Three Reasons why Document-based SE (usually) works better than (most of) MBSE

Jean-Philippe LERAT – Sodius
1 rue André Gide – 75015 – Paris
jplerat@sodius.com

Why this provocative subject ?

After more than a decade at INCOSE, it is time to assess the evolution of SE community.

- Did the 1985 figures of the Standish Group really evolve?
  - 87% of projects are not « successes »… Did it change?
- Is the SE practice better now than 10/15 years ago?
  - No Sure… (even if RM is now everywhere)
- Is the SE practice better now than 40 years ago?
  - Much less certain…

Is MBSE really practiced ?

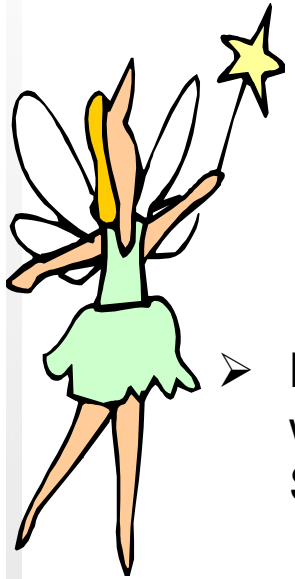
It is always difficult to figure out how much MBSE is practiced within companies.

The usual statement is that « we all do MBSE ».

By experience, this assertion must be challenged, because many companies confuse two different notions:

- Model-Based engineering activities, which is widely spread in companies for most technical disciplines (CAD/CAM, EDA, Software, planning, manufacturing…)
- MBSE, which drives the engineering from a repository of architectural data.

Why focusing on MBSE ?



- Year after year, new magic wonders come to the market, bringing redemption to Systems Engineers
  - Product Data Management
  - OO methods – Formal methods - …
  - UML 1.x - UML 2.x
  - SysML
  - …
- Many projects claimed a drastic improvement due to these magic wonders, grouping them under the generic name of Model-Based Systems Engineering.
- Many companies are currently implementing MBSE to get the same improvement.

But are we sure that MBSE is a magic wonder ?
Here is the challenge!

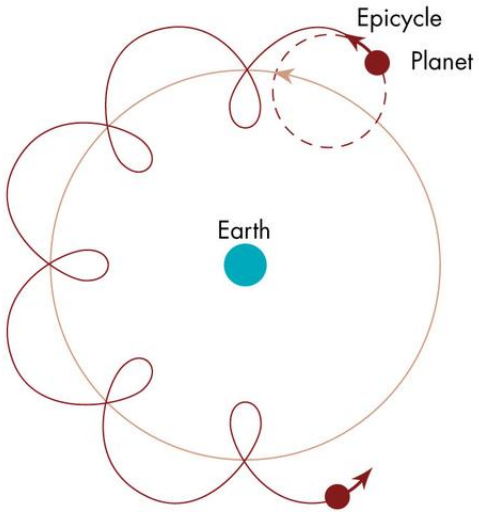
MBSE as a prediction tool

Regardless of its implementation (Matlab – Statemate – UML – SysML – CATIA – …), MBSE is a kind of prediction technique to look into the future and obtain predictive information.

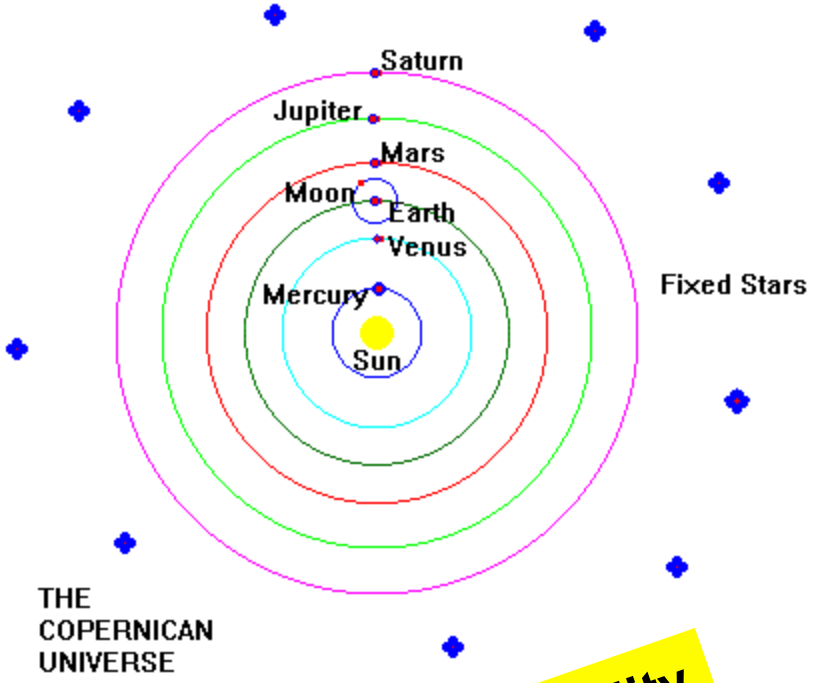
So, the most important quality of a MDSE approach is its ability to provide accurate predictions, because all subsequents trade-off will be based on it, as well as the management of the project.

In which extent does MBSE provide good predictions?

Prediction in models: what does it mean?



The Ptolemy model



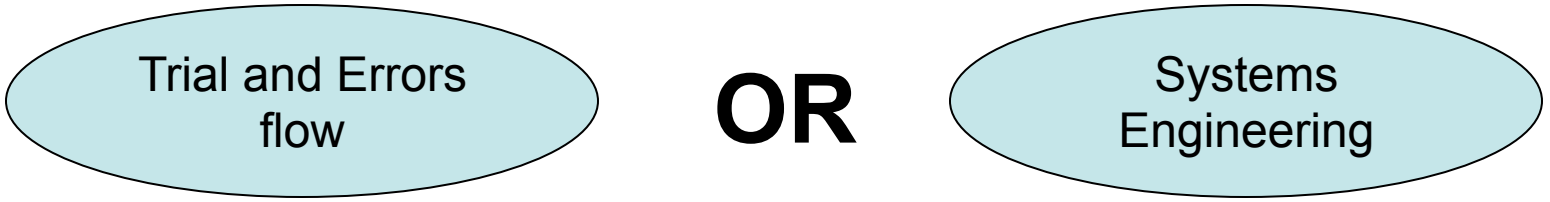
More accurate

Closer from reality

Until Kepler, the Ptolemy model worked better than Copernicus. Until Galileo and the optical discovery of the phases of Venus, both models are only speculative.

MBSE in Systems Engineering : the input

The two ways to design a system :



Prerequisite Knowledge for Systems Engineering:

The Mission
The Environment
The consequences of each technical decision

Until these three domains are covered with sufficient accuracy, Systems Engineering is not an option and the process will return to « Trial and Errors ».

MBSE in Systems Engineering : the output

The output of Systems Engineering is an architecture, which gathers a set of ***principles*** :

- Principles to perceive the environment
- Principles to react to the environment
- Principles of systems organization

As principles are abstract matters, it is crucial to bring them to some « concrete state », so that they can be shared, investigated, challenged and exchanged between the different people within the project.

Documents or Models are the usual vehicules used to bring the information between the different project actors.

Three usual problems of MBSE

By my experience, system-level modeling usually ignores some essential pitfalls:

- MBSE usually focuses more on products-level than on systems with dangerous OO-methods.
- MBSE does not recognize the hierarchy of systems, which generates complexity rather than reducing it.
- MBSE does not support the human and organizational dimension of project management and fosters a wrong idea of collaboration.

MBSE may accelerate the creation of problems, but also may hide them within the models.

Problem #1 : Product vs system modeling

Most MBSE approaches are currently based on Object-Oriented (OO) methods, directly derived from software processes.

However, what works fine for software may hide some tricks that may drive to severe misuse :

- OO-Methods look for objects, while Systems are not at the level of tangible things.
- OO-Methods imagine isolated objects, while they are usually not.
- OO-Methods do not take into account the effect of technical decisions on the environment, leading to the breach of one of the most important source of derived requirements

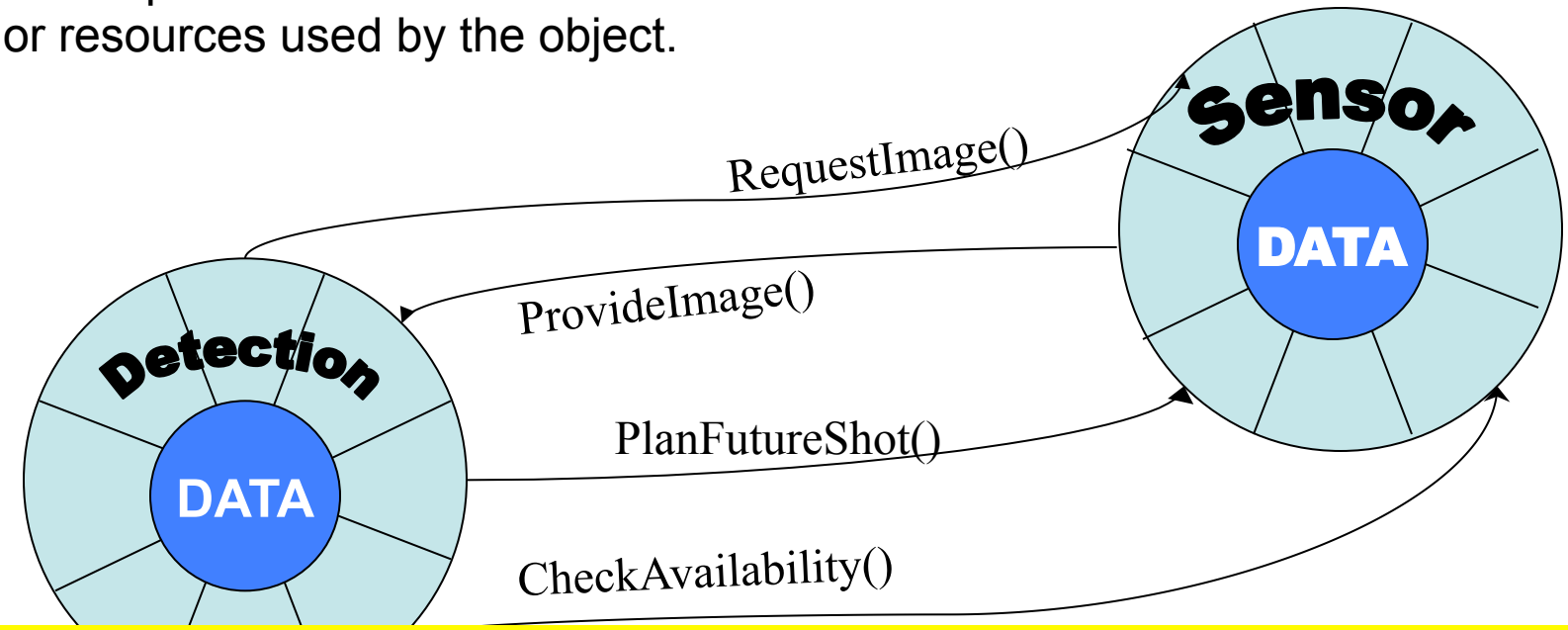
Problem #1.1 : where are the objects ?

- Objects are the basis of most of MBSE processes, like SysML.
- However, when starting to design a system, no physical organization exists.
- One of the main pitfalls of these methods is to « force » the specification of objects, far before they reveal being relevant or not.

In the first days of a project, the physical structure can be frozen through the objects of the model and cannot be challenged later.

Problem #1.2: Isolated object?

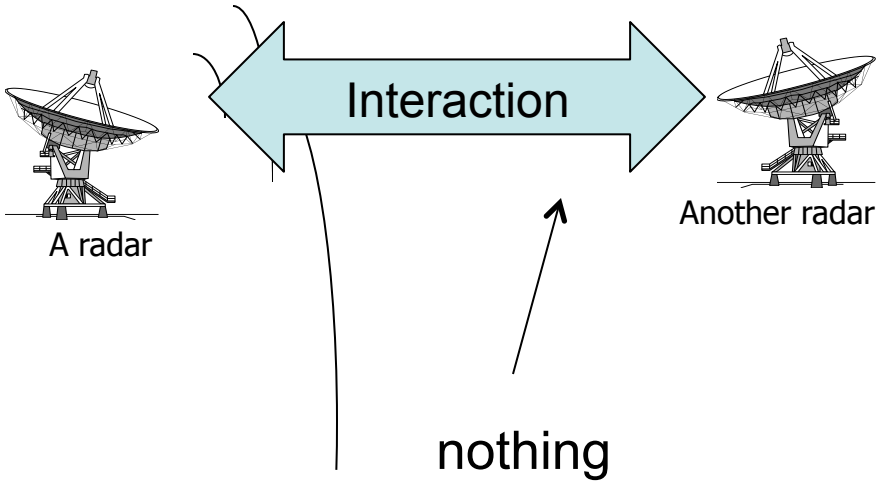
- Objects are extremely well defined and easy to manipulate through their interfaces.
- One of the most interesting points of OO-methods consists in hiding everything beyond the interface : the way the functions operate is not the business of the user, as well as the persistance methods used to store the information or resources used by the object.



At the system level, information about the implementation of functions and about the storage of information & resources are critical

Problem #1.3: The environment feedback

- One of the most disturbing problems in Systems Engineering is to take into account the impact of the system on its environment during the design phase.



Interaction at the system level may not be addressed, since objects are usually not considered to be effected by themselves.

Problem #2 : Generating complexity

The of MBSE is to the help of models to reduce complexity.

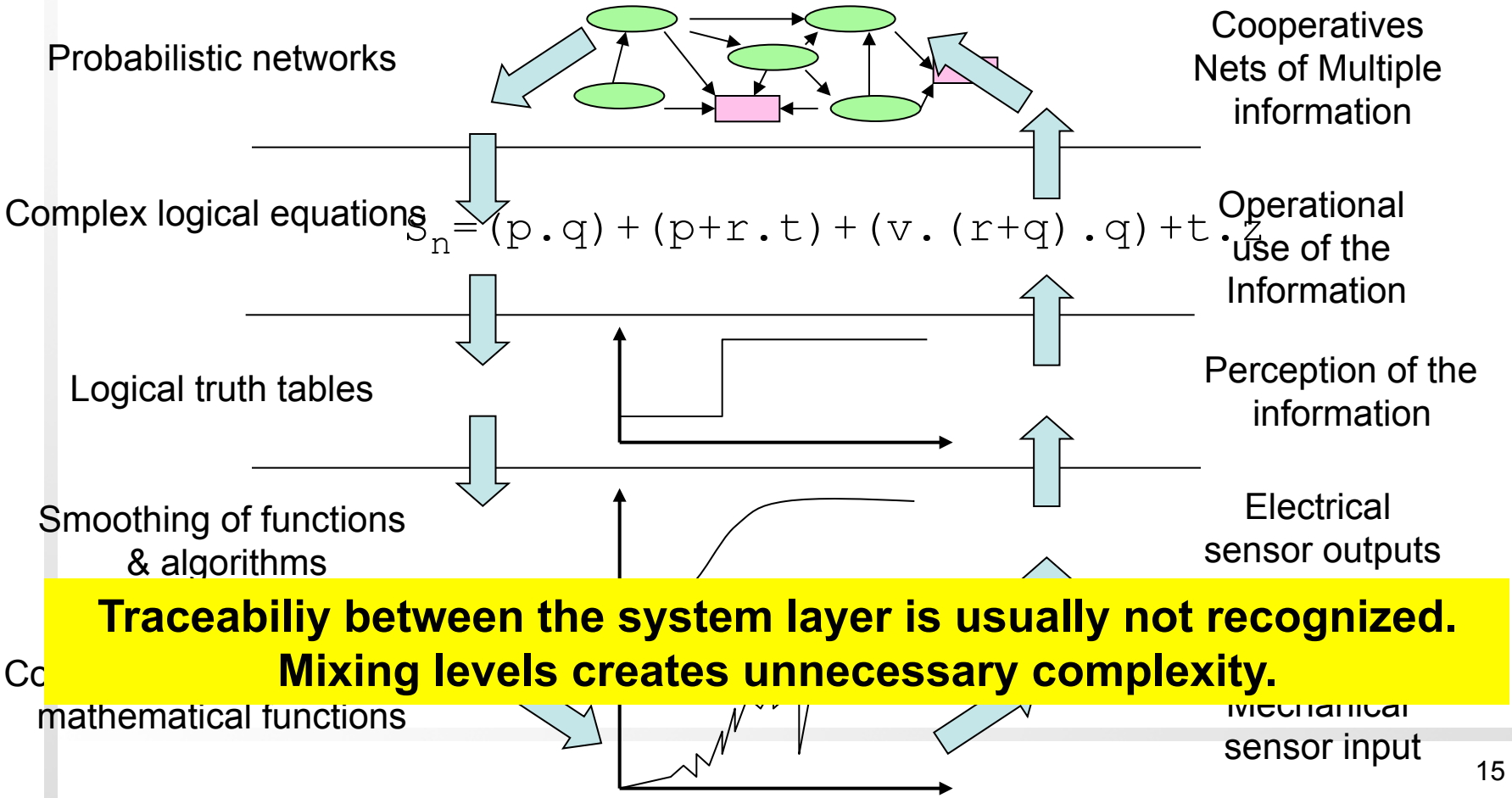
However the contrary may come from bad modeling: the complexity unnecessarily increases through the model for several reasons :

- The « hierarchy of systems » is not considered
- Statemachines are defined without states
- Functions of subsystems are confused with subfunctions of the system.

MBSE accelerates the creation of problems, but also hide them within the models.

Problem #2.1: The hierarchy of systems

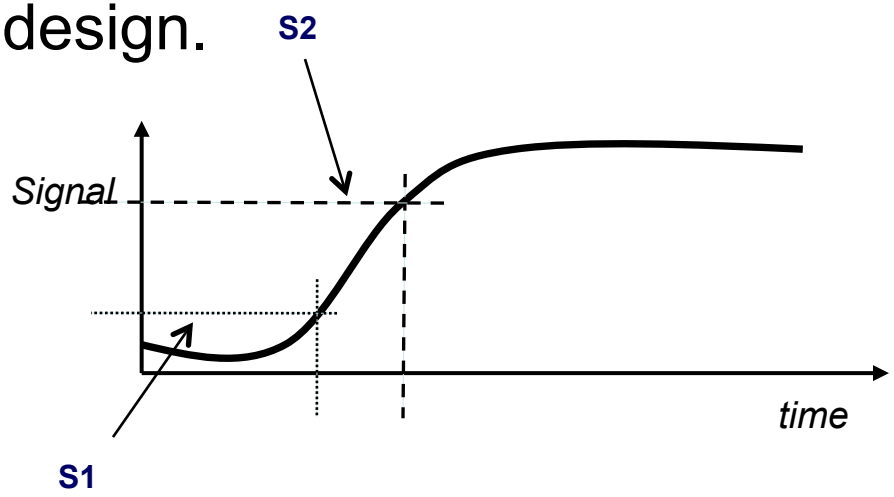
Large systems are usually not composed of a single block, but rather of a stack of systems. However, this stack embeds interactions between the different layers.



Traceabiliy between the system layer is usually not recognized.
Mixing levels creates unnecessary complexity.

Problem #2.2: Statemachines without states

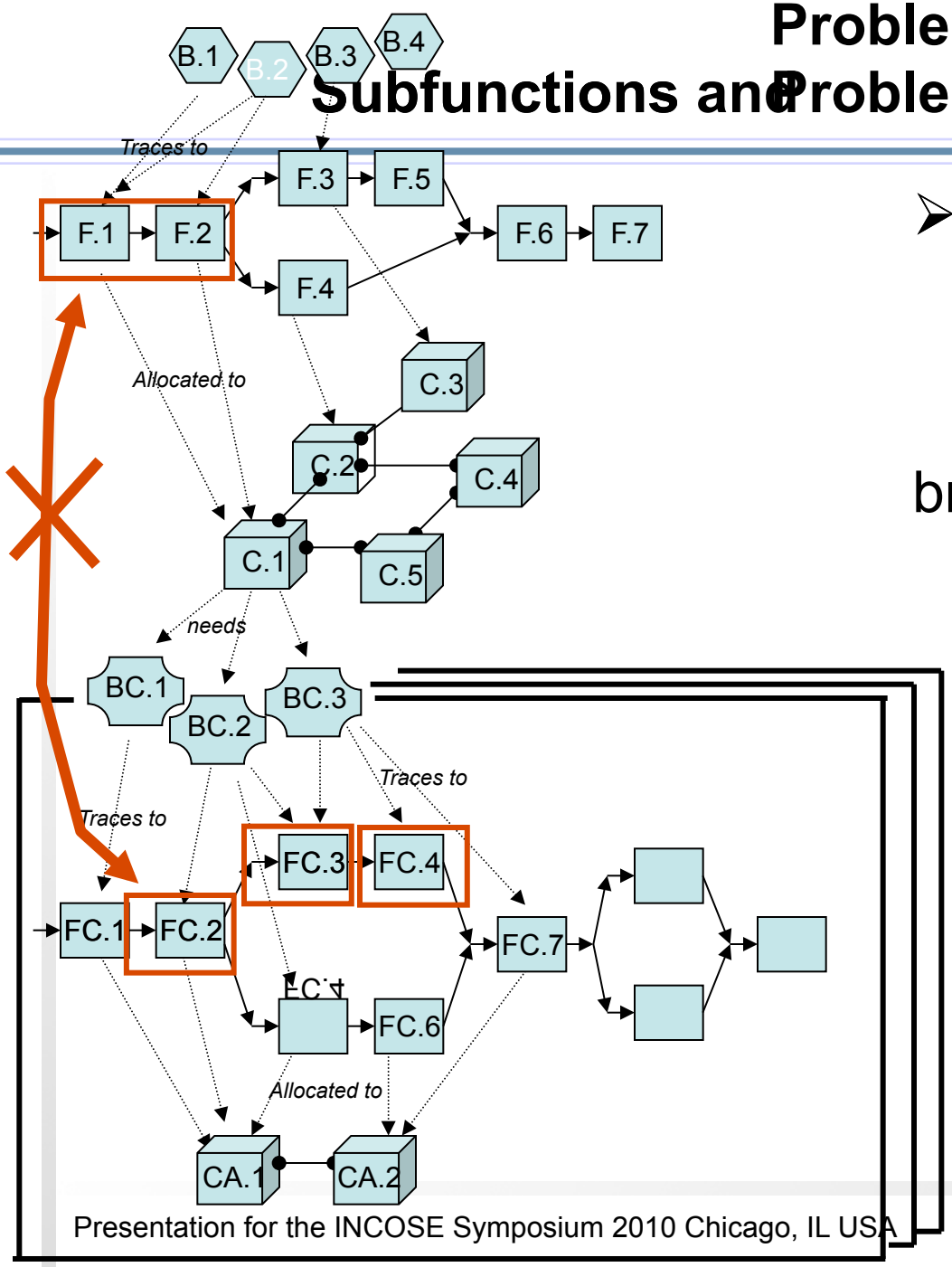
- Most MSBE processes use statemachines as a way to model dynamic characteristics.
- However, state discrimination is one of the pains of system engineers, since discrimination is very difficult.
- Moreover, this situation is complicated by design, since the limit between two states may be part of the system's design.



S1: Better reactivity
S2: Better reliability

Dynamics of systems is usually addressed by MBSE using software based tools, leading to unreliable models.

Problem #2.3: Subfunctions and Problem #2.3:



- Each rupture within
  Each rupt systems
  hierarchy of within the
  systems
  breaches any
  decomposition.
- Most MBSE
  recognize does not
  these
  ruptures, leading to

**The functions -
of a sub- -
systems of the system.**

#3 : Information sharing

- The major issue with MBSE is not on the technical side, but rather on the organizational side.
- Like any other technical standard, MBSE suffers from its variations within the projects within the same companies
- Whatever MBSE standard, MBSE tools are usually not following projects same interpretation, leading to incompatible tools.

Even if MBSE uses only one formalism (like SysML), and even if all the models are available to others through PDM/PLM/ALM applications, Most of these models are stricly incompatible.

Conclusion: Doc-based or MBSE ?

As a starting point, it must be clear MBSE is not only
many arenas, it is a « must-do ».
is many arenas, it is « must-do not only convenient, in accelerator work, which enables the
creation
However, MBSE is a drastic accelerator of work, which
enables
creation of deliverable
the
products

- Document-based processes are slower, but sometime more safe: the time to elaborate the document helps in reviewing its larger number of people.
  Statistically, Document-based processes are slower, but sometime more safe: the time to Statistically, architectura helps in reviewing its content by a larger number of people. reserved to some
  problems

does hardly are more likely detected at this phase which is so
- MBSE
  - MBSE exist to non- is reserved to some model-makers. modeling activities,
  like freezing an architecture made of reversible bad leads
  that,

many of MBSE is difficult, situation. the dissemination is so
like freezing an architecture made, but are rather is from, modeling activities,
detected objects
- Recovery of MBSE failures
is

Conclusion

take advantage

– advantage of MBSE

– Better control of architecture

capability to collaborate

➢ Projects that are not architecture- taking

Bad SE practices are not rescued by MBSE

serious risks. are not architecture-centric are

serious risks of the notion of architecture leads to very

concentrate on a system’s products.

Before implementing any MBSE plan,
make sure that your SE practices are good…

Time to debate…

Please, make sure that you left all your weapons at the entrance of the room…