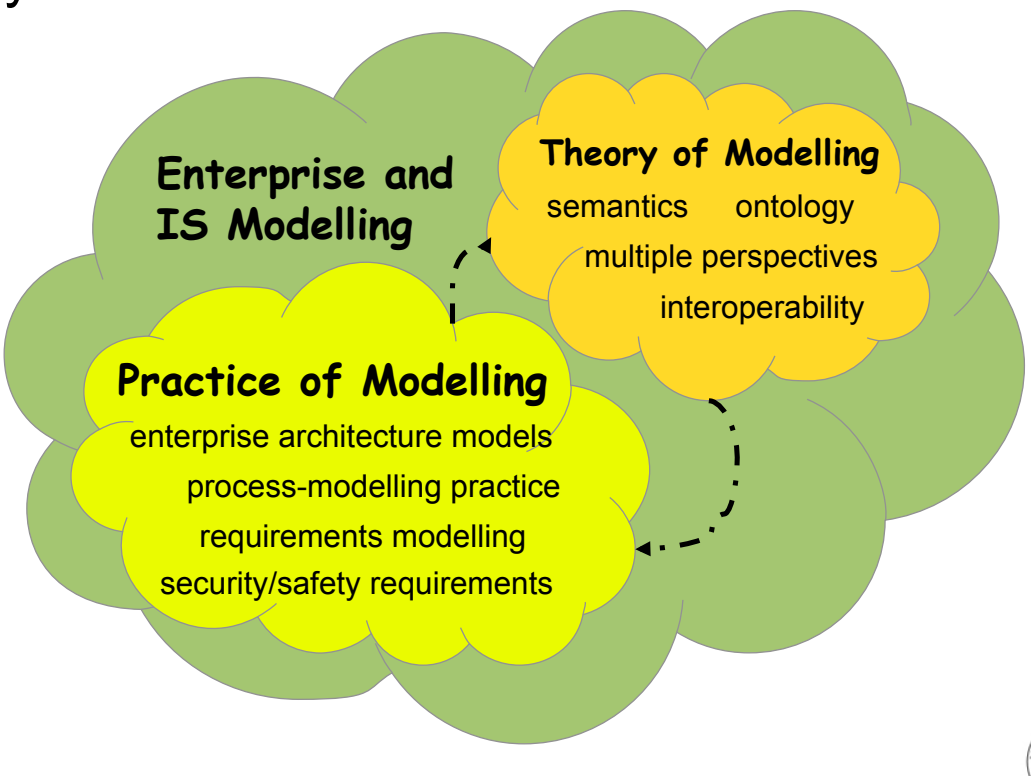
Identifying and Visualising Dependability Concerns: Results from a Requirements Project with Applications to Business Process Work

Andreas L Opdahl
University of Bergen, Norway

My research interests



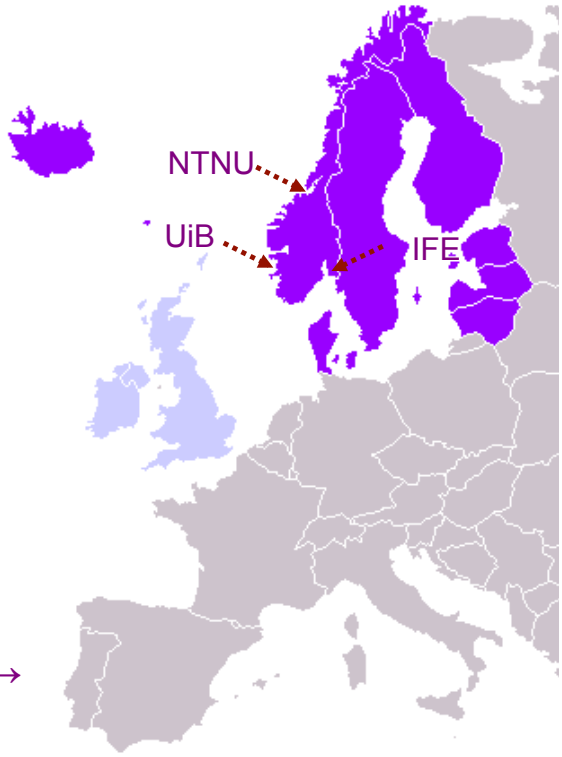
The **ReqSec** project!

Methods and tools for security requirements engineering:

- involve non-experts
- visualisation for inclusion
- lightweight, integrated
- industrially evaluated

Funded by the Norwegian Research Council (NFR), 2008-2012

Avoiding unwanted behaviours → dependability requirements



The **ReqSec** people!


Guttorm Sindre


Andreas L Opdahl


Christian Raspotnig


Peter Karpati


Vikash Katta
(sup. Tor Stålhane)

...lots of joint work!

What is dependability?

Dependability

- ability to deliver a service that can be justifiably trusted *(J.C. Laprie)*
- traditionally: availability, reliability and maintainability
- more recently: safety, security and privacy
- common theme: what we do *not* want to happen

Our focus:

- *security* – resilience to *intended* threats
- *safety* – resilience to *unintended* hazards

The importance of dependability

Several related developments:

- pervasive IS (and crucial parts of business processes)
- tightly integrated IS (and business processes)
- more complex intertwined business processes
  - parallelism, interactions, stakeholders, boundaries
- digitalisation and standardisation
- many types of dependability for same IS
- interactions between dependability types

Working with dependability requirements

Started with Misuse Cases (MUC)

- ...from 1999 *(Sindre & Opdahl, REJ 10, 2005)*
  - initial focus on *security*
  - **Negating** constructs from Use Case diagrams
  - visual notation for *abuse cases* *(McDermott & Fox)*

Related work:

- experiments
- case studies, design research
- tools (NTNU)
- methods (e.g., CORAS)
- uses for safety *(Stålhane, Sindre, ...)*

Misuse Case diagram example

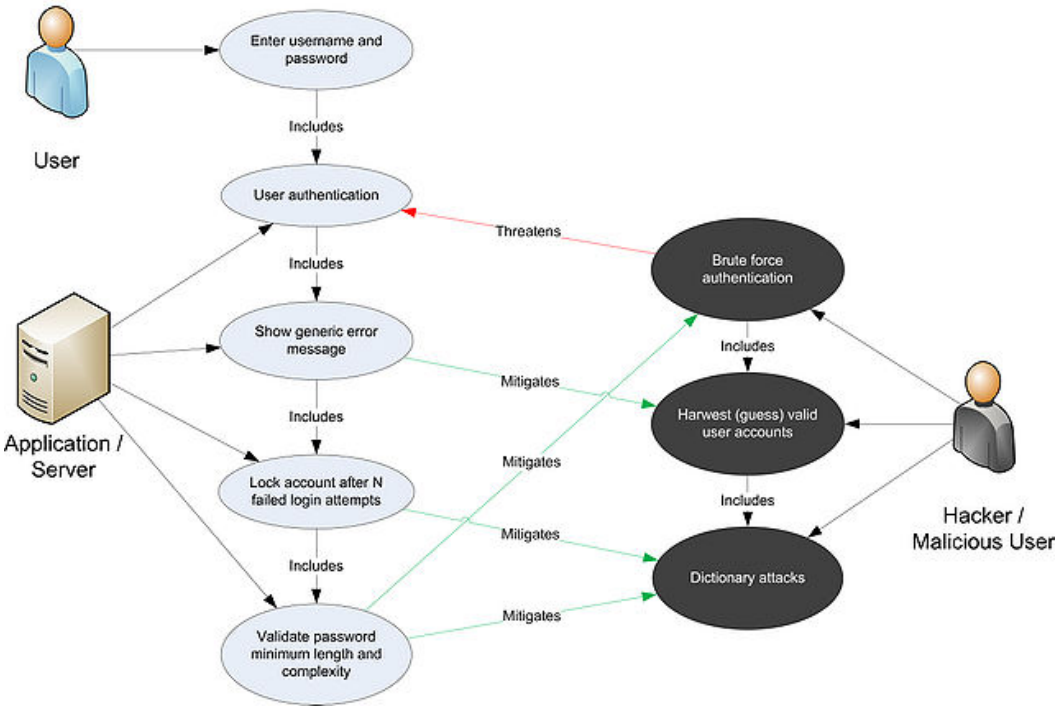


Diagram from owasp.org

Anti-behaviours in other notations

i* extensions *(Liu, Yu, Mylopoulos) (Elahi)*
Secure Tropos *(Mouratidis, Giorgini)*
Secure KAOS *(van Lamsweerde, ...)*
Abuse frames *(Lin, Nuseibeh, Ince, Jackson, Moffett)*
Mal-Activity Diagrams *(Sindre)*

Less focus on:

requirements and architecture
detailed analysis of attack sequence
integrated dependability method

Dependability requirements and architecture

System security models:

focus on single, monolithic systems
similar for safety

Security architecture frameworks (SABSA, TOGAF):

high-level views, enterprise security architecture
not a focus for safety

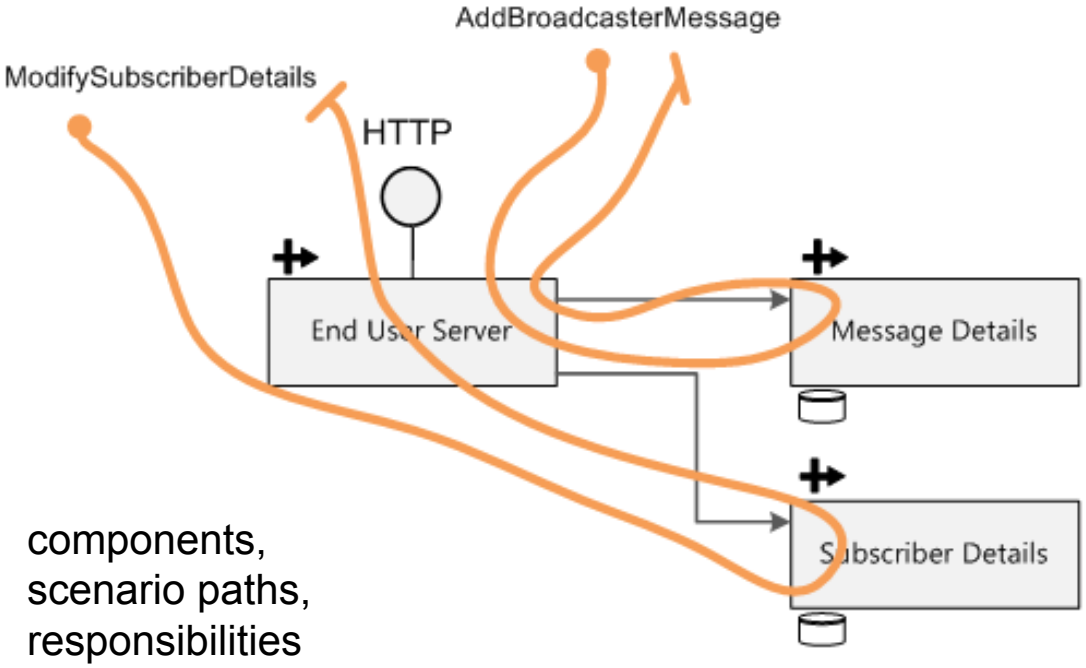
Need for intermediate solutions:

architectural security modelling, e.g., for SOA

Could we build on Use Case Maps (Buhr, Aymot, ...)?

Joint work with *Peter Karpati and Guttorm Sindre*

Use Case Map example



components,
scenario paths,
responsibilities

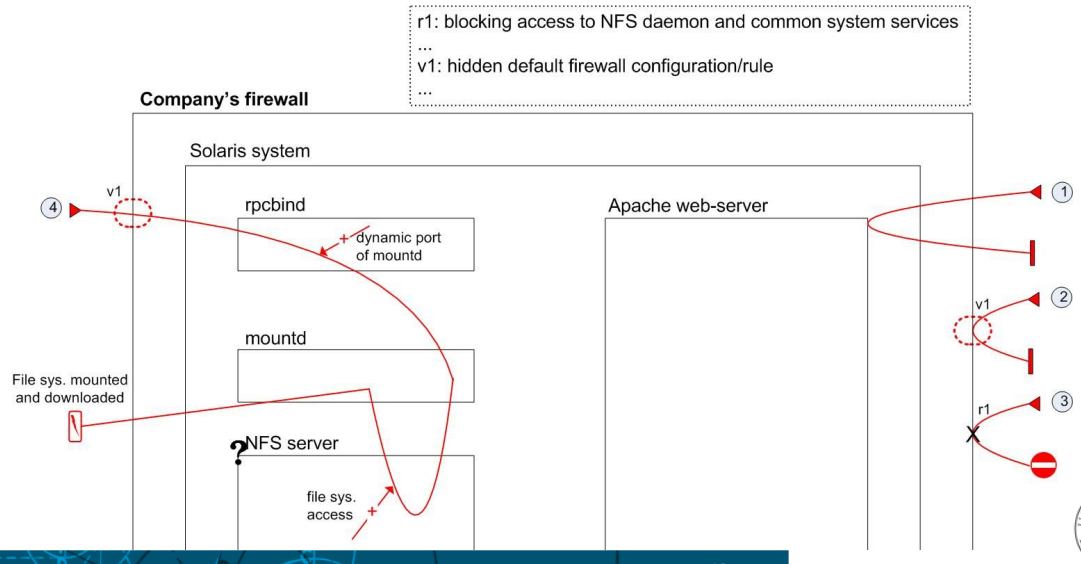
Diagram from www.softwarepractice.org

Misuse Case Maps (MUCM)

Misuse Case Maps:

vulnerabilities, exploit paths, vulnerable responsibility

– anti-behaviour in **red**, rather than **negated**



Misuse Case Maps (MUCM)

Research approach:

- working out cases (Mitnick's «The Art of Intrusion»)
- several experiments
- tool development (NTNU)

Conclusions:

- facilitates better understanding
- somewhat more productive than separate diagrams
- not clearly better liked

Also usable for safety?! *(Wu, Kelly)*

- guiding words?
- multiple failure modes?

Dependability requirements and sequence

Existing notations notations:

- few visualise *attack/failure sequence* in detail
- Mal-Activity Diagrams are an exception...

Could we build on *sequence diagrams*?

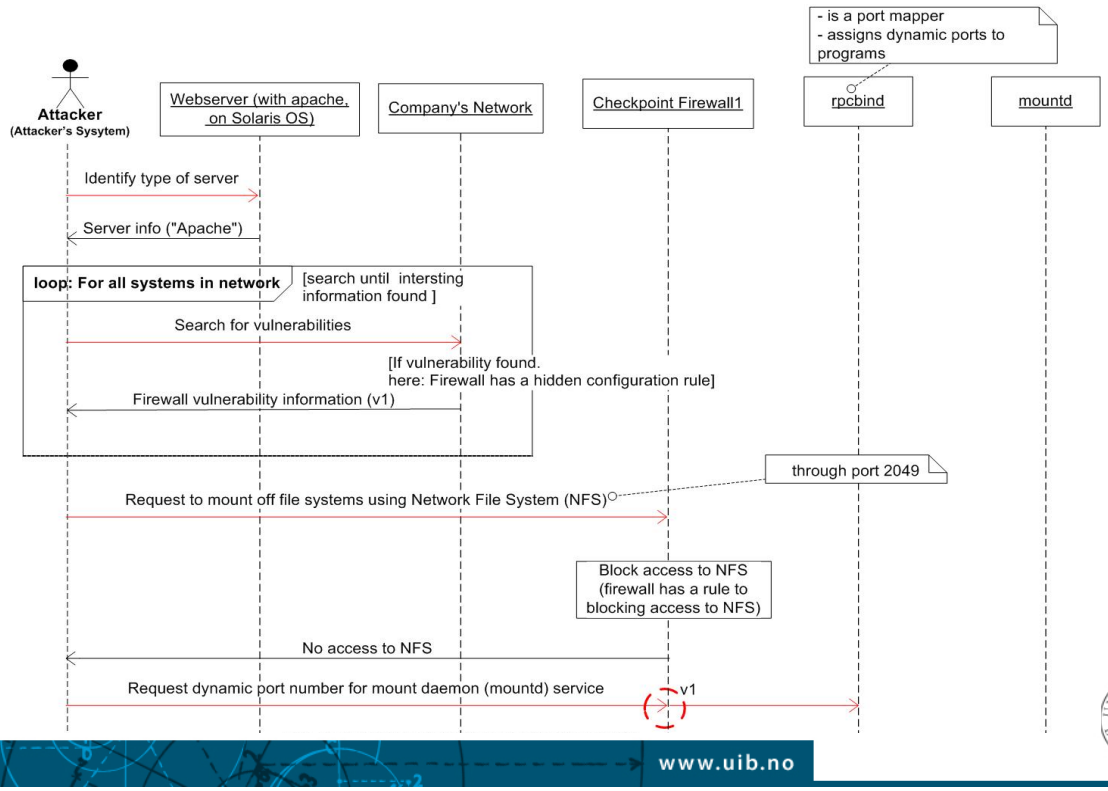
from: actor, object/component, action, event/ message

to: attacker, vulnerability, exploit action/event/ message

anti-behaviour again in **red**

Joint work with *Vikash Katta, Peter Karpati, Christian Raspotnig and Guttorm Sindre*

Misuse Sequence Diagrams (MUSD)



Misuse Sequence Diagrams (MUSD)

Research approach:

- working out cases
- experiments

Conclusions:

- complements MUCMs for understanding
- similarly effective to MUCMs
- better liked than MUCMs

Failure Sequence Diagrams (FSD)

The “safety variant” of MUSD

- similar notation, but safety terms

Used in air-traffic control:

- sequence of real live workshops
- combined with Failure Mode and Effect Analysis (FMEA)
- *Do FSD and FMEA combine well?*

Conclusions:

- more interactive analysis
- good for understanding *propagation*
- but the FSDs get complex
  - may not work for multiple failures

Joint work with *Christian Raspotnig*

Comparison of techniques

Around 5 safety and 5 security techniques

Systematic comparison through a framework:

- stakeholders, timing, type of system, application area, process, scalability, interoperability...

Systematic differences:

- maturity, visual notation,
- integration with development,
- structured method, cue words

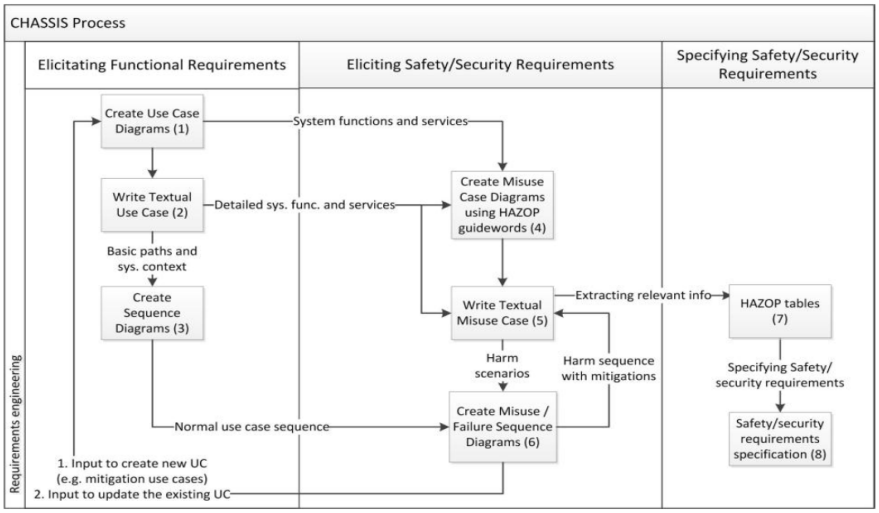
Towards an integrated

- conceptual model
- …and a *method*

Joint work with *Christian Raspotnig*

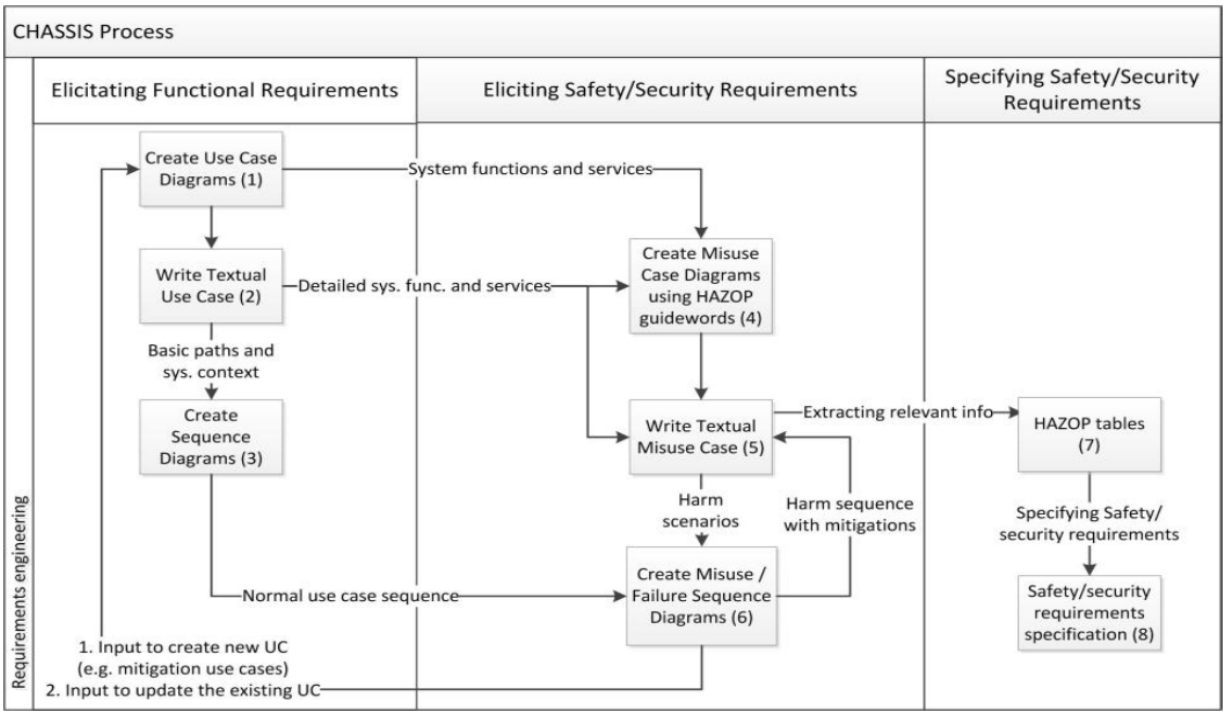
Integrated safety and security method

CHASSIS: Combined Harm Assessment of Safety and Security for Information Systems



Work by *Raspotnig, Karpati & Katta*

Diagram from (Raspotnig, Karpati & Katta)



HAZOP:

NOT, MORE, LESS, AS WELL AS, PART OF, REVERSE, OTHER THAN…

Applications to process work

Business processes are closely intertwined with the information systems that support them

The ideas behind and results from ReqSec should be readily transferred to business process work

Consider dependability early

no learning the hard way

avoid costly rework

control project risk

the best solutions may involve functional or architectural trade-offs

Involve many competencies

Safety, security, privacy etc. is not (only) an expert matter

...and not only a technical matter

Customers, suppliers, process owners and participants, systems users, developers etc. know the assets, hazards and threats best

...and understand the possible trade-offs better

Use visualisations

Central to involve multiple stakeholders

Central in the early development stages

Architecture/organisational structure and sequences

Broad, integrated handling of dependability

Dependability issues are becoming more important

More types of dependability are becoming important for the same systems

The different types interact – they must be investigated together

Using closely related (or the same) techniques and tools will make things simpler

Integrate risk assessment

The dependability types interact

- so their risks are dependent on one another
- integrated risk assessement is made easier when similar techniques and tools are used for different dependability types

Using boundaries

- Look for vulnerabilities, threats and hazards whereever a scenario path crosses a component boundary
- In business processes, organisational units are similar to components
- Hence *pools* and *swimlanes* are similar to *components* in UCMs and MUCMs
- Can be combined with *guiding words* *(Ubayashi & Kamei)*

Guiding words

Guiding words are central in safety

- *HAZOP: NOT, MORE, LESS, AS WELL AS, PART OF, REVERSE, OTHER THAN…*
- underused in security *(Srivatanakul, Winther et al.)*?
- ...and in process work?

A driving process that is both structured and encourages creativity

Use the semantics of process modelling constructs:

- dedicated guiding words, e.g., for actors and roles, swimlanes, actions, message flows, sequence flows, timers, alarms...

Remedies are potential vulnerabilities

Every mitigation must be analysed for dependability issues of its own
(Alexander)

Main points

Dependability is becoming more important

Many similarities between the dependability types

…but the fields are (largely) unrelated

We need new integrated techniques and methods

Empirical grounding through

real textbook cases, experiments with students and industry, industrial cases, design research

THANK YOU! :-)

Selected papers

Sindre, Guttorm; Opdahl, Andreas L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering 10(1) 2005.

Sindre, Guttorm; Opdahl, Andreas L.: Misuse Cases for Identifying System Dependability Threats. Journal of Information Privacy and Security 4(2) 2008.

Opdahl, Andreas L.; Sindre, Guttorm: Experimental Comparison of Attack Trees and Misuse Cases for Security Threat Identification. Information and Software Technology 51(5) 2009.

Karpati, Peter; Opdahl, Andreas Lothe; Sindre, Guttorm: Experimental Comparison of Misuse Case Maps with Misuse Cases and System Architecture Diagrams for Eliciting Security Vulnerabilities and Mitigations. Proc. Sixth Int' Conf. on Availability, Reliability and Security. IEEE Comp. Soc. 2011.

Karpati, Peter; Opdahl, Andreas Lothe; Sindre, Guttorm: Experimental evaluation of misuse case maps for eliciting security requirements. 1st Security Conf. Europe 2010.

Karpati, Peter; Sindre, Guttorm; Opdahl, Andreas Lothe: Characterising and Analysing Security Requirements Modelling Initiatives. Proc. Sixth Int. Conf. on Availability, Reliability and Security. IEEE Comp. Soc. 2011.

Karpati, Peter; Sindre, Guttorm; Opdahl, Andreas Lothe: Visualizing Cyber Attacks with Misuse Case Maps. LNCS 6182, Springer 2010.

Karpati, Peter; Sindre, Guttorm; Opdahl, Andreas Lothe: Towards a hacker attack representation method. Proc. Fifth Int. Conf. on Software and Data Technologies. INSTICC Press 2010.

Katta, Vikash; Karpati, Peter; Opdahl, Andreas Lothe; Raspotnig, Christian; Sindre, Guttorm: Comparing two techniques for intrusion visualization. LNBIP 68, Springer 2010.

Raspotnig, Christian; Opdahl, Andreas Lothe: Improving Security and Safety Modelling with Failure Sequence Diagrams. International Journal of Secure Software Engineering 3(1) 2012.

Raspotnig, Christian; Opdahl, Andreas Lothe: Supporting Failure Mode and Effect Analysis: A Case Study with Failure Sequence Diagrams. Proc. REFSQ'12. LNCS 7195. Springer 2012.