MOGENTES

MOdel-based GENeration of Tests for Embedded Systems

#216679 FP7-ICT-2007-1-3.3 Embedded Systems Design

SEVENTH FRAMEWORK PROGRAMME

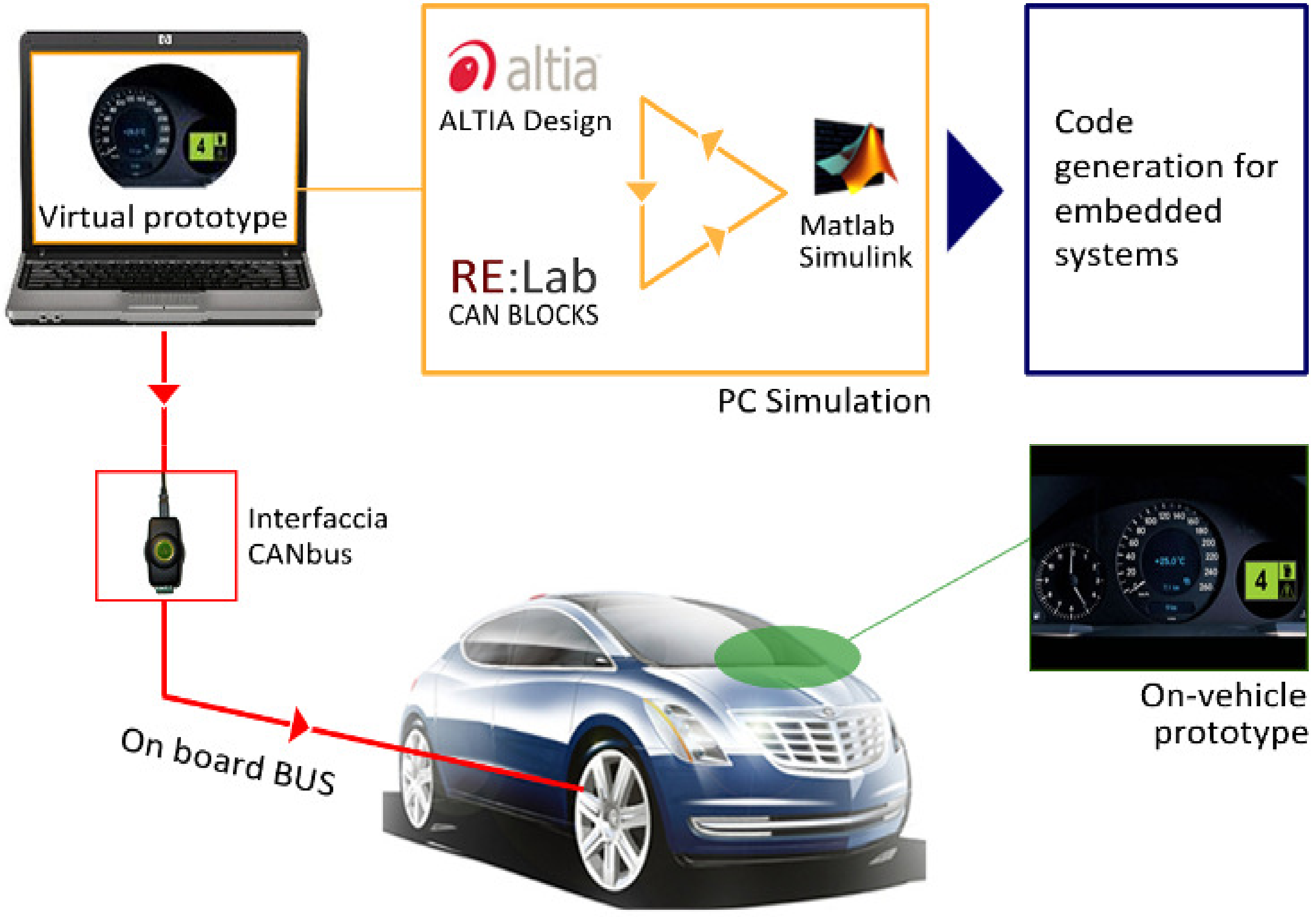
Model-based generation of tests for embedded systems: the MOGENTES EU project

Stefano Marzani, Lorenzo Fantesini

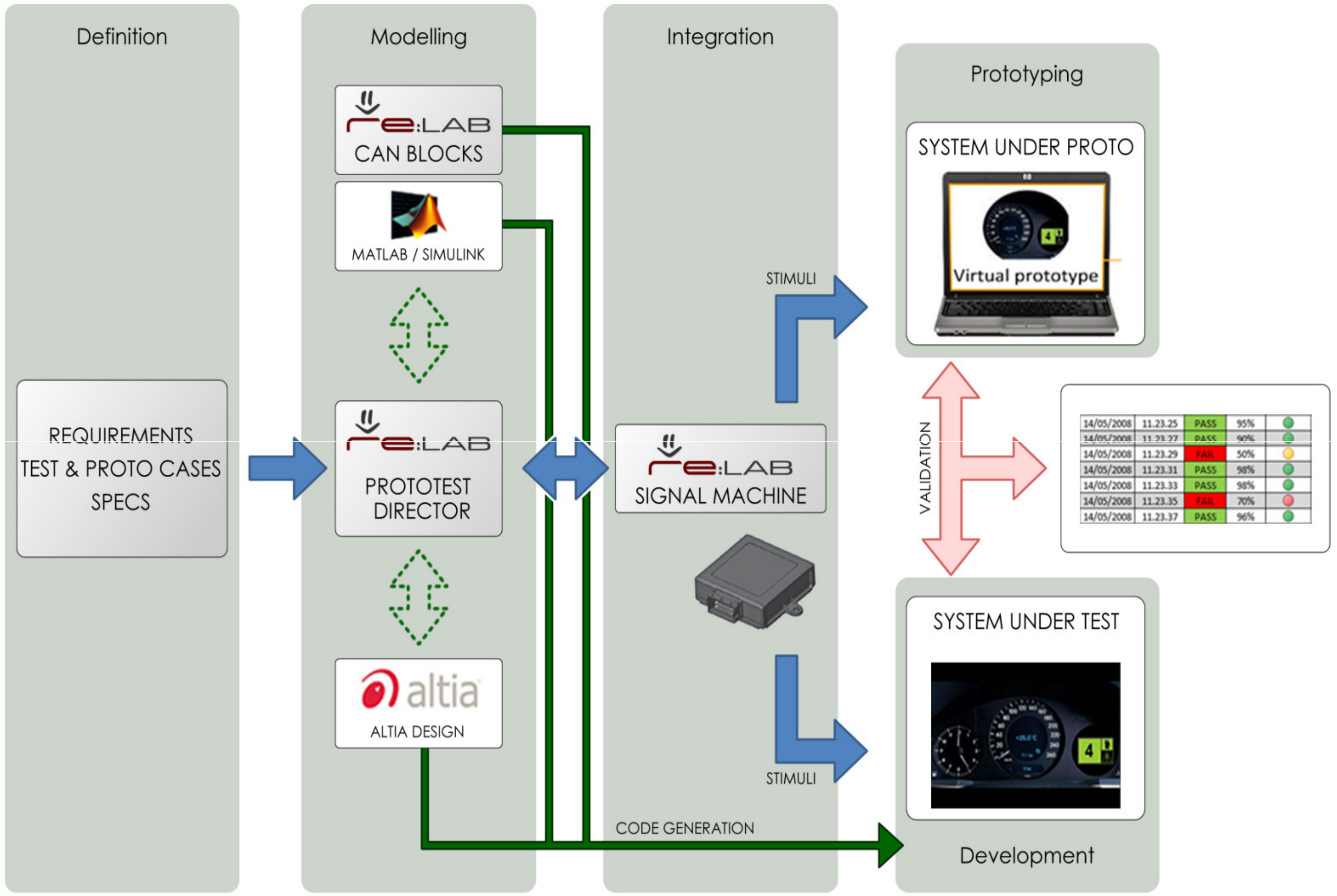
s.marzani@re-lab.it, l.fantesini@re-lab.it

Workshop SPIN, 04/06/2009

Background: RE:Lab MBD approach (1)



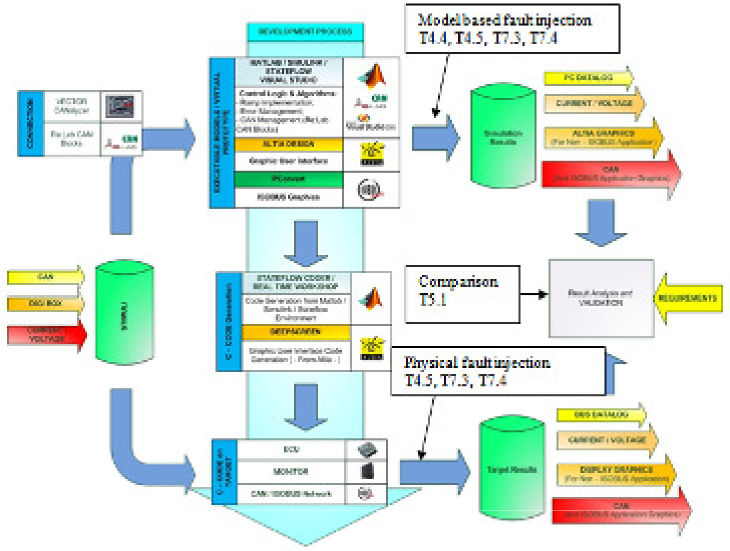
Background: RE:Lab MBD approach (2)



14/05/2008	11.23.25	PASS	95%	🟢
14/05/2008	11.23.27	PASS	90%	🟢
14/05/2008	11.23.29	FAIL	50%	🟡
14/05/2008	11.23.31	PASS	98%	🟢
14/05/2008	11.23.33	PASS	98%	🟢
14/05/2008	11.23.35	FAIL	70%	🔴
14/05/2008	11.23.37	PASS	96%	🟢

MOGENTES overview

- Model-Based GEneration of TEst-Cases for dependable Embedded Systems
- **Objective**:
  - automatic generation of test cases to enhance *testing* and *verification* of dependable embedded systems
  - target applications: trains, agricultural machines, cars
  - a common model-based approach *(framework)* for test and validation



MOGENTES Consortium

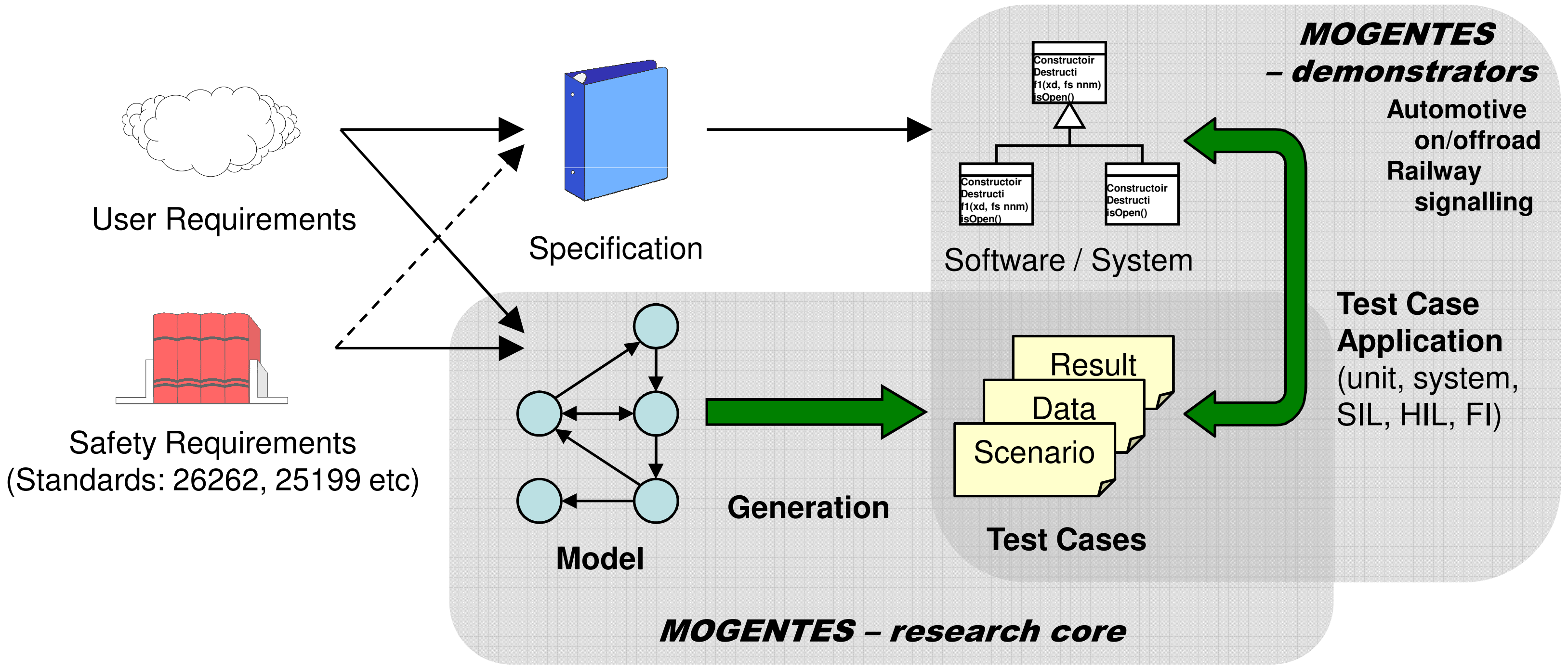
Budget: 4.436.511 €

Partners:

- Austrian Research Centers Gmbh - ARC
- Swiss Federal Institute of Technology Zurich / University of Oxford
- Ford GmbH
- Budapest University of Technology and Economics
- Graz University of Technology
- PROLAN
- Prover Technology AB
- SP Technical Research Institute of Sweden
- Thales Rail Signalling Solutions
- RE:Lab

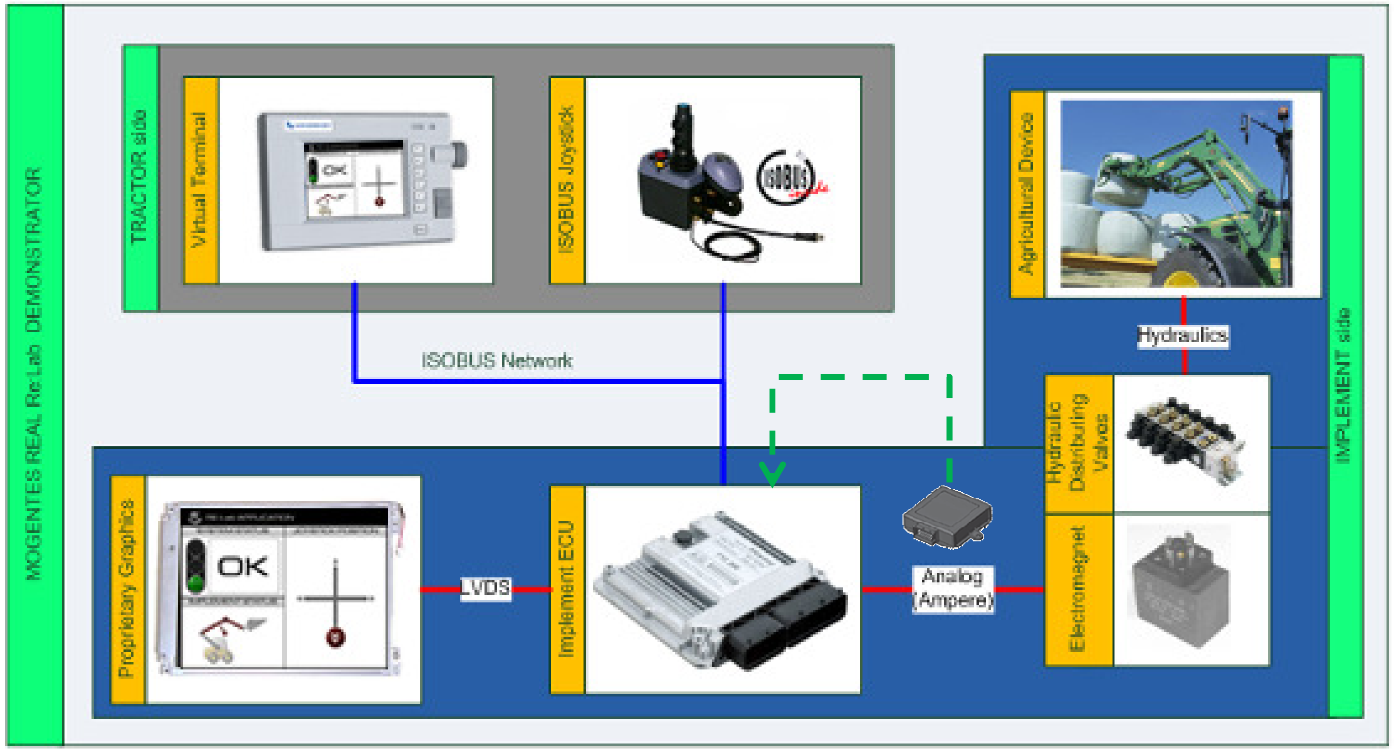
MOGENTES overall framework

- Automatic generation of efficient test cases to verify system safety correctness using formal methods and fault injection



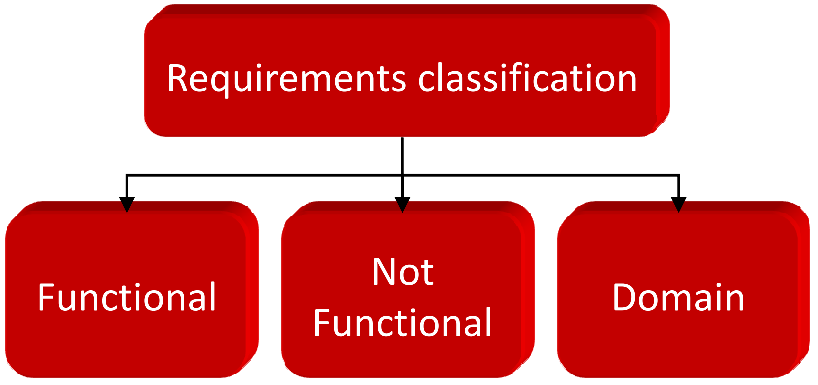
Framework application

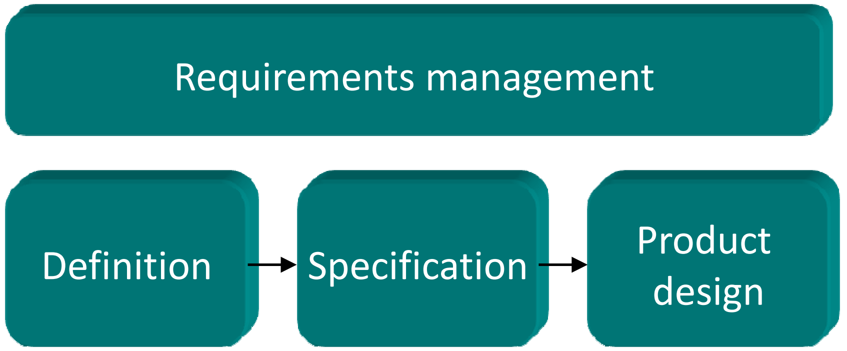
- Bucket: off-highway machine



Phase 1: user requirements and specs

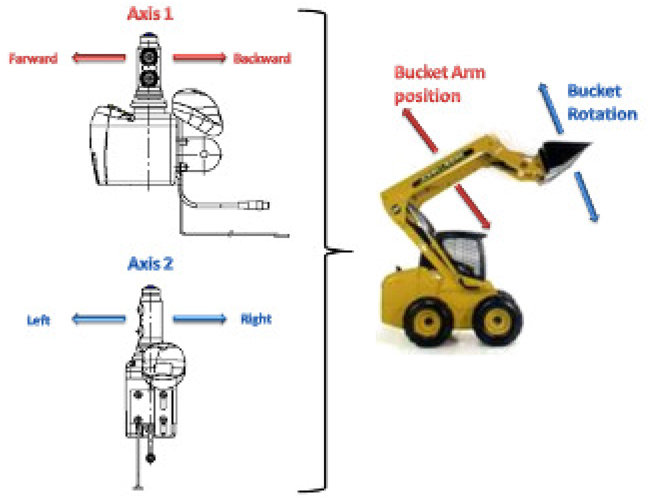
- Highlight, analyze and validate system requirements
  - *Classification*: requirements are grouped according to their characteristics
  - *Management*: system specifications are generated and translated into product design guidelines



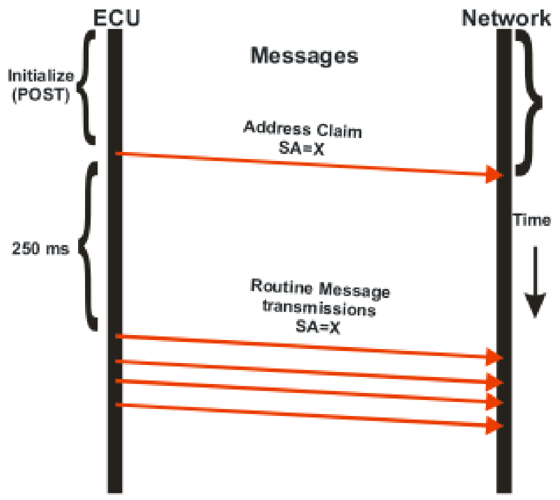


Phase 1: user requirements and specs

- Functional
  - *System Behaviour*
    - The system must read the incoming CAN messages input signals of the control joystick each time a message is received.
    - The ISOBUS joystick must sent a status message at least every 100 milliseconds

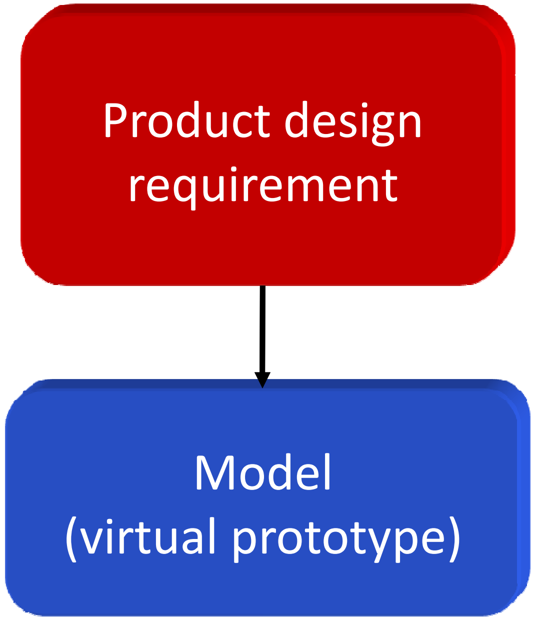


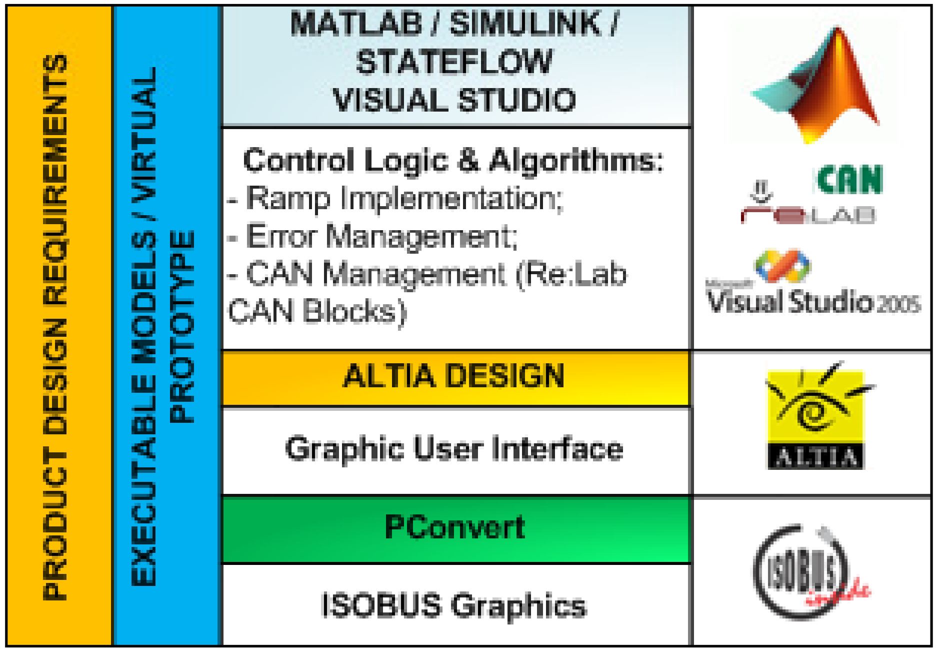
- Non functional (e.g. ISOBUS requirements)
  - *ECU initialization*
    - Initialization of an ECU with address claim and no contention



Phase 2: model development

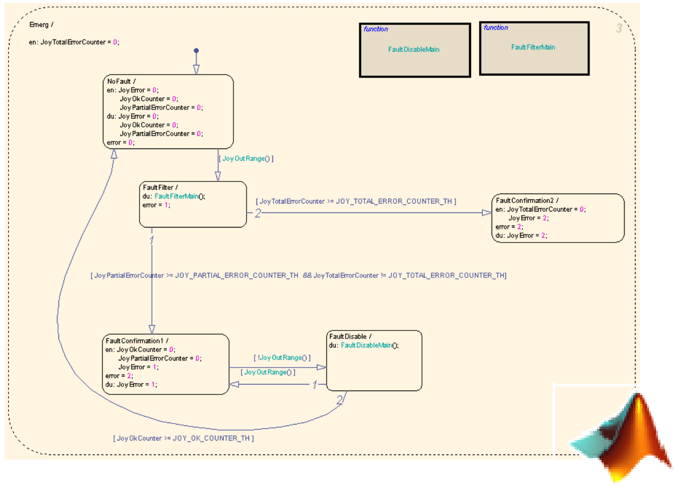
- Basing on system specifications, the model (prototype) of the final system is developed.
  - ♦ Behavioural specifications modelling (Matlab simulink/Stateflow, Visual Studio)
  - ♦ Isobus specifications implementation
  - ♦ User Interface graphics implementation (Altia, PConvert)



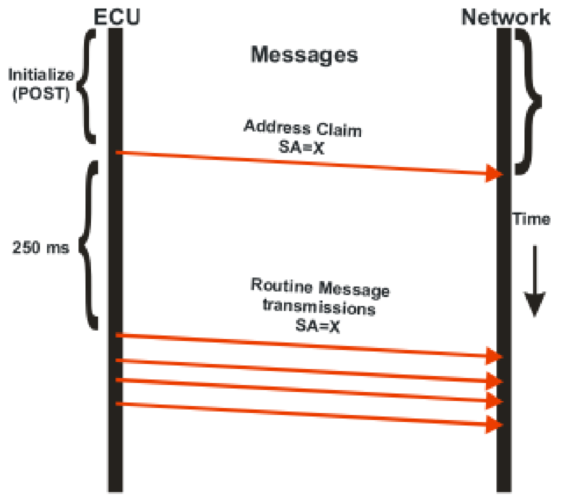

PRODUCT DESIGN REQUIREMENTS
EXECUTABLE MODELS / VIRTUAL PROTOTYPE
MATLAB / SIMULINK / STATEFLOW VISUAL STUDIO
Control Logic & Algorithms:
- Ramp Implementation;
- Error Management;
- CAN Management (Re:Lab CAN Blocks)
CAN
re:LAB
Visual Studio 2005
ALTIA DESIGN
Graphic User Interface
ALTIA
PConvert
ISOBUS Graphics
ISOBUS

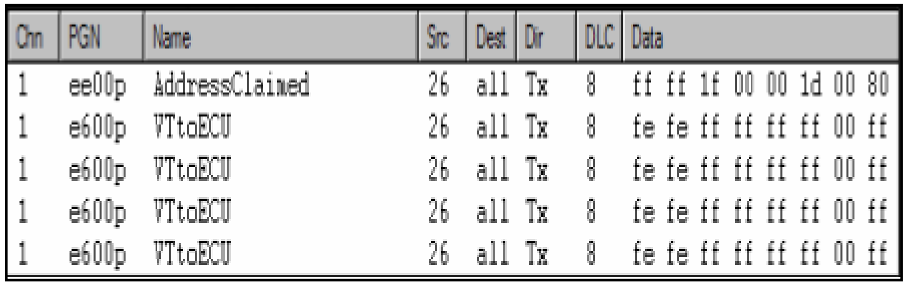
Phase 2: model development (examples)

- System behaviour modelling (State Flow)



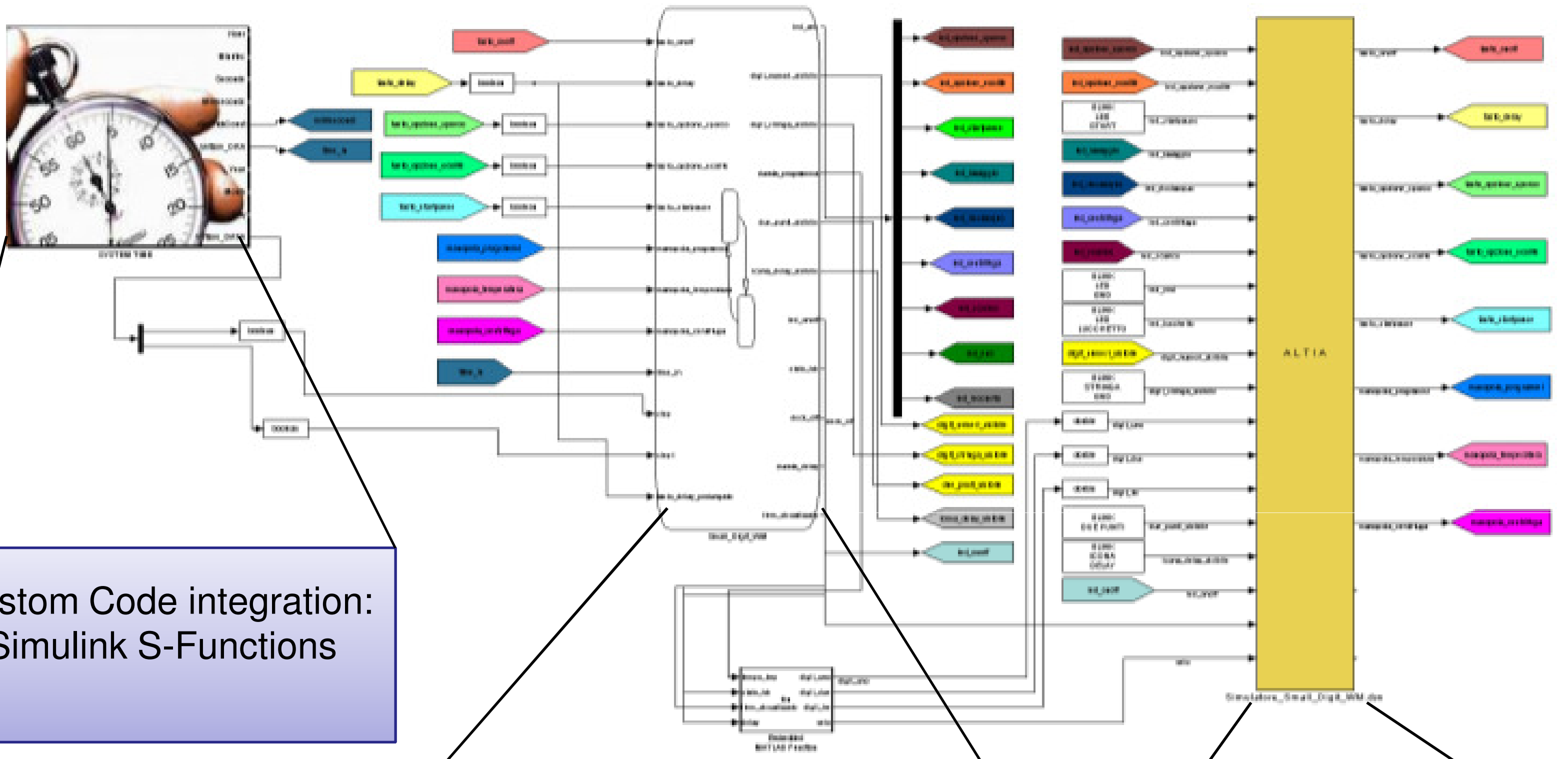
- ISOBUS implementation





Chn	PGN	Name	Src	Dest	Dir	DLC	Data
1	ee00p	AddressClaimed	26	all	Tx	8	ff ff 1f 00 00 1d 00 80
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff

Phase 2: model development – Overall



Custom Code integration:
Simulink S-Functions

- Control Logic and Algorithms
-ISOBUS Protocol and Graphic

Stateflow

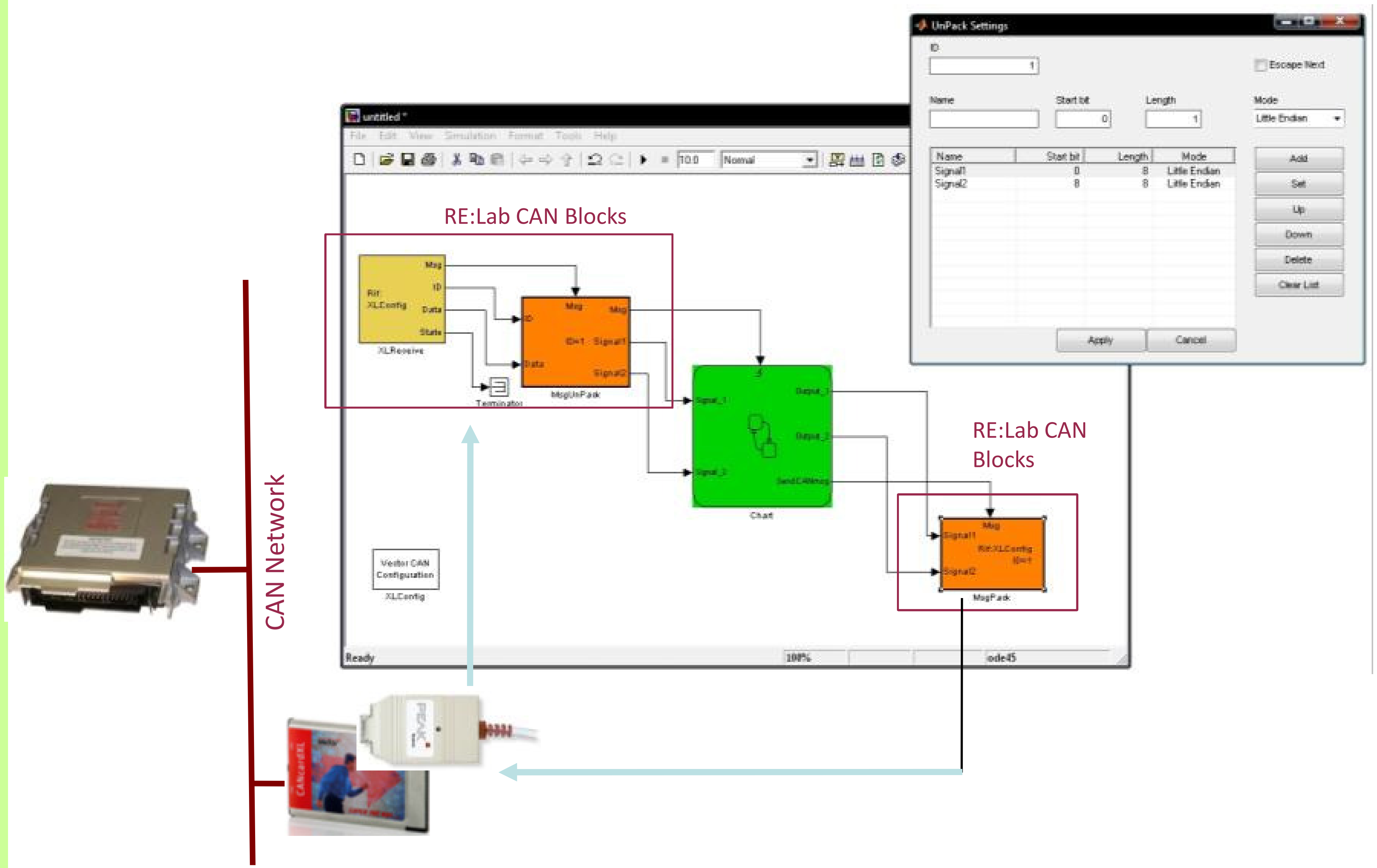
Proprietary Graphic
Development:
Altia Design

MOGENTES

MOdel-based GENeration of Tests for Embedded Systems

#216679 FP7-ICT-2007-1-3.3 Embedded Systems Design

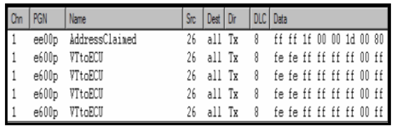
Phase 2: model development – CAN bus



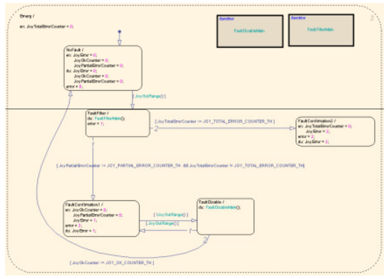
Name	Start bit	Length	Mode
Signal1	0	8	Little Endian
Signal2	8	8	Little Endian

Phase 2: model development

- User Interface graphics implementation

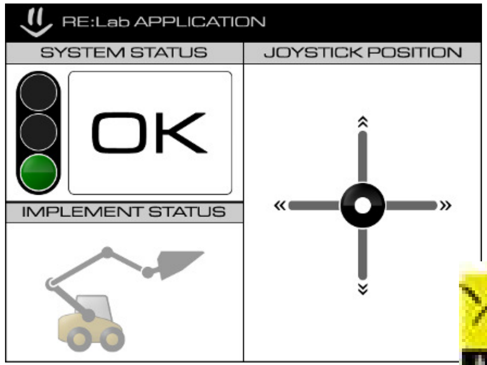


Chn	PGN	Name	Src	Dest	Dir	DLC	Data
1	ee00p	AddressClaimed	26	all	Tx	8	ff ff 1f 00 00 1d 00 80
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff



System behaviour model and ISOBUS network

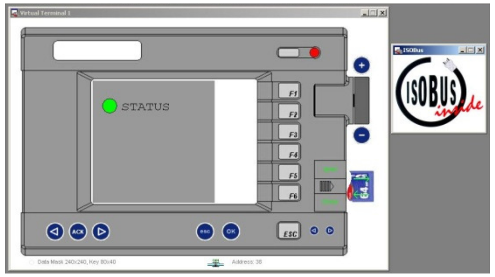
Proprietary graphics



TFT

- Not strictly related to CAN messages

ISOBUS graphics



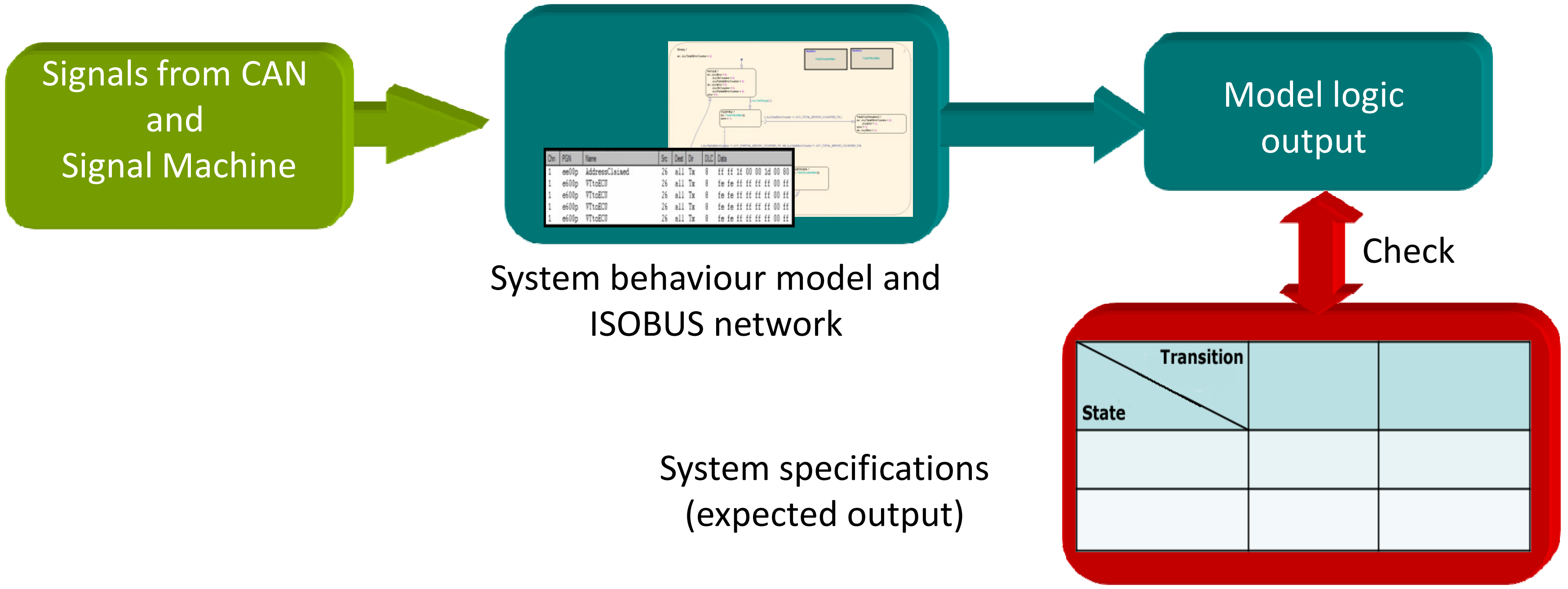
VIRTUAL TERMINAL

- Each object pool related to a CAN message

Phase 2.1: model validation

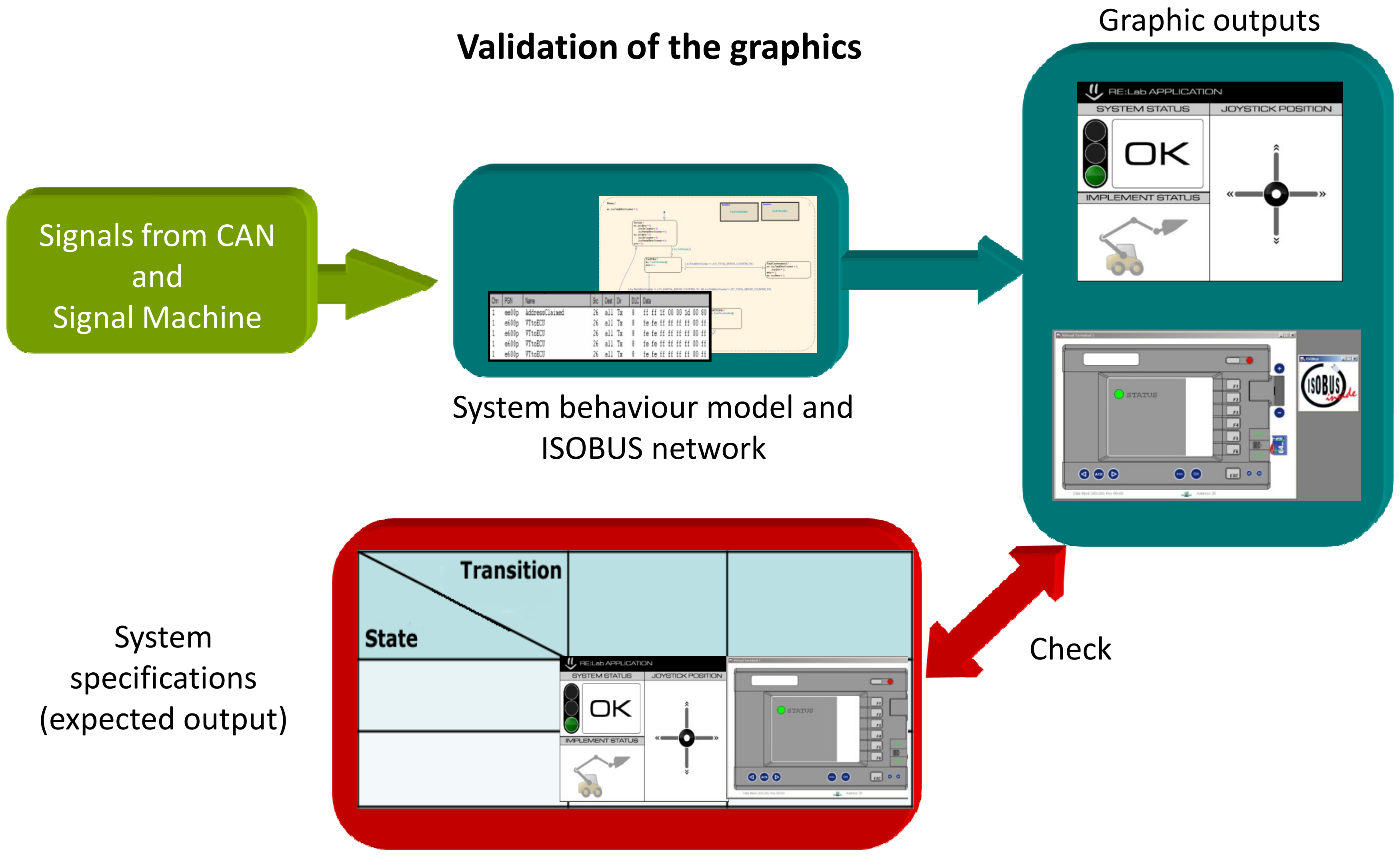
- At the end of the process, the model becomes the reference for the validation of the final system
- Before, the model should be validated

Model Checking Technique - Validation of the system logic



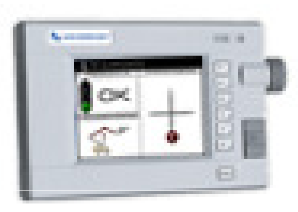
Chn	PGN	Name	Src	Dest	Dir	DLC	Data
1	ee00p	AddressClaimed	26	all	Tx	8	ff ff 1f 00 00 1d 00 80
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff
1	e600p	VTtoECU	26	all	Tx	8	fe fe ff ff ff ff 00 ff

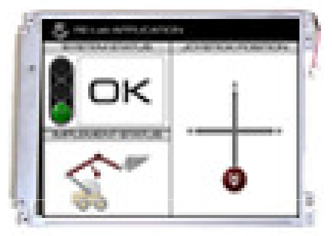
Phase 2.1: model validation


Validation of the graphics
Graphic outputs
Signals from CAN and Signal Machine
System behaviour model and ISOBUS network
RE:Lab APPLICATION
SYSTEM STATUS
JOYSTICK POSITION
OK
IMPLEMENT STATUS
STATUS
Transition
State
System specifications (expected output)
Check

Phase 3: code generation

- Code generation: from the validated model to the target final system

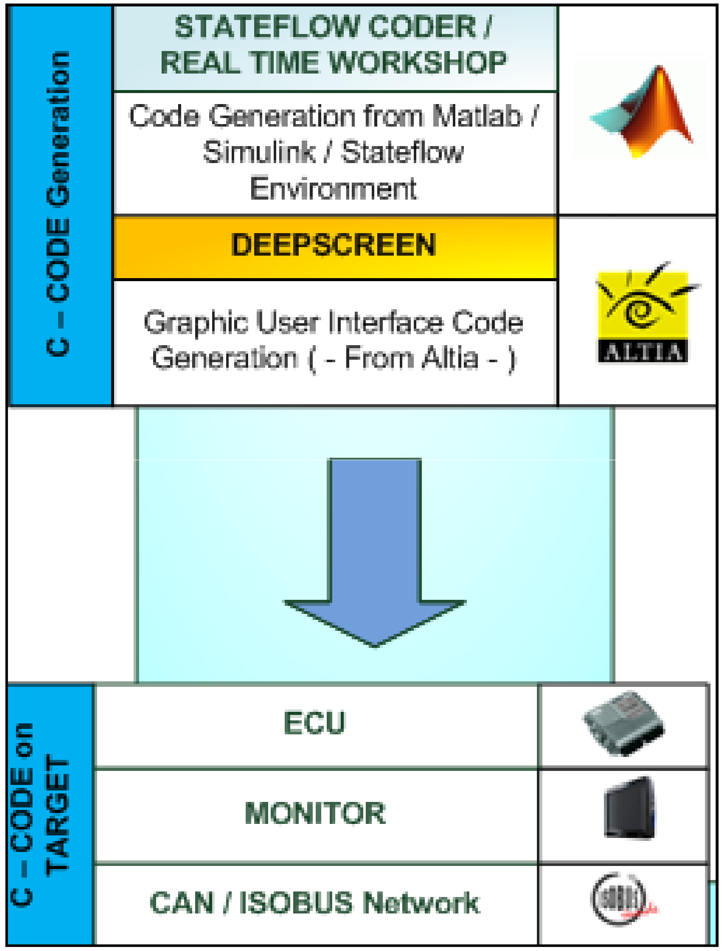

VT



TFT

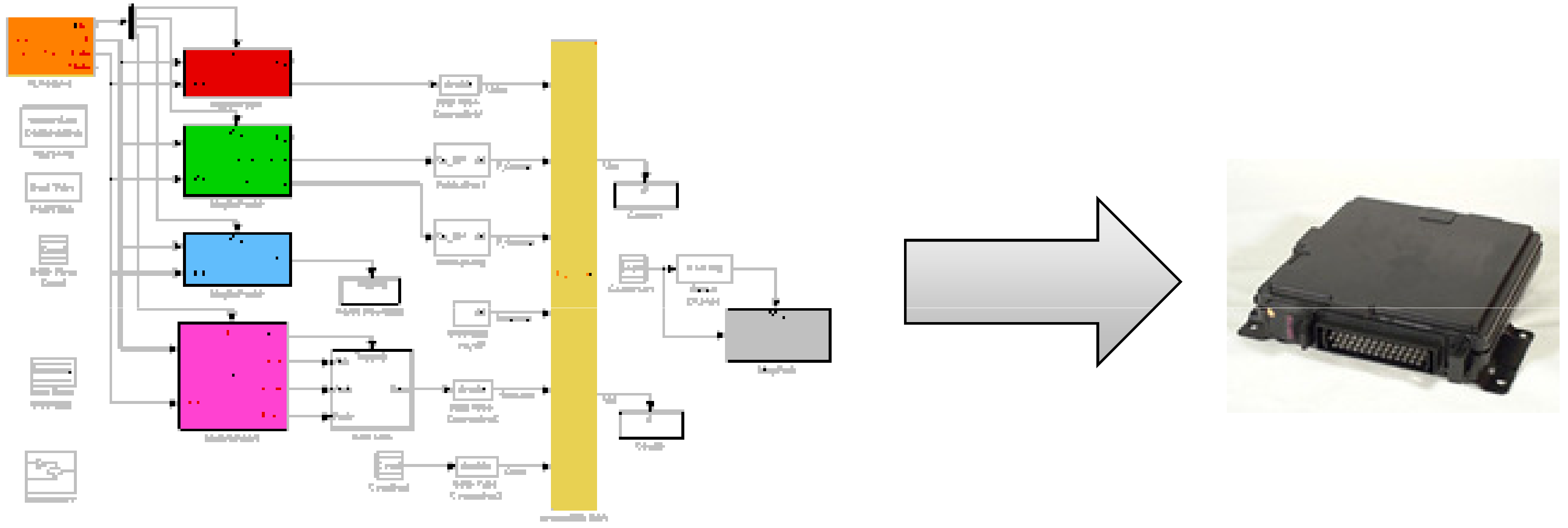
Joystick

ECU


C – CODE Generation
STATEFLOW CODER / REAL TIME WORKSHOP
Code Generation from Matlab / Simulink / Stateflow Environment
DEEPSCREEN
Graphic User Interface Code Generation (- From Altia -)
ALTIA
C – CODE on TARGET
ECU
MONITOR
CAN / ISOBUS Network

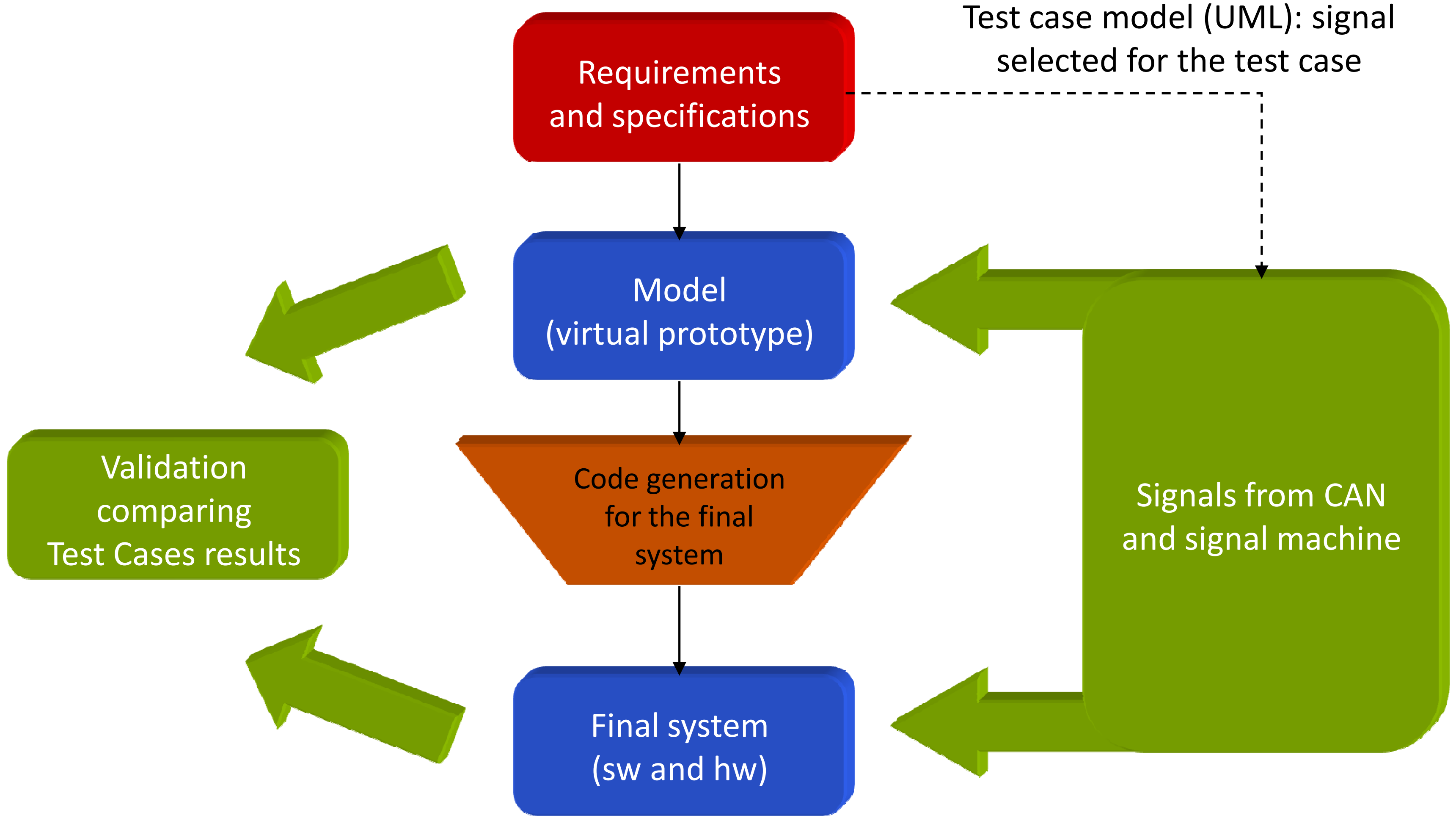
Phase 3: code generation

- From CAN blocks, Matlab/Simulink/StateFlow and Altia it is possible to generate code for specific embedded targets.



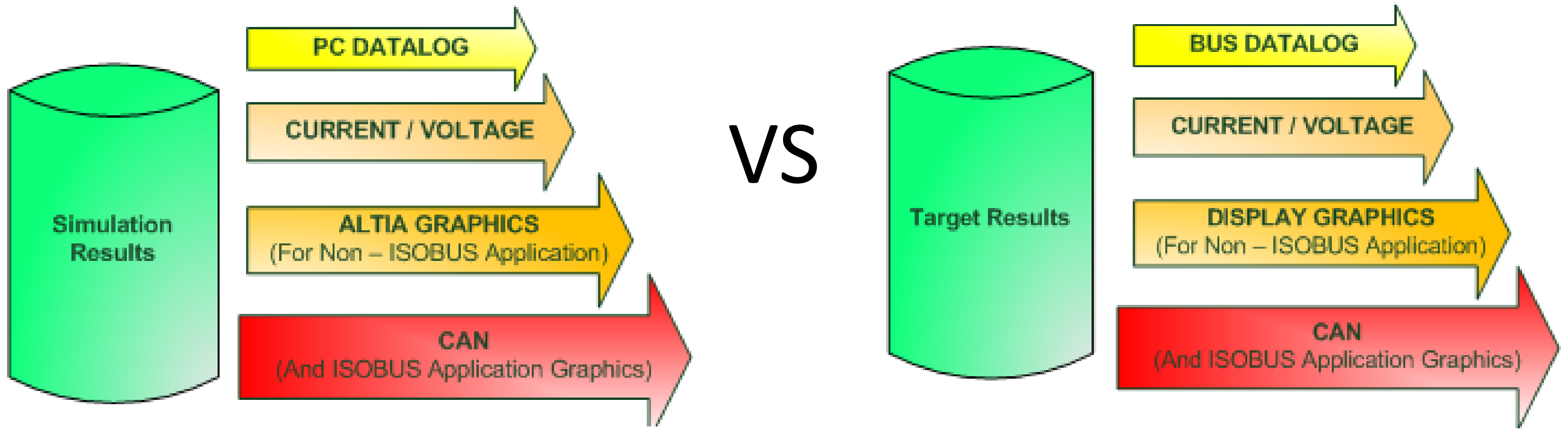
- Using this target it is possible to generate the C code firmware for a specific microcontroller and to create directly a project file ready for compilation with the proprietary toolchain.

Phase 4: final system validation


Requirements and specifications
Test case model (UML): signal selected for the test case
Model (virtual prototype)
Code generation for the final system
Final system (sw and hw)
Signals from CAN and signal machine
Validation comparing Test Cases results

Phase 4: final system validation

- Inputs to the model and the final system are:
  - CAN messages, sent using the Vector CANoe tools together with the generable CAN Blocks inside the Matlab/Simulink model environment.
  - Voltage and Current signals, generated by a Signal Machine.
- Outputs

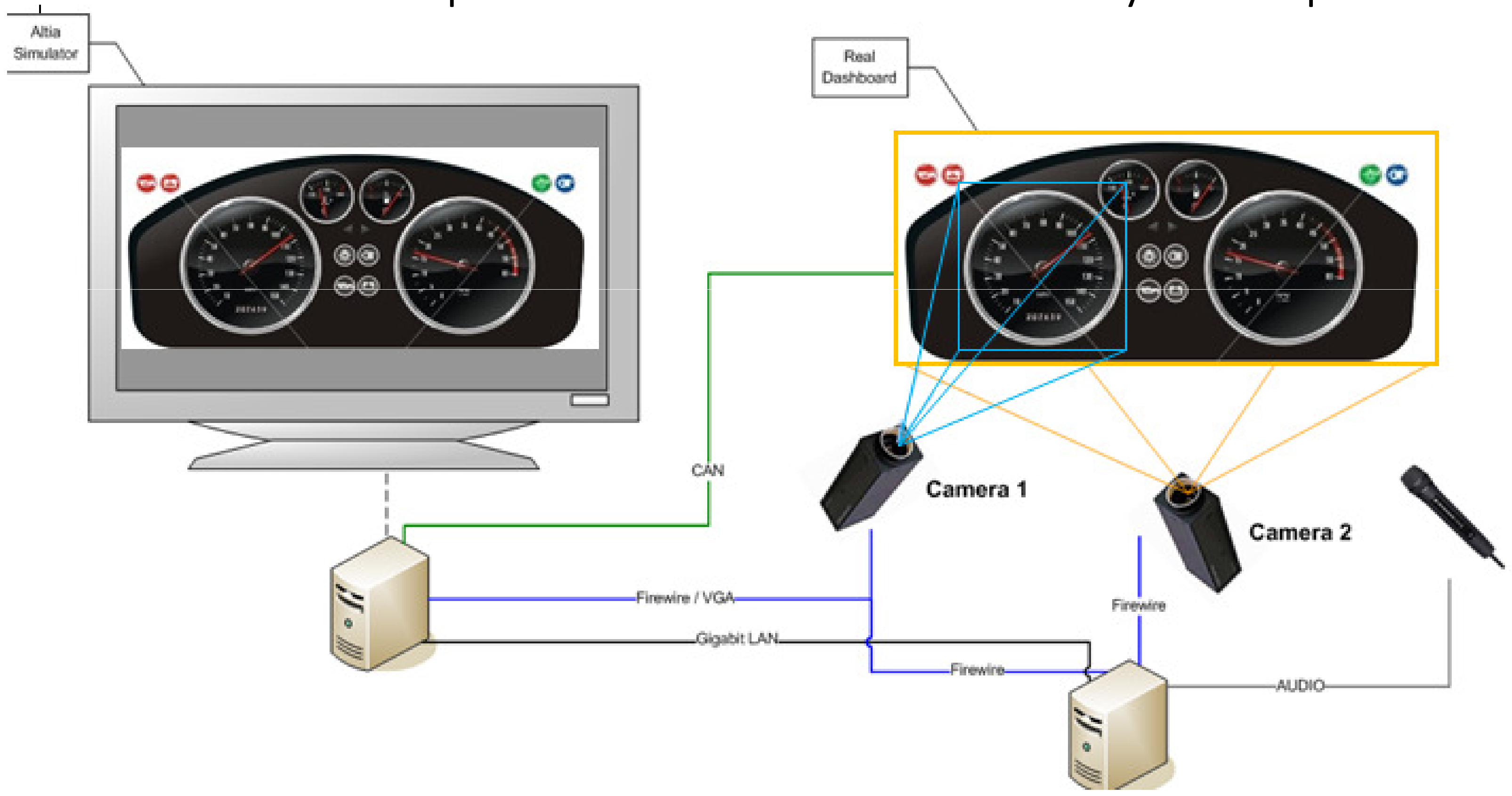

Simulation Results
PC DATALOG
CURRENT / VOLTAGE
ALTIA GRAPHICS (For Non – ISOBUS Application)
CAN (And ISOBUS Application Graphics)
VS
Target Results
BUS DATALOG
CURRENT / VOLTAGE
DISPLAY GRAPHICS (For Non – ISOBUS Application)
CAN (And ISOBUS Application Graphics)

Phase 4: final system validation

- Proprietary graphics validation

Model outputs

Final system outputs

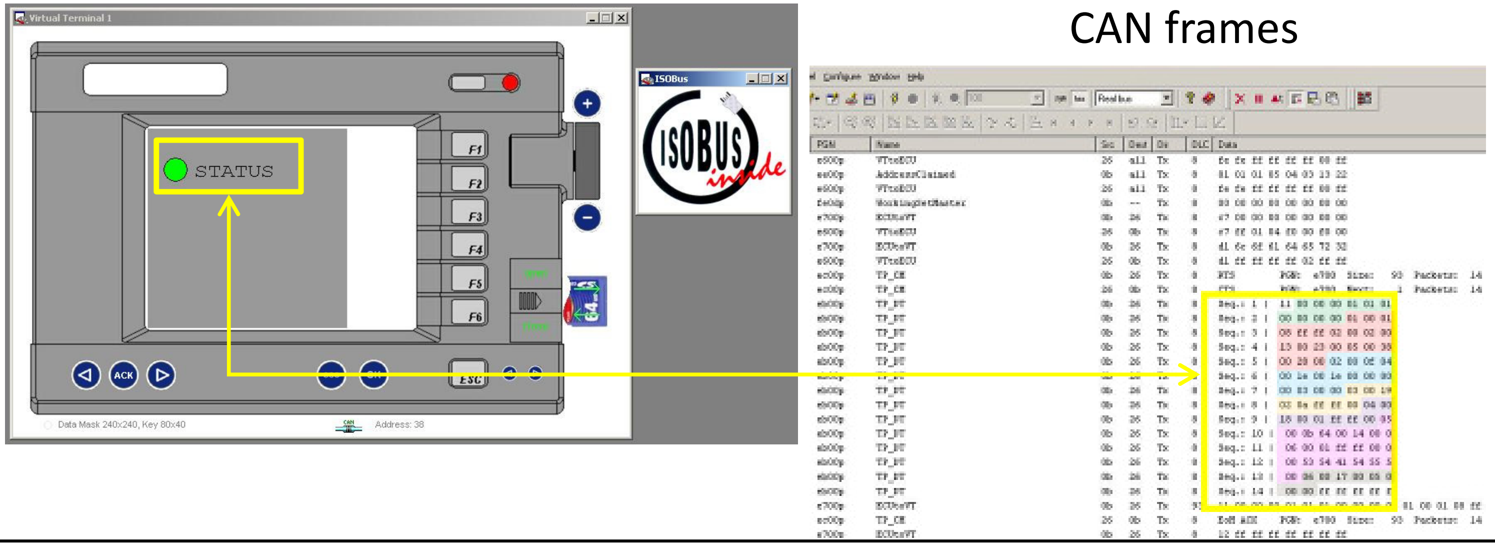


Phase 4: final system validation

- ISOBUS graphics validation
  - Object Pools test is performed by testing the CAN frames through the network bus and verifying if the displayed Object Pool is the correct one with respect to the Standard ISOBUS (ISO 11783).

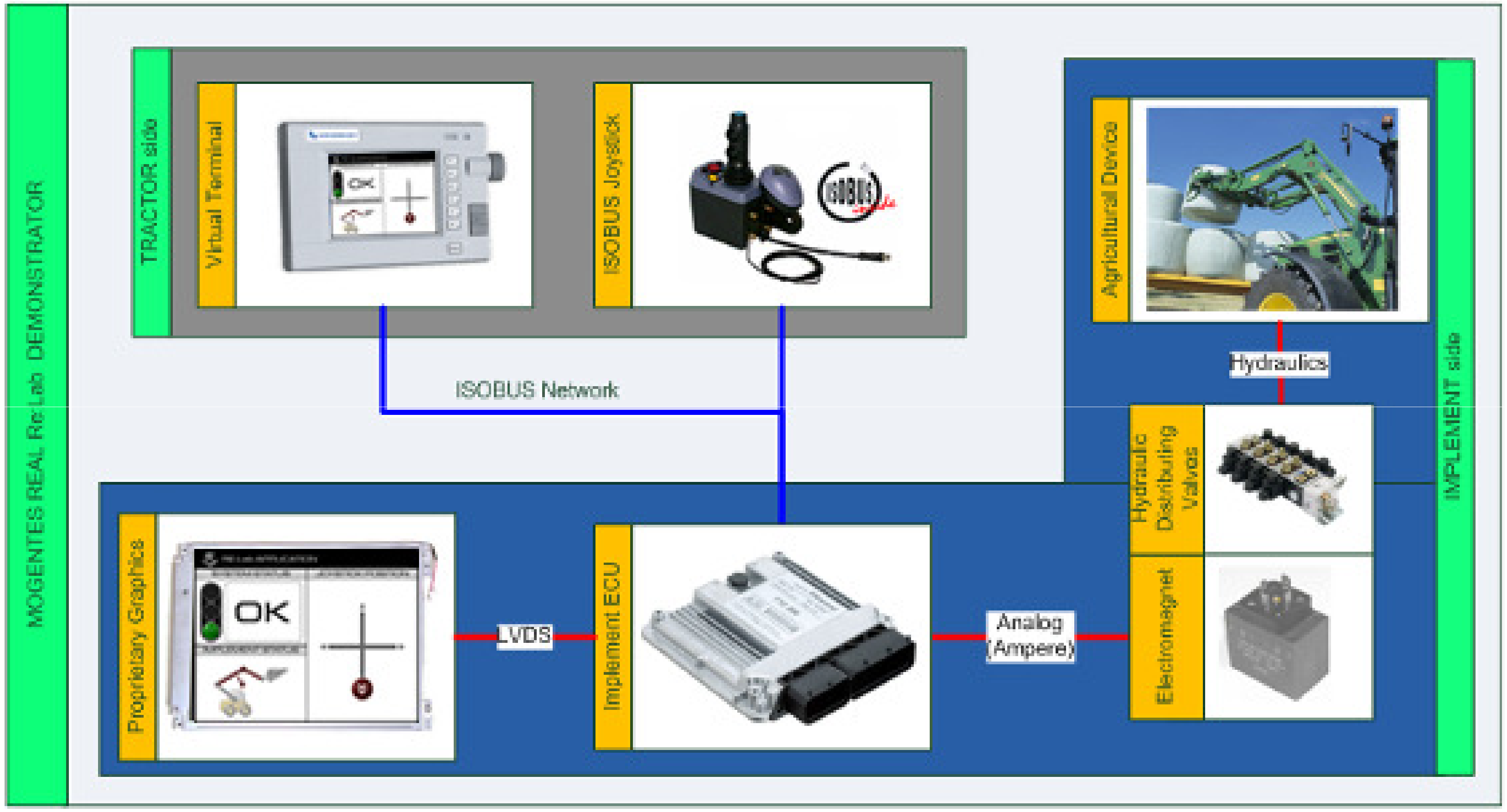
Object pool (highlighted graphic element)

CAN frames



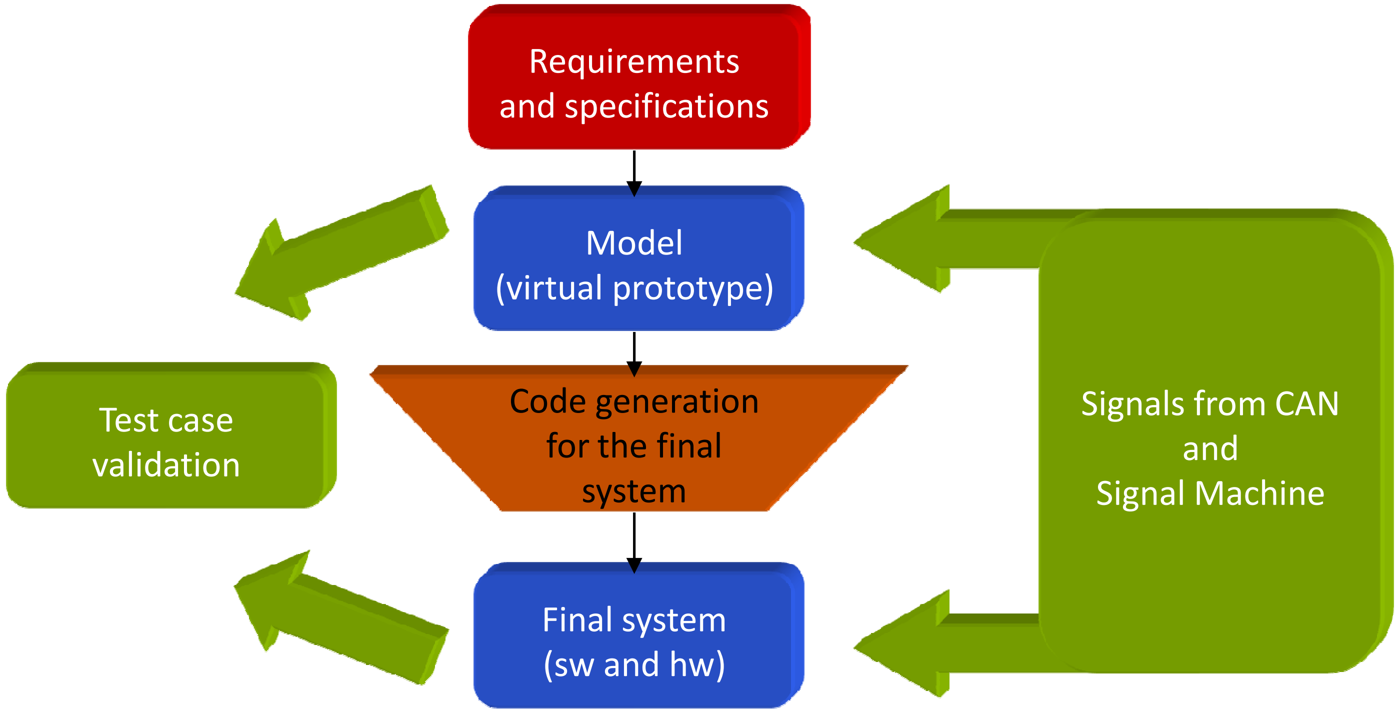
```
PGN     Name               Src  Dest  Dir  DLC  Data
e600p   VTtoECU            26   all   Tx   8    fe fe ff ff ff ff 00 ff
ee00p   AddressClaimed     0b   all   Tx   8    81 01 01 85 04 03 13 22
e600p   VTtoECU            26   all   Tx   8    fe fe ff ff ff ff 00 ff
fe0Dp   WorkingSetMaster   0b   --    Tx   8    00 00 00 00 00 00 00 00
e700p   ECUtoVT            0b   26    Tx   8    c7 00 00 00 00 00 00 00
e600p   VTtoECU            26   0b    Tx   8    c7 ff 01 04 f0 00 f0 00
e700p   ECUtoVT            0b   26    Tx   8    d1 6c 6f 61 64 65 72 32
e600p   VTtoECU            26   0b    Tx   8    d1 ff ff ff ff 02 ff ff
ec00p   TP_CM              0b   26    Tx   8    RTS      PGN: e700  Size:  93  Packets: 14
ec00p   TP_CM              26   0b    Tx   8    CTS      PGN: e700  Next:   1  Packets: 14
eb00p   TP_DT              0b   26    Tx   8    Seq.: 1  | 11 00 08 00 01 01 01
eb00p   TP_DT              0b   26    Tx   8    Seq.: 2  | 00 00 08 00 01 00 01
eb00p   TP_DT              0b   26    Tx   8    Seq.: 3  | 08 ff ff 02 00 02 00
eb00p   TP_DT              0b   26    Tx   8    Seq.: 4  | 13 00 23 00 05 00 36
eb00p   TP_DT              0b   26    Tx   8    Seq.: 5  | 00 28 08 02 00 0f 04
eb00p   TP_DT              0b   26    Tx   8    Seq.: 6  | 00 1e 08 1e 00 00 00
eb00p   TP_DT              0b   26    Tx   8    Seq.: 7  | 00 03 08 00 03 00 19
eb00p   TP_DT              0b   26    Tx   8    Seq.: 8  | 02 0a ff ff 00 04 00
eb00p   TP_DT              0b   26    Tx   8    Seq.: 9  | 18 00 01 ff ff 00 05
eb00p   TP_DT              0b   26    Tx   8    Seq.: 10 | 00 0b 64 00 14 00 0
eb00p   TP_DT              0b   26    Tx   8    Seq.: 11 | 06 00 01 ff ff 00 0
eb00p   TP_DT              0b   26    Tx   8    Seq.: 12 | 00 53 54 41 54 55 5
eb00p   TP_DT              0b   26    Tx   8    Seq.: 13 | 00 06 00 17 00 00 0
eb00p   TP_DT              0b   26    Tx   8    Seq.: 14 | 00 00 ff ff ff ff f
e700p   ECUtoVT            0b   26    Tx   93   11 00 00 [illegible] 81 00 01 00 ff
ec00p   TP_CM              26   0b    Tx   8    EoM ACK  PGN: e700  Size:  93  Packets: 14
e700p   ECUtoVT            0b   26    Tx   8    12 ff ff ff ff ff ff ff
```

Final validated system



Conclusions

- MOGENTES: automatic generation of test cases to enhance testing and verification of dependable embedded systems
- Target RE:Lab application: agricultural machine bucket
- Framework for test and validation


Requirements and specifications
Model (virtual prototype)
Code generation for the final system
Final system (sw and hw)
Test case validation
Signals from CAN and Signal Machine

Contacts

re:LAB
INTERACTION ENGINEERING

www.re-lab.it
s.marzani@re-lab.it

MOGENTES

www.mogentes.eu