

UMLsec: Extending UML for Secure Systems Development

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** well-being of people and organizations.

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

Networked computers need to be **secure**.



Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development

2

Problems, Causes

Many **flaws** found in designs 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“.



Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development

3

Previous approaches

„Penetrate-and-patch“:

- **insecure**
- **disruptive**

Traditional formal methods: **expensive**.

- **training** people
- **constructing** formal specifications.



Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development

4

Goal: Security by design

Consider security

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



Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development

5

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.



Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development

6

Used fragment of UML

- Activity diagram
- Class diagram
- Sequence diagram
- Statechart diagram
- Deployment diagram
- Package
- Stereotypes, tags, constraints

Current: UML 1.4

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 7

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**

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 8

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.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 9

<<Internet>>, <<encrypted >>, <<LAN >>, ...

Kinds of communication **links** resp. system **nodes**.

For adversary type *A*, stereotype *s*, have set $Threats_A(s) \in \{delete, read, insert, access\}$ of actions that adversaries are capable of.

Default attacker:

Stereotype	Threats _{default()}
Internet	{delete, read, insert}
encrypted	{delete}
LAN	∅
smart cart	∅
POS device	∅

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 10

<<secure dependency>>

Ensure that <<call>> and <<send>> dependencies between components **respect** security requirements on communicated data given by tag {high}.

Constraint: for <<call>> or <<send>> dependency from *C* to *D*:

- Msg in *D* is {high} in *C* if and only if also in *D*.
- If msg in *D* is {high} in *C*, dependency stereotyped <<high>>.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 11

Example <<secure dependency>>

Violates <<secure dependency>>: Random generator and <<call>> dependency do not provide security level for random() required by key generator.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 12

<<no down-flow>>

Enforce secure **information flow**.

Constraint:

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

Formalize by referring to formal behavioural semantics.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 13

Example <<no down-flow>>

The diagram shows a state machine for a Bank account with the property <<no down-flow>> (high={wb,rb,balance}). It features two states: ExtraService and NoExtraService. Transitions are labeled with guards and actions. A transition from ExtraService to NoExtraService is guarded by `wb(x)[x<10000]` and has the action `/balance:=x`. A transition from NoExtraService to ExtraService is guarded by `wb(x)[x<10000]` and has the action `/balance:=x`. A self-loop on ExtraService is guarded by `wb(x)[x>=10000]` and has the action `/balance:=x`. A self-loop on NoExtraService is guarded by `wb(x)[x<10000]` and has the action `/balance:=x`. Both states have a receive event `rx()` and a return event `rb()/return(balance)`. The return event of NoExtraService is guarded by `/balance:=0`.

<<no down-flow>> violated: partial information about input of high `wb()` returned by non-high `rx()`.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 14

Formal semantics for UML: Why

Meaning of diagrams stated informally in (OMG 2001).

Possible ambiguities problem for

- tool support
- establishing behavioral properties (e.g. security)

Use mathematically precise semantics for used part of UML, especially for security requirements.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 15

Formal semantics for UML: How

Diagrams in **context** (using subsystems).
 Model **actions** and internal **activities** explicitly.

Message exchange between objects or components (incl. event dispatching).

For **UMLsec**: include **adversary** arising from threat scenario in deployment diagram.

Use Abstract State Machines (pseudo-code).

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 16

Security Analysis

Model classes of **adversaries**.

May **attack** different parts of the system in a specified way.

Example: **insider** attacker may intercept communication links in LAN.

To evaluate security of specification, execute jointly with adversary.

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 17

Connection with analysis tool

Commercial modelling tools: only **syntactic** checks and **code-generation**.

Current work: link to **verification** tools via **XMI**.

So far: industrial CASE tool with UML-like notation (AUTOFOCUS).

TUM Jan Jürjens, TU Munich: UMLsec: Extending UML for Secure Systems Development 18

Conclusion

Defined UML extension **UMLsec** for model-based development of security-critical systems.

Successfully used in security consulting.

Work in progress: tool-support.

Extend to other criticality requirements.



Resources

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

More information (also slides, papers etc.):

<http://www.jurjens.de/jan>

Thanks for your attention !

