Model-Driven Development & Embedded Systems

Sebastian Martin Aguilar
Modeling Tool Expert, IBM
sebastian.martin.aguilar@es.ibm.com

Salvador Trujillo, Ph.D.
IKERLAN, strujillo@ikerlan.es

David Gonzalez
IKERLAN, dgonzalez@ikerlan.es

Innovate2010
The Rational Software Conference

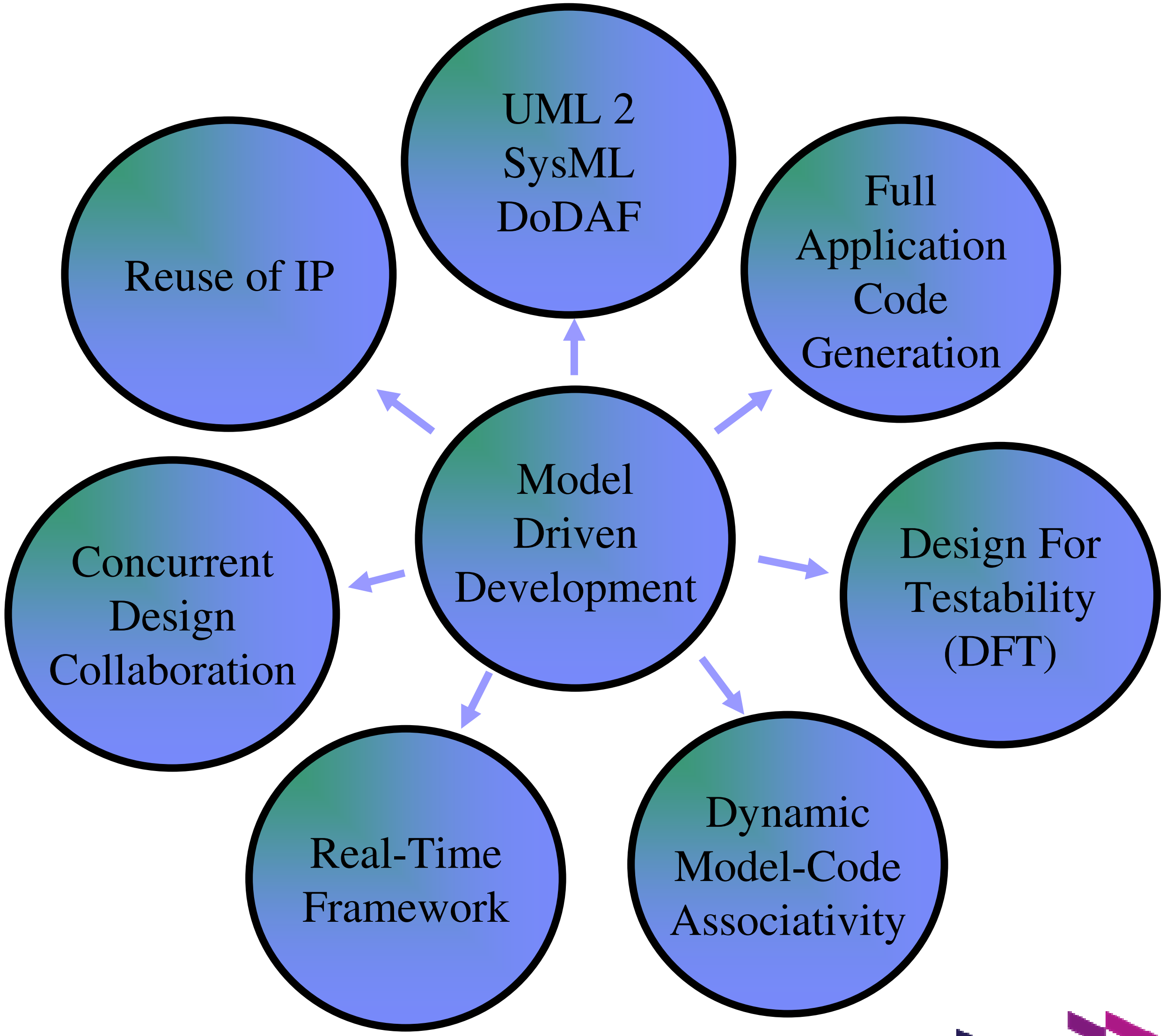
Let's build a smarter planet.

The premiere software and product delivery event.
4 de Noviembre, Madrid

ikerlan
ik4 research alliance

IBM

Model Driven Development



Model Driven Development

UML 2 SysML DoDAF

Full Application Code Generation

Design For Testability (DFT)

Dynamic Model-Code Associativity

Real-Time Framework

Concurrent Design Collaboration

Reuse of IP

- **Background**
  - R&D on embedded systems for industrial control
  - Application domains: energy, transportation, industrial control, etc
- **Paradigms & Methodologies**
  - SysML, Code Generation, DSL, Modeling
  - Experience: benefits, challenges
- **Dependability: safety-critical**
  - SysML & SIL4, Certification
  - Experience: benefits, challenges
- **Future: Team Orchestration**
  - Ongoing work: Towards **Jazz**

Embedded industry

- **Remains reliant on the craftsmanship of skilled individuals**
  - Labor intensive manual tasks (no automation, no product-line)
- **Increasing market pressure**
  - Reduce time-to-market, reduce associated cost for hardware, reduce maintenance cost, increasing complexity, multiple disciplines, increasing certification needs in safety domains, etc
- **Undergoing Paradigm shift**
  - MDD: Intensive use of models that can be transformed to code / test cases
    - Acceleration of timing, use of abstractions
  - Safety-critical systems with stringent requirements
    - Certification needs are increasingly relevant (e.g. SIL4)
  - SPL: a family instead of individual products, family-oriented reuse
    - Improved management, variability handling, business-oriented development

Green Energy

Engineering Challenges

A domain where reuse is a key success factor

Innovate2010

The Rational Software Conference

Let's build a smarter planet.

The premiere software and product delivery event.
June 6–10 Orlando, Florida

ikerlan
ik4 research alliance

IBM

Railway Systems

Engineering Challenges

A domain where safety certification is a must

Innovate2010

The Rational Software Conference

Let's build a smarter planet.

The premiere software and product delivery event.
June 6–10 Orlando, Florida

ikerlan
ik4 research alliance

IBM

General challenges

- Time-to-market
- Decrease the time to develop a control system
- **Customization** and roll-out
  - Decrease the time needed to customize an individual system
- Scalability and System **complexity**
  - The size and number of elements to be controlled by the control system is far from trivial (and still growing)
- Focus on value
  - Reduce the time devoted to maintenance activities
- **Multi-disciplinary teams**
  - Different engineers working together on highly complex problems
- Dependability & Certification
  - Develop high-integrity systems and attain certification

Paradigms

Model-Driven Development

Modeling	Design your system
Metamodeling	Design your modeling concepts
Model transformations	Specify your code generation (RulesComposer)

Diversity of Design Artifacts

- SysML, UML, DSL, Code generation, Code, Testing, etc

Variability into the System Design

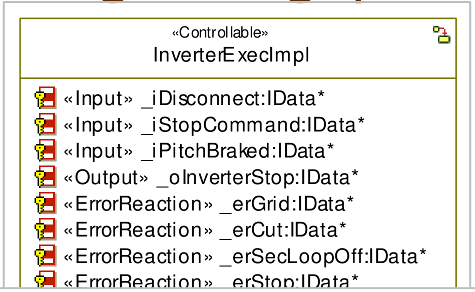
- SysML & Variability [Innovate 2010 talk]

DSL

- Visual DSL Editors

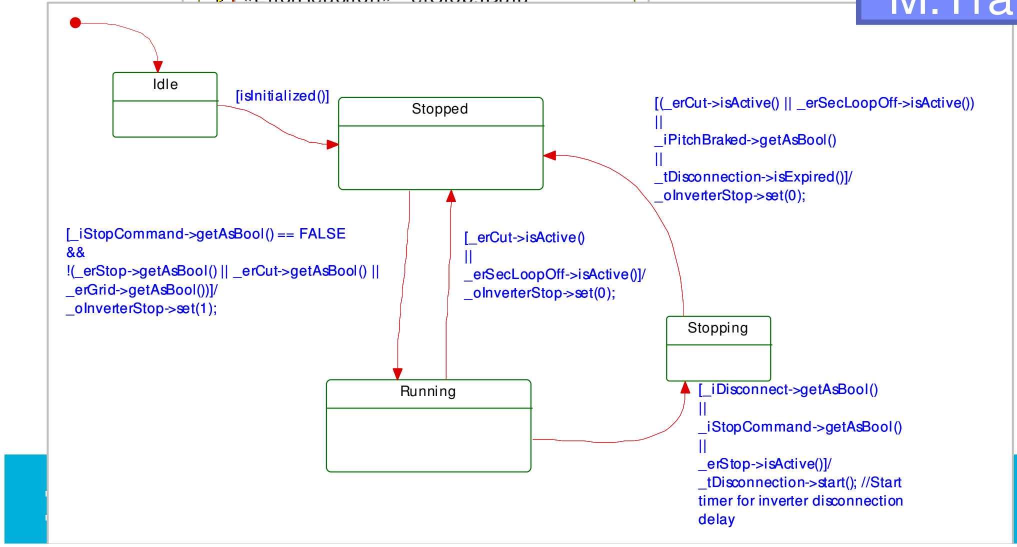
Model-Driven Development

- Development driven by intensive use of models.
- Automate the generation of code from models.
- Some models are UML, others are specific (a.k.a.DSL).
- Bridge the gap between modeling and implementation.



«Controllable»
InverterExecImpl

«Input» _iDisconnect:IData*
«Input» _iStopCommand:IData*
«Input» _iPitchBraked:IData*
«Output» _oInverterStop:IData*
«ErrorReaction» _erGrid:IData*
«ErrorReaction» _erCut:IData*
«ErrorReaction» _erSecLoopOff:IData*
«ErrorReaction» _erStop:IData*



[isInitialized()]

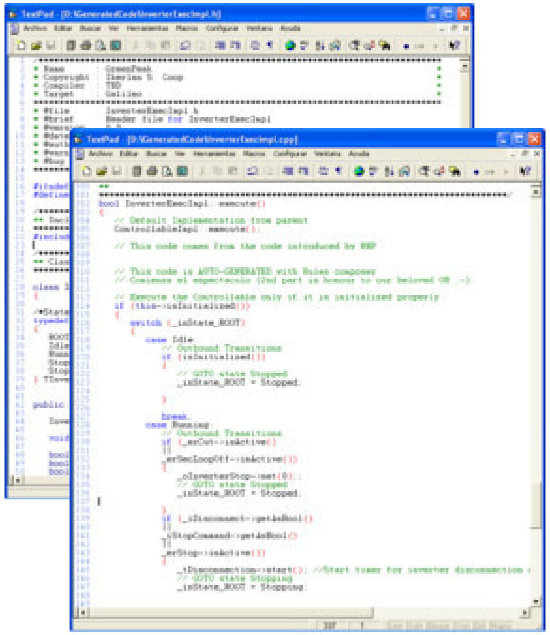
[(_erCut->isActive() || _erSecLoopOff->isActive())
||
_iPitchBraked->getAsBool()
||
_tDisconnection->isExpired()]/
_oInverterStop->set(0);

[_iStopCommand->getAsBool() == FALSE
&&
!(_erStop->getAsBool() || _erCut->getAsBool() ||
_erGrid->getAsBool())]/
_oInverterStop->set(1);

[_erCut->isActive()
||
_erSecLoopOff->isActive()]/
_oInverterStop->set(0);

[_iDisconnect->getAsBool()
||
_iStopCommand->getAsBool()
||
_erStop->isActive()]/
_tDisconnection->start(); //Start timer for inverter disconnection delay

M.Transf.



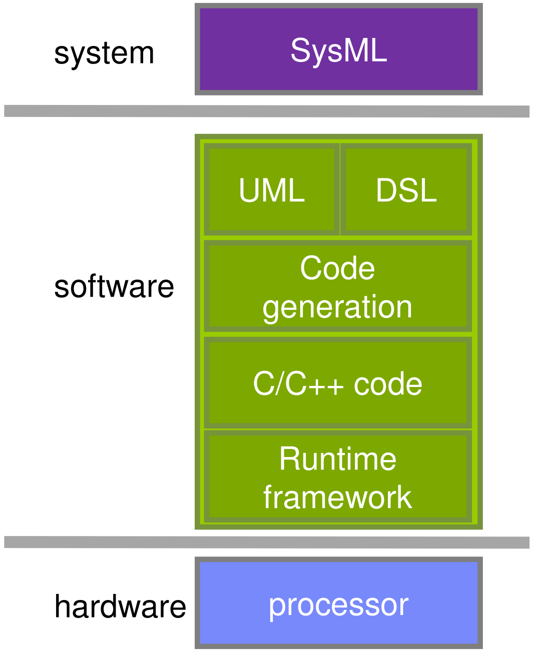
TextPad - [D:\GeneratedCode\InverterExecImpl.h]

TextPad - [D:\GeneratedCode\InverterExecImpl.cpp]

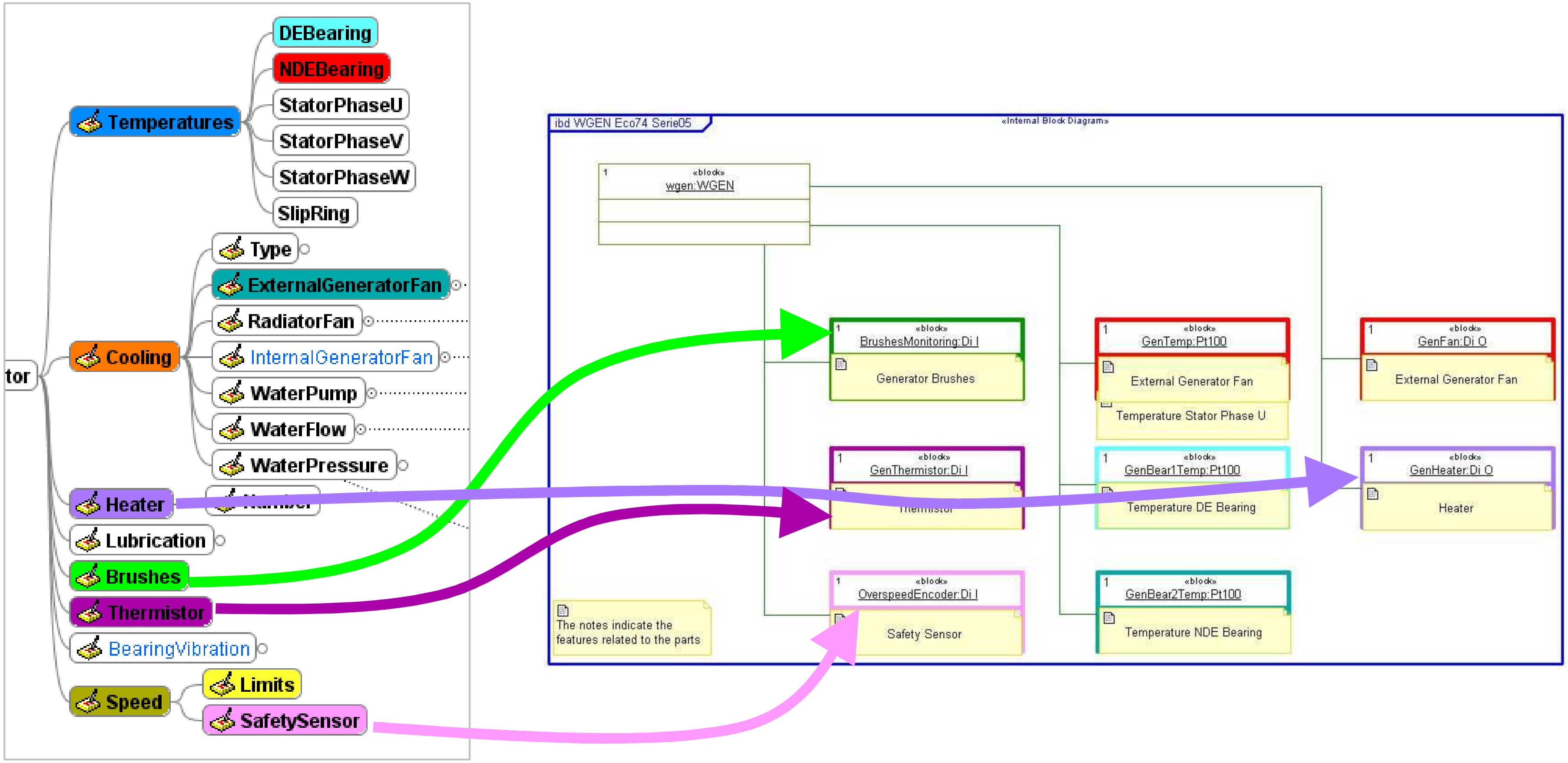
```
**
*******************************************************************/
bool InverterExecImpl::execute()
{
   // Default Implementation from parent
   ControllableImpl::execute();

   // This code comes from the code introduced by RHP

   // This code is AUTO-GENERATED with Rules composer
   // Comienza el espectáculo (2nd part is honour to our beloved GR ;-)

   // Execute the Controllable only if it is initialized properly
   if (this->isInitialized())
   {
      switch (_isState_ROOT)
      {
         case Idle:
            // Outbound Transitions
            if (isInitialized())
            {
               // GOTO state Stopped
               _isState_ROOT = Stopped;

            }

            break;
         case Running:
            // Outbound Transitions
            if (_erCut->isActive()
            ||
            _erSecLoopOff->isActive())
            {
               _oInverterStop->set(0);
               // GOTO state Stopped
               _isState_ROOT = Stopped;

            }
            if (_iDisconnect->getAsBool()
            ||
            _iStopCommand->getAsBool()
            ||
            _erStop->isActive())
            {
               _tDisconnection->start(); //Start timer for inverter disconnection
               // GOTO state Stopping
               _isState_ROOT = Stopping;
```

Diversity of Design Artifacts

- System
  - SysML & Variability [Innovate 2010's talk]
- UML
  - Design your software using *Universal* modeling notation
- DSL
  - Design your software with "your language"
- Code Generation
  - Write the code that specifies how your code is to be generated
- Code, etc
  - Remember this is *really* running in a dedicated hardware!



Variability into the System models

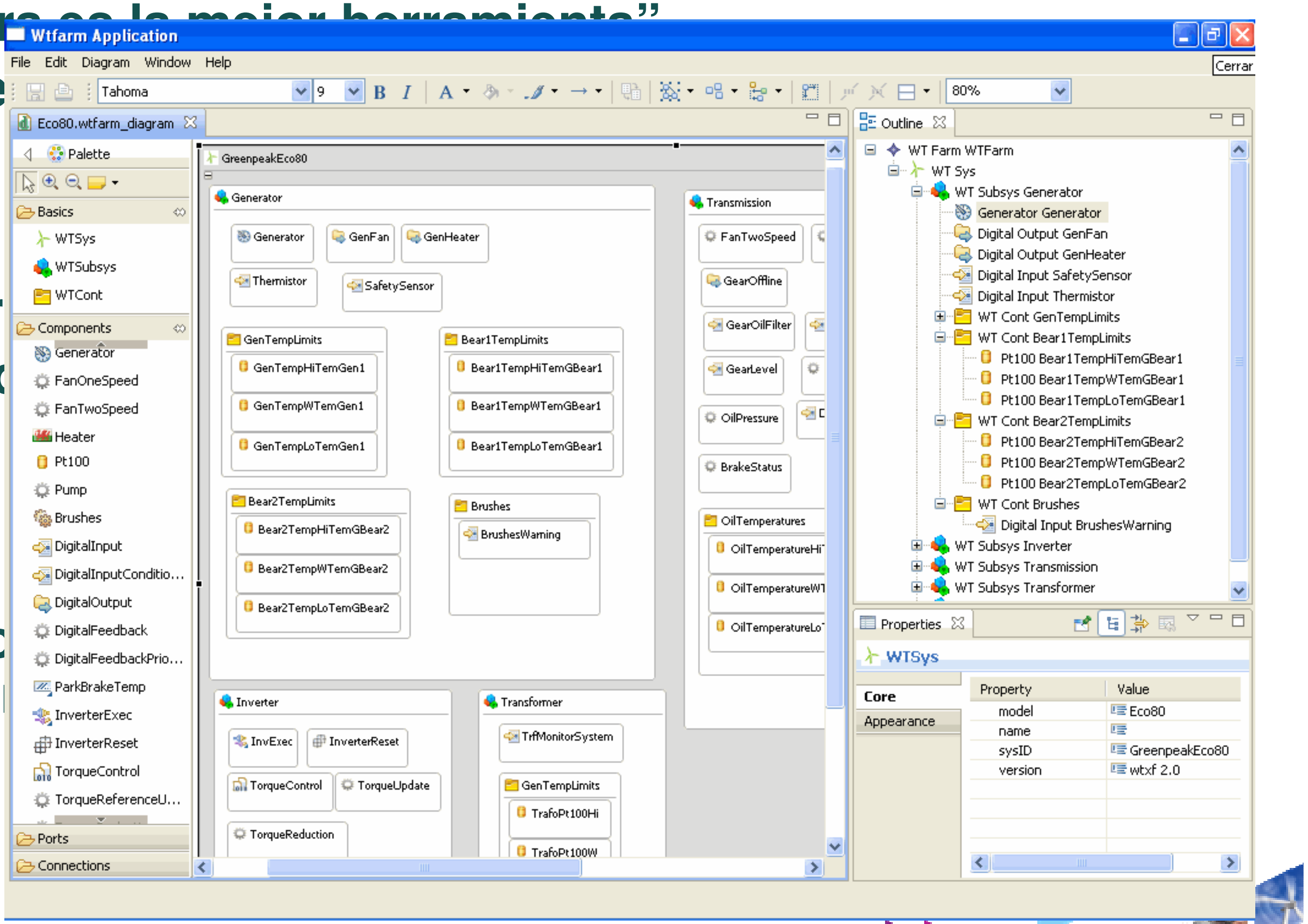


Feature Model

SysML's Block Definition Diagram

Domain Specific Languages & DSL Editors

- **Antes:** **Sin abstracciones**
  - Diseño: “la pizarr
  - Desarrollo: imple
- **Ahora:**
  - Diseño: en la her
  - Desarrollo: el cóc charla de hoy)
- **Bottom-line:**
  - Es más abstracto lenguaje de prog



Property	Value
model	Eco80
name	
sysID	GreenpeakEco80
version	wtxf 2.0

Dependability & Safety-critical Systems

- **Dependability & SysML**
  - Standards are important
- **IK's methodology under certification**
  - Metodologia base y customizable para el nivel SILx
  - dentro de RHP: reference profile, etc
  - What is new when using SILx
  - Why are we using SysML? Why code generation?
  - Explicar proceso de forma general
- **TERESA project: dependable design patterns**
  - Metodologia base y customizable para el nivel SILx
  - Security design patterns (within the project)

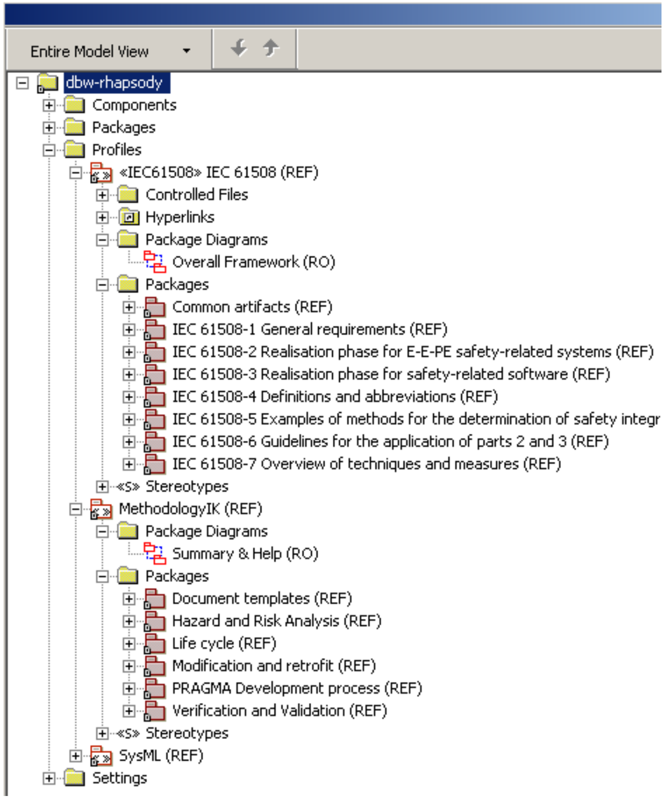
Dependability & SysML

- **There is an increasing need for dependable Embedded systems in many application domains (railway, automotive, elevation, etc.)**
  - Dependability reference standard is IEC61508
- **Why SysML?**
  - SysML improves documentation and model traceability
- **According to the IEC61508, SysML can be used in:**

Part 1	Documentation
Parts 2/3	Traceability
Part 3 - B.5	Modelling
Part 3 - A.2	Structured diagrammatic methods
Part 3 - A.2	Automatic software generation
Part 2 - B.1, B.2; Part 3 - A.1, A.2, A.4	Computer aided specification and design tools
Part 2 - B.6	Simulation

Methodology & SIL Certification

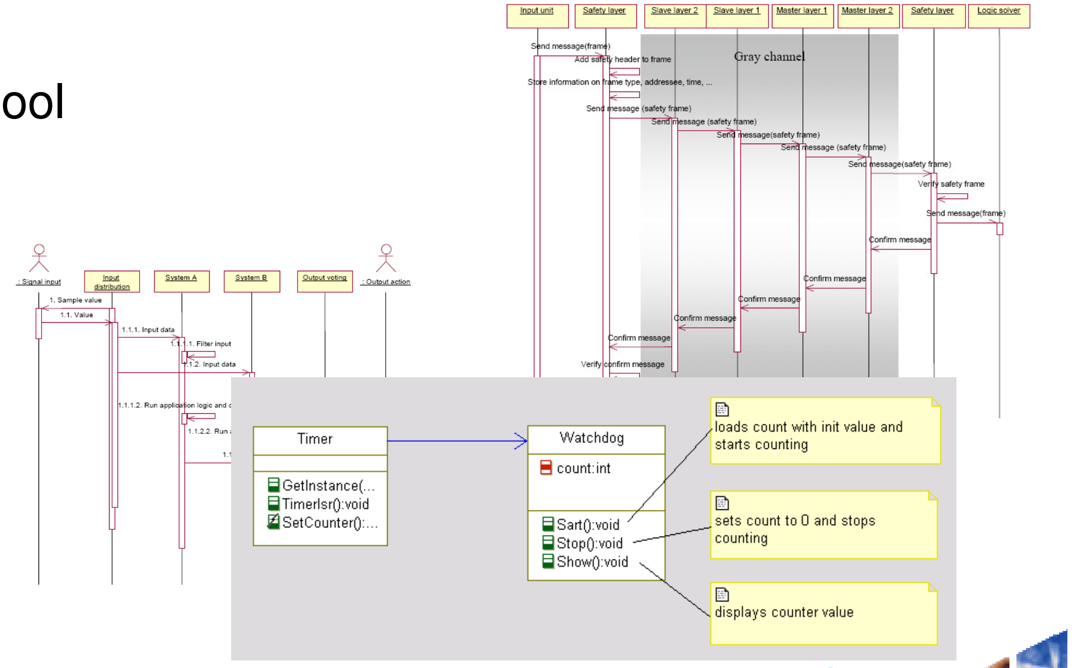
- **Methodology is a key point in order to achieve certification**
- **IK methodology guides the developer to the SIL expected level**
  - Based on PRAGMA and IEC61508
  - Initially defined in UML
  - Recently extended to SysML
  - **Integration within the tool**
  - IK methodology can be imported into RHP as a reference profile
- **IK methodology under certification**
  - Work in progress…



```
Entire Model View
dbw-rhapsody
  Components
  Packages
  Profiles
    «IEC61508» IEC 61508 (REF)
      Controlled Files
      Hyperlinks
      Package Diagrams
        Overall Framework (RO)
      Packages
        Common artifacts (REF)
        IEC 61508-1 General requirements (REF)
        IEC 61508-2 Realisation phase for E-E-PE safety-related systems (REF)
        IEC 61508-3 Realisation phase for safety-related software (REF)
        IEC 61508-4 Definitions and abbreviations (REF)
        IEC 61508-5 Examples of methods for the determination of safety integr
        IEC 61508-6 Guidelines for the application of parts 2 and 3 (REF)
        IEC 61508-7 Overview of techniques and measures (REF)
      «S» Stereotypes
    MethodologyIK (REF)
      Package Diagrams
        Summary & Help (RO)
      Packages
        Document templates (REF)
        Hazard and Risk Analysis (REF)
        Life cycle (REF)
        Modification and retrofit (REF)
        PRAGMA Development process (REF)
        Verification and Validation (REF)
      «S» Stereotypes
    SysML (REF)
  Settings
```

Design patterns

- **We are taking part in FP7 TERESA (Trusted computing Engineering for Resource constrained Embedded Systems Applications)**
  - The goal is to build a repository to store frequently used design patterns for secure and/or dependable (S&D) applications
- **Main ideas:**
  - Use of MDD (UML/SysML)
  - RHP Repository integrated in the tool
  - Definition of the engineering process in order to effectively use the created patterns



Ongoing Safety Projects

- **Railways signaling system**
  - Development a high-integrity system with ongoing **SIL 4** level certification
  - GOALS: manage complexity, reuse components, reduce testing time
- **Wind turbine systems**
  - New safety product-line platform for offshore wind farms
- **Elevators**
  - Catalog of components based on MDE
  - Ongoing **SIL3** certification
- **More**
  - Integral Methodology with HW, SW, FPGA, Comms, etc
  - Incorporating Model Based Testing

DEMO

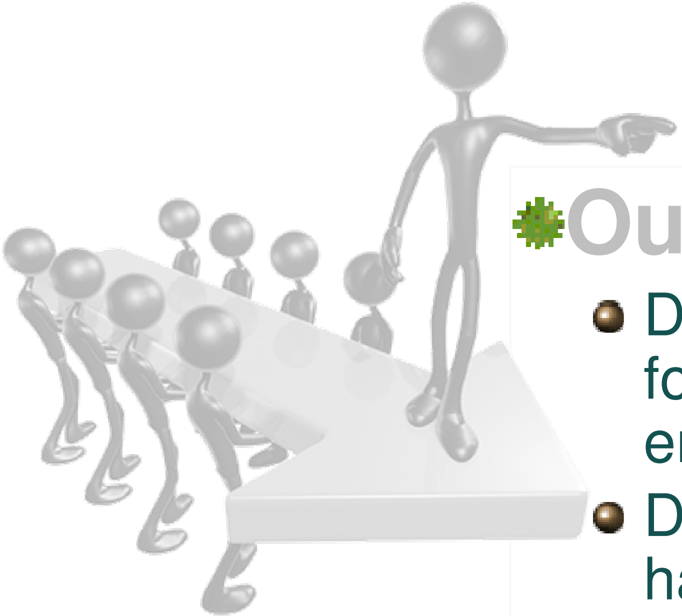
Demo time

Conclusions

Lessons Learned

- System-driven software design
  - The ability to delay or later modify design decisions is very relevant
  - software have to fit the system elements it is controlling
- Market orientation is critical
  - The participation of domain, market, and technical experts is a critical success factor
- System-software boundaries
  - Recognizing the significance of separating systems from runtime software seems trivial
- Design as a continuum
  - each discipline focuses on certain design parts develop in parallel disciplinary teams
  - Introducing variability further accelerates this
- Living with inconsistencies
  - … intermediate stages during the development in which certain inconsistencies …
- The Art of Bootstrapping
  - Start small, and iterate

Conclusions



Our talk

- Described the line of work several teams are following in IKERLAN for the development of embedded systems
- Different application domains and customers, have different demands and so different techniques

Future work

- Widespread internal adoption (not yet)
- Automation into the development process
- Incorporation of novel techniques

Questions

Learn more at:

- IBM Rational software
- Rational launch announcements
- Rational Software Delivery Platform
- Accelerate change & delivery
- Deliver enduring quality
- Enable enterprise modernization
- Ensure Web security & compliance
- Improve project success
- Manage architecture
- Manage evolving requirements
- Small & midsized business
- Targeted solutions
- Rational trial downloads
- developerWorks Rational
- Leading Innovation
- IBM Rational TV
- IBM Business Partners
- IBM Rational Case Studies

Model-Driven Development & Embedded Systems

Sebastian Martin Aguilar
Modeling Tool Expert, IBM
sebastian.martin.aguilar@es.ibm.com

Salvador Trujillo, Ph.D.
IKERLAN, strujillo@ikerlan.es

David Gonzalez
IKERLAN, dgonzalez@ikerlan.es

Innovate2010
The Rational Software Conference

Let's build a smarter planet.

The premiere software and product delivery event.
4 de Noviembre, Madrid