NFR을 이용한 자체 전파 웜의 스테레오 타입 정의에 관한 연구

조규형, 이혁준, 임종인, 문종섭
고려대학교 정보경영공학 전문대학원
{mathbank, rainhalt, jilim, jsmoon}@kore.ac.kr

Self-propagation Worm definition of stereo type using NFR

Kyuhuyng Cho, Hyukjoon Lee, Jongin Lim, Jongsub Moon
Graduate School of Informaiton Management and Security

요약

네트워크 상에서 활동하는 웜을 모델링하는 연구는 특정 웜에 한정되어 있다. 따라서 기존에 발표된 웜의 확산 모델링 연구는 그 범위를 다른 수많은 웜으로 확장하기에 어려움이 따르며, 이를 위한 표준화 연구도 부족한 실정이다. 따라서 본 연구에서는 Non-fuction requrirement(NFR)의 개념을 이용하여 웜의 속성을 정의하고 이 정의를 바탕으로 자체 전파되는 웜의 표현 기법을 제안한다. 현재로서는 사용자의 추가적인 작동을 요구하지 않는 자체 전파 웜에 대하여 한정하고 있으나, 이를 확장하면 다양한 형태의 웜을 표현할 수 있는 도구가 될 수 있다.

1. 서론

인터넷의 발달로 인한 통신의 편리함은 네트워크에 악영향을 미치는 웜이 편리하게 활동할 수 있는 환경을 제공하게 되었다. 네트워크에서 악성 코드의 활동은 사회 전반적으로 큰 영향을 미칠 수 있음이 현실적인 일련의 사건으로 인하여 증명되었으므로 이에 대한 대책 마련에 부심하고 있는 실정이다. 일반적으로 웜이나 바이러스 등의 악성 코드가 발생될 경우 그 파급력이 어느 정도나 될 것인가를 쉽게 예측하기는 어렵지만, 이를 보완하기 위하여 다양한 형태의 모델링 및 시뮬레이션 연구가 진행 중이다. 그러나 대부분의 연구에서는 웜의 감염 정도 등에 국한하여 실제 감염되었을 경우와 유사한 형태의 전파율을 파악하고, 이를 모델링 하는데 초점을 두고 있다[1][2][3]. 그러다보니 보다 정밀한 시뮬레이션 결과를 도출하는 과정에서 추가적으로 고려해야 할 웜의 특성이나 활동 등은 충실하게 적용하기에 어려움이 따른다. 본 논문에서는 웜의 활동 상황을 가급적이면 포괄적으로 표현할 수 있도록 Non-Fuctinal Requrirement(NFR)[4] 개념을 이용하여 웜을 기능별로 추상화 하고 웜의 속성을 정의한다. 이러한 정의는 추후에 웜을 다양한 언어로 표현하기 위한 기준으로서 그 의의를 가질 수 있다.

본 논문의 순서는 다음과 같다. 2장에서는 웜의 특징 분류를 위한 기존 연구를 서술하고, 웜의 활동을 NFR을 사용하여 정의할 수 있는 연구에 대하여 고찰한다. 3장에서는 웜의 각 행동 패턴들을 NFR에 맞게 기능분류를 하고 이를 스테레오 타입으로 정의한다. 이후 4장에서 본 논문의 결론을 내리도록 한다.

2. 관련 연구

가. 웜의 분류

웜에 관한 연구는 웜이 전파되는 방식에 따른 분류와 웜의 행동 패턴에 따른 분류로 진행되고 있다. 웜의 전파방식에 따른 분류 기법은 시만텍 사의 Darrell M. Kienzle 등의 연구[5]에 따르면 웜을 전파방식에 따라 E-mail 웜, Winodws File Sharing Worm, 전통적인 웜 등으로 분류할 수 있다. Nazario 등의 연구[6]에서는 웜의 미래를 이야기하면서 웜의 기능을 7가지로 분류하였는데, 첫째, 자신을 전파하는 기능, 둘째, 웜 네트워크의 형성, 셋째, 감염된 웜끼리 통신을 위한 구성, 넷째, 통신 방법, 다섯 번째, 웜을 전파하기 위한 스캔, 여섯번째, 전파된 호스트 상에서의 행동, 일곱 번째, 웜을 업데이트하기 위한 방법 등으로 세분화 될 수 있다. Kienzle의 분류가 웜의 전파 방식에 초점을 둔 반면, Nazario의 연구는 웜의 기능을 전체적으로 나누어 분류했다는 것에 의의가 있다. Weaver는 웜을 5단계의 활동요소로 분류하고 각 활동요소에서 발생하는 구체적인 기능들을 정의하였다. Target Discovery 단계에서는 공격 대상을 차기 위하여 스캔하는 행위를 수행하며, Carier 단계에서는 웜 코드를 공격 대상 호스트에 전송한다. Activation 단계에서는 웜이 감염시킨 호스트에서 활동을 시작하기 위한 조건을 구성하며, Payloads 단계에서는 웜 자체의 전파 이외에 부수적인 공격 활동 및 명령의 실행을 수행한다. 이와 함께 웜을 제작하게된 동기를 Motivation and Attackers로 분류하였다. Weaver는 웜의 시작단계인 취약한 호스트를 찾는 단계부터 마지막 단계인 공격의 행위까지 과정을 체계적으로 분류하여 웜이 활동하는 과정에서 수행되는 모든 단계를 규정하였다. 이와 같이 앞에서 진행된 연구들에서는 모두 웜에 대하여 활동이나 구성 요소등을 규정할 수 있는 일정한 형식을 구성하기 위하여 노력하였다.

나. NFR을 이용한 웜의 표현

NFR은 Non-Functional Requirements의 약자로 비기능적 요구명세를 지칭한다. NFR은 시스템이 정상적으로 작동하기 위한 제약사항들로 성능, 안정성, 보안 등의 범주를 다룬다. NFR은 시스템 전체에 걸쳐 표현되고 매우 다양한 상태를 나타낼 수 있기 때문에 복잡하고 추상적인 특징이 있다. 예를 들어 운영체제에 대한 NFR을 서술할 경우 "이 운영체제는 안전하다" 라고 표현된다. 여기서 "안전하다"라는

표현이 추상적이다. 이렇게 추상적으로 표현되는 NFR의 특징으로 인하여 정형화 혹은 적합성을 검증하는 것은 쉽지 않다[4]. 특히 NFR의 비가시적인 특징으로 인해 많은 NFR이 단순히 글로 표현되고 있다. 기존 연구에서는 NFR을 Functional Requirement[FR]과 비교하여 NFR의 특징을 찾고 정형화 하는 연구가 있다[7].

MIT에서는 여러 가지 공격에 대한 데이터를 기반으로 네트워크 공격을 표현하기위한 NFR 스테레오 타입을 제작하였다. 스테레오 타입이란 "동일한 형태이나 다른 의도로 기존에 존재하는 모델링 요소를 변형할 수 있게 하는 것이다[8]". 즉 이러한 스테레오 타입을 통해 기존의 UML 모델링에 새로운 요소를 도입할 수 있다. 기존의 UML은 특정 공격에 대한 부분을 자세하게 표현하려면 주석처리나 많은 Note들이 필요하지만, 스테레오 타입의 정의를 통해서 정형화 된 틀 안에서 공격을 간단히 UML로 표현할 수 있다.

Mohammed Hussein 등은 MIT에서 분석한 공격 데이터를 이용하여 침입 시나리오를 NFR로 정의하고 이것을 UNL로 재구성하여 UMLintr 스테레오타입을 정의하였다[7].

표 1 UMLintr stereotypes

Stereotype	Base class	Description
⟨⟨DOS⟩⟩	Package	Denial of service attacks
⟨⟨RemoteToUser⟩⟩	Package	Remote to user attacks
⟨⟨UserToRoot⟩⟩	Package	User to root attacks
⟨⟨Probe⟩⟩	Package	Probe attacks
⟨⟨attacker⟩⟩	Actor	Attacker actor
⟨⟨victim⟩⟩	Actor	Attacker actor
⟨⟨abuse feature⟩⟩	Use-case	Methods of exploitation
⟨⟨deny⟩⟩	Connector	Actions performed by attacker

표 2 UMLintr tagged values

Tagged value	Description
initial-privilege	Represents the initial access level of the attacker
gained-privilege	Represents the gained access level of the attacker
skill	The skill of the attacker
action	The action of the attacker
security	The security level of the victim
severity	The severity level of the attack
method	The exploitation method of the attack
listeners	Events needed to detect attacks

UMLintr에서는 NFR을 통하여 공격을 표현하기 위해서 UML을 도입하였으며 4개의 다이어그램(Use-case, Package, State-machine, Class Diagram)을 사용하였으며[표 1 참조], 5개의 엔티티(Package, Actor, Use-case, Class, Connector)를 사용하였다. MIT에서 분석한 공격을 4가지로 나누고 각각의 4가지 타입의 공격을 Package로 표현하였다. 또한 다이어그램에서 공격자와 피해자를 Actor로 표현하였으며, 피해 상태를 Use-case로 표현하였다. 그리고, 공격에 대한 정적 구조를 Class 다이어그램으로 표현하여, <<attacker>>, <<victim>>, <<intrusion>>의 클래스를 구성하였다. Connector는 공격에 대한 구체적인 형태를 표현하는데 이용하였으며, 공격에 대한 동적 행위는 Use-case, State-machine으로, 정적인 상태는 Package, Class 다이어그램으로 표현하였다.

3. 스테레오 타입 정의

가. 웜 기능의 추상화

일반적인 웜을 모델링하기 위하여 각 기능을 추상화 하고 객체지향 언어인 UML을 이용하여 세부 기능을 구성할 경우 해결해야 할 문제점이 발생된다. 우선 웜의 특징을 명확하게 정의하지 않은 상태에서 UML로 표현을 하게 되면 많은 주석들과 제한으로 인하여 UML이 상당히 복잡하게 된다. 또한, 웜에 포함되는 많은 기능들을 표현하기 위해 함수를 사용하게 된다면 Class 다이어그램의 함수 부분은 시스템의 고유한 함수들로 채워지게 되므로 UML로 웜을 표현하는 것은 매우 어려운 작업이 된다. 이러한 이유로 인하여 웜의 동작을 기능별로 분류하고, 각 동작을 체계적으로 추상화 하는 작업이 필요하게 된다.

본 논문에서는 기존에 발견된 웜을 기능별로 분석하고, 리버스엔지니어링을 통하여 도출한 데이터를 이용하여 웜의 행위를 스테레오 타입에 적합한 형태로 표현한다.

본 논문에서는 웜의 행위를 표현하기 위하여 다음과 같은 방식으로 웜의 기능을 추상화 한다.

표 3 웜 기능의 추상화

단계	설명
System occupation	웜이 취약한 시스템을 공격하여 점령하는 행위
Code transfer	공격 대상 호스트로 웜의 코드를 전송
Payload	시스템 점령 이후의 행위

웜을 이상과 같은 3단계로 분류하게 되면 현재 웜의 동작 상태를 단계별로 구분지을 수 있으며, 각 단계에 따라서 웜의 특징을 구분하는 요소를 표현하기가 용이하다. 특히, 위와 같은 단계를 기준으로 특징의 범주를 규정함으로서 현재 진행중인 웜의 행위가 각 단계에서 어떠한 단계인지를 파악하고 모델링할 수 있게 된다.

나. 스테레오 타입 정의

본 연구에서는 Package, Class, Actor 및 Connector를 도입하여 각 단계에 포함되어있는 웜의 행위를 표현한다. UML에서 Package는 의미적으로 관련된 것을 하나로 묶음으로써 구조화된 모델을 제공한다[9]. 시스템이 점령당하는 단계인 System occupation은 하나의 웜이 수행하는 여러 단계의 행위들을 포함해야 하므로, 최종적인 목적을 의미하는 시스템 점령을 달성하고자 수행되는 일련의 과정들이 포함되

어야 한다. 따라서 이러한 일련의 행위들을 포괄할 상위 범주의 개념을 표현하기 위하여 Package로 규정하였다.

스테레오 타입을 정의하는 형식은 <<...>> 안에 스테레오 타입을 정의하는 형식으로 이루어진다. 이러한 스테레오 타입을 통해서 기존의 UML 과 다르게 웜을 위한 모델링이 가능하게 되었다. 이러한 패키지 System occupation을 표현하기 위한 스테레오 타입을 다음과 같이 정의한다.

표 4 System occupation 스테레오 타입 정의

Stereotype	Base class	Description
<<System occupation>>	Package	시스템을 점령하는 단계

이후에는 System occupation 단계에서 웜이 수행하는 행위를 세부적으로 구분하여 각 행위들을 위의 스테레오 타입 정의와 유사한 형식으로 정의한다. <<Vulnerability>>는 웜이 호스트를 공격할 경우 이용하는 보안취약성을 의미한다. 취약성이 존재하지 않는 호스트는 웜에 감염되지 않으며, 취약한 호스트는 공격을 당할 수 있는 가능성이 있으므로 웜의 공격 이전에 공격에 의하여 피해를 당할 수 있는지 여부를 판단하는 기준이 된다. 이와 연동하여 현재 호스트가 웜에 감염되었는지 여부를 판단하는 기준으로 <<Check me>>를 정의한다. 이 항목에서는 현재 호스트가 웜에 감염되어 있는지를 확인하며, 이미 웜에 감염되어있는 호스트에는 재감염 동작이 수행되지 않으므로 확인 결과에 따라 웜의 감염 행위의 실행 여부를 결정하게 된다. <<Register code>>는 감염된 호스트 상에서 웜이 지속적으로 활동할 수 있도록 시스템 재부팅시 자동으로 실행되는 기능의 추가를 의미한다. 이상의 단계는 웜의 감염에 관련된 정적인 동작이므로 class를 기본 범주로 한다. 실제 활동하는 객체인 취약한 호스트와 웜 코드는 각각 <<Infected Host>>와 <<Malignant>>로 정의한다. 이 단계는 실제 활동의 주체가 되는 요소들이므로 Actor 범주에 포함시킨다. 취약성을 찾고 공격을 하는 웜의 실질적인 행동은 웜의 종류에 따라 다양하게 구분될 수 있다. 공격 단계와 실질적인 코드 전송 단계가 동시에 진행되는지 여부에 따라서 취약성을 찾는 과정, 호스트를 공격하는 과정 및 공격 코드의 전송을 조합할 수 있게 된다. 실제로 Sasser 웜의 경우에는 취약성을 찾는 스캔 과정이 진행된 이후에 공격을 수행하게 되고, Blaster 웜의 경우에는 스캔 활동과 동시에 공격이 수행된다. 또한 Slammer 웜의 경우에는 스캔과 공격, 코드 전송이 동시에 이루어지므로 이러한 웜들을 각각 구분할 수 있는 스테레오 타입의 정의가 필요하게 된다. 따라서 웜의 활동을 표현하는 단계를 Connector로 정의하고 다음과 같이 세부 활동을 정의한다.

스캔 후 공격 : <scan> + <attack>
스캔과 동시 공격 : <scan + attack>
스캔과 동시에 공격 및 웜코드 전송 : <scan + attack + code>
스캔 후 공격 및 웜 코드 전송 : <scan> + <attack + code>

이 상태에서 추가로 웜 코드의 전송과 직접적으로 관련된 스테레오 타입은 다음과 같이 정의 한다.

웜 코드 요청 : <request code>
웜 코드 전송 : <transfer code>

위와 같이 정의한 웜의 활동은 모두 Connector 범주에 포함된다. 이상과 같은 단계를 모두 수행하게 되면 시스템의 상태는 웜에 의하여 감염이 완료된 상태로 표현된다.

다음으로는 웜 코드를 전송하는 Code Transfer에 대한 스테레오 타입 정의가 필요하다. Kienzle의 연구에 의하면 웜 코드의 전송은 취약점을 이용하는 방법, 윈도우 공유 폴더를 이용하는 방법, 메일을 통한 방법 등으로 나눌 수 있다[5]. 이를 근거로 하여 다음과 같은 형식으로 스테레오 타입을 정의한다.

표 5 Code Transfer의 스테레오 타입 정의

Stereotype	Base class	Description
<<Code Transfer>>	Package	코드를 전송하는 단계
<<TargetHostIP>>	Class	IP를 생성하는 방법
<<Scan>>	Class	취약점을 스캔하는 방법
<<Send mail>>	Class	메일을 통해 전파
<<Window File Share>>	Class	파일공유를 통하여 전송
<<Passive>>	Class	다른 웜 침투할 때 그 웜을 삭제하고 자기가 장악
<<Transfer Malignant Code>>	Class	악성코드를 전달하는 방법
<<Malignant Code>>	Actor	악성 코드 (밑에 이름을 붙임)
<<Infected Host>>	Actor	악성코드에 감염된 호스트
<<Vulnerability Host>>	Actor	취약한 호스트
<<Non-Vulnerability Host>>	Actor	취약점이 패치된 호스트

먼저 취약점을 통해서 코드를 전파하는 전통적인 웜의 경우 IP를 생성하고 생성된 IP를 통해 취약점을 가진 호스트를 찾는다. IP를 생성하는 방법은 여러 가지가 있는데, <<TargetHostIP>>는 이러한 아이피를 생성하는 것을 나타낸 스테레오 타입이다. 자체적으로 전파되는 웜은 웜 자체에 IP를 생성하는 기능이 포함되어있는 경우가 많기 때문에, 이러한 구체적인 기능을 분석하여 포함시키게 된다. 그러나 이메일 등을 이용하여 전파되는 웜에서는 감염 호스트 IP에 대한 정보가 필요하지 않기 때문에 이부분이 삭제될 수 있다. IP를 생성한 후 전파할 대상을 찾는 행위가 나타나는데 <<Scan>>은 취약점을 가진 호스트를 찾는 것을 나타내는 스테레오 타입이다. 웜에 따라서 Scan 방식 역시 다양하게 적용될 수 있으므로 웜을 분석한 결과가 그대로 이 항목에 적용되어야 한다. IP를 생성하여 취약점을 통해 전파하는 웜과 달리 전자 메일을 통해서 전파하는 경우 <<Send mail>>을 통해 표현하며, 윈도우 공유폴더의 취약점을 사용하여 전파하는 경우는 <<Window File Share>>으로 스테레오 타입을 표현한다. 메일을 보내서 전파하는 방법과 윈도우 파일 공유폴더를 이용해 전파하는 방법의 경우 사회공학적인 방법으로 시스템의 취약성을 이용하여 전파하는 것이 아닌 컴퓨터를 사용하는 사용자의 행위를 기반으로 웜의 전파에 이용될 수 있는 요소를 이용하여 전파하는 방식이다. <<Passive>>는 다른 웜이 현재 감염되어있는 호스트에 침투해 오면 침투해온 Infected Host에 있는 웜은 삭제하고 자신 복제하여 감염시키는 행위를 표현한다.

이러한 웜들은 <<Passive>> 스테레오 타입으로 표현한다. 이와 같이 웜 전송 방식은 코드를 전송하는 단계에서는 IP를 다양한 방법으로 생성하고 생성된 IP를 가지고 웜을 전송하는 방법, Spam 메일을 통해서 웜을 전송하는 방법, 파일 공유를 통해서 웜을 전송하는 방법, 그리고 다른 웜 침투할 때 그 웜을 삭제하고 자기가 들어서는 방법으로 정의하였다. 그리고 웜 코드는 취약점을 가진 <<Vulnerability Host>>에 전파되며, 취약점이 없는 <<Non-Vulnerability Host>>에는 전파 활동이 수행될 수 는 있으나 실질적으로는 감염이 되지 않는다. 웜의 전파 행위를 표현하는 부분은 웜코드의 명령어에 따른 실행을 의미하기 때문에 웜 코드를 표현하는 스테레오타입은 <<Malignant Code>>이며, 웜코드를 전송해주는 주체는 웜에 이미 감염된 호스트인 <<Infected Host>>이다.

마지막으로, 웜의 전송 후 감염된 호스트에서 공격자가 원하는 행위를 수행하는 단계인 Payload의 스테레오 타입을 정의한다. Payload는 공격자가 웜을 통하여 수행하는 다양한 명령이 포함될 수 있으므로 일정한 규칙을 만들기가 쉽지 않다. 그러므로 현 단계에서는 기존에 분석된 데이터들을 통하여 각 행위들을 일부분 표현하는 방식으로 스테레오 타입을 정의한다.

표 6 Payload의 스테레오 타입 정의

Stereotype	Base class	Description
<<Payload>>	Package	공격자의 명령행위
<<IRC Backdoor>>	Class	IRC 서버에 접속하여 IRC요청에 따라 행동
<<FTP Server>>	Class	FTP 서버의 역할
<<Send Spam>>	Class	스팸 메일 전송
<<IRC Server>>	Actor	IRC-Bot의 경우 IRC 좀비들에게 명령
<<Infected Host>>	Actor	감염된 호스트 봇의 좀비
<<Client>>	Actor	FTP서버 열때 접속하는 호스트
<order>	Connector	공격자의 명령
<send data>	Connector	공격자의 요청하는 데이터 전송
<connection>	Connector	FTP 서버 열때 접속

<<IRC Backdoor>>는 공격자가 웜을 전파한 후에 웜에 감염된 호스트에 IRC Backdoor 설치하여 공격자가 원하는 의도대로 감염된 호스트를 원격 조정한다. 이것은 웜의 변종인 봇으로 감염된 호스트를 좀비라 하며 공격자는 감염된 호스트를 IRC Backdoor를 통해 감염된 좀비를 자신이 원하는 방식으로 제어 할 수 있다. <<FTP Server>>는 감염된 호스트를 FTP 서버로 만들어 클라이언트들이 FTP 서버를 통하여 파일을 업/다운로드 할 수 있는 기능을 제공한다. 특히 일부 웜의 경우에는 FTP 서버를 통하여 웜 코드를 전송하기도 한다. 또한 공격자가 호스트를 불법적인 파일 공유의 용도로 이용하기 위하여 FTP 서버를 설치하기도 한다. <<Send Spam>>은 공격자가 감염된 호스트를 통하여 스팸 광고를 전송하는 활동을 표현한다. 일반적으로 스팸 메일을 전달하는 웜의 경우에 적용할 수 있다. 이상의 정의들은 모두 <<Payload>> Package 하위에 Class로 규정된다. <<IRC Server>>, <<Infected Host>>, <<Client>>는 감염된 시스템의 역할을 표현하는 것이므로 Actor로 규정하였다. 또한 웜의 네트워크 활동과 관련된 <order>, <send data>, <connection>는 Connector로 정의하였다.

4. 결론

본 연구에서는 웜의 활동을 3단계로 분류하여 각 단계에 해당하는 웜의 활동을 세분화 하여 NFR을 이용한 스테레오 타입을 정의하였다. 이러한 정의는 웜을 분석하고 표현하는데 있어서 기본 틀로 제시될 수 있으며, 실질적으로는 시뮬레이션 과정에서 웜을 모델링 할 경우 다양한 웜을 표현하기 위한 기본 구조로 사용될 수 있다.

이러한 정의는 그 표현방식이 정규적으로 규정될 수 있으므로 웜의 행위에 따른 데이터가 표현이 가능하다. 따라서 각 행위에서 발생되는 이벤트들을 부가적으로 표현할 수 있으며 공격 형태에 따른 분류가 가능하므로 웜의 활동을 전체적으로 파악할 수 있다. 또한 필요에 따라서는 각 부분의 통합이나 추가가 용이하므로 새로운 웜이 발견된다 하더라도 융통성 있게 적용할 수 있는 장점이 있다.

본 연구에서는 자체적으로 전파되는 웜만을 연구의 대상으로 하였으나, 다른 매체 혹은 방법을 통하여 전파되는 웜의 경우에는 더욱 많은 요소들에 대한 정의가 필요하다. 특히 최근 들어 확산 빈도가 높은 이메일 웜의 경우에는 전파 기능이 자체 전파되는 웜과 매우 상이하기 때문에 이와 관련된 연구가 추가적으로 진행되어야 한다.

참고문헌

[1] Matthew M. Williamson, Jasmin Leveille "An epidemiological model of virus spread and cleanup",Hewlett-Packard Company, 2003

[2] Z. Chen, L. Gao and K. Kwiat, Modeling the Spread of Active Worms, IEEE INFOCOM 2003

[3] Jeffery K. and Steve W., Directed-graph Epidemiological Models of Computer Viruses, In IEEE Symposimum on Security and Privacy, 1991

[4] Mohammed Hussein and Mohammad Zulkernine "UMLintr: A UML Profile for Specifying Intrusions" 2006 IEEE

[5] Kienzle DM. Elder MC. "Recent Worms: A Survey and Trends". in Proceeding of the 2003 ACM workshop on Rapid Malcode Oct 2003. pp1-10

[6] Nazario J. Anderson J. Wash R and Connelly C. "The Future of Internet Worms" 2001 Blackhat Briefings. LasVegas. NV. July 2001 pp.13-20

[7] Rosa, N., Cunha, P., and Justo, G., "ProcessNFL: A Language for Describing Non-Functional Properties", Proceedings of the 35th Hawaii International Conference on System Science, Hawaii, 2002

[8] 아일라 노이슈타트 "실용적 객체지향 분석과 설계를 위한 UML과 UP", Pearson Education Korea. 2003.

[9] http://www.oracle.com/technology/global/kr/pub/columns/use_case.html