Use Case Oriented Development of Security-Critical Systems

Gerhard Popp, Jan Jürjens, Guido Wimmel

Department of Computer Science
Munich University of Technology, Germany
(popp|juerjens|wimmel@in.tum.de)

Overview

- Introduction
  - Critical System Development
  - Use Case Development
- Security Use Case Development
  - Methodical Concept
  - Security Data Modeling
  - Security Use Case Modeling
  - Integration and Further Steps
- Conclusion

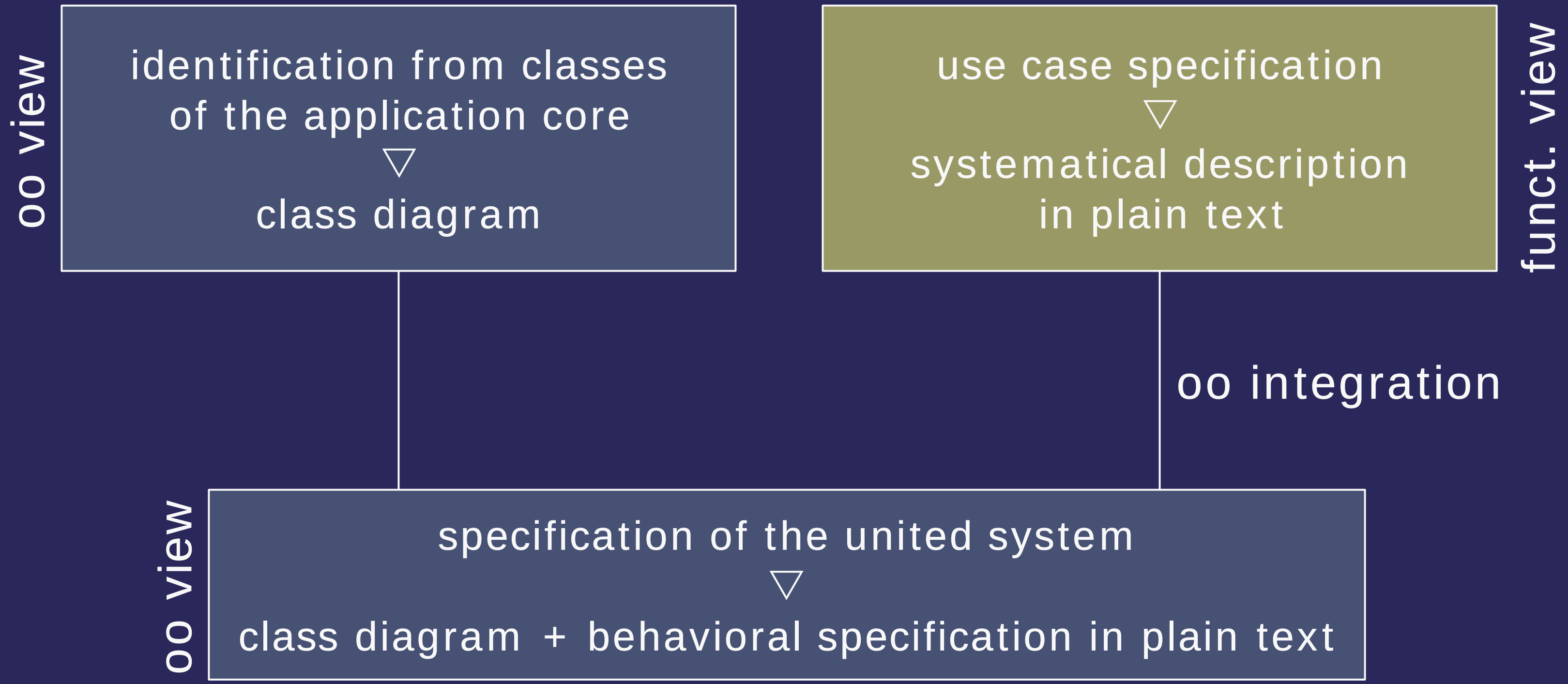
Critical Systems Development

- High quality development of critical systems (dependable, security-critical, real-time, ...) is difficult.
- Many systems developed, fielded, used that do not satisfy their criticality requirements, sometimes with spectacular failures.
- Security is mostly an add-on to the common system development.

Use Case Oriented Development

- Developers and customers think both about objects and about tasks in the early phases.
- Object-Oriented modeling of the data model.
- Use Cases for the modeling of dynamic aspects.

Methodical Concept for Use Case Oriented Development



oo view

identification from classes of the application core

▽

class diagram

use case specification

▽

systematical description in plain text

funct. view

oo integration

oo view

specification of the united system

▽

class diagram + behavioral specification in plain text

Use Case Description

- Which actor is involved?
- Which data or objects does the actor exchange with the system?
- Which classes of the core system are changed by the execution?
- Which expected behavior does the system show?
- Which variants of the expected behavior do exist?

Security Use Case Oriented Development

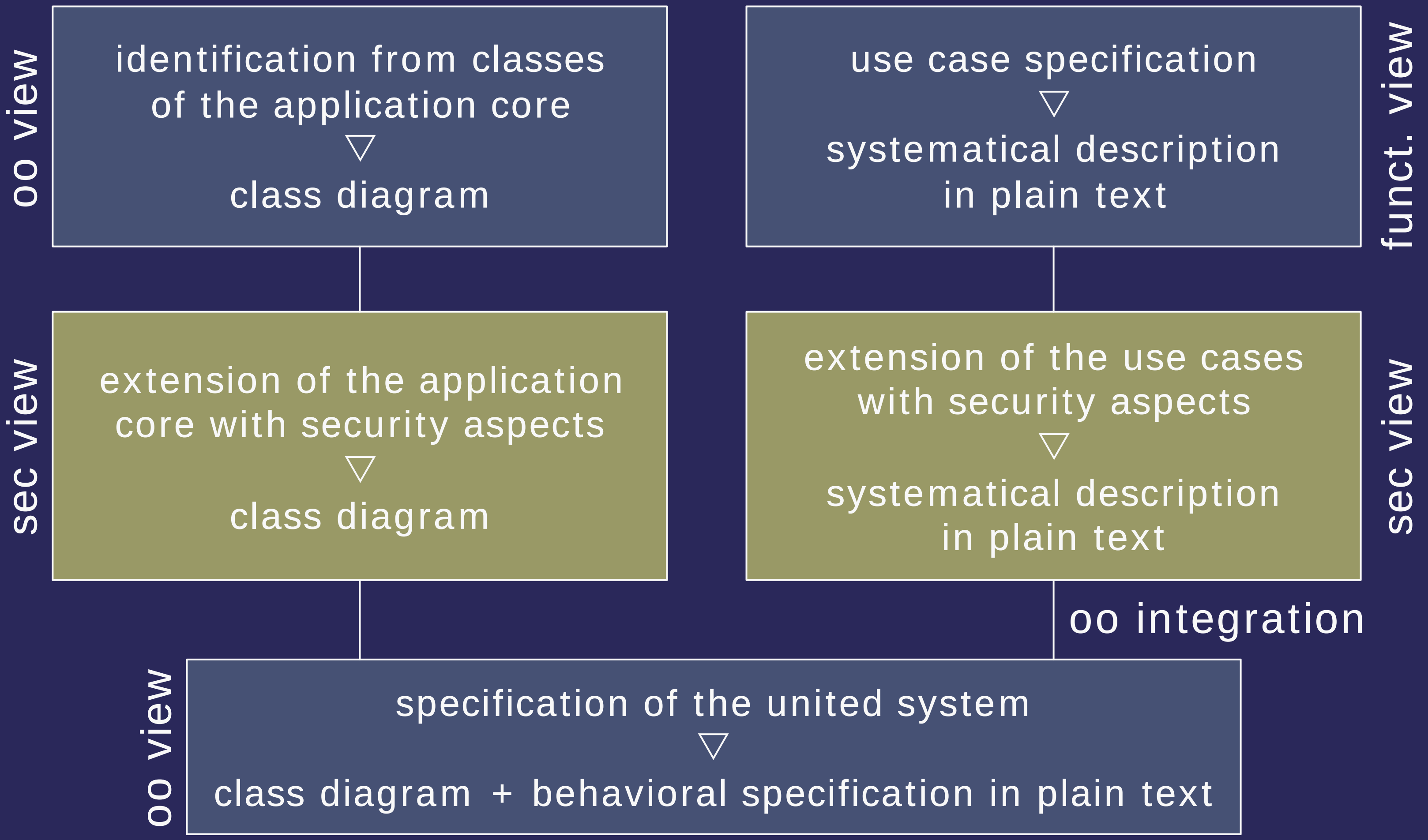
- Security is a complex non-functional requirement which can only be guaranteed by the interaction of many parts in the system.

- Described as non-functional requirement in plain text.

- Security in manageable pieces.

- Integration in an object-oriented development process.

Methodical Concept for Specifying Security-Critical Systems



oo view

identification from classes of the application core
▽
class diagram

funct. view

use case specification
▽
systematical description in plain text

sec view

extension of the application core with security aspects
▽
class diagram

sec view

extension of the use cases with security aspects
▽
systematical description in plain text

oo integration

oo view

specification of the united system
▽
class diagram + behavioral specification in plain text

Static Security Data Modelling with UMLsec

- Data modeling with class diagrams.
- Add security related information to class diagrams.
- Extension of class diagrams with:
  - Stereotypes << >>
  - Tags {tag=value}
  - Constraints

Extension of Class Diagrams for Secure Data Modeling

Stereotype	Base Class	Tags	Description
secrecy	dependency		assumes secrecy
integrity	dependency		assumes integrity
high	dependency		high sensitivity
critical	object	secrecy integrity high	Critical object

Secure Data Modeling <<secure links>>

- Ensures that physical layer meets security requirements on communication.
- Constraint:
  For each dependency d with stereotype

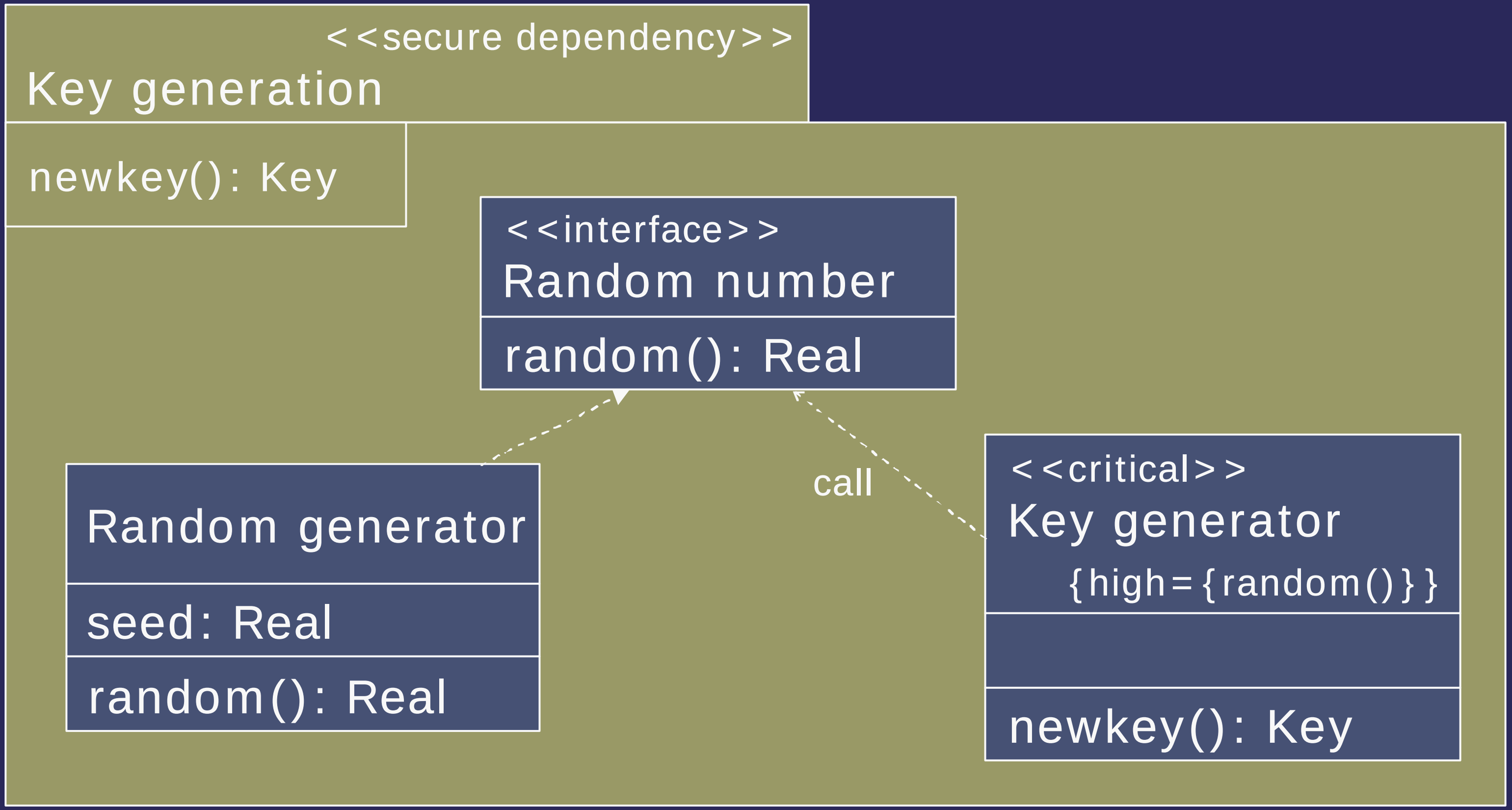
  s ? {<<secrecy>>, <<integrity>>, <<high>>}

  between components on nodes n?m, have a communication link l between n and m with stereotype t such that:
  - if s = <<high>>: have Threats$_A$(t) =Ø .
  - if s = <<secrecy>>: have read ? Threats$_A$ (t).
  - if s = <<integrity>>: have insert ? Threats$_A$(t).

Secure Data Modeling <<secure dependency>>

- Ensures that <<call>> and <<send>> dependencies between components respect security requirements on communicated data given by tags {secrecy}, {integrity}, {high}.
- Constraint:
  Given <<call>> or <<send>> dependency from C to D:
  - Any message n in D appears in {secrecy} in C if and only if does so in D.
  - If message in D appears in {secrecy} in C dependency stereotyped <<secrecy>>.
  - Analogously for {integrity} and {high}.

Secure Data Modeling Example: Key Generation Subsystem Instance



Security Use Case Modeling

- Mapping of the overall security aspects to use cases.
- Extension of the use cases, if necessary.
- 3 types of (security) use cases:
  - not security-critical
  - security critical
  - new use case for security aspects

Security Extension of Use Cases – Questions (1)

- Which risks are connected with the actor who is involved in or starts the use case?
- Which input and output data of the use case is security-critical (with respect to security aspects)?
- Which classes have to be modified and how strongly does the security aspect affect the classes?

Security Extension of Use Cases – Questions (2)

- How does the modified system behavior look like?
- Do we have additional action sequences due the specification of security aspects?

Security Extension of Use Cases – Description Techniques

- Description in structured text.
- Additionally specification of the behavior in a sequence diagram, if necessary.
- Security aspects can be emphasized with curly brackets.

Security Use Case Extension – Establish a Connection (Example) (1)

- Risk associated with the actor
  - The actor establishes a connection without any encryption, so there is no way to identify the sender and the connection can be eavesdropped.
- Security I/O Data
  - Input: Phone number
    {The phone number is critical, because it can be attacked.}
  - Output: Message ready or aborted
    {The messages are critical, because they can be attacked.}

Security Use Case Extension – Establish a Connection (Example) (2)

- Modified Classes
  - Connection
- Added Classes
  - Key generation, Key storage, Crypt message
- Modified System Behavior
  - The system has to exchange the data over a secure channel.
  - After the caller selects a phone number, the system generates a session key.
  - The system exchanges the session key.
  - The system can encrypt and decrypt the data.

Outline of the Integration

- Extension of the class diagram with a class for every use case.
- Mapping of security aspects:
  - Add data structure for input and output data, if necessary.
  - Map the security aspects of the input and output data.
  - Need of special security classes.

Resources

- Book on related UMLsec:
Jan Jürjens, Secure Systems Development with UML, Springer-Verlag, 2003

Thanks for your attention!