Modelling and Analysing Resilience as a Security Issue within UML

Ricardo J. Rodríguez, José Merseguer and Simona Bernardi
{rjrodriguez, jmerse}@unizar.es, bernardi@di.unito.it

Universidad de Zaragoza
Zaragoza, Spain

Università di Torino
Torino, Italy

15th September 2010

SERENE'10: 2nd International Workshop on Software Engineering for Resilient Systems

- Introduction
- MARTE-DAM background
- Security Analysis and Modeling profile (SecAM)
- Advanced firewall running example
- Obtaining a formal model
- Experiments and discussion of the results
- Related work and conclusions

Introduction (I)

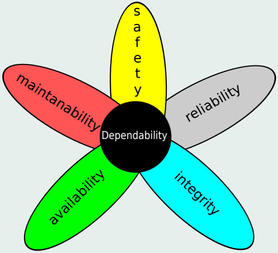
- Security requirement specification as a part of system modeling
- Broad and heterogeneous field (hardware issues, coding bugs...)
- Security requirements as non-functional properties (NFPs)
- Exploit the UML profiling capabilities to support their specification

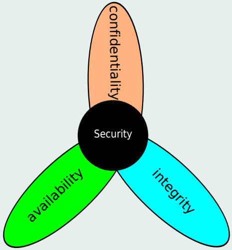
- UML: well-known solution and comprehensive modelling language
- Tailored for specific purposes: profiling
- The OMG standard MARTE profile
  - Performance and schedulability analysis for RT and embedded systems
- The Dependability and Analysis Modelling (DAM) profile
  - MARTE specialization to support dependability NFPs
- MARTE + DAM: performance and/on dependability requirements

$\rightarrow$ missing security aspects

Introduction (II)

- Relation between dependability-security



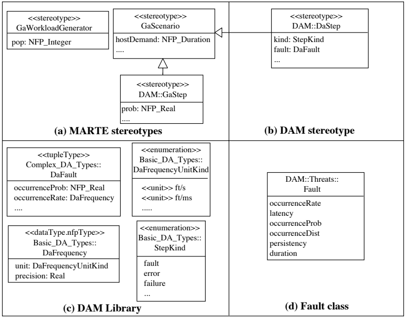


- Security specification can be included in the MARTE-DAM framework
- MARTE-DAM: stereotypes and tagged to express NFPs
  - Attached to those UML model elements they affect
- Security Analysis and Modelling (SecAM) profile
  - Allow the expression of security NFPs

Background

MARTE: Modelling and Analysis of RT Embedded systems

- UML *lightweight* extension
- Provides support for schedulability and performance analysis
- NFPs with VSL (Value Specification Language) syntax
- Design model element extending its semantic



MARTE-DAM

- DAM stereotypes specialise MARTE stereotypes
- MARTE NFP types
  - *value*
  - *expr* (VSL expression)
  - *source* (*req*, *est*, *assm*)

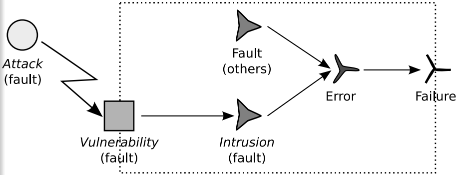
SecAM profile (I): *Resilience* package (1)

Domain model definition

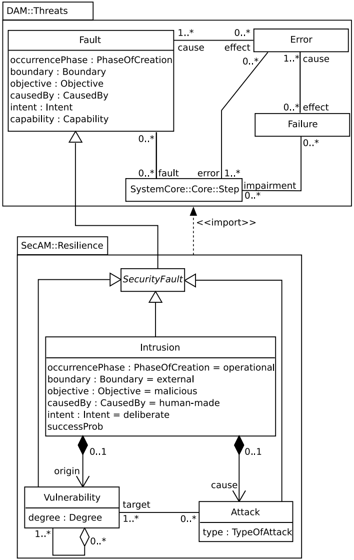
- Comprehensive modelling of security issues
- Domain model for each relevant security aspects
  - e.g., confidentiality, resilience or integrity
- In this work: *Resilience* package

Threats

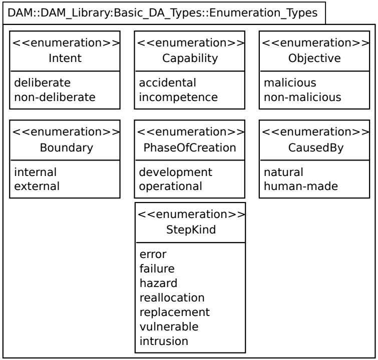
- From dependability:
  - Fault → Error → Failure
- From security:
  - Attack → Vulnerability → Intrusion
- AVI as a refinement of FEF



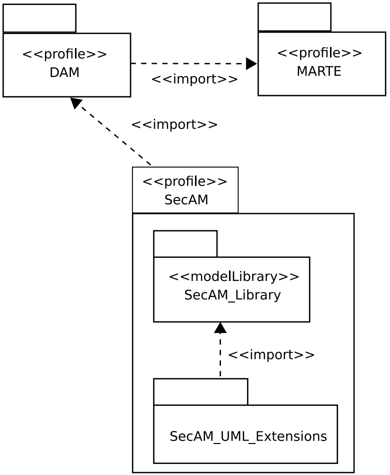
SecAM profile (I): *Resilience* package (2)

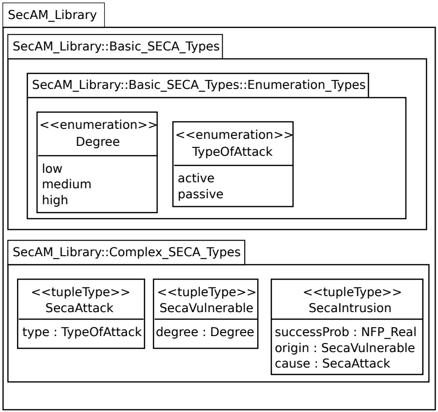


- *Fault* class from *DAM::Threats*: extension with new attributes
- OCL to restrict source of errors



SecAM profile (II): building the profile (1)





SecAM profile (II): building the profile (2)

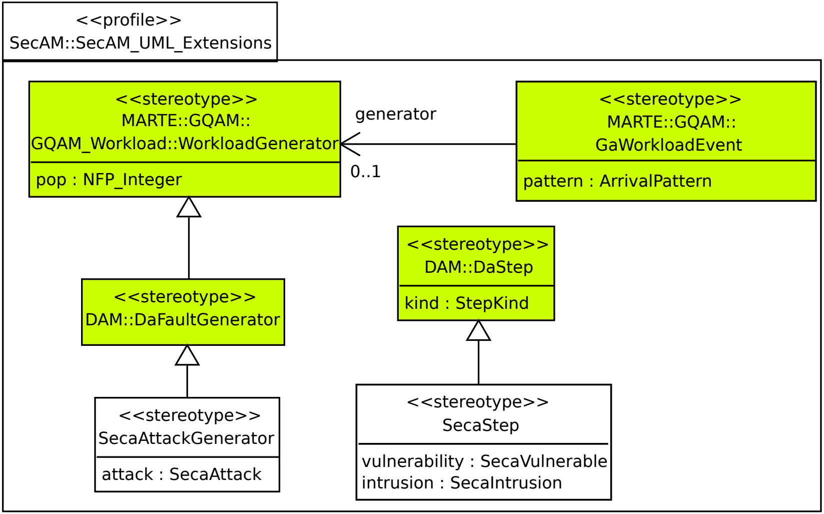


Figure: SecAM UML extensions

Example (I): system physical view and class diagram

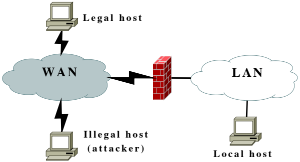


Figure: System physical view

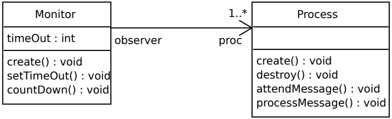


Figure: Class diagram

- How to use SecAM from a modeler viewpoint
- Advanced firewall
  - Exposed to attacks → vulnerable
  - Attend messages from WAN and forwarded them to LAN
  - Critical information systems (e.g. MAFTIA, CRUTIAL, OASIS)
- Includes a monitor
  - Tamper-proof embedded system → invulnerable
  - Its mission: to check firewall processes and to clean up those hung

Example (II): UML state-charts (1)

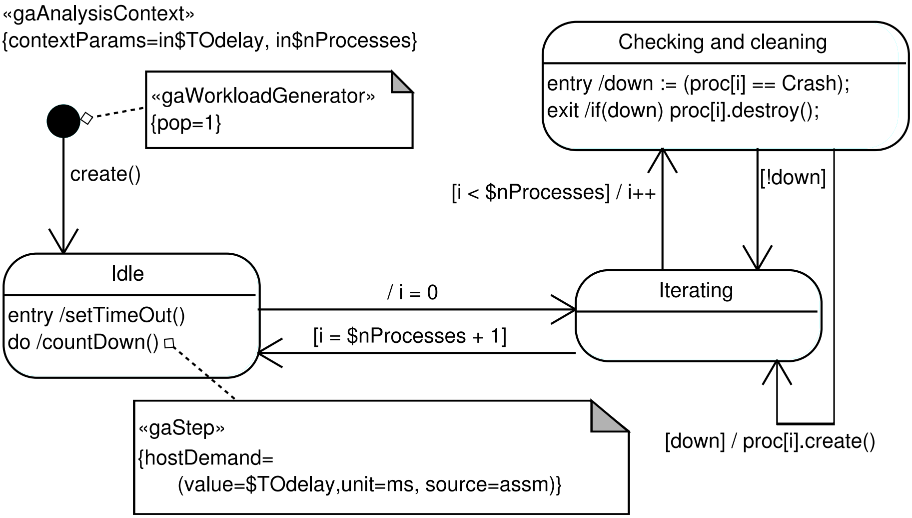


Figure: Monitor state-chart diagram.

Example (II): UML state-charts (2)

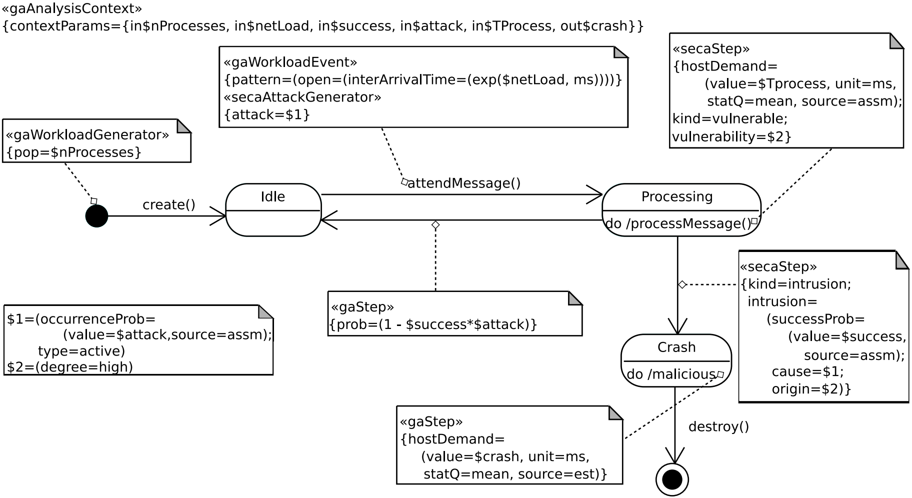


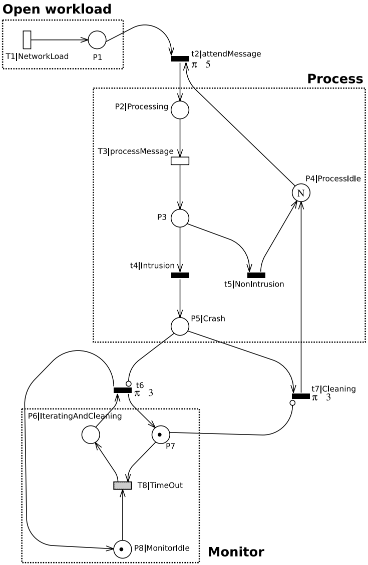
Figure: Process state-chart diagram.

Obtaining a formal model (I): Conversion of UML-SC

- Translation proposed by Merseguer et al. (*WODES'02*)
- Given for performance analysis purposes
- ArgoSPE tool: UML-SC annotated with SPT (precursor of MARTE)
- General ideas:
  - SC simple state → PN place
  - Entry and exit actions → immediate transitions
  - Do-activity actions → timed transitions
  - Stochastic resolution of transition conflicts
- Communication via events → PN places modelling event mailboxes

- Working out the PN to incorporate DAM and SecAM annotations
- Open workload: manually produced
- Simplified the subnets → gaining readability

Obtaining a formal model (II): Obtained DSPN



Place	Initial marking	Value
P4|Idle	*nProcesses*	6

Transition	Parameter (type)	Value(s)
T1|NetworkLoad	1/*netload* (rate)	0.01, 0.05, 0.1/ms
T3|processMessage	1/*Tprocess* (rate)	0.2/ms
T8|TimeOut	*TOdelay* (delay)	1, 100ms
t4|Intrusion	$attack \cdot success$ (weight)	
t5|NonIntrusion	$1 - attack \cdot success$ (weight)	

Parameter	Values
attack	[0.01 . . . 0.5]
success	[0.01 . . . 0.5]

Description of the experiments

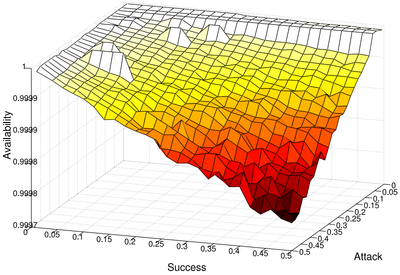
Availability

- At DSPN model level:

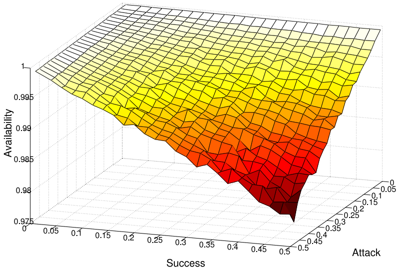
$$\frac{MTTF}{MTTF + MTTDI} = 1 - \frac{E[P5|Crash]}{N} \qquad (1)$$

  - $MTTF$: Mean Time To Failure
  - $MTTDI$: Mean Time To Detect an Intrusion
  - $E[P_i]$: mean number of tokens in place P_i
  - $P5|Crash$: unavailable state of the process
- Under different assumptions:
  - Three types of network loads: low, high, very high (0.01, 0.05, 0.1/ms)
  - Two types of time-out durations: short, long (1, 100 ms)
  - Probabilities of attacks and successful attacks from 1% up to 50%

Results (I): under low workload

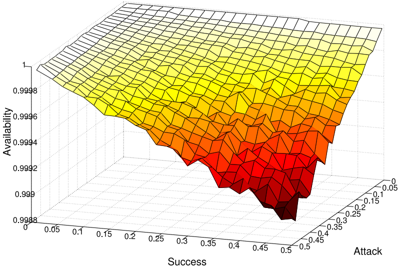


(a) short time-out

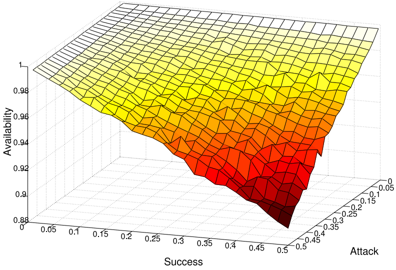


(b) long time-out

Results (II): under high workload

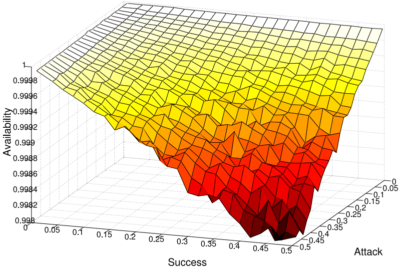


(a) short time-out

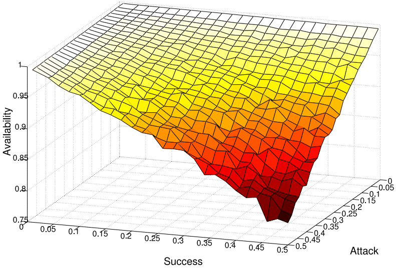


(b) long time-out

Results (III): under very high workload



(a) short time-out



(b) long time-out

Discussion

Availability

- Inverse proportion to probability of attacks and of successful attacks
- Decreasing factor: sensitive to the network workload and monitor time-out assumptions
    - Higher for higher workloads and for longer time-out duration (e.g., 0.021% in case of low network workload and short time-out duration, 20.9% when very high network workload and long time-out duration)
- Incoming messages are potential attack carriers → frequency of attacks increases from low to very high network workload → higher availability decreasing factor
- Short time-out duration → promptly detection → higher availability
- Isolated hills close to 100% (low workload, short time-out)
    - Due to simulation accuracy (their height is lower than 0.01%)
- False alarms (i.e., time-out expires and no process is crashed)
    - Do not provoke side effects in the system

Related work and conclusions (I)

Related work

- SecureUML (*T. Lodderstedt et al.*)
  - Just focused on annotating static UML design models
- UMLsec (*J. Jürjens*)
  - Not worry on influence on the throughput of the system

  Both approaches focus on the design phase and allow model-checking
- Other work close (*D. C. Petriu et al.*)
  - Not focussed on giving a unified framework
- Dependability and SPNs
  - *A. E. Rugina et al.*
    - Exclusively for the dependability field
    - Very bound to AADL (Architecture Analysis & Design Language)
  - Several works of *Bondavalli et al.*
    - Dependability attributes in early design phases of the system
    - Construct a Timed PN using graph transformation techniques in structural UML diagrams

Related work and conclusions (II)

Conclusions

- Proposal profile $\subset$ MARTE-DAM profile
- Analysis of relevant dependability-security aspects
- Considering the system performance characteristics
  - e.g., to measure the real impact of introducing more security layers

Future work

- Tools supporting the SecAM approach
  - Reuse of existing tools for UML and MARTE
- Effort focused on the security analysis on top of existing tool sets
- Extend SecAM adding more security fields to its domain
  - Easy fit: SecAM-MARTE-DAM fit already done
- ...

Modelling and Analysing Resilience as a Security Issue within UML

Ricardo J. Rodríguez, José Merseguer and Simona Bernardi

{rjrodriguez, jmerse}@unizar.es, bernardi@di.unito.it

Universidad de Zaragoza
Zaragoza, Spain

Università di Torino
Torino, Italy

15th September 2010

SERENE'10: 2nd International Workshop on Software Engineering for Resilient Systems