Systemx
INSTITUT DE RECHERCHE TECHNOLOGIQUE

Accélérateur
de la transformation
numérique

Model Identity Car (MIC)

A method to support modeling and simulation of complex system

SMS Community meeting - 8th of December 20

Eric Landel (eric.landel@irt.systemx.fr)

Context of work

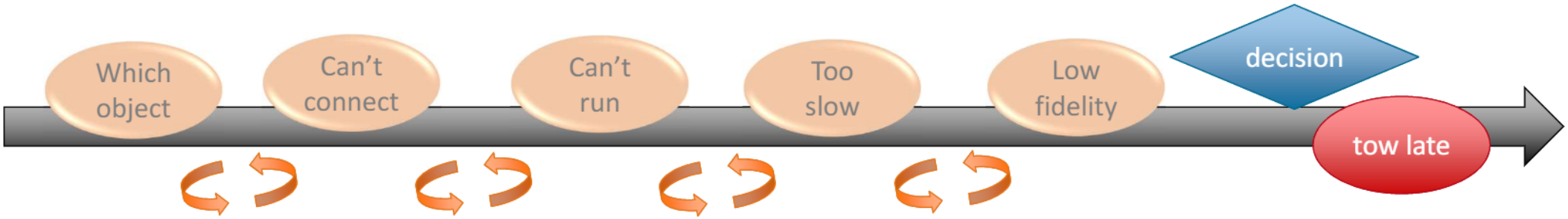
- **Renault : expertise workshop and thesis (G.Sirin 2015)**
- **SystemX : projects SIM and AMC**
  - Automotive (Renault, PSA, Valeo)
  - Aeronautic (Airbus, Arianne group)
- **Cooperation in automotive consortium**
  - French (SystemX)
  - German (prostep-ivip)
  - Japan (METI)

A Model Identity Card to Support Simulation Model Development Process in a Collaborative Multidisciplinary Design Environment January 2015 IEEE Systems Journal (G.Sirin, C.Paredis, B.Yannou, E.Coatanéa, E.Landel)

Objectives of MIC

- **For supporting simulation activities**
  - of complex systems : many components (many links between many components)
  - in complex organization : many stakeholders (internal and external)
- **In a way to support decision making :**
  - Delivering information at the right time (T)
  - With the right level of credibility (Q)
  - For a reasonable budget (C)
- **MIC is built as a generic concept**
  - Easy to apply
  - Comprehensible
  - Independent of tools

How to perform simulation of complex systems



Aggravating factor : models are elsewhere,… not on the computer

Concerns during activities of system simulation:

1. Identification of the system : can't link the model to the object (consistency)
2. Connecting/assembling : can't connect : port definition
3. Environment : can't run : library, OS,..
4. Numerical : can't deliver results : sampling, convergence,…
5. Execution : execution is too slow (RT ?)
6. Accuracy : fidelity is not adapted (for decision)

- **Concerns during activities :**

1. Identification of the system
2. Connecting/assembling
3. Environment
4. Numerical
5. Execution
6. Accuracy

=> Integration

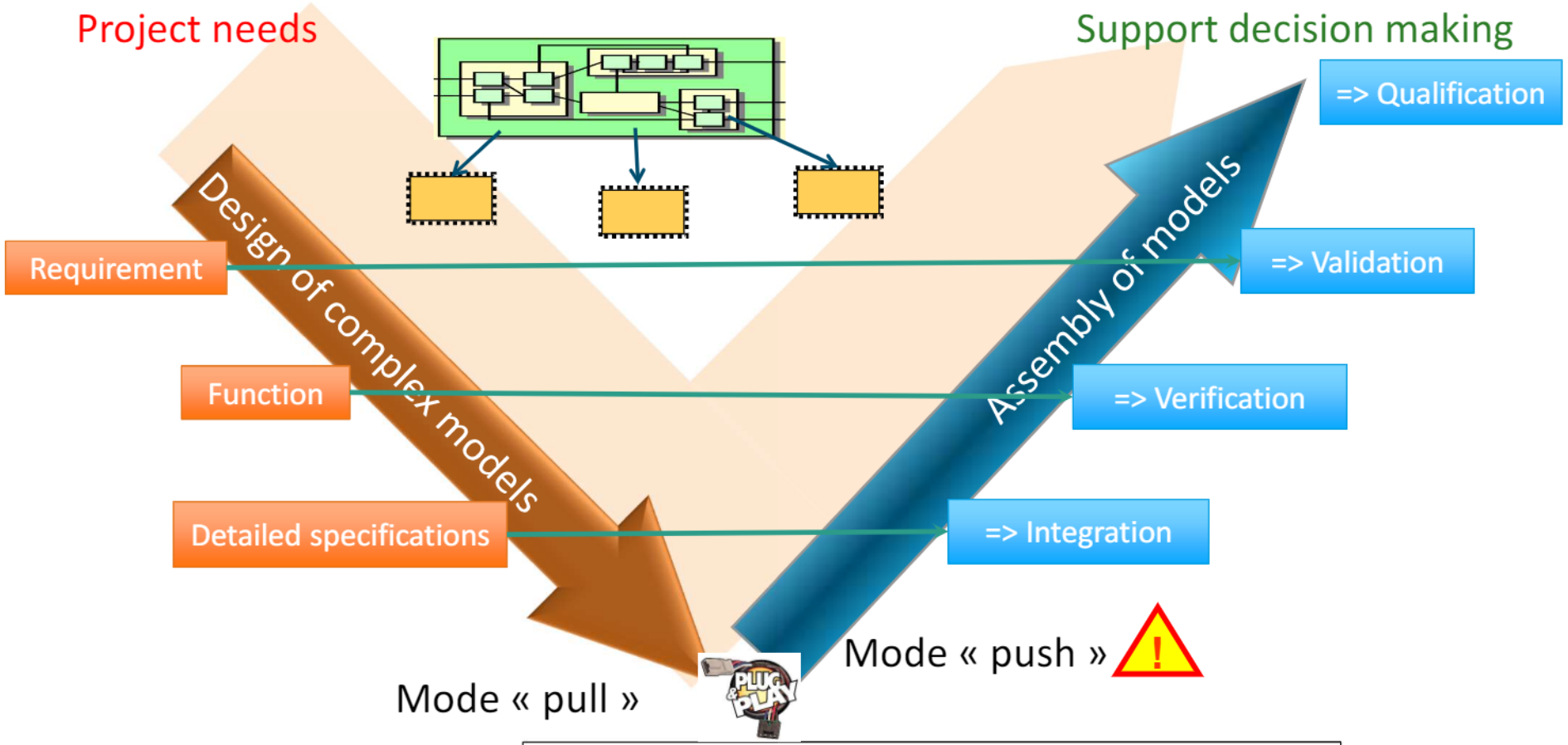
=> Verification

=> Validation

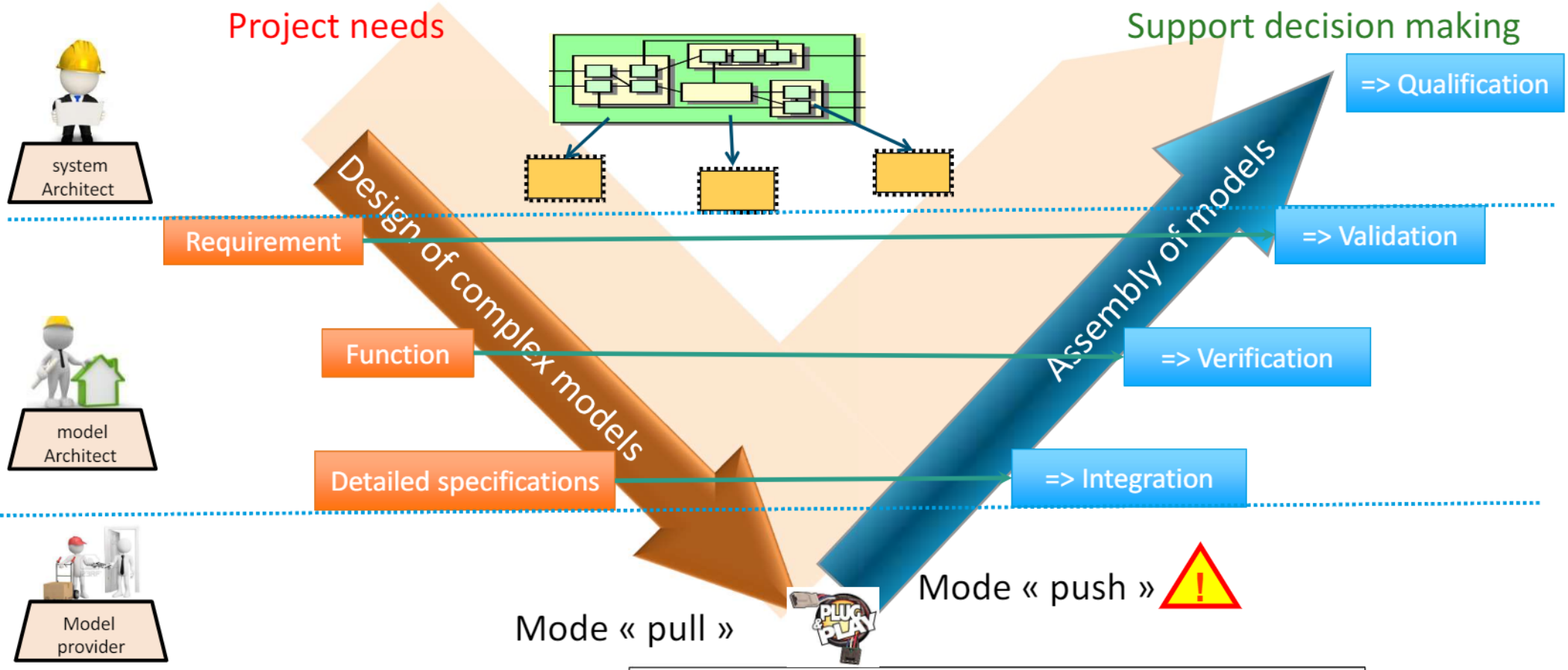
=> Qualification

- **Concerns can be treated by anticipation of IVVQ activities**

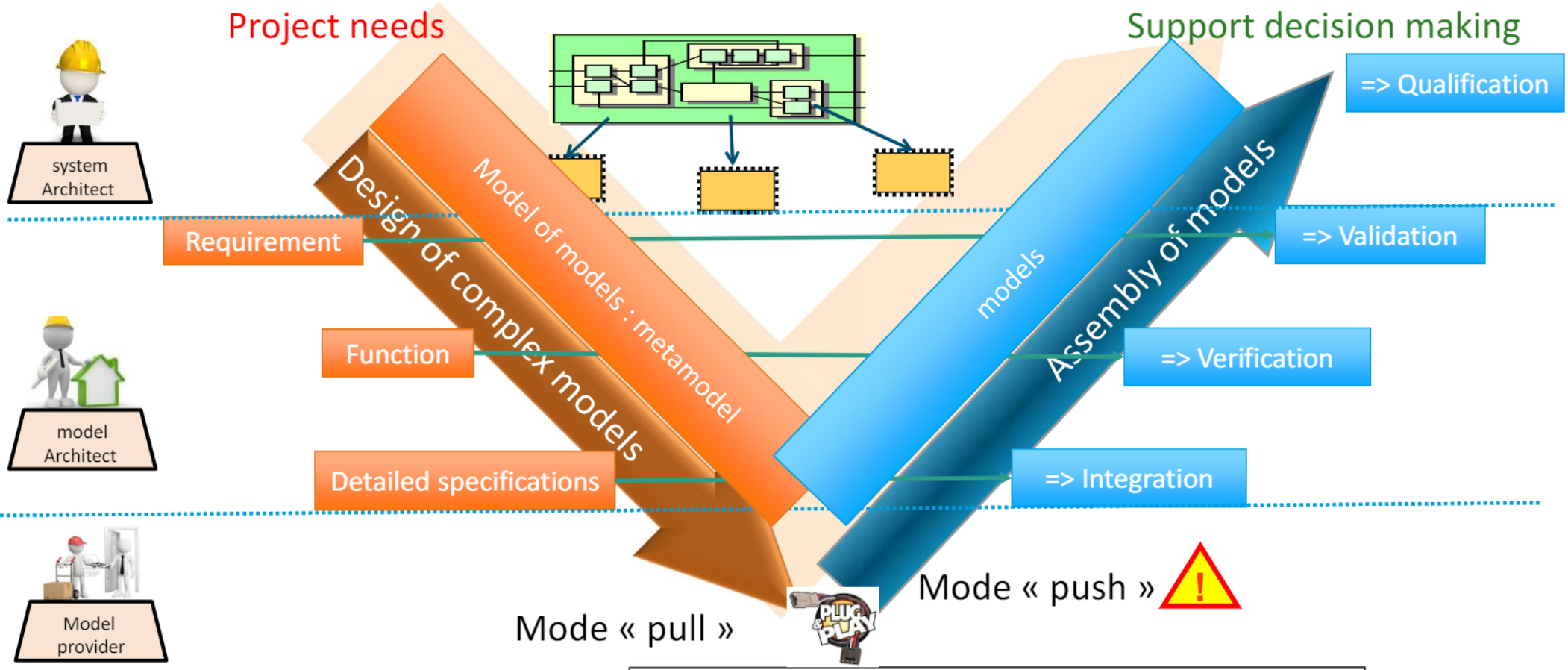
Simulation of Systems driven as a system of simulation



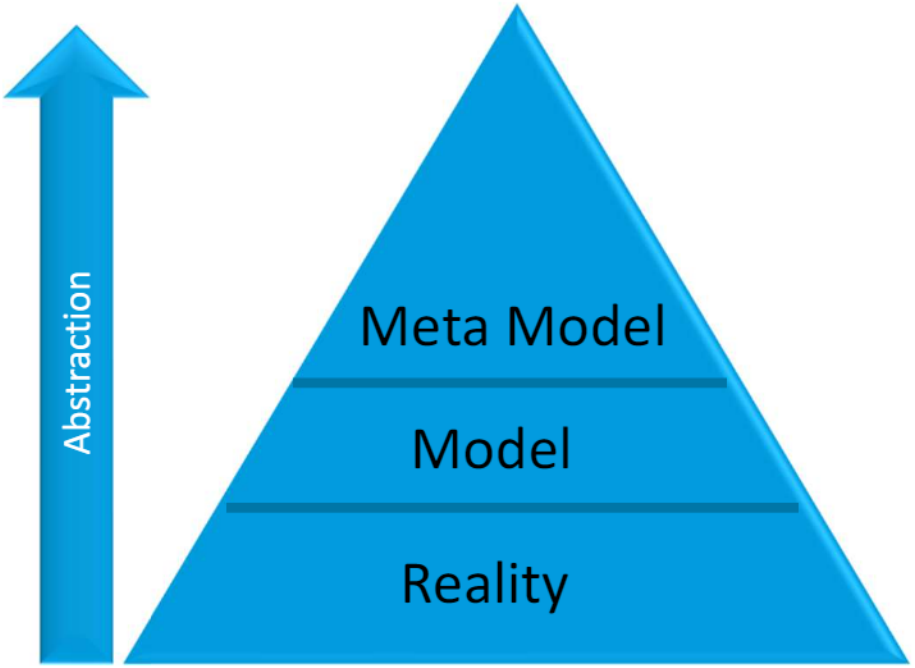
Simulation of Systems driven as a system of simulation



Simulation of Systems driven as a system of simulation

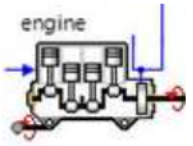


Model Identity Card is a metamodel to support design activities



MIC : a language to describe a model

Metamodel of the engine

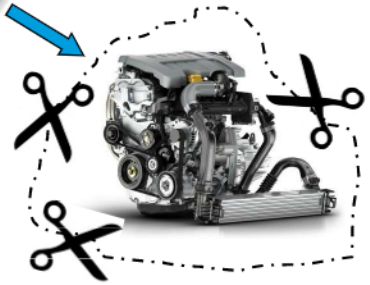


Model of the engine

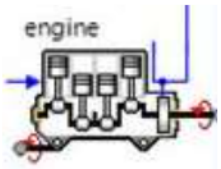
The engine in the car

Model Identity Card : 5 sets of items

Real object



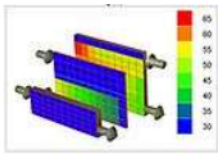
Model



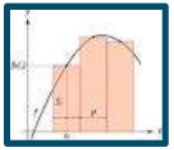
Meta- model

interface

object / physic

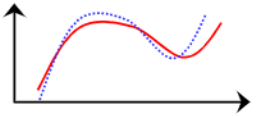


methods



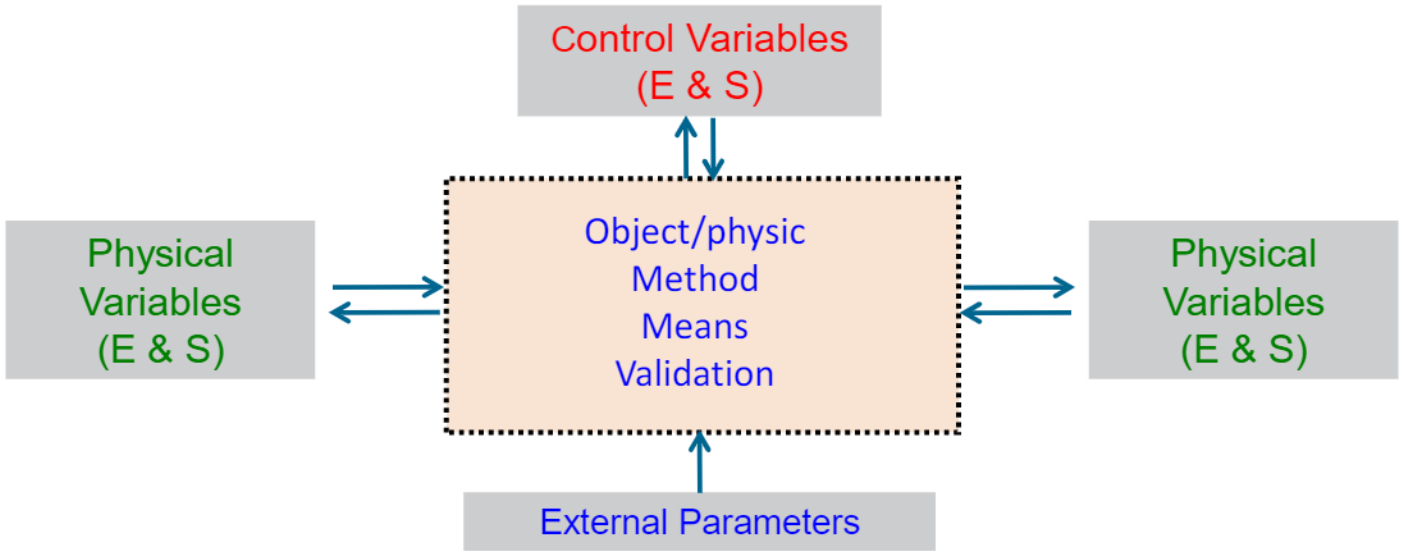
means

validation



Model Identity Card : a model of the numerical model

MIC is integrated in a block diagram



Integration : definition of interfaces

List of the port and set of ports

- Type
- Unit
- In/out
- Range of validity

Name	**Name of the variable. The variable can be directly at the interface of the model (as in an FMU), or it can be accessible through a port.**
Description	Description of the variable.
Type	Data type of the variable. The variable can be a multidimensional array made of a unique type of data or of multiple types of data. This attribute is typed as "string", but its use can be facilitated with a set of basic values (e.g. data types from languages like Python).
Unit	Unit of the variable. Typed as "string", but the symbols of the SI base units (s, m, kg, …) and derived units (rad, Hz, N, …) are recommended as a basic choice. Permit to know the branch of physics associated to the variable. Can potentially be used for an array of values with different units.
Default value	Default value of the variable. If the variable is an output, the default value represents the initial value which cannot be computed at the first time step. If the default value is used for an input variable, its meaning should be specified in the documentation of the simulation model. Units should not be repeated, as they are already defined in their dedicated attribute. Can potentially be an array of values.
In/Out	Defines whether the variable is an input or an output, in case of causal simulation.
Validity domain	Validity domain of the variable. If the variable is an input, the validity domain represents the values which have been taken into account during the development of the simulation output. If the model has been verified and validated, the outputs are only guaranteed for inputs within their validity domain. If the validity domain is used for an output variable, its meaning should be specified in the document of the simulation model.

To secure connection and avoid ambiguities

=> Integration

MIC : general information about the object

Description of the model :

- The object of the model
- The reference of the model
- The author of the model

To secure the traceability

=> Qualification

Attribute	Description
Name	Name of the simulation model. Typically short, clear, and based on what the model represents or computes. Is not necessarily a unique identifier for the simulation model (if a unique identifier is needed, it can be managed outside the MIC).
Description	Summarizes what the simulation model represents, and what it is used for. Should be as unambiguous as possible (e.g. with a clear identification of the represented system). Can potentially summarize information from the rest of the MIC (such as modelling hypothesis). Can reflect multiple uses.
Owner	Person or institution responsible for the simulation model. Can be the developer or author. The owner can potentially provide further information about the simulation model, or make the simulation model available.
Life cycle state	Life cycle state the simulation model is currently in. Can permit to distinguish a MIC used for the preliminary specification of a simulation model which has not been developed yet, a MIC used to describe a simulation model whose development is complete, or a MIC used for the modification of a simulation model, for example.
Version	Version of the simulation model (not of the MIC). Not used during the preliminary specification of a simulation model. Updated when the development is complete. External versioning solutions can also be used to further track modifications (during the preliminary specification, or during the development).
Version date	Release date of this version of the simulation model in the following format : YYYY-MM-DD for a day, YYYY-MM for a month, or YYYY for a year.
Confidentiality	Identifies who has access to the simulation model. Can be set to "Public" if anyone has access. Can include multiple names of persons or institutions. Can also be set to a confidentiality level defined on a clear scale (e.g. a within a given company), but this scale needs to be referenced. Can also identify a legal document such as a Non Disclosure Agreement (NDA).
License	Defines the rules governing the access to the simulation model : royalties to pay, restriction to noncommercial use or nonmilitary use, right to modify, etc. A legal contract can be referenced.

Verification : Numerical

Numerical scheme

- Fields
- Scheme
- Time scale

Attribute	Description
Explicative text	Modelling choice explained in natural language. Can be a modelling hypothesis or a note about the limits of the results. There can be multiple modelling choices, but there is only one explicative text per modelling choice
Modelling field	The modelling choice is often about taking into account or neglecting phenomena in different modelling field. This first attribute permits to associate the modelling choice to a modelling field. It is typed as a "string", but the following values are recommended as a basic choice: "Solid mechanics", "Mechanics of materials", "Fluid mechanics", "Acoustics and vibrations", "Electromagnetics", "Thermal", "Chemistry", "Optics", "Biology", "Sociology".
Type of choice	This second attribute permits to explain whether the modelling choice is about taking into account (i.e. modelling) or neglecting a phenomenon in this modelling field.
Time scale	This attribute measures the time scale of the phenomenon which has been taken into account or neglected. Fast or transient dynamics (e.g. during a motor startup, during the energization of a transformer, during a wind burst, …) can require more complex modelling than steady states but they are required to obtain realistic values when the simulation is run with short time steps. The time scale should have a clear unit (e.g. ms).
Behavior specification	Documentation regarding the behavior of the simulation model. Can be system requirements. Can be a text (formatted or not), or a link to a document.
Model type	Type of simulation model, defined with keywords. The attribute is a "string", but the following values are recommended as a basic choice: "Discrete", "Continuous", "Deterministic", "Stochastic", "Static", "Dynamic", "Causal", "Acausal", "Bond graph", "Block diagram", "Transfer function", "State Machine diagram", "Neural network", "Empirical data"
Default Solver name	Indicates a default solver which can be used for the computation of the model's states over time when a set of ordinary differential equations is define
Step size	Indicates a default step size representing the time intervals between the computations the model's states by the solver. Can be defined either as "Variable", or as a fixed numerical value. In the latter case, the step size should be associated to clear time units (e.g. ms). The step size should be consistent with the time scales of the modelled and neglected physics.
Embedded	Indicates whether the solver is embedded in the model's file (as in an FMU for Co-Simulation) or not. If not, the solver must be provided by a simulation tool.

To prevent numerical concerns

=> Verification

Integration and verification : Hardware

Characteristics of hardware

- CPU
- RAM
- Data storage

Attribute	Description
Name of the reference hardware	A reference hardware is a machine which is known to be adapted to the simulation model. It is possible to identify it with an understandable name which helps to quickly understand the type of machine which can be used to run the simulation. The multiplicity of the parent element ("Reference hardware and performance") permits to provide several examples, but this should not be used to archive the full history of the simulation execution.
Time factor	Ratio of the time it takes to run the simulation model (with the reference hardware) and the simulated time, in case of dynamic simulation computed as a function of time. Measures the performance of the simulation model. Can be used to estimate the usability of the simulation model with Hardware In the Loop (HIL). To be used with full simulation models, not with simulation blocks to be integrated in a greater simulation model.
CPU	Description of the Central Processing Unit (CPU) or main processor in the reference hardware. Performance measures should be associated to clear units (e.g. GHz). The description of the CPU may not be limited to the clock frequency.
RAM	Description of the Random-Access Memory (RAM) in the reference hardware. Performance measures should be associated to clear units (e.g. Go). The description of the RAM may not be limited to its capacity in octets or bits.
Data storage	Description of the data storage in the performance hardware. Performance measures should be associated to clear units (e.g. Go). The description of the data storage may not be limited to its capacity in octets or bits.
Other hardware characteristics	Hardware characteristics other than CPU, RAM, and data storage. Can be used for High Performance Computing (HPC).

To secure execution

=> Verification

Integration and Verification : software

Characteristic of software

- Programming language
- Tool characteristic
- Operating system
- Compiler

To secure execution

=> Verification

Attribute	Description
Language name	Programming language of the simulation model. Can be used for noncompiled or compiled simulation models. The multiplicity of the parent element ("Language") permits to handle simulation models with multiple simulation languages.
File format name	File format of the simulation model. A given language can lead to different file formats. For example, a simulation model in Simulink can be saved as an SLX, an FMU, or a DLL. Thus, the file format can show whether a simulation model is compiled or not. The multiplicity of the parent element ("File format") permits to handle simulation models split in multiple files
Tool name	Name of the simulation tool required to run the simulation. Permits the external management of the software licenses constraining the use of the simulation tools and compilers. The multiplicity of the parent element ("Required simulation tool") permits to handle simulations where multiple tools are required, such as in a cosimulation (e.g. with Simulink and Amesim).
Tool version	Version of the simulation tool required. If no version is specified, it is assumed that the simulation can run with any version of the simulation tool.
Operating system name	Name of the operating system required to run the simulation. Permits the external management of the software licenses constraining the use of the operating systems. The multiplicity of the parent element ('Required operating system") permits to handle simulations requiring multiple computers with different operating systems (e.g. with Windows on one computer and Debian on another).
Operating system version	Version of the operating system required. . If no version is specified, it is assumed that the simulation can run with any version of the operating system.
Compiler name	Name of the compiler required to run the simulation. A compiler should be specified for simulation models programmed in languages like C, independently from any simulation tool. A compiler should also be specified in some simulation tools like Amesim, in particular in case of cosimulation with other tools. The licenses constraining the use of the compilers can be externally managed. The multiplicity of the parent element ("Required compiler") permits to handle simulations where multiple compilers are required.
Compiler version	Version of the compiler required. If no version is specified, it is assumed that the simulation can run with any version of the tool or compiler.

Validation/Qualification

Evidence of quality

- **Proof of evaluation**
- **Scale of fidelity**

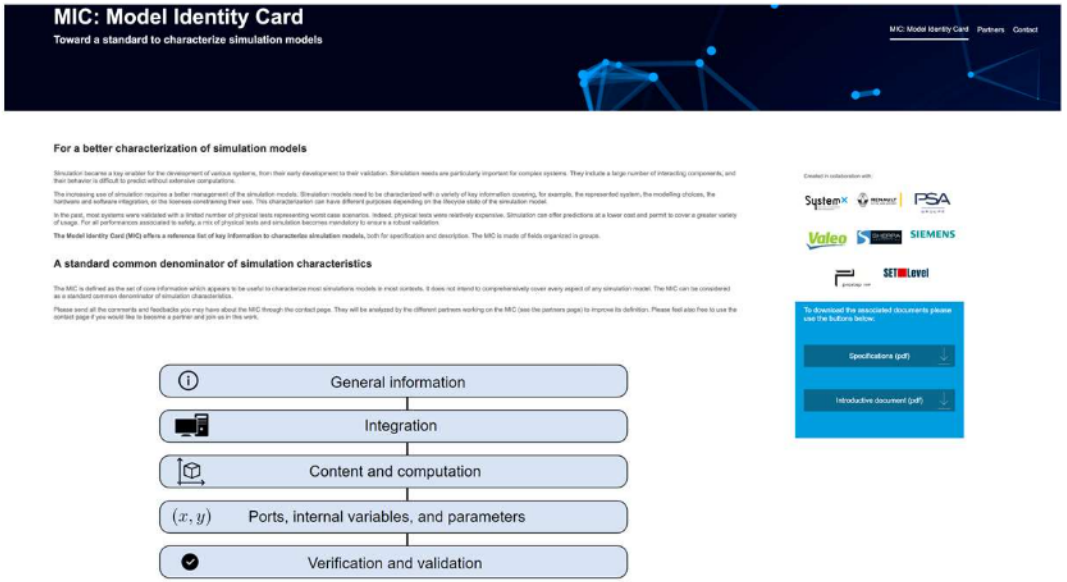
Attribute	Description
Method	The verification and validation of a simulation model can be based on multiple tests. Each test follows a certain method. The method is described here with a key word.
Documentation	Identifies a document describing the test applied to the simulation model (test procedure, test results, …).
Name	It is sometimes decided to indicate, after the verification and validation, how good the simulation model is. It can follow different metrics, and the score may be arbitrary. This attribute permits to identify a metric by its name.
Score	Score obtained by the simulation model for a given metric.

To tag fidelity of models

=> Validation

Publication of a white paper

- **MIC specification document available at :**
  - https://mic.irt-systemx.fr/#/mic



Conclusion

- **MIC is a simple and easy concept and become a reality now**
- **Can be implemented in any tool**
- **In the future, it could become a standard**

Time for Q&A