CHARACTERISING AND ANALYSING SECURITY REQUIREMENTS MODELLING INITIATIVES

Peter Karpati[1], *Guttorm Sindre*[1], *Andreas L. Opdahl*[2]
[1] Norwegian University of Science and Technology,
[2] University of Bergen

SecSE Vienna, 22-26 August 2011

NTNU
Institutt for datateknikk
og informasjonsvitenskap

UNIVERSITAS BERGENSIS

Outline

1. Motivation
2. Characterizing dimensions
    A. Introduction
    B. The 10 dimensions
3. Conclusion and future work

1. Motivation

- Eliciting and modelling security requirements are (should be) the most fundamental activities for engineering secure systems
- Many initiatives can be found in this area
- To describe, compare, characterize them to match their abilities to the needs of stakeholders
- 10 papers examined which were containing surveys, reviews, comparisons of SRE initiatives

About the ten papers

- From 2005-2010
  - more than half from 2009-2010
- Technical reports, conference and journal papers, magazine articles
- Number of identified SRE initiatives varies between 9 and 64
  - It seems that the authors concentrate more on deeper investigation of the identified initiatives than including many of them in their analysis

Cont.

- New conceptual frameworks are developed with sophistication often based on previous, well-established frameworks for the analysis and comparison
- Different groups tend to use different sets and definitions of basic SRE notions and charcterizing features (if any) though a slow convergence can be observed
- Not a complete collection

2. Characterizing dimensions

- Papers selected based on a thorough search of the literature (but no systematic process was followed which is a limitation)
- Focus on classification and comparison frameworks for security engineering initiatives (later narrowed to SRE)
- After eliciting the characterizing dimensions from he papers, they were grouped according their focus
- Main dimensions with sub-dimensions were synthesised per group based on alignment of their concepts
- Final result: 9(+1) synthesised main dimensions each including some sub-dimensions

Running example: misuse cases

- Misuse cases (MUC)
  - complement use cases (UC) for security purposes by extending them with *misusers*, *misuse cases* and *mitigation use cases*, as well as new relations like *threatens* and *mitigates.*
- A stepwise process to develop a use case diagram including misuse cases was defined
- A five steps process to elicit security requirements with MUC was also defined

Representation perspective

- defines the type of approach according to the construct that it is founded on (based on *Nhlabatsi et al. [9])*
- Type of approaches
  - Goal-based
  - Model-based
  - Problem-oriented
  - Process-oriented
- Example: misuse cases (MUC) are classified as a problem-oriented initiative

Kind of SRE tasks/activities

- defines of which parts of the security requirement development process are covered by the initiative. The most commonly recommended tasks or activities are considered (based on *Tøndel et al.* [1] and *Du et al.* [6]).
- (a) security objectives; (b) identification and modeling of assets, vulnerabilities and threats; (c) elicitation and analysis of SRs; (d) specification, documentation of SRs; (e) verification and validation support
- MUC: (a – partially), (b), (c), (d)

Specification criteria for SRE

- In the context of Sw. Eng., specification is a description of externally known features, a complete behaviour. The fulfilment of a specification criterion can partially help to achieve the fulfilment of several technical criteria. (From Villarroel et al. [3] and Mellado et al. [10].)
- (a) understandable, (b) unambiguous, (c) complete, (d) consistent, (e) correct, (f) verifiable, (g) validateable, (h) modifiable, (i) traceable, (j) appropriate

Technical criteria for SRE

- A software specification technique is a method to achieve the desired purpose or product. The fulfilment of a technical criterion must generate the fulfilment of all specification criteria related to that criterion. (From Villarroel et al. [3] and Mellado et al. [10].)
  - internal verification support (b,c,d,e,g,h,i),
  - external validation support (e,g),
  - support for documentation generation (a),
  - standards integration (a,c,d,f),
  - requirements reuse (d,h,j),
  - support for other development stages (c,h,i),
  - help support (-),
  - easy to use (-)

Specification and technical criteria – example (MUC)

- internal verification support
  - +: unambiguous, complete, correct, validateable, modifiable
  - P: consistent, traceable
- external validation support
  - +: correct
- support for documentation generation
  - P: understandable
- requirements reuse
  - +: consistent, modifiable; P: appropriate
- support for other development stages
  - +: traceable; P: complete, modifiable
- help support: +; easy to use: +; standards integration: –

Modelling language criteria

- Useful distinction between the *modelling language* and the *modelling process* of a technique. Further, the techniques can be organized into a *method* with its own steps of the application of the techniques.
- Modelling language criteria for security specification languages/techniques (from Khan and Zulkernine [8])
  - ability to formulate basic security requirements (MUC: +)
  - ability to represent usage scenarios (MUC: +)
  - ability to represent security mechanisms and low level security requirements (MUC: -)
  - similarity with software specification languages (MUC: +)
  - reuse of provided artefacts in later phases (MUC: testing)
  - tool support (MUC: +)

Modelling and method process criteria

- The modelling process of deriving security requirements using a specification language should be considered though it is discussed only on the base of the involved activities in Khan and Zulkernine [8].
- The method process criteria for secure software development (SSD) processes
  - development resources (MUC: -)
  - reusable artefacts (MUC: +)
  - usage in the industry (MUC: +)

Software evolution support

- how much is software evolution management possible in S(R)E initiatives
- Sub-dimensions (0: no support; 3: full support)
  - Modularity
    - MUC – 2: modules are use cases
  - Component architecture
    - MUC – 1: no explicit support
  - Change propagation
    - MUC – 0: focuses on identifying misuses rather than interactions between functions
  - Change impact analysis
    - MUC – 2: implicitly, it is possible to identify MUC for UC

Relevant SRE notions

- Fabian et al. [2] presents a conceptual framework for security engineering with strong focus on security requirements elicitation and analysis
- Basic notions used for comparison
  - Security goal (MUC: ~)
  - Security requirement (MUC: -)
  - Specification (MUC: security req.)
  - Stakeholder (MUC: ~Actor)
  - Domain knowledge (MUC: -)
  - Asset (MUC: ~)
  - Threat (MUC: ~)
  - Vulnerability (MUC: -)
  - Risk (MUC: ~)
  - This set might be extended with additional concepts like mitigation.

Central concepts of Fabian et al.'s framework

- Criteria (+: considered explicitly; - not considered explicitly)
  - CIA triad: MUC +
  - Other than security requirements: MUC +
  - Stakeholders view: MUC -
  - Multi-lateral view: MUC -
  - Orientation towards the technical IT system: MUC -
  - Orientation towards to its environment: MUC +
  - Inclusion of threats: MUC +
  - Inclusion of risk analysis: MUC +
  - Means for quality assurance: MUC -
  - Means for formal verification: MUC -

3. Summary

- Representation perspective: needs extension
- Kind of SRE tasks/activities: might need details
- Specification criteria: ok
- Technical criteria: ok
- Modelling language criteria: might need ext.
- Method process criteria: needs extension
- Modelling process criteria: needs investigation
- Sw. evolution support: ok
- Relevant SRE notions: needs ext.
- Central concepts of Fabian et al.'s framework : needs further clarification

Conclusion and further work

- Conclusion
  - Clearer definitions needed often
  - The set of dimension has the potential to provide detailed knowledge about the relevant aspects of SRE initiatives without having to know them e.g for decision support and reasoning about a choice
- Further work
  - Build a uniform characterising framework from the set of dimensions based on an organizing concept
  - Apply it for SRE initiatives comparison
  - Try it with industrial partners requiring consultancy in this area

THANK YOU FOR YOUR ATTENTION!

RESERVE SLIDES

The ten papers

- Tøndel I, Jaatun M, Meland P (2008) Security requirements for the rest of us: a survey. Software IEEE 25(1):20–27
- B. Fabian, S. F. Gürses, M. Heisel, T. Santen, H. Schmidt: A comparison of security requirements engineering methods. Requir. Eng. 15(1): 7-40 (2010)
- R. Villarroel, E. Fernández-Medina, M. Piattini, Secure information systems development — a survey and comparison, Computers & Security, 2005, pp. 308–321.

Cont.

- G. Yee, Recent Research in Secure Software, unpublished report, available as of Jan. 20, 2006 from: http://www.georgeyee.ca
- M. A. Hadavi, V. S. Hamishagi, H. M. Sangchi, Security Requirements Engineering; State of the Art and Research Challenges, Proceedings of The International MultiConference of Engineers and Computer Scientists 2008 , pp985-990
- Jing Du, Ye Yang, Qing Wang: An Analysis for Understanding Software Security Requirement Methodologies. Proceedings of The Third IEEE International Conference on Secure Software Integration and Reliability Improvement, SSIRI 2009: 141-149

Cont.

- Sunyaev, Ali; Tremmel, Florian; Mauro, Christian; Leimeister, Jan Marco; and Krcmar, Helmut, "A Reclassification of IS Security Analysis Approaches" (2009). *AMCIS 2009 Proceedings*. Paper 570.
- M.U.A. Khan and M. Zulkernine, “A Survey on Requirements and Design Methods for Secure Software Development,” Technical Report 2009-562, School of Computing, Queen’s University, Canada, 2009.
- Armstrong Nhlabatsi, Bashar Nuseibeh, Yijun Yu: Security Requirements Engineering for Evolving Software Systems: A Survey. International Journal of Secure Software Engineering (IJSSE), 1(1): 54-73 (2010)
- Daniel Mellado, Carlos Blanco, Luís Enrique Sanchez, Eduardo Fernández-Medina: A systematic review of security requirements engineering. Computer Standards & Interfaces 32(4): 153-165 (2010)