신뢰와 평판 기반의 소프트웨어 보증 시스템 구현†

박대명○ 이석민 유대훈 최웅철
광운대학교 컴퓨터 과학과
{eoanddl, nenunena, yo2dh, wchoi}@kw.ac.kr

Implementation of Software Assurance System Based on Trust and Reputation

DaeMyeong Park○, SeokMin Lee, Daehun Yoo, WoongChul Choi
Department of Computer Science, KwangWoon University

요 약

소프트웨어 보증은 소프트웨어의 신뢰성, 예측 가능한 실행으로 정의할 수 있다. 신뢰성은 악의적인 의도로 발생할 수 있는 버퍼 오버플로, 메모리 릭 등의 보안 취약점이 존재하지 않아야 한다는 것이고, 예측 가능한 실행은 소프트웨어가 개발 시 의도한대로 실행되어야 한다는 것 이다. 소프트웨어 보증을 위한 작업은 소프트웨어 개발 생명 주기의 개발과 유지보수 단계 모두에서 수행되어야 한다. 국외 기관인 NIST, SANS 등은 개발 단계에서의 보증을 위해 필요한 툴, 프로그래밍 가이드라인, 오류 식별 문서 등을 공개하고 관련 프로젝트를 지원하였다. 본 논문에서는 상대적으로 연구가 부족한 유지보수 단계에서의 소프트웨어 보증을 위해 신뢰와 평판 기반의 소프트웨어 보증 시스템을 구현하였다. 본 시스템은 사용자에게 해당 소프트웨어에 대한 다른 사용자의 평판과 다양한 의견을 제공하고, 이에 따른 신뢰도를 함께 제공한다. 이는 사용자가 안전한 소프트웨어를 선별하고 사용하는데 도움을 준다.

1. 서 론

본 논문의 핵심은 소프트웨어 보증(Software Assurance)이다. 소프트웨어 보증은 국내 보다는 국외에서 많은 관심을 가지고 연구되고 있다. DHS(Department of Homeland Security), DoD(United States Department of Defense), NIST(National Institute of Standards and Technology), SANS(SysAdmin, Audit, Network, Security) 등과 같은 기구 및 포럼에서 '소프트웨어 보증'에 대한 연구 개발을 활발하게 진행하고 있다. 여기서 각 기구에서 소프트웨어 보증에 대한 정의를 하고 있는데 약간의 차이가 있지만 다음과 같은 두 가지의 내용을 포함한다[1][2][3][4]. 첫째, 신뢰성(trustworthiness)을 가져야 한다. 악의적이거나 의도하지 않은 원인 등으로 발생하는 취약성이 존재하지 않아야 한다. 둘째, 개발된 소프트웨어는 예측 가능한 실행(Predictable Execution)을 해야 한다. 소프트웨어는 의도한 기능을 실행했을 때 올바르게 계획대로 실행해야 한다. 이러한 내용은 어떻게 보면 당연하다고 생각되는 부분이지만, 해외의 다양한 사례와 연구 내용들을 보면 매우 중요하다는 사실을 알 수 있다. 국내의 경우에는 해외에 비해 소프트웨어 보증에 대한 관련 자료 및 연구단체 등을 찾기가 매우 어렵다. 국외의 경우에는 소프트웨어 보증에 대한 중요성을 인지하고 다양한 기구 및 연구단체 등을 만들어 활발한 연구 개발이 진행 중이다.

NIST와 SANS는 가장 활발하게 연구 개발을 하고 있는 소프트웨어 보증 기구이다. SANS에서는 CWE(Common Weakness Enumeration)와 함께 개발자의 실수나 공격자(Attacker)의 공격으로 인한 프로그래밍 오류 700개 이상을 식별하여 공개하였다[5]. 공개된 오류는 단순히 오류의 내용만을 공개한 것이 아니라, 오류가 발생 했을 때의 증상, 오류 발생 예, 예방 방법 등 유용한 정보를 제공하여 중대한 프로그래밍 오류를 인지하고 예방 할 수 있도록 도움을 준다. NIST에서는 DHS 국제 사이버 보안부서와 협력하여 소프트웨어 보증 툴을 개발하는 SAMATE(Software Assurance Metrics And Tool Evaluation) 및 R&D 요구 검증 프로그램 프로젝트를 지원하였다. 이와 동시에 사용자와 개발자에게 잘 알려진 보안 약점을 정리한 데이터집합을 제공하는 SRD(SAMATE Reference Dataset) 프로젝트를 수행하였다[6]. 이러한 연구들은 소프트웨어 개발 생명 주기(SDLC:Software Development Life Cycle)에서 개발 단계에서의 소프트웨어 보증을 위한 것 이다. 이에 비해 소프트웨어를 배포한 이후, 즉 유지보수 단계에서의 소프트웨어 보증에 대한 연구는 상대적으로 부족한 상황이다.

SDLC의 유지 보수 단계에서의 소프트웨어 보증의 하나로, 배포된 소프트웨어의 신뢰성을 평가하여 제공하는 것을 생각할 수 있다. 많은 소프트웨어가 인터넷 블로그나 P2P, 웹디스크 등을 통해 전파되면서 바이러스나 악성코드가 함께 전파되어 사용자에게 피해를 주고 있는 것이 사실이다. 이러한 피해를 막기 위해서 해당 소프트웨어를 식별하고 신뢰할 수 있는지 판단하는데 도움을 줄 수 있는 소프트웨어 보증 시스템을 구현 하였다.

소프트웨어의 신뢰도를 평가하기 위해 대규모 오픈 시

† 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2009-0090108).

스템에서 널리 사용 중인 사용자의 피드백에 기반한 신뢰 및 평판 메커니즘을 사용한다. 본 논문에서 제안하는 소프트웨어 보증 시스템은 사회망(Social Network)을 통해 이뤄지는 사용자 평가를 기반으로 소프트웨어에 대한 신뢰도와 다른 사용자들의 의견 정보를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 살펴본다. 3장에서는 신뢰와 평판 기반의 소프트웨어 보증 시스템의 설계, 4장에서는 구현에 대해 기술한다. 이어서 5장에서는 결론을 내린다.

2. 관련연구

소프트웨어의 신뢰도는 비기능적인 측면에 속한다. 이러한 비기능적인면을 평가하기 위한 방법으로 사용자의 피드백에 기반한 신뢰 및 평판 메커니즘이 있다[7]. 일반적으로 평판은 서비스에 대한 특징, 정직성, 능력, 신뢰도 등에 대한 대중의 집합적인 의견을 말하며, 신뢰는 한 개인의 의견이 반영된 주관적인 값을 의미한다[8]. 이와 같은 신뢰와 평판 메커니즘은 전자상거래[9], P2P [10], 멀티 에이전트 시스템[11]과 같은 대규모 오픈 시스템에서 널리 적용되고 있다. 대표적인 P2P 프로그램인 당나귀[12]나 프루나[13]와 같이 파일을 신고하는 방식도 신뢰와 평판 메커니즘을 사용한 예라고 볼 수 있다. 프루나의 경우 가짜 자료로 신고된 파일명 앞에 "?"를 표시하며, 신고 건수와 의견정보를 보고 사용자 스스로 신뢰성을 판단하게 하는 방식을 사용한다.

소프트웨어의 신뢰도를 평가하기 위해서는 소프트웨어를 식별할 수 있어야 한다. P2P 프로그램인 프루나의 경우 해시함수 등을 사용한 "파일 ID" 라고 부르는 키 값을 이용해 파일을 구분한다. 하지만 소프트웨어는 여러 개의 파일로 구성되는 경우가 많아 단순히 파일하나를 구분하기 위해 키 값을 만드는 방식은 사용할 수 없다.

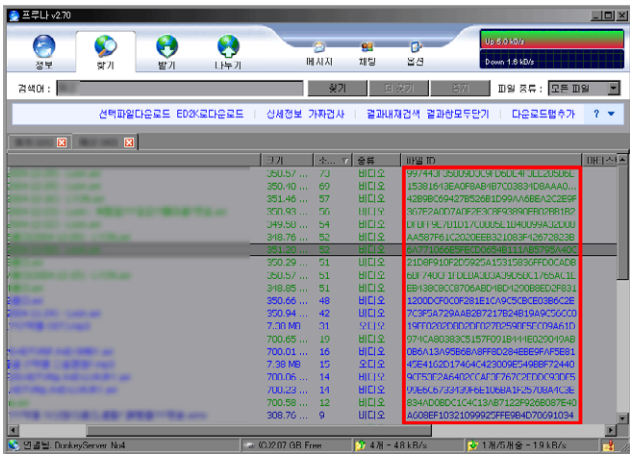

그림 1. 프루나의 파일 ID

3. 신뢰와 평판 기반 소프트웨어 보증 시스템의 설계

신뢰 및 평판 기반 소프트웨어 보증 시스템은 어떤 특정 소프트웨어를 사용자가 설치하기 전에 이것이 믿을 만한 소프트웨어인지 검증하기 위해 사용한다. 그림 2는 소프트웨어 보증 시스템의 시퀀스 다이어그램을 보여준다.

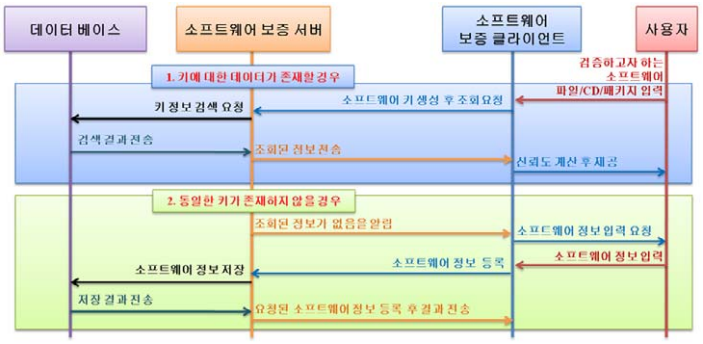

그림 2. 소프트웨어 보증 시스템 시퀀스 다이어그램

사용자는 소프트웨어 보증 클라이언트를 활용하여 검증하고자하는 소프트웨어(파일/CD/패키지)를 입력 후 검사 요청을 한다. 클라이언트는 소프트웨어 키를 생성 후 조회를 요청하고, 서버는 동일한 키에 대한 데이터가 존재할 경우 관련 정보를 클라이언트에게 전송한다.클라이언트는 응답받은 정보를 이용하여 해당 소프트웨어에 대한 신뢰도, 다른 사용자들의 신고, 댓글내용을 제공한다. 신고내용은 소프트웨어 사용 후 피해를 입은 사례, 사용 시 문제점 등을 포함하고, 댓글 내용은 사용 후기, 의견 등을 포함한다. 동일한 키가 존재하지 않을 경우에는 사용자에게 해당 소프트웨어 정보입력을 요청하고 응답을 받아서 소프트웨어 정보를 등록한다.

본 논문에서 제안하는 시스템 사용의 실제적인 예로써 공개 소프트웨어 자료실이나 블로그 등에서 소프트웨어 설치 파일을 다운로드 했다고 가정해 보자. 이때, 해당 소프트웨어가 개발사가 정식으로 배포한 것인지 확인하기 위해 사용할 수 있다. 이 시스템은 해당 소프트웨어에 대한 신뢰도를 제공한다. 또한 해당 소프트웨어 이미지에 대한 다양한 사람들의 의견 정보를 공유할 수 있다. 이것을 통해 구체적인 피해 사례 등을 확인 할 수 있으며, 신뢰도가 높다고 하여도 신빙성 있는 정보 인지 확인할 수 있는 장점을 갖는다. 이러한 시나리오를 달성하기 위해 다음과 같은 기능들을 제공한다.

첫째, 소프트웨어에 따라 파일, CD, 패키지 등을 통해 고유한 키를 생성하여 소프트웨어를 식별 할 수 있는 기능을 제공한다. 이러한 키를 통해 같은 이름의 소프트웨어라도 버전에 따라 혹은 변경된 소프트웨어인지 구분이 가능하다. 둘째, 데이터베이스 서버를 구축하여 소프트웨어 고유 번호를 관리하고, 이에 대한 신뢰도를 계산할 수 있는 정보를 관리한다. 또한, 해당 소프트웨어에 대한 댓글을 등록 혹은 조회를 할 수 있도록 제공하여 일반 사용자들에게 소프트웨어에 대한 검증 정보를 제공한다. 마지막으로, 소프트웨어의 신뢰도 계산을 위해 데이터베이스에 저장한 정보를 활용하여 신뢰도를 계산한다.

3.1 고유키 생성 모듈

신뢰와 평판 기반의 소프트웨어 보증 시스템에서는 소프트웨어 보증 서버의 데이터베이스에 저장된소프트웨어와 검사 대상이 되는 소프트웨어를 식별 및 비교하여 일치 여부를 판단한다. 소프트웨어의 식별을 위해서는

데이터 해싱 알고리즘(SHA-256)을 활용해 해당 소프트웨어(파일/CD/패키지 등)의 64자리의 고유한 키 값을 생성한다. 키 값을 생성해내는 과정에서는 파일의 크기가 커지면, 너무 많은 양의 데이터를 바탕으로 키 값을 생성해내야 하기 때문에 오랜 시간이 걸리는 문제점이 존재한다. 그래서 해싱 알고리즘을 사용할 때 파일 사이즈가 256Mb 이상이 되는 데이터는 1Kb단위로 샘플링 기법을 사용한다. 샘플링 기법에는 확률 샘플링과 비확률 샘플링이 있지만, 본 논문의 개발 시스템에서는 동일한 표본에 대해서 동일한 결과 값이 나와야 하기 때문에 비확률 샘플링을 사용한다. 그래서 확률 샘플링방법 중 하나인 Systematic 샘플링 기법에서 시작점을 고정하여 비확률 샘플링으로 변경 후 사용한다[14].

$$k = \frac{N}{262144} \quad \cdots\cdots (1)$$

(k= 샘플링 표본의 간격, N= $1Kb$단위의 파일 사이즈)

샘플링 기법은 (1)과 같은 식을 사용하였고, 표본의 시작점은 고정된 위치 1로 하였다. 예를 들어 512MB인 파일을 사용하는 경우, 표본은 1, 3, 5, ..., 1 + 262143 * 2번째가 된다. 하지만 확률 샘플링 방법을 수정한 방식이기 때문에 표본이 주기적 성질을 갖는 데이터일 경우 좋지 않은 표본이 추출될 수 있다. 그래서 이러한 문제점을 보완하고자 해싱 알고리즘 중간에 파일사이즈와 해당 파일의 상대경로와 같은 정보들을 데이터에 포함하여, 고유한 키를 생성한다.

3.2 데이터베이스

그림 3은 서버 프로그램에서 자료 저장 및 관리를 위해 사용하는 데이터베이스의 ER(Entity-Relationship) 다이어그램을 나타낸다.

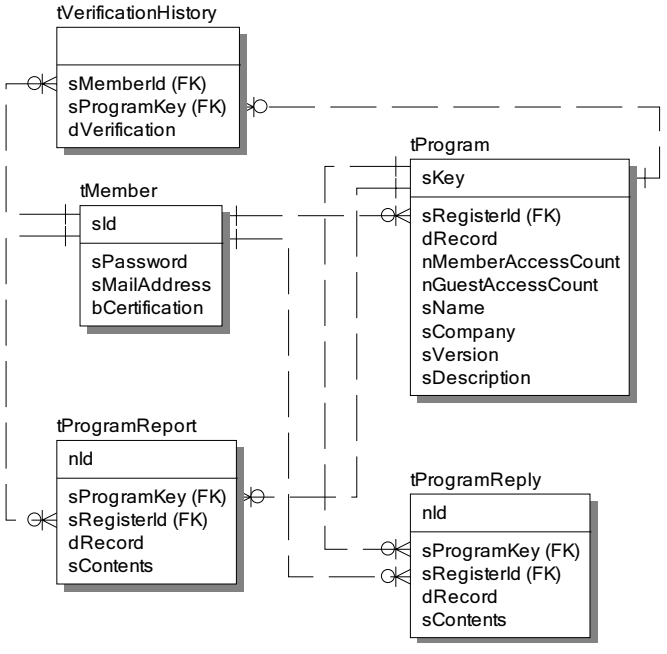

그림 3. 데이터베이스 ER 다이어그램

데이터베이스는 다음 표 1과 같이 총 5개의 테이블로 구성된다.

표 1. 데이터베이스 테이블 설명

테이블 명	설명
tMember	■회원정보 관리용 테이블 ■회원 아이디, 패스워드, 이메일 주소, 회원 인증 여부의 정보를 저장
tProgram	■소프트웨어 정보 관리용 테이블 ■검증 횟수는 신뢰도 계산에 사용되고, 소프트웨어 이름, 회사, 버전, 세부정보 등의 정보를 저장
tProgram Replay	■댓글 정보 관리용 테이블 ■댓글을 단 사용자, 댓글 내용, 댓글이 달린 시간 정보를 저장
tProgram Report	■신고 내용 관리용 테이블 ■신고를 한 사용자, 신고 시간, 신고 내용 정보를 저장
tVerificationHistory	■소프트웨어의 검사 기록 관리용 테이블 ■검사를 요청한 사용자, 검사 대상 소프트웨어, 검사 시간 정보를 저장

3.3 신뢰도 계산 모듈

신뢰도 계산 모듈은 검사 횟수, 신고 횟수, 전체 회원수 정보를 이용하여 신뢰도를 계산한다.

$$\text{신뢰도}(\%) = 50 - (n*10) + \frac{k}{4*m*(n+1)} \quad \cdots\cdots (2)$$

(n= 신고 횟수, m= 전체 회원수, k= 검사횟수)

수식 (2)에서 나타난 바와 같이 기본 신뢰도는 50%에서 시작하고, 신고횟수에 따라 10%씩 감소한다. 검사횟수는 신고횟수와 전체 회원수에 반비례한다. 즉, 신뢰도는 검사횟수가 증가할수록 증가하고, 신고횟수와 전체회원수가 증가할수록 감소한다. 이는 검사횟수의 증가에 따른 신뢰도의 급격한 증가를 예방할 수 있다.

본 시스템에서는 수식 (2)를 활용하여 회원과 손님의 신뢰도를 개별적으로 계산하고, 이를 더해서 전체 신뢰도를 계산한다. 전체 신뢰도는 회원과 손님의 신뢰도를 7:3의 비율로 적용한 후 더한 값으로 회원의 의견이 더 큰 비중을 차지하도록 한다.

4. 신뢰와 평판 기반 소프트웨어 보증 시스템의 구현

신뢰와 평판 기반 소프트웨어 보증 시스템은 서버-클라이언트 모델을 활용하여 구현하였다. 서버 프로그램은 사용자, 소프트웨어 관련 정보를 관리하는 데이터베이스와 연동하여 클라이언트 프로그램의 요청이 있을 때 해당 정보를 응답한다. 클라이언트 프로그램은 응답받은 정보를 바탕으로 신고내역, 댓글, 신뢰도 등의 정보를 사

용자에게 제공한다. 또한, 신고내용, 댓글, 해당 소프트웨어 정보 등록을 서버 프로그램에 요청하여 데이터베이스에 저장 및 관리할 수 있다.

4.1 서버 프로그램

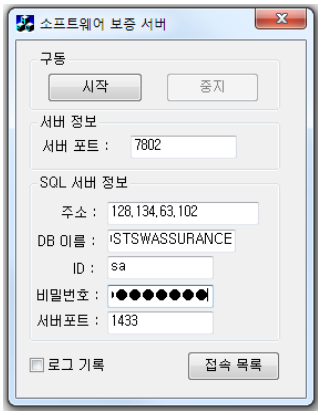

그림 4. 서버 프로그램

본 연구에서 개발한 서버 프로그램은 그림4와 같다. 서버 프로그램은 서버 포트를 설정하는 서버정보 부분 데이터베이스 접속을 위해 입력하는 주소, 이름, 아이디, 비밀번호, 포트를 설정하는 SQL 서버정보 부분으로 구성되어 있다. 서버 프로그램에서는 데이터베이스와의 연동을 위해서 데이터베이스 풀을 사용하였다. 그 유는 데이터베이스 연결에 필요한 라이센스의 수가 제한적이고 하나의 연결로 모든 클라이언트들의 처리를 해결할 수 없기 때문이다. 데이터베이스 풀은 스레드 풀과 같이 미리 자원을 할당해놓고 필요할 때 할당받아 쓰고 반환하는 정책과 같다. 여기서 말하는 자원을 데이터베이스와의 연결을 의미한다. 또한, 통신의 신뢰성 있는 패킷의 처리를 위해 우드블록 및 버퍼링 처리 루틴을 구현 하였다. 또한, 고용량의 데이터 처리를 위한 데이터의 압축 해제를 통한 패킷 송신/수신 루틴을 구현 하였다.

4.2 클라이언트 프로그램

그림 5. 클라이언트 프로그램

클라이언트 프로그램의 개략적인 화면은 그림5와 같다. 본 프로그램은 크게 검사화면과 등록화면으로 나눠진다. 검사화면에서는 해당 소프트웨어를 선택한 후 검사를 할 수 있고, 이에 따라 신뢰도 정보를 얻을 수 있다. 이와 동시에 해당 소프트웨어의 문제점이 발견된 이력이 있는지 신고내역을 열람할 수 있고, 사용 후 신고를 할 수도 있다. 또한, 관련댓글을 열람 및 등록 할 수 있다. 등록화면에서는 아직 데이터베이스에 등록되지 않은 소프트웨어를 등록한다. 클라이언트 프로그램은 사용자에게 입력받은 해당 소프트웨어의 이름, 회사, 버전 등의 정보와 생성된 키 값을 서버 프로그램에 전송하여 등록을 요청한다. 등록된 소프트웨어는 향후 사용자들이 해당 소프트웨어를 검사할 때 사용된다.

신뢰와 평판 기반 소프트웨어 보증 시스템은 사용자들의 평판 및 이를 바탕으로 한 신뢰도가 중요한 요소이기 때문에 클라이언트 프로그램에서의 주요 기능은 검사, 댓글, 신고 관련 기능을 제공 및 관리하는 것이다.

4.2.1 검사 기능

그림 6은 D:\설치\java\eclipse.tar을 선택하고, 검사시작 버튼을 눌러 해당 소프트웨어에 대한 검사를 수행한 후의 화면이다.

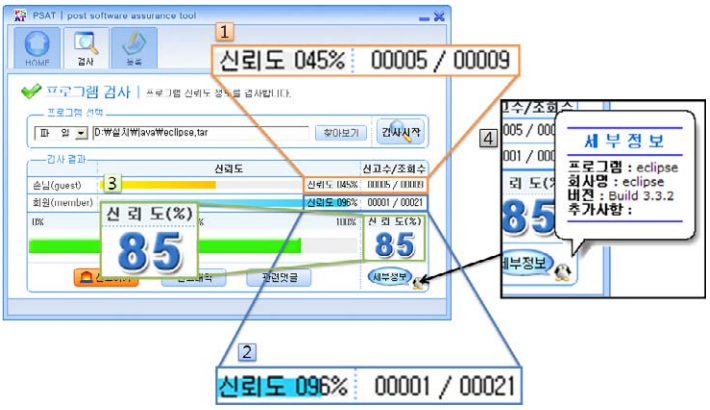

그림 6. 검사 화면

다음은 그림 6에서 1~4 까지 표시된 부분의 구성요소에 대한 설명이다.

① 손님(guest)의 신뢰도 정보
그림 6에서 1번으로 표시된 부분으로 해당 프로그램에 대한 손님의 조회수는 9이고, 신고횟수는 5이다. 이 정보를 바탕으로 계산된 신뢰도는 45%이다. 그래프는 45%에 해당하는 위치까지만 나타나고, 그래프 끝부분에 자세한 신뢰도가 숫자로 나타난다.

② 회원(member)의 신뢰도 정보
손님의 신뢰도 정보와 마찬가지로 보는 방식은 동일하고, 현재 조회수는 21번 신고횟수는 1번으로 신뢰도가 96%이다.

③ 통합 신뢰도
통합 신뢰도는 손님과 회원의 신뢰도를 바탕으로 각각에 가중치를 적용한 후 더한 결과로써 현재 이 프로그램에 대한 신뢰도는 85%에 해당한다. 통

합 신뢰도 또한 수치 및 그래프를 보여준다

④ 세부 정보

해당 프로그램에 대해 다른 사용자가 남긴 세부 정보로써, 현재 정보 상으로는 프로그램명이 eclipse이고, 회사명도 eclipse 버전은 Build 3.3.2 버전이다. 그리고 추가사항이 등록되어 있지 않다 만일 세부 정보 중 등록되지 않은 부분이 있으면 검사가 완료되었을 때 추가등록 여부를 묻는 메시지가 나타난다. 여기서 예를 선택하면, 세부정보에 등록되지 않은 나머지부분을 등록할 수 있다

4.2.2 댓글관리 기능

소프트웨어 보증 시스템은 다른 사용자들이 소프트웨어의 사용 후 의견을 입력한 댓글 내역을 확인 할 수 있고, 직접 댓글을 등록할 수 도 있다. 그림 7은 댓글보기 및 등록 화면이다.

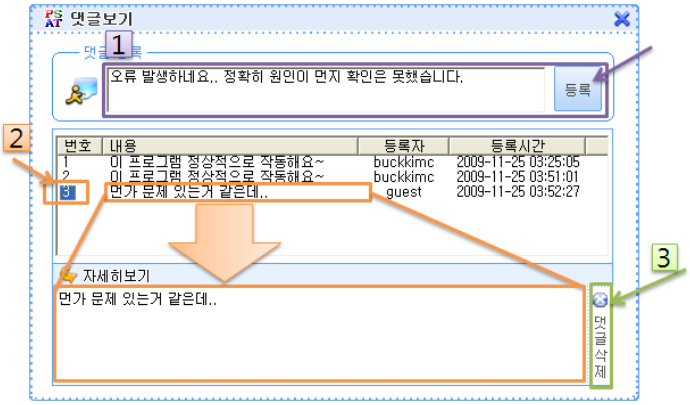

그림 7. 댓글 보기 화면

다음은 그림 7에서 1~3 까지 표시된 부분의 구성요소에 대한 설명이다.

① 댓글 등록

그림 7에서 1번으로 표시된 부분으로 원하는 내용의 댓글을 입력하고 댓글 버튼을 누름으로써 댓글을 등록 할 수 있다.

② 댓글 자세히 보기

자세히 보고 싶은 댓글을 선택하면 선택된 댓글에 대한 내용이 자세히 보기 부분에 나타난다

③ 댓글 삭제

삭제하고자 하는 댓글을 선택한 후 댓글삭제 버튼을 누르면, 등록자를 확인한 후에 등록한 사람과 같으면 삭제여부의 확인 후에 해당 내용을 삭제할 수 있다. 만일 등록자와 삭제하려고 시도하는 사용자가 다르다면, 삭제할 수 없다는 경고 메시지를 띄운다.

4.2.3 신고관리 기능

신고내역 보기는 사용자가 소프트웨어 보증 시스템을 사용할 때 가장 중요하게 봐야할 부분으로 사용자가 사용하고자 하는 프로그램을 검사한 후 이 프로그램에 신뢰도가 좋지 않으면, 신고내역을 봄으로써 다른 사용자들이 어떠한 평가를 했는지 확인 할 수 있다 이 내용들을 바탕으로 프로그램의 사용여부를 결정하는데 도움을 준다.

신고하기는 해당 소프트웨어의 신뢰도를 평가하는 가장 중요한 수단으로, 사용자가 프로그램을 사용하다 문제점을 발견하면, 해당 소프트웨어를 신고하고 내역을 남긴다. 이와 같은 과정을 통해 다른 사용자들이 문제가 있는 소프트웨어를 사용하지 않도록 예방할 수 있다

그림 8은 신고내역 보기 및 신고하기 화면이다

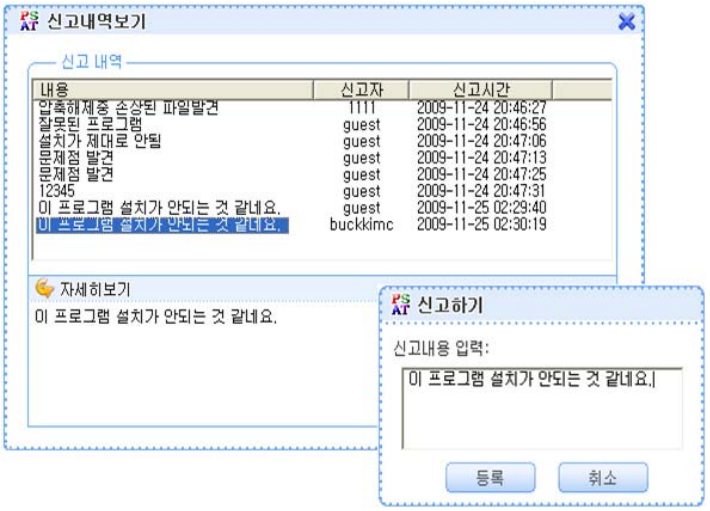

그림 8. 신고내역 보기 및 신고하기 화면

5. 결론

소프트웨어 보증에 대한 연구는SDLC의 개발 단계에 집중되어 있어 유지보수 단계에서의 연구가 부족했다

본 논문에서는 사용자 피드백을 통한 신뢰와 평판 기반의 소프트웨어 보증 시스템을 구현하여SDLC의 유지보수 단계에서의 소프트웨어 보증을 제공할 수 있는 방법을 제시하였다. 이러한 소프트웨어의 신뢰도 평가는 사회망(Social Network)을 이용한 것으로써, 클라이언트 프로그램을 통해 보다 쉽게 평가정보가 공유될 수 있도록 하여 안전한 소프트웨어인지 판단하는데 도움을 준다

참 고 문 헌

[1] DHS(Department of Homeland Security), “http://www.dhs.gov/index.shtm”

[2] DoD(United States Department of Defense), “http://www.defense.gov/”

[3] NIST(National Institute of Standards and Technology), “http://www.nist.gov/index.html”,

[4] SANS(SysAdmin, Audit, Network, Security), “http://www.sans.org/”

[5]. Bob Martin(MITRE), Mason Brown(SANS), Alan Paller(SANS), “http://cwe.mitre.org/top25/”, (CWE)Common Weakness Enumeration, 2009

[6]. “http://samate.nist.gov/index.php/Main_Page.html”, NIST – SAMATE Project

[7] 하미수, 이경호, “신뢰와 평판 기반의 웹 서비스 선

택", 한국정보과학회 2008가을 학술발표논문집, 제35권, 제2호, pp.17-21, 2008
[8] Y. Wang and J. Vassileva, "Toward Trust and Reputation Based Web Service Selection: A Survey", International Transactions on Systems Science and Applications, Vol.3, No.2, pp. 118-132, 2007
[9] G. Zacharia, A. Moukas, and P. Maes, "Collaborative Reputation Mechani는 in Electronic Marketplaces", Decision Support Systems, Vol. 29, No. 4 pp.371-388, 2000
[10] Y. Wang, and J. Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks", Proc. 3rd Int'l Conf. Peer-to-Peer Computing, pp 150. 2003
[11] T. T. Hyunh and N. R. Jennings, "An integrated trust and reputation model for open multi-agent systems", Autonomous Agents and Multi-Agent Systems, Vol. 13, No. 2, pp. 119-154, 2006
[12] 당나귀P2P, "http://www.donkeyp2p.com/"
[13] PRUNA, "http://www.pruna.com/"
[14] wikipedia, "http://en.wikipedia.org/wiki/Systematic_sampling"