Model-based Security with UMLsec

Jan Jürjens

Software & Systems Engineering

Informatics, Munich University of Technology

Germany

jan@jurjens.de

http://www.jurjens.de/jan

A Need for Security

Society and economies rely on computer networks for communication, finance, energy distribution, transportation...

Attacks threaten economical and physical integrity of people and organizations.

Interconnected systems can be attacked anonymously and from a safe distance.

Networked computers need to be secure.

Problems, Causes

Many flaws found in design or implementation of security-critical systems, sometimes years after publication or use.

- Designing secure systems is difficult.
- Designers often lack background in security.
- Security as an afterthought.
- Cannot use security mechanisms „blindly“.

Previous approaches

„Penetrate-and-patch“:

- insecure
- disruptive

Traditional formal methods: expensive.

- training people
- constructing formal specifications.

Goal: Security by design

Consider security

- from early on
- within development context
- taking an expansive view
- in a seamless way.

Secure design by model analysis.

Secure implementation by test generation.

Using UML

UML: unprecedented opportunity for high-quality critical systems development feasible in industrial context:

- De-facto standard in industrial modeling: large number of developers trained in UML.
- Relatively precisely defined.
- Many tools in development.

Used fragment of UML

Activity diagram

Class diagram

Sequence diagram

Statechart diagram

Deployment diagram

Package

Stereotypes, tags, constraints

Current: UML 1.5

UML Extension mechanisms

Stereotype: specialize model element using ≪label≫.

Tagged value: attach {tag=value} pair to stereotyped element.

Constraint: refine semantics of stereotyped element.

Profile: gather above information.

UMLsec

UMLsec: extension for secure systems development.

- evaluate UML specifications for vulnerabilities
- encapsulate security engineering patterns
- also for developers not specialized in security
- security from early design phases, in system context
- make certification cost-effective

The UMLsec profile

Recurring security requirements as stereotypes with tags (secrecy, integrity,...).

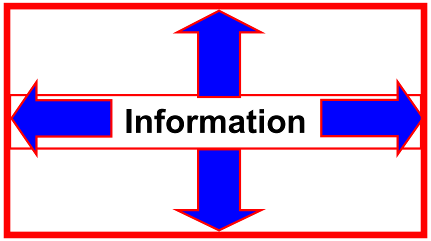
Associated constraints to evaluate model, indicate possible vulnerabilities.

Ensures that stated security requirements enforce given security policy.

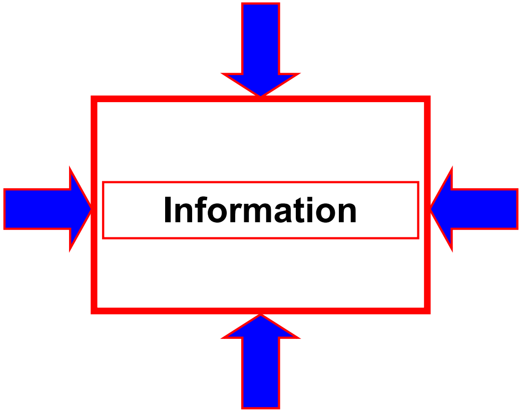
Ensures that UML specification provides requirements.

Basic Security Requirements

Secrecy



Integrity



≪Internet≫, ≪encrypted≫, …

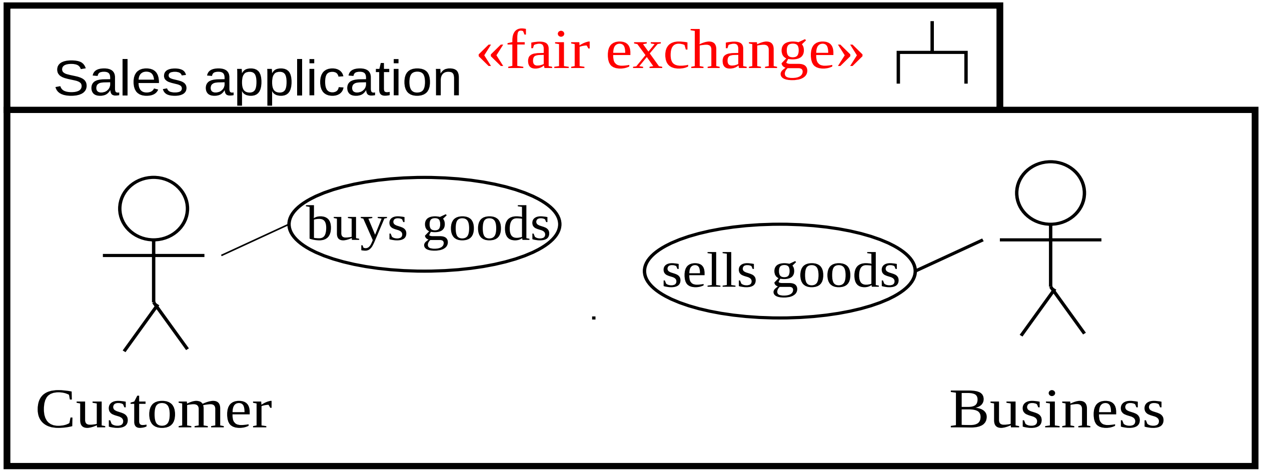
Kinds of communication links resp. system nodes.

For adversary type A, stereotype s, have set $\mathrm{Threats}_A(s) \subseteq$ {delete, read, insert, access} of actions that adversaries are capable of.

Default attacker:

Stereotype	$\mathrm{Threats}_{default}()$
Internet	{delete, read, insert}
encrypted	{delete}
LAN	∅
smart card	∅

Requirements with use case diagrams



Capture security requirements in use case diagrams.

Constraint: need to appear in corresponding activity diagram.

≪fair exchange≫

Ensures generic fair exchange condition.

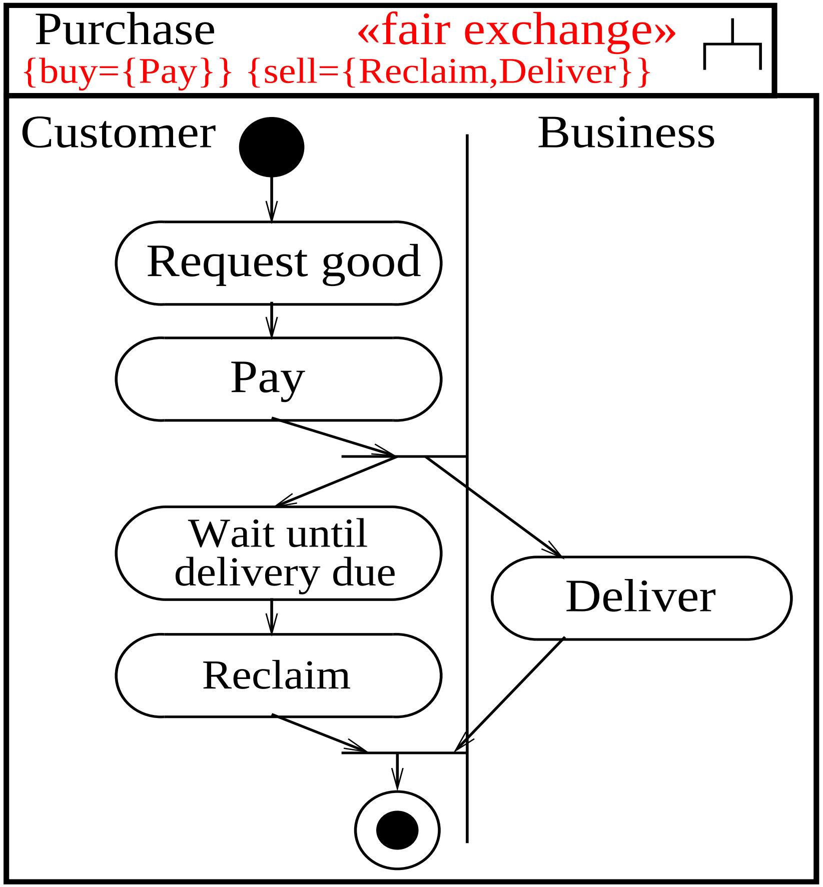
Constraint: after a {buy} state in activity diagram is reached, eventually reach {sell} state.

(Cannot be ensured for systems that an attacker can stop completely.)

Example ≪fair exchange≫

Customer buys a good from a business.

Fair exchange means: after payment, customer is eventually either delivered good or able to reclaim payment.



≪secure links≫

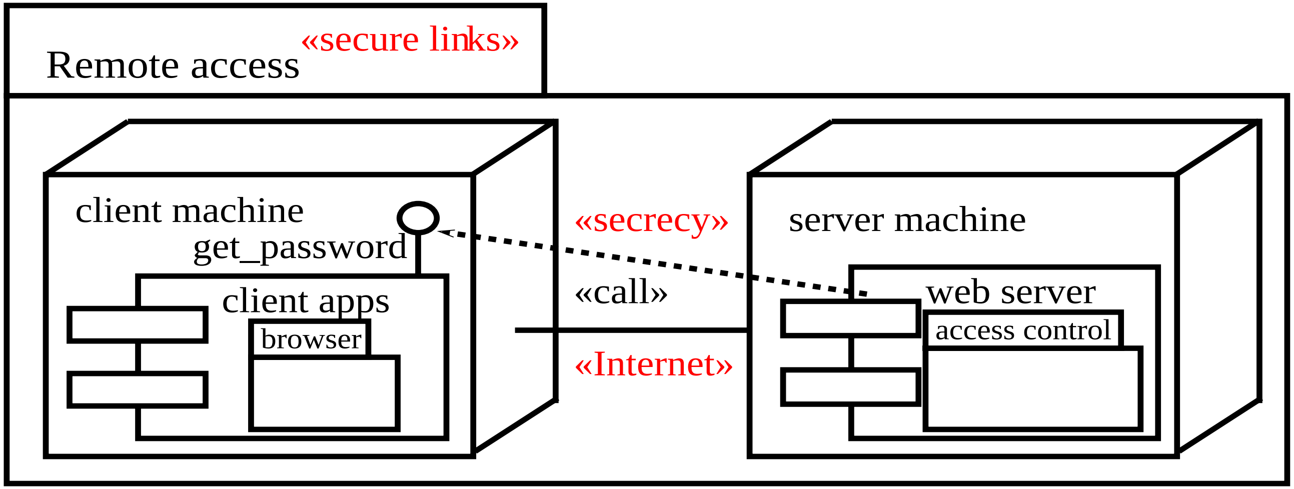
Ensures that physical layer meets security requirements on communication.

Constraint: for each dependency *d* with stereotype *s* ∈ {≪secrecy≫, ≪integrity≫} between components on nodes *n*≠*m*, have a communication link *l* between *n* and *m* with stereotype *t* such that

- if *s* = ≪secrecy≫: have read ∉ Threats *(t)*.
- if *s* = ≪integrity≫: have insert ∉ Threats *(t)A*.

A

Example ≪secure links≫



Given default adversary type, constraint for stereotype ≪secure links≫ violated: According to the $\text{Threats}_{\text{default}}(\text{Internet})$ scenario, ≪Internet≫ link does not provide secrecy against default adversary.

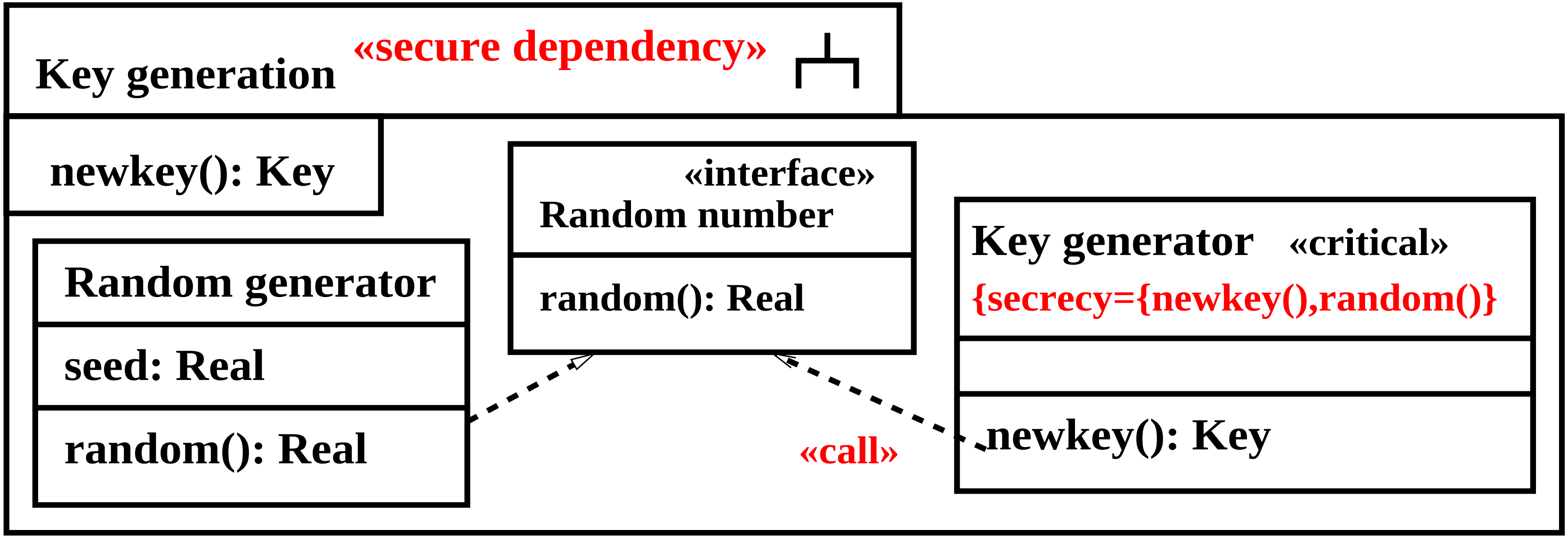
≪secure dependency≫

Ensure that ≪call≫ and ≪send≫ dependencies between components respect security requirements on communicated data given by tags {secrecy}, {integrity}.

Constraint: for ≪call≫ or ≪send≫ dependency from *C* to *D* (and similarly for {secrecy}):

- Msg in *D* is {secrecy} in *C* if and only if also in *D*.
- If msg in *D* is {secrecy} in *C,* dependency stereotyped ≪secrecy≫.

Example ≪secure dependency≫



Violates ≪secure dependency≫: Random generator and ≪call≫ dependency do not give security level for random() to key generator.

≪no down–flow≫

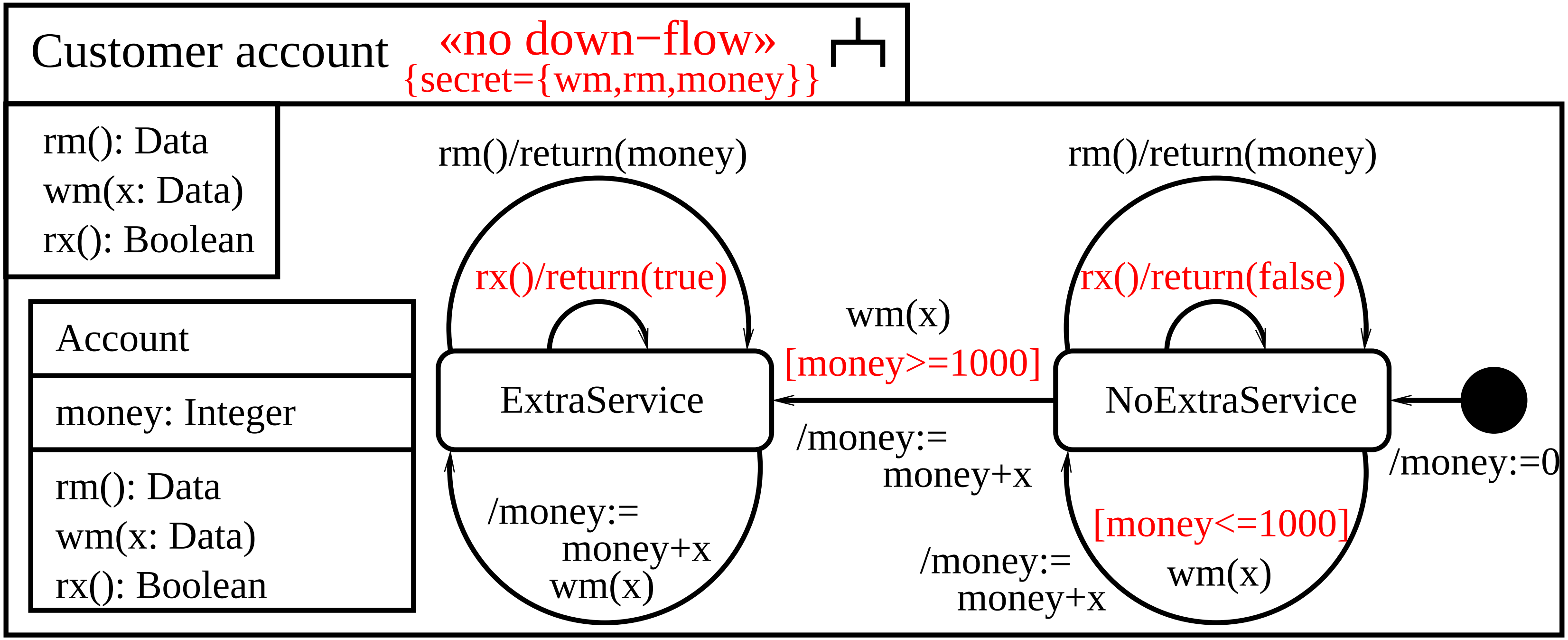
Enforce secure information flow.

Constraint:

Value of any data specified in {secrecy} may influence only the values of data also specified in {secrecy}.

Formalize by referring to formal behavioural semantics.

Example ≪no down-flow≫



≪no down–flow≫ violated: partial information on input of high wm() returned by non-high rx().

≪data security≫

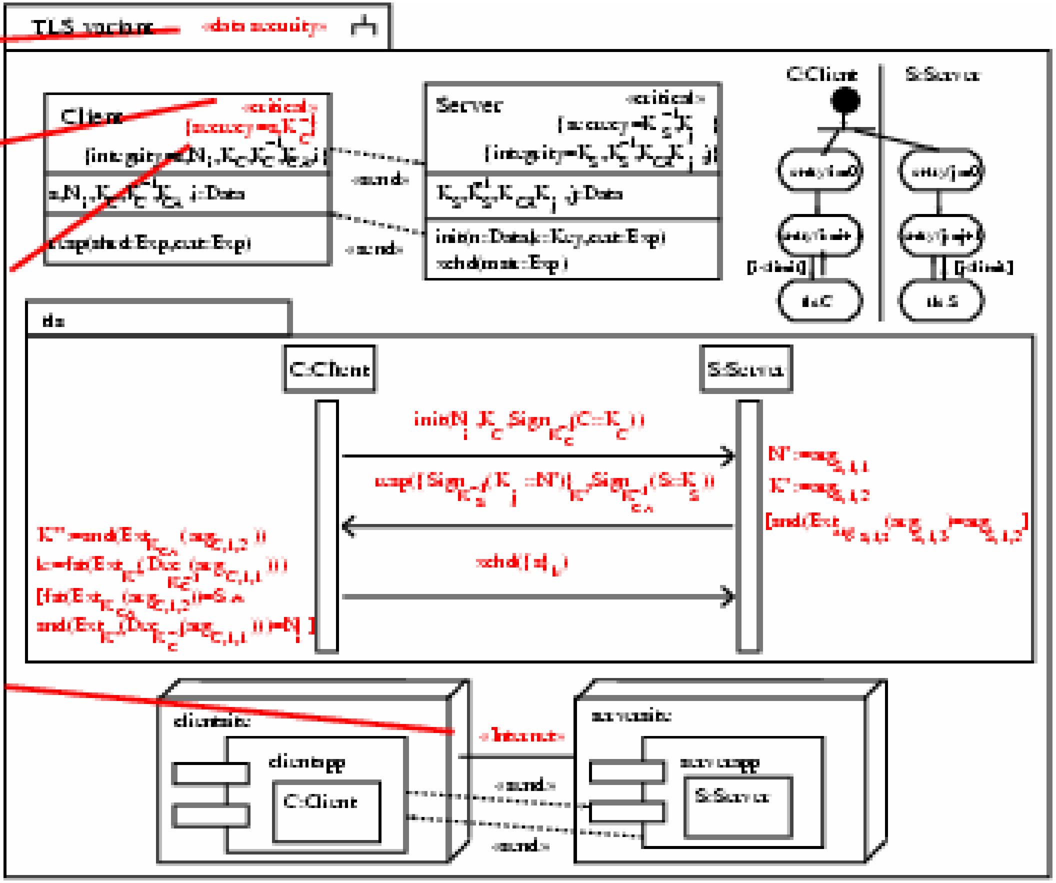
Security requirements of data marked ≪critical≫ enforced against threat scenario from deployment diagram.

Constraints:

Secrecy of {secrecy} data preserved.

Integrity of {integrity} data preserved.

Example «data security»



«data security»

«critical»

$\{secrecy = \{s, K_C^{-1}\}\}$

Variant of TLS (INFOCOM`99).

Violates {secrecy} of s against default adversary.

«Internet»

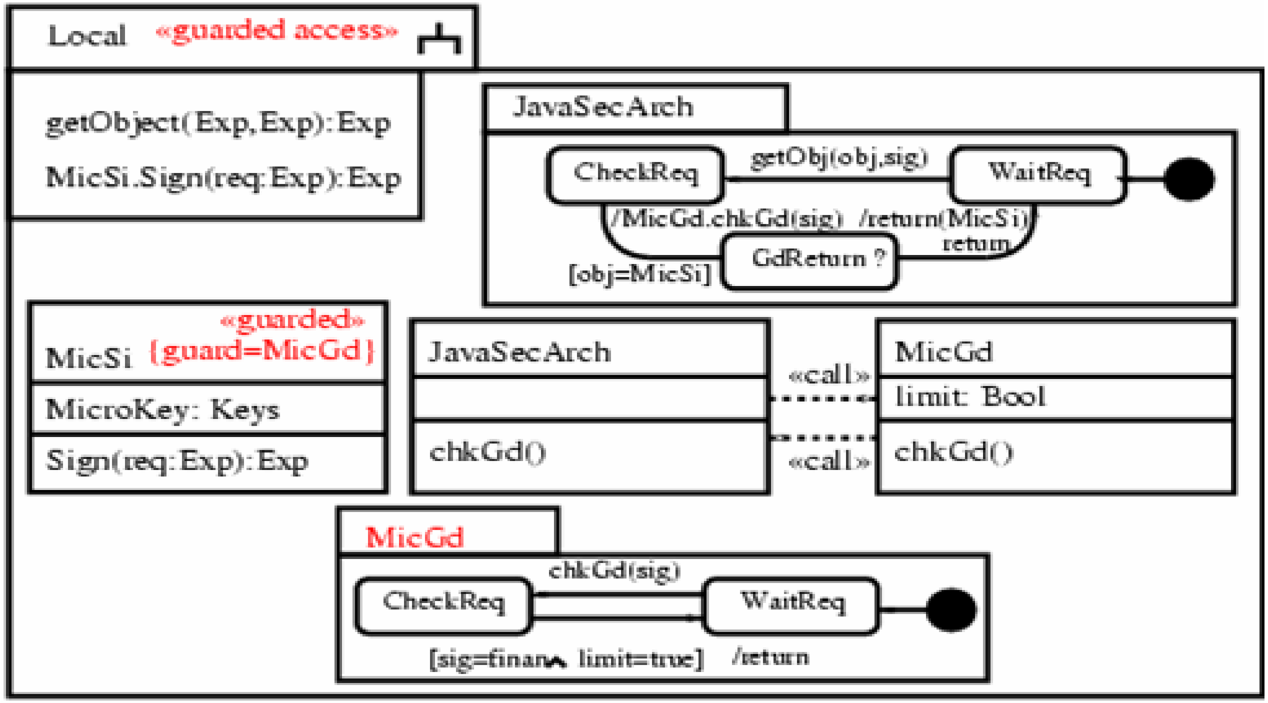
≪guarded access≫

Ensures that in Java, ≪guarded≫ classes only accessed through {guard} classes.

Constraints:

- References of ≪guarded≫ objects remain secret.
- Each ≪guarded≫ class has {guard} class.

Example ≪guarded access≫



Provides ≪guarded access≫:

Access to MicSi protected by MicGd.

Concepts covered by UMLsec

Security requirements: ≪secrecy≫,…

Threat scenarios: Use Threats$_{adv}$(ster).

Security concepts: For example ≪smart card≫.

Security mechanisms: E.g. ≪guarded access≫.

Security primitives: Encryption built in.

Physical security: Given in deployment diagrams.

Security management: Use activity diagrams.

Technology specific: Java, CORBA security.

Security Analysis

Model classes of adversaries.

May attack different parts of the system according to threat scenarios.

Example: insider attacker may intercept communication links in LAN.

To evaluate security of specification, simulate jointly with adversary model.

Tool-support: Concepts

Meaning of diagrams stated informally in (OMG 2001).

Possible ambiguities problem for

- tool support
- analysis of behavioral properties (such as security)

Use precise semantics for part of UML defined as pseudo-code. Include adversary model for simulation.

Tool-support: Technology

Commercial modelling tools: so far mainly syntactic checks and code-generation.

Goal: more sophisticated analysis; connection to analysis tools.

Several possibilities:

- General purpose language with integrated XML parser (Perl, …)
- Special purpose XML parsing language (XSLT, …)
- Data Binding (Castor; XMI: e.g. MDR)

Data-binding with MDR

Extracts data from XMI file into Java Objects, following UML 1.4 meta-model.

Access data via methods.

Advantage: No need to worry about XML.

Connection with analysis tool

Industrial CASE tool with UML-like notation: AUTOFOCUS (http://autofocus.informatik.tu-muenchen.de)

- verification
- code generation
- test-sequence generation

Connect UML tool to underlying analysis engine.

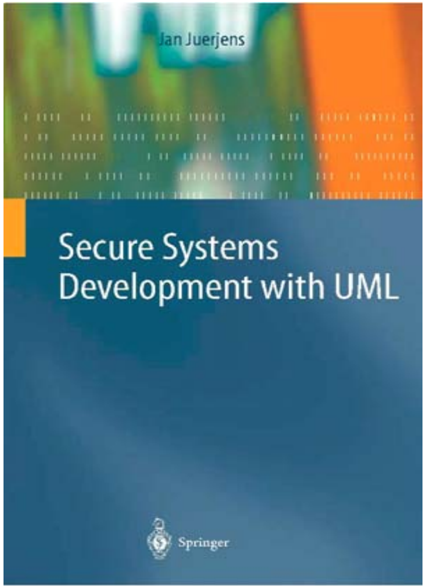
Applications

- Common Electronic Purse Specifications
- Analysis of multi-layer security protocol for web application of major German bank
- Analysis of SAP access control configurations for major German bank
- Risk analysis of critical business processes (for Basel II / KontraG)
- …

Some resources

Book: Jan Jürjens, Secure Systems Development with UML, Springer-Verlag, due 2003

Tutorial @ CSS'03, Cancun (Mexico), 19-21 May.



More information:

http://www.jurjens.de/jan

Finally

We are always interested in industrial challenges for our tools, methods, and ideas to solve practical problems.
More info: http://www.jurjens.de/jan

Contact me here or via Internet.

Thanks for your attention !