 증가하고 있는 추세이다(Hubbard and Forcht, 1998).

따라서 현재의 바이러스는 형태는 바이러스로부터 네트워크를 통하여 스스로 전파되고 피해를 입히는 웜 형태로 변화해 나가고 있다.

우리나라의 인터넷 이용자 10명중 6명은 최근 1년 사이에 자신의 컴퓨터가 바이러스에 감염됐던 경험을 갖고 있는 것으로 드러났다(전자신문, 2003a).

그림[2-1]에서 보는 것과 같이 컴퓨터 바이러스의 침해 사건 수는 계속해서 증가하고 있다.



[그림 2-1] 컴퓨터 바이러스 침해 사건 수
자료원 : CERT and CERT Coordination Center Carnegie Mellon Universitv (2003)

2. 안티바이러스 소프트웨어에 대한 연구

2.1 안티바이러스 소프트웨어의 정의

안티바이러스 소프트웨어는 일반적으로 파일에서 악성코드(바이러스, 트로이목마, 웜)를 진단하고 복구해 주는 프로그램으로, 현재까지 악성 코드예방과 퇴치에 가장 효과적인 것으로 평가되고 있다(차민석, 2002).

2.2 안티바이러스 소프트웨어의 기능

안티바이러스 소프트웨어는 일반적으로 다음과 같이 크게 세 가지로 구분된다.

① 바이러스 감염을 사전에 방지하기 위한 예방 프로그램
② 바이러스에 감염이 되었는지를 검사하는 진단 프로그램
③ 바이러스에 감염된 것이 확인되었을 경우 파일을 원래의 상태로 복구하는 치료 프로그램

예방용 프로그램은 바이러스 공격으로부터 바이러스를 사전에 막을 수 있는 프로그램이다(한국정보보호센터 편, 2000). 그러나 예방용 프로그램 자체가 바이러스에 감염이 되면 악성 바이러스보다 더욱 큰 피해를 입을 수도 있으므로 주의를 하여야 한다.

진단용 프로그램은 이미 알려져 있는 바이러스에 대한 진단과 새로운 바이러스에 대한 진단을 위하여 각기 다른 방법을 사용한다. 알려진 바이러스에 대한 진단 방법에는 문자열 비교와 내부 알고리즘 추적 방법이 있으며, 새로운 바이러스에 대한 진단 방법에는 특정 문자열의 존재 여부를 확인하는 방법과 인공지능 방법 등이 있다.

치료용 프로그램은 바이러스에 감염된 것이 확인하면 해당 파일을 치료하여 복구를 하는 프로그램이다.

일반적으로 사용자는 안티바이러스 소프트웨어를 설치하는 것으로 바이러스의 침해 사고를 예방할 수 있다고 생각한다. 하지만, 실제 안티바이러스 소프트웨어는 새로운 바이러스에 대한 정보를 가지고 있지 않기 때문에 기능을 수행하는데 여러 가지 한계점을 가지고 있다.

첫째, 안티바이러스 소프트웨어의 가장 큰 한계는 해당 프로그램이 처리할 수 있는 바이러스만 진단, 퇴치가 가능하다는 점이다. 사용자가 소프트웨어를 설치한 후에도 신종 바이러스는 계속 제작되고 유포되기 때문에 안티바이러스 소프트웨어의 엔진을 계속 업데이트 해주어야 한다. 안티바이러스 소프트웨어 제작사들은 신종 바이러스의 진단/치료 기능이 추가될 때 마다 이에 대한 정보를 업데이트해 고객들에게 제공한다. 최신 엔진을 사용해도 진단되지 않는 신종 악성코드에 감염 위험은 있지만 최선의 예방책은 지속적으로 엔진을 업데이트해주고 시스템 감시 기능을 사용하는 것이다.

둘째, 시스템 감시기능은 바이러스의 접근을 차단하는 것이지, 네트워크를 통한 바이러스의 공격을 예방하는 것은 아니다. 시스템 감시기능은 사용자가 접근하는 파일의 감염 여부를 검사해 사전에 사용자가 감염된 파일에 접근하는 것을 차단해 준다. 이런 특징으로 인해 바이러스에 감염된 파일을 다른 컴퓨터의 공유 폴더를 이용하여 파일을 복사할 때, 파일을 받는 컴퓨터 쪽 백신의 시스템 감시 기능은 파일이 전송되는 중간에 파일을 검사하지 않고 파일이 모두 복사되고 파일이 닫히는 시점에서 파일을 검사/진단하여 바이러스 유무를 사용자에게 알려준다. 이렇게 악성코드 감염여부를 검사하는 시점이 파일에 대한 모든 작업이 완료된 후이다 보니, 사용자는 현재 사용하는 프로그램의 문제로 악성코드의 감염을 차단하지 못했

다고 오해할 수 있는 것이다. 이러한 이유로 현재 네트워크를 통해 전달되는 내용의 안전성을 검사하지 못하는데, 이것은 인터넷으로 파일을 모두 다운로드 받은 후에 바이러스 감염여부를 확인하기 때문이다. 이러한 문제를 해결하기 위해서 안티바이러스 소프트웨어와 함께 침입탐지 프로그램 등이 함께 사용되어야 한다.

그 외에도 안티바이러스 소프트웨어 제품마다 바이러스 분류가 다르기 때문에 나타날 수 있는 한계나, 안티바이러스 소프트웨어도 하나의 프로그램이기 때문에 바이러스에 감염 될수 있다는 한계를 가지고 있다.

2.3 국내 안티바이러스 소프트웨어

2003년도 세계 안티바이러스 소프트웨어 시장은 시만텍(Symantec)과 맥피아(Mcfee), 트렌드마이크(Trendmicro) 등 3 개 업체가 시장의 70% 이상을 점유한 가운데 국내 업체 안철수연구소가 전년 대비 22.7%의 견실한 성장률을 나타냈다(박상훈, 2004).

최근 IDC가 발표한 2003년 전 세계 안티바이러스 시장 점유율과 2004~2008 시장 전망 보고서에 따르면 시만텍은 전년대비 35.9%가 성장한 10억 8800만 달러 매출로 시장 점유율 40.4%를 기록하며 안티바이러스 업계 1위 자리를 지켰으며, 국내에서는 유일하게 안철수연구소가 전년대비 22.7% 상승한 2080만 달러 매출로 톱 10에 이름을 올렸다(IDC, 2004).

IDC(2004)의 보고서에 의하면 지난해 안티바이러스 업계와 보안 위협과 관련해 몇 가지 특징적인 흐름이 있었다고 지적했다. 가장 큰 특징은 기존에 알려진 보안 취약점을 겨냥한 웜과 전통적인 대량 메일 바이러스가 서로 혼합된 것이다. 대용량 메일 바이러스의 공격은 전년과 큰 차이가 없었으나, 네트워크 웜은 기업 네트워크를 위협하는 가장 큰 위협으로 급속히 대두되고 있다고 분석했다.

두 번째는 바이러스의 활동 방식과 이를 배포하는 동기가 변했다는 점이다. 과거의 바이러스와 웜이 철없고 어리석은 비전문 개발자들에 데이터를 파괴하는 방식이었다면, 최근에는 공격방식도 복잡해지고 동기도 금적적인 이익을 위해 신용카드 번호나 계좌 정보, 타인의 개인 정보 등을 노리는 경우가 많아졌다.

2.4 기존의 안티바이러스 소프트웨어 평가요인에 관련된 선행연구

Dunham(2003)는 그의 연구에서 안티바이러스 소프트웨어 선정에 주요한 5가지의 고려사항들을 설명하고 있다.

① 안티바이러스 프로그램의 인식
② 주요 소프트웨어에 대한 인식과 개인적인 테스트
③ 안티바이러스 소프트웨어의 기능과 이해에 대한 평가
④ 안티바이러스 소프트웨어의 기능성에 대한 검정
⑤ 개인적인 필요를 위한 최선의 안티바이러스 소프트웨어의 선정

또한 비용, 시스템 요구사항, 인터페이스와 리더쉽, 성능, 스캔(scan) 옵션, 제거와 복구 옵션, 지원 등이 안티바이러스 소프트웨어의 주요한 평가 기준으로 제시되고 있다.

Sherif & Gilliam(2003)의 연구에서는 주된 안티바이러스 소프트웨어 패키지에 대한 평가를 위한 평가 항목으로, 멀티 플랫폼 지원, 업데이트, 지속적인 개발, 헬프데스크 지원, 스캐닝, 치료, 시스템 감염에 대한 응답, 스캔율, 메모리 요구 수준 등의 평가 기준을 제시하고 있으며, 주된 평가기준으로 기능, 신뢰, 유지, 비용의 4가지의 기준을 제시하고 있다.

Mamaghani(2002)는 안티바이러스와 필터링 소프트웨어의 평가와 선택에 관한 연구를 통해 선택의 주요한 기준으로 설치(Installation), 운영(Operation), 관리(Administration), 통보/로그(Notification/Logging), 안티바이러스(Anti-virus), 내용 필터링(Content Filtering)의 6가지의 기준을 제시하고 있다. 이러한 기준들 중 연구 결과를 통해 안티바이러스와 내용필터링이 안티바이러스 선택에 가장 주요한 기준이 된다고 보여 지고 있다.

황진욱(2002)는 안티바이러스 소프트웨어를 소비자 관점에서 선호도 분석을 하였다. 연구를 위해 인터넷을 사용하는 기업 및 사용자를 대상으로 설문조사가 실시되었고, 제품인지도가 도출되었으며, 인지도를 구성하기 위해 요인 분석과 선호도 회귀 분석이 사용되었다. 요인 분석 결과, 핵심품질, 효율성, 주변요소의 3가지 공통요인이 도출되었다. 그 중에 핵심품질은 세가지요인 중에서 가장 중요한 것으로 나타났다.

[표 2-3] 안티바이러스 소프트웨어 평가요인관련 선행 연구

연구자/연구기관	평가기준
Dunham(2003)	비용, 시스템 요구사항, 인터페이스와 리더쉽, 성능, 스캔옵션, 제거와 복구옵션, 지원
Sheirf & Gilliam(2003)	기능, 신뢰, 유지, 비용
PC magazine(2002)	설치, 인터페이스, 지원, 바이러스 스캐닝, 스케줄 스캐닝, 업데이트
Managhani(2002)	설치, 운영, 관리, 통보/로그, 안티바이러스, 내용 필터링
황진욱(2002)	온라인지원, 신종바이러스에 대한 신속한 대응,편리한 메뉴구성, 디자인, 프로그램업데이트, CPU 및 메모리 점유율, 스캔시간, 프로그램습득, 스캔 엔진의 업데이트, 가격대비 성능

3. 기존 소프트웨어 평가 모형 연구

3.1 Boehm의 연구

Boehm은 최초로 소프트웨어 제품의 품질을 정량적으로 측정 및 평가 할 수 있는 품질모형을 제시하였다(Boehm, 1978). 이 모형에 의하면 품질을 외부 관점에서 본 중간구조와 내부관점의 기초구조의 2차원으로 구성되어 있다(Gilles, 1992).

이 모형에서의 품질특성은 초기운용, 유지보수, 이식성의 세 가지로 분류하고 중가구조에서는 이식성, 신뢰성, 효율성, 인간공학, 시험성, 이해성, 변경성으로 품질특성과 연관지어 분류하였다. 내부특성으로는 장치 독립성, 자기 포함성, 완전성, 확고 및 무결성, 일관성, 설명성, 장치 효율성, 통신성, 자기 기술성, 구조성, 간결성, 명료성, 확대성 등 14가지로 분류하고 중간구조와의 연관성을 나타내었다.

3.2 McCall의 연구

McCall(1978)은 미 공군의 요청으로 연구를 시작하였고, 미 국방성의 품질표준으로 채택되는 등 현재의 많은 품질측정 기초로 이용되고 있다.

McCall의 모형은 소프트웨어 품질 요구사항을 명확히 설정하고 개발된 제품의 품질을 평가관리하기 위한 방안으로, 사용자 관점에서 요구된 품질수준을 구체화하고 소프트웨어 개발과정 중 요구된 품질의 평가 가능성을 판단하는 지침으로 활용되기도 한다.

또한 열한 가지의 품질요인을 소프트웨어 제품의 개정, 전이 및 운용의 관점에서 구분하여 제안하였다. 제품제정에는 유지보수성, 유연성, 시험성이 있고, 제품전이에는 상호 운용성, 재사용성, 이식성이 있으며, 제품운영을 위해서는 정확성, 효율성, 무결성, 신뢰성, 사용성의 품질특성이 있다.

3.3 Evans의 연구

Evans(1987)는 12개의 품질특성을 선택한 후, 소프트웨어 개발 수명주기를 중심으로 성능, 설계, 개선의 3개 범주로 구분하였다. 성능은 사용자에 의해 언급된 요건들이 소프트웨어의 운영방법을 구체적으로 설명하는가의 측면을 다루고, 설계에서는 사용자의 소프트웨어 요구사항을 구현하기 위한 종합적인 설계과정, 개선은 사용자와 유지보수자에게 영향을 주는 수명주기 상에서 고려되는 사항으로, 소프트웨어가 얼마나 용이하게 사용자의 요건을 추가 ·변경하고 다른 프로그램에서 재사용이 가능할 것인가를 다루고 있다.

3.4 소프트웨어 평가 관련 국제표준

ISO/IEC 9126(Information Technology-Software Quality Characteristics and metrics)은 품질특성 및 메트릭을 정의하고 있는 표준으로 각 품질 특성별로 세부 메트릭을 제시하고 있다(ISO/IEC 9126, Information Technology-Software Quality Characteristics and metrics, McGRAW-HILL Book Company Europe, 1995).

ISO/IEC 9126에서는 소프트웨어 품질 특성 및 품질 부특성의 항목들을 계층구조로 표현하는 품질 모델을 제시하고 있다. 이는 소프트웨어 품질과 관련된 체크리스트로써 사용 가능하고 품질 부특성들은 내부 메트릭이나 외부 메트릭에 의해 측정 가능하다. 각 품질 특성 및 부특성들을 정의하면 [표 2-4]과 같다.

III. 연구모델의 구성

1. 계층적 분석 절차 (Analytic Hierarchy Process : AHP)

1.1 AHP기법의 정의와 특징

Saaty(1972)에 의해 창안된 AHP(Analytic

[표 2-4] ISO/IEC 9126 품질 특성에 대한 정의

품질 특성	부특성	정의
기능성 Functionality 소프트웨어가 특정조건에서 사용될 때, 명시된 요구와 내재된 요구를 만족하는 기능을 제공하는 소프트웨어 제품의 능력	적합성 suitability	명시된 과업과 사용자 목적에 적합한 기능의 집합들을 제공해 줄 수 있는 능력
	정확성 accuracy	정확한 규격, 일치된 결과, 정확한 효과를 나타내는 소프트웨어의 능력, 계산에 있어서의 정확성
	상호운용성 interoperability	명시된 하나 또는 그 이상의 시스템들과 상호운영 될 수 있는 능력
	부합성 compliance	관련 표준, 법률상의 규칙 또는 협정·규정 등에 부합되는 능력
	보안성 security	정보나 프로그램에 대한 권한 없는 접근을 방지하는 능력
신뢰성 Reliability 명시된 조건에서 사용될 때, 성능 수준을 유지할 수 있는 소프트웨어 제품의 능력	성숙성 maturity	소프트웨어의 결함으로 인한 오류를 회피하는 능력
	오류허용성 fault tolerance	명시된 인터페이스의 위반 또는 소프트웨어의 오류 발생시에 소프트웨어가 명시된 수준으로 성능을 유지하는 능력
	복구성 recoverability	고장시에 소프트웨어가 직접 피해를 입은 데이터를 복구하고 성능 수준을 회복하는 능력
사용성 Usability 명시된 조건에서 사용될 경우, 사용자에 의해 이해되고, 학습되고, 사용되고 선호될 수 있는 소프트웨어 제품의 능력	이해성 understandability	특정 과업과 사용환경에서 어떻게 사용되는지와 적합한지를 사용자가 이해할 수 있도록 하는 능력
	학습성 learnability	사용자에게 소프트웨어 적용법을 학습하도록 하는 소프트웨어의 능력
	운용성 operability	사용자가 운영하고 통제하도록 하는 소프트웨어의 능력
효율성 Efficiency 명시된 조건에서 사용되는 자원의 양에 따라 요구된 성능을 제공하는 소프트웨어 제품의 능력	시간효율성 time behavior	명시된 조건하에서, 기능 수행시 적절한 응답과 처리시간 또는 처리율을 제공해 주는 능력
	자원활용도 resource utilization	명시된 조건하에서, 기능 수행시 적절한시간동안 적절한 자원을 사용하는 능력
유지보수성 Maintainability 소프트웨어 제품이 변경되는 노력. (환경과 요구사항 및 기능적 명세에 따른 소프트웨어의 수정, 개선, 혹은 개작 등)	해석성 analysability	소프트웨어의 고장원인 또는 결함을 진단하거나, 수정되어야 할 부분을 찾아내는 능력
	변경성 changeability	소프트웨어가 명시된 수정을 이행하는 능력, 코딩, 설계·문서화의 변경 포함
	안정성 stability	소프트웨어의 수정에 의해 야기되는 예기치 못한 영향을 최소화하는 능력
	시험성 testability	수정된 소프트웨어의 타당성을 시험하는 능력
이식성 Portability 한 환경에서 다른 환경으로 전이될 수 있는 소프트웨어 제품의 능력	적응성 adaptability	이식 목적에 해당되는 작업이나 수단 이외의 것은 이용하지 않고, 명시된 다른 환경으로 이식될 수 있는 능력
	설치성 installability	소프트웨어가 명시된 환경에서 설치되는 능력
	부합성 conformance	소프트웨어가 이식성에 관련된 표준이나 협정에 부합하는 능력
	치환성 replaceability	소프트웨어가 명시된 다른 소프트웨어와 치환되어 그 환경에서 사용되는 능력

자료원 : 한국전산원, "국가정보화 촉진을 위한 품질정착연구", 연구보서, 1997, pp.32-34

Hierarchy Process : 이하 AHP) 기법은 다속성 의사결정도구로, 기업이나 군사에 관련된 계획, 의사결정, 제한된 자원의 배분 등과 관련된 문제를 해결하기 위하여 개발되었다.

AHP기법은 객관적인 평가요인은 물론 주관적인 평가요인도 수용하는 매우 유연한 의사결정기법으로서 수학적인 이론보다도 직관을 바탕으로 하기 때문에 그 논리가 매우 쉽다는 장점을 가지고 있다(Scholl *et al.*, 2004; 윤재곤, 1997).

AHP는 특히 집단의사결정 문제 등에 유용하여 1980년대 이후 경영과학 분야의 주요 의사결정기법으로 인정받아 왔다. 일반적으로 의사결정 문제는 서로 불완전한 정보와 제한된 자원 하에서 목적과 기준에 일치되는 최적의 대안을 선택해야 하는 문제를 가지고 있다. 이러한 관점에서 AHP는 최종적인 목적아래 하위기준들을 수립하고, 상위

목표의 관점에서 하위 기준을 평가하여 가중치를 부여하는 방식이다.

1.2 AHP기법의 적용절차

AHP기법은 다기준 의사결정문제(Multi-Criteria Decision Making Problem)를 최종목적(Overall Goal), 평가기준(Criteria), 하위평가기준(Subcriteria), 해결 가능한 대안(Alternatives)의 순서로 하는 계층구조로 모형화 한다(Saaty, 1990; Saaty., 1982).

AHP기법은 목표들 사이의 중요도(Weight)를 단계적으로 나누어 파악함으로써 각 대안들의 우선순위를 산정하는 기법이다(Satty and Vargas, 1982).

AHP기법의 적용 방법은 다음과 같다.

1단계 : 문제를 정의하고 목적이나 목표를 결정한다.

2단계 : 계층구조를 만든다. (Identify Decomposition의 원리를 적용한다.)

3단계 : 각각의 비교 행렬을 만든다.

4단계 : 3 단계에서 만들어진 행렬들에 주관적으로 n(n-1)/2 회의 비교를 하여 상대적 중요도를 평가한다.

이때 사용되는 상대적 중요도의 크기를 나타내는 방법은 다음과 같다.

[표 3-1] 이원 비교시 중요도의 측정

척도	정의	설명
1	동등하게 중요	두 개의 요소가 똑같이 중요함
3	약간 더 중요	한 요소가 다른 요소보다 약간 더 중요함
5	더욱 더 중요	한 요소가 다른 요소보다 더욱 더 중요함
7	대단히 더 중요	한 요소가 다른 요소보다 대단히 더 중요함
9	절대적으로 중요	한 요소가 다른 요소에 비해 절대적으로 중요함
2,4, 6,8	위에서 정의된 척도들의 중간	위에서 정의된 척도들 사이의 값이 요구될때 사용

자료원 : Satty(1978), pp.152 부분인용

5단계 : 상대적 중요도를 합성하고 일관성 지수(Consistency Index, C.I.), 일관성 비율(Consistency Ration, C.R.)를 구한다. 계산과정이 복잡하므로 대개의 경우 컴퓨터 프로그램이 이 과정을 대신해 준다.

6단계 : 3,4,5 단계를 계층 구조의 최고 수준의 Priority vector를 구할 때까지 반복한다.

7단계 : 최종 수준의 행렬에서 C.R이 20%를 넘지 않으면 이 분석을 인정하고 그렇지 않으면 3단계에서부터 모든 과정을 다시 반복한다.

1.3 AHP기법의 계산방법

1.3.1 이원비교(pair-wise comparison)

다 속성 의사결정일 때는 각 속성의 상대적인 중요도를 모두 고려하여 중요도를 정하기가 어렵다. 다라서 AHP에서는 속성들을 두개씩 뽑아 쌍대비교를 한다. 어떤 계층에 있는 한 기분(속성 또는 요소)의 관점에서 직계하위계층에 있는 기준들의 상대적 중요도(선호도 또는 우월정도)를 평가하기 위해 기준들 간에 쌍대비교를 행하고 그 결과를 행렬로 나타내는 과정이다. 따라서 문제의 각 단계에서의 구성요소들을 비교할 때 각각 다음과 같은 정방행렬이 입력 자료로 쓰인다.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} \end{bmatrix}$$

위와 같은 입력행렬은 $A_{ji} = \frac{1}{A_{ij}}$ 이고, 정방행렬의 주대각선의 원소들이 역수행렬 (reciprocal matrix)이다. 역수속성을 지닌 정방행렬은 고유벡터(eigenvector)와 고유값(eigenvalue) 이라는 성질을 이용하면 쉽게 해를 구할 수 있다.

쌍대비교의 과정에는 평가기준 등에 대한 의사결정자의 선호도(preference)를 먼저 어의적인 표현에 나타내고 이를 계량화 과정에 포함시킨다.

1.3.2 가중치의 추정

상대적 중요도를 평가하는 쌍대비교를 행한 후에는 각 계층에 대하여 비교대상 평가요소들이 갖는 상대적인 가중치를 추정한다.

1.3.3 논리적 일관성

이원비교에 의해서 얻어진 행렬 A의 원소인 A_{ij}가 W_i/W_j의 값을 갖고 있다면 기수적 일치성 즉 $a_{ij} \times a_{jk} = a_{ik}$ 이 성립되어야 한다. 그러나 이원비교에 의해서 행렬A를 얻는 방법은 각 열의 요소의 중요도를 1로 기준한 후 대각선 상위에 있는 행의 요소들의 상대적 중요도를 결정하고 있다. 따라서 행렬 A의 원소들의 논리적 모순성 정도를 검증하는데 기수적 일치성을 알아볼 필요가 있다.

행렬 A가 기수적으로 정확히 일치하는 경우는 $\lambda max = n$ 이 된다. 일치하지 않는 경우는 λmax 는 언제나 n보다 큰 값을 갖는다.

1.3.4 가중치의 종합

AHP기법의 마지막 단계는 하위계층에 있는 평가요소들의 가중치를 구하기 위해서 각 계층에서 계산된 평가기준들의 가중치를 종합하는 과정이다, 즉, 상위계층에 있는 의사결정문제의 궁극적인 목표에 미치는지 또는 어느 정도의 중요성을 갖고 있는지를 알아보기 위해 평가요소들의 종합 가중치를 구한다. 이것은 대안의 상대적 비중 또는 우선순위라고도 하며 대안의 선택기준이 된다.

2. 연구모형의 설계를 위한 AHP문제 구성

2장과 3장을 통해 안티바이러스 소프트웨어 평가와 관련된 기존 연구들을 고찰하고 일반적인 소프트웨어 평가에 관한 여러 가지 기준들을 살펴보았다.

소프트웨어 품질평가와 우선순위를 결정하는데 있어서 여러 방법이 사용될 수 있으나 정호원과 이종무(1997) 및 이종무(1997)의 연구에서 AHP기법을 이용한 연구의 객관성이 입증되고, ISO/IEC 9126에서 제시하고 있는 계층 구조가 AHP기법을 이용하는데 알맞으며, 측정의 단위와 관련없는 다속성의 비교가 가능한 AHP기법을 본 연구에서 이용하기로 한다.

2.1 안티바이러스 소프트웨어의 주요 평가요인에 관한 내용

안티바이러스 소프트웨어에 대한 평가기준을 작성하기 위해서 3가지의 기준을 토대로 하였다.

첫 째로, 소프트웨어의 품질관련 국제 표준 중 ISO/IEC 9126에서 제시하고 있는 소프트웨어 품질기준을 토대로 본 연구 모델의 기본적인 기준을 제시한다. 두 번째로, 앞의 문헌연구에서 살펴보았던 안티바이러스 소프트웨어 평가와 관련된 여러 가지 선행 연구를 통해 공통적인 평가 요소를 도출해 낸다. 마지막으로 소프트웨어 외적인 선정기준을 토대로 AHP에서 평가기준으로 사용할 주특성과 하부특성을 도출하도록 한다.

2.2 안티바이러스 소프트웨어의 평가 요인선정방법 및 절차

2.2.1 소프트웨어 자체 품질 측면에서의 선정기준

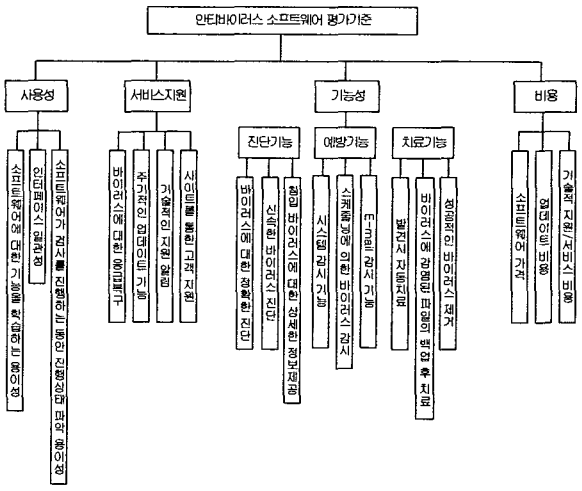
본 연구에서는 ISO 9126에서 정의된 품질특성을 기준으로 안티바이러스 소프트웨어의 특성을 반영하는 방법으로 평가기준을 도출하였다.

ISO 9126는 현재 국제적으로 사용 효과성을 인정받고 있는 품질평가 모형이다. 따라서 본 연구에서는 ISO 9126 품질특성 중 사용자는 모든 특성을 고루 갖춘 소프트웨어를 찾아볼수가 없다. 그 이유는 6가지의 품질특성이 서로 상호의존적이거나 상호배타적인 성질을 가지고 있기 때문이다. 따라서 본 연구에서는 사용자가 품질특성의 평가를 할 수 있는 기능성, 사용성의 품질특성 두 가지를 사용하도록 한다.

2.2.2 안티바이러스 소프트웨어 평가 측면에서의 선정기준

본 연구에서는 2장에서 살펴본 안티바이러스 소프트웨어 평가의 선행 연구를 통하여 안티바이러스 소프트웨어 선정의 공통적이고 일반적인 요인들을 도출해 냈다. 그 중에서 Dunham(2003)의 연구에서 안티바이러스 소프트웨어를 평가하기 위한 설문의 세부항목을 참조하여 본 연구에서 측정항목으로 사용하도록 한다.

2.3 AHP기법 평가를 위한 안티바이러스 소프트웨어 계층구조



[그림 3-1] 안티바이러스 소프트웨어 계층구조

2.4 평가요인의 조작적 정의

[표 3-2] 안티바이러스 소프트웨어의 평가기준별 내용

	평가기준	내용
사용성	기능학습의 용이성	안티바이러스 소프트웨어의 기능을 학습하는 용이성
	인터페이스 일관성	안티바이러스 소프트웨어의 프로그램 화면의 일관성
	진행상태 파악 용이성	안티바이러스 소프트웨어의 설치, 검사 기능과정에서 어떤 진행상태가 되고 있는지에 대한 것
운용지원	바이러스에 대한 응급복구	바이러스 감염 시 치료해 줄 수 있는 시간
	주기적인 업데이트	새로운 바이러스에 대한 계속적인 최신의 치료나 스캔 정보의 제공
	기술적인 지원	바이러스를 자동으로 치료하지 못했을때 공급업체에서 치료를 위해 지원할 수 있는 기술
	고객 트레이닝	이메일이나 알림을 통한 고객에게 바이러스 위험에 대한 정보제공 정도
기능성	바이러스 발견시 자동치료	바이러스가 발견되면 자동으로 치료하는 기능
	바이러스에 감염된 파일의 백업 후 치료	감염된 파일을 치료하기 전에 파일을 저장해 주는 기능
	성공적인 바이러스 제거	바이러스 감염 시 바이러스가 제거된 것
	시스템 감시기능	컴퓨터가 켜짐과 동시에 계속적으로 바이러스에 대한 감시를 실시
	스케줄링 스캐닝	사용자가 지정한 시간에 맞추어 바이러스의 유무에 관한 검사를 실시
	E-mail 스캐닝	컴퓨터로 E-mail을 확인하기 전 바이러스의 유무에 관한 체크
	바이러스에 대한 정확한 진단	바이러스에 감염 되었을시 감염된 바이러스의 이름과 치료 방법을 정확하게 진단하는 기능
	신속한 바이러스 진단	바이러스에 감염 되었을시 감염된 바이러스를 시간의 지체없이 빠른시간 내에 진단하는 기능
	침입 바이러스에 대한 상세한 정보제공	감염된 바이러스에 대한 상세한 정보가 제공되는 기능
비용	구입비용	안티바이러스 소프트웨어의 구입비용
	업데이트 비용	안티바이러스 소프트웨어를 업데이트 할 때드는 비용
	기술적인 지원/서비스 비용	고객이 기술적인지원과 서비스를 받을 시 지출되는 비용

Ⅳ. 실증 분석

본 연구는 다음에서 제시하는 과정으로 진행된다.

1. 연구도구 개발

본 연구를 수행하기 위해서는 설문지를 이용한 우편조사와 현장조사를 수행하여야 한다. 설문조사를 위해서 설문지를 구성하는 설문문항이 연구하고자 하는 바에 정확하게 적용되어야 한다. 따라서 설문지를 통한 연구도구의 개발을 위해 기본적으로 앞에서 보였던 문헌연구를 통하여 평가항목구성요인들을 도출하였고, 각각의 평가 항목을 측정하기 위해 평가항목의 세부 문항을 3~4개 정도로 구성하였다. 그리고 9점 척도를 통한 측정을 위해 설문지를 구성하였다.

2. 예비 설문조사(pilot study)

본 연구에서는 조사하고자 하는 내용의 관한 선행연구가 거의 없는 실정이다. 따라서 연구도구인 설문항목이 응답자들에게 본 연구의 의도가 제대로 전달되는지 확인하기 위하여 사전조사를 수행하였다. 사전조사의 설문대상은 안티바이러스 소프트웨어를 사용하고 있는 대학생과 대학원생, 일반관리자를 대상으로 설문지를 통한 설문을 실시하였다. AHP기법이 많은 설문지를 필요로 하지 않는 기법이기 때문에 성실하지 않은 답변을 한 설문을 제외한 10부의 설문을 사전조사에 사용하였다. 이 사전조사를 통하여 설문문항의 측정수치를 수정하였고, 명확하지 않는 설문문항에 대해서는 일부를 수정하거나 삭제 또는 용어의 설명을 통해 변경하였다.

3. 설문실시

3.1 자료 수집 및 표본의 특성

설문의 대상은 안티바이러스 소프트웨어를 사용하고 있는 대학원이상의 사용자와 대학교의 전산실을 관리하는 관리자를 주 대상으로 하였다. 설문지의 조사 방식은 우편조사와 e-mail, 방문조사를 병행하였다.

4. 실증분석결과

4.1 조사결과의 분석

4.1.1 연구 대상자들의 특성 분석

4.1.2 측정도구의 신뢰성 검증

4.2 계층구조별 평가기준의 중요도 산출

5. 평가요인 분석 결과에 대한 논의

V. 결론

1. 연구결과 요약 및 시사점

2. 연구의 한계점 및 향후 연구과제

참고문헌

이종무, 소프트웨어 품질 평가 투입 요소 결정에 관한 연구, 고려대학교 대학원, 박사학위논문, 1997.

박상훈, 보안 최악의 해-안티바이러스 업계 경영 성적표, ZDNet Korea, 2004.

박호인, 정호원, "소프트웨어 제품을 위한 평가 선정 모형의 조사 및 적용성에 관한 연구," 한국정보처리학회지, 제4권 제7호, 1997, p.1706-1718.

윤재곤, "AHP기법의 적용효과 및 한계점에 관한 연구 : MIS 성공요인평가를 위한 3가지 통계기법 비교중심," 한국경영과학, 제21권 제3호, 1996, pp. 109-125.

정관진, 이희조, "인터넷 웜과 바이러스의 진화와 전망," 안철수 연구소, 2003.

차민석, *바이러스 백신과 악성코드에 대한 몇가지 오해*, http://info.ahnlab.com/securityinfo/, 2002.

한국정보보호센터, 정보보호개론, 한국정보보호센터, 2000.

황진욱, An Analysis of Consumers' Preference in Virus Vaccine Programs, 한국정보통신대학원 석사학위논문, 2002.

전자신문, 바이러스 감염경로와 피해, 2003(a). 04. 09.

Dunham, K., "Evaluating Anti-Virus Software:Which Is Best?," Telecommunications and Network security, Jul/Aug, 2003, pp. 17-28.

Evans, M. W. and J. J. Marciniak, Software Quality Assurance and Management, John Wiley & Sons, 1987.

Gilles, A. C., Software Quality : Theory and Management, Chapman and Hall, 1992.

Hubbard, J. and K. Forcht, "Computer viruses : how companies can protect their systems," Industrial Management & Data Systems, Vol. 98, No. 1, 1998, pp. 12-16.

ICSA Labs, Computer virus Prevalence Survey, 2003.

ISO/IEC 9126 Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for Their Use, ISO, 1991.

Mamaghani, F., "Evaluation and selection of an antivirus and content filtering software," Information Management & Computer Security, Vol. 10, No. 1, 2002, pp. 28-32.

McCall, J. A., "A Framework for the Measurement of Software Quality : The Proceeding of the ACM Software Quality Assurance," 1978.

Saaty, T. L., "Modeling unstructured decision problems - The theory of analytical hierarchies," Mathematics and Computers in Simulation, 1978, pp.147-158.

Saaty, T. L., "How to Make a Decision : the Analytic Hierarchy process," European Journal of Operational Research, Vol. 48, No. 1, 1990, pp. 9-26.

Saaty, T. L. and L. G. Vargas, The Logic of Priorities, Kluwer-Nijhoff Publishing, London, 1982.

Sherif, J.S. and D. Gilliam, "Deployment of anti-virus software : a case study," Information Management & Computer Security, Vol. 11, No. 1, 2003, pp.5-10.

Vaiday O. S. and S. Kumar, "Analytic hierarchy process : An overview of applications," European Journal of Operational research, 2004, pp.1-29.