WILLERT.
pioneers in embedded software engineering

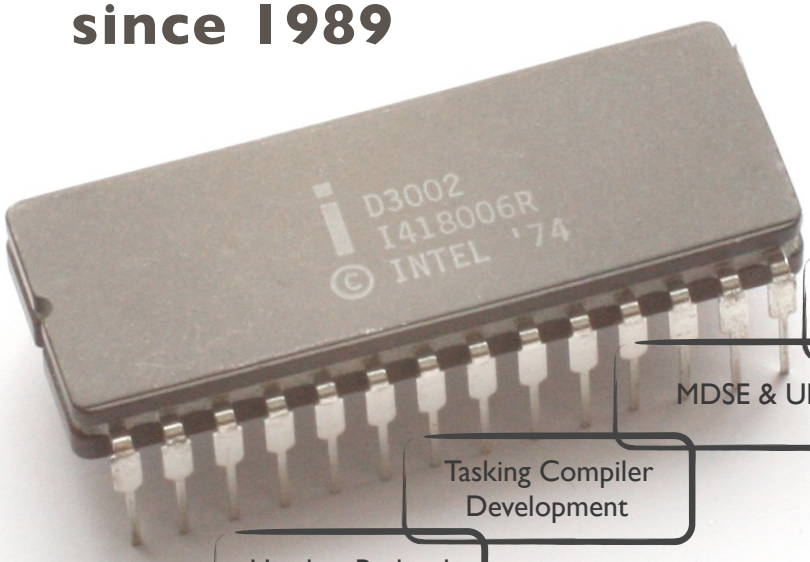
UML MODELING VS CODING

WALTER VAN DER HEIDEN
WILLERT SOFTWARE TOOLS

ABOUT WILLERT SOFTWARE TOOLS

- THE WILLERT SOFTWARE TOOLS COMPANY SPECIALIZES SINCE 1992 IN TOOLS FOR SOFTWARE ENGINEERING
- WITH RESPECT TO REALTIME EMBEDDED SYSTEMS. WILLERT HAS SET ITS GOAL TO OFFER THE BEST AVAILABLE TECHNOLOGIES,
- TO ENABLE CUSTOMERS TO SUCCESSFULLY DEVELOP SOFTWARE AND SUPPORT THEM IN ADAPTING THE REQUIRED TOOLS. FOR THIS, WILLERT OFFERS METHODS AND TOOLS PLUS TRAINING AND COACHING BASED ON,
- THOROUGH KNOWLEDGE OF REALTIME EMBEDDED SYSTEMS.
- OUR CUSTOMERS INCLUDE THE AUTOMOTIVE INDUSTRY, AEROSPACE, MEDICAL DEVICES, PUBLIC TRANSPORT MANUFACTURERS, TELECOMMUNICATIONS, ENERGY AND INFRASTRUCTURAL ENVIRONMENT.
- WILLERT SOFTWARE TOOLS HAS HELPED WITH NUMEROUS PRODUCT SPECIFICATION, -DEVELOPMENT AND QUALITY ASSURANCE, FOR EXAMPLE: VEHICLE CONTROL DEVICES, COFFEE MACHINES, FREQUENCY CONVERTERS, SATELLITES, HEARING INSTRUMENTS, DOOR CONTROLS, FIRE ALARMS, X-RAY DEVICES, POSITIONING SYSTEMS AND ROBOTICS.

Embedded Software Experience since 1989



Processes and Standards (26262, Automotive SPICE , 61508 ....)

Req. Engineering

MDSE & UML

Tasking Compiler Development

Hewlett Packard Test & Measurement

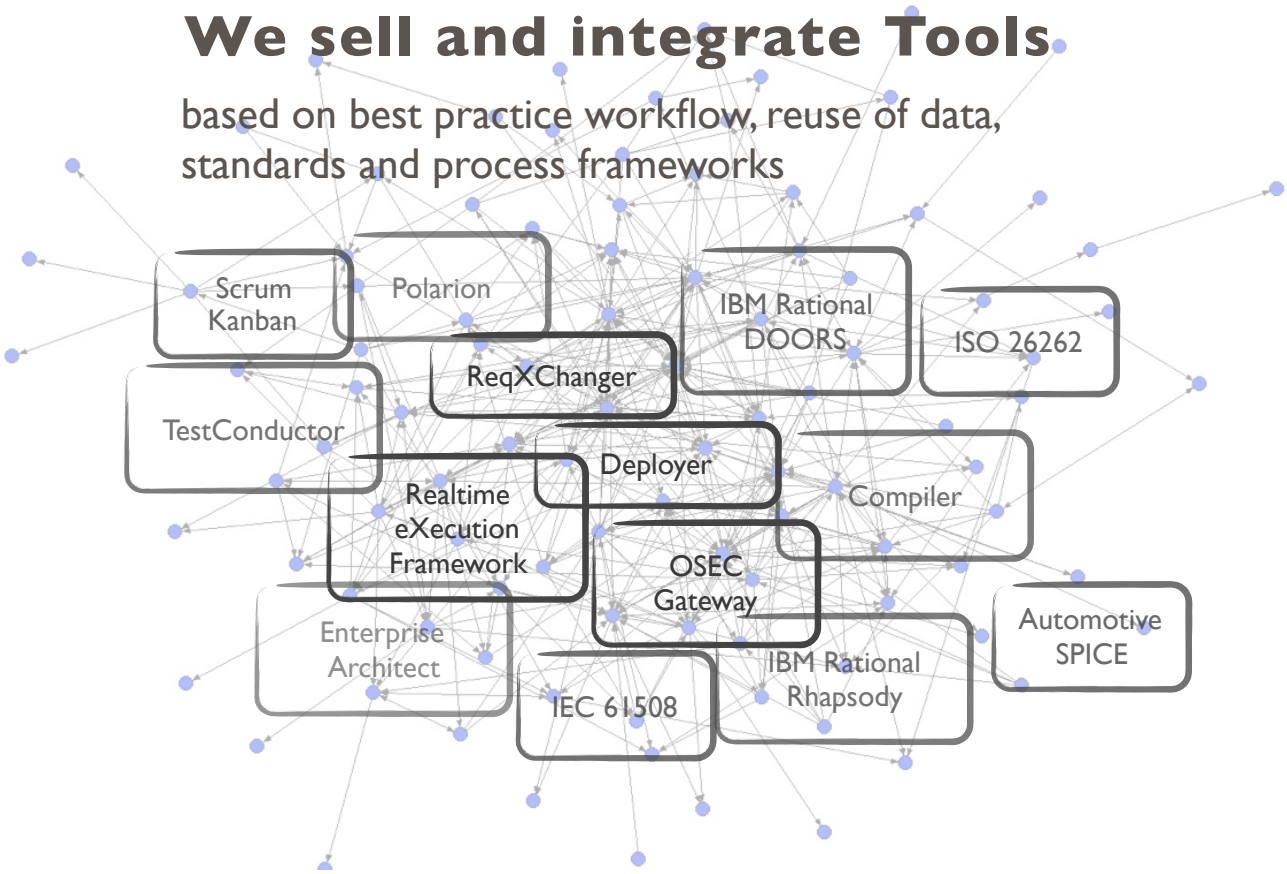
INTEL 8051 Assembler

Quality and Safety

Limited resources

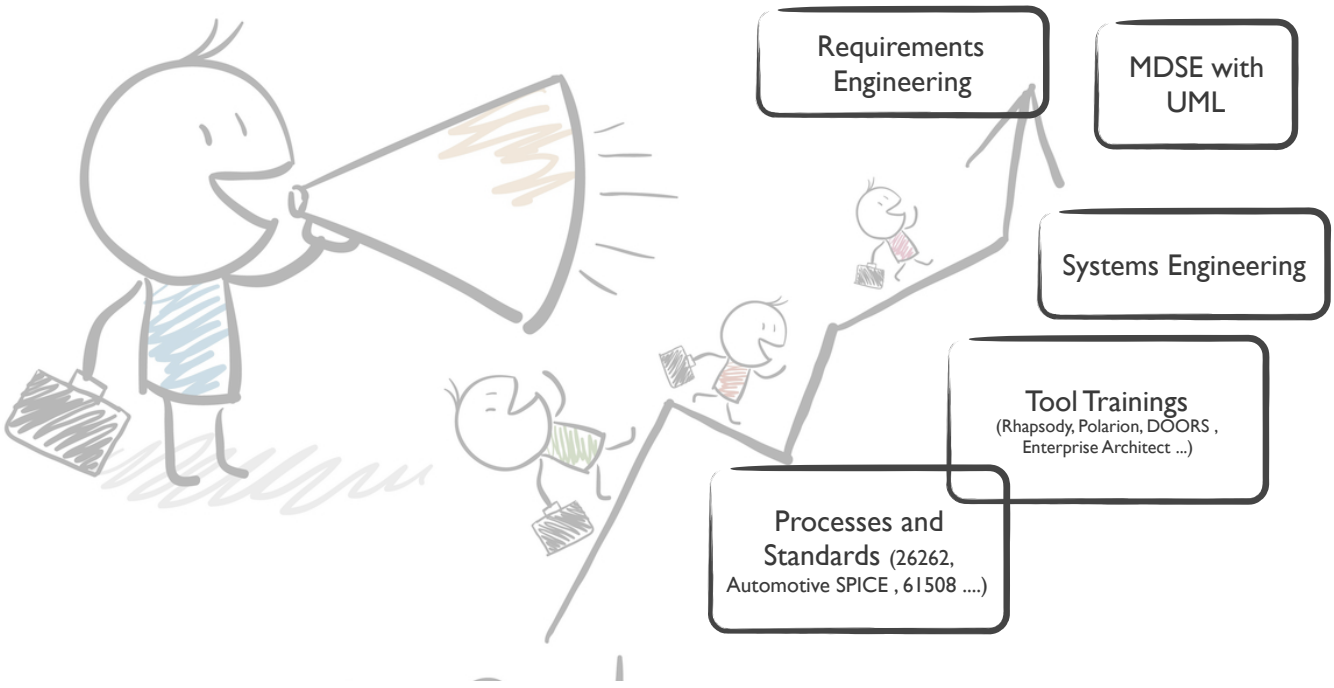
We sell and integrate Tools

based on best practice workflow, reuse of data, standards and process frameworks

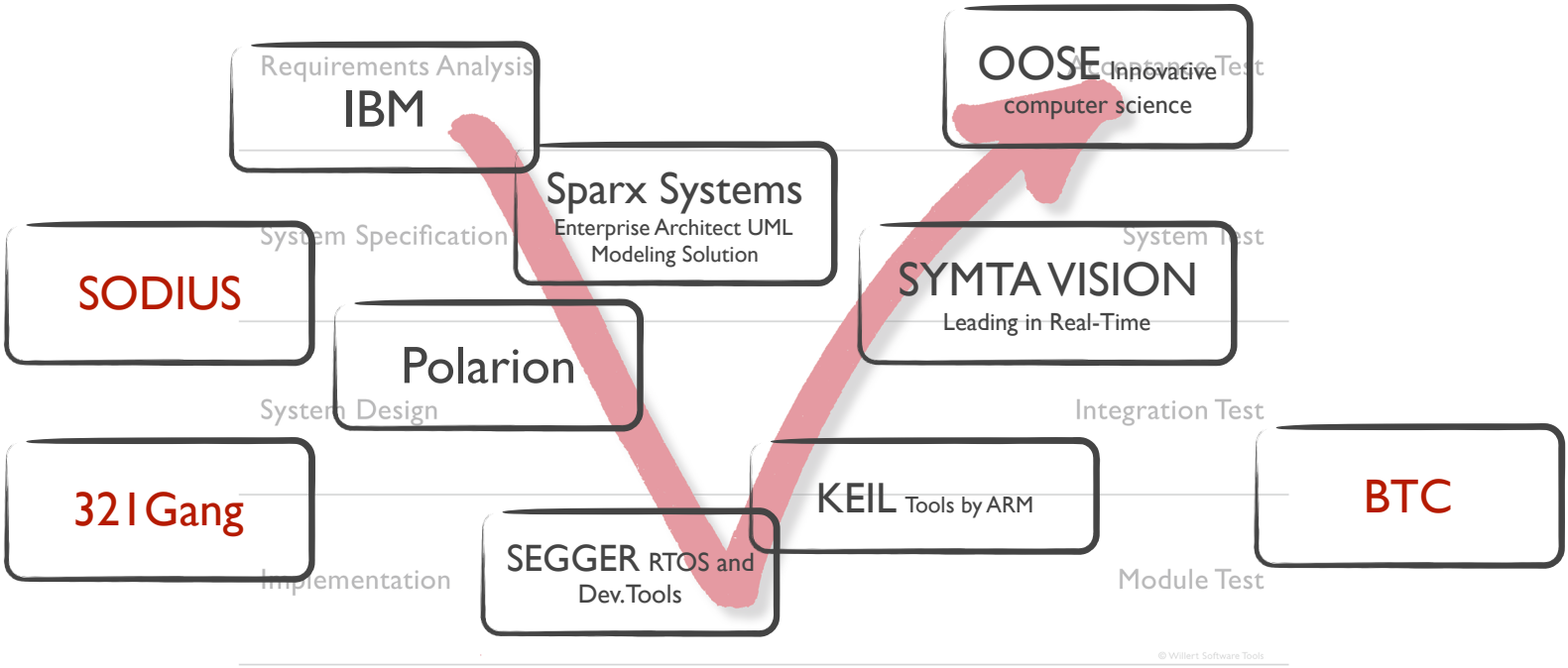


We train Knowledge

in real life and practical situations....

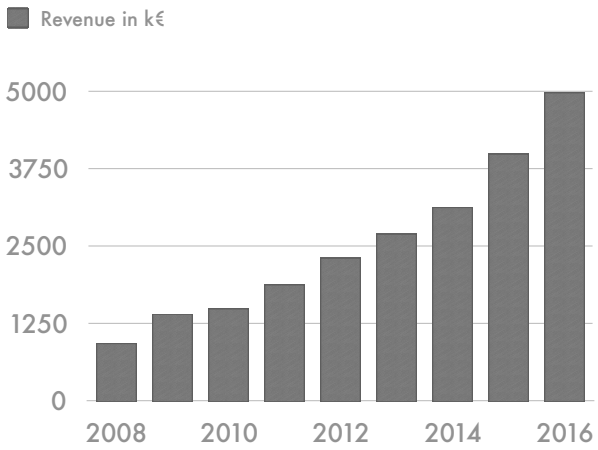


Our Partners



Preferred IBM Partner

Company Data



EUR 5 Million Revenue in 2016

20 Employees (11/2017)

Our Successful Customers

What is modeling?

SOURCE:
HTTP://WWW.DICTIONARY.COM

modeling

mod·el·ing [**mod**-l-ing]

noun

- the act, art, or profession of a person who models.

- the process of producing sculptured form with some plastic material, as clay.

- the technique of rendering the illusion of volume on a two-dimensional surface by shading.

- the treatment of volume, as the turning of a form, in sculpture.

- the representation, often mathematical, of a process, concept, or operation of a system, oftenimplemented by a computer program.

What is a model?

SOURCE:
HTTP://WWW.DICTIONARY.COM

model

mod·el [mod-l] mod·eled,mod·el·ing or (especially British) mod·elled, mod·el·ling.

noun

- a standard or example for imitation or comparison.

- a representation, generally in miniature, to show the construction or appearance of something.

- an image in clay, wax, or the like, to be reproduced in more durable material.

- a person or thing that serves as a subject for an artist, sculptor, writer, etc.

- a person whose profession is posing for artists or photographers.

verb (used with object)

- to form or plan according to a model.

- to give shape or form to; fashion.

- to make a miniature model of.

- to fashion in clay, wax, or the like.

- to simulate (a process, concept, or the operation of a system), commonly with the aid of a computer.

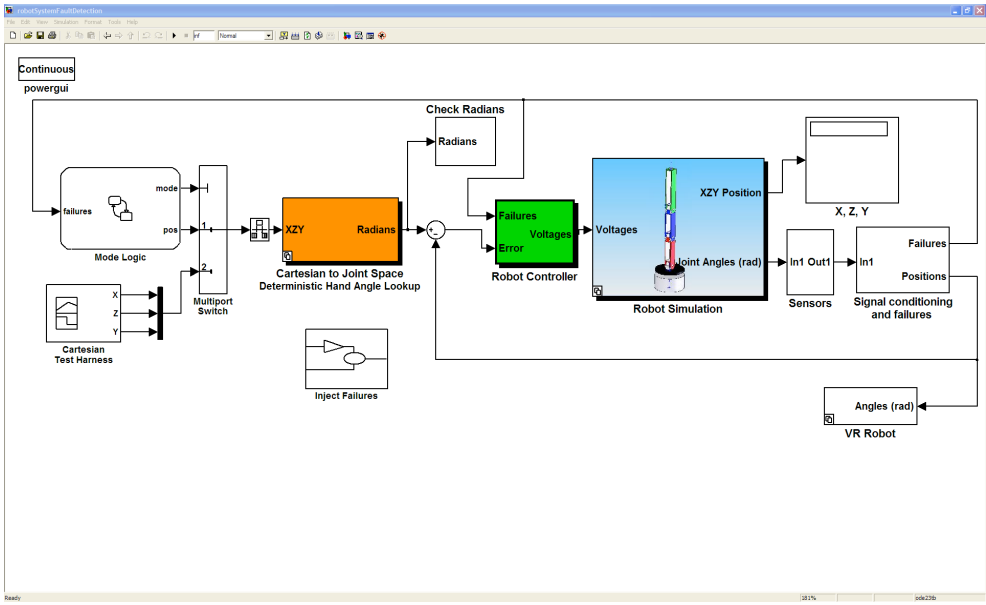
Modeling Tools

The MathWorks MatLab Simulink

MathWorks®

Accelerating the pace of engineering and science

MATLAB®
SIMULINK®

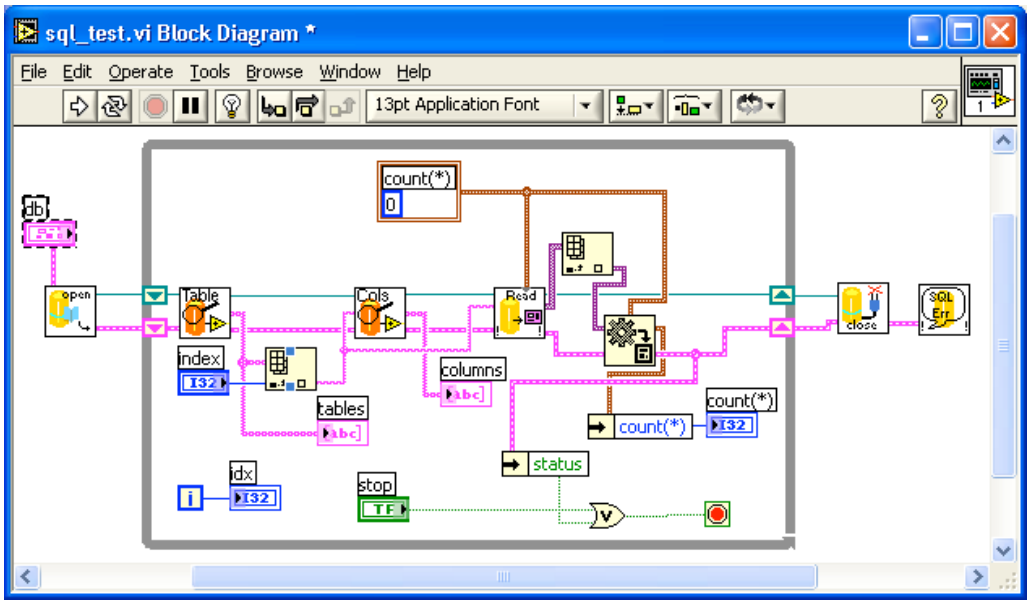


Modeling Tools

National Instruments LabView

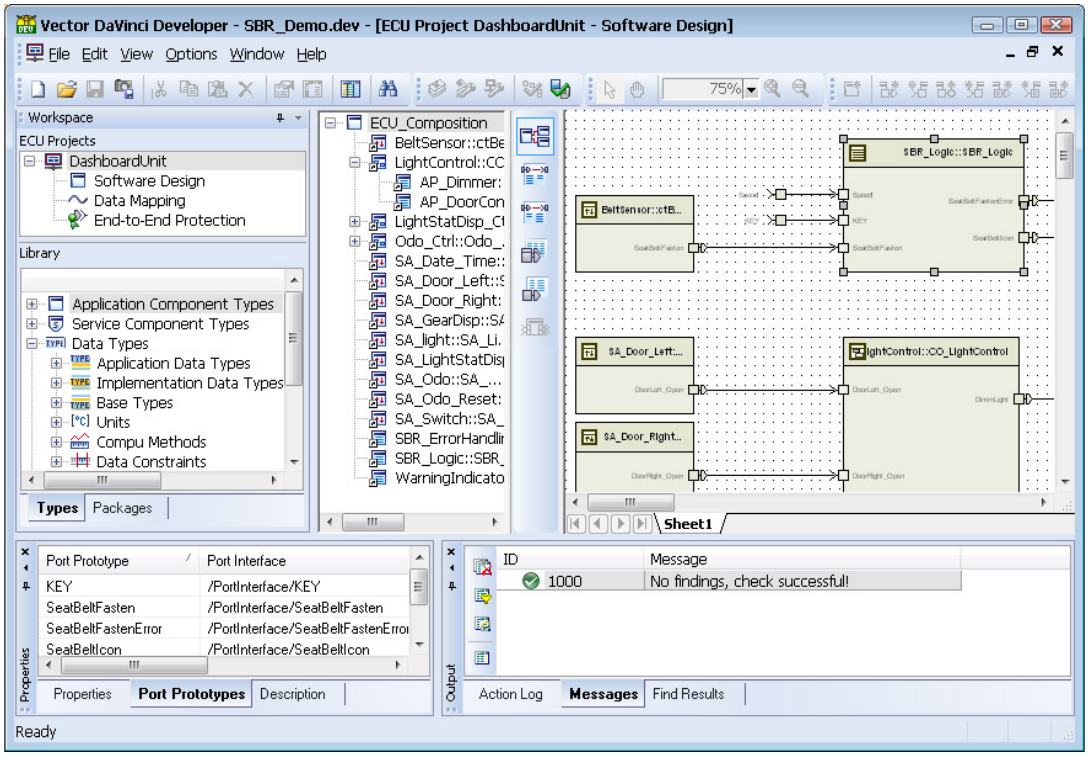
NATIONAL INSTRUMENTS LabVIEW™

Certified Developer



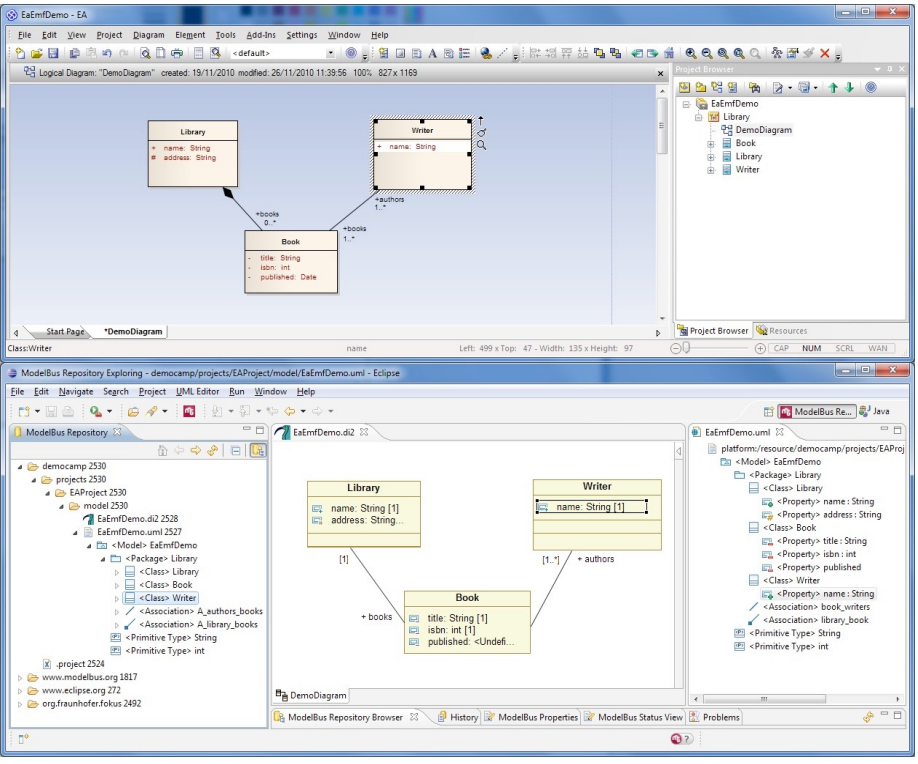
Modeling Tools

AUTOSAR - DaVinci developer



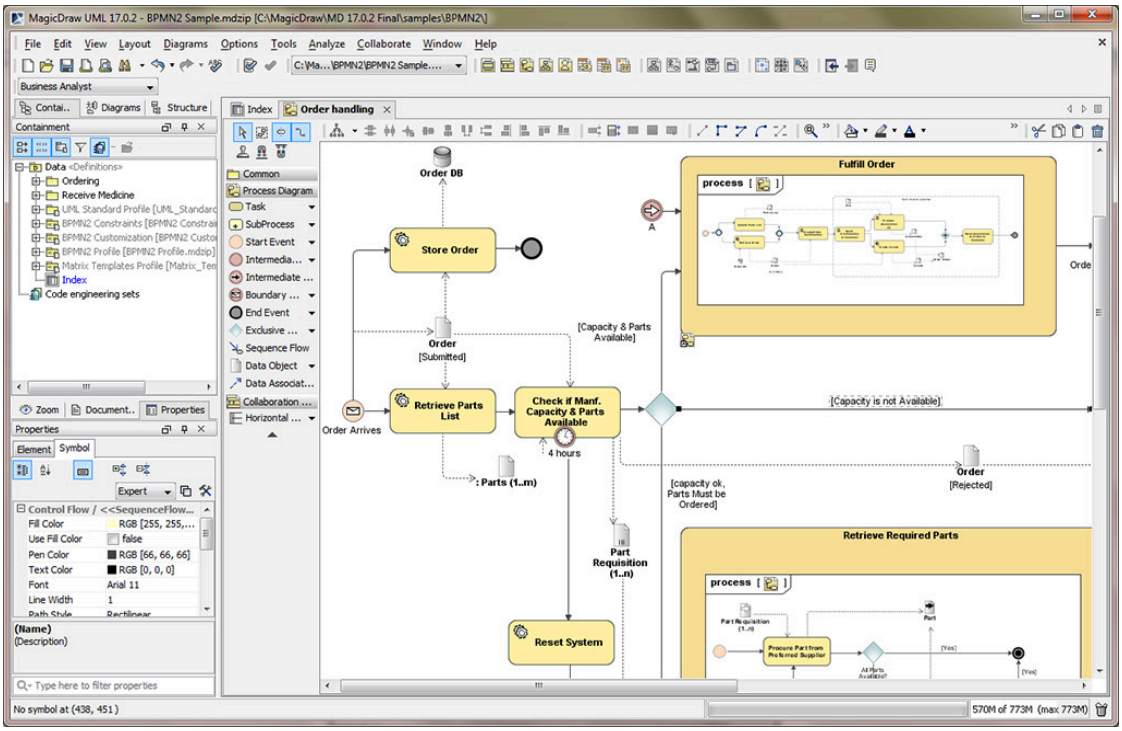
Modeling Tools

UML - Enterprise Architect



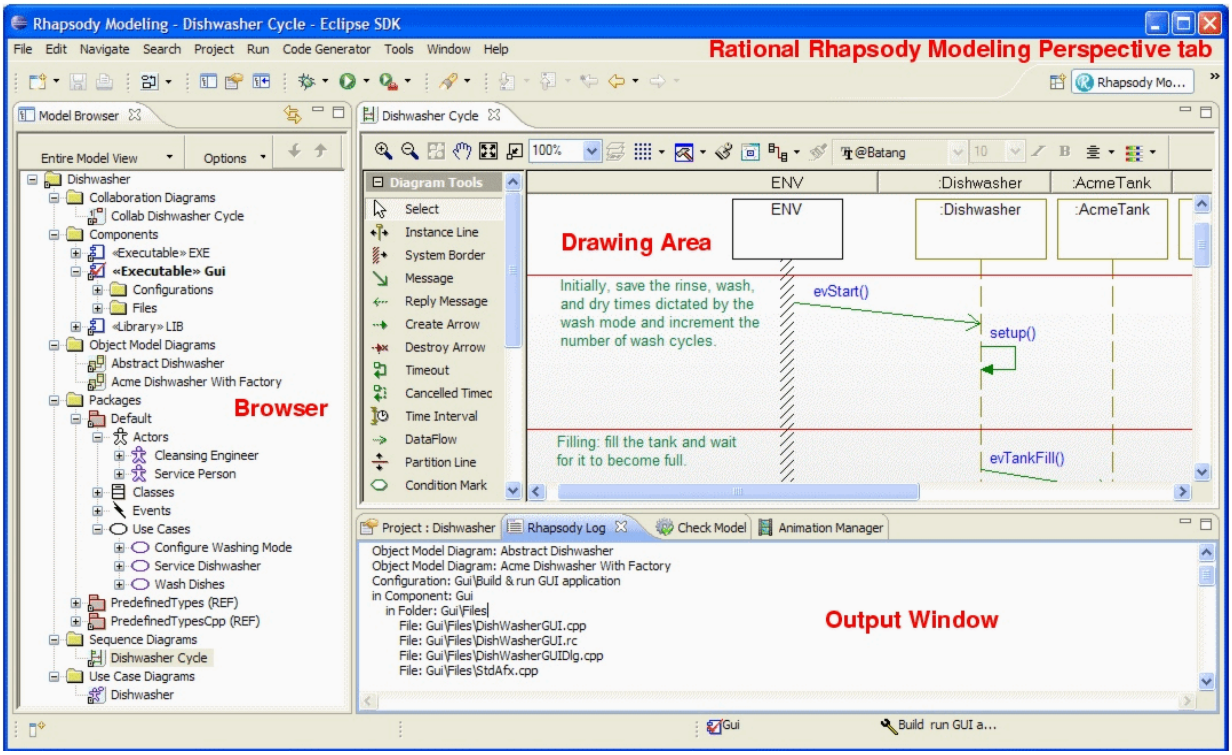
Modeling Tools

No Magic - MagicDraw



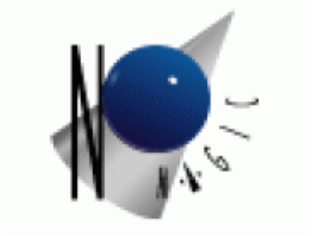
Modeling Tools

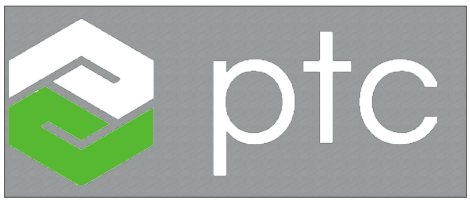
UML - IBM Rational Rhapsody



Modeling Tools

Many many more UML and proprietary tools.





Which tool should I select then?

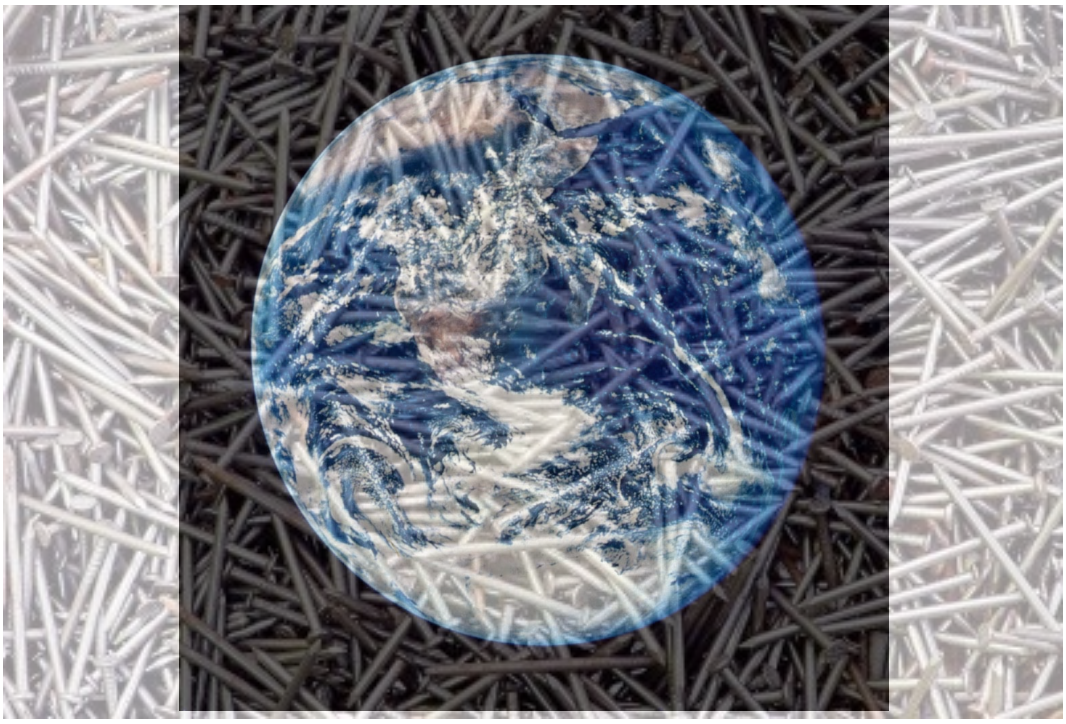
Perhaps you need more then just one tool....

Perhaps you need more then just one tool....

OK.... it'll work if you try....

But using the right tool for the right job is a lot easier..

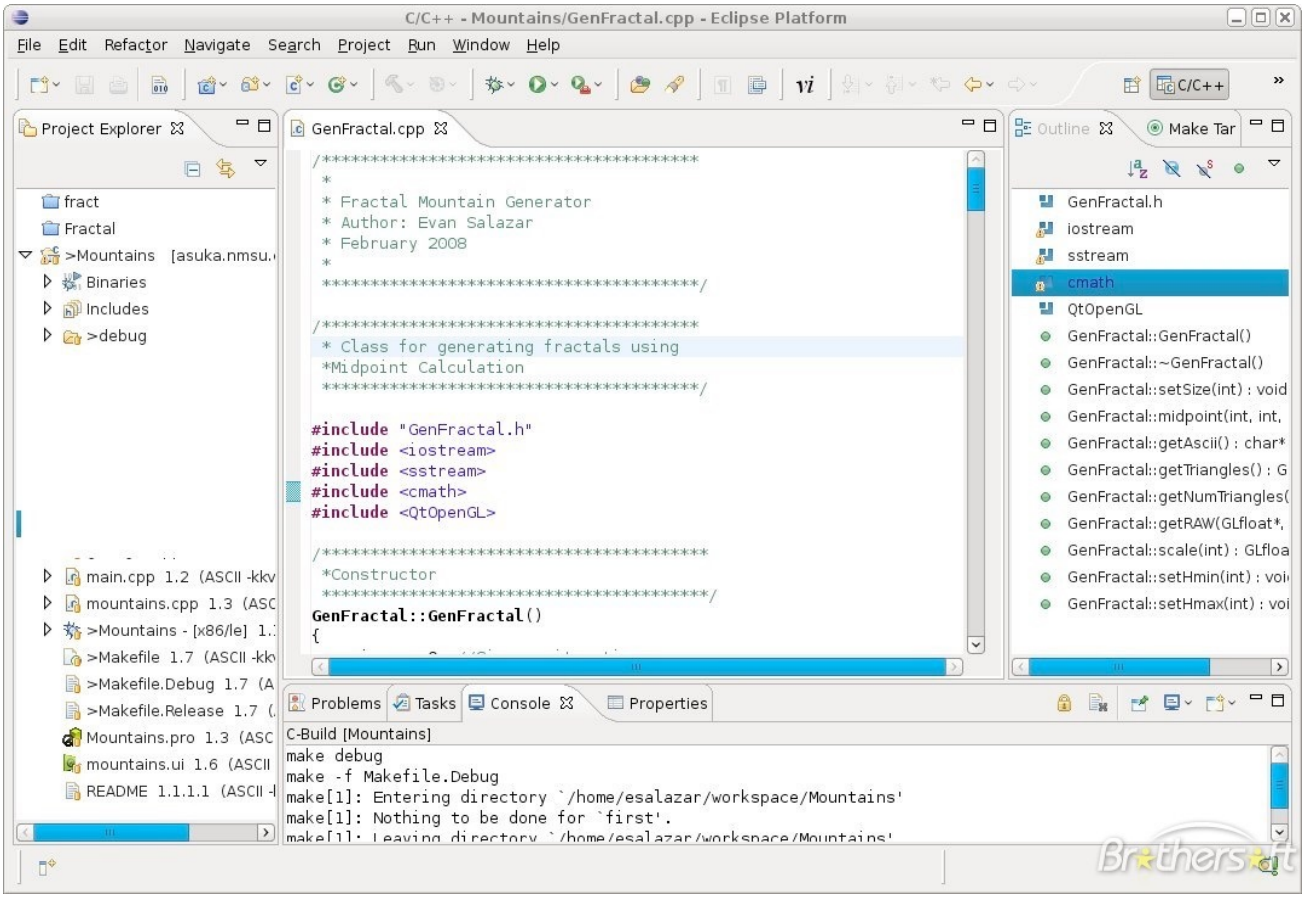
With a hammer in the hand...



The whole world looks like a nail

WHY MODELING?

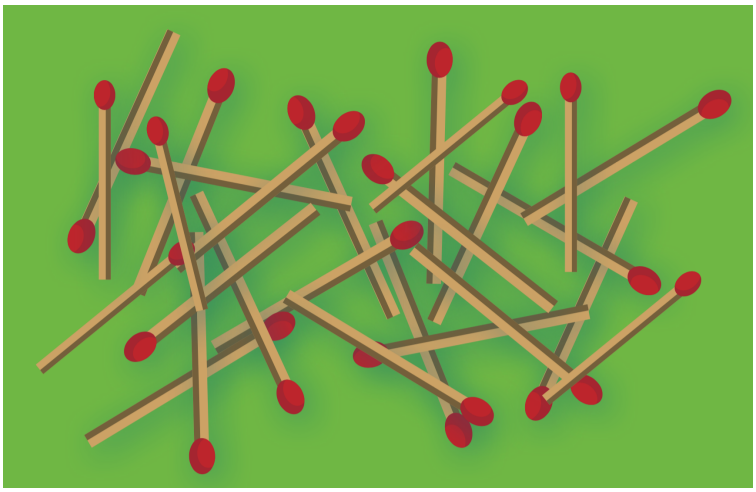
Who needs UML when you have Eclipse?



Better handling of complexity by abstraction

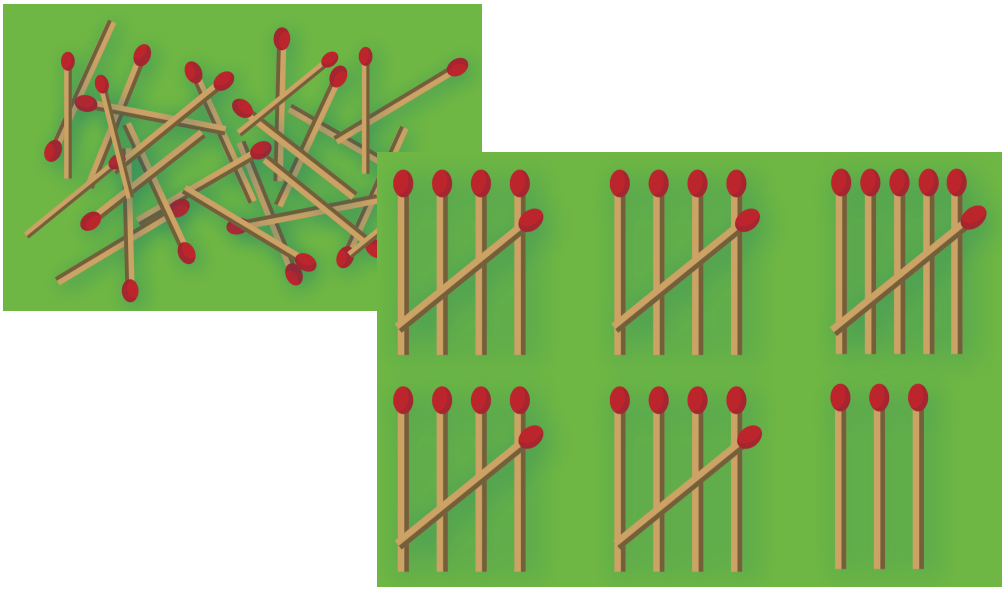
Abstraction

How many matches are in this picture?



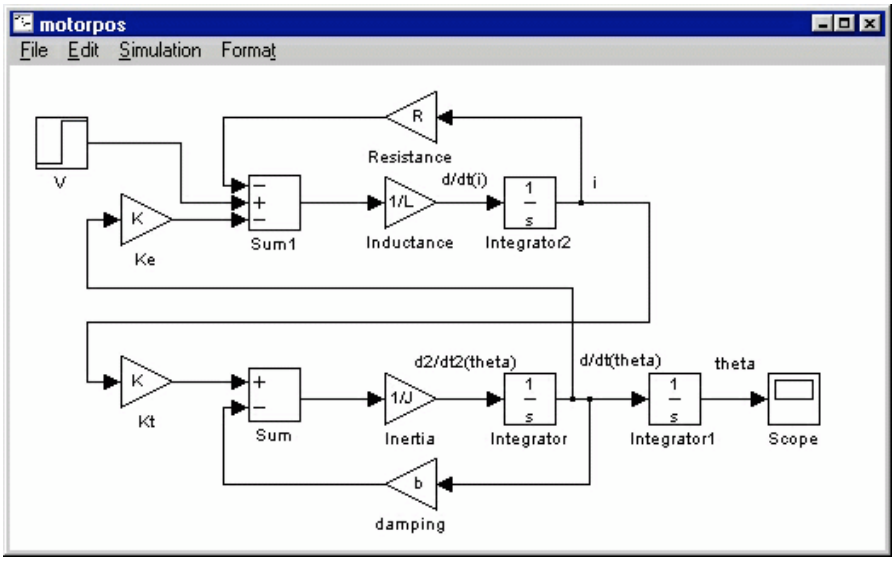
Abstraction

And now?



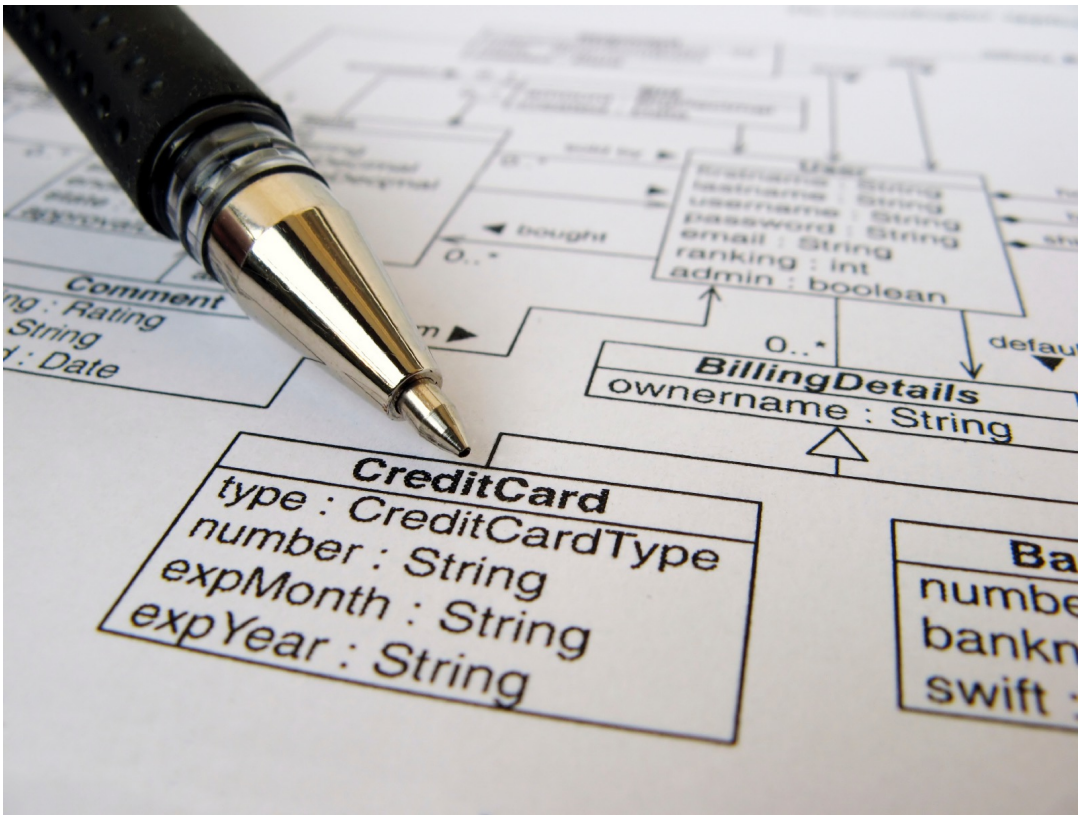
Abstraction

This is something engineers understand immediately



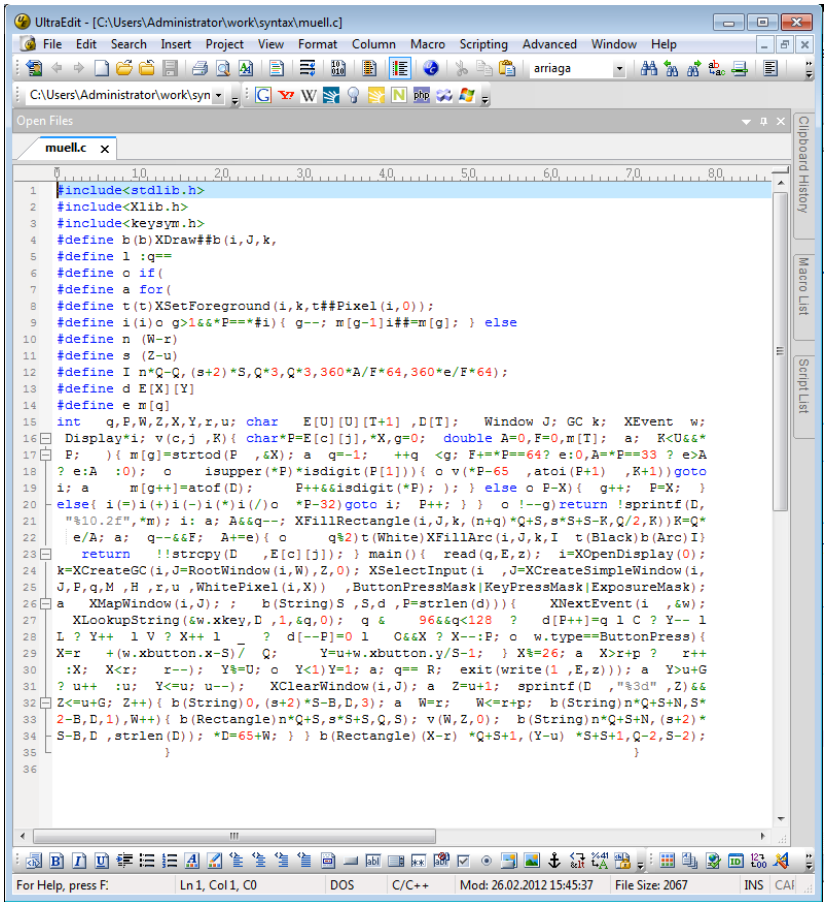
Abstraction

This too



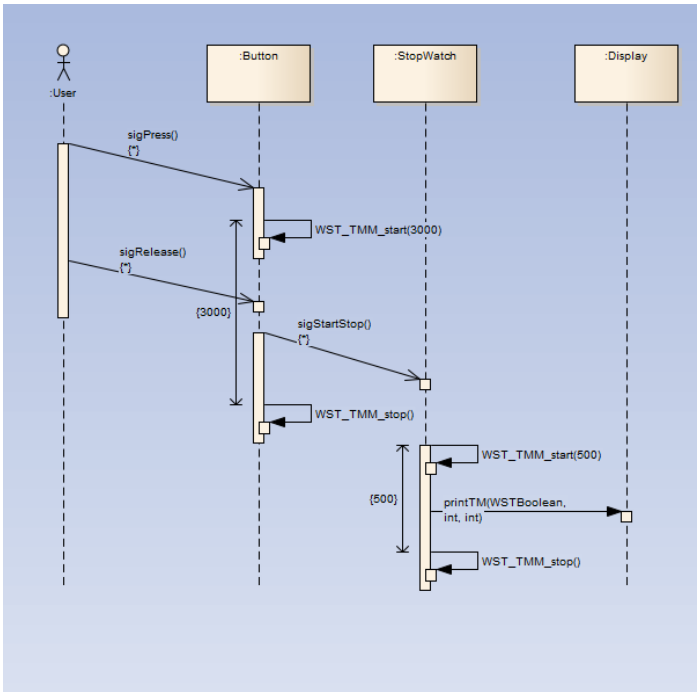
Code

But this is much more difficult to understand.

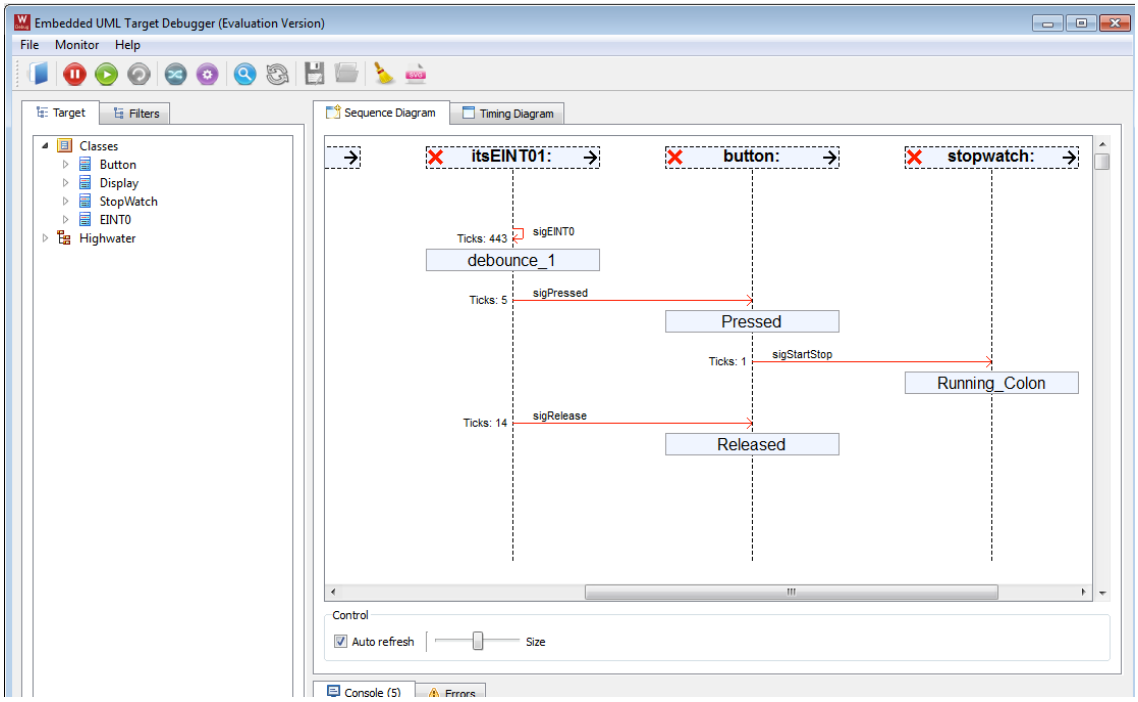


Better communication between developers

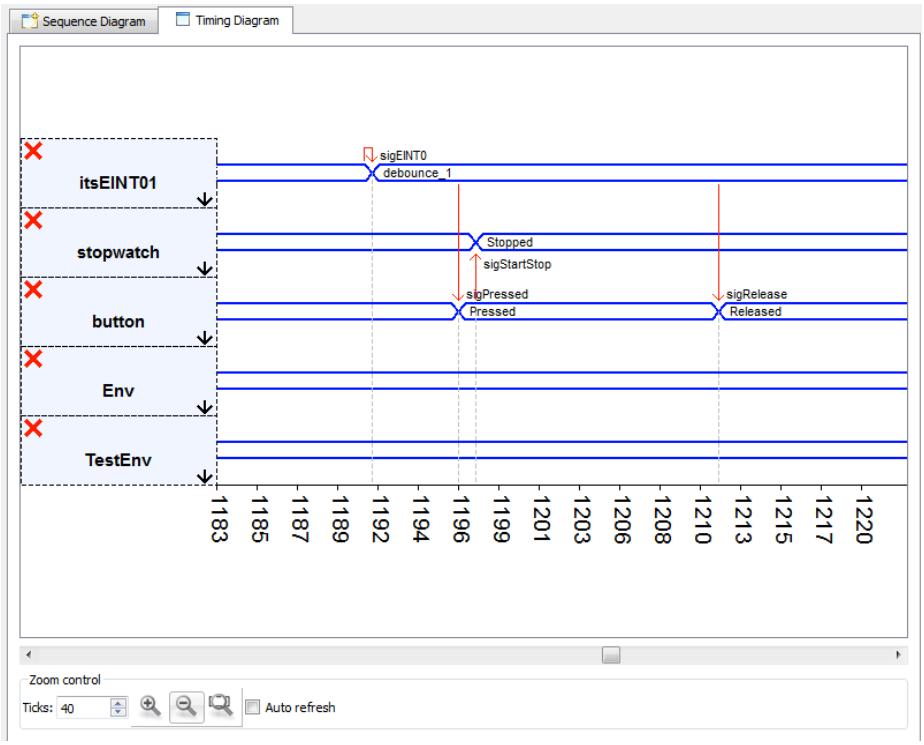
Diagrams to communicate



Diagrams to communicate

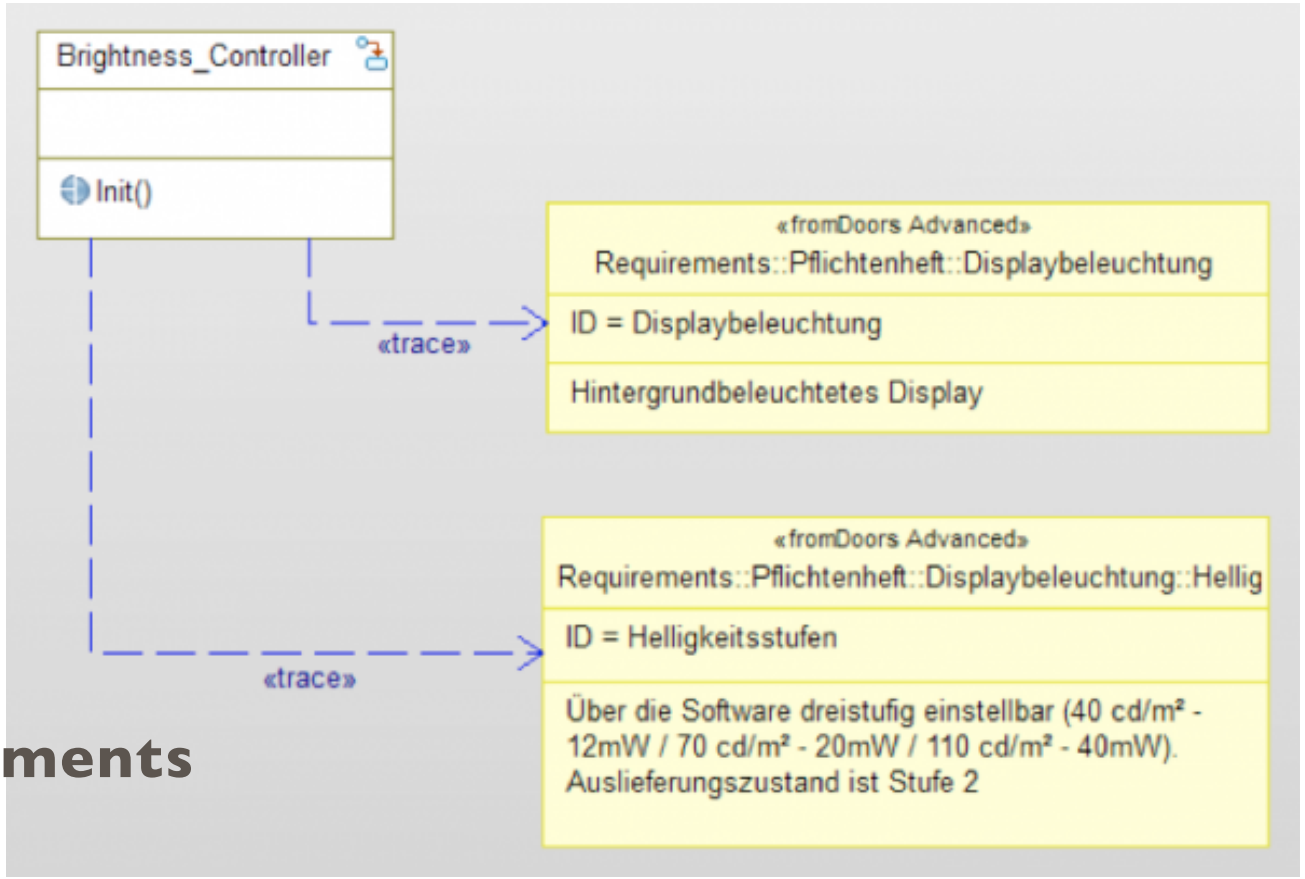


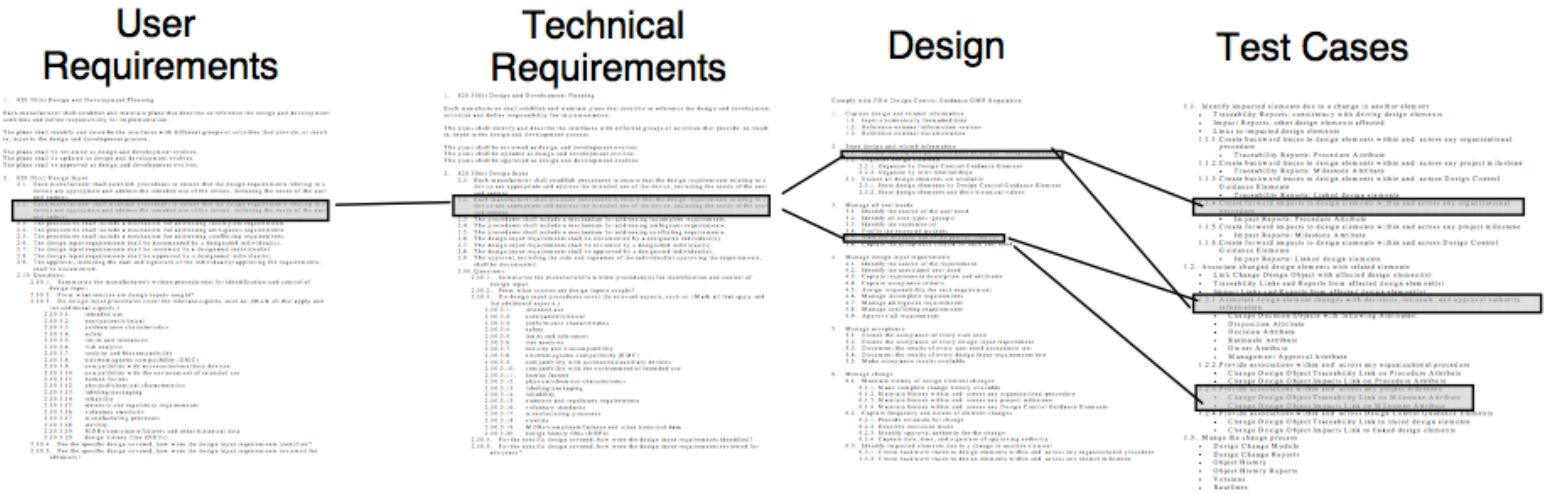
Diagrams to communicate



Link model elements with other information

Requirements

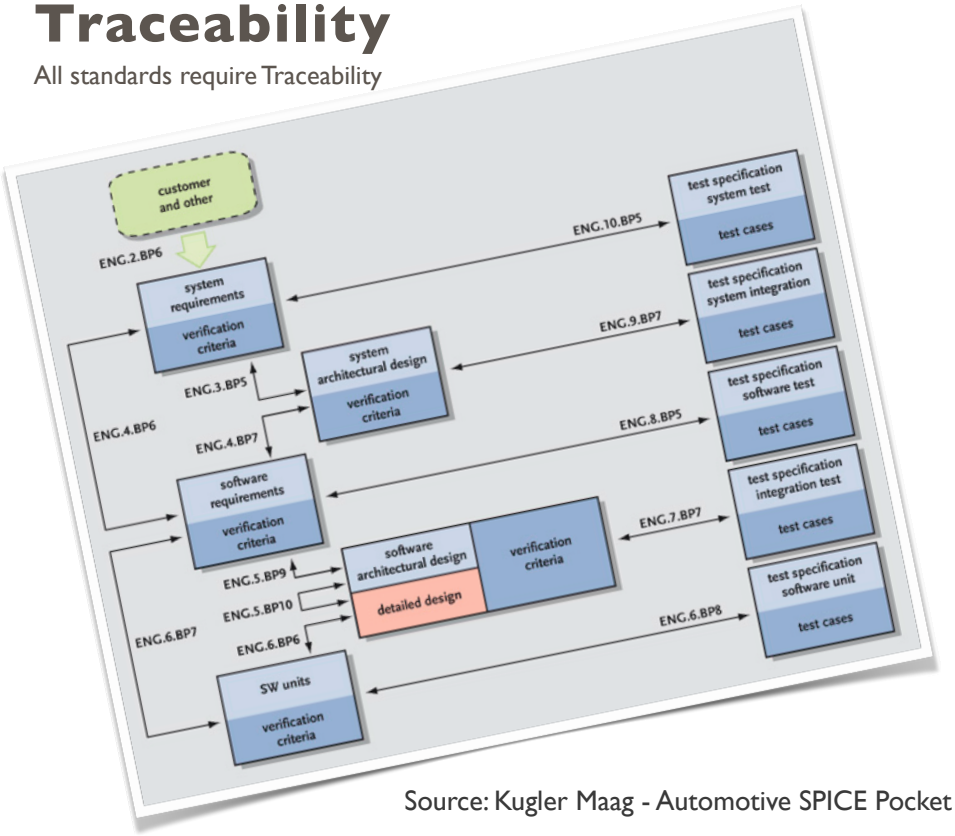




Requirements

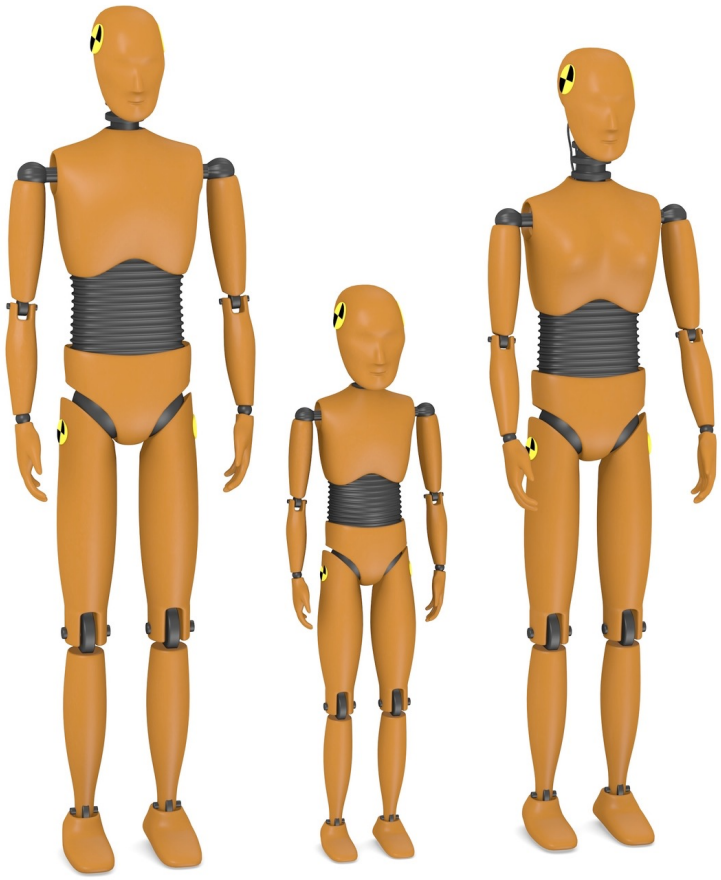
Traceability

All standards require Traceability



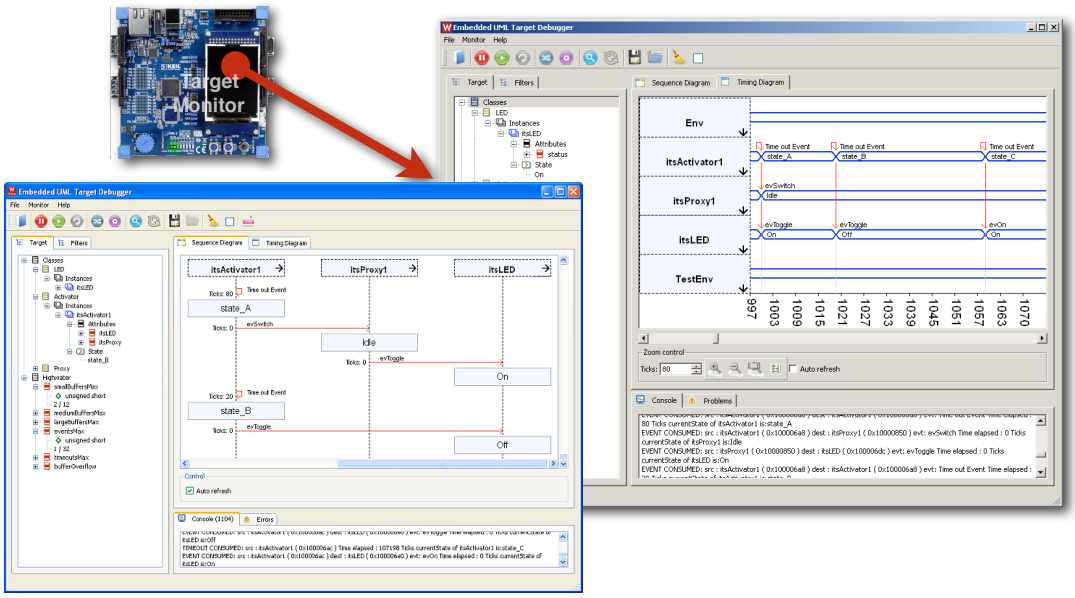
Source: Kugler Maag - Automotive SPICE Pocket Guide

Simulation

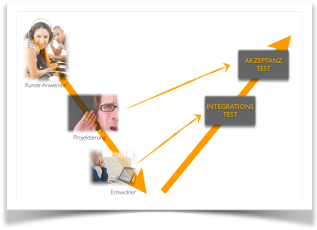


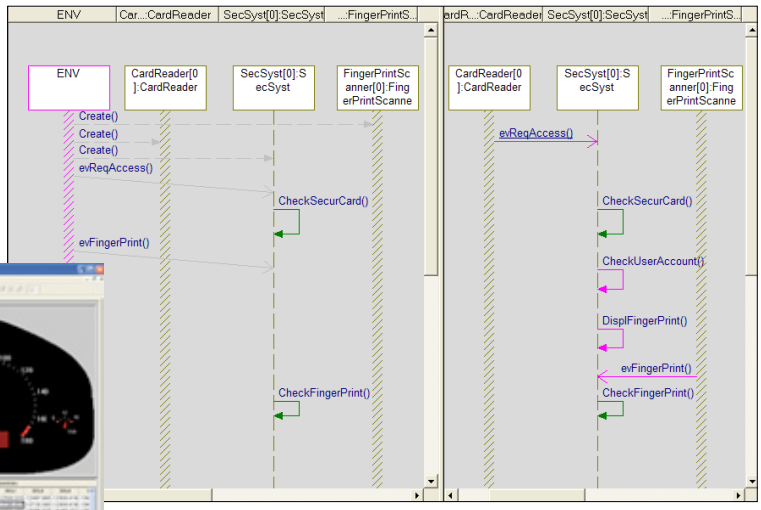
Backannotation

Simulation and Target Execution



MODEL TRANSFORMATION - SIMULATION





ANIMATION AND COMPARE SEQUENCE DIAGRAM

Better Documentation

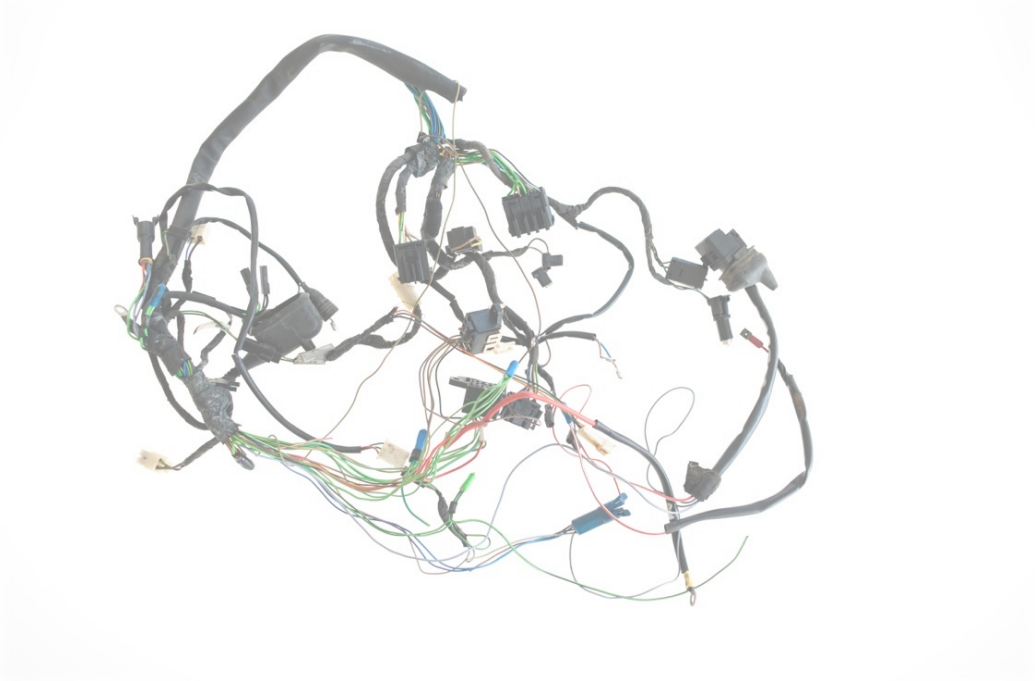
E.G: SW DOCUMENTATION

A: DOCUMENTATION IS OK

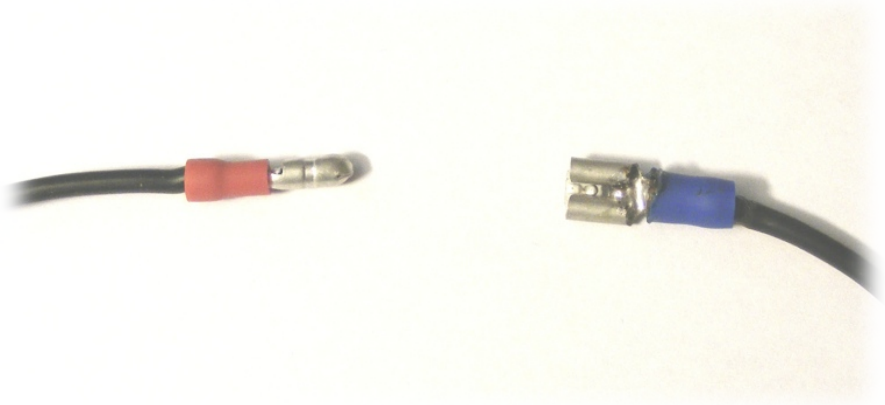
B: DOCUMENTATION IS GENERALLY POORLY

C: DOCUMENTATION DOESN’T REPRESENT THE STATUS OF THE SOFTWARE

Better reUse of Software

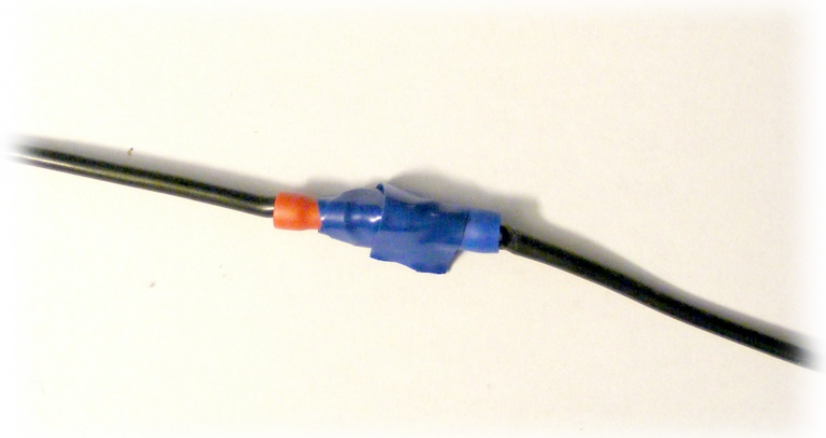


reUse in Software engineering?



reUse in Software engineering?

ReUse in Software engineering?



‚Casting' the duct tape of SW-Engineering

CASTINGS

Castings dienen dazu, die Bedeutung von Daten anders zu interpretieren, bzw. der Programmiersprache zu sagen, dass man sich bewusst ist, was man da eigentlich tut.

–Inhaltsverzeichnis

Als Beispiel:

```
char c = 4;
int i;

i = c;          // kein Problem: denn char (8 Bit) ist vollständig
                //               in int (32 Bit) speicherbar;
c = i;          // Problem:      der Computer kann die 32 Bit des ints nicht
                //               in den 8 Bit des chars speichern:
                //               Es könnten Daten verloren gehen.
```

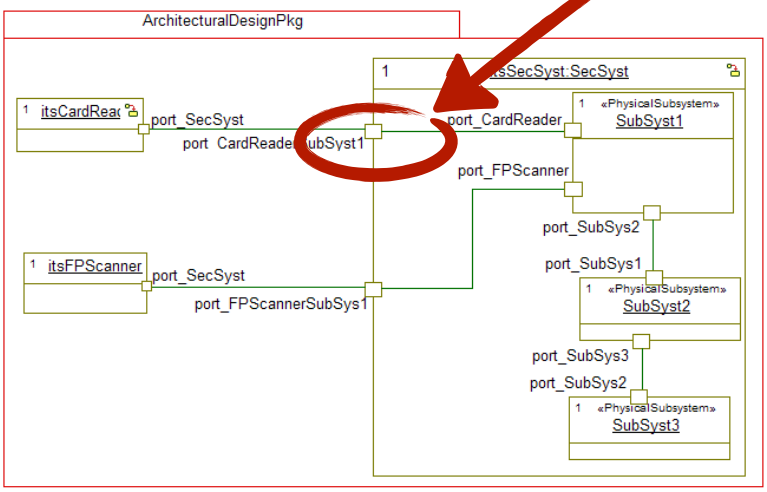
Ein C-Casting löst dieses Problem wie folgt:

```
c = (char) i; // Problem gelöst.
              // Ich behaupte, ich weiß, was ich da tue und deswegen soll der Compiler
              // das nun ohne zu meckern durchführen.
```

Also wird der Compiler nun die niedrigsten 8 Bit des Integers in die Character-Variable kopieren. Sollte die Zahl größer sein, so gehen diese Informationen verloren. Dass das nicht passiert - oder dass das gewollt ist, dafür übernimmt der Entwickler durch das Casting die Verantwortung.

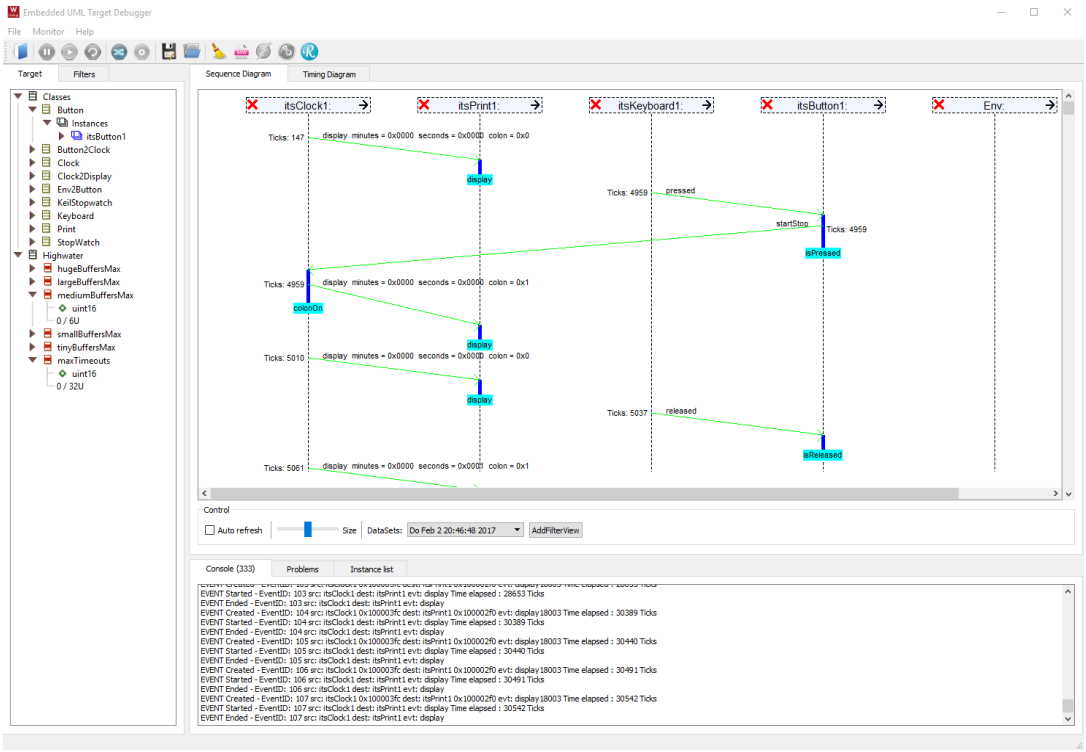
Alternatives

Interfaces based on UML Ports



Code Generation

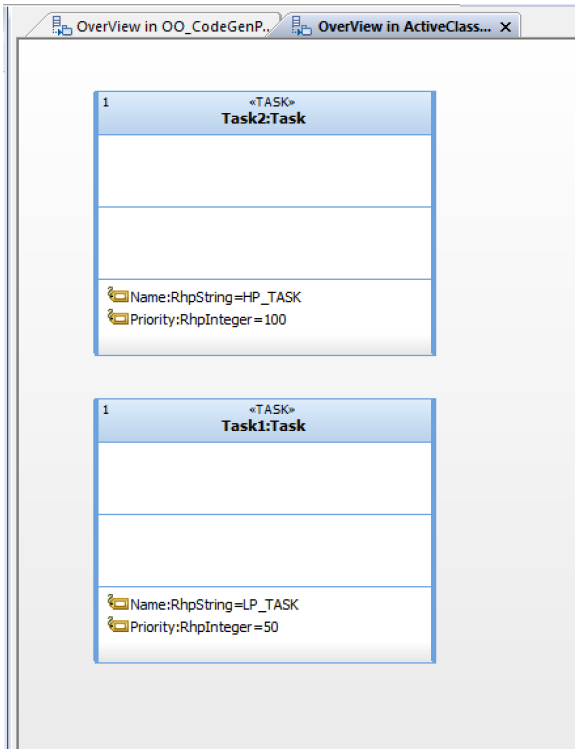
Target execution



Start 2 tasks in C (using Segger embOS)

```
****** main() ********************************************************/
int main(void) {
  OS_IncDI();
  OS_InitKern();
  OS_InitHW();
  /*
/* Initially disable interrupts  */
/* Initialize OS                 */
/* Initialize Hardware for OS    */
  *  Create the extended tasks just as normal tasks.
  *  Note that the first paramater has to be of type OS_TASK
  */
  OS_CREATETASK(&TCBHP.Task, "HP Task", MyTask, 100, StackHP);
  OS_CREATETASK(&TCBLP.Task, "LP Task", MyTask,  50, StackLP);
  /*
  *  Give task contexts individual data
  */
  TCBHP.Timeout = 200;
  TCBHP.pString = "HP task running\n";
  TCBLP.Timeout = 500;
  TCBLP.pString = "LP task running\n";
  OS_Start();                     /* Start multitasking            */
return 0; }
```

Start 2 tasks in UML (using Rhapsody)



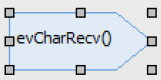
Sending a message in C (using embOS)

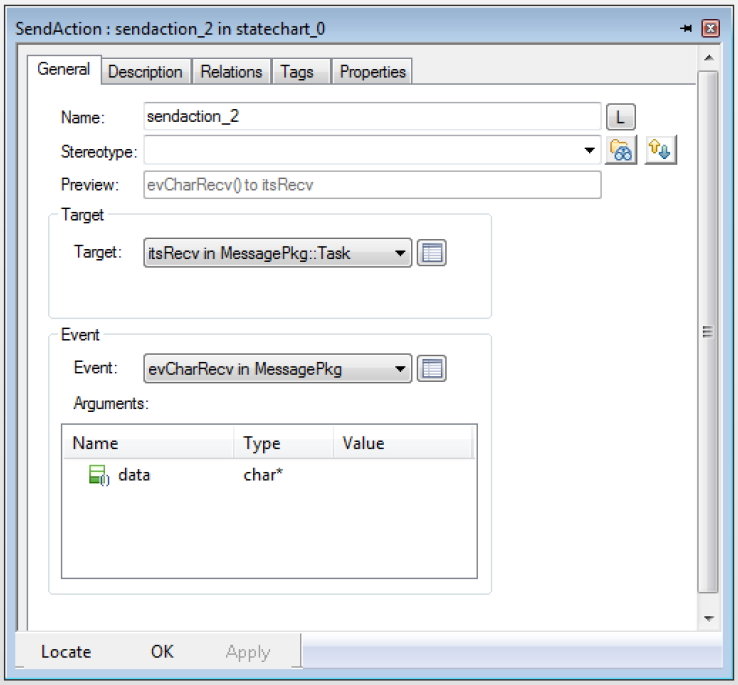
```
OS_MAILBOX MBKey;
     char MBKeyBuffer[6];

void InitKeyMan(void)
{
       /* Create mailbox, functioning as type ahead buffer */
       OS_CreateMB(&MBKey, 1, sizeof(MBKeyBuffer), &MBKeyBuffer);
}

void KEYMAN_StoreKey(char k)
{
       OS_PutMail1(&MBKey, &k);  /* Store key, wait if no space in buffer */
}
```

Sending a message in UML (using Rhapsody)





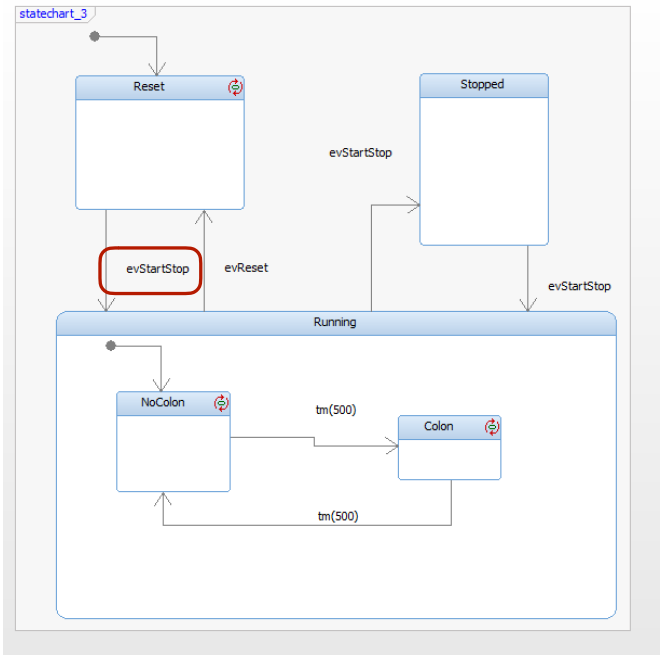
Receiving a message in C (using embOS)

```
OS_MAILBOX MBKey;
     char MBKeyBuffer[6];

void InitKeyMan(void)
{
       /* Create mailbox, functioning as type ahead buffer */
       OS_CreateMB(&MBKey, 1, sizeof(MBKeyBuffer), &MBKeyBuffer);
}

char WaitKey(void) {
       char c;
       OS_GetMail1(&MBKey, &c);
return c; }
```

Receiving a message in UML (using Rhapsody)



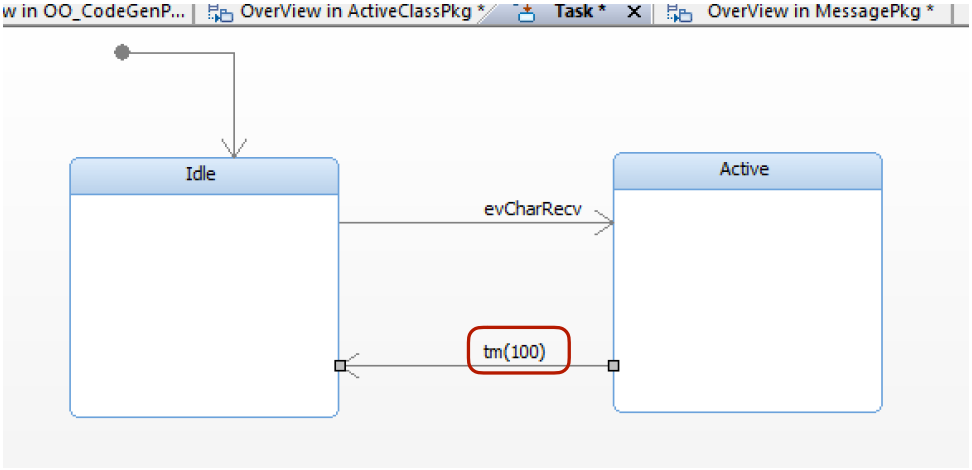
Setting a timer in C (using embOS)

```
OS_TIMER TIMER100;

void Timer100(void)
{
  LED = LED ? 0 : 1;              /* Toggle LED  */
  OS_RetriggerTimer(&TIMER100);  /* Make timer periodical */
}
void InitTask(void)
{
  /* Create and start Timer100  */
  OS_CREATETIMER(&TIMER100, Timer100, 100);
  OS_StartTimer(&TIMER100);

}
```

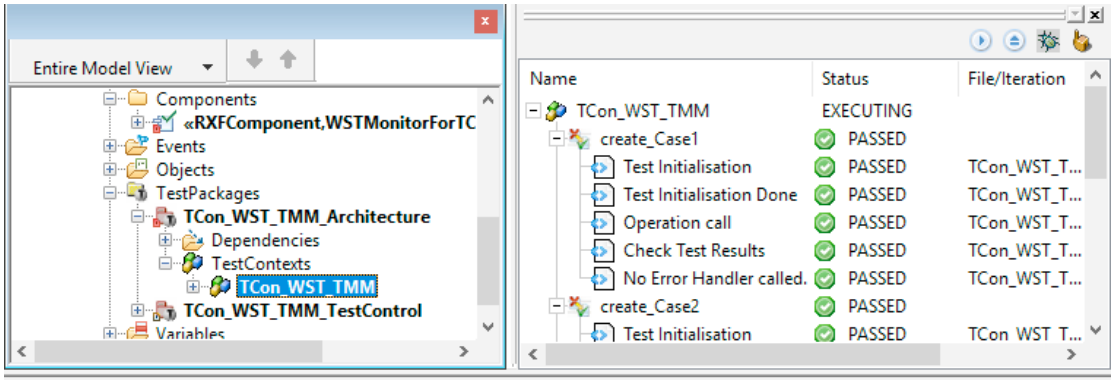
Setting a timout in UML (using Rhapsody)



Automated Test

TestConductor

Test Execution on Target with Feedback on Host (1)



TestConductor

Test Execution on Target with Feedback on Host (2)

Coverage Statistics

	Goals	Covered	
Statement Coverage	80	80	100%
Decision Coverage	32	32	100%
Condition Coverage	8	8	100%
Condition/Decision Coverage	46	46	100%
Modified Condition/Decision Coverage	46	46	100%

Quick Links

Goal Statistics

Statement Coverage

Coverage Goals		
total	80	
covered	80	100%
unknown	0	0%

Coverage Items (1 Goal)		
total	80	
covered (completely)	80	100%
covered (partially)	0	0%
uncovered	0	0%

Detailed Coverage Results

Decision Coverage

Coverage Goals		
total	32	
covered	32	100%
unknown	0	0%

Coverage Items (2 Goals)		
total	16	
covered (completely)	16	100%
covered (partially)	0	0%
uncovered	0	0%

Detailed Coverage Results

Condition Coverage

Coverage Goals		
total	8	
covered	8	100%
unknown	0	0%

Coverage Items (2 Goals)		
total	4	
covered (completely)	4	100%
covered (partially)	0	0%
uncovered	0	0%

Detailed Coverage Results

C/DC and MC/DC

Coverage Goals		
total	46	
covered	46	100%
unknown	0	0%

Function Coverage

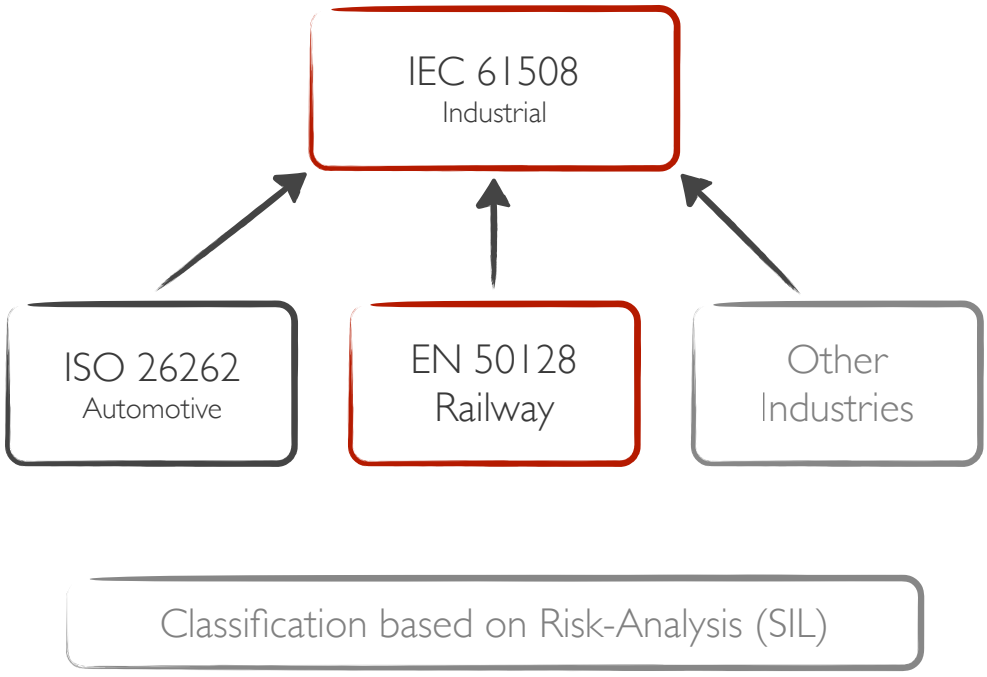
Coverage Goals		
total	6	
covered	6	100%
unknown	0	0%

Switch-Case Coverage

Coverage Goals		
total	0	
covered	0	n.a.
unknown	0	n.a.

IEC 61508 / EN 50128

Standards



AUTOSAR

ABOUT MARQUARDT

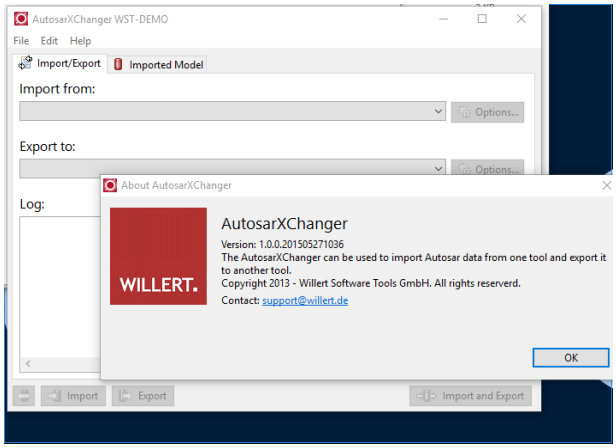
- MARQUARDT IS AN INDEPENDANT INTERNATIONAL AND SUCCESSFUL FAMILY OWNED COMPANY AND LEADING
- MANUFACTURER OF ELECTRO-MECHANIC AND ELECTRONIC SWITCHES AND -SYSTEMS. MARQUARDT PRODUCTS
- ARE WIDELY USED BY MULTIPLE CAR MANUFACTURERS. MARQUARDT ALSO MANUFACTURES DEVICES USED
- IN HOUSES OR FOR INDUSTRIAL APPLIANCES AND IS WORLDWIDE MARKET LEADER IN THESE AREAS.

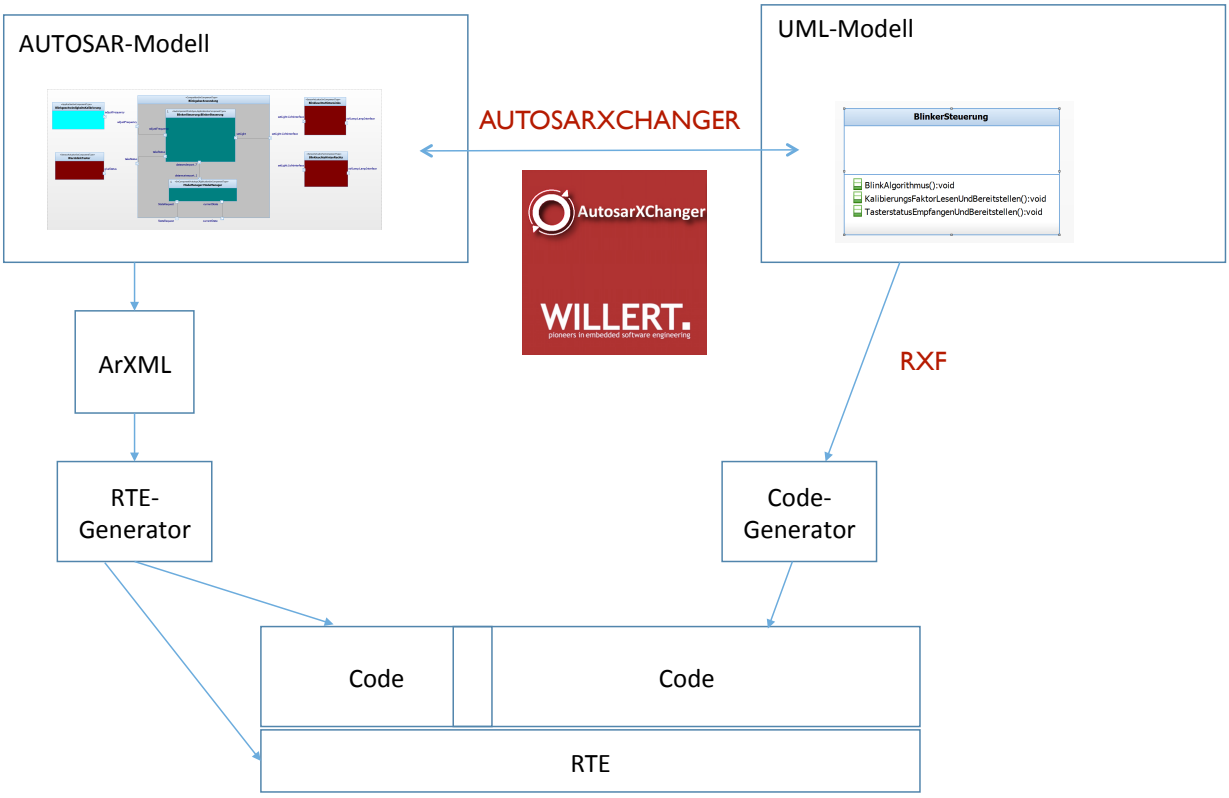
THE SYSTEM
KEYLESS GO AND ENTRY

- ☐ KEY
  - ☐ 16-BIT 64K ROM 4K RAM (-16K ROM FOR ENCRYPTION)
  - ☐ IAR COMPILER
  - ☐ REQUIREMENTS XCHANGER
  - ☐ RHAPSODY IN C
  - ☐ RXF FOR RENESAS RL78
  - ☐ NO AUTOSAR! OO-RTX INSTEAD
- ☐ ECU
  - ☐ 16-BIT 128K ROM 8K RAM (-16K ROM FOR ENCRYPTION)
  - ☐ GREENHILLS COMPILER
  - ☐ REQUIREMENTS XCHANGER
  - ☐ RHAPSODY IN C
  - ☐ AUTOSAR XCHANGER
  - ☐ RXF FOR RENESAS V850
  - ☐ AUTOSAR USING OO-RTX AS SINGLE TASK

WORKFLOW

- ☐ REQUIREMENTS IN DOORS
- ☐ USING REQUIREMENTS EXCHANGER FROM WILLERT TO SYNCHRONISE REQUIREMENTS BETWEEN DOORS AND RHAPSODY
  - ☐ TRACE LINKS ARE ADDED IN RHAPSODY
- ☐ FOR ECU USING THE WILLERT AUTOSAR EXCHANGER
  - ☐ TAKES ARXML FILE AND CONVERTS THE AUTOSAR ARTEFACTS TO RHAPSODY/UML
- ☐ DEVELOPMENT IS IN RHAPSODY USING UML AND CODE GENERATION TO THE WILLERT REF FRAMEWORK
- ☐ TESTING IS DONE USING TEST CONDUCTOR




AUTOSAR-Modell
UML-Modell
AUTOSARXCHANGER
BlinkerSteuerung
BlinkAlgorithmus():void
KalibierungsFaktorLesenUndBereitstellen():void
TasterstatusEmpfangenUndBereitstellen():void
AutosarXChanger
WILLERT.
pioneers in embedded software engineering
ArXML
RXF
RTE-
Generator
Code-
Generator
Code
Code
RTE

THE BIG WIN

"Although most people think of it as something new, keyless entry technology has actually been around for hundreds of years!"

- ☐ LAST MINUTE CHANGE
  - ☐ THEFT PREVENTION BY USING RADIO
  - ☐ IN THE LAST STAGE OF DEVELOPMENT
  - ☐ DOUBLE KLICK SWITCHES OF KEYLESS ENTRY
  - ☐ FOUR BLINKS AS FEEDBACK
  - ☐ IMPLEMENTATION TIME: 1 DAY
  - ☐ TESTING TIME: 1 NIGHT