UML as a tool for modelling of risks

1st iNTeg-Risk Conference

Swerea IVF AB, Mikael Ström

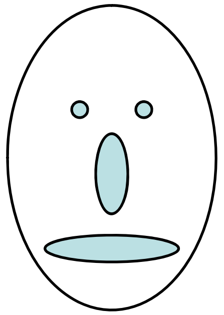
June 3, 2009
Stuttgart, Germany

Content

Models are often simplified representations of reality

Reality

Model



We select modelling method depending on what we want to model and what we want to enlighten/ emphasise/ communicate/ understand.

What is UML (1)

UML is an abbreviation of Unified Modeling Language

UML was created by OMG™ – Object Management Group is an international, open membership, not-for-profit computer industry consortium. OMG Task Forces develop enterprise integration standards for a wide range of technologies, and an even wider range of industries. OMG's modeling standards enable powerful visual design, execution and maintenance of software and other processes.

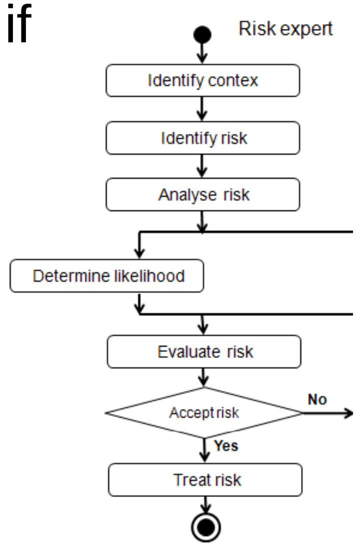
Fist version UML 1.0 came in 1997

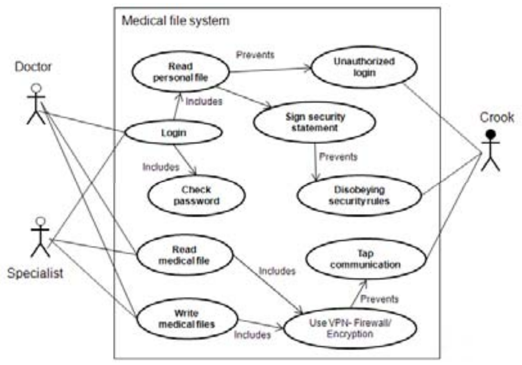
Today we have UML 2.1 and its definition and other useful documents are free to download from: www.omg.org

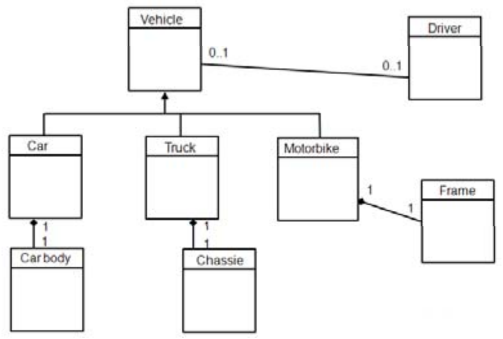
What is UML (2)

UML :

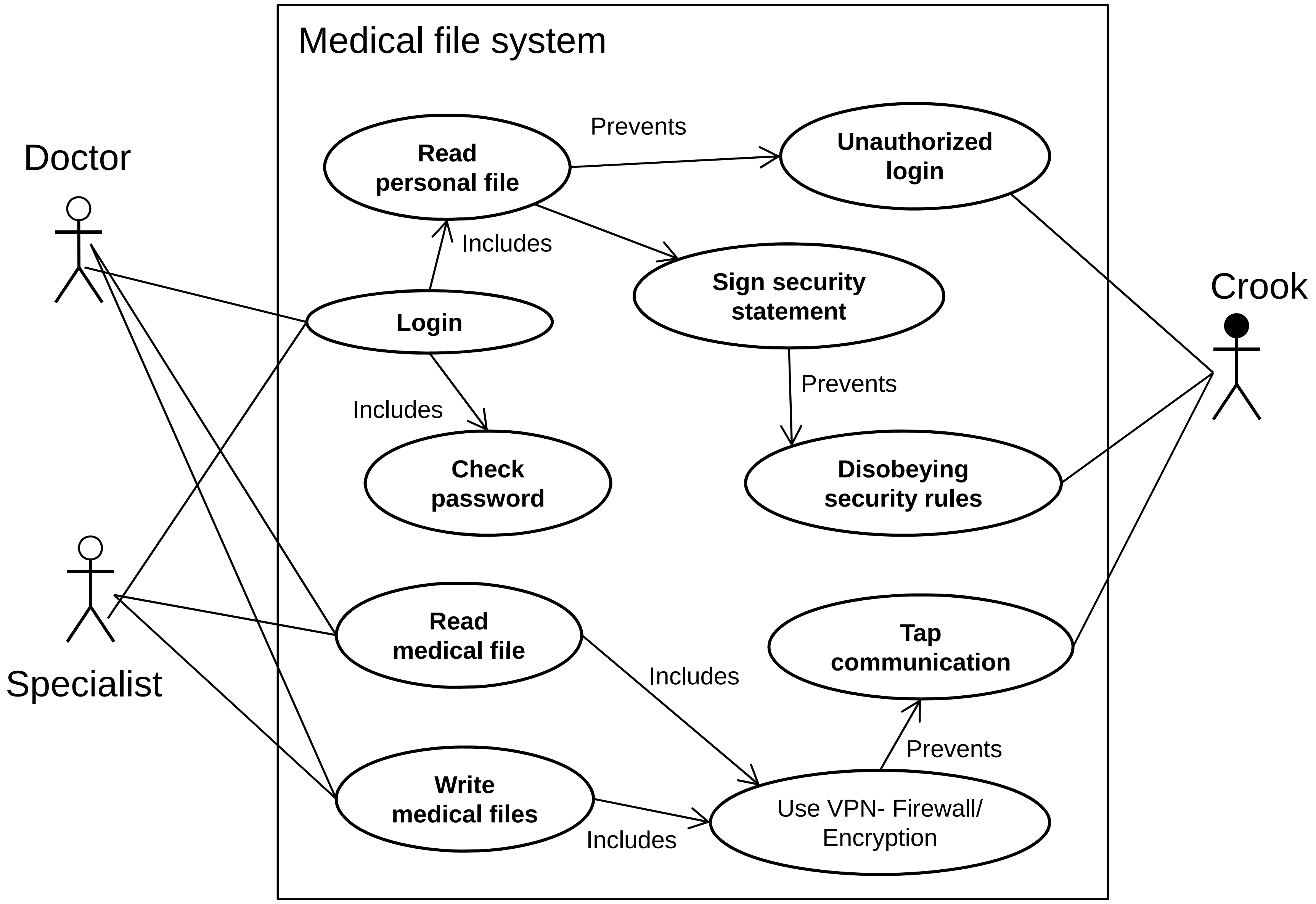
- Is a modelling method and has a graphical notation
- Has 13 different types of diagrams
- Is highly object oriented
- Can be used to model business processes, software systems, physical systems, information and many other things
- Is not limited in any way to software and electronics even if the origin of UML is in that area



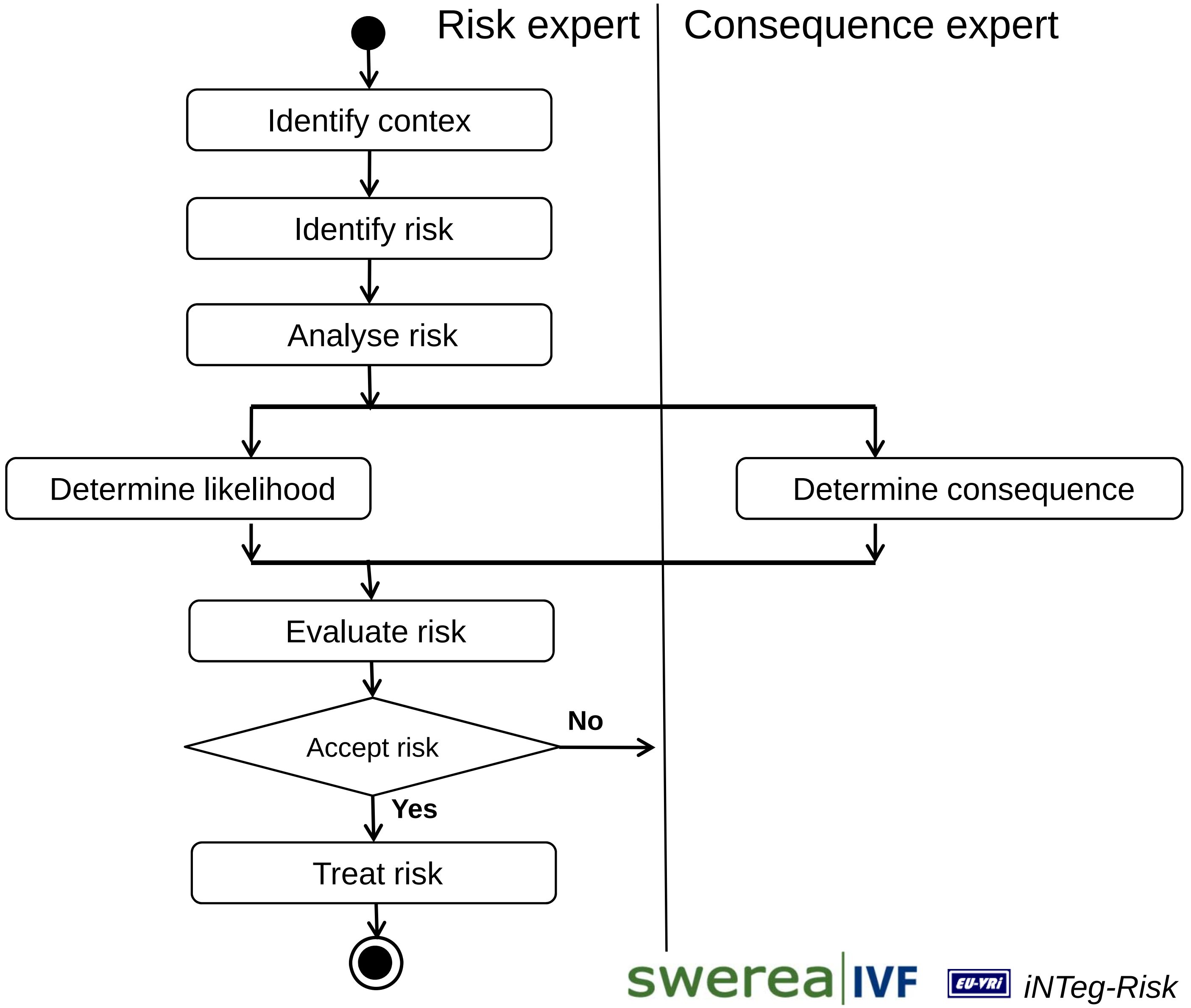




Use case diagram



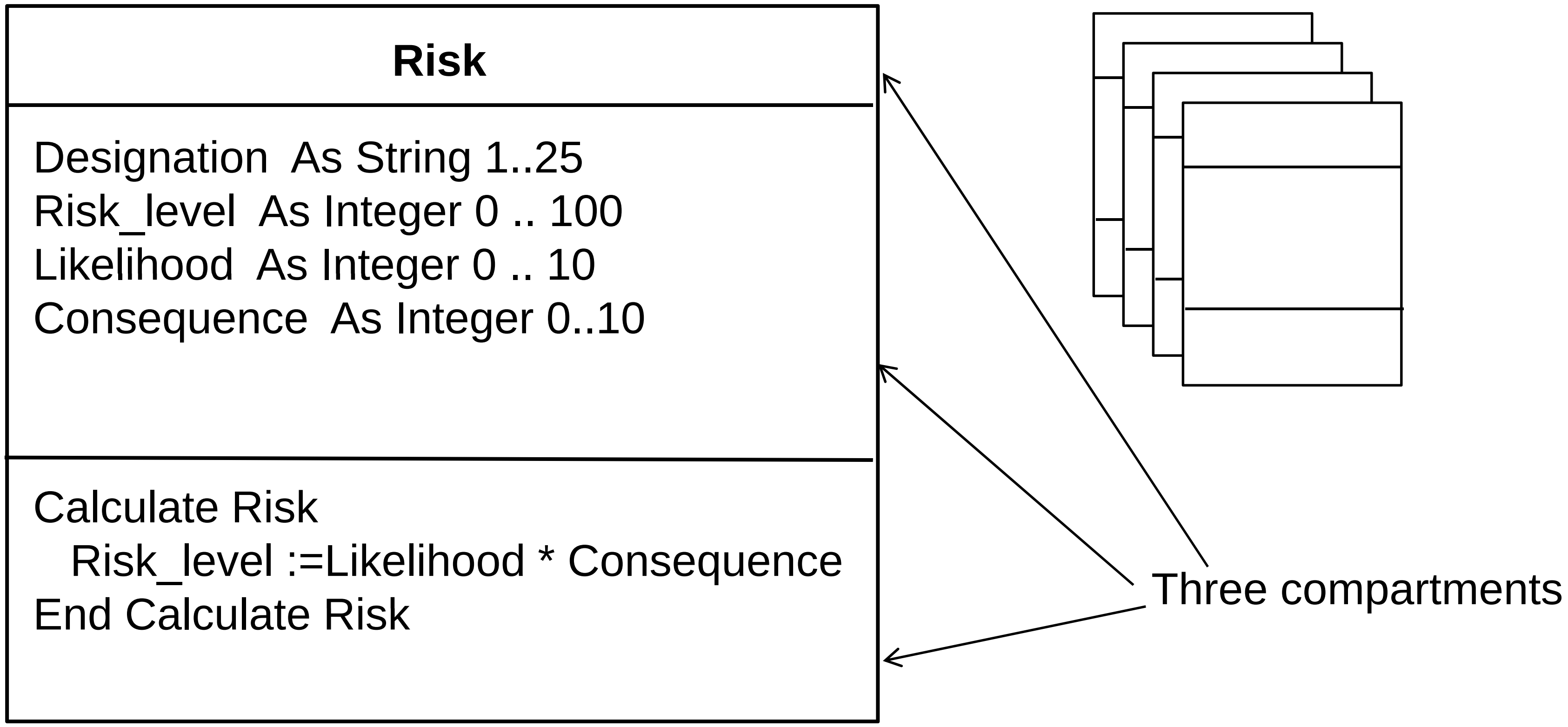
Activity diagram



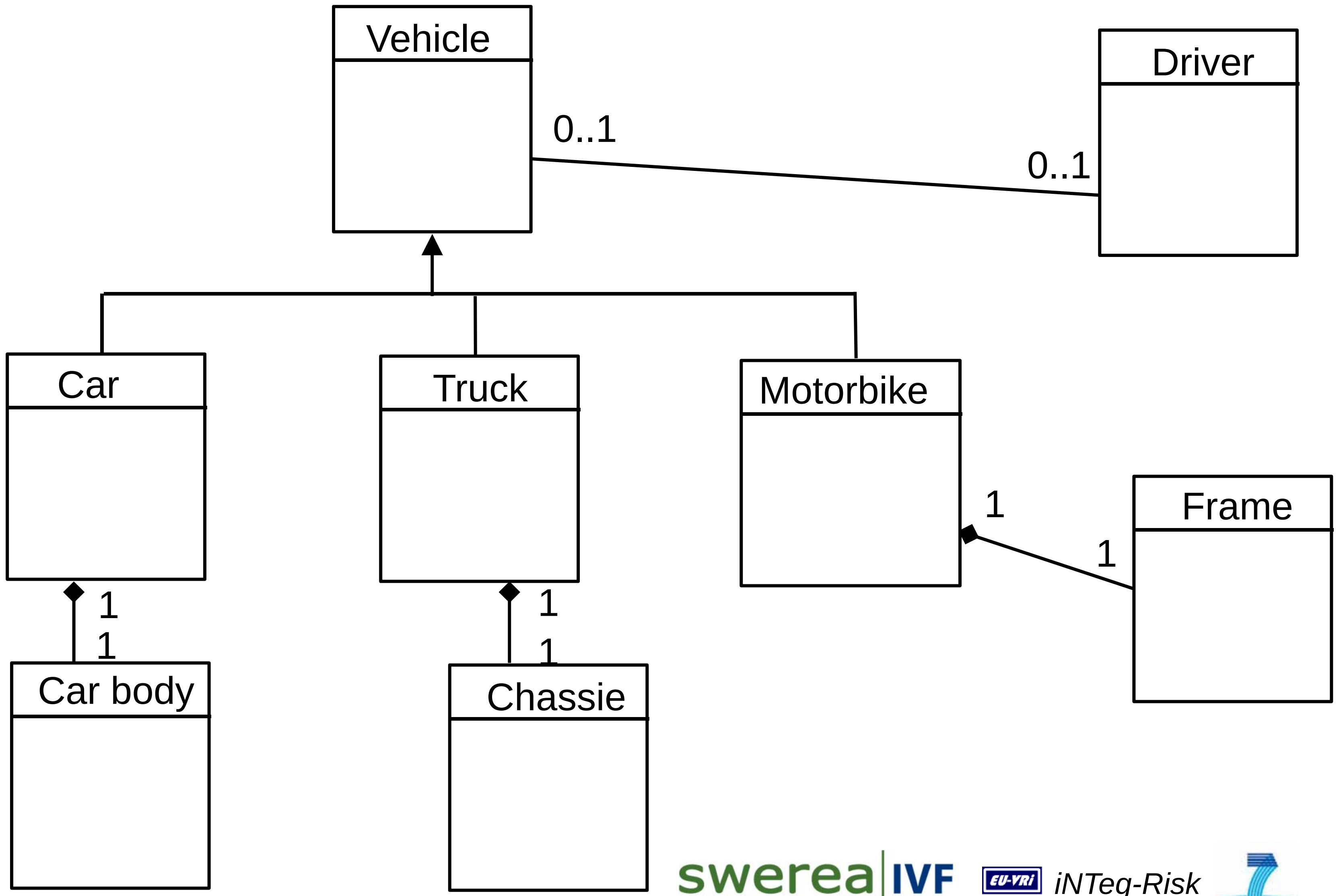
Most common is the Class diagram

One Class definition (simplyfied example)

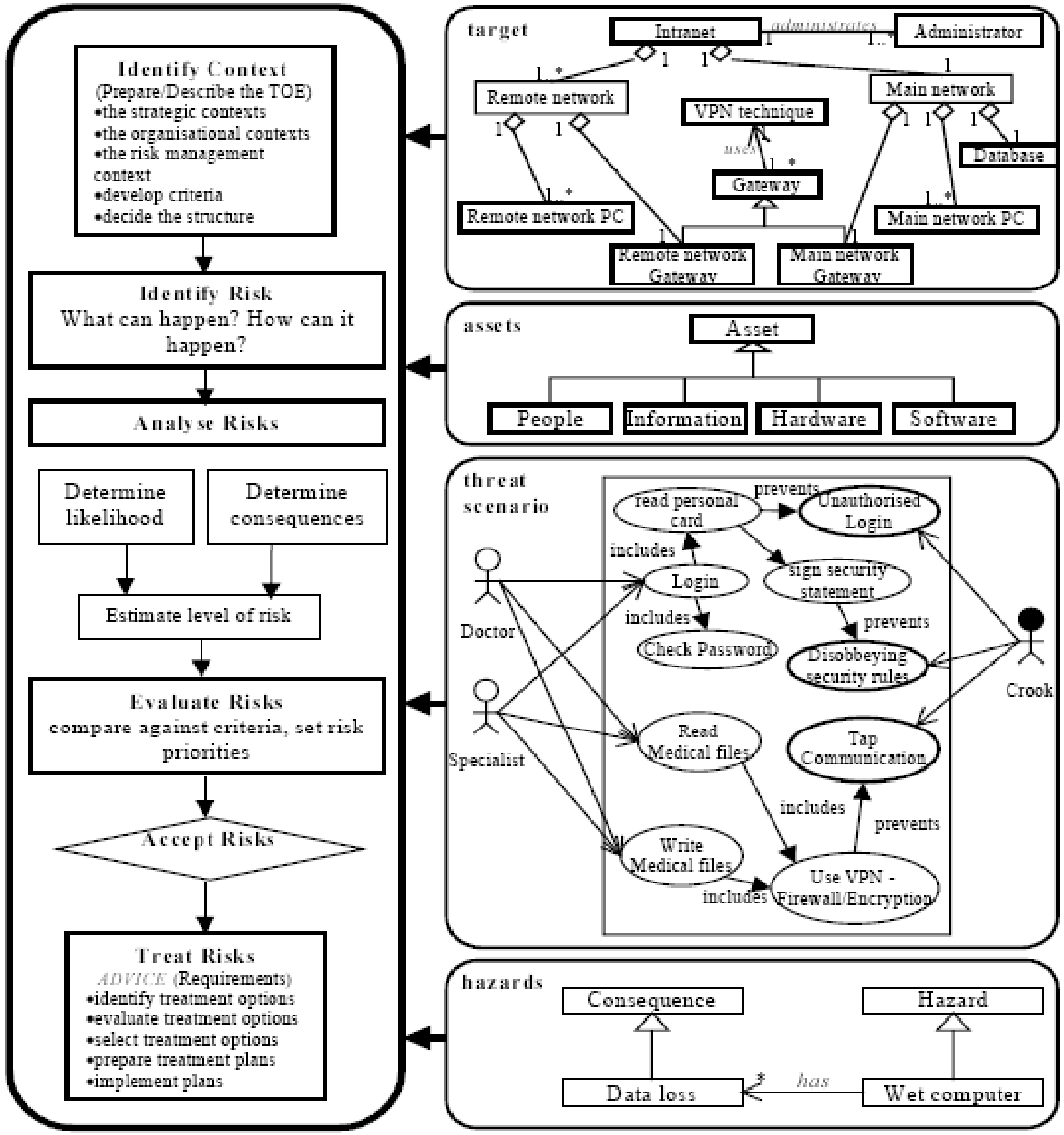
Many instances / objects


Risk
Designation As String 1..25
Risk_level As Integer 0 .. 100
Likelihood As Integer 0 .. 10
Consequence As Integer 0..10
Calculate Risk
Risk_level :=Likelihood * Consequence
End Calculate Risk
Three compartments

Relations of Class diagrams



Risk management process



Target

Assets

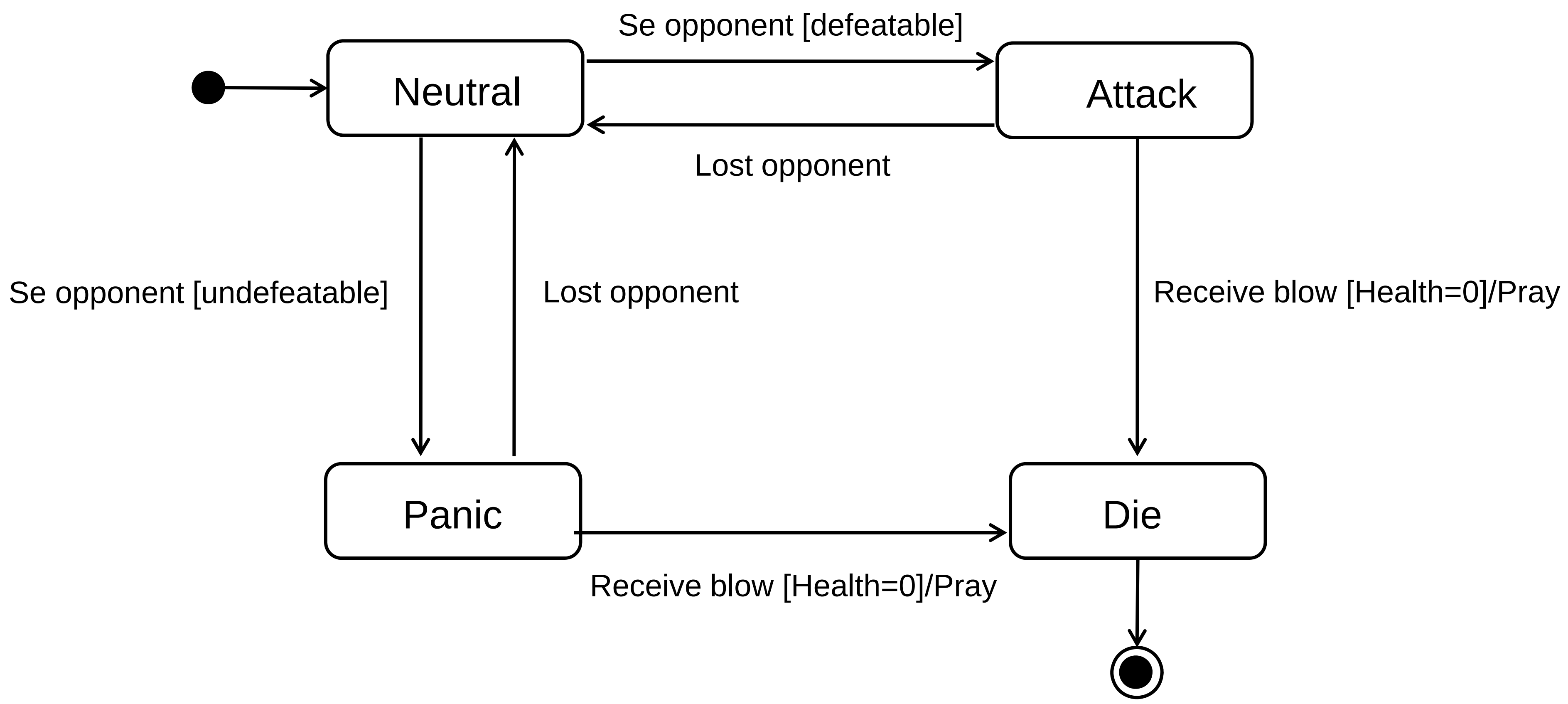
Threat scenario

Hazards

Ref: Coras

State chart diagram

Troll in First-Person Shooter (FPS) game


Se opponent [defeatable]
Neutral
Attack
Lost opponent
Se opponent [undefeatable]
Lost opponent
Receive blow [Health=0]/Pray
Panic
Die
Receive blow [Health=0]/Pray

The CORAS Language as an example

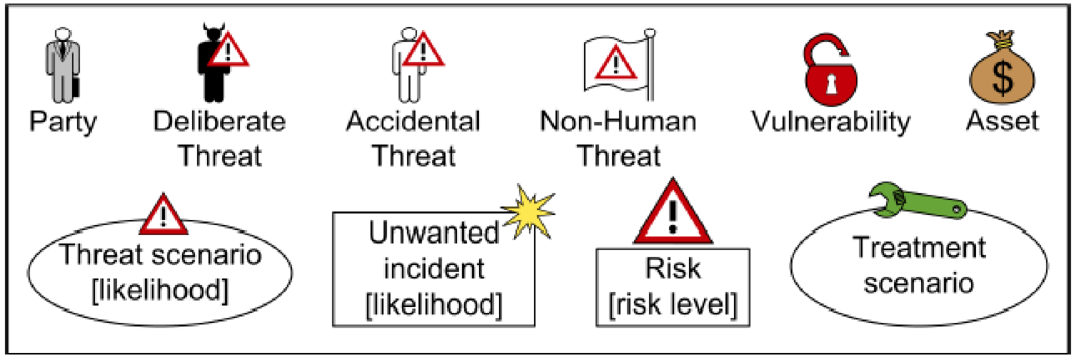


Figure 1: Basic building blocks of the CORAS diagrams

Treatment Category	Abbreviation
avoid	av
decrease likelihood	dl
decrease consequence	dc
share	sh
retain	re

Table 2: Treatment category abbreviations

Vertex	Instance
party	p
asset	a
deliberate threat	dt
accidental threat	at
non-human threat	nht
threat scenario	ts
unwanted incident	ui
risk	r
treatment scenario	trs

Annotation	Instance
vulnerability	$v = \{v\} = V_1$
vulnerability set	$V_n = \{v_1, \ldots, v_n\}$
likelihood	l
consequence	c
risk value	rv
risk function	rf

Table 1: Naming conventions

CORAS Security Risk Modeling Language

The syntax is an extension of UML

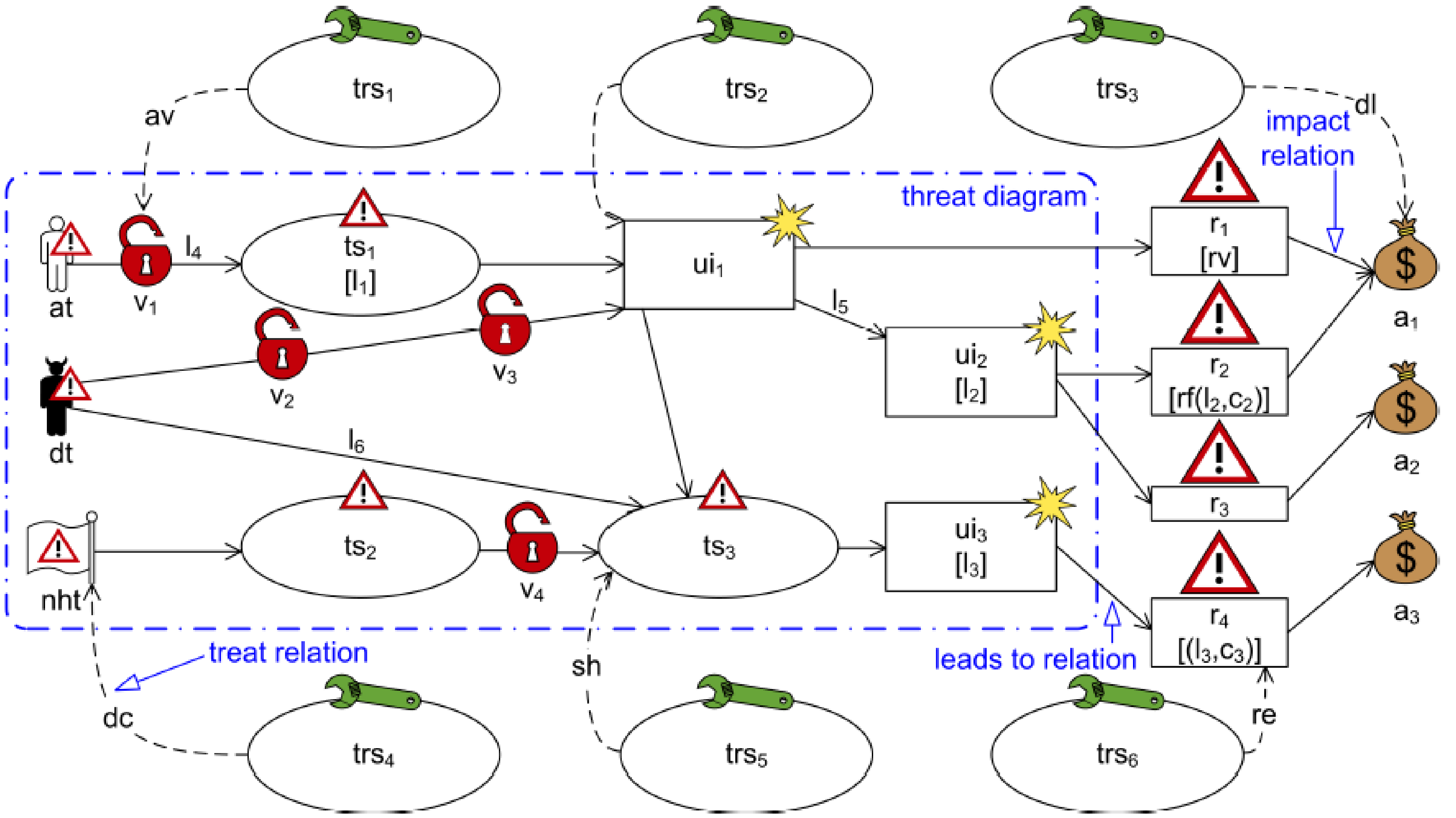
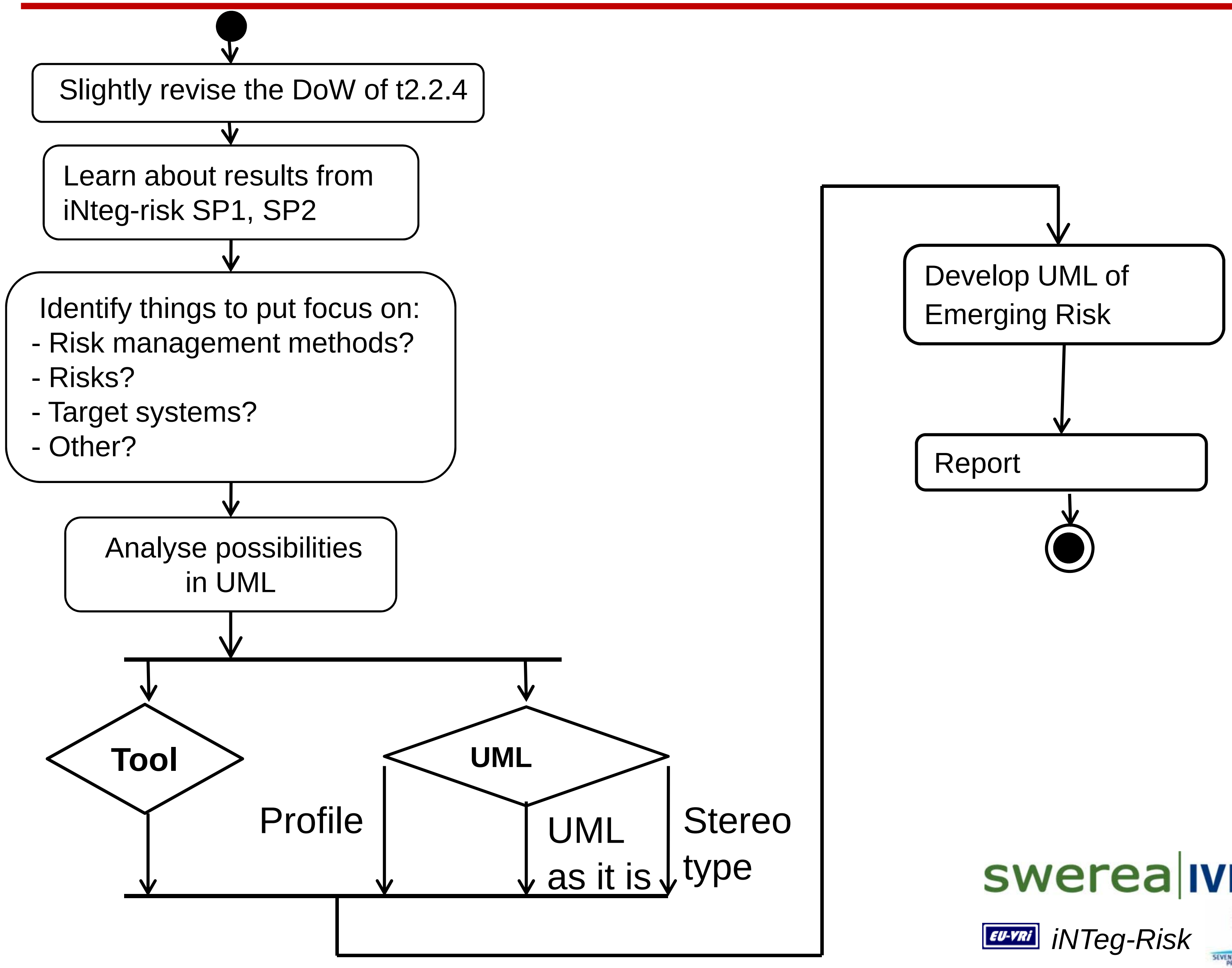


Figure 12: Graphical syntax of treatment diagrams

With UML of today we can model:

What to model	Diagram to use ?
Target Systems	Class diagram, Componet diagrams, Deployment diagrams, Object diagrams
Target system behaviour	State diagrams, Use cases, Sequence diagrams
Risks	Class diagrams
Risk management processes	Activity diagrams, Use cases
Dependencies	Class diagrams, object diagrams, State diagrams
Things that we want to put focus on regarding risks in emerging technologies	**iNteg-Risk developed diagram, stereotypes**

Work process of Task 2.2.4


Slightly revise the DoW of t2.2.4
Learn about results from iNteg-risk SP1, SP2
Identify things to put focus on:
- Risk management methods?
- Risks?
- Target systems?
- Other?
Analyse possibilities in UML
Tool
UML
Profile
UML as it is
Stereo type
Develop UML of Emerging Risk
Report

swerea | IVF

EU-VRi iNTeg-Risk

13 different UML diagrams

Diagram type	What can be modeled
Use Case	Interaction between target system and users or external systems
Activity	Sequential and parallel activities within a system
Class	Classes, types, interfaces and relationships
Object	Object instances between classes defined in class diagrams of a system
Sequence	Interaction between objects where the order of the interaction is important
Communication	The way objects interact and the connections that are needed to support interaction
Timing	Interactions between objects where timing is essential
Interaction Overview	Used to bring sequence communication, and timing diagrams together
Composite Structure	The internals of a class or component
Componen	Components within a system and the interfaces they use to interact with each other
Package	The hierarchical organization of groups of classes and components
State Machine	The state of an object throughout its lifetime and the event that can change the state
Deployment	How your system is finally deployed in a given real world situation

Tank you

Questions ?

Mikael.strom@swerea.se