



<http://liuppa.univ-pau.fr>

From SysML to UML


Does SysML improve Software Components Design in a System Engineering Project?



UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR



Nicolas BELLOIR
Associate Professor at the University of Pau, France



About the Speaker

- **Nicolas Belloir**
 - Is Associate Professor at the UPPA
 - **Not a native speaker!!**
 - **Has worked on RT project in Industry**
 - Latecoere (Human centrifuge)
 - Airbus (Integration and Development Flight Simulator)



UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 2



About the Document

- **Targeted audience**
 - Master Degree or Ph. D. students
 - Basic knowledge in UML
- **Format**
 - Short speaking (3H)
- **Special Thanks to**
 - Jean-Michel Bruel, Professor ,Toulouse University
 - Laurent Nonne, Teacher, Blagnac IUT
 - Pascal Roques, Senior consultant, ARTAL Inc.





UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 3



About Pau








PAYS DE L'ADOUR
SLIDE 4



The University of Pau (UPPA)

A network of four campuses





The LIUPPA Laboratory

Research Organization



Computer Science Laboratory

Director : Pr Conduc Pham

2 Research Teams

Software Engineering and Distributed Systems

Information Processing and Interactions

5 Project-Teams

SLIDE 6




LIUPPA

Software Engineering and Distributed Systems

- Model-Driven Engineering (MDE)
- Agents, Components & Software Services
- Security, Time-Stamping, Digital Right Management
- Communication Protocols, High-Speed Networks, Congestion Control
- Wireless Sensor Networks (WSN)
- Multimedia and Quality of Service



UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 7




LIUPPA


Software Engineering and Distributed Systems

Application Domains

Time-Stamping Infrastructure	UML 2 - SysML
Congestion Control For GBIT/S Networks, Optimized TCP	Multimedia Supervision Platform
Java Management Extension (JMX)	Service-Oriented Computing and Autonomic Computing
Learning Grid Infrastructure	Management of Wireless Mobile Devices
Video Transport On WSN	Deny of Service, Intrusion Detection System




UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 8



Need of modeling

- “Essentially, all models are wrong, but some are useful”
 - Professor George E. P. Box (1987)
- But... Modeling is still a key feature
 - To increase systems
 - Reliability
 - Adaptability
 - To make generative programming realistic
 - To increase reusability


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 9



Use of DSM to Modeling Systems

- Domain Specific Modeling
 - Tends to support higher-level abstractions than General-Purpose modeling languages
 - Require less effort to specify a system
- UML to define DSM
 - UML includes the profile mechanism to be constrained and customized for specific domains and platforms
 - Stereotypes
 - Tagged values
 - Constraints
 - SoaML for web service engineering
 - SysML for System modeling WSML


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 10



Content

- **Part I**
 - Introduction to System Engineering
 - Introduction to SysML
 - From UML to SysML
- **Part II**
 - Case studies
 - From SysML to UML
 - Use of Software Components in System Engineering
 - Questions and discussions

UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 11



System Engineering

- **Not Software Engineering...**
- **...Before Software Engineering!**
 - Historically
 - In the development process

UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
2008/2009




A Complex System

A complex system is very different of a simple software system

- **Integrated to provide services to it environment corresponding to the system finality**
- **Interacting between themselves and the environment**



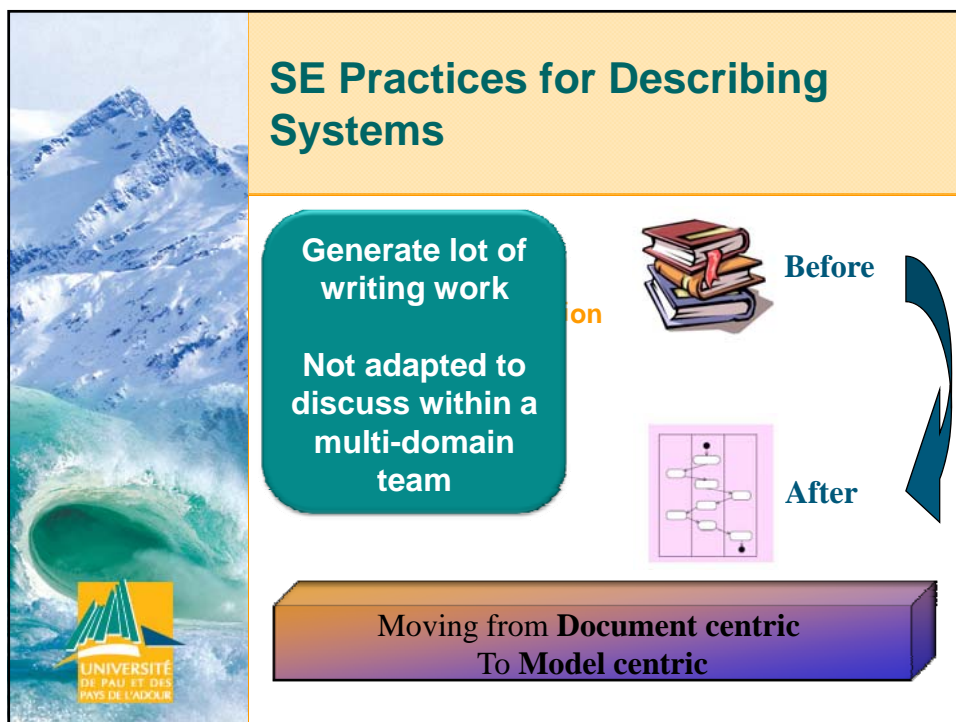
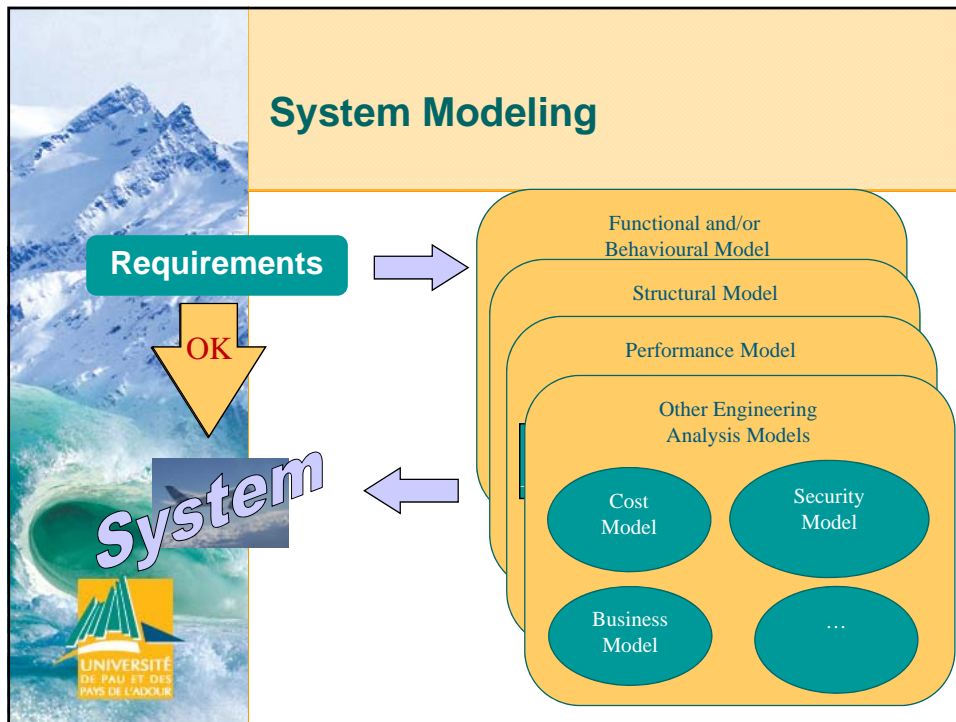





Systems of Systems

- **A system**
 - **Should manage interactions between parts**
 - **Support expected behavior**
 - **Handle unexpected ones**









UML for modeling systems?

- **UML 1.x, not really:**
 - Mainly for Information Systems
 - Poor link between diagrams
 - Poor link between models and requirements
- **UML 2.x, neither, although:**
 - Implication of Systems Eng. in the definition
 - Composite structure diagram




UML for modeling systems (cont.)?

- **UML a good basis**
 - *De facto* standard in Software Engineering
 - Provides a lot of useful concepts to describe a complex system
 - Stable and extensible - can be adapted for the SE needs
 - Lot of available tools
- **But...**
 - Lack of several key concepts of SE
 - UML vocabulary is too « software-like » to be used by System Engineers
 - Too many diagrams (13 different kinds)




System Engineering

- Any question?




UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR




Content

- Part I
 - Introduction to System Engineering
 - Introduction to SysML
 - From UML to SysML
- Part II
 - Case studies
 - From SysML to UML
 - Use of Software Components in System Engineering
 - Questions and discussions




UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 20



UML capabilities for SE modeling

- **UML 1.x fails to support SE**
- **UML 2 increases UML 1.x capabilities for SE modeling**
 - **Collaboration between SE and the OMG**
 - **Allowing hierarchical modeling of both structural and behavioral features**
 - *Composite structure diagram*
- **But No support for :**
 - **Linking models to requirements**
 - Need other software like Doors
 - **Modeling continuous flows**
 - ...


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 21




UML and SE

- **UML a good basis**
 - *De facto* standard in Software Engineering
 - Provides a lot of useful concepts to describe a complex system
 - Stable and extensible - can be adapted for the SE needs
 - Lot of available tools
- **But...**
 - Lack of several key concepts of SE
 - UML vocabulary is too « software-like » to be used by System Engineers
 - Too many diagrams (13 different kinds)

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 22

	<h2>SysML: identity card</h2>
	<ul style="list-style-type: none">• Date of birth: 2001!• Current version: 1.1 (03/12/2008)• Father: OMG/UML• Leading authors<ul style="list-style-type: none">- Conrad Bock- Cris Kobryn- Sanford Friedenthal


	<h2>SysML Objectives</h2>
	<ul style="list-style-type: none">• Based on UML 2 : but must be able to design a large panel of systems (hw, sw, data, ...)• Standard : provide a modeling language for SE in order to:<ul style="list-style-type: none">- Analyze, specify, design and verify complex systems- Allowing information exchange throughout tools- Bringing the semantic gap between all SE disciplinary• Simple :<ul style="list-style-type: none">- easy to learn for system engineers- Supported by designing tools• Extensible : for specific domains (automotive, aerospace ...)



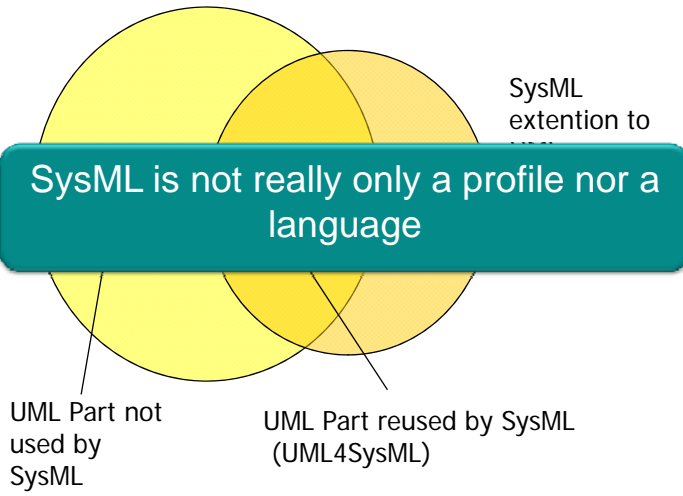
SysML: Who is behind?

- **Industry**
 - American Systems, BAE Systems, Boeing, Deere & Company, EADS Astrium, Eurostep, Israel Aircraft Industries, Lockheed Martin, Motorola, NIST, Northrop Grumman, oose.de, Raytheon, Thales
- **Tools Companies**
 - Artisan, EmbeddedPlus, Gentleware, IBM, I-Logix, Mentor Graphics, PivotPoint Technology, Sparx Systems, Telelogic, vitech
- **Academia, organizations**
 - AP-233, INCOSE, Georgia Institute of Technology, AFIS

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 25



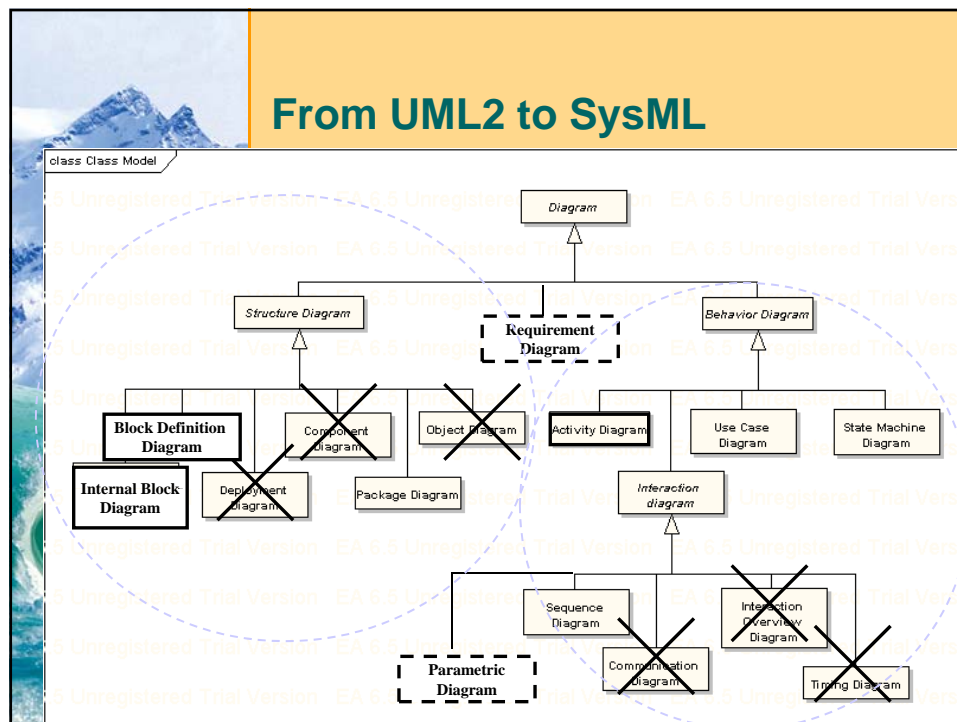
SysML, a UML2 profile?




SysML
extention to
UML

SysML is not really only a profile nor a language

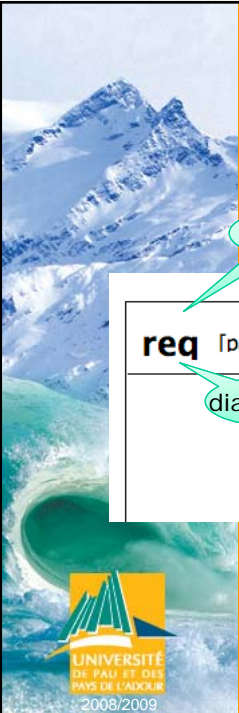
UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 26



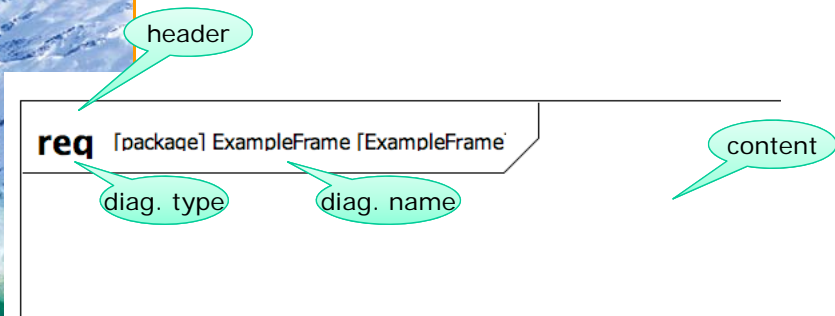


SysML diagram frames

- Each SysML diag. represents a model element
- Each SysML diag. must have a Diagram Frame
- Diagram context is indicated in the header:
 - Diagram kind (req, act, bdd, ibd, sd, etc.)
 - Model element type (package, block, activity, etc.)
 - Model element name
 - User defined diagram name or view name
- A separate diagram description block is used to indicate if the diagram is complete, or has elements elided



SysML diagram frames (e.g.)



header

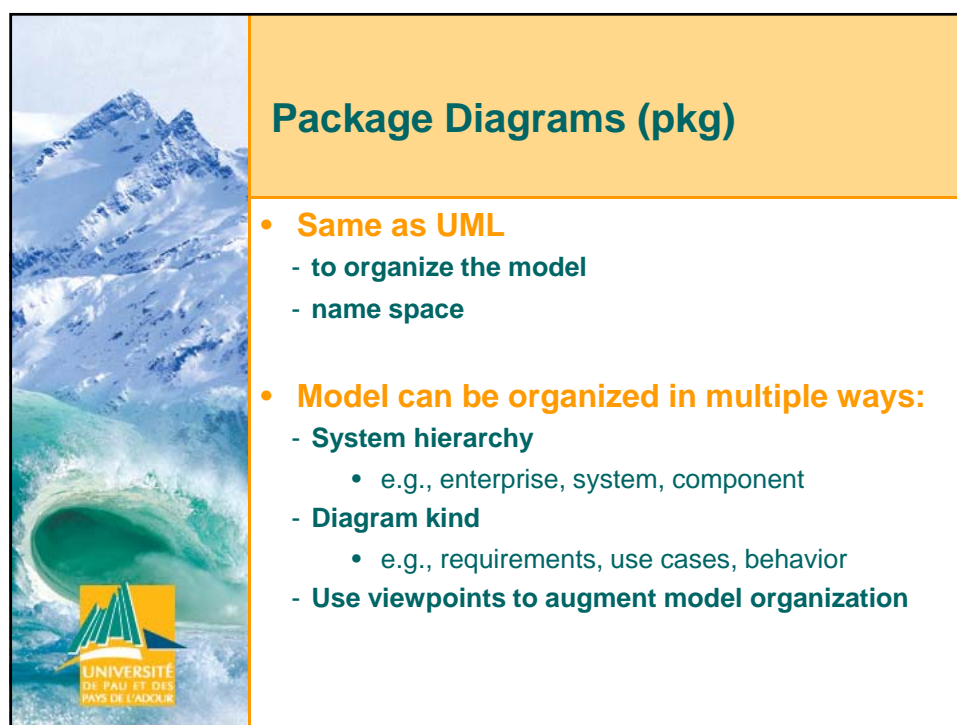
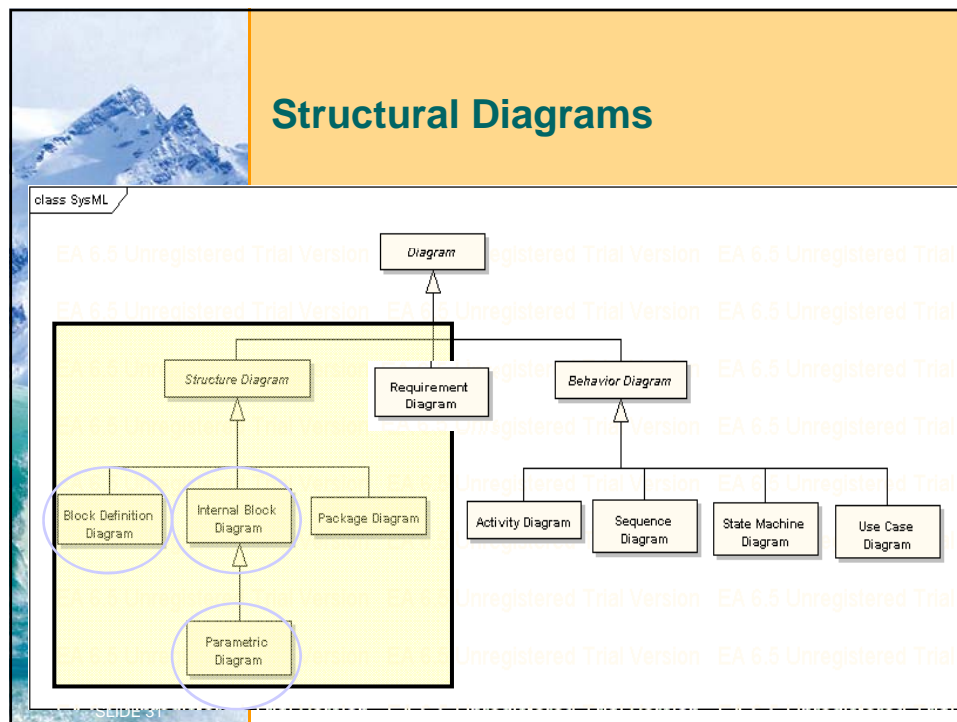
req [package] ExampleFrame [ExampleFrame]

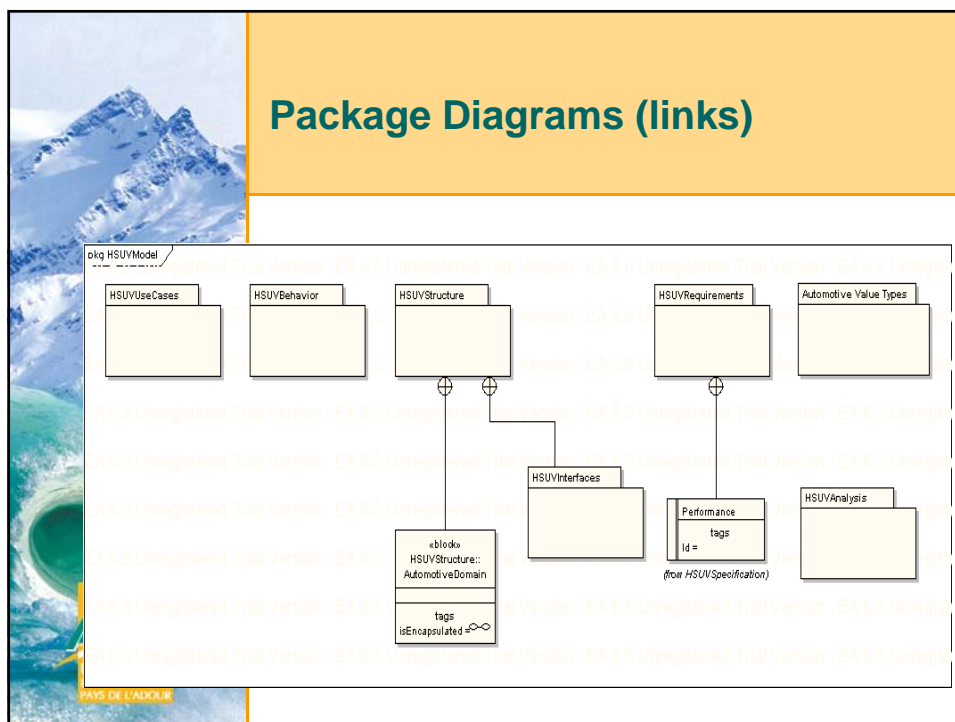
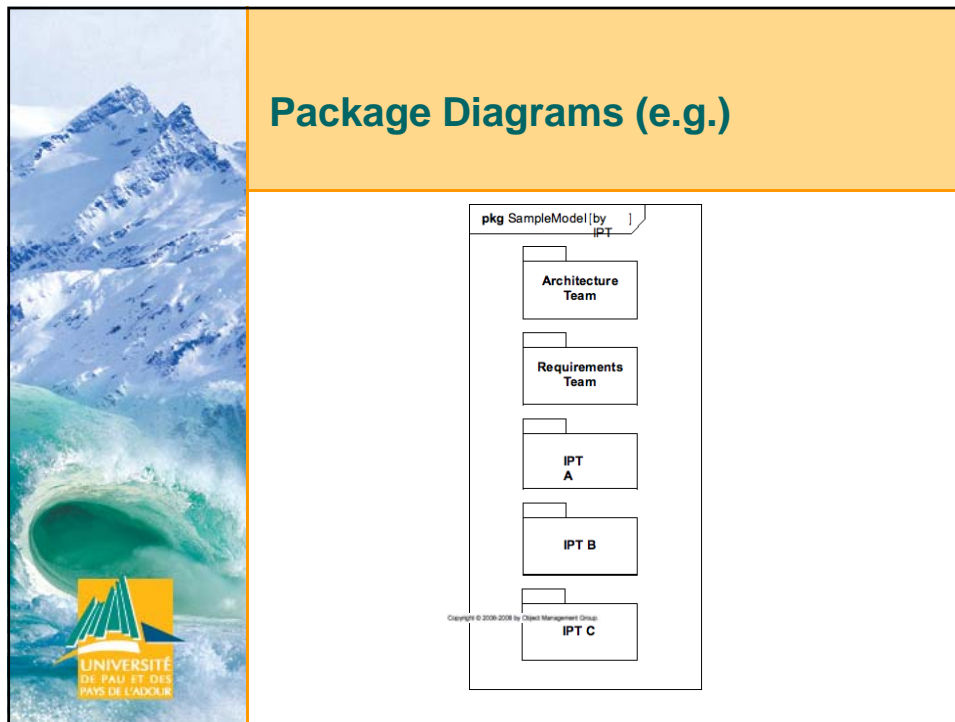
diag. type

diag. name

content

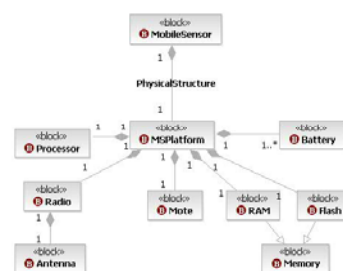
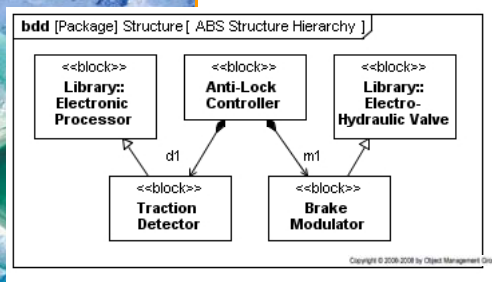
bruel@irit.fr





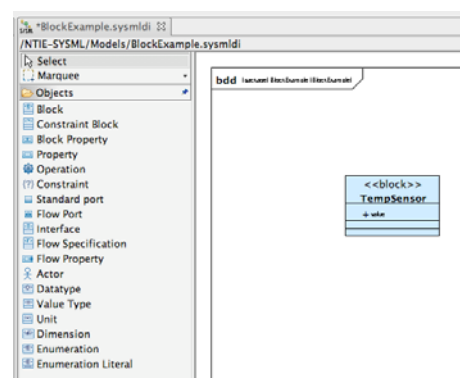
Block Definition Diagrams (bdd)

- Classes are dead... welcome to blocks!
 - Can be anything (System, Hardware, Software, Data, Procedure, Facility, Person)
 - Satisfy Systems Engineers



Block Definition Diagrams (bdd)

- Compartments
 - Properties
 - Operations
 - Constraints
 - Allocations
 - Requirements
- User defined!



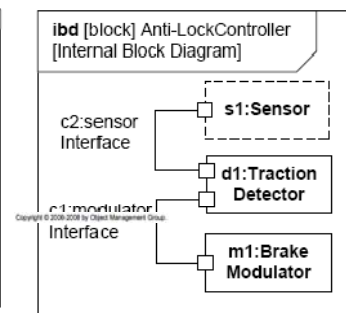
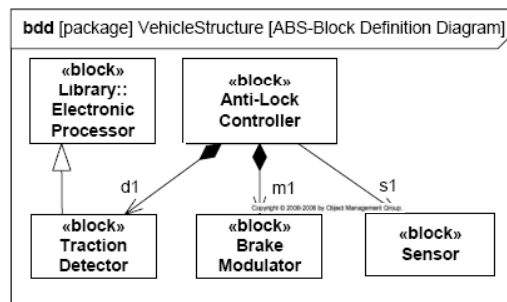
Block Definition vs. Usage

■ Block Definition Diagram (BDD)

- Describes relations between blocks (composition, inheritance...)

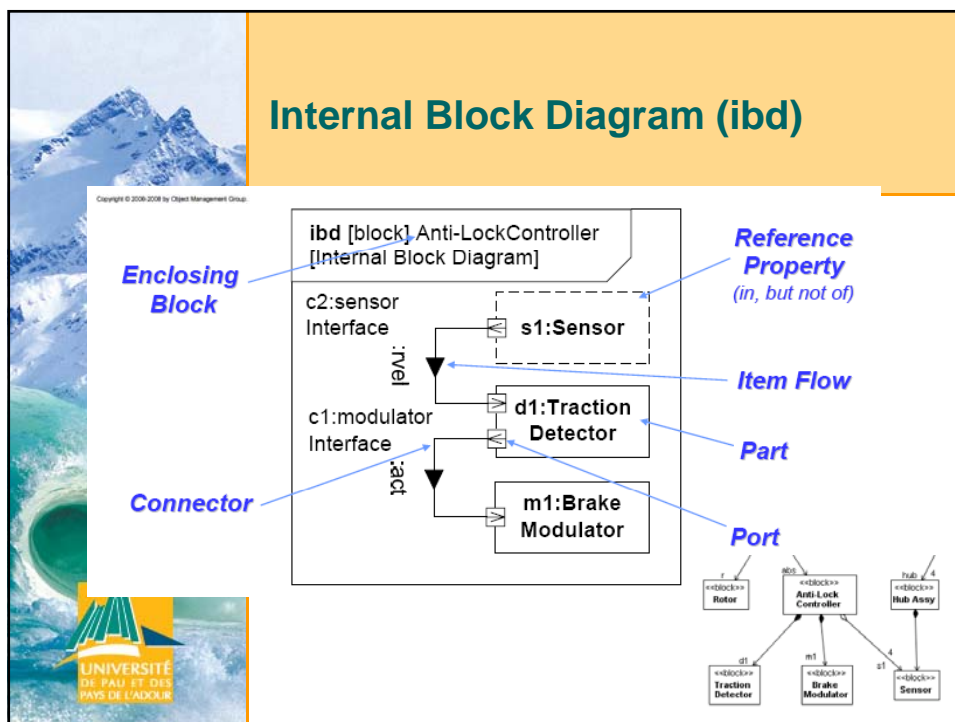
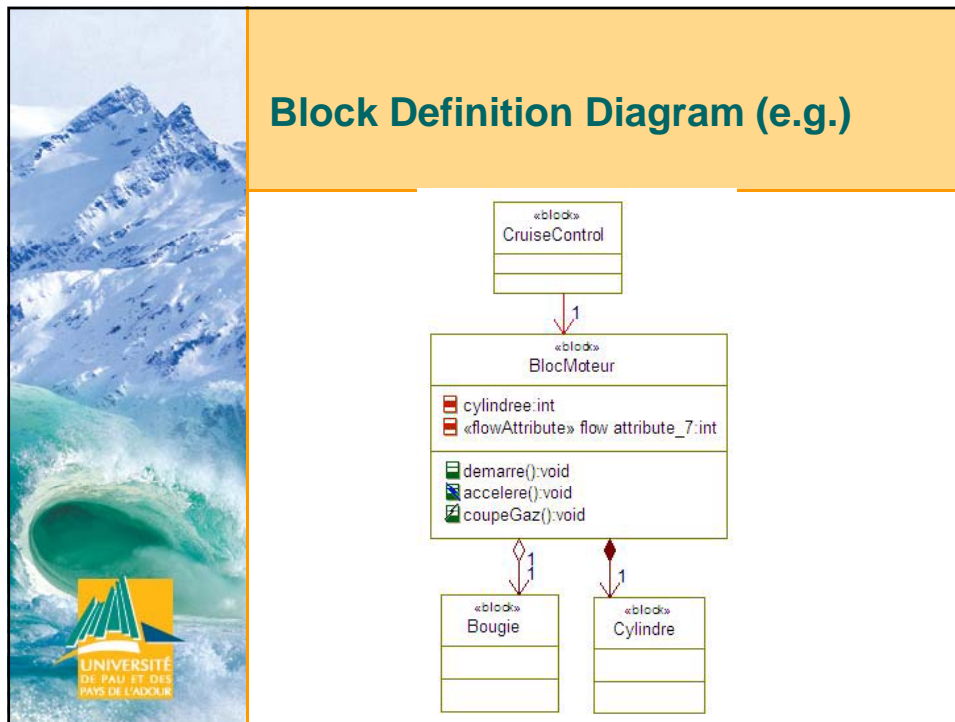
■ Internal Block Diagram (IBD)

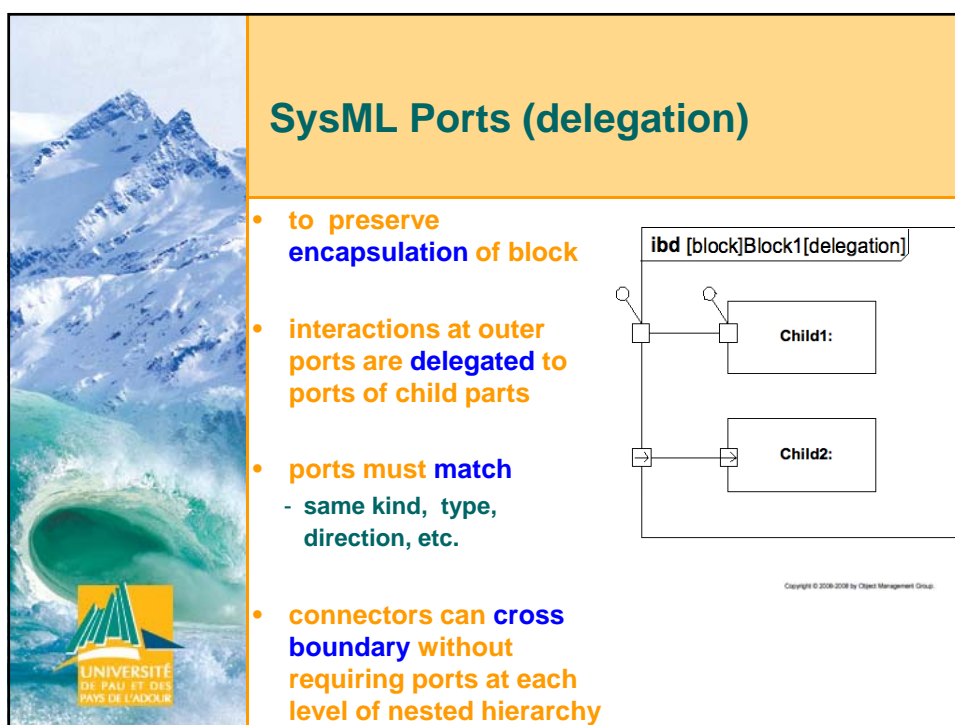
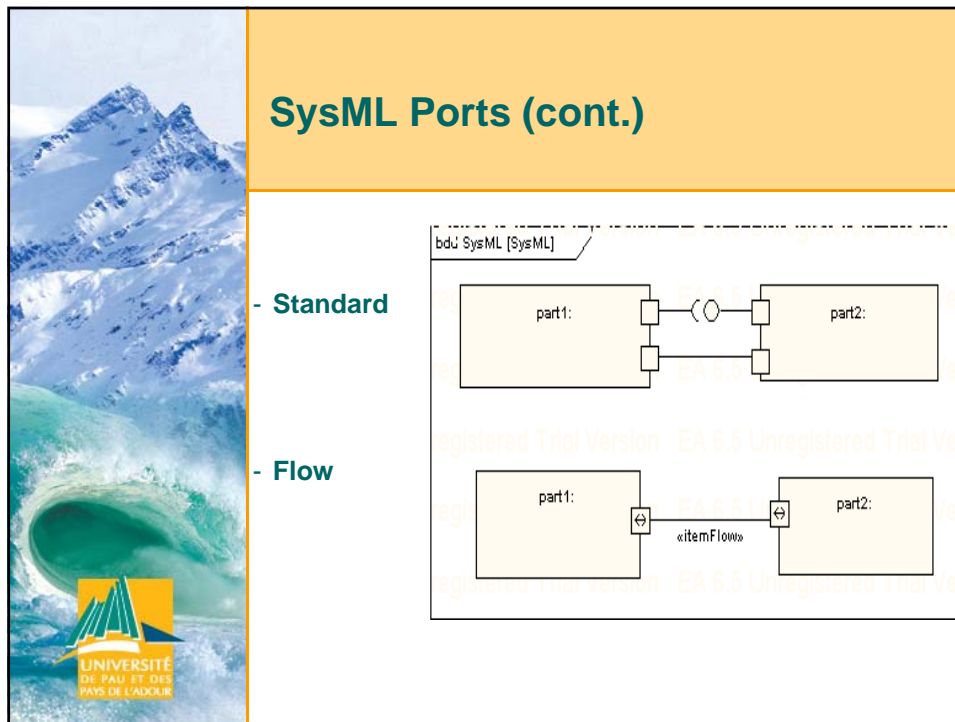
- Describes block internal structure with *parts*, *ports* et *connectors*.

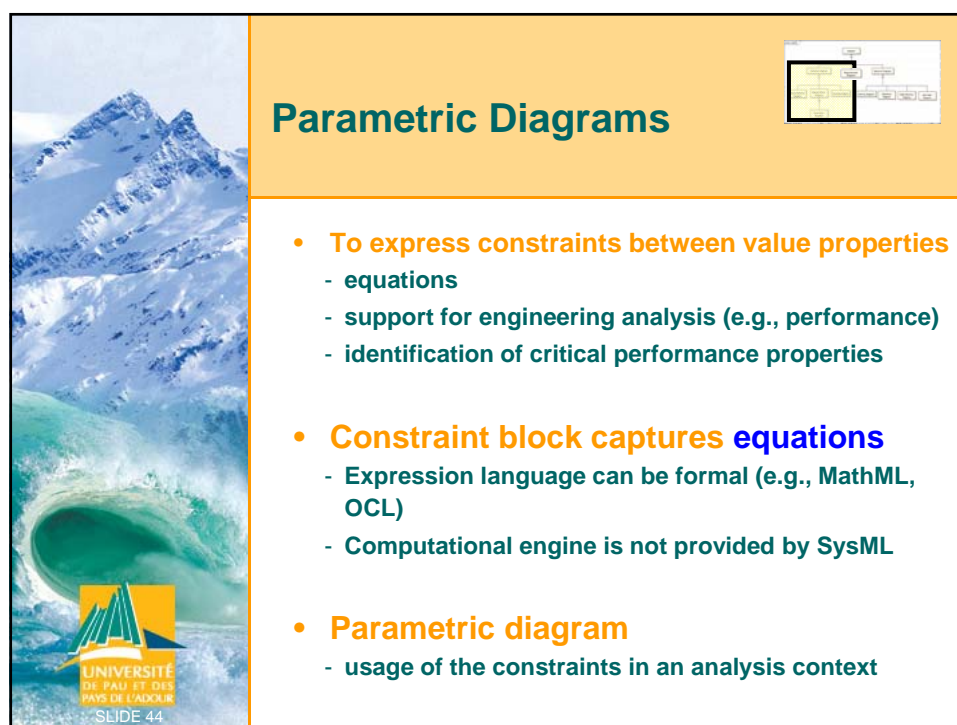
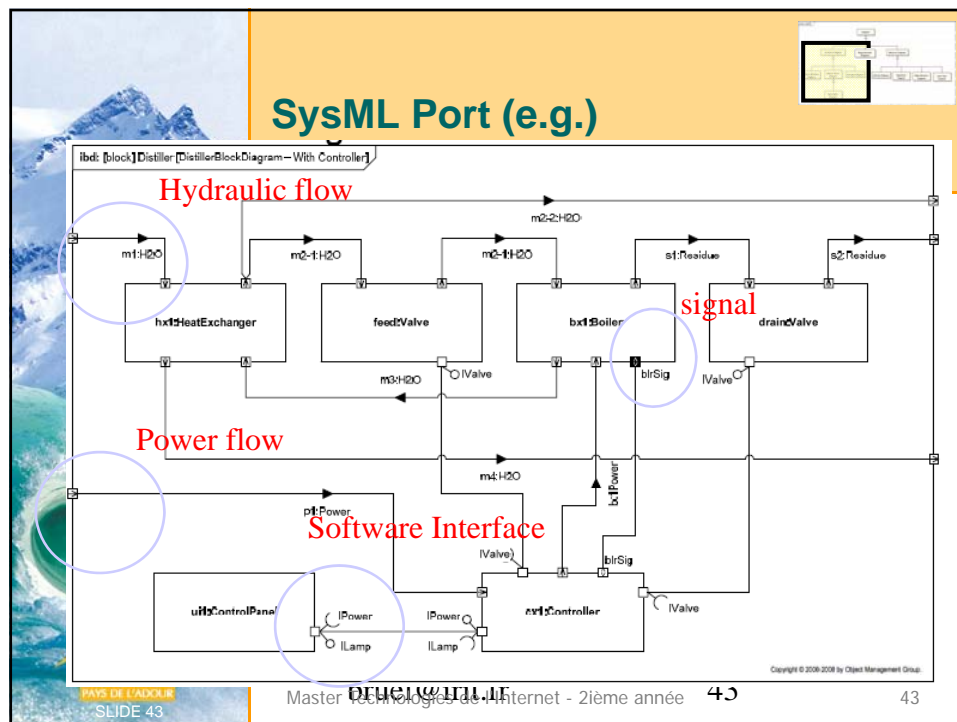


Block example

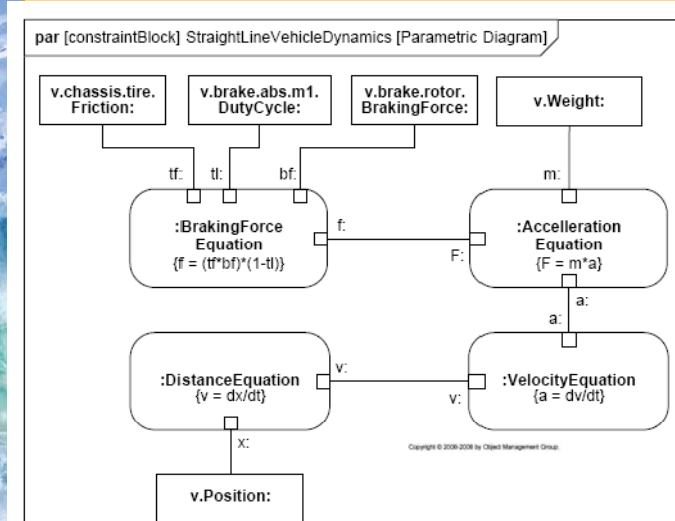




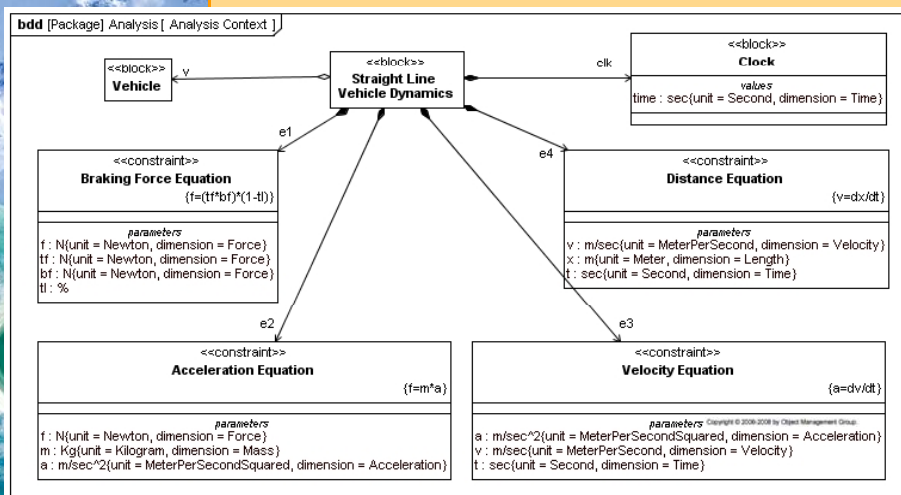


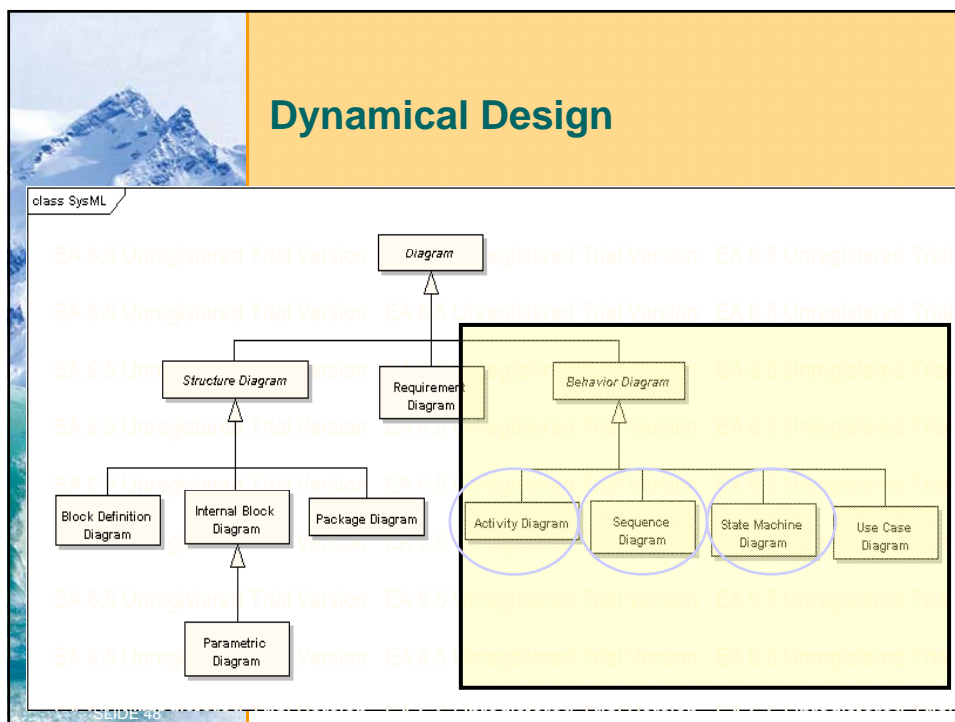
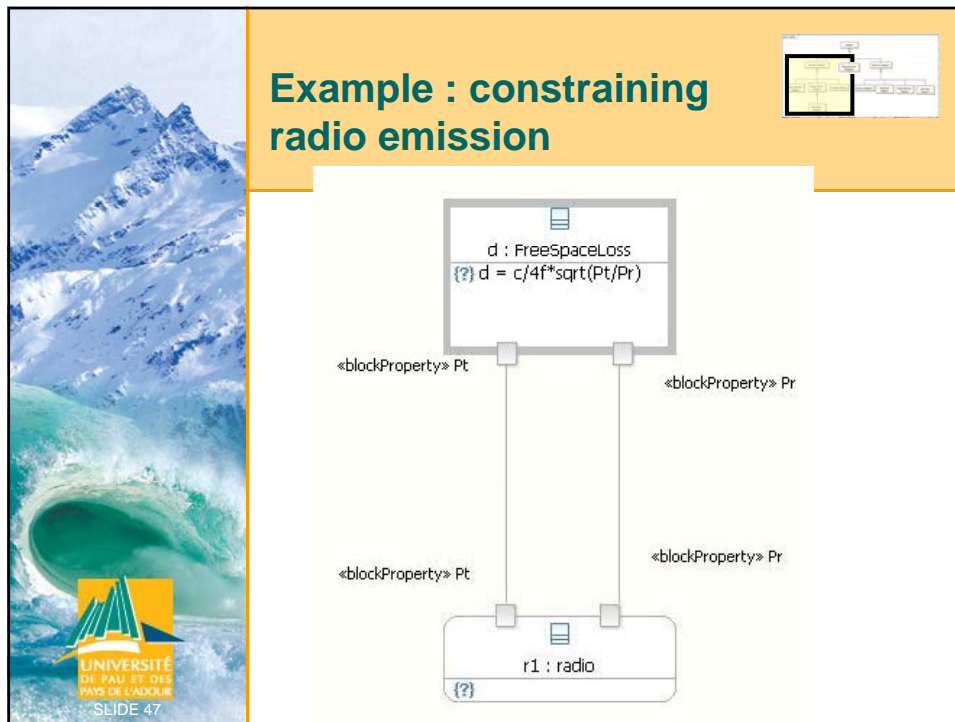



Parametrics (e.g. 1)




Parametrics (e.g. 2)

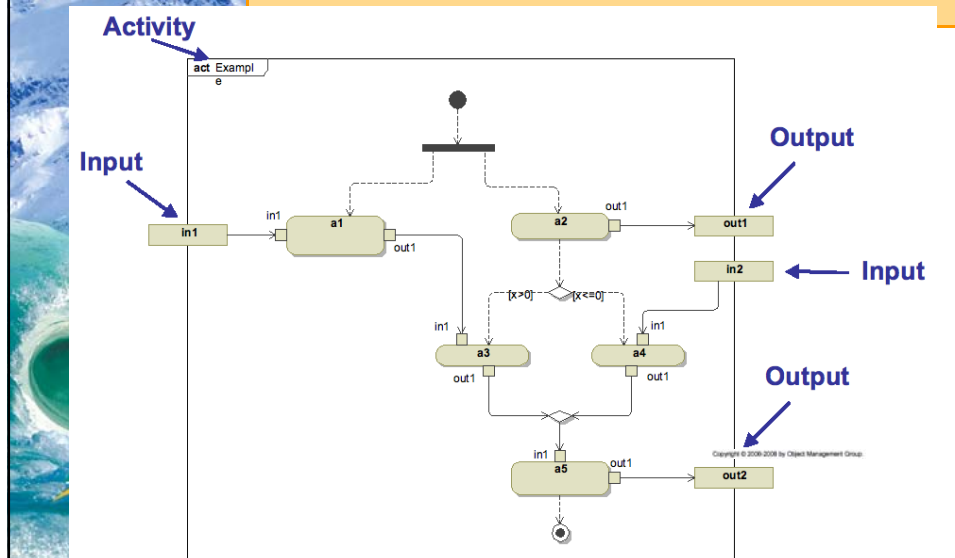




	<h2>Activity Diagrams (act)</h2>
	<ul style="list-style-type: none">• To specify<ul style="list-style-type: none">- controlled sequence of actions- the flow of inputs/outputs- control, including sequence and conditions for coordinate activities• Swimlanes<ul style="list-style-type: none">- to show responsibility of the activity

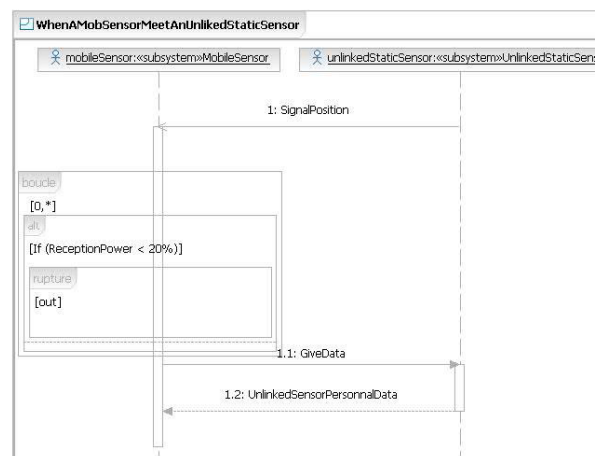
	<h2>Activity Diagrams (cont.)</h2>
	<ul style="list-style-type: none">• Improvements from UML:<ul style="list-style-type: none">- Continuous or discrete flow- Control operators<ul style="list-style-type: none">• to start/stop other actions- Overwrite and NoBuffer ports<ul style="list-style-type: none">• for continuous flows- Probabilities on transitions or parameters

Activity Diagrams (e.g.)



Sequence Diagrams (sd)

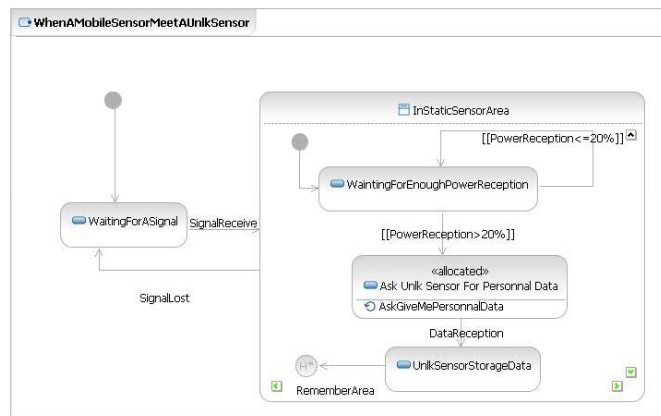
- Like UML





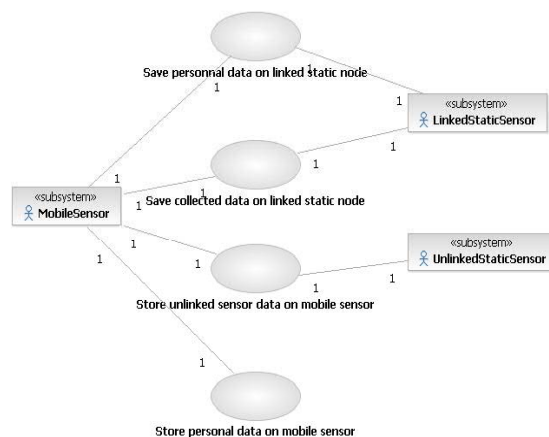
State Machine Diagrams (stm)

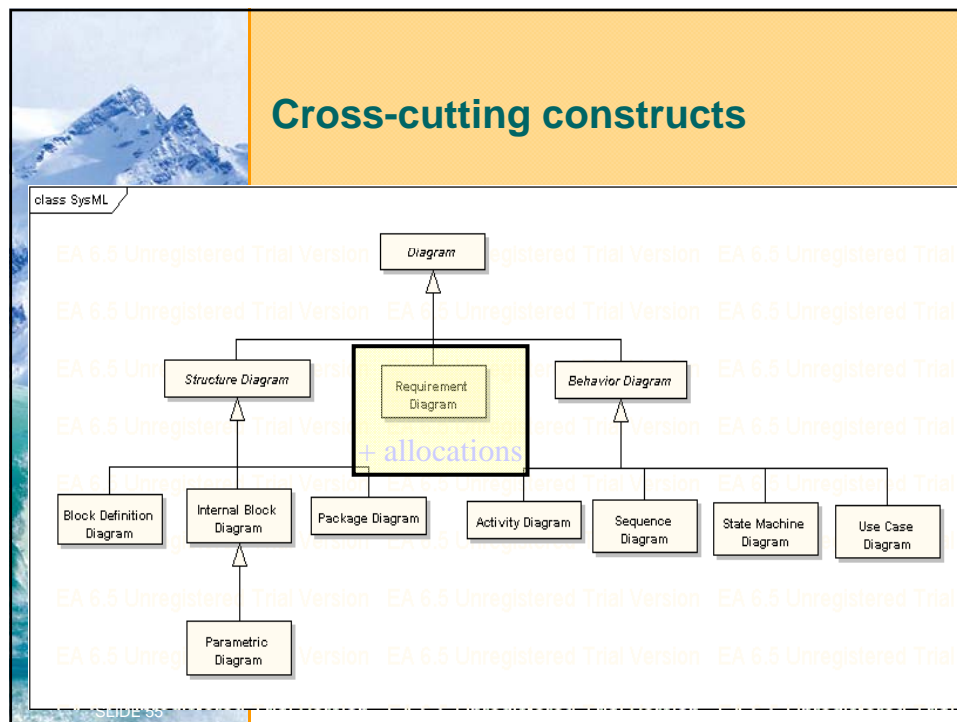
- Like UML




Use Case Diagrams (uc)

- Like UML





Allocation concept

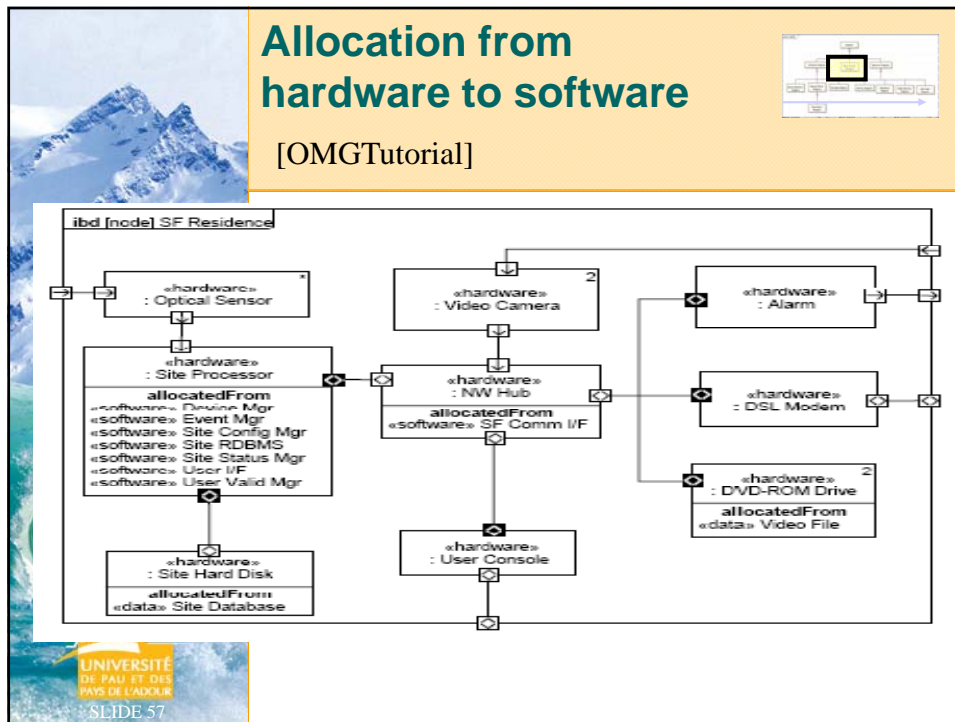


- General relationship between two elements of the model
- Different kinds of allocation
 - Functionality - component
 - Logical component – physical component
 - Software – hardware
 - ...
- Usable in a lot of different diagram kinds
- Also usable under graphical or tabular representation

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 56

Allocation from hardware to software

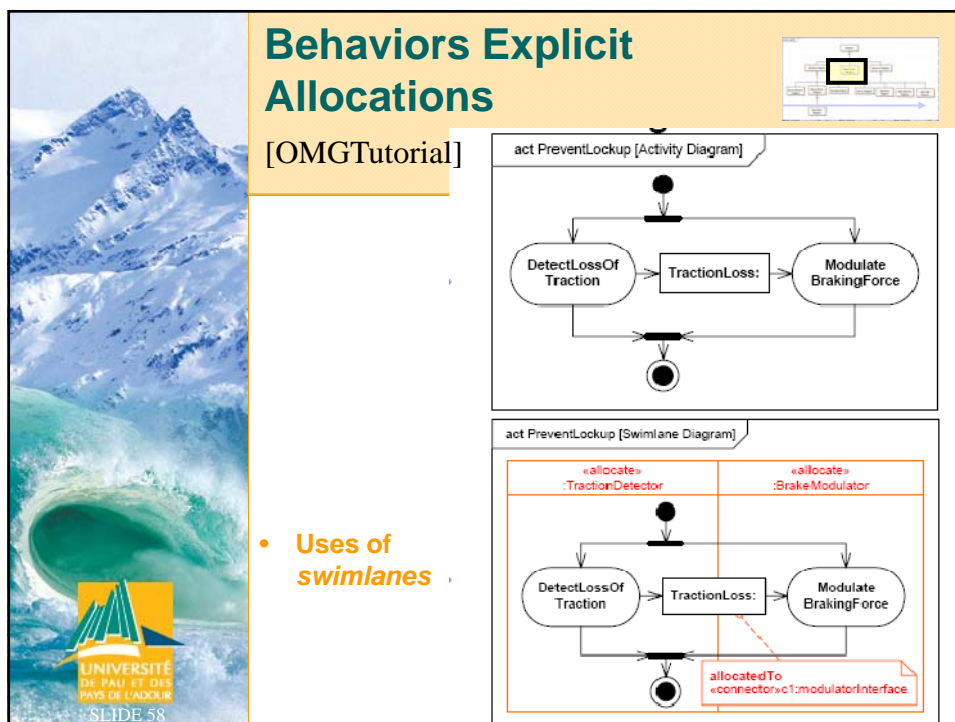
[OMGTutorial]



UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 57


Behaviors Explicit Allocations

[OMGTutorial]




- Uses of swimlanes


UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 58




Requirement diagrams

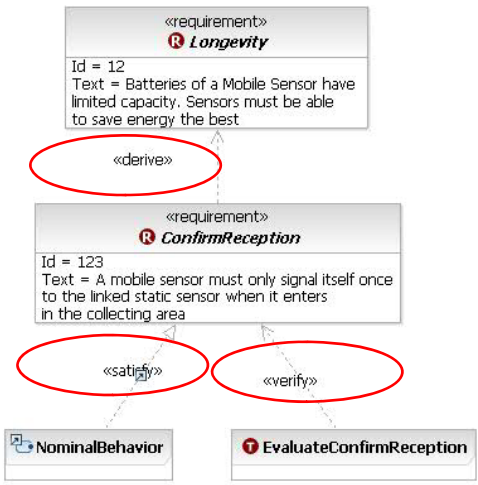


- **<<requirement>>** allows to represent a text based requirement
 - Includes one identifier *id* and some textual properties
 - Can add user defined properties
 - Can add user defined requirement categories
- **Requirements can be**
 - decomposed
 - specialized
- **Requirement relationships**
 - « deriveRqt », « refine »
 - « satisfy », « verify »
 - « trace », « copy »


 UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 59




Requirement Diagrams (e.g.)



```

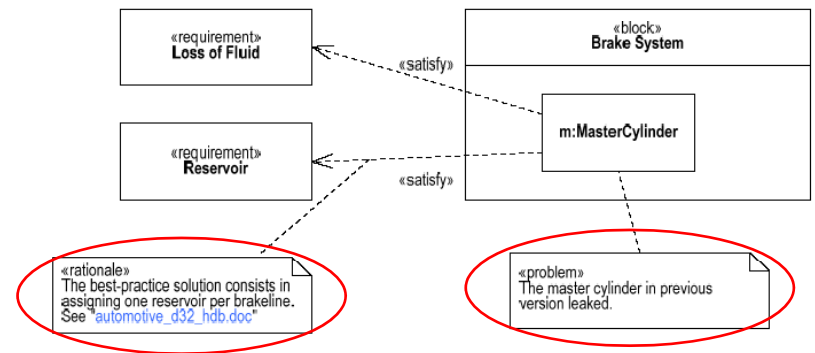
graph BT
    LR["<<requirement>>  
Ⓡ Longevity  
Id = 12  
Text = Batteries of a Mobile Sensor have limited capacity. Sensors must be able to save energy the best"]
    CR["<<requirement>>  
Ⓡ ConfirmReception  
Id = 123  
Text = A mobile sensor must only signal itself once to the linked static sensor when it enters in the collecting area"]
    NB["NominalBehavior"]
    ECR["EvaluateConfirmReception"]

    LR -.->|«derive»| CR
    CR -.->|«satisfy»| NB
    CR -.->|«verify»| ECR
          
```


 UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
2008/2009

Rational and Problems

bdd Master Cylinder requirements



bruel@irit.fr

Requirements Tables

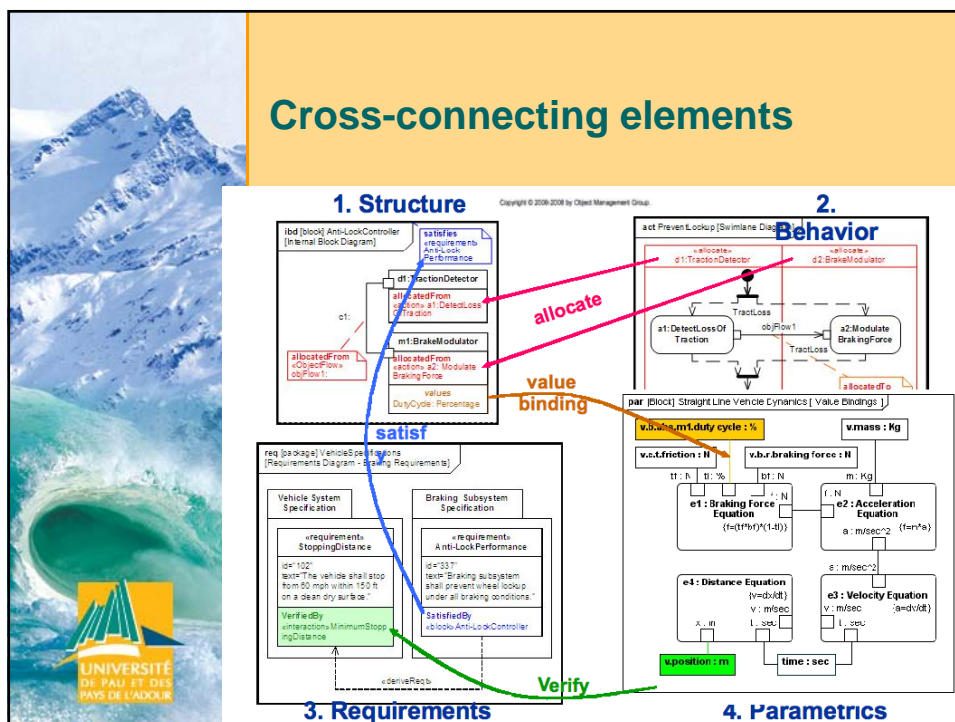
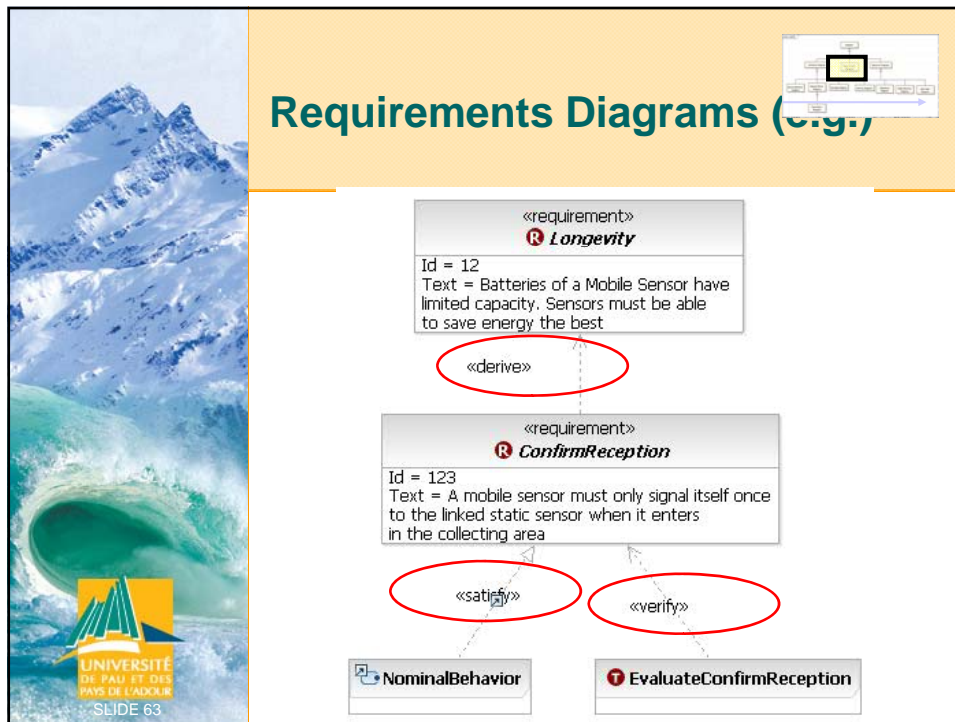
table [requirement] Performance [Decomposition of Performance Requirement]

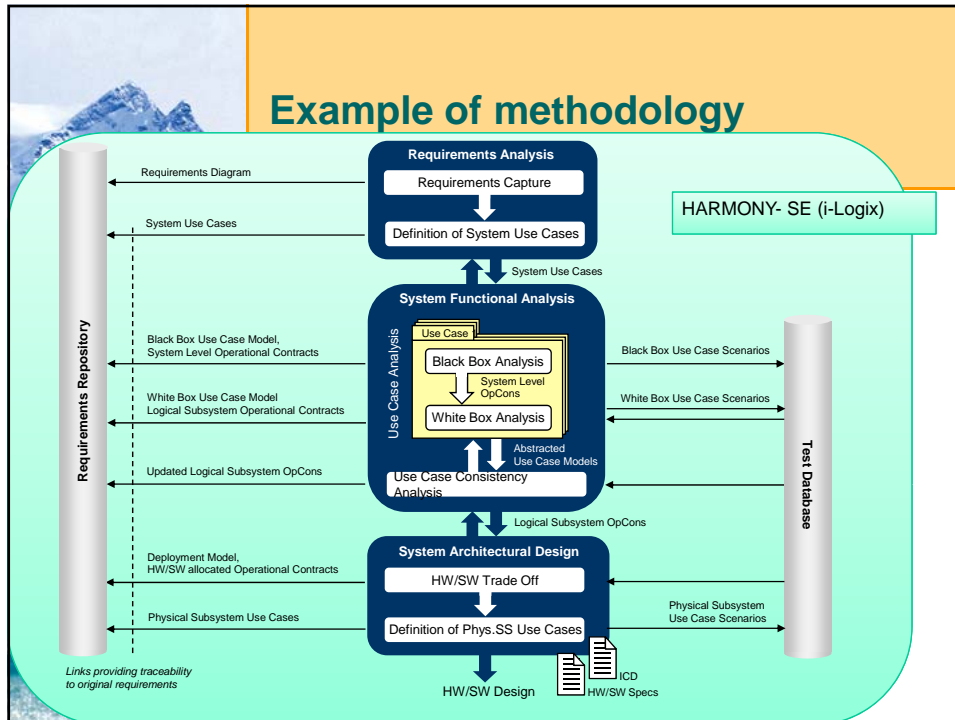
id	name	text
2	Performance	The Hybrid SUV shall have the braking, acceleration, and off-road capability of a typical SUV, but have dramatically better fuel economy.
2.1	Braking	The Hybrid SUV shall have the braking capability of a typical SUV.
2.2	FuelEconomy	The Hybrid SUV shall have dramatically better fuel economy than a typical SUV.
2.3	OffRoadCapability	The Hybrid SUV shall have the off-road capability of a typical SUV.
2.4	Acceleration	The Hybrid SUV shall have the acceleration of a typical SUV.

table [requirement] Performance [Tree of Performance Requirements]

id	name	relation	id	name	relation	id	name
2.1	Braking	deriveReq	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReq	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReq	d.2	Range			
4.2	FuelCapacity	deriveReq	d.2	Range			
2.3	OffRoadCapability	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement
2.4	Acceleration	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement
4.1	CargoCapacity	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement








Introduction to SysML


- Any question?



Content

- **Part I**
 - Introduction to System Engineering
 - Introduction to SysML
 - From UML to SysML
- **Part II**
 - Case studies
 - From SysML to UML
 - Use of Software Components in System Engineering
 - Questions and discussions


UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 67





From UML to SysML

- **Only some advices:**
 - Forget about class and objects
 - Think like an engineer
 - Focuss on the requirements
 - Play with SysML tools
 - Stay tuned (sysml forums and lists)

UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR

	<h2>From UML to SysML</h2>
	<ul style="list-style-type: none">• Any question?• Time for a break? <p>??</p>

	<h2>Content</h2>
	<ul style="list-style-type: none">• Part I<ul style="list-style-type: none">- Introduction to System Engineering- Introduction to SysML- From UML to SysML• Part II<ul style="list-style-type: none">- Case studies- From SysML to UML- Use of Software Components in System Engineering- Questions and discussions

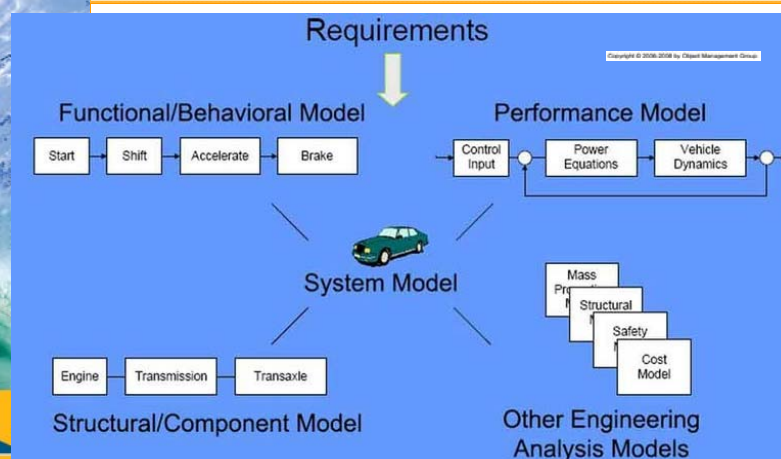
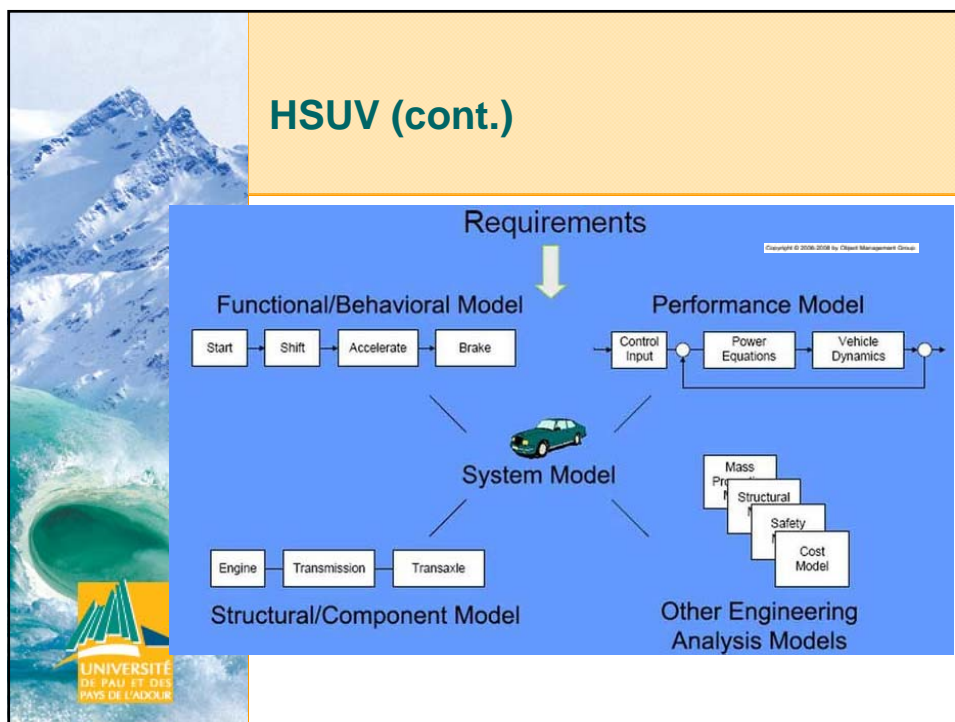


Hybrid Sport Utility Vehicle

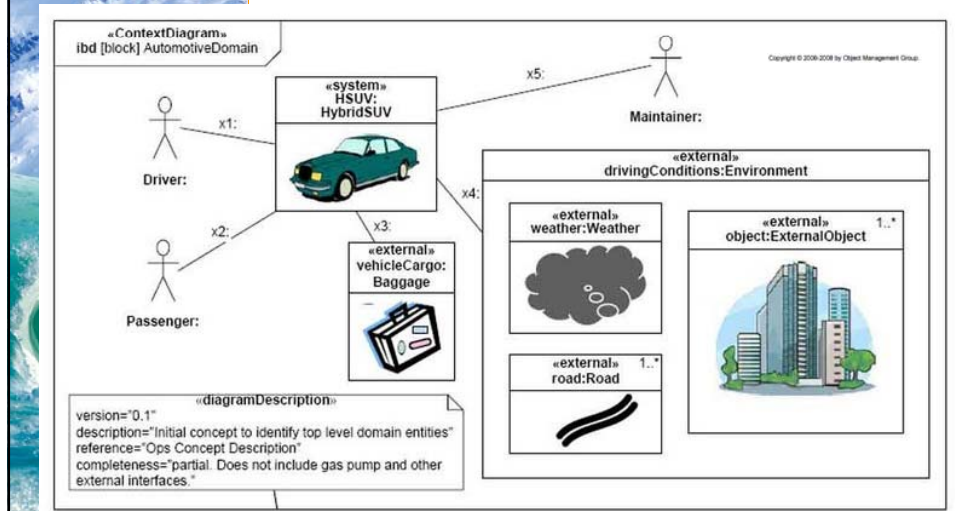
- From <http://www.uml-sysml.org/sysml>
- 2TUP process



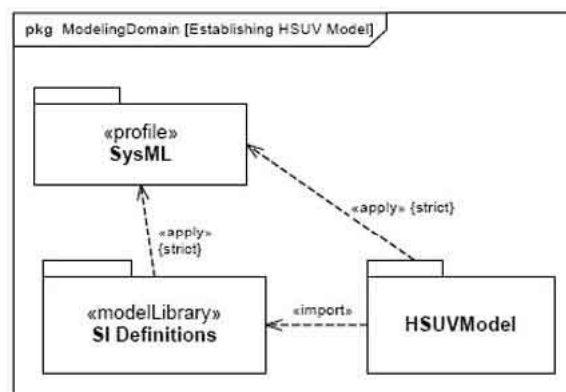
UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR



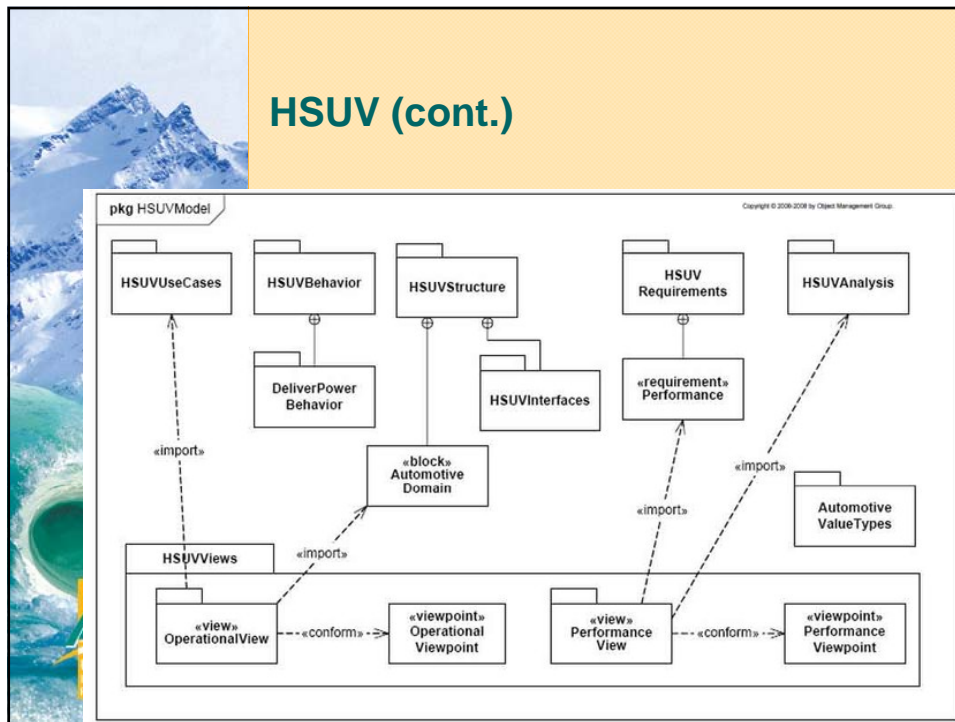
HSUV (cont.)



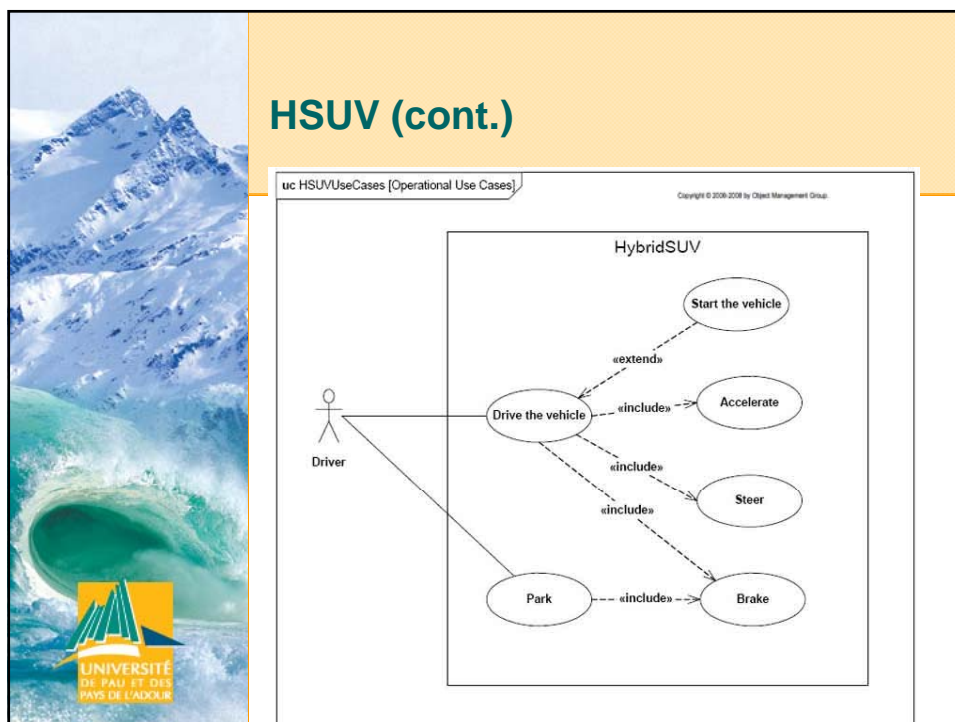
HSUV (cont.)



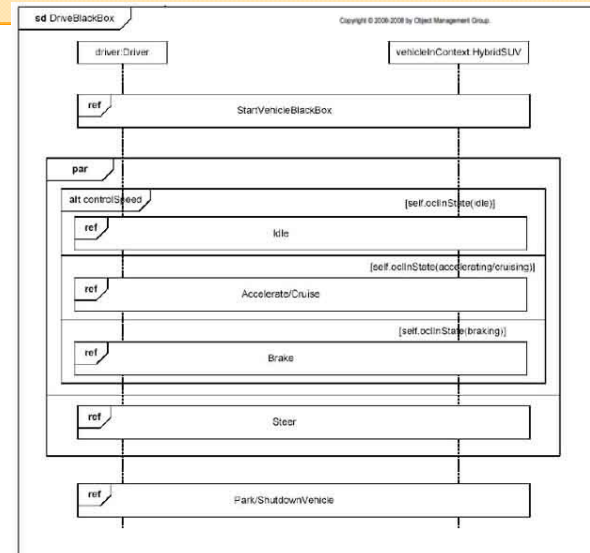
HSUV (cont.)



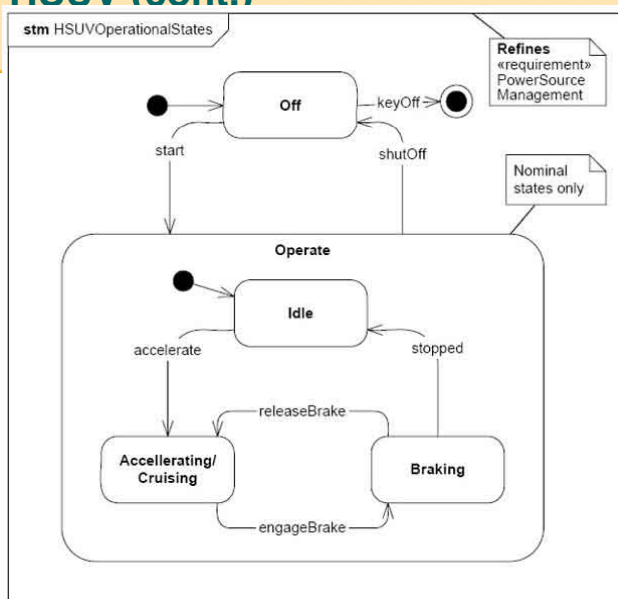
HSUV (cont.)



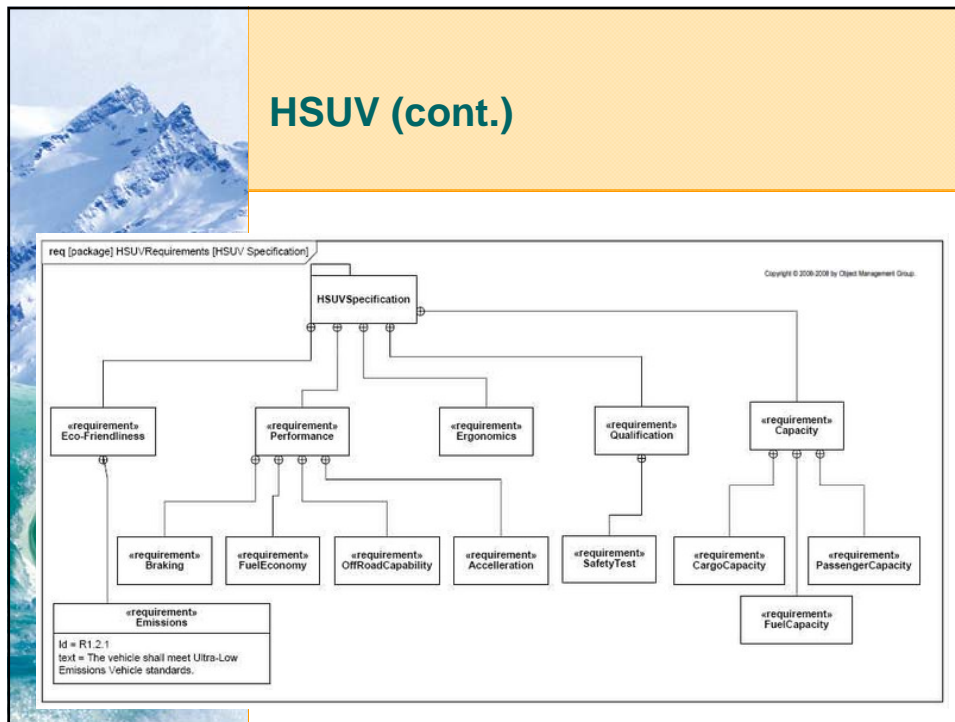
HSUV (cont.)



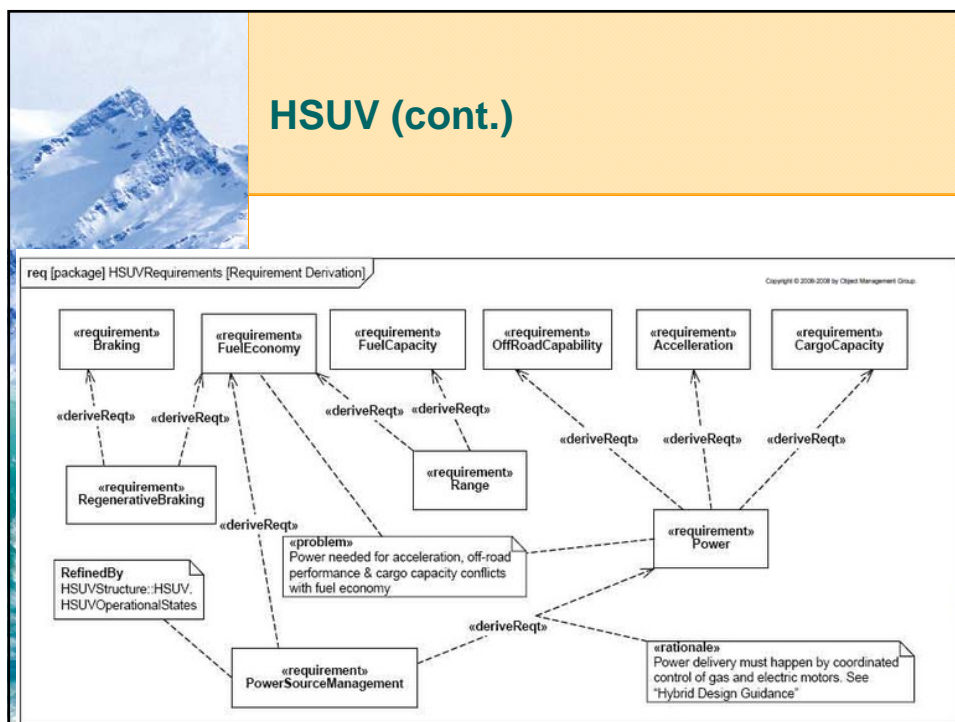
HSUV (cont.)

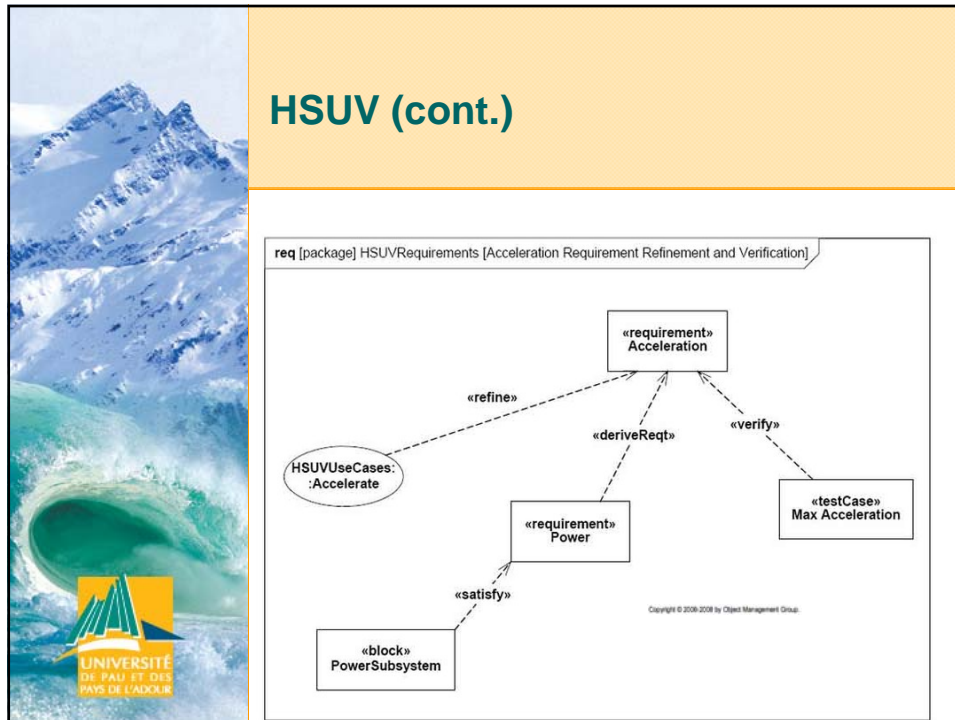


HSUV (cont.)



HSUV (cont.)






HSUV (cont.)

Copyright © 2006-2008 by Object Management Group.

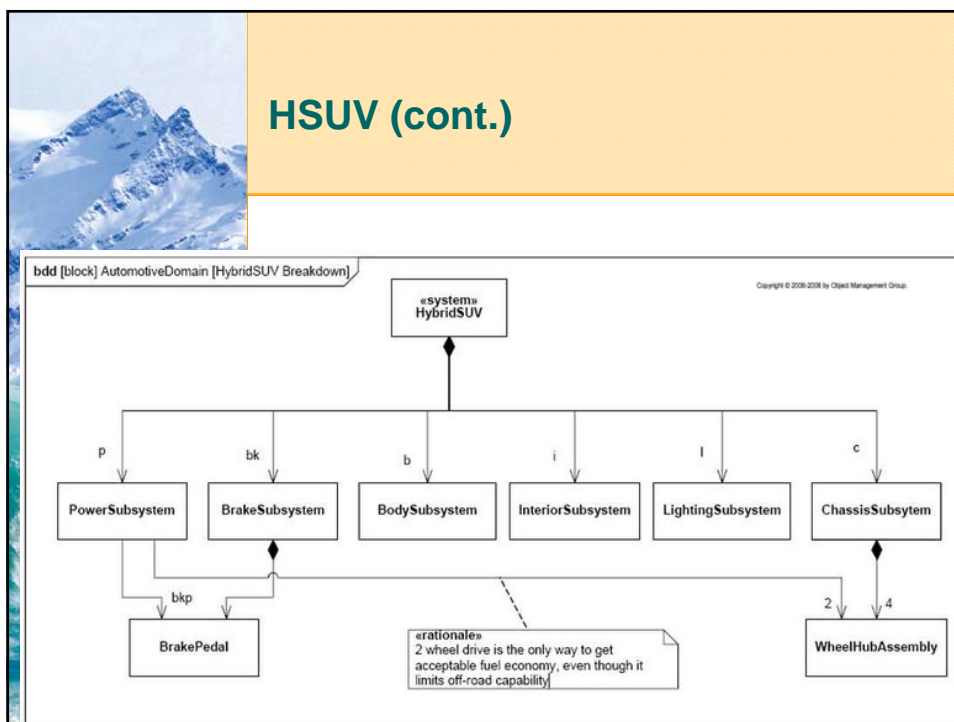
id	name	text
2	Performance	The Hybrid SUV shall have the braking, acceleration, and off-road capability of a typical SUV, but have dramatically better fuel economy.
2.1	Braking	The Hybrid SUV shall have the braking capability of a typical SUV.
2.2	FuelEconomy	The Hybrid SUV shall have dramatically better fuel economy than a typical SUV.
2.3	OffRoadCapability	The Hybrid SUV shall have the off-road capability of a typical SUV.
2.4	Acceleration	The Hybrid SUV shall have the acceleration of a typical SUV.

id	name	relation	id	name	relation	id	name
2.1	Braking	deriveReq	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReq	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReq	d.2	Range			
2.2	FuelEconomy	deriveReq	d.2	Range			
2.3	OffRoadCapability	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement
2.4	Acceleration	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement
4.1	CargoCapacity	deriveReq	d.4	Power	deriveReq	d.2	PowerSourceManagement



HSUV (cont.)

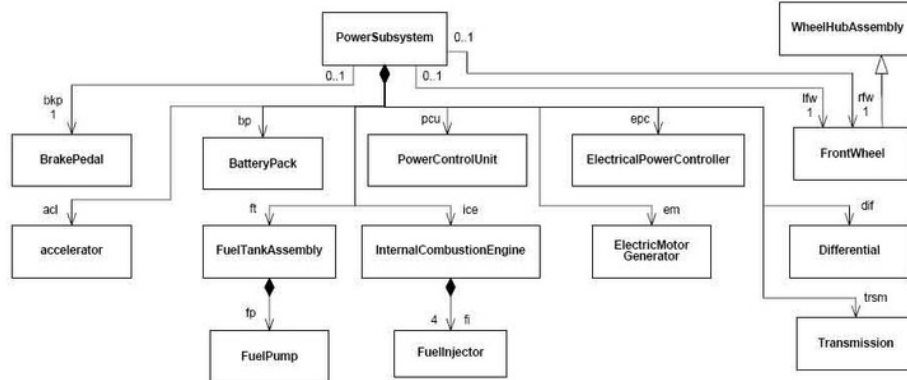
- Requirements have been fully studied
- Let's go inside the box!



HSUV (cont.)

bdd [block] HSUV [PowerSubsystem Breakdown]

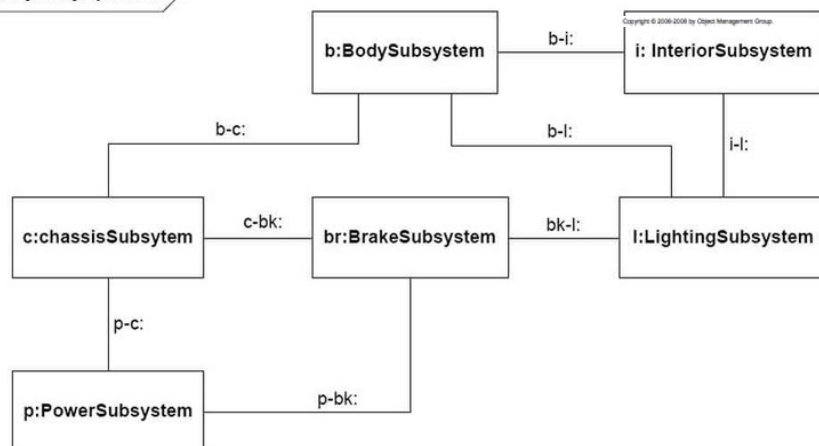
Copyright © 2006-2008 by Object Management Group.



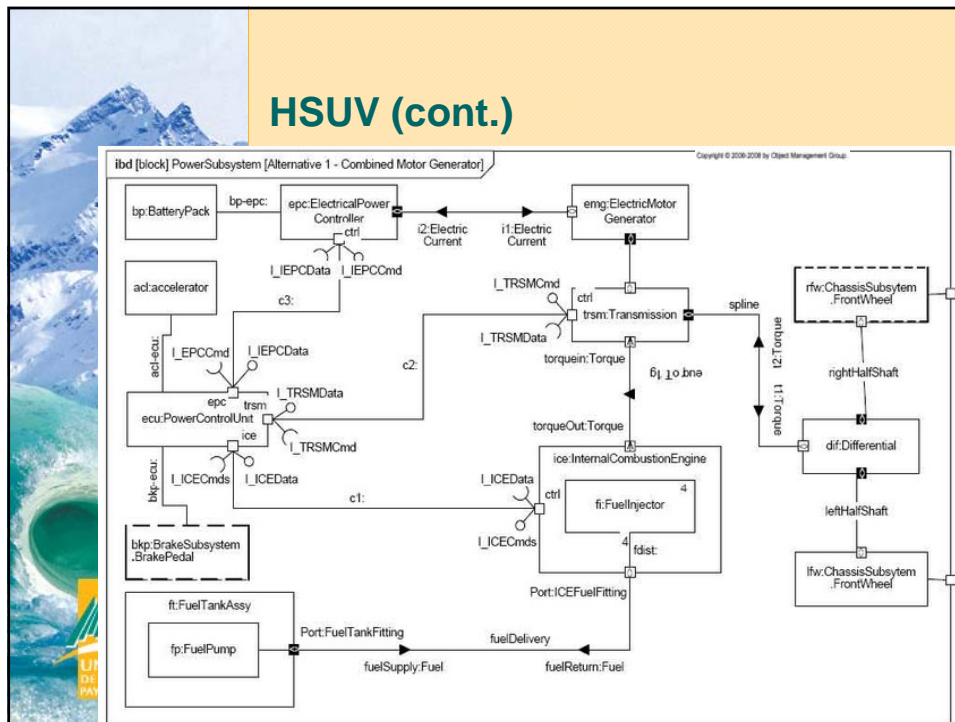
HSUV (cont.)

ibd [block] HybridSUV

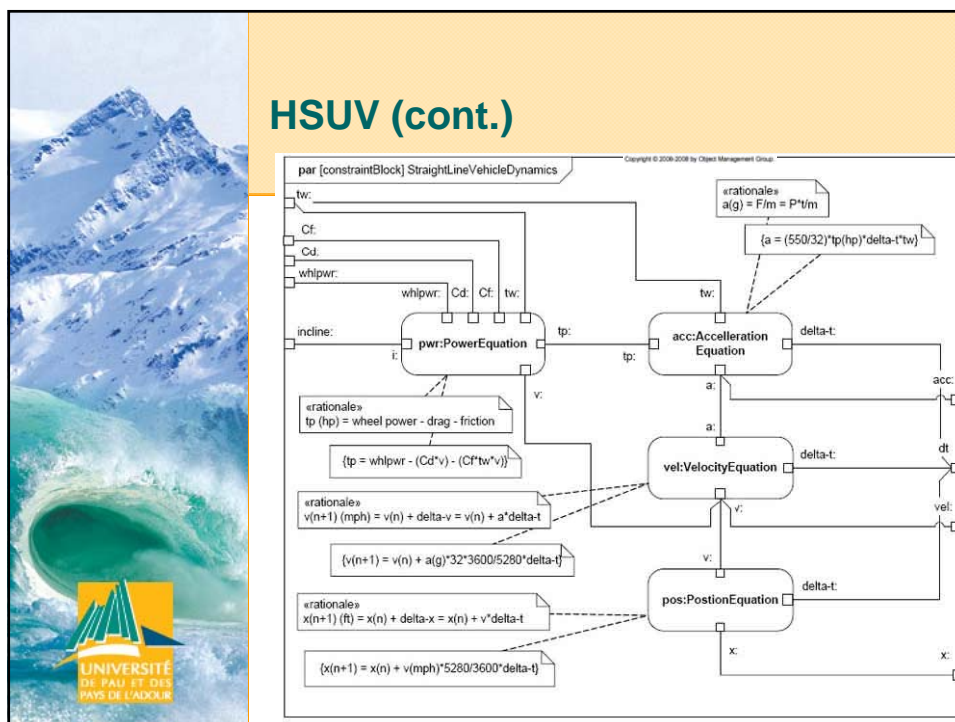
Copyright © 2006-2008 by Object Management Group.




HSUV (cont.)



HSUV (cont.)






Content

- **Part I**
 - Introduction to System Engineering
 - Introduction to SysML
 - From UML to SysML
- **Part II**
 - Case studies
 - From SysML to UML
 - Use of Software Components in System Engineering
 - Questions and discussions


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 89



From SysML to UML

- **For SysML some viewpoints will provide information to be allocated to several engineering domains**
 - Hardware, software, ...
- **Part of the software specification is include in the system specification**
 - Inheritance of system specification
 - Problem of the software specification grain
 - Problem of retro-specification from Sw to System


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 90



From SysML to UML

- **A Wild subject**
 - No existing formalized method nor cookbook
 - Just some good practices
 - Examples of requirements and deployment [Art08]


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 91



Requirement Traceability

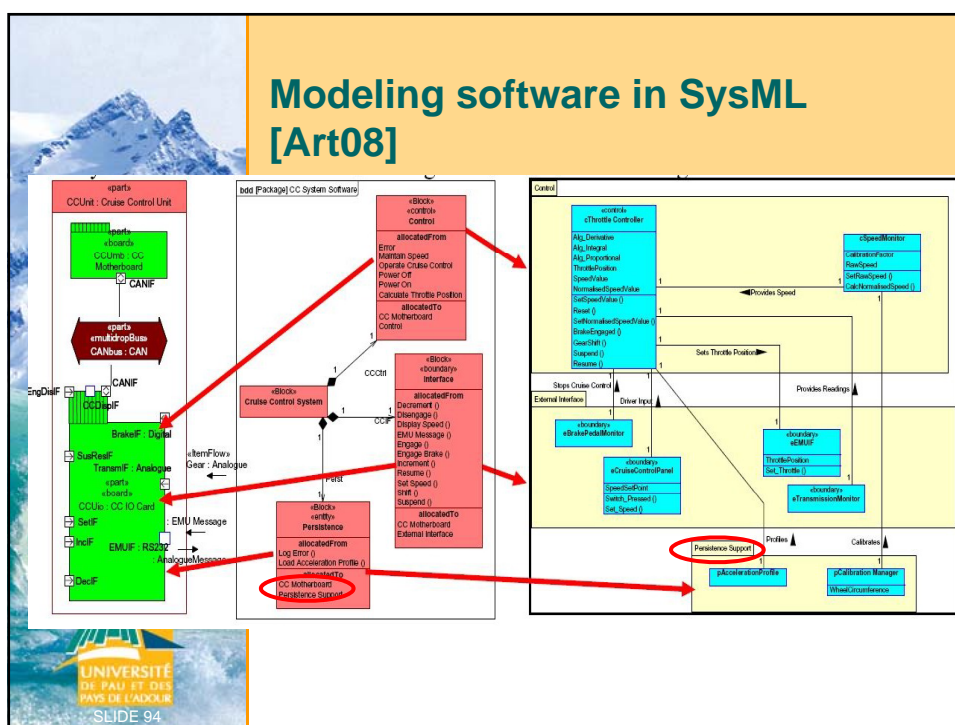
- **Software requirements specification**
 - Normally be created as part of SE process
 - Integrating into System requirement into SysML/UML model will improve traceability
- **Principe**
 - Add some relationships between SysML and UML models
 - <<Trace>>
 - << Satisfy>>
 - <<Refine>>


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 92



Modeling software in SysML


- A block can represent a wide kind of things
 - System, sub-system, logical or physical components...
- A rapid approach consists in
 - Mapping directly between a block and a class
 - But ... the level of abstraction is not at the same level
 - Mapping one to one is not realistic
- Use of <<allocation>> concept






Use cases

- For simple systems, mapping is easy
- For large scale systems (systems of systems)
 - Problem of level of abstraction
- Example
 - System = *Drive Vehicle* use case
 - Sub-system = cruise control system
- SW engineer will analyze CC functional requirements and create a subset of use cases to the level of the CC software




SLIDE 95

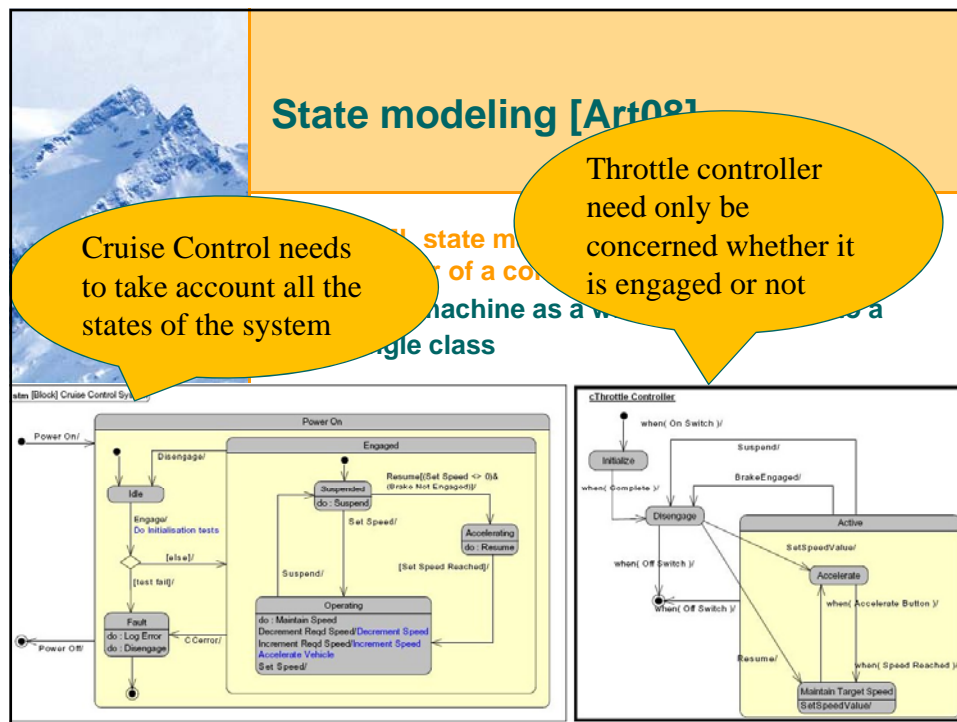


Flows, signal and events

- Events and signals are generally atomic operations
 - Will usually map directly to the UML model
 - Will be necessary to consider any linked state transitions or behavior invocations
- Problem will occur within continuous flows




SLIDE 96



Content

- **Part I**
 - Introduction to System Engineering
 - Introduction to SysML
 - From UML to SysML
- **Part II**
 - Case studies
 - From SysML to UML
 - Use of Software Components in System Engineering
 - Questions and discussions

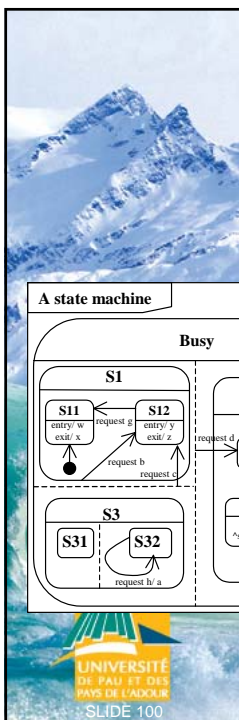
UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 98



UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 99

And Software Components?

- **System-level reuse is**
 - relatively immature compared to software reuse
- **Use of SW components in a system process**
 - Problem of philosophy (how to be sure that the reused component exactly corresponds to requirements)
 - Problem of correspondence between System block and Software components
 - What about systems norms (e.g. DO178)?

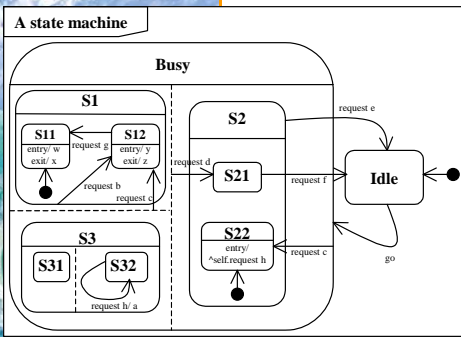


UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 100

A solution with PauWare?

- **Building components from state machines**

A state machine



UML Component Diagram

My component

```

<<component>>
-+()
-request_h()
-vr()
-x()
-y()
-z()

```

A provided interface

```


go()
request_b()
request_c()
request_d()
request_e()
request_f()
request_g()

```

1 -- 1

1 -- behavior


A state machine



Content

- **Part I**
 - Introduction to System Engineering
 - Introduction to SysML
 - From UML to SysML
- **Part II**
 - Case studies
 - From SysML to UML
 - Use of Software Components in System Engineering
 - Questions and discussions


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 101



SysML, What Else?

- **Modeling and Analysis of Real-Time and Embedded Systems (MARTE)**
 - A UML profile
 - Real-Time Oriented
- **Architecture Analysis and Design Language (AADL)**
 - A architecture description language
 - Verification and Validation using tools
 - Extension mechanisms (parser)
 - <http://www.aadl.info/>


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 102



Opened Questions 1/

- **What maximum granularity level Using SysML?**
- **How do you translate a SysML model in a UML model?**
 - How do you keep links between requirements and corresponding model elements?
 - Model transformation?
- **Methodological consideration**
 - Which existing method could be adapted within SysML language?


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 103



Opened Questions 2/

- **SysML genericity : would be a matter?**
 - Generic blocks must be stereotyped
- **What about the timing concerns?**
 - Cooperation/integration with Marte?
- **Lack of verification tools for requirement validations**


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 104



Conclusion 1/

- **SysML is**
 - a specific language for complex systems
 - strongly UML-Based
 - focusing on the analysis step
- **SysML is not**
 - a language designing to be used with a specific method
 - Able to be used without more specific languages (hw, sw..)


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 105



Conclusion 2/

- **SysML has improved**
 - The links between models elements and requirements)
 - The capability to include mathematical features in models
 - Continuous flows design
- **SysML has failed**
 - Covering all the system design lifecycle
 - Fully interacting with UML

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 106



Conclusion 3/

- **SysML helps**
 - To identify high granularity software components
 - To define access points/drivers between software and hardware components
- **SysML doesn't help**
 - To directly translate a software component designed in SysML to a UML software component


UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 107




SysML Tools

- **Artisan Software / Real-time Studio**
 - <http://www.artisansw.com/>
- **Embedded Plus / SysML Toolkit for RSDP**
 - <http://www.embeddedplus.com/>
- **I-Logix / Rhapsody**
 - <http://www.ilogix.com/sublevel.aspx?id=53>
- **SparxSystems / Enterprise Architect**
 - <http://www.sparxsystems.com/sysml>
- **Telelogic / Tau G2**
 - <http://www.telelogic.com/products/tau/index.cfm>

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 108







UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 109

Sources 1/

- **Web sites**
 - <http://www.omgsysml.org/>
 - <http://www.sysml.org/>
- **Tutorials**
 - [OMGTutorial] **OMG Systems Modeling Language (OMG SysML™) Tutorial**, *Sanford Friedenthal, Alan Moore, Rick Steiner*

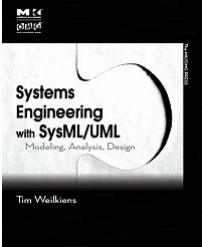




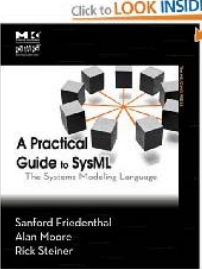
UNIVERSITÉ
DE PAU ET DES
PAYS DE L'ADOUR
SLIDE 110

Sources 2/


- **Books**
 - **A Practical Guide to SysML: The Systems Modeling Language**, Sanford Friedenthal, Alan Moore, and Rick Steiner, ISBN-13: 978-0123743794
 - **Systems Engineering with SysML/UML: Modeling, Analysis, Design**, by Tim Weilkiens, ISBN-10: 0123742749



Systems
Engineering
with SysML/UML
Modeling, Analysis, Design
Tim Weilkiens




Click to **LOOK INSIDE!**
A Practical
Guide to SysML
The Systems Modeling Language
Sanford Friedenthal
Alan Moore
Rick Steiner



Sources 2/

- **Documentations**
 - [Sys08] « The OMG SysML specification v1.1 », OMG partners, formal/2008-11-01
 - [Art08] “Building Bridges Between System and Software With SysML and UML”, Matthew Hause, Francis Thom, INCOSE Intl. Symposium, June ' 2008
- **To go farer**
 - [HR04] D. Harel and B. Rumpe, "Meaningful Modeling: What's the Semantics of 'Semantics'?", *Computer* 37:10 (October 2004), IEEE Press, 64-72

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 111



Any questions?

nicolas.belloir@univ-pau.fr

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
SLIDE 112

