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

Outline

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

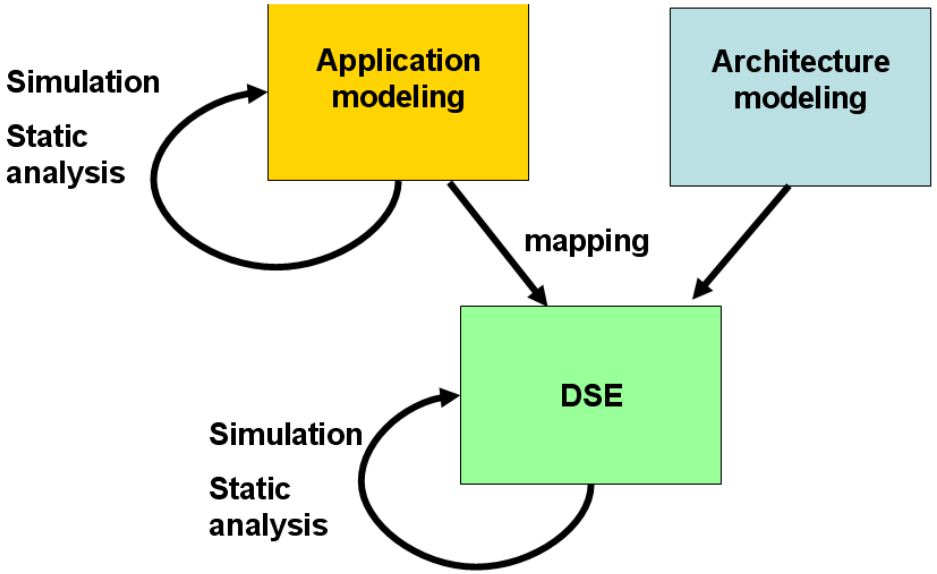
- Open-source toolkit mainly developed by Telecom ParisTech
- 8 UML profiles
  - DIPLODOCUS, AVATAR
- Support from academic (e.g. INRIA) and industrial partners (e.g., Freescale)

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

AVATAR in a Nutshell

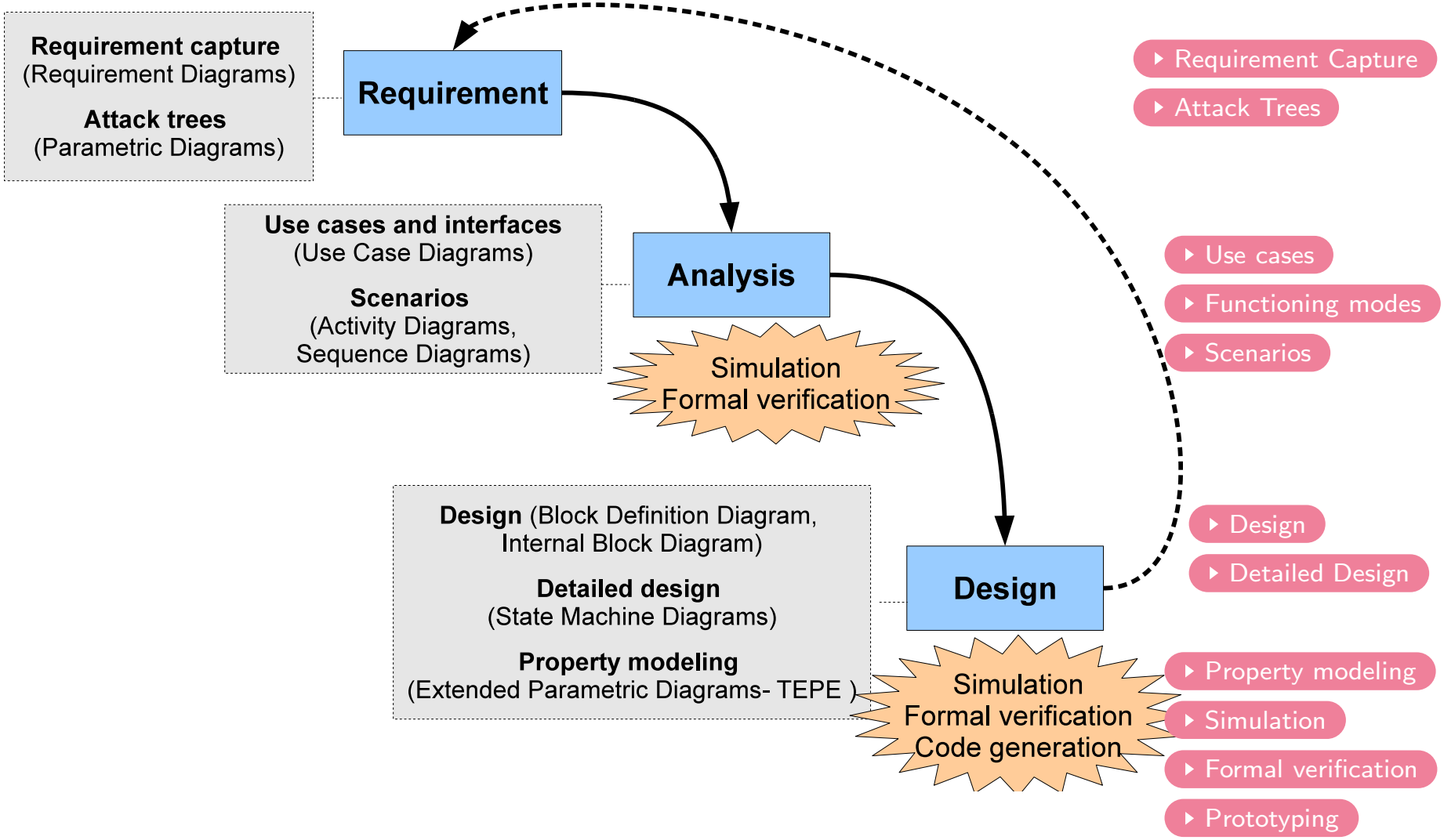
Former contribution: TURTLE (1999)

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

AVATAR (2010)

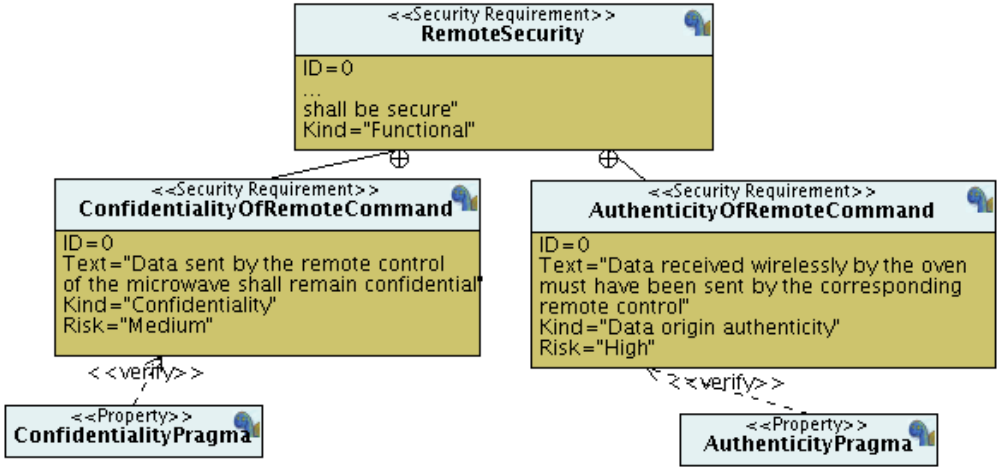
- SysML environment supporting all methodological phases
- Graphical capture of properties
- Integrated simulation
- Safety and **security** proofs at the push of a button
- C-POSIX code generation
- TURTLE is now deprecated!

Methodology



Requirement Capture

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



Modeling assumptions

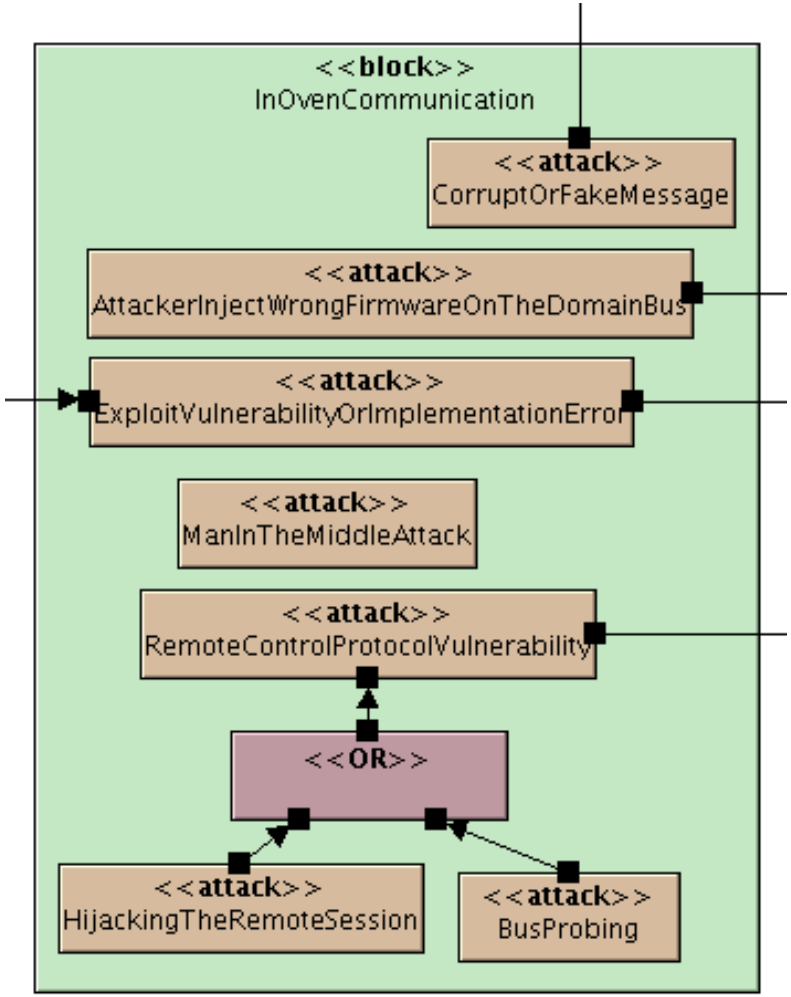
The controller never fails.
None of the devices connected to the controller ever fails.
The oven is correctly initialized.
The oven is permanently connected to 220V AC.
There is no pet in the oven during operation.

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

Back to methodology

Attack Trees

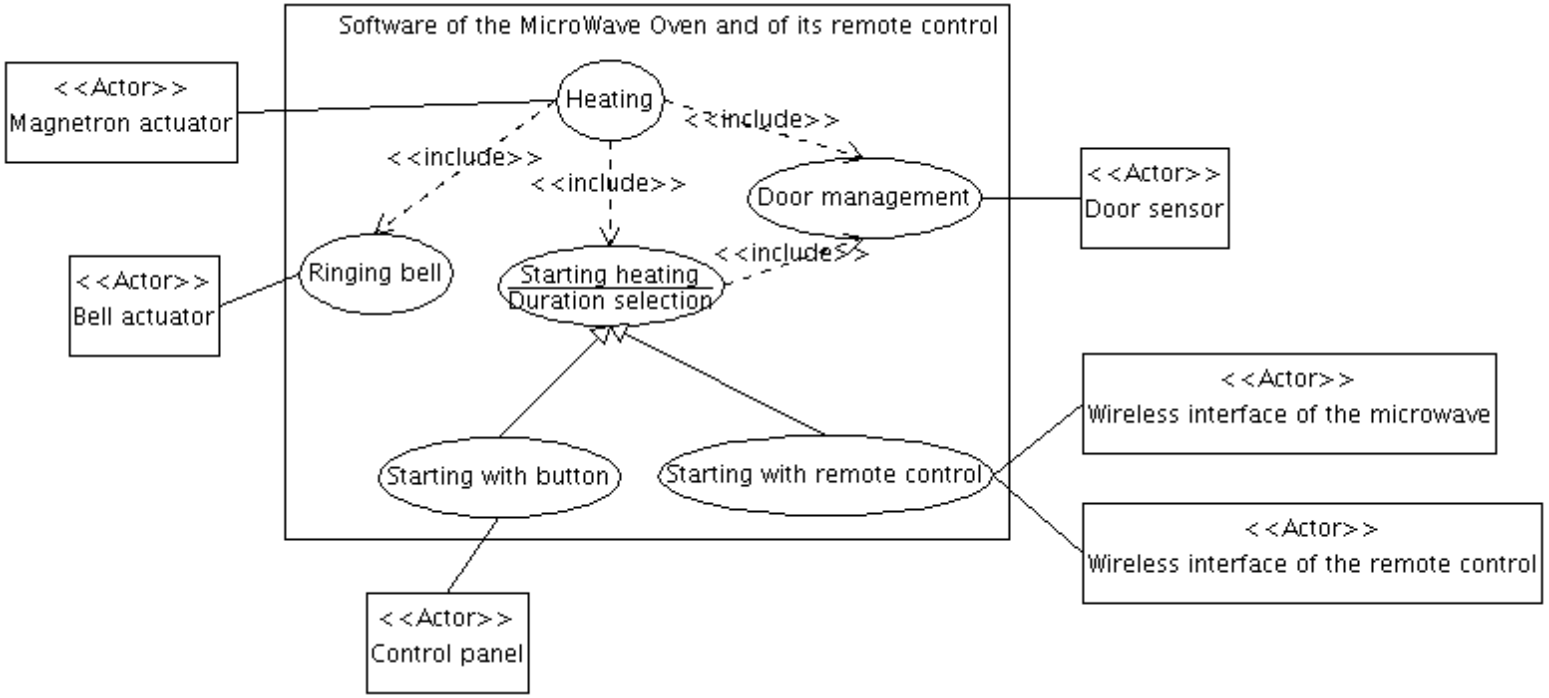
- Represent all possible attacks on the system
  - And relations between those attacks: OR, AND, SEQUENCE, BEFORE, AFTER, etc.
- SysML Parametric Diagrams



Back to methodology

Use Cases

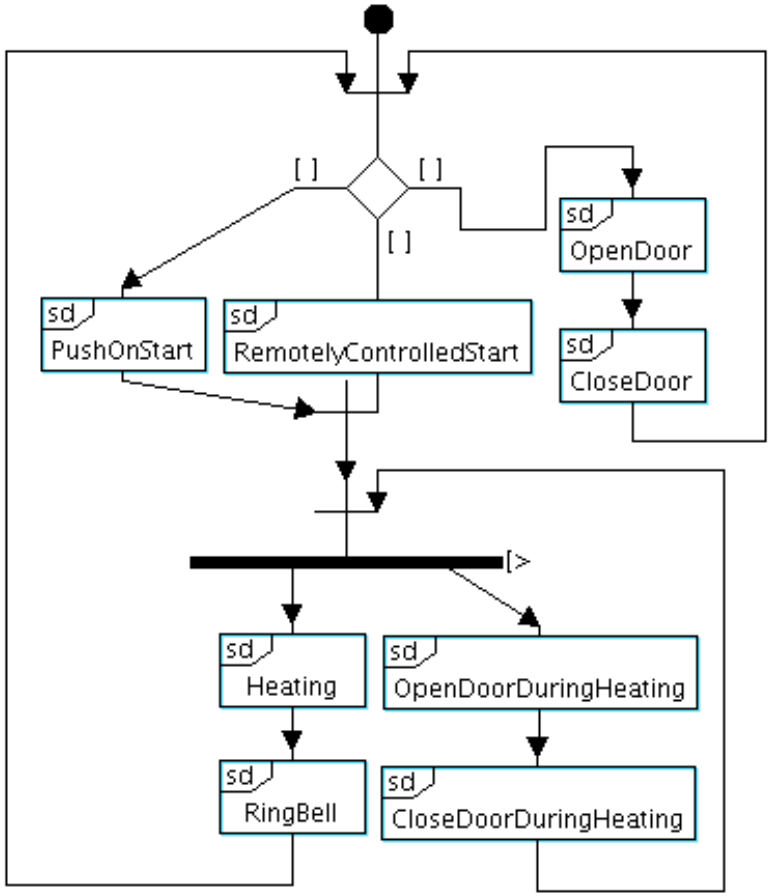
- System boundary, actors, and main functions (use cases) provided by the system
  - And high-level links between use cases
- SysML Use Case Diagrams



Back to methodology

Main Functioning Modes

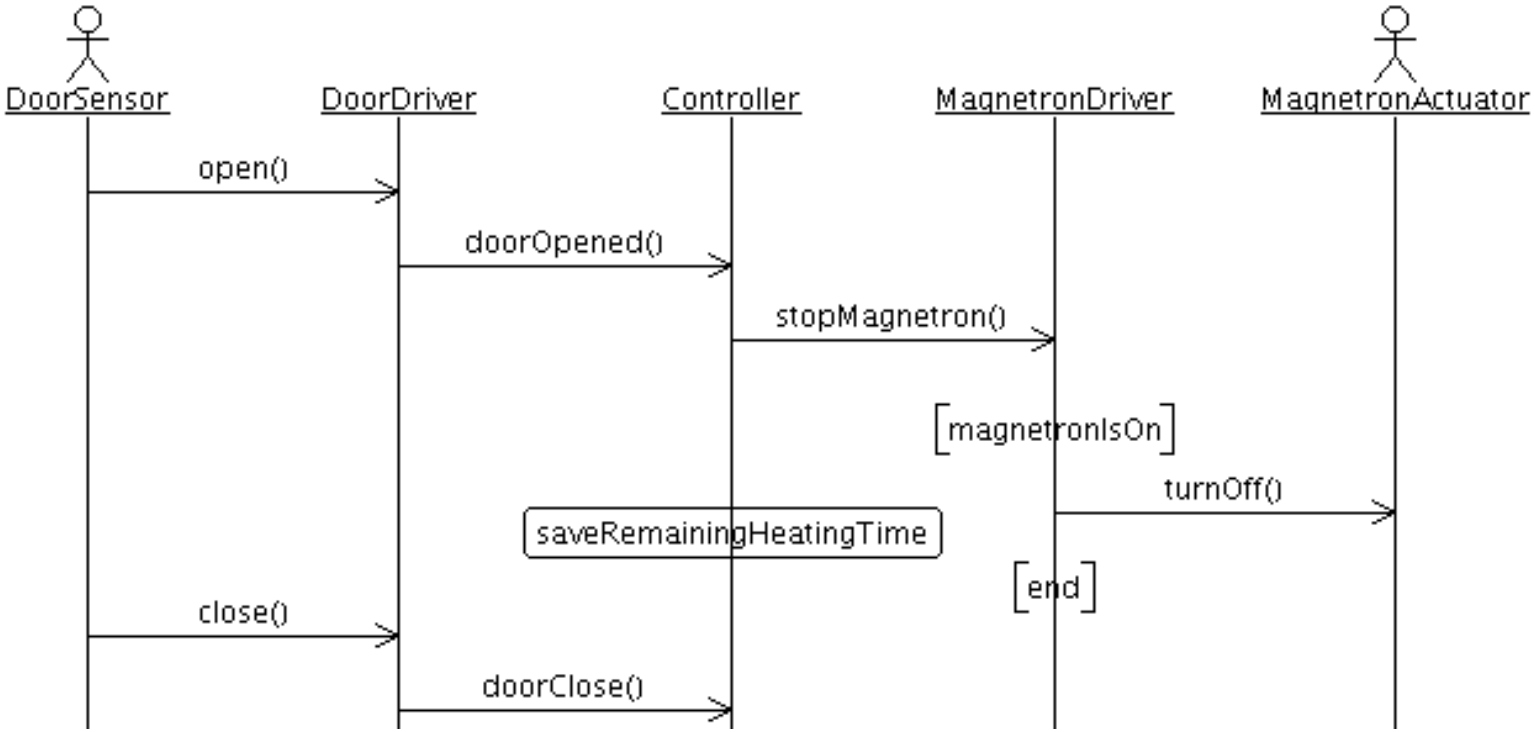
- Identify various system functioning modes
- Represent relations between functioning modes
  - Sequence, choice, preemption, parallel
- (Slightly extended) SysML Activity Diagrams



Back to methodology

Scenarios

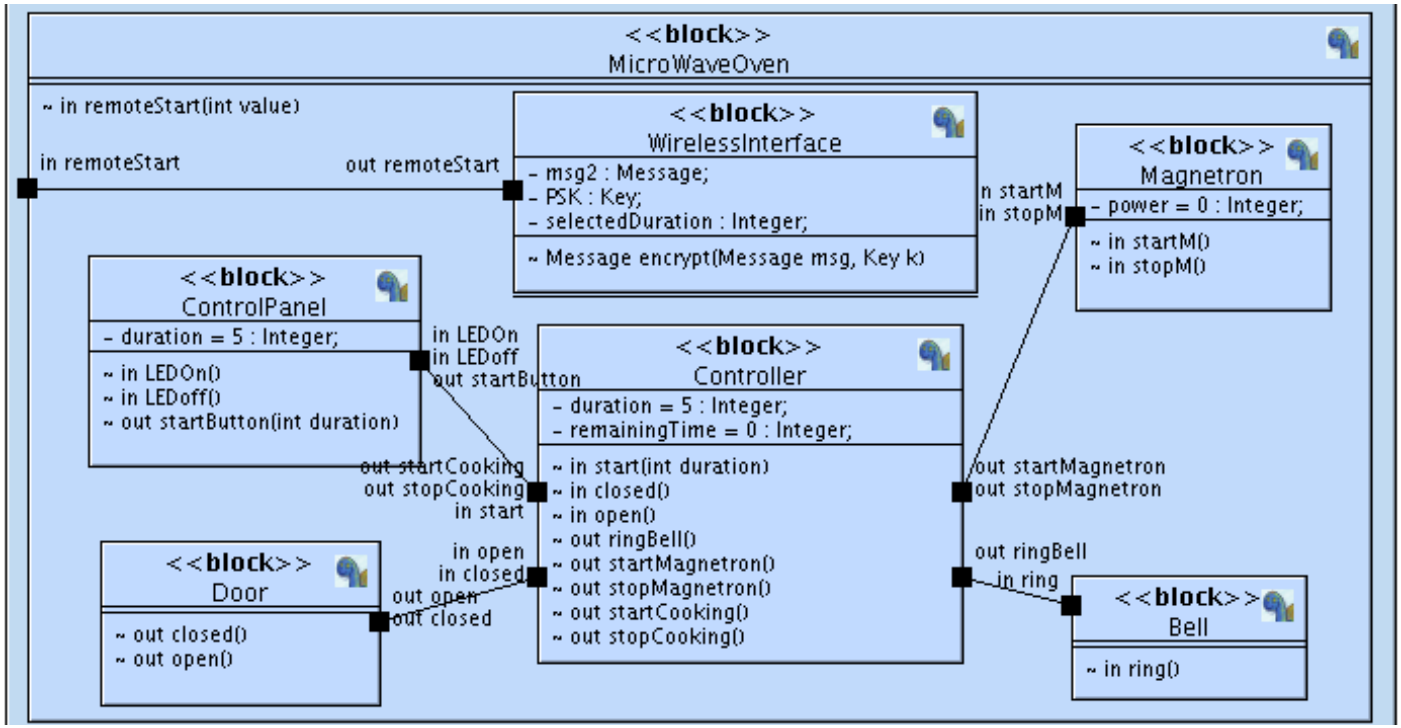
- Identify a specific trace of the system
- SysML Sequence Diagrams



Back to methodology

Design: Architecture

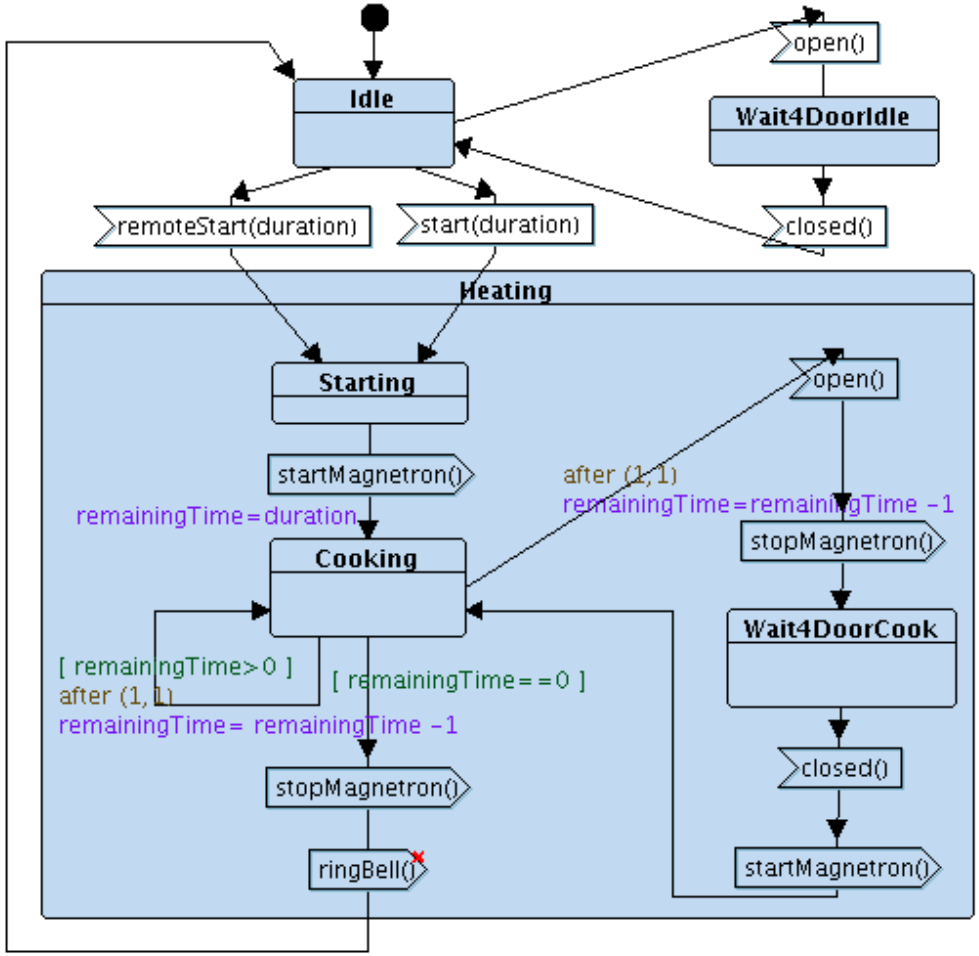
- SysML Block Definition and Internal Block Diagrams
- Block = attributes, methods, in/out signals, behaviour



Back to methodology

Detailed Design

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

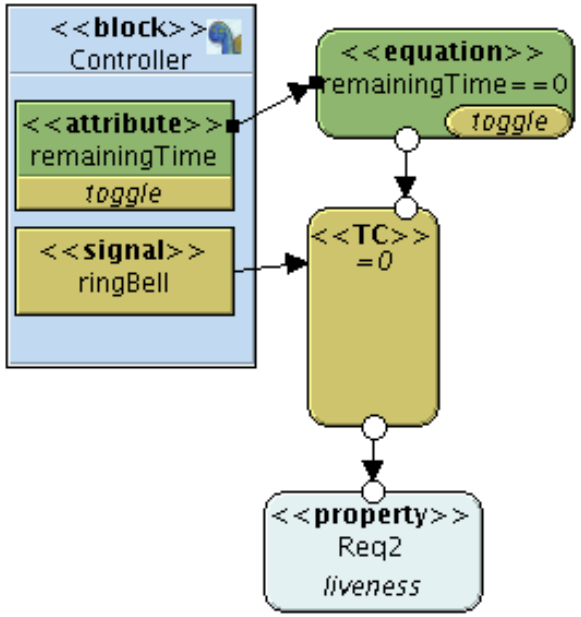


Back to methodology

Property Modeling

Safety properties

- Customized Parametric Diagrams (TEPE)



Security properties

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

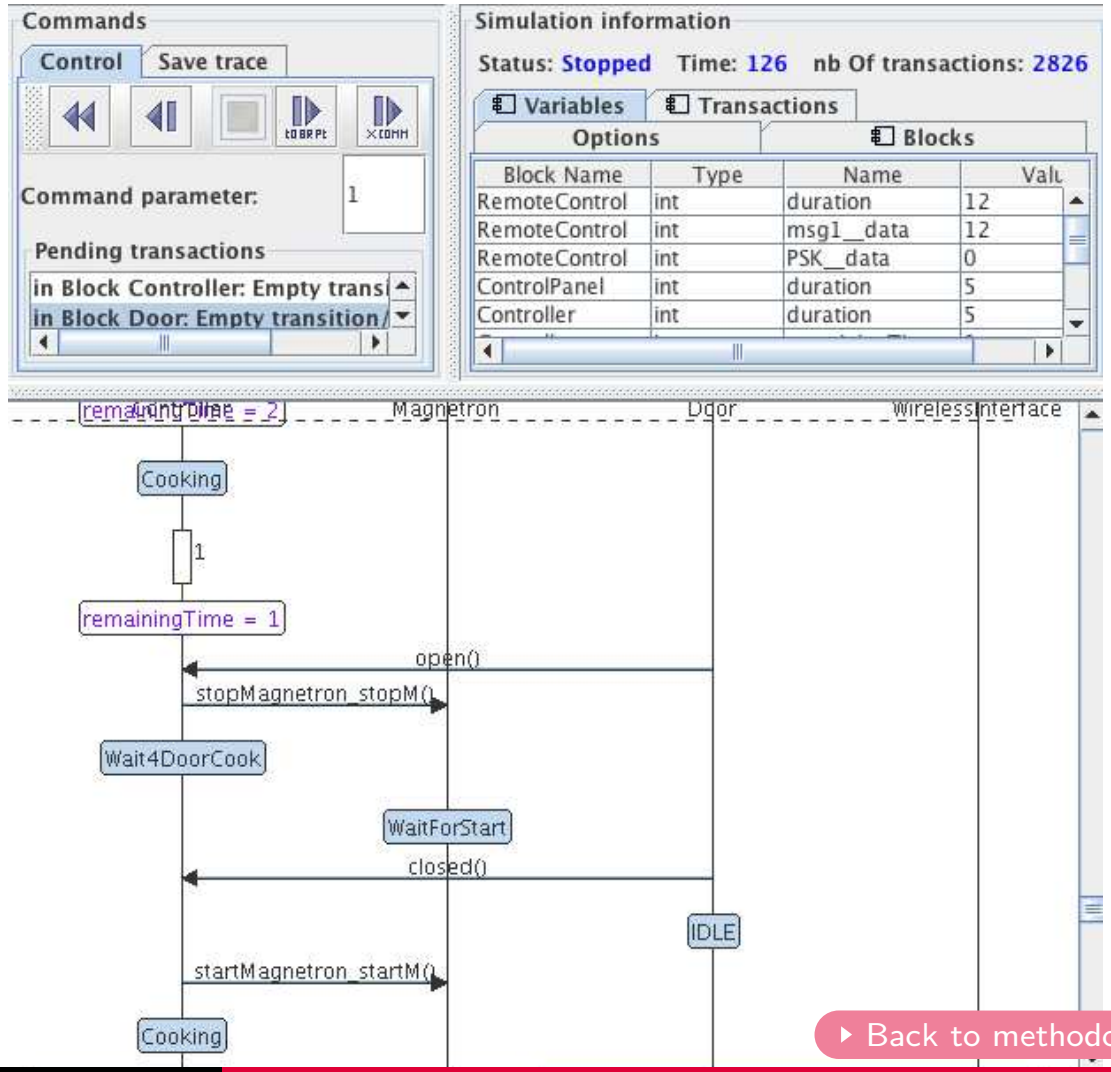
```
#Confidentiality RemoteControl.duration
#Authenticity RemoteControl.SendingRemoteOrder.msg1 WirelessInterface.gotWirelessOrder.msg2

#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



Back to methodology

Formal Verification

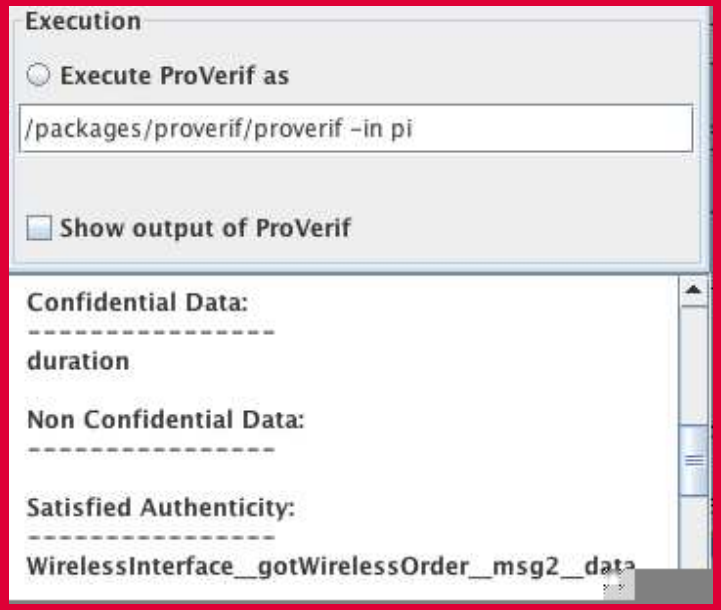
- Push button approach, both for safety and security properties!

Safety properties

- UPPAAL based

Security properties

- ProVerif based



Back to methodology

Prototyping

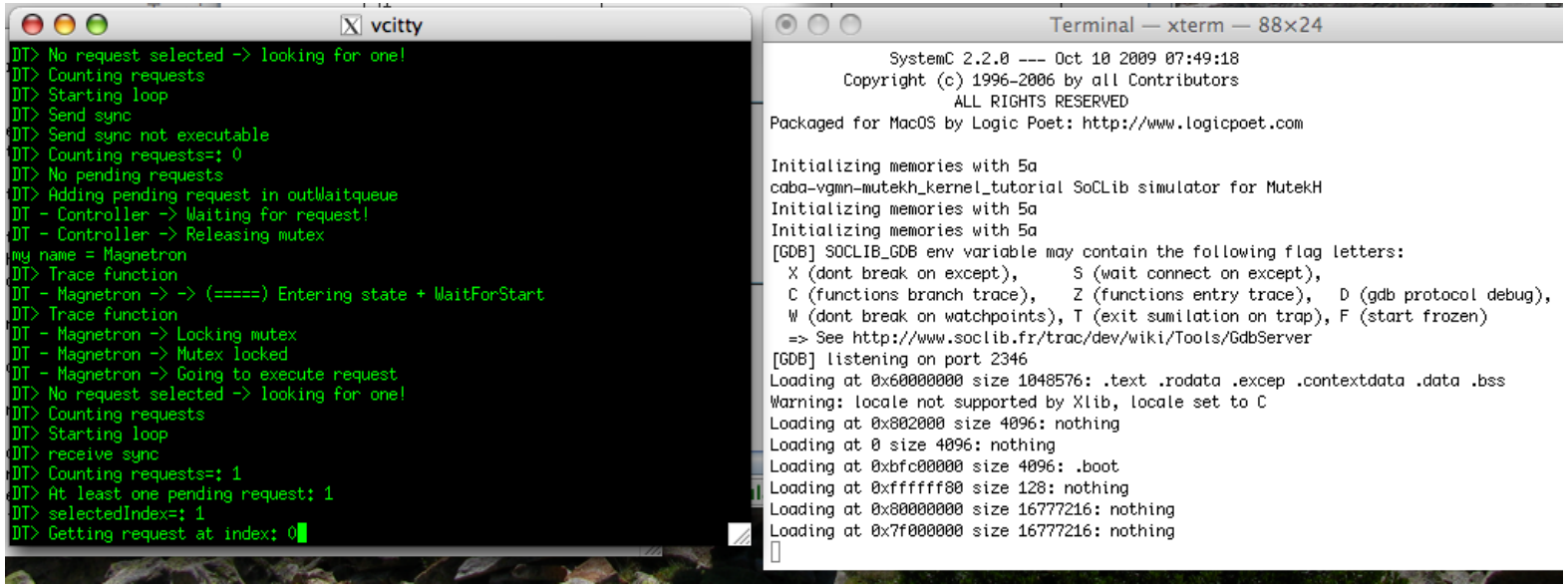
- C-POSIX code generation from design
  - Compiled and executed on localhost (e.g. Windows, MacOS, Linux)
  - Prototyped with the SocLib + MutekH platform

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



Back to methodology

Outline

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