# Teaching How to Write Security Target of Common Criteria Using the i* Methodology

田原 康之[†]，田口 研治[†]，本位田 真一[††]

[†] 国立情報学研究所　[††] 国立情報学研究所/東京大学

セキュリティは，今や情報化社会における最大の問題点の１つとして，広く認識されている．セキュリティに限らず，ソフトウェア欠陥への対策の基本方針としては，可能な限り開発プロセスの上流の段階，特に要求分析工程において，潜在的な欠陥を少なくすべきである，という認識が浸透している．そこで国立情報学研究所では，高度ソフトウェア技術者の育成を行っているトップエスイープロジェクトにおいて，セキュリティ要求分析講座を開講している．特に，セキュリティ要求分析技術の実務における適用への橋渡しとして，セキュリティ要求仕様の文書化の標準である，セキュリティ評価に関する国際標準ISO/IEC15408 (CC)を扱い，評価基準を記した文書であるセキュリティ目標(ST)の作成方法を教育している．本論文では，ゴール指向要求分析手法であるi*の適用によるST作成手法の教育について述べる．

Teaching How to Write Security Target of Common Criteria Using the i* Methodology

Yasuyuki Tahara[†] Kenji Taguchi[†] Shinichi Honiden[††]

[†] National Institute of Informatics [††] National Institute of Informatics/The University of Tokyo

Security is one of the most widespread and common problem in everyday life. Security breach at companies is reported almost everyday and users of computer systems are busy updating security patches against vulnerabilities of their computer systems. These problems are caused by human errors and faults of physical devices but the majority of them are due to the defects of the software systems. The best way to reduce them is to find and fix them in earlier stage of the software development, especially in the requirements elicitation and analysis phases. Thus we incorporate a security requirements analysis course in our project, called TopSE, to teach advanced software engineers. In particular, we treat ISO/IEC15408 (Common Criteria for Information Technology Security Evaluation, CC) and teach a method to write Security Target (ST) that is a type of document used in the security evaluation process of CC. In this paper, we describe how our teach the method based on i* that is a goal-oriented requirements analysis methodology.

## 1 Introduction

Security is one of the most widespread and common problem in everyday life. Security breach at companies is reported almost everyday and users of computer systems are busy updating security patches against vulnerabilities of their computer systems. These problems are caused by human errors and faults of physical devices but the majority of them are due to the defects of the software systems. The best way to reduce them is to find and fix them in earlier stage of the software development, especially in the requirements elicitation and analysis phases. Thus we incorporate a security requirements analysis course in our project, called TopSE, to teach advanced software engineers. In particular, we treat ISO/IEC15408 (Common Criteria for Information Technology Security Evaluation, CC) and teach a method to write Security Target (ST) that is a type of document used in the security evaluation process of CC. In this paper, we describe how our teach the method based on i* that is a goal-oriented requirements analysis methodology.

This paper is organized as follows. Section 2 briefly explains i*. Section 3 explains our method. Section 4 gives some concluding remarks and future work. Appendix A includes the i* models treated in our course.

## 2 i*

i* is a goal-oriented requirements engineering methodology originally proposed by Yu[4]. Its main feature is to handle social aspects of system requirements such as relationships of stakeholders. The details of i* are as follows.

- Before requirements analysis of the system to be implemented, i* analyzes the as-is status, that is, the social environments in which the system is going to be settled.
- The analysis of as-is status results in identification of dependency relationships between actors, that is, i* model constructs representing the stakeholders and the systems that already exist or are going to be developed. Such dependency relationships are used to identify trade-offs of the requirements of the to-be systems.
- i* uses the following two types of models. *Strategic Dependency* (SD) models focus on the organizational aspects of the system environments by consisting of dependency relationships between actors. *Strategic Rationale* (SR) models express the details of the concerns of each actor.

i* models include the two types of goals: hard goals and softgoals. They are distinguished by the definition of satisfaction. A hard goal is totally satisfied or not at all satisfied. A softgoal has a degree of satisfaction represented by a number between 0 and 1. i* models also include two other types of constructs: tasks and resources. A task represents a routine to satisfy goals. A resource is a physical or informational entity used to satisfy goals or carry out tasks. A dependency relationship between actors is involved in a goal, a task, or a resource. i* models also include other various types of relationships such as AND-OR decompositions and contributions representing influence of a constructs to satisfaction of a softgoal. Examples of i* model appear in the latter parts of this paper.

## 3 Application of i* to Common Criteria

Our security requirements analysis course aims at enabling students to apply the advanced requirements analysis techniques to their workplace in practice. This course, in the same way as other several courses of TopSE, treats an international standard ISO/IEC 15408 as a framework of security requirements specifications and a catalog of security functionality requirements specifications. This standard is called "Common Criteria for Information Technology Security Evaluation" and *Common Criteria* or *CC* for short. As the name shows, this standard is provided for evaluation of security of IT systems. However, the objective of evaluation is not the security strength but the "level of confidence that the security functionality and its assurance measure meet the specified requirements" [1) . Therefore the requirements specifications are as important as the system itself for CC evaluation. CC specifies the format of the requirements specifications called *Security Target* or *ST* for short. Our course teaches how to write an ST efficiently using the learned techniques, in particular, an i*-based security requirements analysis approach.

### 3.1 i*-Based Security Requirements Analysis Approach

We describe here the security requirements analysis approach based on i* and risk analysis approaches [3) . In detail, our approach is to combine the following techniques.

- Liu et al [2) proposed a security RE technique based on i* that analyze extensive constructs of security requirements model including vulnerability, threats, attacks, and countermeasures. Our approach is mainly based on this technique (we call Liu's method hereafter).
- HazOp (HAZard and OPerability study) is a widely-known risk analysis technique. Recently it is becoming applied to security risk analysis for software. We found they can be effectively used if we combine it with Liu's method.

We first explain our approach and illustrate the exercise using an example of HD (hard disk)/DVD recorders. Due to the readability, all the figures of i* models are put into Appendix A.

Our approach here is to enhance Liu's method with risk analysis techniques. Liu's method is summarized as follows (Figure 1).

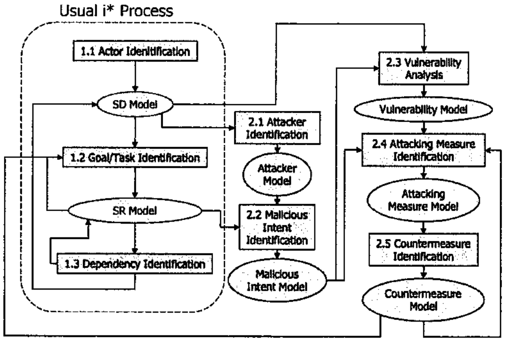
Usual i* Process
1.1 Actor Idenitification
SD Model
1.2 Goal/Task Identification
SR Model
1.3 Dependency Identification
2.1 Attacker Identification
Attacker Model
2.2 Malicious Intent Identification
Malicious Intent Model
2.3 Vulnerability Analysis
Vulnerability Model
2.4 Attacking Measure Identification
Attacking Measure Model
2.5 Countermeasure Identification
Countermeasure Model

Fig. 1 Liu's Method [2)

**Usual i* Process** The i* process included in Liu's method (called *domain requirements analysis process* in the literature[2) ) is divided into the three steps, (1.1) Actor Identification, (1.2) Goal/Task Identification, and (1.3) Dependency Identification. (1.1) and (1.3) transform the SD model and (1.2) transforms the SR model.

**2.1 Attacker Identification** Identify attacker actors. The main idea of this step is to imagine that any actor could behave as an attacker. This does not only mean that an unknown attacker could behave as if he is a legitimate actor by the masquerading attack, but also include the case in which the attacker makes use of intent represented goals, system functionalities or human behaviors represented by tasks, and resources available to the attacker in carrying out the attacking measures.

The resulting model includes the attacker actors and is called an *attacker model.* In this paper, we distinguish model constructs involved in the attacker actors such as malicious intent goals and attacking measure tasks from others. We call a model including such constructs an `attack model`, while we call a model with no such constructs a `domain model`.

**2.2 Malicious Intent Identification** Identify the malicious intent of the attackers as goals of the attacker actors. The resulting model includes such goals and called a *malicious intent model.*

**2.3 Vulnerability Analysis** Identify a part of the SD model that has the possibility of being attacked as vulnerability. The resulting model is called a *vulnerability model.*

**2.4 Attacking Measure Identification** Identify the actions realizing the malicious intent as attacking measure tasks. The resulting model is called an *attacking measure model.* HazOp is applied to this step for systematic identification.

**2.5 Countermeasure Identification** Identify the countermeasures against the attacking measures as tasks. The resulting model is called a *countermeasure model.* Note that a countermeasure model is a domain model as the attacking measure tasks are transformed into softgoals representing the degree of avoidance of the attacks and other constructs about the attacks are removed.

Our course uses an example of copyright protection problem for HD/DVD recorders. Though the copyright issue is not exactly a security problem, it is closely related to various security issues and functionalities. For example, many software products treating texts or multimedia data include access control mechanisms to prevent the data from being copied. Most of DRM (Digital Rights Management) mechanisms use security-related technologies such as cryptography and digital watermarks. For this reason, we treat the copyright problem as a topic of security requirements analysis.

The details of the problem are as follows.

- If a user tries to copy multimedia data recorded in an HD recorder to a DVD recorder, she must not violate the copyright of the TV station for the data.
- The students of this course need to identify functionalities of the HD recorder and the DVD recorder and issues brought by the functionalities.

The taught process is described as follows.

### 3.1.1 Domain Requirements Analysis

The steps here apply the usual i* process and create a model of the environments and the requirements other than security. Though the entire model for HD/DVD recorders would be very large, we handle only a part of it involved in the security RE process. The detailed scenario is as follows.

1. Identify actors. They include not only the user and the TV station easily identified from the problem descriptions, but also the devices such as the HD recorder, the DVD recorder, and the TV set.
2. Identify the goals. We identify a hard goal "Copy Contents" of the user and a softgoal "Protect Copyright" of the TV station here.
3. Analyze the "Copy Contents" goal as follows.
    - Delegate the "Copy Contents" goal to the DVD recorder, that is, identify the dependency of the user to the recorder about the goal, because the recorder is expected to achieve the goal completely.
    - Divide the goal of the DVD recorder to the two subgoals "Receive Contents" and "Record Contents to DVD".
    - Identify the goal "View Contents Later" of the user as the supergoal of the "Copy Contents" goal.
    - Proceed analyses for each goal.

As the results of the analyses, we can create the domain model of Figure 5.

### 3.1.2 Attacker and Malicious Intent Identification

We show the case of treating the user as an attacker as an example. The attacker identification and malicious intent identification are carried out as follows.

1. Identify the attacker actor "Attacker" and a malicious goal "Violate Copyright" as an intent to attack.
2. Change the type of the "User" actor to "Role" and create a "Play" association between them (Figure 6). The purpose of these operations is to indicate the user is considered as an attacker.
3. Create an actor "User As Attacker" by putting "User" and "Attacker" together and put their goals into the new actor to clarify the situation in which "Attacker" "Play"s the "User" role and proceed the analyses. As the results, the model shown in Figure 7 is created.

### 3.1.3 Vulnerability Identification

This step identifies the range of influence of the identified malicious goals, their subgoals, and their subtasks. For our example, the "Protect Copyright" softgoal delegated from the TV station to the user is identified as vulnerable.

### 3.1.4 Attacking Measure Identification

This step identifies attacking measures as the tasks to achieve the malicious goals. It is allowed to use (or *misuse* in this case) the domain model constructs to carry out the attacking measures. As the original Liu's method does not provide specific techniques to identify the measures, our course teaches a technique using HazOp shown in 3.1.5. For example, the attacking measure model shown in Figure 8 is created.

### 3.1.5 Application of HazOp

HazOp is a risk analysis technique whose main feature is to help analysts identify potential risks in systems. This technique is becoming used in security analysis recently. Our course applies it to increase the practical applicability of Liu's method.

HazOp is based on a typical risk analysis procedure in which (1) the targets of the analysis are identified first, (2) then the potential risks are identified, and (3) finally the risk reducing measures are identified. HazOp mainly focus on the step (2). Our security requirements analysis approach shown before lacks the detailed techniques for the task corresponding to this step. Thus we consider the combination of these approaches is very useful.

The main feature of HazOp focused on the step (2) above is application of guide words. A guide word is "a short word to create the imagination of a deviation of the design/process intent" [3] . A deviation and a design/process intent correspond to a malicious intent goal and a domain goal of i* models respectively. Thus HazOp is used to identify malicious intent goals and attacking measures. Table 1 shows a typical set of guidewords and their usage in our approach.

Table 1 Guidewords and Their Usage

| | |
|---|---|
| No | Do not carry out a task or satisfy a goal |
| More | Carry out a task or satisfy a goal too much by repetition, concurrent execution, or using extreme values |
| Less | Carry out a task or satisfy a goal too little |
| As well as | Carry out an additional task or satisfy an additional goal |
| Part of | Carry out a task or satisfy a goal only partially |
| Other than | Carry out a task to satisfy an inappropriate goal or satisfy a goal to satisfy an inappropriate supergoal |

Table 2 shows an example of application of HazOp to our example. Thus the initially identified malicious intent goals are obtained by application of HazOp.

Table 2 Example of Application of HazOp

| Domain goal | Guideword | Malicious intent goal |
|---|---|---|
| Copy Contents | More | Copy Contents Many Times |
| | Other than | Copy Copy-Protected Contents |
| Record Contents | Other than | Record Record-Protected Contents |

## 3.2 CC and ST

We explain the details of CC and how our approach is used to create an ST document efficiently.

### 3.2.1 Organization of CC

CC consists of the following three parts.

**Part 1: Introduction and general model** describes the backgrounds and the fundamental ideas of security evaluation and specifies the general model of assets, countermeasures and evaluation. In particular, the format of ST is specified here. In addition, another type of document called *Protection Profile* or *PP* for short is also specified. A PP is a subset of ST and is used as a reusable module for writing STs.

**Part 2: Security functional components** is a catalog of security functionalities. The functionalities are divided into eleven classes such as security audit, communication, cryptographic support, and user data protection. It is recommended to choose security functionalities from here when writing an ST or a PP.

**Part 3: Security assurance components** is a catalog of inspection tasks to make sure if the security functionalities are correctly implemented. Such tasks are divided into ten classes. This part also specifies *Evaluation Assurance Levels* or *EALs* for short that is a set of assurance requirements. There are seven levels of EALs (from EAL1 to EAL7). These levels do not represent the security strength but the ranges of targets of inspection and the degrees of inspection.

### 3.2.2 ST Organization

In CC, an ST is defined as an implementation-dependent statement of security needs for a specific identified TOE (Target Of Evaluation). The organization of ST is as follows.

1. ST introduction: describing the TOE in a narrative way on three levels of abstraction: ST and TOE references, TOE overview and TOE description

2. Conformance claims: describing how the ST conforms with the Common Criteria itself, Protection Profiles (if any), and Packages

3. Security problem definition: defining the security problem that is to be addressed

4. Security objectives: a concise and abstract statement of the intended solution to the problem defined by the security problem definition
5. Extended components definition: defining components that are not based on those in CC Part 2 or CC Part 3.
6. Security requirements: consists of security functional requirements (SFRs) and security assurance requirements (SARs)

   SFRs consist of a translation of the security objectives for the TOE into a standardised language.

   SARs consist of a description of how assurance is to be gained that the TOE meets the SFRs
7. TOE summary specification: describing the general technical mechanisms that the TOE uses to satisfy all the SFRs

In creating an ST using security requirements analysis approaches, we use the correspondence between an ST and a security requirements model shown in Figure 2. As this table shows, the most part of an ST can be written using security requirements analysis approaches.

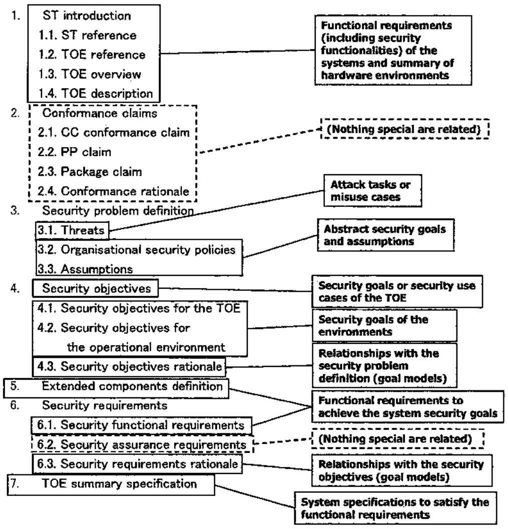
1.
ST introduction
1.1. ST reference
1.2. TOE reference
1.3. TOE overview
1.4. TOE description
2.
Conformance claims
2.1. CC conformance claim
2.2. PP claim
2.3. Package claim
2.4. Conformance rationale
3.
Security problem definition
3.1. Threats
3.2. Organisational security policies
3.3. Assumptions
4.
Security objectives
4.1. Security objectives for the TOE
4.2. Security objectives for the operational environment
4.3. Security objectives rationale
5.
Extended components definition
6.
Security requirements
6.1. Security functional requirements
6.2. Security assurance requirements
6.3. Security requirements rationale
7.
TOE summary specification
Functional requirements (including security functionalities) of the systems and summary of hardware environments
(Nothing special are related)
Attack tasks or misuse cases
Abstract security goals and assumptions
Security goals or security use cases of the TOE
Security goals of the environments
Relationships with the security problem definition (goal models)
Functional requirements to achieve the system security goals
(Nothing special are related)
Relationships with the security objectives (goal models)
System specifications to satisfy the functional requirements

Fig. 2 Correspondence between ST and Security Requirements Model

To apply Liu's method to writing an ST, we establish the mapping from the models created using Liu's method according to the relationships between i* and ST. The mapping consists of the following constructs.

- Relationships between security requirements analysis in general and ST and the following
- i* goals and tasks to various requirements
- In writing "3.1. Threats",
  - i* actors (attackers) to threat agents
  - i* resources (to be protected) to assets
  - i* tasks (of attacking) to adverse actions
- In writing Rationales (4.3., 6.3.), i* goal models are mapped to table representations of relationships between requirements items.

We need to note in writing an ST that goals or tasks corresponding to the following sections should be refined to the level of CC Part 2: 5. Extended components definition, and 6.1. Security functional requirements.

As an exercise of writing an ST, we use the following example of document management system.

- Users can access the documents
- The extent of users is specified for each document
- Each user can access the documents permitted to the user
- TOE is the document management system

  Other entities such as the users are considered as the environments

We assume that we create an attack model shown in Figure 3 using Liu's method.

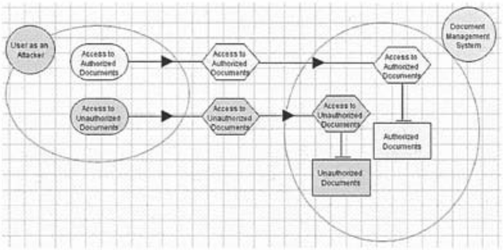
User as an Attacker
Access to Authorized Documents
Access to Unauthorized Documents
Access to Authorized Documents
Access to Unauthorized Documents
Document Management System
Access to Authorized Documents
Access to Unauthorized Documents
Authorized Documents
Unauthorized Documents

Fig. 3 Attack Model of Document Management System

From this model, we can write "3. Security problem definition" of the ST as follows.

3.1. Threats

- T.UNAUTHORIZED_ACCESS_TO_DOCUMENTS

  Attacker may access documents with no access permissions

Note that a prefix "T." is attached to each item of this section. We have other similar prefixes.

3.2. Organisational security policies: None

3.3. Assumptions: None

After that, we assume that we create a domain requirements model including security countermeasures shown in Figure 4. The symbols such as *FMT_MSA_dot_3* represents SFRs adopted from CC Part 2. Though this SFR is denoted as FMT_MSA.3 in CC, we use _dot_ to represent a dot "." because ST-Tool cannot handle a dot.

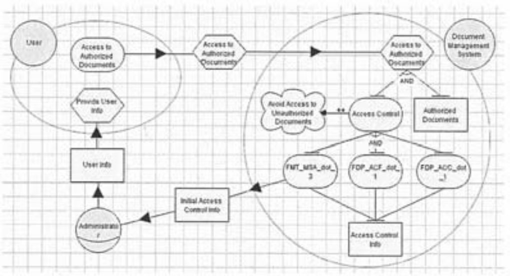
User
Access to Authorized Documents
Access to Authorized Documents
Access to Authorized Documents
Document Management System
AND
Provide User Info
Avoid Access to Unauthorized Documents
Access Control
Authorized Documents
AND
User Info
FMT_MSA_dot_3
FDP_ACF_dot_1
FDP_ACC_dot_1
Initial Access Control Info
Administrator
Access Control Info

Fig. 4 Domain Requirements Model of Document Management System

From this model, we can write the following corresponding part of the ST.

4.1. Security objectives for the TOE

- O.ACCESS_CONTROL

  The extent of users is specified for each document

  Each user can access the documents permitted to the user

4.3. Security objectives rationale

| | T.UNAUTHORIZED _ACCESS_TO _DOCUMENTS |
|---|---|
| O.ACCESS_CONTROL | x |

6.1. Security functional requirements

- FDP_ACC.2 Complete access control

  FDP_ACC.2.1: The TSF shall enforce the *ACL SFP (or any other access policies)* on *the users and the documents* and all operations among subjects and objects covered by the SFP.

  – The emphasized words are concrete assignments to the original CC descriptions.

  – We also have FDP_ACC.2.2.

- FDP_ACF.1 Security attribute based access control

- FMT_MSA.3 Static attribute initialisation

6.3. Security requirements rationale

| | O.ACCESS_CONTROL |
|---|---|
| FDP_ACC.2 | x |
| FDP_ACF.1 | x |
| FMT_MSA.3 | x |

The students also work on a larger exersize using an example of cellular phones with contactless IC chips for the electronic money facility.

## 4 Conclusions

In this paper, we described how our teach the method to write Security Target (ST) of CC based on i*. We have several issues in writing CC documents using security requirements analysis approaches. The first one is treatment of PPs (protection profiles). As a PP is created for somewhat general purpose in an application domain, writing the PP demands domain analysis. We also need to investigate how to use PPs efficiently in writing an ST. The second issue is how to analyze SARs (security assurance requirements). Because an SAR is not a requirement to a system but to security assurance activities, we need some software process approaches. Finally, considering that CC documents are formal ones and are targets of official inspection, improvement of their quality is important. Tool support would be useful to that purpose.

## Acknowledgement

This work is fully sponsored by the Special Coordination Funds for Promoting Science and Technology, Fostering talent in emergent research field by MEXT.

## 参考文献

1) Common criteria for information technology security evaluation part 1: Introduction and general model. http://www.commoncriteriaportal.org/public/developer/index.php?menu=2, Sep. 2006.

2) L. Liu, E. Yu, and J. Mylopoulos. Security and privacy requirements analysis within a social setting. In *Proc. of RE'03*, pages 151–161, 2003.

3) M. Rausand and A. Høyland. *System Reliability Theory; Models, Statistical Methods and Applications (Second Edition)*. Wiley, 2004.

4) E. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *Proc. of RE'97*, pages 226–235, 1997.

## A i* Models

The figures appear from the next page.

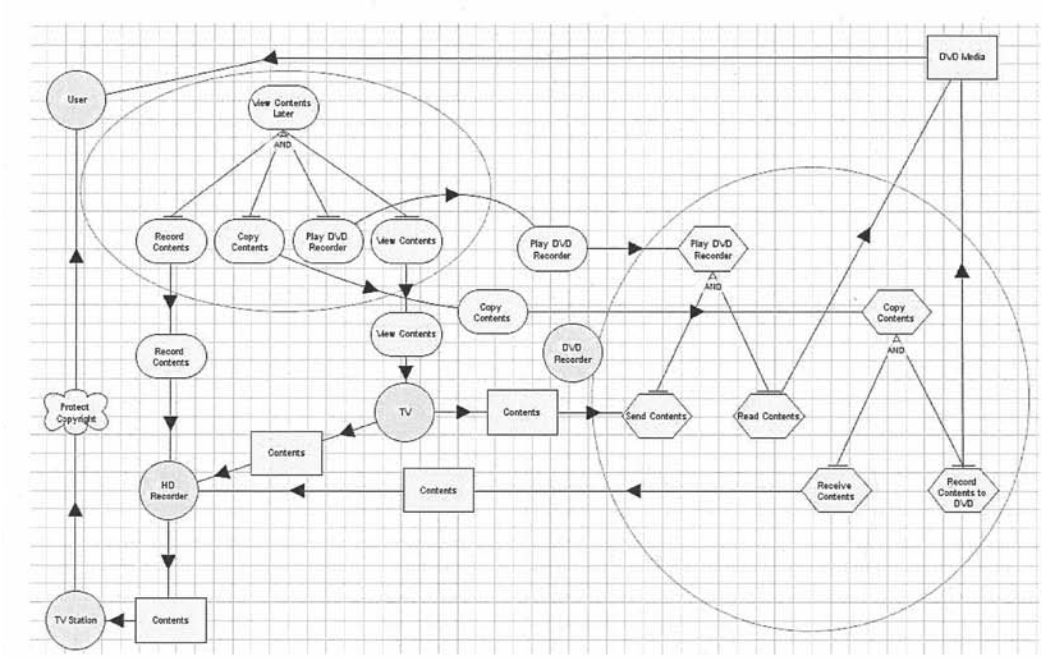

Fig. 5 Domain Model

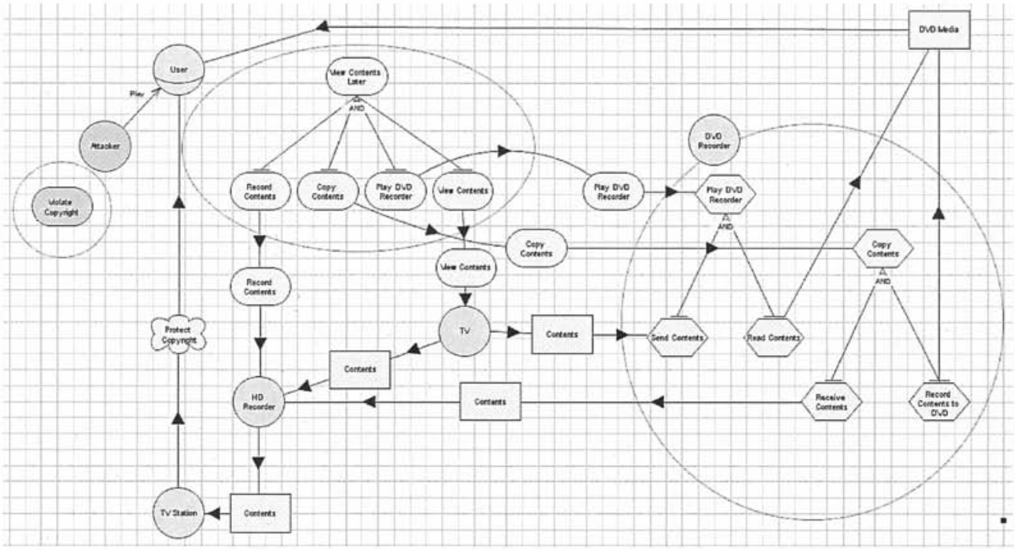

Fig. 6 Introduction of Attacker Actor

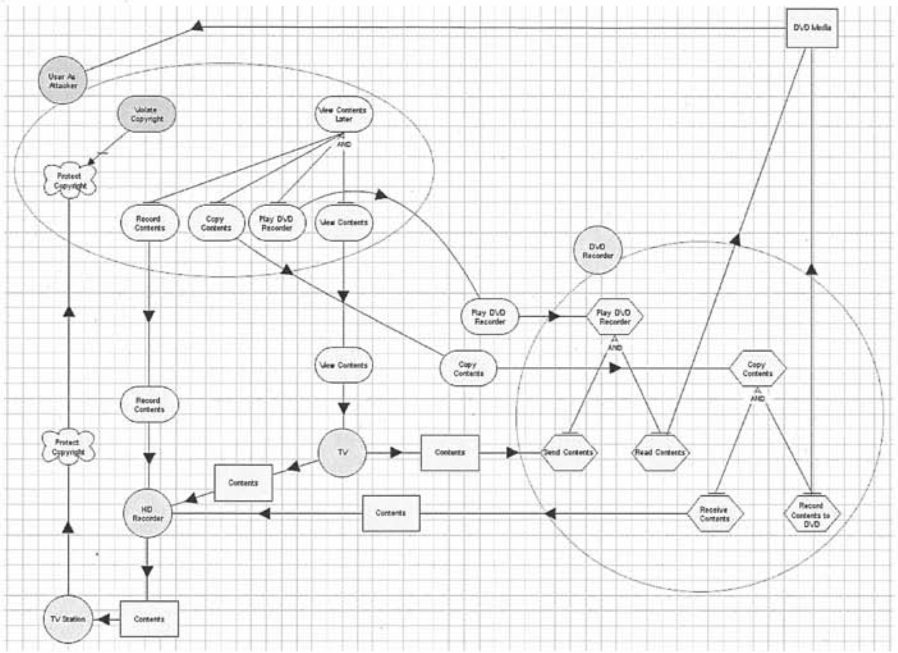

Fig. 7 Attacker Model

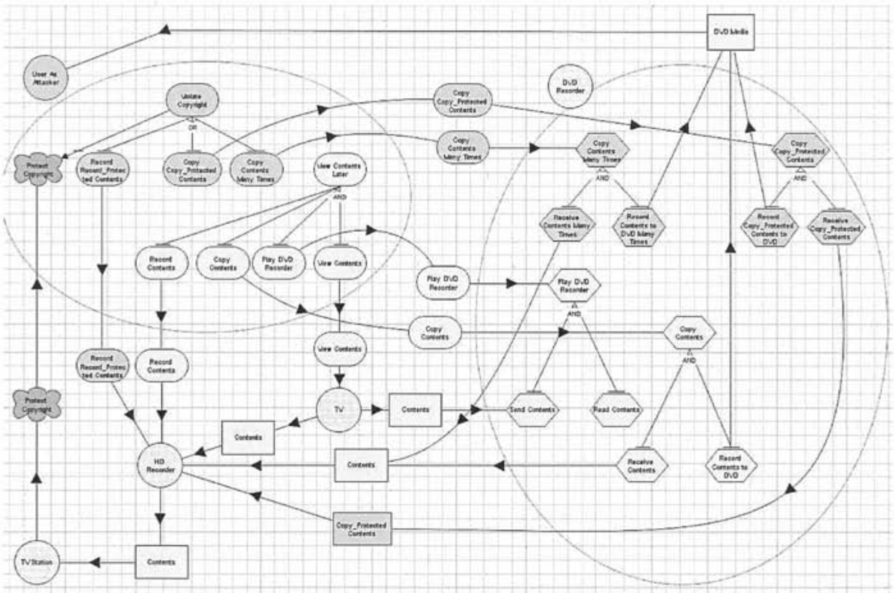

Fig. 8 Attacking Measure Model