AVATAR-TTool
A SysML Environment for the Proof of Safety and Security Properties

Ludovic Apvrille

Telecom ParisTech
ludovic.apvrille@telecom-paristech.fr

April 2011

Outline

Introduction
- Model-Driven Engineering
- TTool
- DIPLODOCUS

AVATAR: From Requirements to Prototyping
- Introduction
- Requirements
- Analysis
- Design

Conclusions, References
- Conclusions
- References

Outline

Introduction
- Model-Driven Engineering
- TTool
- DIPLODOCUS

AVATAR: From Requirements to Prototyping

Conclusions, References

Model Driven Engineering

Definition

- Process based on abstract representations for a given domain (domain model)
- Notion of patterns
- Should enhance team working and exchanges between clients / system-level teams and development teams

UML and SysML

- MDE is commonly based on UML profiles
  - Profiles defined at OMG's (e.g., SPT, MARTE, SysML)
  - Profiles defined by tool vendors (e.g. in Rhapsody, Artisan)
  - User-defined and company-defined models

A Multi Profile Platform: TTool

TTool

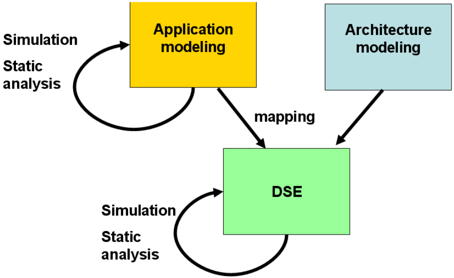
- Open-source toolkit mainly developed by Telecom ParisTech
- 8 UML profiles
  - DIPLODOCUS, AVATAR
- Support from academic (e.g. INRIA) and industrial partners (e.g., Freescale)

Main ideas

- Lightweight, easy-to-use toolkit
- Simulation with model animation
- Formal proof at the push of a button

The DIPLODOCUS UML Profile

- Partitioning
  - Finding the best SW / HW function repartition
- Follows the Y-Chart approach
- Ultra-fast simulation and verification
  - Up to 100 times the real-time execution
  - Variable simulation coverage



Outline

Introduction

AVATAR: From Requirements to Prototyping

- Introduction
- Requirements
- Analysis
- Design

Conclusions, References

AVATAR in a Nutshell

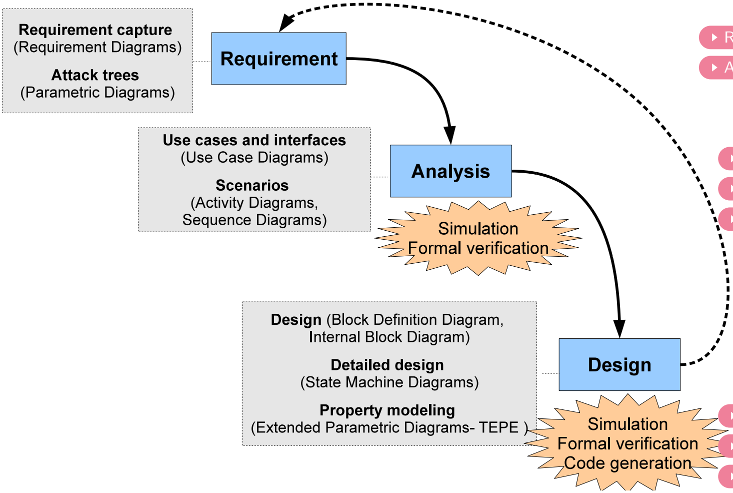
Former contribution: TURTLE (1999)

- Formally defined UML profile (RT-LOTOS, UPPAAL)
- Enhanced with requirement (2006), analysis (2003) and deployment phases (2006)

AVATAR (2010)

- SysML environment supporting all methodological phases
- Graphical capture of properties
- Integrated simulation
- Safety and **security** proofs at the push of a button
- C-POSIX code generation
- TURTLE is now deprecated!

Methodology



Requirement capture
(Requirement Diagrams)

Attack trees
(Parametric Diagrams)

Requirement

Use cases and interfaces
(Use Case Diagrams)

Scenarios
(Activity Diagrams,
Sequence Diagrams)

Analysis

Simulation
Formal verification

Design (Block Definition Diagram,
Internal Block Diagram)

Detailed design
(State Machine Diagrams)

Property modeling
(Extended Parametric Diagrams- TEPE)

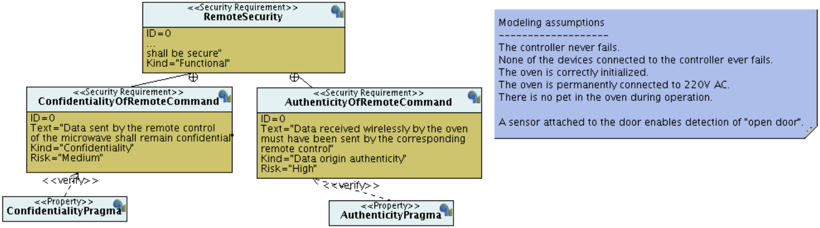
Design

Simulation
Formal verification
Code generation

- Requirement Capture
- Attack Trees
- Use cases
- Functioning modes
- Scenarios
- Design
- Detailed Design
- Property modeling
- Simulation
- Formal verification
- Prototyping

Requirement Capture

- SysML Requirement Diagrams
- Specialization for security-related requirements (e.g., confidentiality, privacy, etc.)
- Modeling assumptions inside notes



<<Security Requirement>>
RemoteSecurity
ID=0
...
shall be secure"
Kind="Functional"

<<Security Requirement>>
ConfidentialityOfRemoteCommand
ID=0
Text="Data sent by the remote control of the microwave shall remain confidential"
Kind="Confidentiality"
Risk="Medium"

<<Security Requirement>>
AuthenticityOfRemoteCommand
ID=0
Text="Data received wirelessly by the oven must have been sent by the corresponding remote control"
Kind="Data origin authenticity"
Risk="High"

<<Property>>
ConfidentialityPragma

<<Property>>
AuthenticityPragma

Modeling assumptions

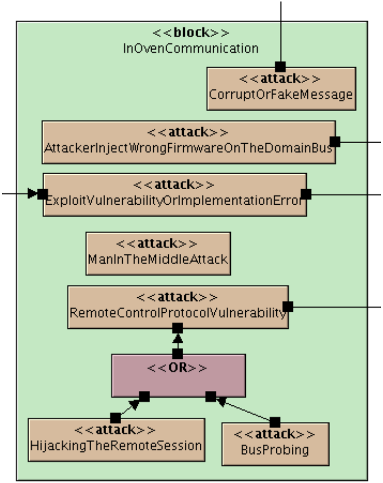
The controller never fails.
None of the devices connected to the controller ever fails.
The oven is correctly initialized.
The oven is permanently connected to 220V AC.
There is no pet in the oven during operation.

A sensor attached to the door enables detection of "open door".

Back to methodology

Attack Trees

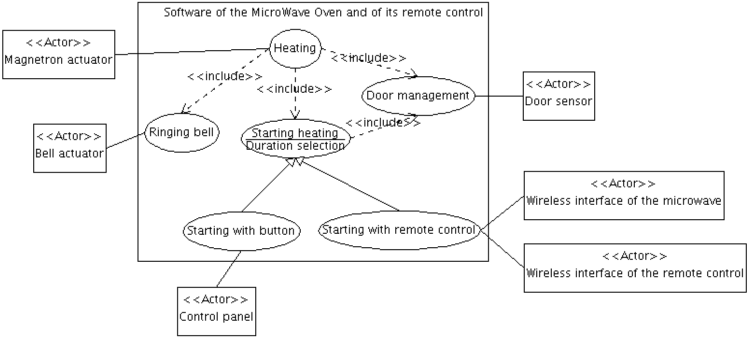
- Represent all possible attacks on the system
  - And relations between those attacks: OR, AND, SEQUENCE, BEFORE, AFTER, etc.
- SysML Parametric Diagrams



▸ Back to methodology

Use Cases

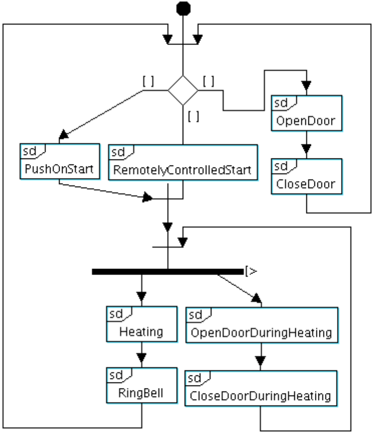
- System boundary, actors, and main functions (use cases) provided by the system
    - And high-level links between use cases
- SysML Use Case Diagrams



Back to methodology

Main Functioning Modes

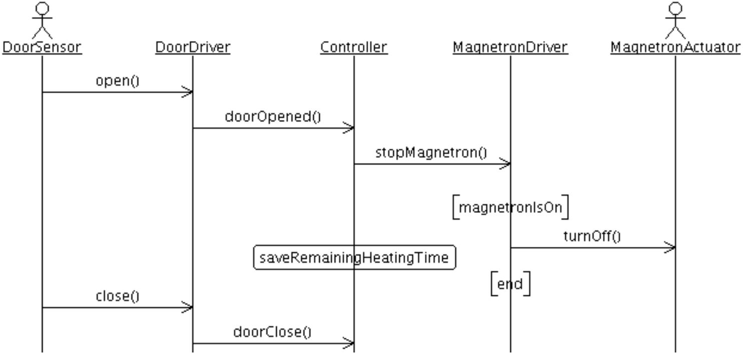
- Identify various system functioning modes
- Represent relations between functioning modes
  - Sequence, choice, preemption, parallel
- (Slightly extended) SysML Activity Diagrams



▸ Back to methodology

Scenarios

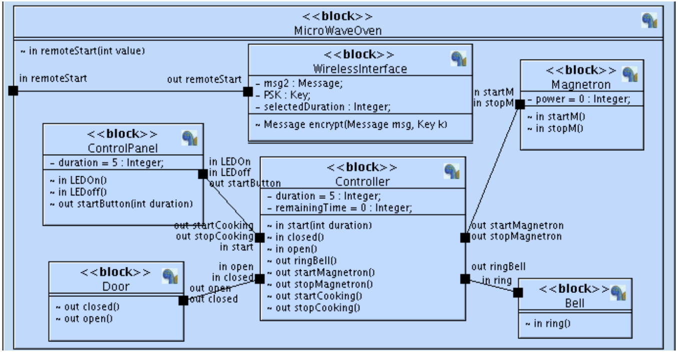
- Identify a specific trace of the system
- SysML Sequence Diagrams



Back to methodology

Design: Architecture

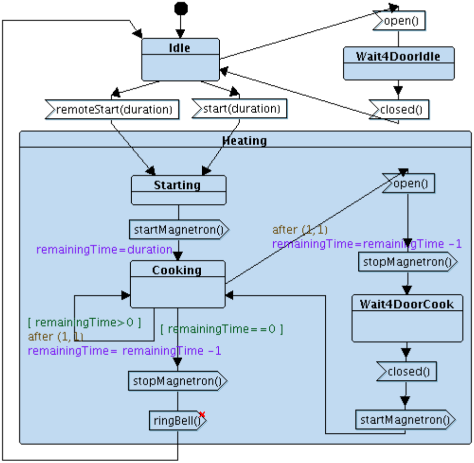
- SysML Block Definition and Internal Block Diagrams
- Block = attributes, methods, in/out signals, behaviour



Back to methodology

Detailed Design

- Block's behaviour is described in terms of SysML State Machine Diagrams
- Non deterministic choices
- Non deterministic temporal operators

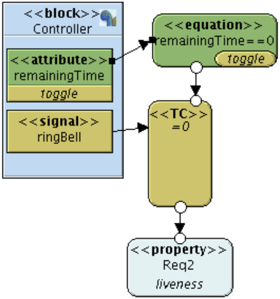


Back to methodology

Property Modeling

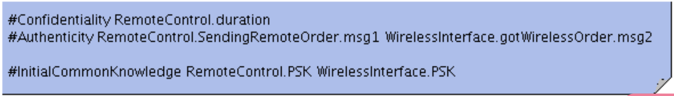
Safety properties

- Customized Parametric Diagrams (TEPE)



Security properties

- Based on basic pragmas
  - Confidentiality of a block attribute
  - Authenticity of interconnected block signals

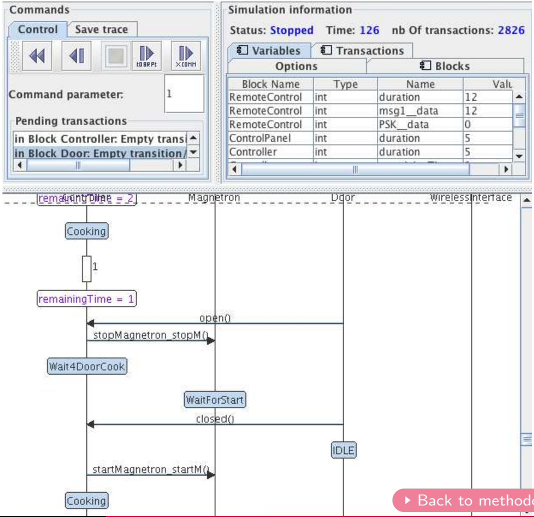


```
#Confidentiality RemoteControl.duration
#Authenticity RemoteControl.SendingRemoteOrder.msg1 WirelessInterface.gotWirelessOrder.msg2

#InitialCommonKnowledge RemoteControl.PSK WirelessInterface.PSK
```

▸ Back to methodology

Simulation

- Integrated in TTool
- Model animation
- Breakpoints, step, backstep, reset, introspection of block variables, etc.
- Simluation traces are displayed as SysML Sequence Diagrams



Commands

Control | Save trace

Command parameter: 1

Pending transactions

in Block Controller: Empty transi
in Block Door: Empty transition/

Simulation information

Status: Stopped Time: 126 nb Of transactions: 2826

Variables | Transactions | Options | Blocks

Block Name	Type	Name	Valu
RemoteControl	int	duration	12
RemoteControl	int	msg1__data	12
RemoteControl	int	PSK__data	0
ControlPanel	int	duration	5
Controller	int	duration	5

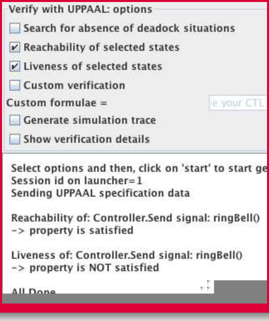
Back to methodology

Formal Verification

- Push button approach, both for safety and security properties!

Safety properties

- UPPAAL based



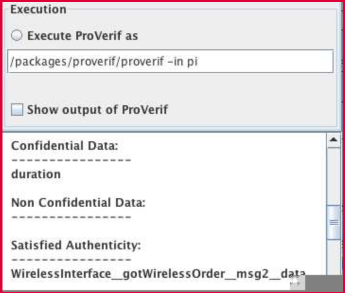
Verify with UPPAAL: options

□ Search for absence of deadlock situations
☑ Reachability of selected states
☑ Liveness of selected states
□ Custom verification
Custom formulae = e your CTL
□ Generate simulation trace
□ Show verification details

```
Select options and then, click on 'start' to start ge
Session id on launcher=1
Sending UPPAAL specification data

Reachability of: Controller.Send signal: ringBell()
-> property is satisfied

Liveness of: Controller.Send signal: ringBell()
-> property is NOT satisfied

All Done
```

Security properties

- ProVerif based



Execution

○ Execute ProVerif as
/packages/proverif/proverif -in pi

□ Show output of ProVerif

```
Confidential Data:
----------------
duration

Non Confidential Data:
----------------

Satisfied Authenticity:
----------------
WirelessInterface__gotWirelessOrder__msg2__data
```

▸ Back to methodology

Prototyping

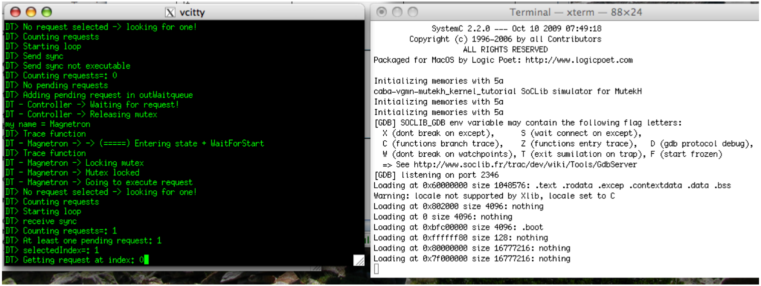
- C-POSIX code generation from design
  - Compiled and executed on localhost (e.g. Windows, MacOS, Linux)
  - Prototyped with the SocLib + MutekH platform

SoClib/MutekH

- SoCLib = virtual prototyping platform (LIP6)
  - Many microprocessors supported (MIPS, ARM, PowerPC, etc.)
  - Embedded Operating System = MutekH
  - Transaction Level Modeling or Cycle Accurate Bus Accurate
- Code is first cross-compiled for the selected microprocessor
- Then, execution of: SocLib, MutekH, application
  - Performance metrics (traces)
  - Easy debugging (gdb: step by step execution, etc.)

Back to methodology

Prototyping with SoCLib



```
DT> No request selected -> looking for one!
DT> Counting requests
DT> Starting loop
DT> Send sync
DT> Send sync not executable
DT> Counting requests=: 0
DT> No pending requests
DT> Adding pending request in outWaitqueue
DT - Controller -> Waiting for request!
DT - Controller -> Releasing mutex
my name = Magnetron
DT> Trace function
DT - Magnetron -> -> (=====) Entering state + WaitForStart
DT> Trace function
DT - Magnetron -> Locking mutex
DT - Magnetron -> Mutex locked
DT - Magnetron -> Going to execute request
DT> No request selected -> looking for one!
DT> Counting requests
DT> Starting loop
DT> receive sync
DT> Counting requests=: 1
DT> At least one pending request: 1
DT> selectedIndex=: 1
DT> Getting request at index: 0
```

```
             SystemC 2.2.0 --- Oct 10 2009 07:49:18
        Copyright (c) 1996-2006 by all Contributors
                    ALL RIGHTS RESERVED
Packaged for MacOS by Logic Poet: http://www.logicpoet.com

Initializing memories with 5a
caba-vgmn-mutekh_kernel_tutorial SoCLib simulator for MutekH
Initializing memories with 5a
Initializing memories with 5a
[GDB] SOCLIB_GDB env variable may contain the following flag letters:
  X (dont break on except),      S (wait connect on except),
  C (functions branch trace),    Z (functions entry trace),   D (gdb protocol debug),
  W (dont break on watchpoints), T (exit sumilation on trap), F (start frozen)
  => See http://www.soclib.fr/trac/dev/wiki/Tools/GdbServer
[GDB] listening on port 2346
Loading at 0x60000000 size 1048576: .text .rodata .excep .contextdata .data .bss
Warning: locale not supported by Xlib, locale set to C
Loading at 0x802000 size 4096: nothing
Loading at 0 size 4096: nothing
Loading at 0xbfc00000 size 4096: .boot
Loading at 0xffffff80 size 128: nothing
Loading at 0x80000000 size 16777216: nothing
Loading at 0x7f000000 size 16777216: nothing
```

▸ Back to methodology

Outline

Introduction

AVATAR: From Requirements to Prototyping

Conclusions, References

- Conclusions
- References

Conclusions

TTool = Open-source solution for MDE

- Academic and industrial involvement on TTool
- Many success stories (e.g., Freescale, EVITA)
- Used for teaching activities

AVATAR

- Integrated simulation
- Formal proof at the push of a button
  - Safety proof (UPPAAL)
  - Security proof (ProVerif)
- Easy-to-use virtual prototyping

Website, Publications

TTool website: http://labsoc.comelec.enst.fr/ttool/

- Under google: "TTool"
- How to install TTool, tutorials, these slides, etc.

Papers

- Gabriel Pedroza, Daniel Knorreck, Ludovic Apvrille, "AVATAR: A SysML Environment for the Formal Verification of Safety and Security Properties", The 11th IEEE Conference on Distributed Systems and New Technologies, Paris, France, May 2011.
- Daniel Knorreck, Ludovic Apvrille, Pierre de Saqui-Sannes, "TEPE: A SysML Language for Time-Constrained Property Modeling and Formal Verification", Proceedings of the Third IEEE International Workshop UML and Formal Methods (UMLFM'2010), Shanghai, China, November, 2010.
- L. Apvrille, W. Muhammad, R. Ameur-Boulifa, S. Coudert and R. Pacalet, "A UML-based Environment for System Design Space Exploration", 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS'2006), Nice, France, December 2006