STF40 A03066 – Unrestricted

REPORT

UML profile for security assessment

Mass Soldal Lund, Ida Hogganvik,
Fredrik Seehusen, Ketil Stølen

SINTEF Telecom and Informatics

December 2003

SINTEF REPORT

SINTEF Telecom and Informatics

Address: NO-7465 Trondheim
NORWAY
Location Trondheim:
S.P. Andersens v 15
Location Oslo:
Forskningsveien 1
Telephone: +47 73 59 30 00
Fax: +47 73 59 43 02

Enterprise No.: NO 948 007 029 MVA

TITLE

UML profile for security assessment

AUTHOR(S)

Mass Soldal Lund, Ida Hogganvik, Fredrik Seehusen, Ketil Stølen

CLIENT(S)

Research council of Norway

REPORT NO. STF40 A03066	CLASSIFICATION Unrestricted	CLIENTS REF. 152839/220		
CLASS. THIS PAGE Unrestricted	ISBN 82-14-03109-5	PROJECT NO. 40332800, 40309900		NO. OF PAGES/APPENDICES 25/0
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.) Ketil Stølen	CHECKED BY (NAME, SIGN.) Jan Øyvind Aagedal	
FILE CODE	DATE 2003-11-13	APPROVED BY (NAME, POSITION, SIGN.) Bjørn Skjellaug, Research director		

ABSTRACT

This report defines a UML 2.0 profile for security assessment. The profile introduces a metamodel that defines an abstract language for supporting model-based risk assessment. Further the profile provides a mapping of classes in the metamodel to UML modelling elements by defining so-called stereotypes, and introduces special symbols (icons) for representing the stereotypes in UML diagrams.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	ICT	IKT
GROUP 2	Risk management	Risikoadministrasjon
SELECTED BY AUTHOR	Modelling	Modellering
	UML	UML
	Security assessment	Sikkerhetsanalyse

TABLE OF CONTENTS

1 Introduction

This report defines a UML 2.0 profile for security assessment. The profile introduces a metamodel that defines an abstract language for supporting model-based risk assessment. Further the profile provides a mapping of classes in the metamodel to UML modelling elements by defining so-called stereotypes, and introduces special symbols (icons) for representing the stereotypes in UML diagrams.

The motivation for the profile is the practical use of UML to support security management in general, and security assessment in particular.

The background of the profile is the EU-project CORAS (IST-2000-25031).[1] CORAS has developed a framework for model-based security assessment. This framework includes:

- A methodology for model-based risk assessment integrating aspects from partly complementary risk assessment methods and state-of-the-art modelling methodology.
- A UML based specification language targeting security risk assessment.
- A library of reusable experience packages.
- A computerised platform providing two repositories; an assessment repository and a repository for the reusable experience packages.
- An XML mark-up for exchange of risk assessment data.
- A component for computerised vulnerability management.

In the model-based risk assessment methodology of CORAS, UML models are used for three different purposes:

- *To describe the target of evaluation at the right level of abstraction.* A proper assessment of technical system documentation is not sufficient; a clear understanding of system usage and its role in the surrounding organisation or enterprise is just as important. UML allows these various aspects to be documented in a uniform manner.
- *To facilitate communication and interaction between different groups of stakeholders involved in a security assessment.* One major challenge when performing a risk assessment is to establish a common understanding of the target of evaluation, threats, vulnerabilities and security risks among the stakeholders participating in the assessment. This motivates a UML profile aiming to facilitate improved communication during security assessments, by making the UML diagrams easier to understand for non-experts, and at the same time preserving the well-definedness of UML.
- *To document security assessment results and the assumptions on which these results depend to support reuse and maintenance.* Security assessments are costly and time consuming and should not be initiated from scratch each time we assess a new or modified system. Documenting assessments using UML supports reuse of assessment documentation, both for systems that undergo maintenance and for new systems, if similar systems have been assessed before.

The UML Profile supports all these objectives, but with a special emphasis on communication and documentation. Documentation is supported because the metamodel of the profile is consistent with data structure of security assessment results in CORAS. Communication is supported by the definition of easy-to-understand icons (symbols) associated with the modelling elements of the profile.

[1] For more information on the CORAS project we refer to the project homepage: http://coras.sourceforge.net/

The reminder of this report is structured as follows: Section 1.1 explains the context and preconditions of the work with the profile and Section 1.2 provides an overview of the history of the profile. In Section 2 the metamodel of the profile is presented and in Section 3 the profile itself is defined. Section 4 introduces the icons (symbols) associated with the various modelling elements of the profile and Section 5 provides examples of the use of the profile. Section 6 provides conclusions. Section 7 contains the references of this report.

1.1 Context and preconditions

UML is the most widely used specification language in the software industry today. A UML profile is a refinement of the basic UML language targeting a more specialised application area. The profile is based on UML version 2.0 [7], which is recently adopted by as a standard of the Object Management Group (OMG). With UML 2.0 as the new standard, the older version of UML now in use (versions 1.3, 1.4 and 1.5) will be outdated, and therefore not suited as a basis for the UML profile.

An important source of inspiration in the development of this profile is the work of Sindre and Opdahl on “misuse cases” [9][10]. A presentation of misuse cases is also found in [1].

The UML profile for security assessment was developed within the 5th Framework EU project CORAS (IST-2000-25031). The CORAS consortium consists of eleven partners from four countries: CTI (Greece), FORTH (Greece), IFE (Norway), Intracom (Greece), NCT (Norway), NR (Norway), QMUL (UK), RAL (UK), SINTEF (Norway), Solinet (Germany) and Telenor (Norway). Telenor and SINTEF are responsible for the administrative and scientific coordination, respectively. The results reported in this paper have benefited from the joint efforts of the CORAS consortium.

1.2 History

The profile has developed over time, and the development of the profile has been iterative with many sources of feedback. One of these sources has been development of the CORAS methodology itself, but other sources of feedback have been equally important. This section provides an overview of the most important ones.

1.2.1 Publications

A preliminary version of the profile was presented in the paper “Towards a UML profile for model-based risk assessment” [3]. Further the book chapter “The CORAS methodology: Model-based risk assessment using UML and UP” [2] provided an introduction in the use of this preliminary version, and in the paper “Maintaining results from security assessments” [4] a brief presentation of the profile was given. In addition, the profile has been presented at a number of seminars and workshops.

1.2.2 Standardisation

The UML profile for security assessment was submitted to the OMG as part of a response to the Request for Proposal (RFP) “UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms” [5]. The proposal [6] was adopted as an OMG standard by the OMG technical meeting in London in November 2003, and has now been sent to finalisation.

The UML profile for security assessment is also adopted as a part of a UML profile developed for the Norwegian Defence.

1.2.3 Use of the profile

The profile has been applied in two large security assessment trials, one targeting a telemedicine system and one targeting an advanced internet application integrated in a telecommunication system. Also the profile has been applied in formalising results from the CORAS trials in the efforts to produce reusable security assessment elements.

The profile has been used in teaching security assessment to Master students through courses at the University of Oslo, and in supervision of Master and PhD students.

2 Metamodel

The metamodel of the profile is divided into six submodels (Figure 1) that support different stages of the risk management process. A security assessment always starts with identifying the context of the assessment. A strengths, weaknesses, opportunities and threats (SWOT) analysis may be part of this. After the context has been established, the reminder of a risk assessment can be divided into the identification and documentation of unwanted incidents, risks and treatments.

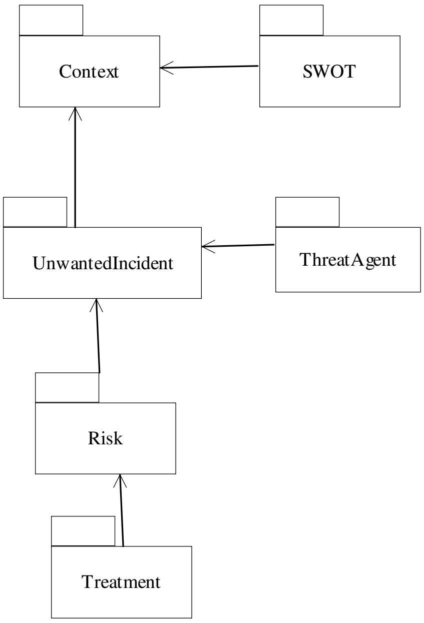


Figure 1 Submodels of the security assessment metamodel

The unwanted incident model is concerned with organising and documenting the threats and vulnerabilities that open for incidents that may harm the system. The risk model quantifies unwanted incidents with respect to the reductions of asset value that they may cause. The treatment model supports documenting ways of treating the system and quantifying the effect of treatments with respect to reducing the potential harm of risks.

In addition there is a submodel TreatAgent associated with the unwanted incident model. This model defines a number of well-known treat agents.

2.1 Context

This submodel defines the context of a security assessment. The context consists of the stakeholders and assets of the system under assessment, which all further assessment is based on. The model is shown in Figure 2.

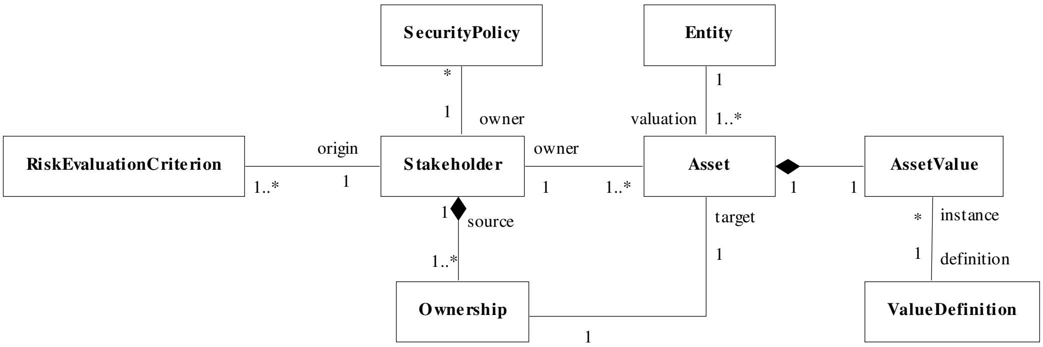


Figure 2 Context submodel

A security assessment is *asset-driven*, which means that assessment is carried out relative to the identified assets. In the general case, an asset may be anything that stakeholders of the system under assessment find to have value.

Each asset may only be related to one stakeholder and should have an unambiguous value assigned by one stakeholder. If two stakeholders view the same entity as an asset, the entity should be documented as two different assets related to the same entity. Two assets are per definition different if valued by different stakeholders. Both the values and the reasons for the valuing may be different.

Below the concepts of the models are described:

- *Stakeholder.* A person or organisation who has interests in the assessed system.
- *SecurityPolicy.* A rule or regulation defined by a stakeholder, related to the security of the system under assessment. A security policy could relate to security aspects like confidentiality, integrity, availability, non-repudiation, accountability, authenticity and reliability, and should provide directions for the assessment.
- *RiskEvaluationCriterion.* A criterion that identified risks are evaluated against in order to decide whether the risk is acceptable or not.
- *Asset.* A part or feature of the system that has value for one of the stakeholders, for example the quality level of a service.
- *Entity.* A physical or abstract part or feature of the system under assessment that becomes an asset when assigned value by a stakeholder, for example a service provided by the system.
- *AssetValue.* The value assigned to an asset by a stakeholder.
- *ValueDefinition.* Definition of value types for various values used in a security assessment, such as asset value.

2.2 SWOT

Strengths, weaknesses, opportunities and threats (SWOT) analysis is as a part of establishing the context of a security assessment. However, SWOT is used for pointing out general directions of the assessment, and its results are only indirectly used in the further assessment. For this reason the concepts of the submodel for SWOT, shown in Figure 3, are not strongly connected to the rest of the metamodel.

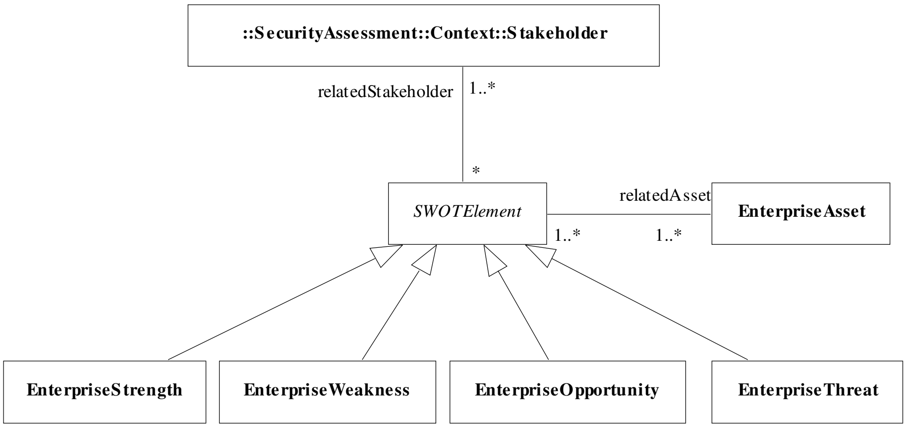


Figure 3 SWOT submodel

A SWOT analysis is concerned with identifying the strategic context of the organisation carrying out a security assessment. The elements of the SWOT model are described below:

- *EnterpriseAsset.* Asset of the organisation from a strategic point of view.
- *EnterpriseStrength.* A strategic strength of the organisation.
- *EnterpriseWeakness.* A strategic weakness of the organisation.
- *EnterpriseOpportunity.* A strategic opportunity of the organisation.
- *EnterpriseThreat.* Something that threatens the strategic position of the organisation.

2.3 Unwanted incident

Identification and documentation of unwanted incidents is concerned with exploring the threats and vulnerabilities of the system under assessment, and how threats and vulnerabilities may combine and lead to potential incidents that may harm the system.
In the metamodel of Figure 4, the notion of threat is decomposed into a threat agent that initiates one or more threat scenarios.

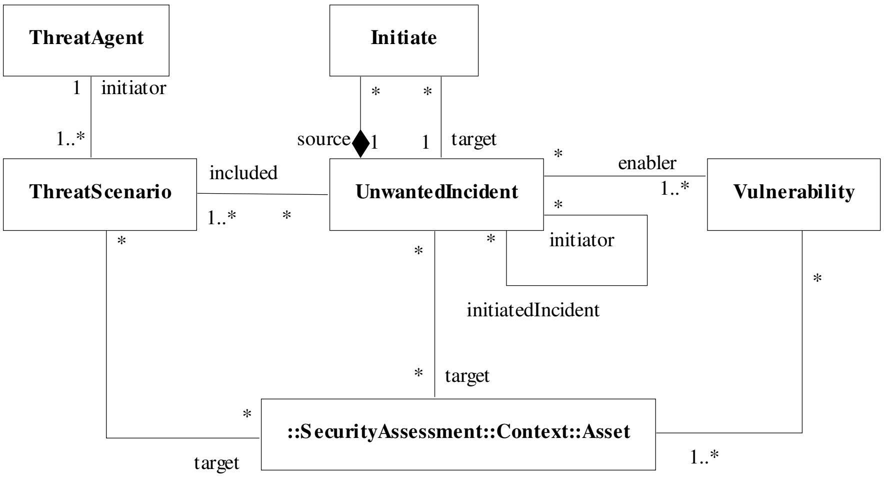


Figure 4 Unwanted incident submodel

The concepts of Figure 4 are described below:

- *ThreatAgent.* A potential cause of an unwanted incident, which may result in harm to a system or organisation and its assets. Threat agents can be external, (e.g., hackers or viruses) or internal (e.g., system failures or disloyal employees).
- *ThreatScenario.* A description of how a threat may lead to an unwanted incident.

- *Vulnerability*. A weakness with respect to an asset or group of assets that can be exploited by one or more threats.
- *UnwantedIncident*. An undesired event that may reduce the value of an asset.
- *Initiate*. An unwanted incident may lead to another unwanted incident. Initiate is a relation for modelling that an unwanted incident acts as an initiator of another unwanted incident.

2.4 Threat agent

The treat agent submodel is presented in Figure 5. This model defines a number of specialisations of the concept of treat agent. The purpose of this model is to support modelling of the variety of threat agent known from literature.

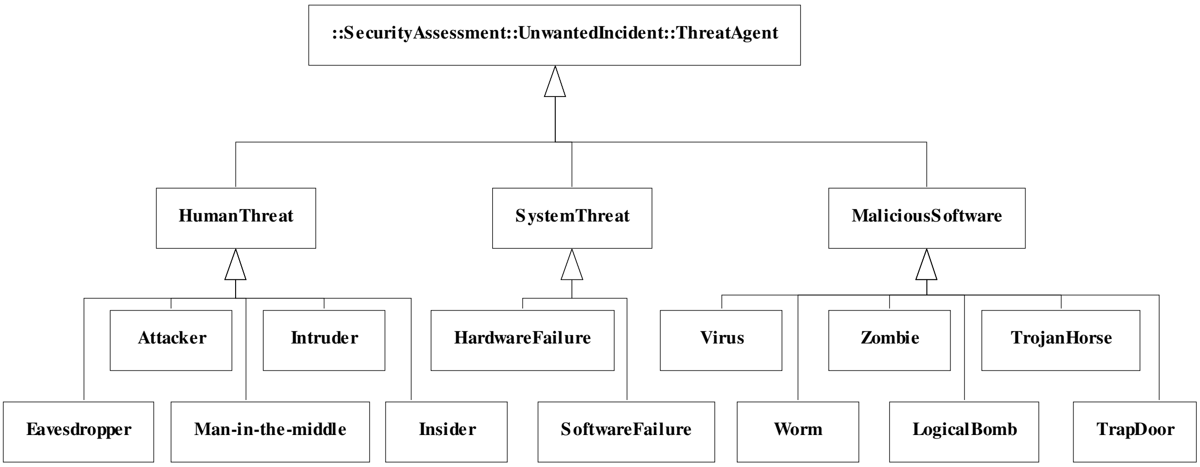


Figure 5 Treat agent submodel

- *HumanThreat*. A person that, with or without intent, potentially may act in a way that leads to unwanted incidents.
  - *Attacker*. An intelligent threat agent that carries out an assault on system security, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system. [8]
  - *Intruder*. An entity that gains or attempts to gain access to a system or system resource without having authorisation to do so. [8]
  - *Eavesdropper*. A person that does a passive wiretapping done secretly, i.e., without the knowledge of the originator or the intended recipients of the communication. [8]
  - *Man-in-the-middle*. A form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data in order to masquerade as one or more of the entities involved in a communication association. [8]
  - *Insider*. An entity inside the security perimeter, i.e., an entity that is authorised to access system resources but uses them in a way not approved by those who granted the authorisation. [8]
- *SystemThreat*. A part of the system under assessment that potentially may act in a way that leads to unwanted incidents.
  - *HardwareFailure*. An error in hardware that may constitute a threat.
  - *SoftwareFailure*. An error in software that may constitute a threat.
- *MaliciousSoftware*. Software made with the intent of harming computerised systems.
  - *Virus*. A hidden, self-replicating section of computer software, usually malicious logic, that propagates by infecting – i.e., inserting a copy of itself into and becoming part of – another program. [8]

- *Worm.* A computer program that can run independently, can propagate a complete working version of itself onto other hosts on a network, and may consume computer resources destructively. [8]
- *Zombie.* A program that secretly takes over another Internet-attached computer and the uses that computer to launch attacks that are difficult to trace to the zombie's creator. [11]
- *TrojanHorse.* A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorisations of a system entity that invokes the program. [8]
- *LogicalBomb.* Malicious logic that activates when specified conditions are met. Usually intended to cause denial of service or otherwise damage system resources. [8]
- *TrapDoor.* A hidden computer flaw known to an intruder, or a hidden computer mechanism (usually software) installed by an intruder, who can activate the trap door to gain access to the computer without being blocked by security services or mechanisms. [8]

2.5 Risk

Figure 6 shows the risk submodel. A risk is an unwanted incident that has been assigned consequence and frequency values. These values are used for calculating a risk value, which represent loss of asset value of the asset the risk is related to. Risks that in some way are related or similar may be categorised into risk themes. A risk theme is itself assigned a risk value based on the risks it contains and is treated like a singular risk with respect to evaluation and treatment. Risk values are evaluated by risk evaluation criteria defined in the context of the security assessment. A risk evaluation criterion states which risk values are acceptable, and which are not – implying the need for treatment.

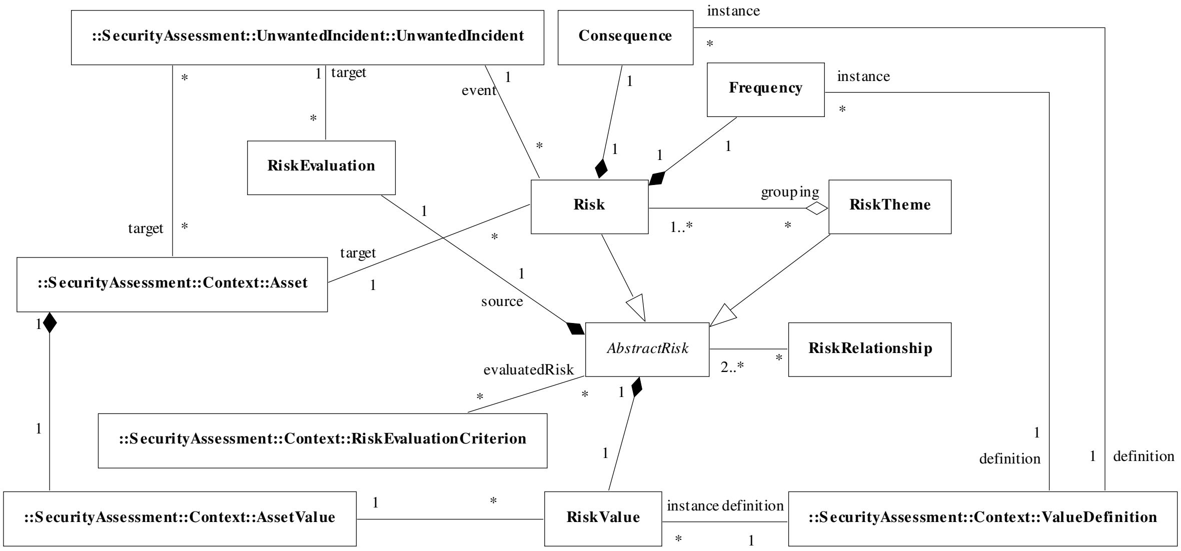

::SecurityAssessment::UnwantedIncident::UnwantedIncident
Consequence
Frequency
RiskEvaluation
Risk
RiskTheme
::SecurityAssessment::Context::Asset
AbstractRisk
RiskRelationship
::SecurityAssessment::Context::RiskEvaluationCriterion
::SecurityAssessment::Context::AssetValue
RiskValue
::SecurityAssessment::Context::ValueDefinition
instance
instance
target
event
grouping
target
target
source
evaluatedRisk
definition
definition
instance definition
1..*
2..*

Figure 6 Risk submodel

The concepts of the submodel of Figure 6 are described in the following:

- *AbstractRisk.* The common properties of risks and risk themes, such as risk value.
- *Risk.* An unwanted incident that has been assigned a consequence value, a frequency value, and a resulting risk value. Threats, vulnerabilities and unwanted incidents may go to several assets, but since a risk may reduce the value of an asset, a risk is only related to one particular asset
- *RiskTheme.* A categorisation of similar risks, assigned its own risk value.
- *RiskRelationship.* The relation between risks or risk themes.

- *RiskEvaluation*. The assignment of a risk or a risk theme to the unwanted incident it evaluates with respect to risk value.
- *Consequence*. The consequence of an unwanted incident happening, relative to an asset.
- *Frequency*. The probability of an unwanted incident occurring within a given period of time.
- *RiskValue*. A value assigned to a risk, reflecting the loss of asset value that the risk represents.

2.6 Treatment

The treatment model (Figure 7) is concerned with documenting and evaluating ways of providing treatments to the system under assessment in order to reduce the value of risks. A treatment may apply to several unwanted incidents. However, when a treatment's capability to reduce risk value is assessed, this is with respect to a single risk or risk theme.

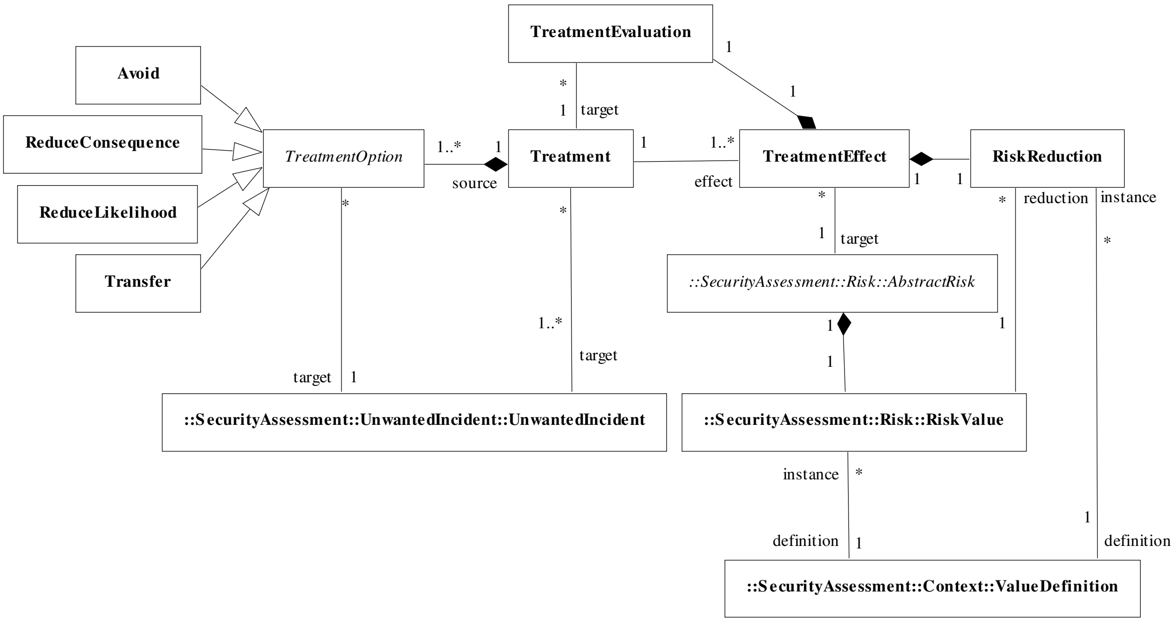


Figure 7 Treatment submodel

The concepts of Figure 7 are described below:

- *Treatment*. Ways of reducing the risk value of a risk or risk theme.
- *TreatmentEffect*. A treatment's capability to reduce the risk value of a particular risk.
- *TreatmentEvaluation*. The assignment of a treatment effect to the treatment it evaluates.
- *RiskReduction*. The value of a treatment effect, i.e., the concrete reduction of a value of a risk.
- *TreatmentOption*. Main classes of providing treatment; used for assigning a treatment an unwanted incident. The options are *Avoid*, *ReduceConsequence*, *ReduceLikelihood*, and *Transfer* (of risk).

3 Profile

In this section the UML profile for security assessment is presented. The profile provides an extension to the UML metamodel, introducing modelling elements for the concepts defined in the security assessment metamodel by defining stereotypes.

Not all classes of the metamodel are assigned stereotypes, more specifically SecurityPolicy, ValueDefinition, AssetValue, RiskEvaluationCriterion, Consequence, Frequency, RiskValue and RiskReduction. In other words, we let modelling of these concepts be undefined. The reason for this is that these classes represent constraints and values and do not need any explicit graphical

notation. Further, there many ways in which these values and constraints may be modelled, and it seems reasonable to leave this up to the users of the profile. In Section 3.7 we do, however, show how stereotypes from the UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms [6] may be used for modelling these concepts.
The structure of the profile, shown in Figure 8, reflects the structure of the metamodel. In the following subsections, these subprofiles are defined.

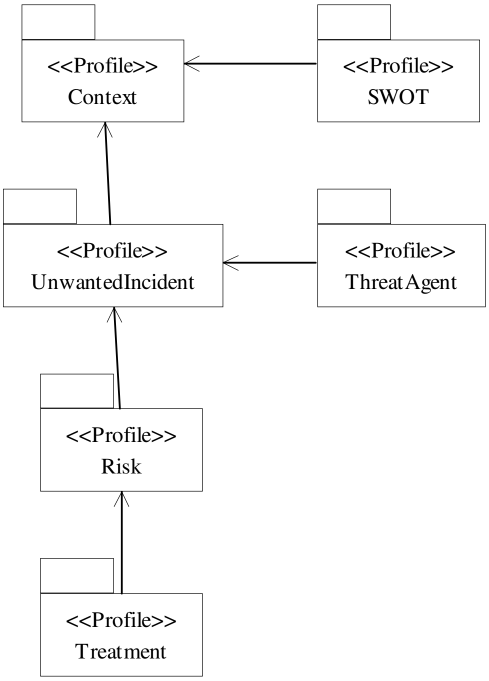


Figure 8 Subprofiles of the security assessment profile

3.1 Context

The subprofile for the context of security assessments is shown in Figure 9.

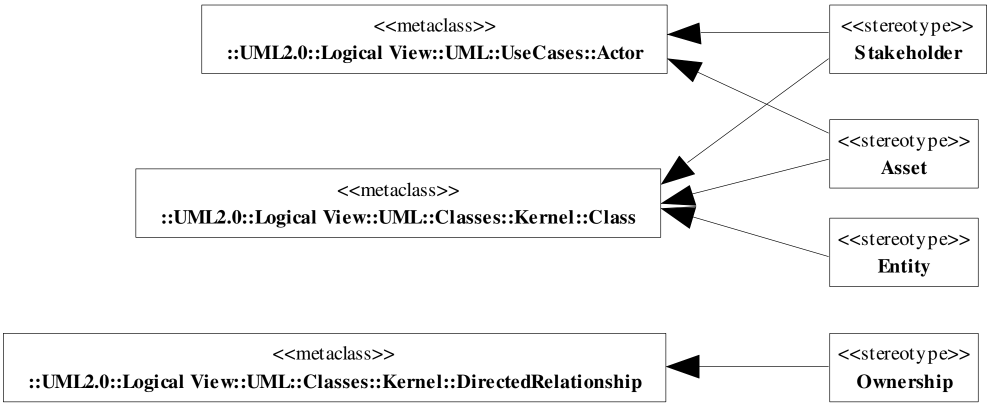

<<metaclass>>
::UML2.0::Logical View::UML::UseCases::Actor
<<metaclass>>
::UML2.0::Logical View::UML::Classes::Kernel::Class
<<metaclass>>
::UML2.0::Logical View::UML::Classes::Kernel::DirectedRelationship
<<stereotype>>
Stakeholder
<<stereotype>>
Asset
<<stereotype>>
Entity
<<stereotype>>
Ownership

Figure 9 Context subprofile

As can be seen from the figure, Stakeholder and Asset may be modelled as both Class and Actor. Documenting assets, stakeholders, and their relationships is most appropriately done in a class diagram, and hence assets and stakeholders are modelled as Class. However, when documenting threats and unwanted incidents in use case diagrams (see Section 2.3) assets and stakeholders should be modelled as Actor. The Ownership relation is modelled using DirectedRelationship.

3.2 SWOT

As seen in Figure 10, SWOTElement is modelled as UseCase and EnterpriseAsset as Actor.

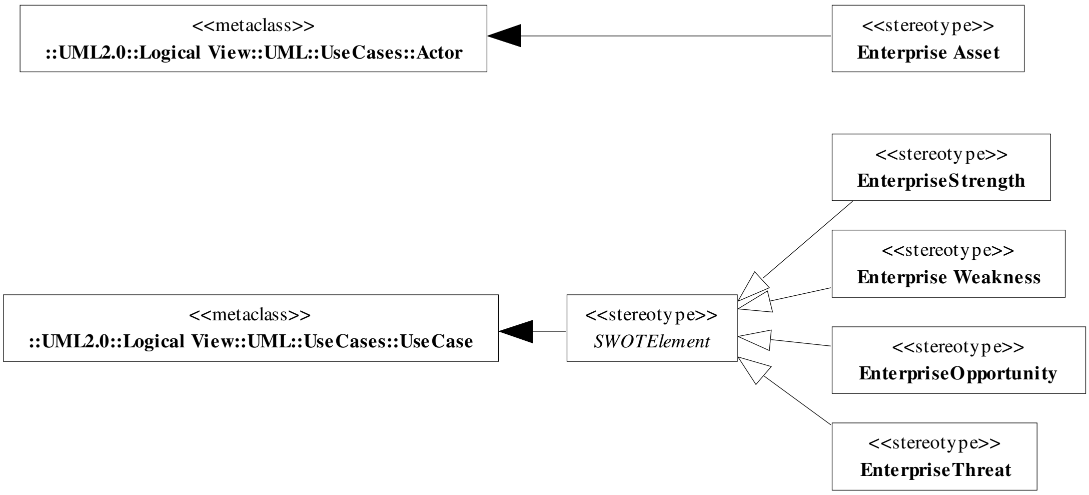


Figure 10 SWOT subprofile

3.3 Unwanted incident

The subprofile for unwanted incidents is shown in Figure 11. A threat is decomposed into a ThreatAgent and a ThreatScenario. As the acting part of a threat, ThreatAgent is modelled as Actor, and ThreatScenario, as the behavioural aspects of the threat, as UseCase. UnwantedIncident, which also may be seen as behaviour, is also modelled as UseCase. Initiate is represented by DirectedRelationship. Vulnerabilities may be seen as (unwanted) features of the assets they apply to, and are modelled as Feature.

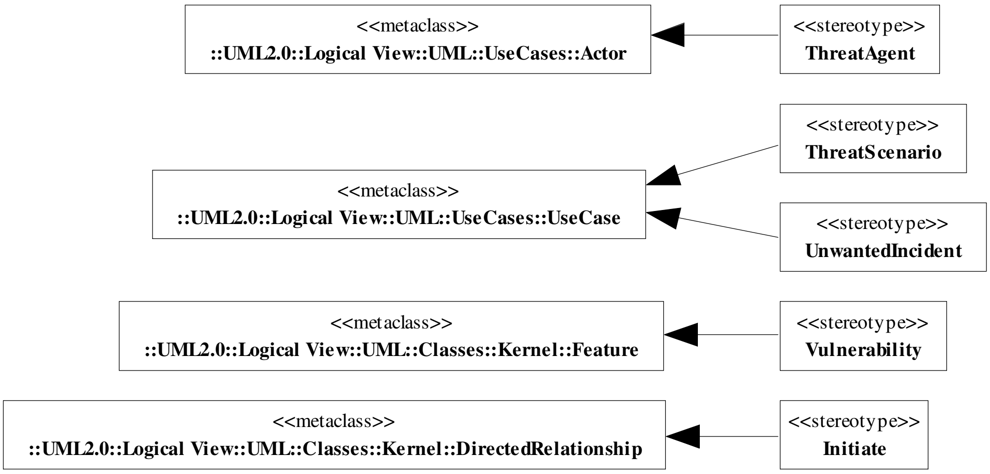

<<metaclass>>
::UML2.0::Logical View::UML::UseCases::Actor
<<stereotype>>
ThreatAgent
<<stereotype>>
ThreatScenario
<<metaclass>>
::UML2.0::Logical View::UML::UseCases::UseCase
<<stereotype>>
UnwantedIncident
<<metaclass>>
::UML2.0::Logical View::UML::Classes::Kernel::Feature
<<stereotype>>
Vulnerability
<<metaclass>>
::UML2.0::Logical View::UML::Classes::Kernel::DirectedRelationship
<<stereotype>>
Initiate

Figure 11 Unwanted incident subprofile

3.4 Threat agent

The submodel for threat agents is shown in Figure 12. As seen in the diagram, all threat agents are specialisations of ThreatAgent, and hence modelled as Actor.

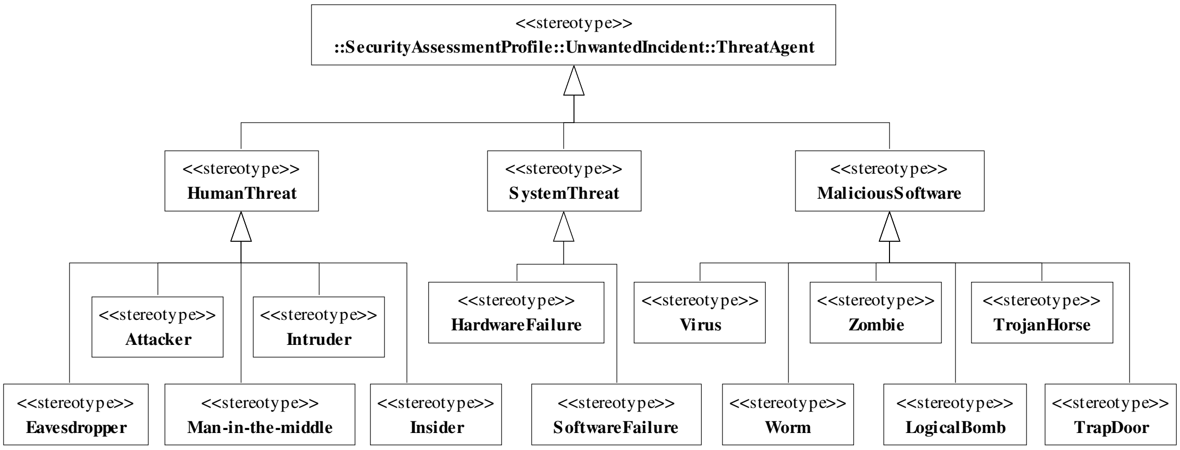


Figure 12 Threat agent submodel

3.5 Risk

The subprofile for risks is shown in Figure 13. Both Risk and RiskTheme are modelled as Class. This makes it possible to capture arbitrary grouping of Risks into RiskThemes by making risks parts in a risk theme. RiskRelationship is modelled as Association, allowing risk themes, with their relations, to be documented in class diagrams. RiskEvaluation assigns a risk to an unwanted incident, and is modelled as DirectedRelationship.

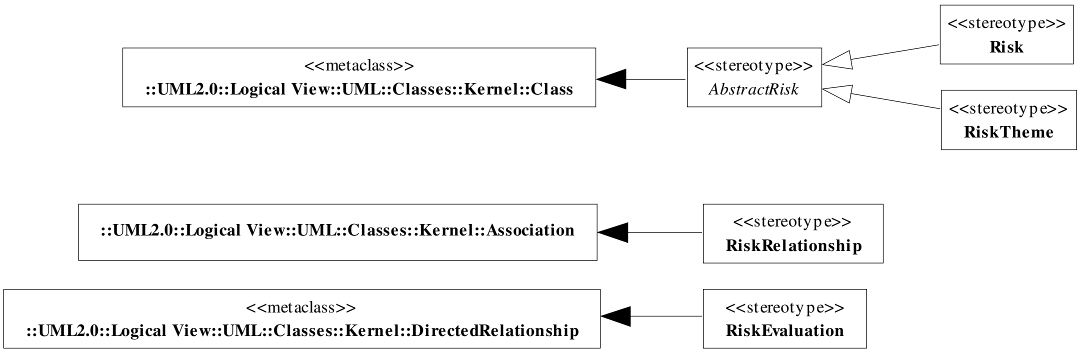

<<metaclass>>
::UML2.0::Logical View::UML::Classes::Kernel::Class
<<stereotype>>
AbstractRisk
<<stereotype>>
Risk
<<stereotype>>
RiskTheme
::UML2.0::Logical View::UML::Classes::Kernel::Association
<<stereotype>>
RiskRelationship
<<metaclass>>
::UML2.0::Logical View::UML::Classes::Kernel::DirectedRelationship
<<stereotype>>
RiskEvaluation

Figure 13 Risk subprofile

3.6 Treatment

The treatment subprofile is shown in Figure 14. Treatment protects against risks, and is modelled as a UseCase. TreatmentEffect is modelled using Class and TreatmentEvaluation using DirectedRelationship. TreatmentOption relates treatments to risks and is modelled using DirectedRelationship.

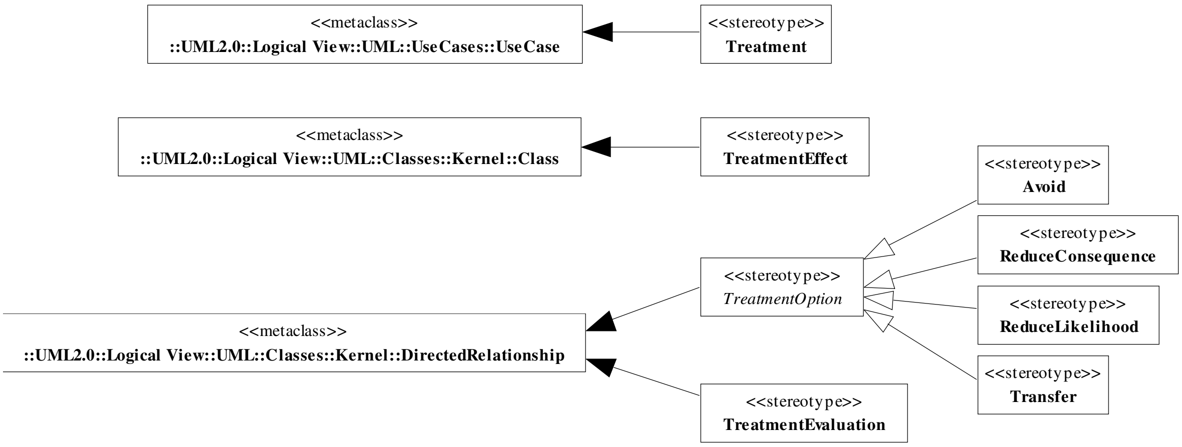


Figure 14 Treatment subprofile

3.7 Using stereotypes from the UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms

As explained above, the classes SecurityPolicy, ValueDefinition, AssetValue, RiskEvaluationCriterion, Consequence, Frequency, RiskValue and RiskReduction from the metamodel are not defined as stereotypes in the profile. In this section we show how stereotypes from the UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms [6] may be used for modelling these classes.

In this approach ValueDefinition is modelled as QoSCharacteristic, since this stereotype may be used for defining value types. AssetValue, Frequency, Consequence, RiskValue and RiskReduction are all modelled as QoSValue, which instantiate QoSCharacteristic. SecurityPolicy is modelled by the means of QoSCharacteristic, which also has the expressiveness to capture various security aspects like availability and integrity. RiskEvaluationCriterion is modelled as QoSRequired, which is used for specifying quality of service requirements.
With this approach we get an extension to the UML Profile for security assessment as shown in Figure 15.

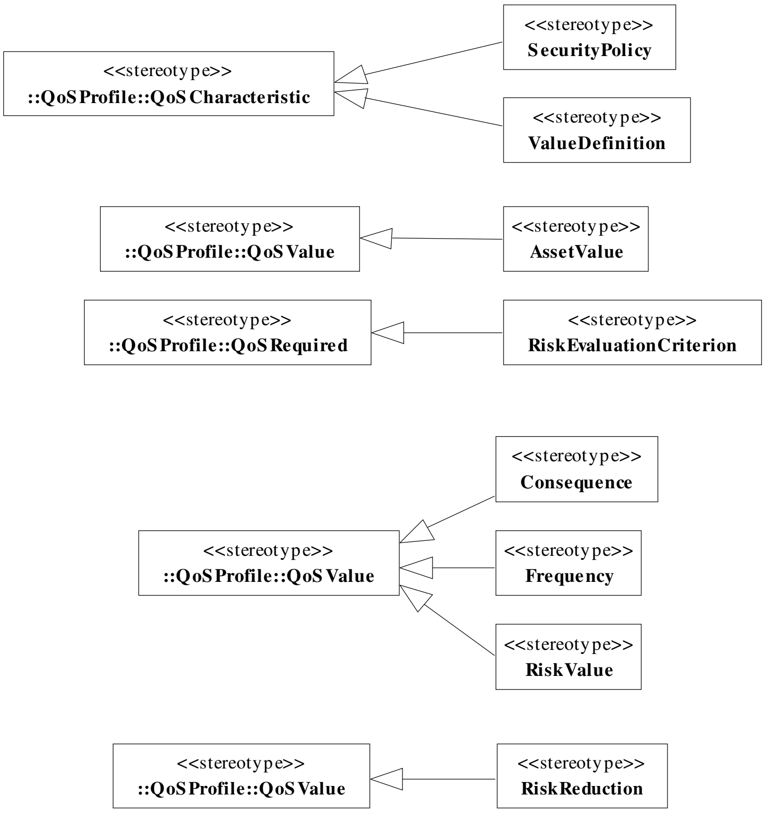


Figure 15 Use of modelling elements from the QoS profile

4 Icons

In order to facilitate communication between various stakeholders and other participants of a security assessment, the stereotypes in the UML Profile are assigned icons, for representing the modelling elements when presented in diagrams. Participants in a security assessment will not in general be familiar with UML, and an important requirement is therefore that these symbols should be intuitively understandable. In the following subsections, icons representing the stereotypes of the UML profile are presented.

4.1 Context

Table 1 presents icons for the context subprofile.

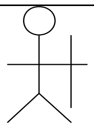
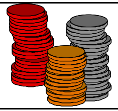
<<Stakeholder>>	
<<Asset>>	
<<Entity>>	No special icon
<<Ownership>>	No special icon

Table 1 Icons for the context subprofile

4.2 SWOT

Table 2 presents icons for the SWOT subprofile.

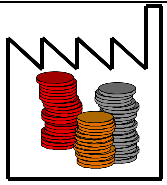
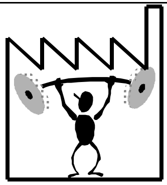
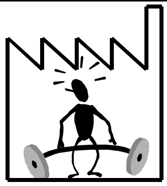
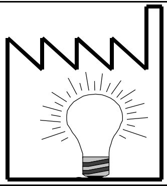
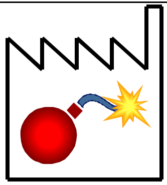
<<EnterpriseAsset>>	
<<EnterpriseStrength>>	
<<EnterpriseWeakness>>	
<<EnterpriseOpportunity>>	
<<EnterpriseThreat>>	

Table 2 Icons for the SWOT subprofile

4.3 Unwanted incident

Table 3 presents icons for the unwanted incident subprofile.

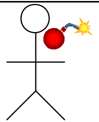
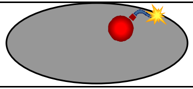
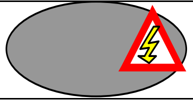
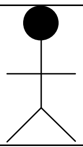
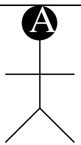
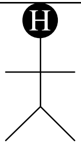
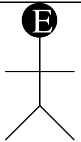
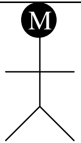
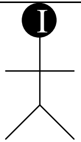
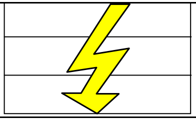
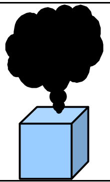
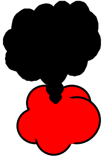
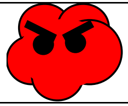
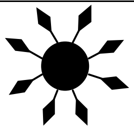
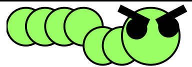
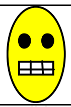
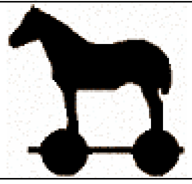
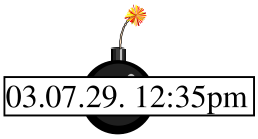
<<ThreatAgent>>	
<<ThreatScenario>>	
<<Vulnerability>>	No special icon
<<UnwantedIncident>>	
<<Initiate>>	No special icon

Table 3 Icons for the unwanted incident subprofile

4.4 Threat agent

Table 4 presents icons for the threat agent subprofile.

<<HumanThreat>>	

<<Attacker>>	A
<<Intruder>>	H
<<Eavesdropper>>	E
<<Man-in-the-middle>>	M
<<Insider>>	 I
<<SystemThreat>>	
<<HardwareFailure>>	
<<SoftwareFailure>>	
<<MaliciousSoftware>>	
<<Virus>>	
<<Worm>>	
<<Zombie>>	
<<TrojanHorse>>	
<<LogicalBomb>>	03.07.29. 12:35pm

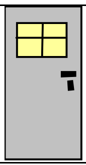
<<TrapDoor>>	

Table 4 Icons for the threat agent subprofile

4.5 Risk

Table 5 presents icons for the risk subprofile.

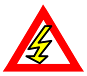
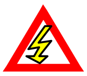
<<Risk>>	
<<RiskTheme>>	
<<RiskThemeRelationship>>	No special icon
<<RiskEvaluation>>	No special icon

Table 5 Icons for the risk subprofile

4.6 Treatment

Table 6 presents icons for the treatment subprofile.

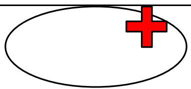
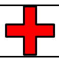
<<Treatment>>	
<<TreatmentEffect>>	
<<Avoid>>	No special icon
<<ReduceConsequence>>	No special icon
<<ReduceLikelihood>>	No special icon
<<Transfer>>	No special icon
<<TreatmentEvaluation>>	No special icon

Table 6 Icons for the treatment subprofile

5 Examples

In the following subsections we present some examples of the use of the UML profile for security assessment. The presentation is structured according to the subprofiles. In the examples all stereotypes are tagged with their stereotype names – also the stereotypes with icons. It is however not necessary to use the stereotype name when stereotypes are represented by icons. In the examples we use stereotypes from the UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms as explained in Section 3.7.

5.1 Context

Figure 16 shows how the stereotype <<ValueDefinition>> is used for defining the value types used throughout a security assessment. In this case all values are enumerations, i.e., values on an ordinal scale, except for "RiskReductionRef" which defines a mapping. An alternative could have been to define asset values and consequences as monetary values and frequencies as probabilities.

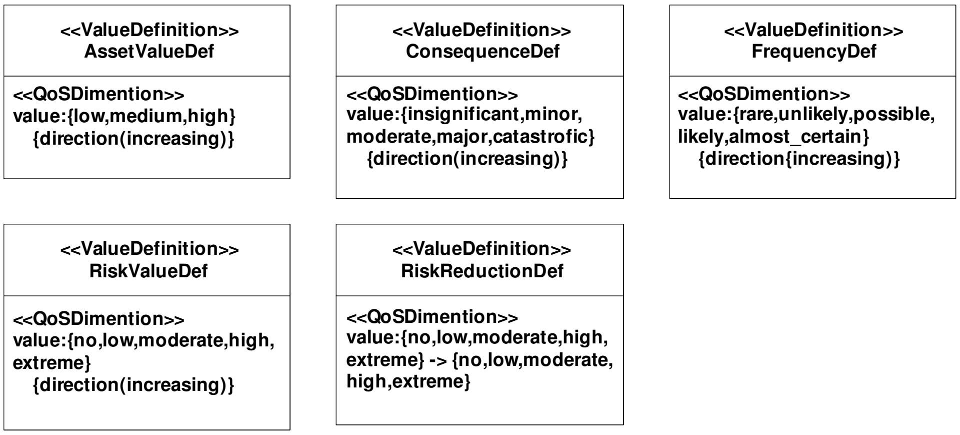


Figure 16 Value definitions

Figure 17 shows definition of an asset. The entity is a service. The asset is defined as the availability of the service. The asset is owned by the stakeholder “Service provider”, and its value is assigned by instantiating the value definition for asset values. Further the diagram shows that asset has one vulnerability.

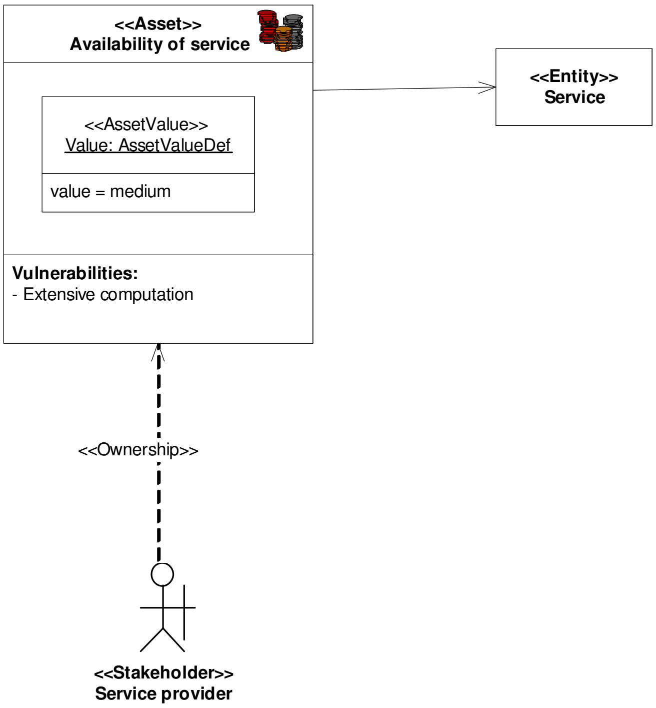


Figure 17 Modelling of assets

5.2 SWOT

Figure 18 shows how the results of a SWOT (Strengths, Weaknesses, Opportunities and Threats) analysis may be modelled using the profile. It may be worth noticing that a SWOT is usually carried out as part of the context identification, and before asset identification.

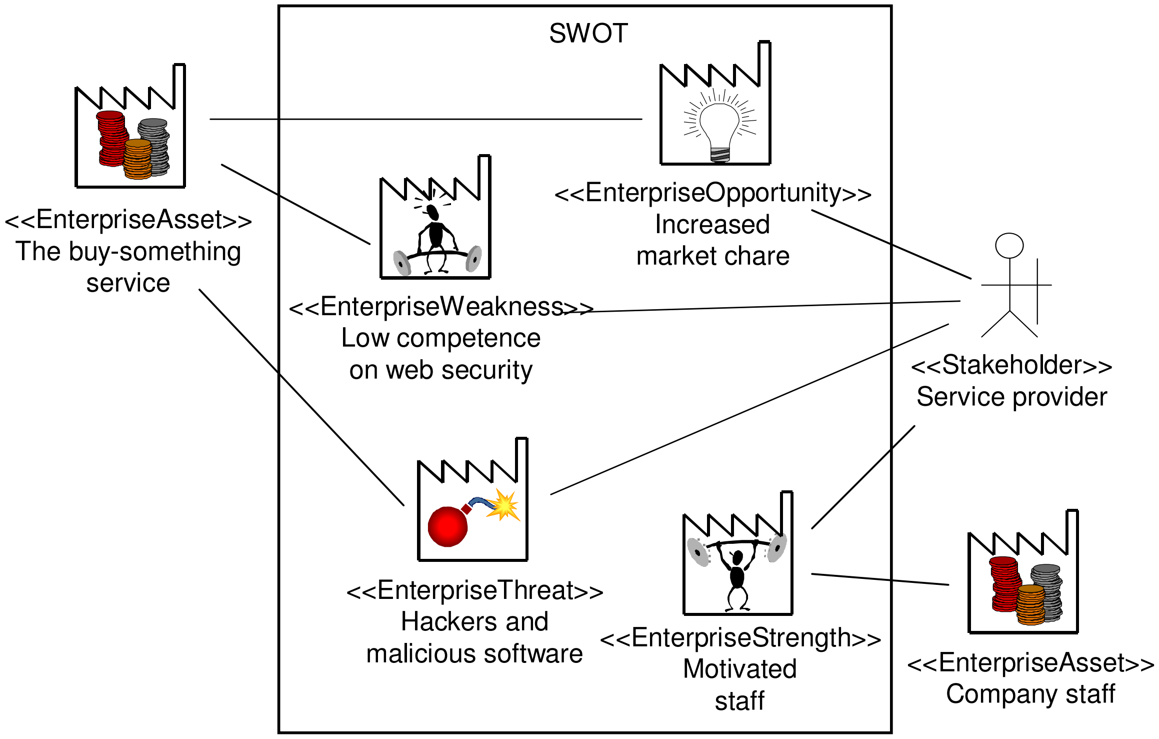


Figure 18 Modelling of SWOT

5.3 Unwanted incident

In Figure 19, modelling of a threat is exemplified. The threat consists of the threat agent, i.e., the acting entity, “Malicious person” and the scenario, i.e., the behaviour, “Flooding”. The threat is related to the asset “Availability of service”. “Malicious person” is modelled as the threat agent specialisation <<Attacker>>.

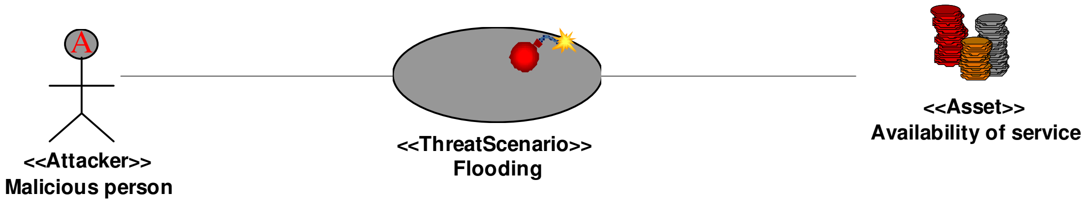


Figure 19 Modelling of threats

Figure 20 illustrates how unwanted incidents are modelled. The unwanted incident “Denial-of-Service” relates to the asset “Availability of service”, and includes the threat scenario from the diagram above. An unwanted incident may lead to other unwanted incidents, and this is shown by use of the stereotype <<Initiate>>. In this case, “Denial-of-Service” initiates the unwanted incident “Loss of customer” which relates to the asset “Customers”.

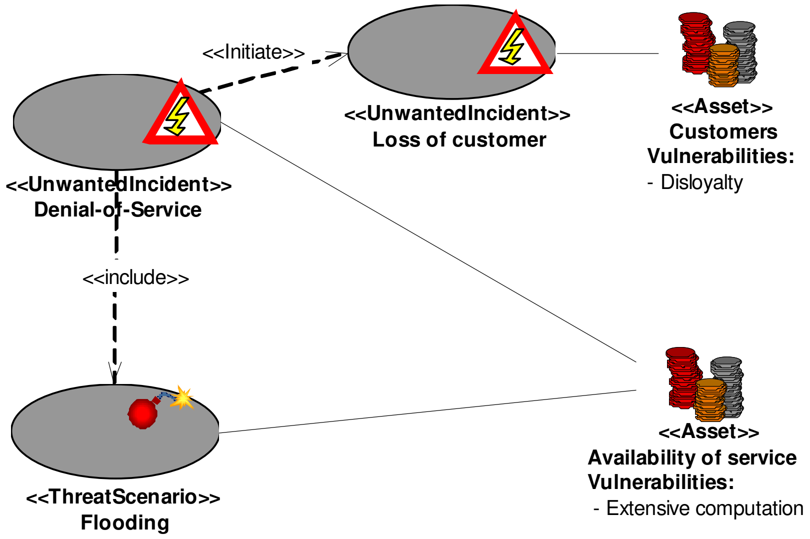


Figure 20 Modelling of unwanted incidents

5.4 Risk

A risk is an assignment of consequence, frequency and risk values to an unwanted incident. Figure 21 illustrates how this is modelled. The values are instances of the corresponding value definitions. The risk "Denial-of-service evaluation" is assigned to the unwanted incident "Denial-of-Service" by the use of the stereotype <<RiskEvaluation>>. The diagram also shows that the risk is related to the asset "Availability of service".

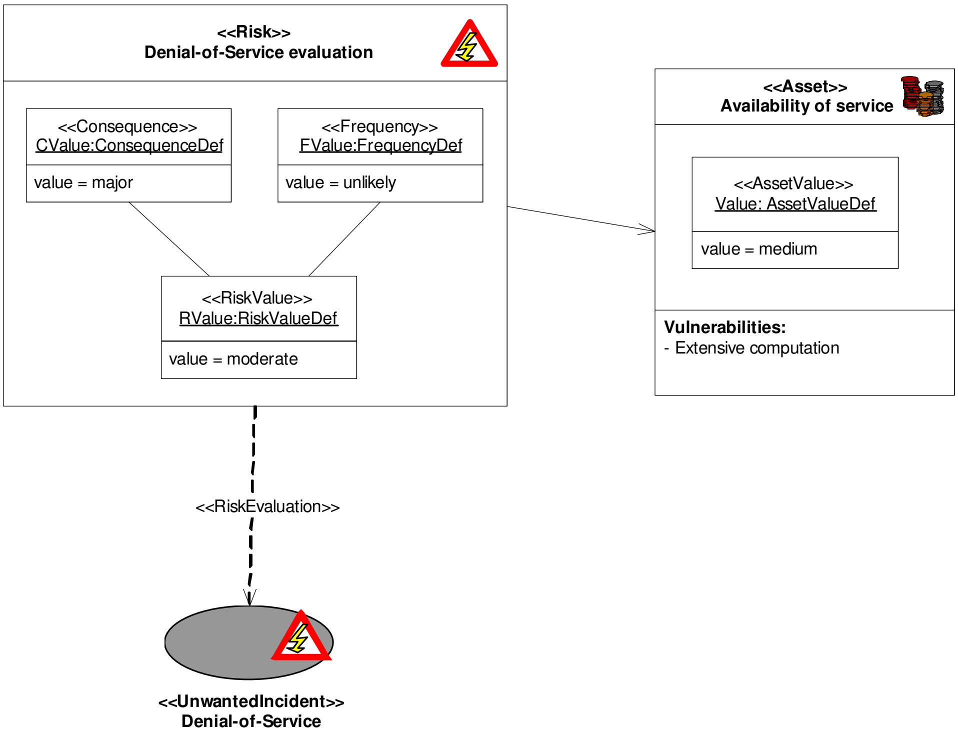


Figure 21 Modelling of risks

Similar risks may be grouped into risk themes. Figure 22 shows how the stereotype <<RiskTheme>> is used to define risk themes of instances of risks. This allows a risk to be a member of several risk themes. In this example, the risks "Denial-of-service evaluation" and "Loss of customer evaluation" are grouped to form the risk theme "DoSRelated". As seen in the example, a risk theme is also assigned a risk value.

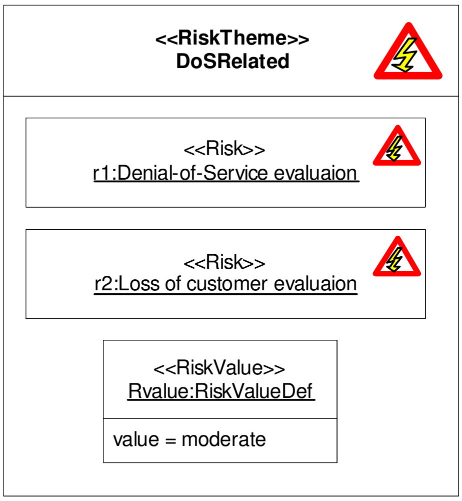


Figure 22 Modelling of risk themes

5.5 Treatment

Figure 23 models "Authentication" as a treatment for the unwanted incident "Denial-of-Service". The stereotype <<Transfer>> (one of the treatment options) explains what kind of treatment "Authentication" is.

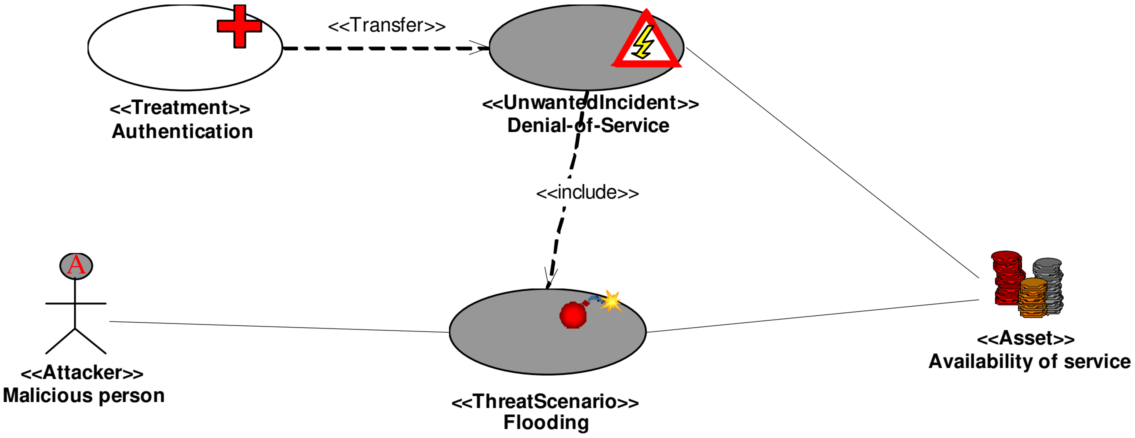


Figure 23 Modelling of treatments

In Figure 24 an example of how a treatment effect is modelled is presented. The treatment effect "DoSTransfer" is bound to the treatment "Authentication" by the use of the stereotype <<TreatmentEvaluation>>. The figure also shows that "DoSTransfer" relates to the risk "Denial-of-Service evaluation". The risk reduction, i.e., the value of the treatment effect, is a mapping from moderate to low, meaning that implementation of the treatment would reduce the risk value of "Denial-of-Service evaluation" from moderate to low.

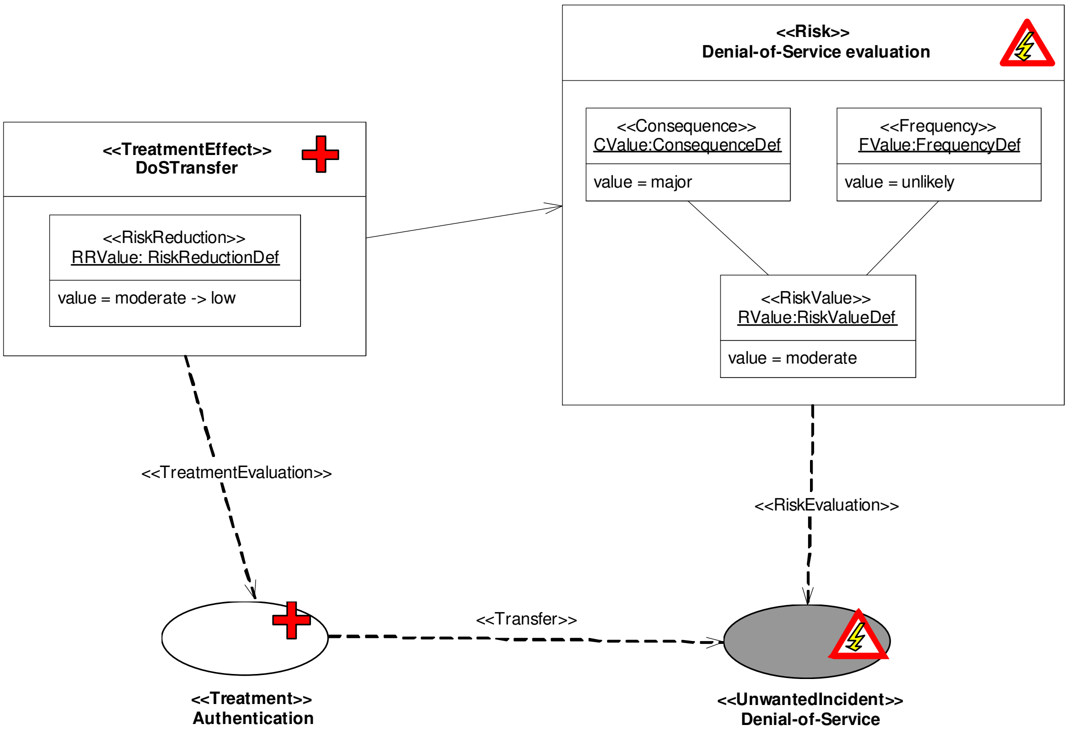


Figure 24 Modelling of treatment effect

6 Conclusions

This paper has presented graphical modelling language, implanted as a UML profile, for supporting security assessments with a special emphasis on communication between the stakeholders and other participants of the assessments. At the same time the profile is founded in a carefully designed metamodel which facilitates automated analysis of the models expressed in the language.

The profile was developed over time with continual feedback from the development of the security assessment methodology, as well as feedback from use of the profile in large scale trials where the methodology has been applied to real systems.

7 References

[1] Alexander, I. Misuse cases: Use cases with hostile intent. IEEE Software, 20(1):58-66, 2003.

[2] den Braber, F, Dimitrakos, T., Gran, B. A., Lund, M. S., Stølen, K., Aagedal, J. Ø. The CORAS methodology: Model-based risk assessment using UML and UP. In Favre, L., editor, UML and the Unified Process, chapter XVII, pages 332-357. IRM Press, 2003.

[3] Houmb, S. H., den Braber, F., Lund, M. S., Stølen, K. Towards a UML profile for model-based risk assessment. In Proc. UML'02 Satellite Workshop on Critical Systems Development with UML, pages 79-91, Munich University of Technology, 2002.

[4] Lund, M. S., den Braber, F., Stølen, K. Maintaining results from security assessments. In Proc. 7th European Conference on Software Maintenance and Reengineering (CSMR'03), pages 341-350. IEEE Computer Society, 2003.

[5] OMG. UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, Request for Proposals. OMG Document: ad/2002-01-07. Object Management Group, 2002.

[6] OMG. UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, Revised submission. OMG Document: realtime/2003-08-06. Object Management Group, 2003.

[7] OMG. Unified Modeling Language: Superstructure version 2.0, 3rd revised submission to OMG RFP ad/00-09-02. Object Management Group, 2003.

[8] Shirey, R., Internet security glossary. RFC 2828. Network Working Group, 2000.

[9] Sindre, G., and Opdahl, A.L, Eliciting Security Requirements by Misuse Cases. In Proc. TOOLS-PACIFIC 2000, pages 120-131. IEEE Computer Society, 2000.

[10] Sindre, G., and Opdahl, A.L., Templates for Misuse Case Description. In Proc. Workshop of Requirements Engineering: Foundation of Software Quality, (REFSQ'01). June, 2001.

[11] Stallings, W., Network security essentials, Application and standards, second edition. Prentice Hall, 2003.