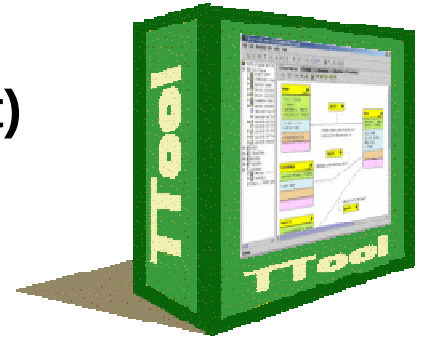
TURTLE
Four Weddings and a Tutorial

L. Apvrille, P. de Saqui-Sannes

ERTS²
Toulouse, France
May 20, 2010

Rationale

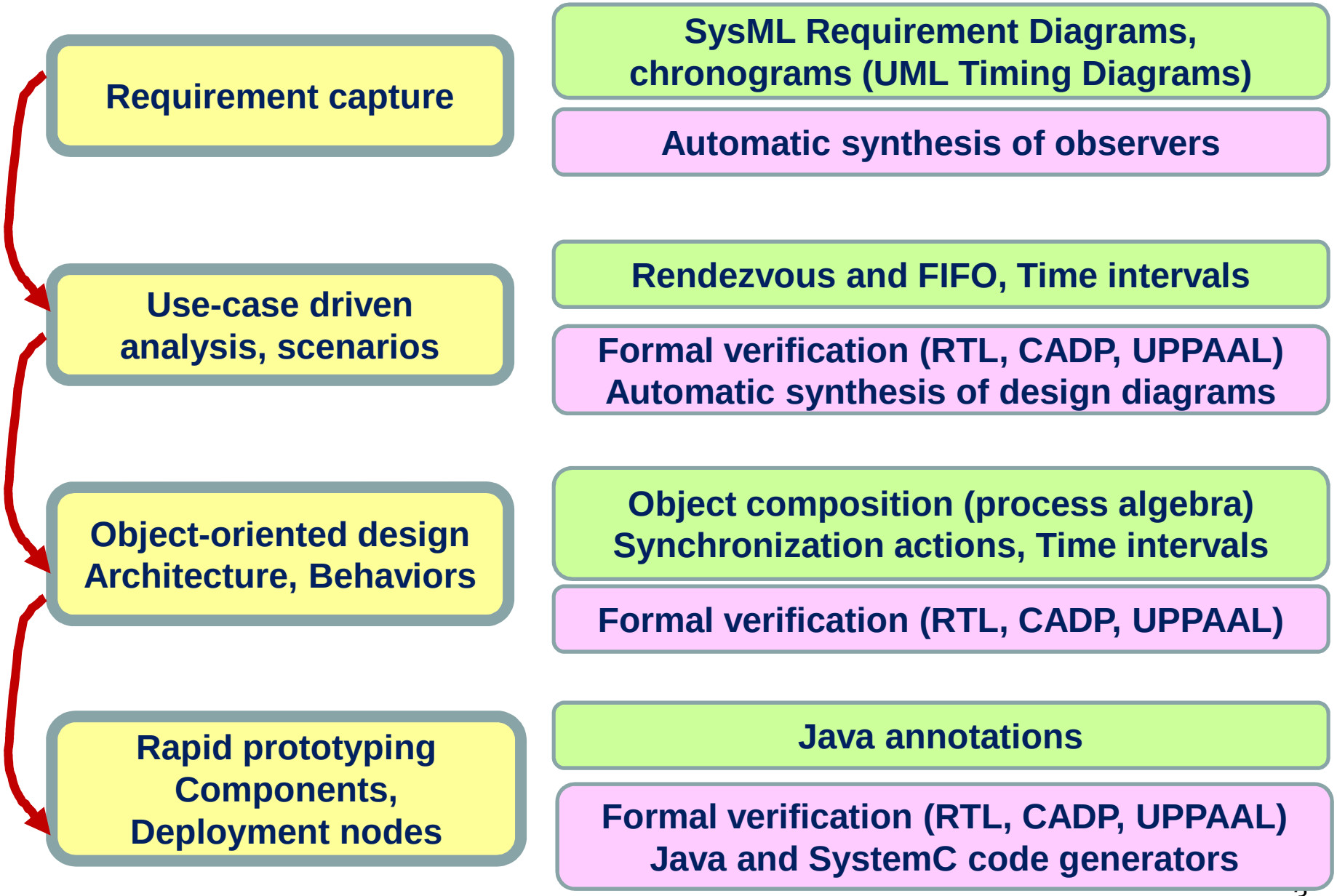
- **Verification-centric modeling of real-time and distributed systems**
  - Real-time UML profile grounded in formal methods
  - Press-button approach hiding the inner workings of verification tools



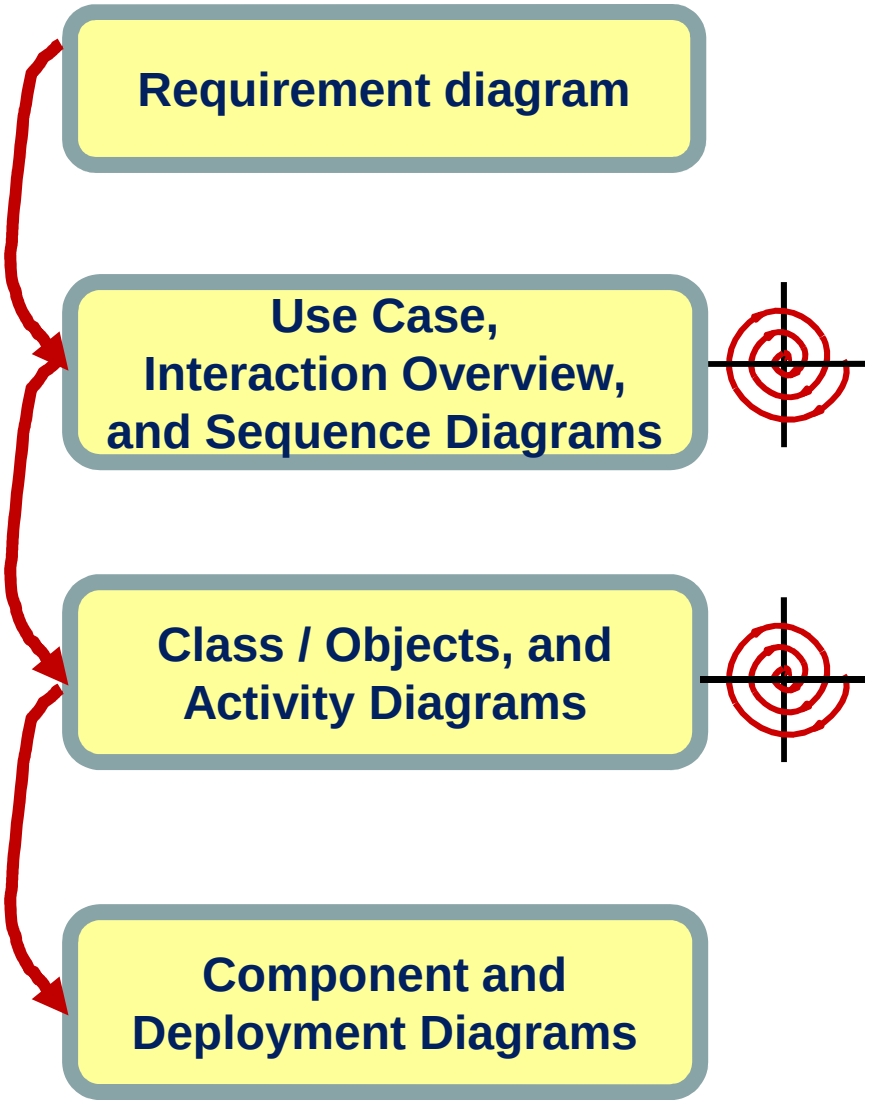
- **TURTLE (Timed UML and RT-LOTOS Environment)**
  - Formal semantics
    - Translation to RT-LOTOS and UPPAAL
  - Tool: TURTLE Toolkit (TTool)
    - Alcatel-Lucent, Texas Instruments, Freescale, DoceaPower, FP7 EVITA project (BMW, Bosch, ..), FP7 SACRA Project (Infineon, …)
  - Method

- **Need for tutorials and educational case studies !**

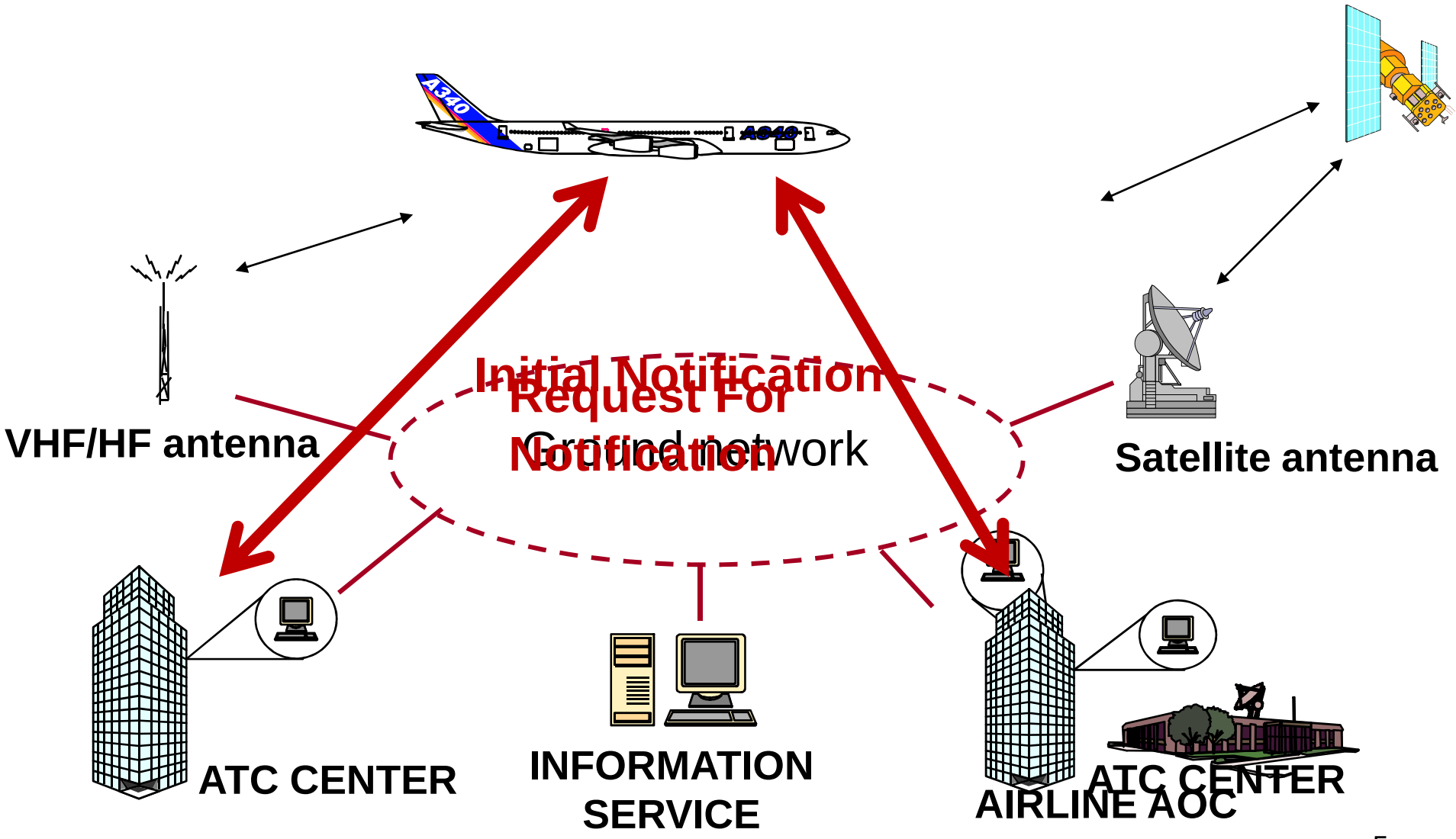
TURTLE : a Formal UML profile supported by TTool



Method and Diagrams

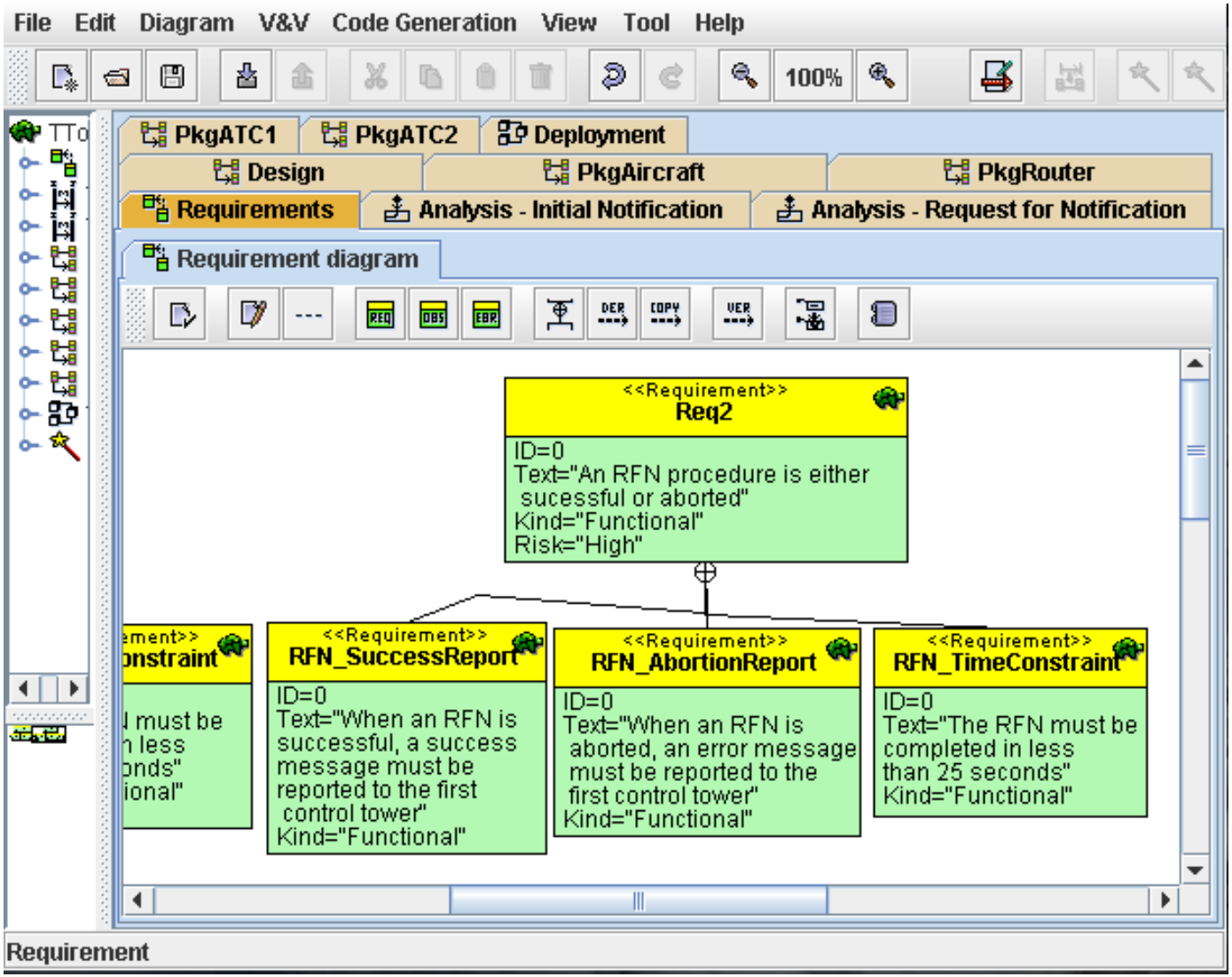


Future Air Navigation System



Requirement Capture

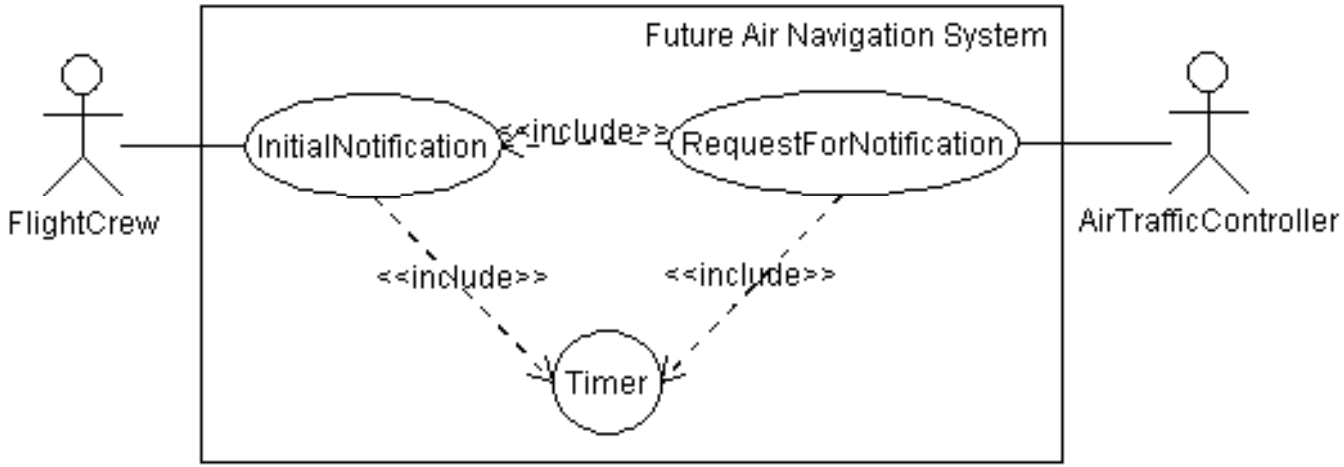
- **The IN procedure either completes within 10 minutes or aborts. The pilot accordingly receives a “success” or “abortion” report.**



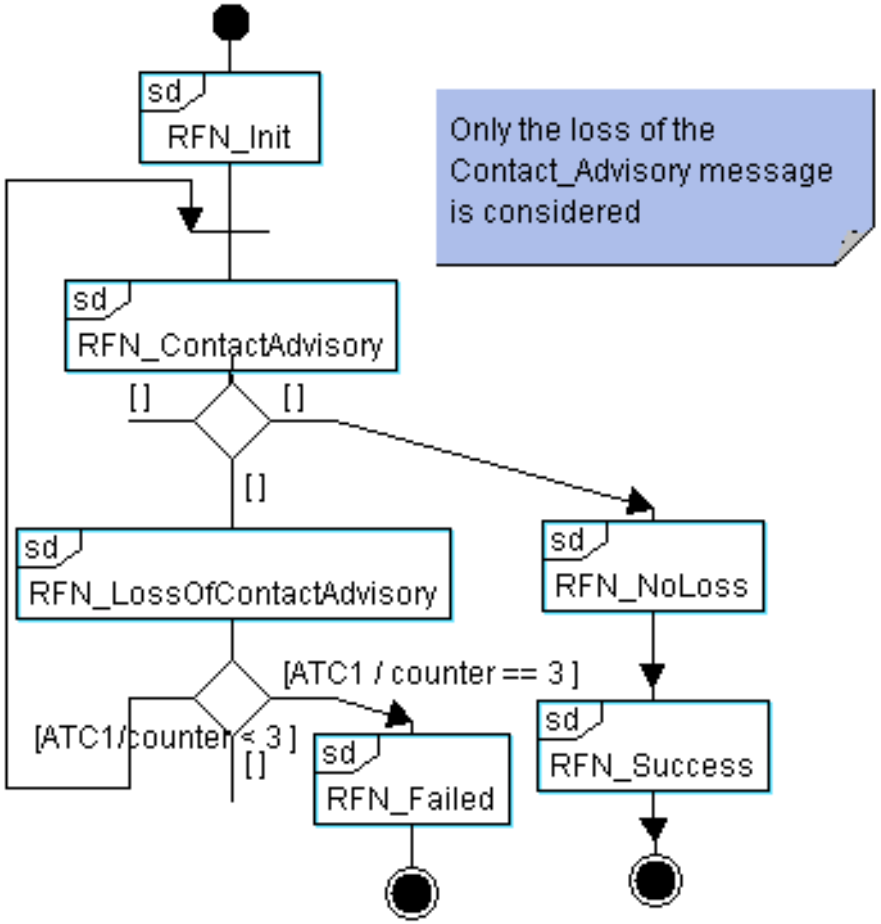
Analysis

- **A Use-Case Diagram without boot, power-off, and maintenance**
- **An Interaction Overview Diagram (IOD) structures elementary scenarios expressed by sequence diagrams**
  - Initial Notification (connection set up, connection not established)
- **Formal verification based on reachability analysis**
  - Initial Notification
    - Req 1 and IN_Abortion_Report
      - Minimization
    - IN_Time_Constraint
      - Dynamic Timed Automaton
      - (observer)

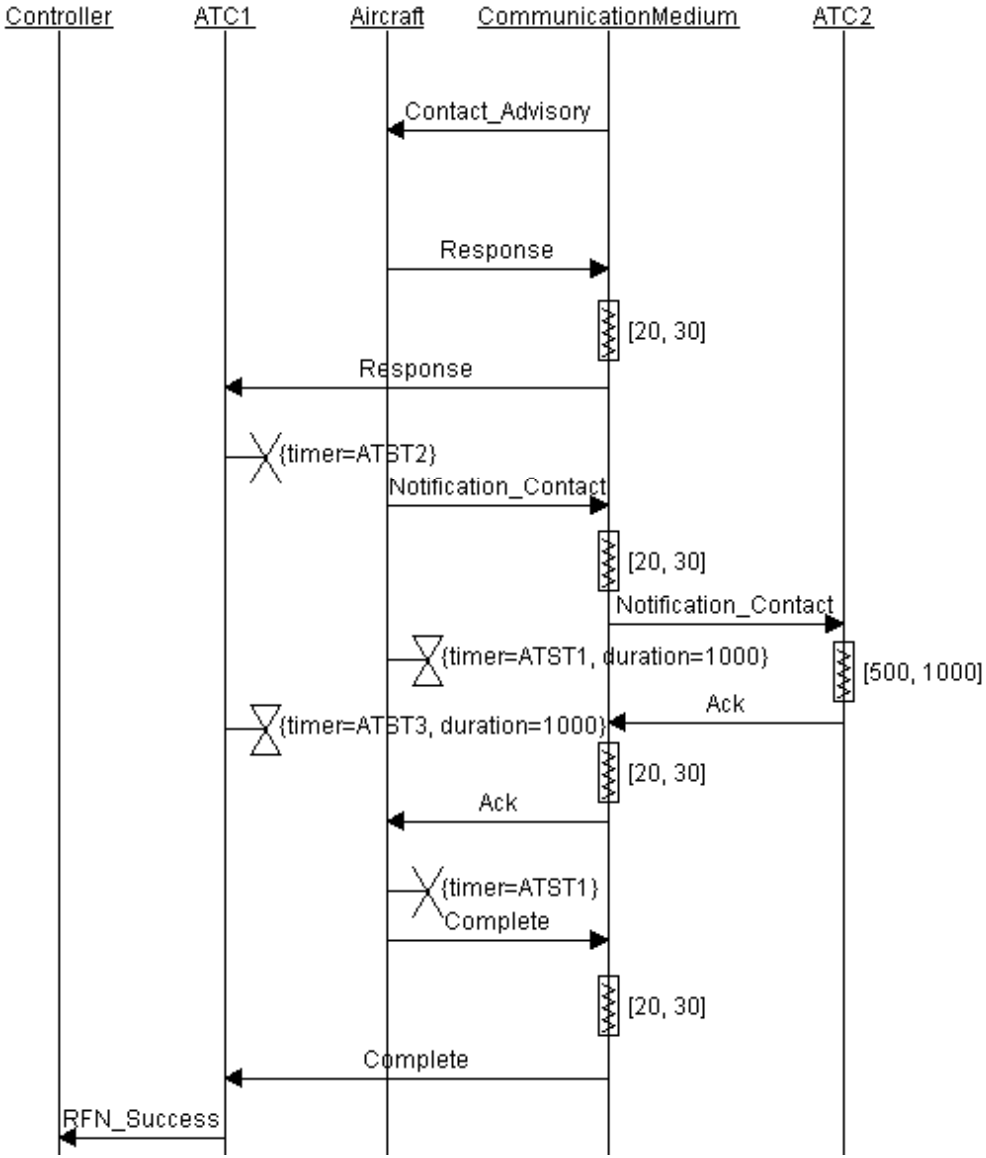
Analysis: Use Case Diagram of RFN



Analysis: IOD RFN

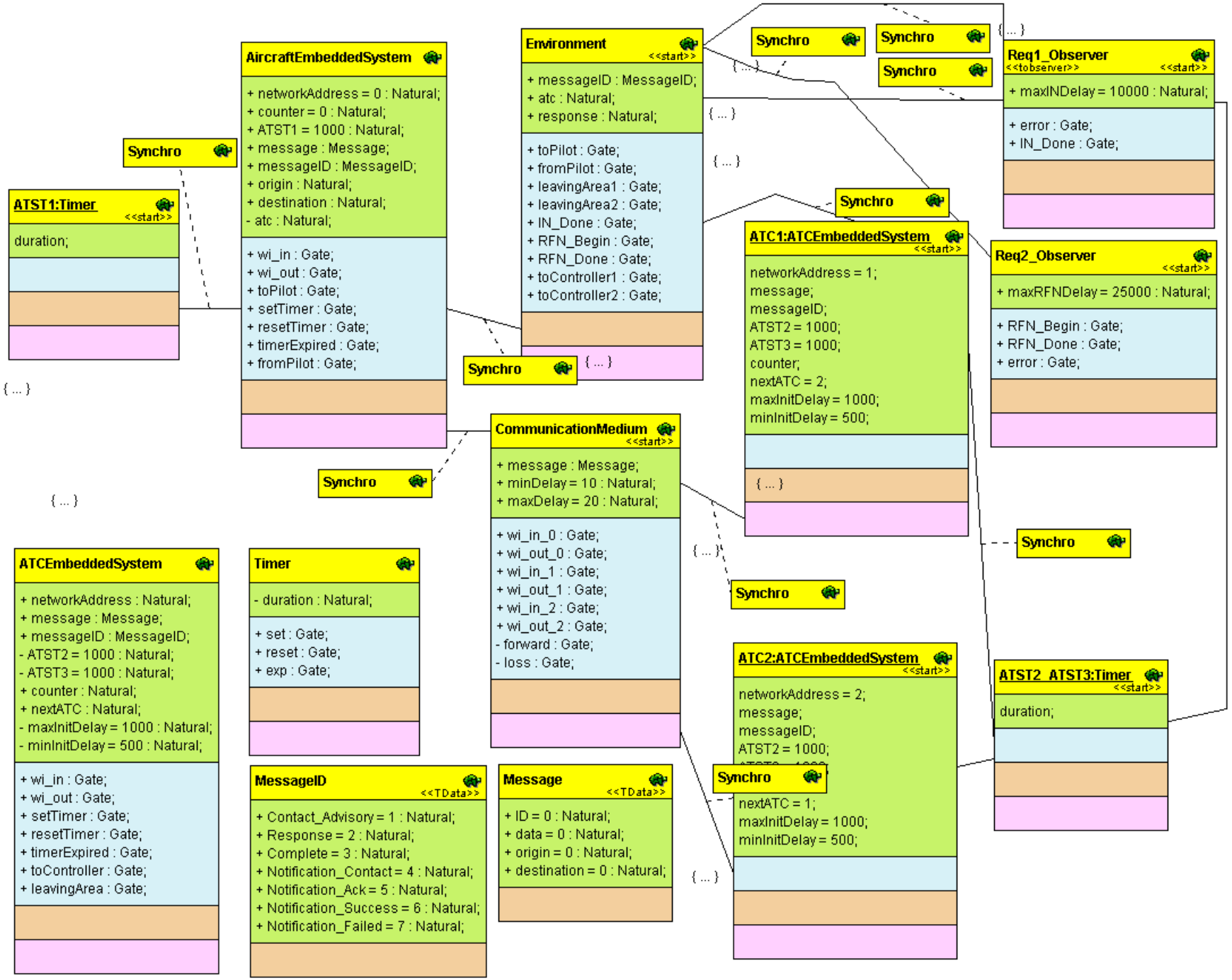


Analysis: SD « RFN_NoLoss »

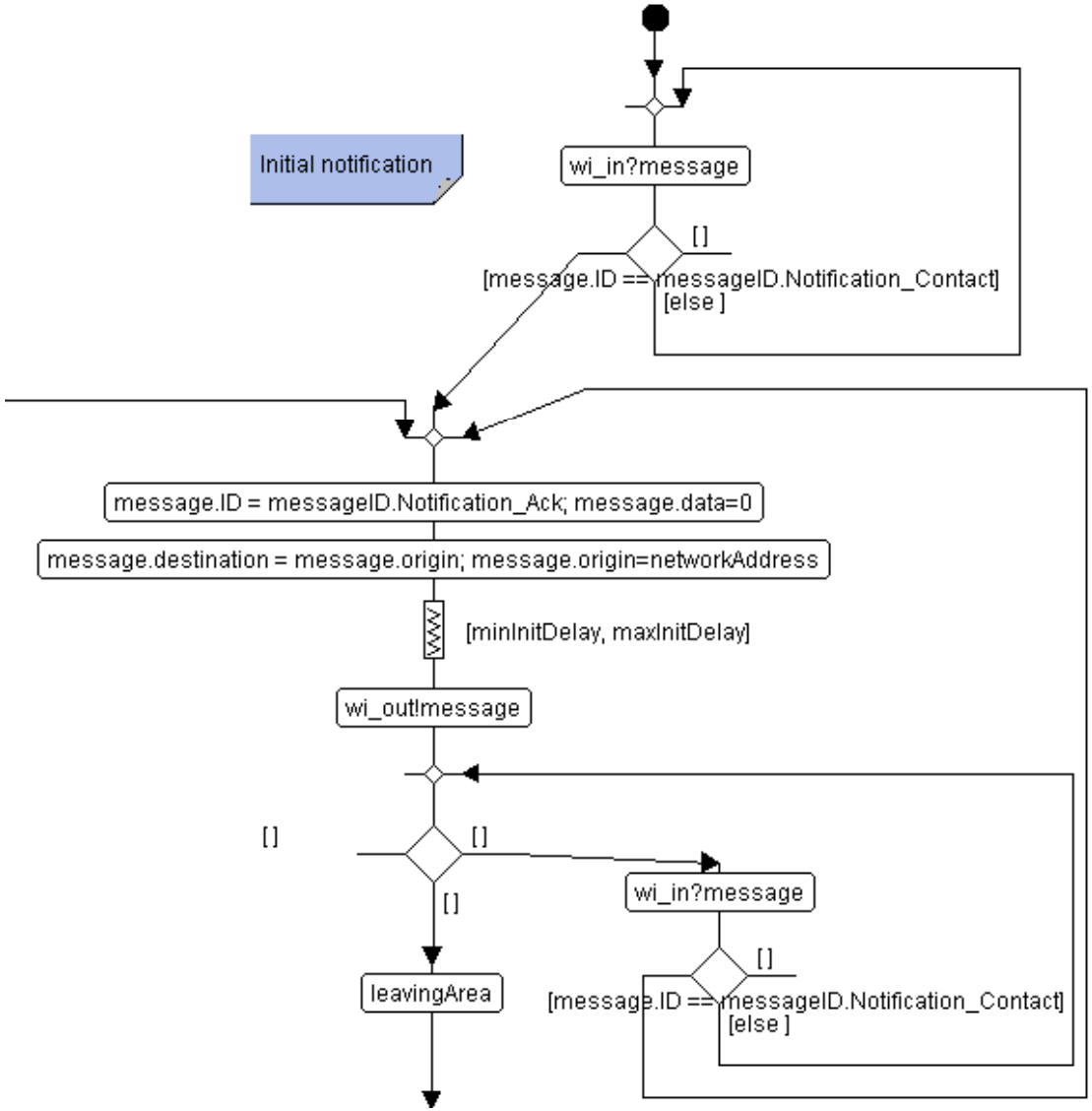


Design

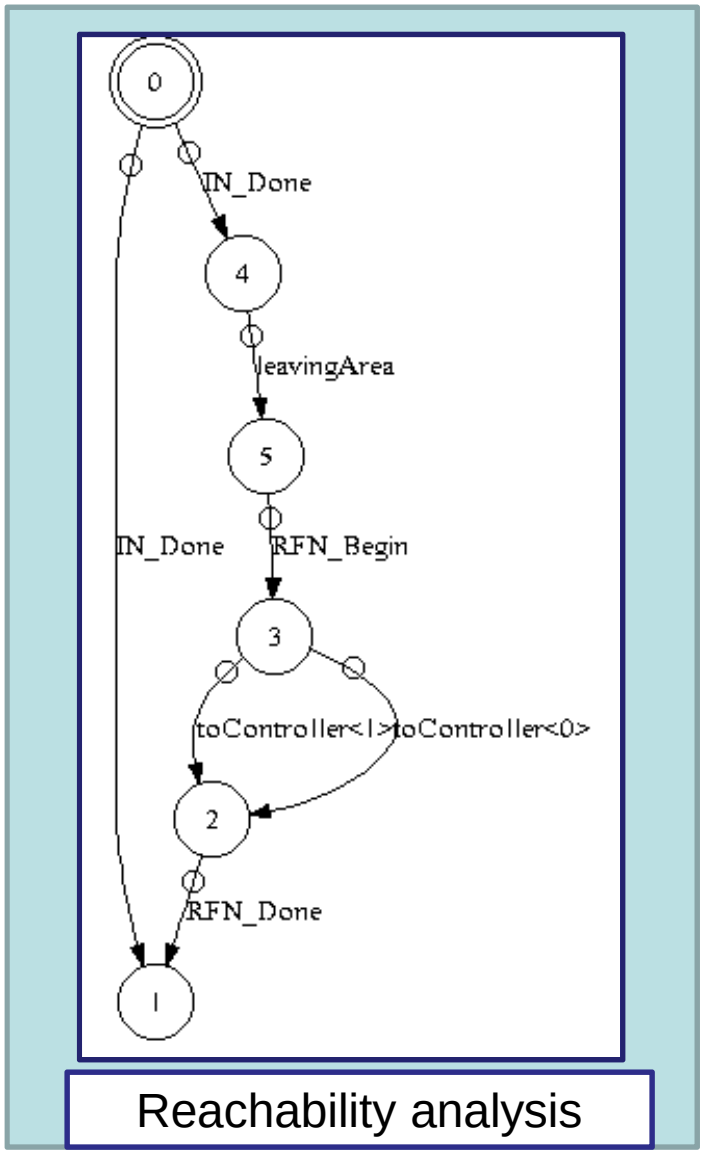
- **Automatic generation of design diagrams**
  - From the Interaction Overview and Sequence diagrams
- **Hand-made enhancements**
  - Class/Object diagram (on next slide!)
    - Parameterized messages with complex data structures
  - Activity diagrams
    - Routing inside the communication medium
- **Formal verification**
  - Graph minimization and model-checking
  - Observers


AircraftEmbeddedSystem
+ networkAddress = 0 : Natural;
+ counter = 0 : Natural;
+ ATST1 = 1000 : Natural;
+ message : Message;
+ messageID : MessageID;
+ origin : Natural;
+ destination : Natural;
- atc : Natural;
+ wi_in : Gate;
+ wi_out : Gate;
+ toPilot : Gate;
+ setTimer : Gate;
+ resetTimer : Gate;
+ timerExpired : Gate;
+ fromPilot : Gate;
Environment
<<start>>
+ messageID : MessageID;
+ atc : Natural;
+ response : Natural;
+ toPilot : Gate;
+ fromPilot : Gate;
+ leavingArea1 : Gate;
+ leavingArea2 : Gate;
+ IN_Done : Gate;
+ RFN_Begin : Gate;
+ RFN_Done : Gate;
+ toController1 : Gate;
+ toController2 : Gate;
Req1_Observer
<<tobserver>>
<<start>>
+ maxINDelay = 10000 : Natural;
+ error : Gate;
+ IN_Done : Gate;
Req2_Observer
<<start>>
+ maxRFNDelay = 25000 : Natural;
+ RFN_Begin : Gate;
+ RFN_Done : Gate;
+ error : Gate;
ATST1:Timer
<<start>>
duration;
ATC1:ATCEmbeddedSystem
<<start>>
networkAddress = 1;
message;
messageID;
ATST2 = 1000;
ATST3 = 1000;
counter;
nextATC = 2;
maxInitDelay = 1000;
minInitDelay = 500;
ATC2:ATCEmbeddedSystem
<<start>>
networkAddress = 2;
message;
messageID;
ATST2 = 1000;
nextATC = 1;
maxInitDelay = 1000;
minInitDelay = 500;
ATST2_ATST3:Timer
<<start>>
duration;
CommunicationMedium
<<start>>
+ message : Message;
+ minDelay = 10 : Natural;
+ maxDelay = 20 : Natural;
+ wi_in_0 : Gate;
+ wi_out_0 : Gate;
+ wi_in_1 : Gate;
+ wi_out_1 : Gate;
+ wi_in_2 : Gate;
+ wi_out_2 : Gate;
- forward : Gate;
- loss : Gate;
ATCEmbeddedSystem
+ networkAddress : Natural;
+ message : Message;
+ messageID : MessageID;
- ATST2 = 1000 : Natural;
- ATST3 = 1000 : Natural;
+ counter : Natural;
+ nextATC : Natural;
- maxInitDelay = 1000 : Natural;
- minInitDelay = 500 : Natural;
+ wi_in : Gate;
+ wi_out : Gate;
+ setTimer : Gate;
+ resetTimer : Gate;
+ timerExpired : Gate;
+ toController : Gate;
+ leavingArea : Gate;
Timer
- duration : Natural;
+ set : Gate;
+ reset : Gate;
+ exp : Gate;
MessageID
<<TData>>
+ Contact_Advisory = 1 : Natural;
+ Response = 2 : Natural;
+ Complete = 3 : Natural;
+ Notification_Contact = 4 : Natural;
+ Notification_Ack = 5 : Natural;
+ Notification_Success = 6 : Natural;
+ Notification_Failed = 7 : Natural;
Message
<<TData>>
+ ID = 0 : Natural;
+ data = 0 : Natural;
+ origin = 0 : Natural;
+ destination = 0 : Natural;
Synchro
{ ... }

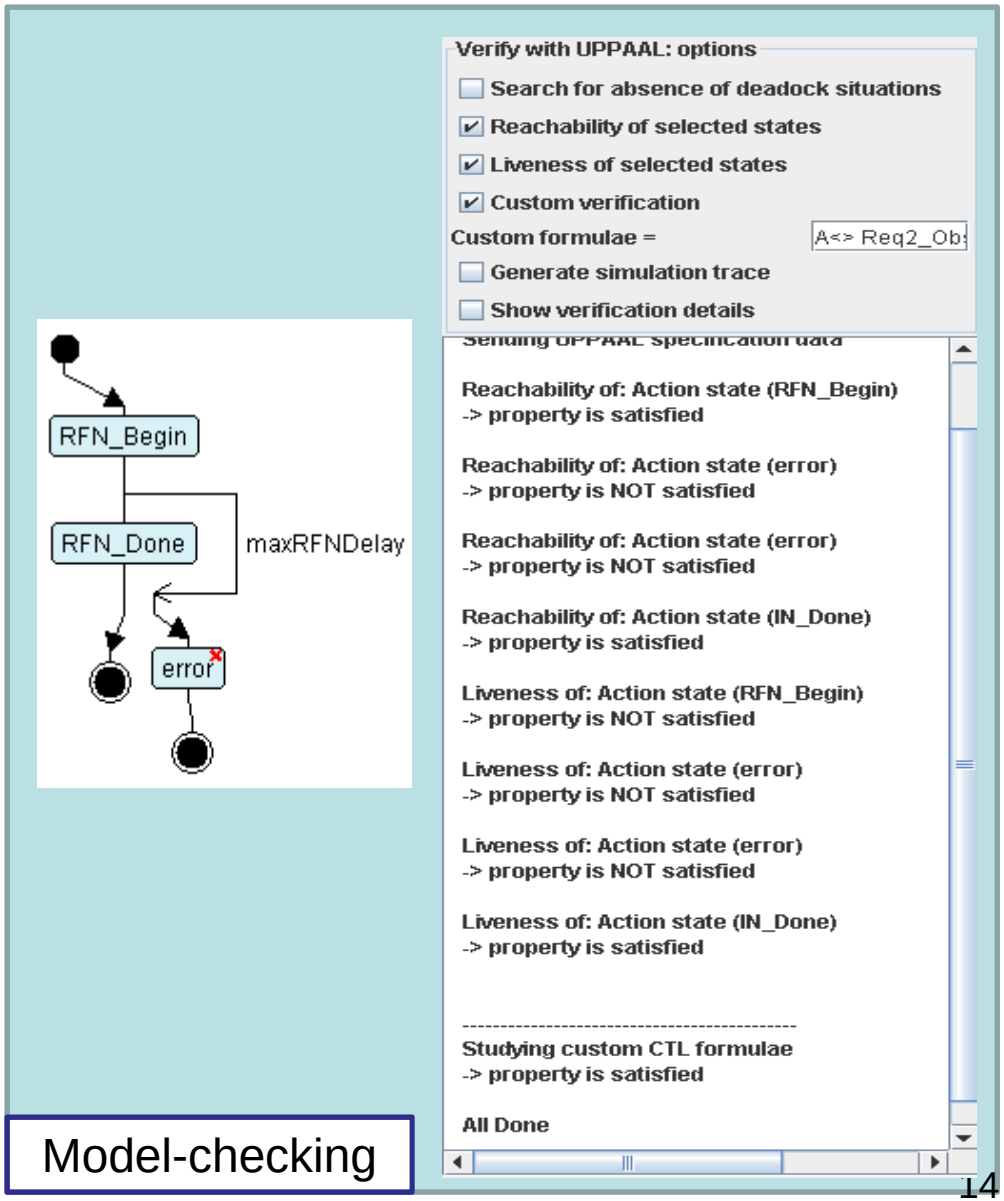
Design: Excerpt of the Activity Diagram of the class « ATCEmbeddedSystem »



Design: Formal Verification



Reachability analysis



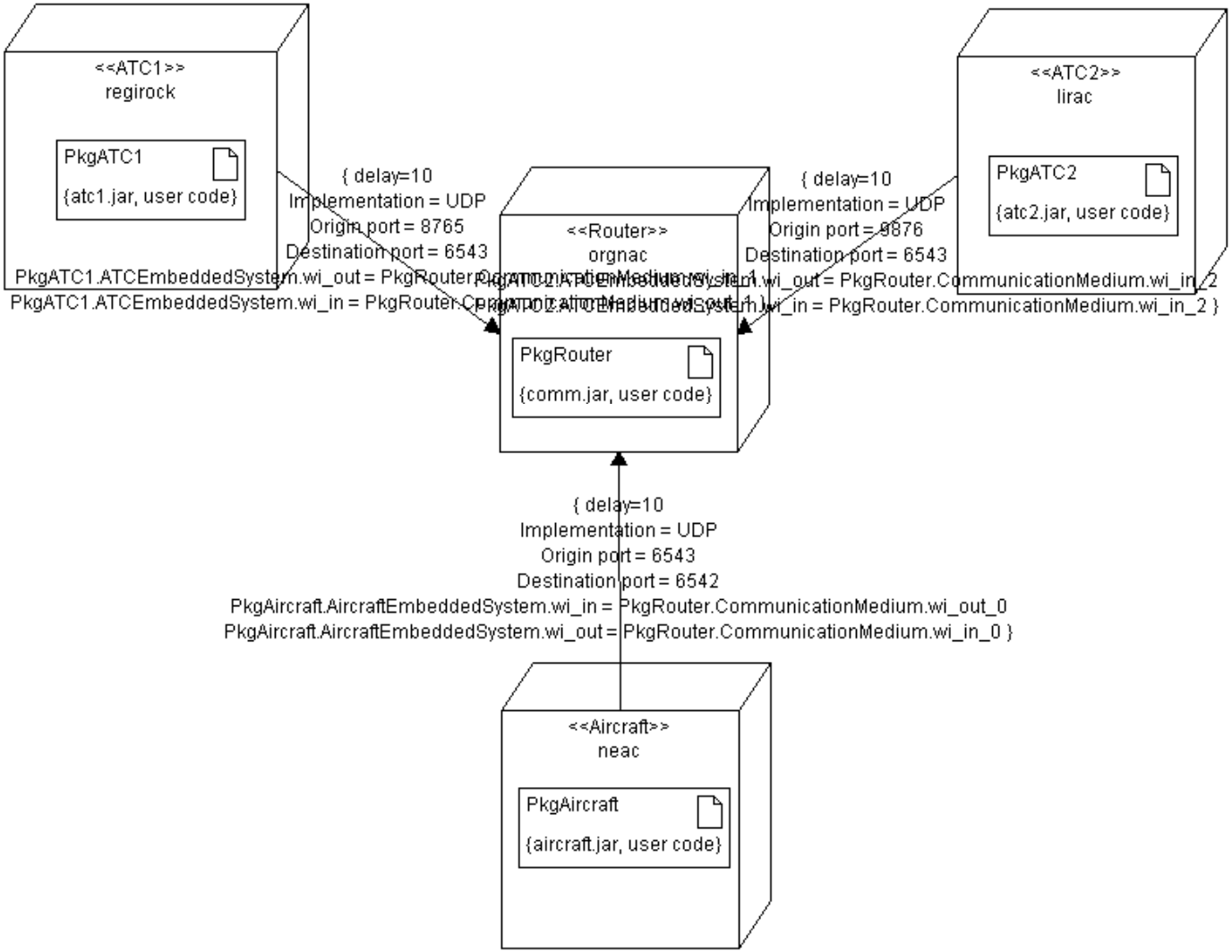
Model-checking

Prototyping

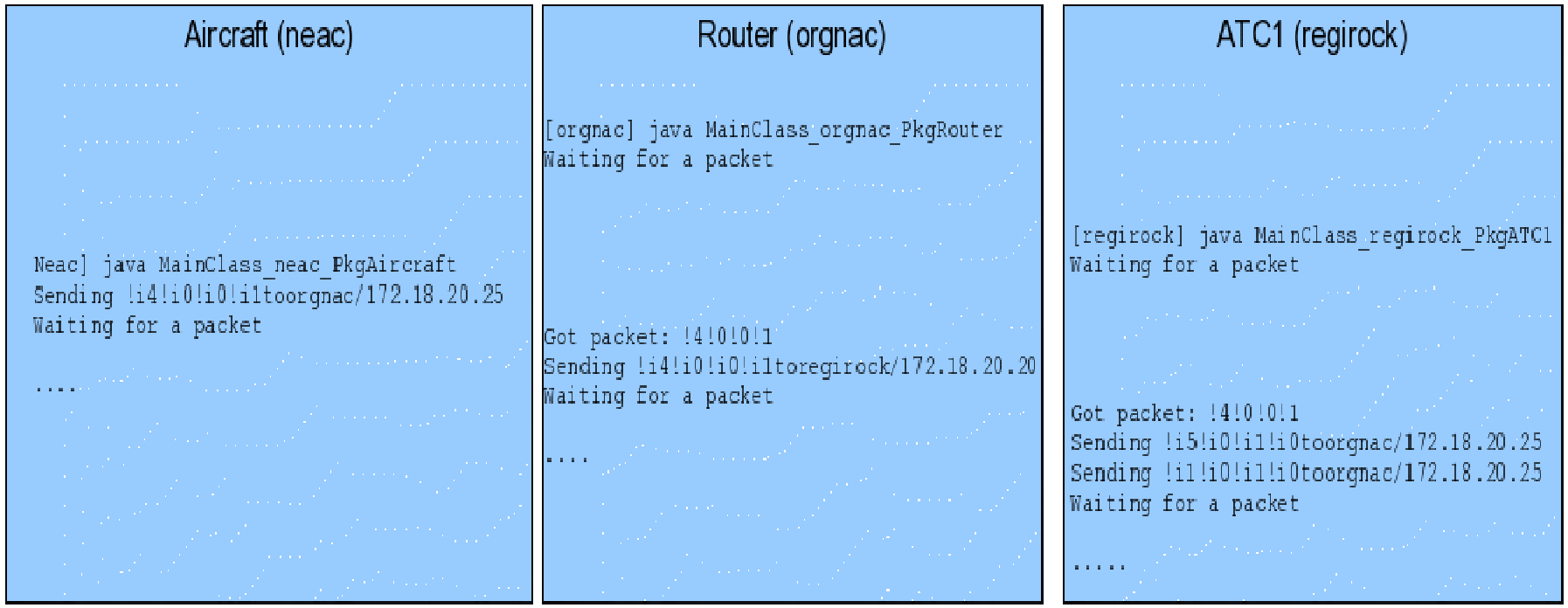
- **Mapping « software components » on execution nodes**
- **Generating prototype code**
- **Deployment diagram**
  - Embedded system of the aircraft
  - Air Control Tower #1
  - Air Control Tower #2
  - Communication system (Routing application)
- **Automatically generated Java code**

- **Execution Trace**

Prototyping: Deployment Diagram



Prototyping: Execution Trace



Conclusions

- **Contributions**
  - A verification-centric UML profile
  - A method for real-time and distributed system design
  - TURTLE toolkit (TTool)
    - http://labsoc.comelec.enst.fr/turtle/ttool.html
- **Projects that used TURTLE and TTool (selection)**
  - SAFECAST (secured group communication system)
  - EVITA (modeling and proof of security properties in vehicle embedded systems)
  - DoceaPower (modeling of power managers)
- **Ongoing work**
  - Extended Requirement Diagrams with security requirements
  - Dimensioning diagrams (Network Calculus)
  - DIPLODOCUS (Modeling and simulation of Systems-on-chip)