ENSTA ParisTech

25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

Towards a Holistic Definition of System Engineering: Paradigm, Modeling and Requirements

Hycham Aboutaleb & Bruno Monsuez

hycham.abou-taleb@ensta-paristech.fr

Introduction

- **Ancient Times**:
  Design used to achieve desired results. Complex human-engineered systems were unique (no repeatability).

- **Medieval Times**:
  A method is applied and a process is followed to produce a human made product.

- **Modern Times**:
  Produced systems are market-driven and follow a set of technical and non technical processes to handle complexity.

Introduction

- **Ancient Times**:
  Design used to achieve desired results. Complex human-engineered systems were unique (no repeatability).

- **Medieval Times**:
  A method is applied and a process is followed to produce a human made product.

- **Modern Times**:
  Produced systems are market-driven and follow a set of technical and non technical processes to handle complexity.

Introduction

- **Ancient Times**:
  Design used to achieve desired results. Complex human-engineered systems were unique (no repeatability).

- **Medieval Times**:
  A method is applied and a process is followed to produce a human made product.

- **Modern Times**:
  Produced systems are market-driven and follow a set of technical and non technical processes to handle complexity.

Motivation

- The approach we follow in a project depends on how the results will be used.

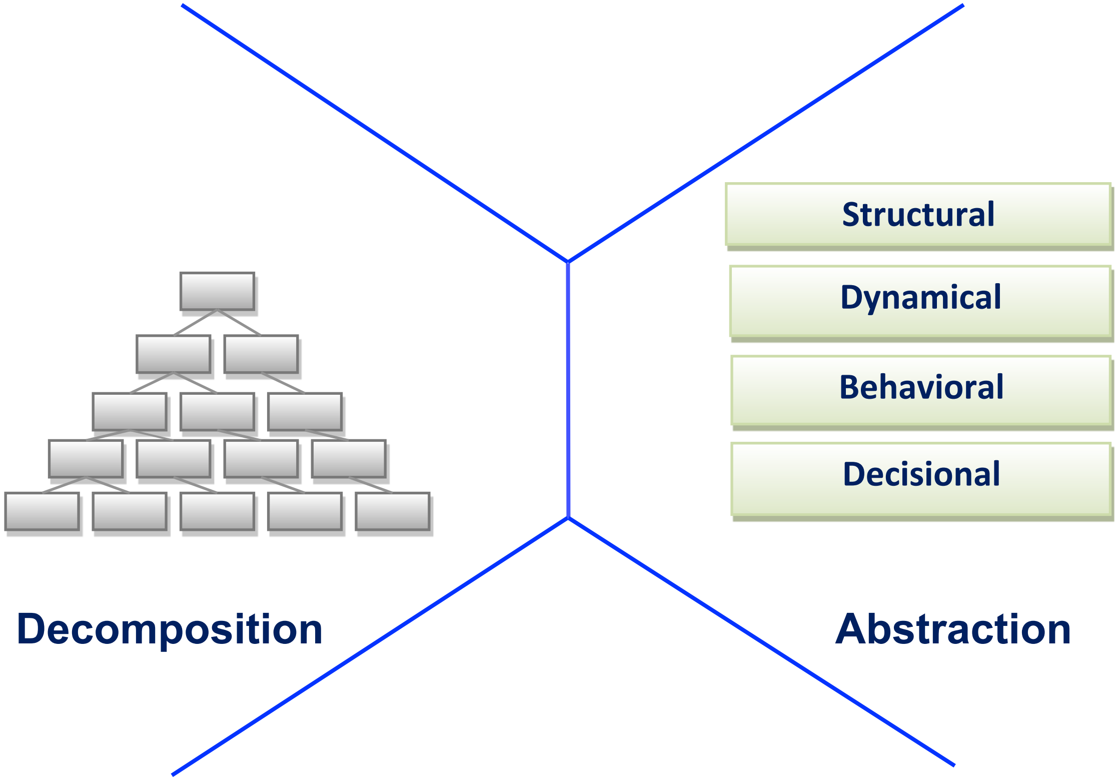
- It is important to have a useful framework to analyze complex systems and study their evolution.

- Main issue: system *real* and *perceived* complexity

Complex Systems Characterization

Decomposition :

Isolate system components for a detailed analysis, given that all information of the context of the analyzed component is regarded.



Structural

Dynamical

Behavioral

Decisional

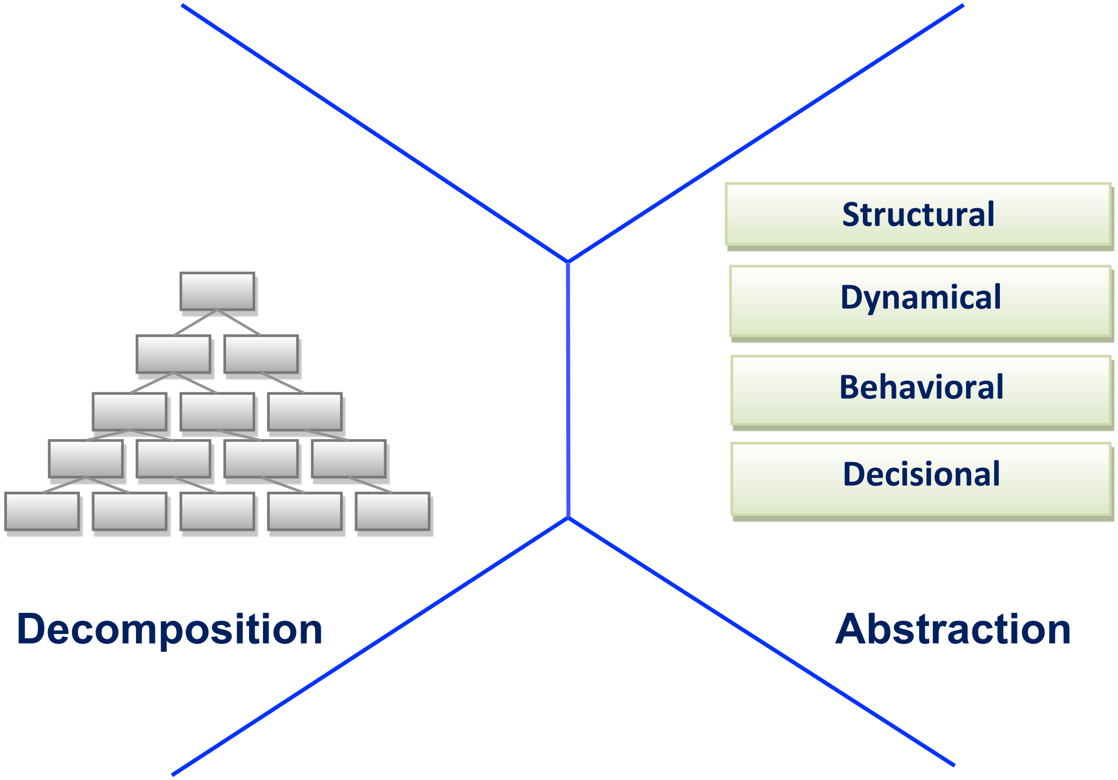
Decomposition

Abstraction

Complex Systems Characterization

Abstraction :

Holistic view of the system that is relative to both the level of detail through decomposition and the type of information captured..



Complex Systems Characterization

Structural layer :

Characterizes the form of the system physical components and their interrelationships. Theses structural aspects include both vertical and horizontal dependencies, such as hierarchy or coupling of the component systems.

Dynamical layer :

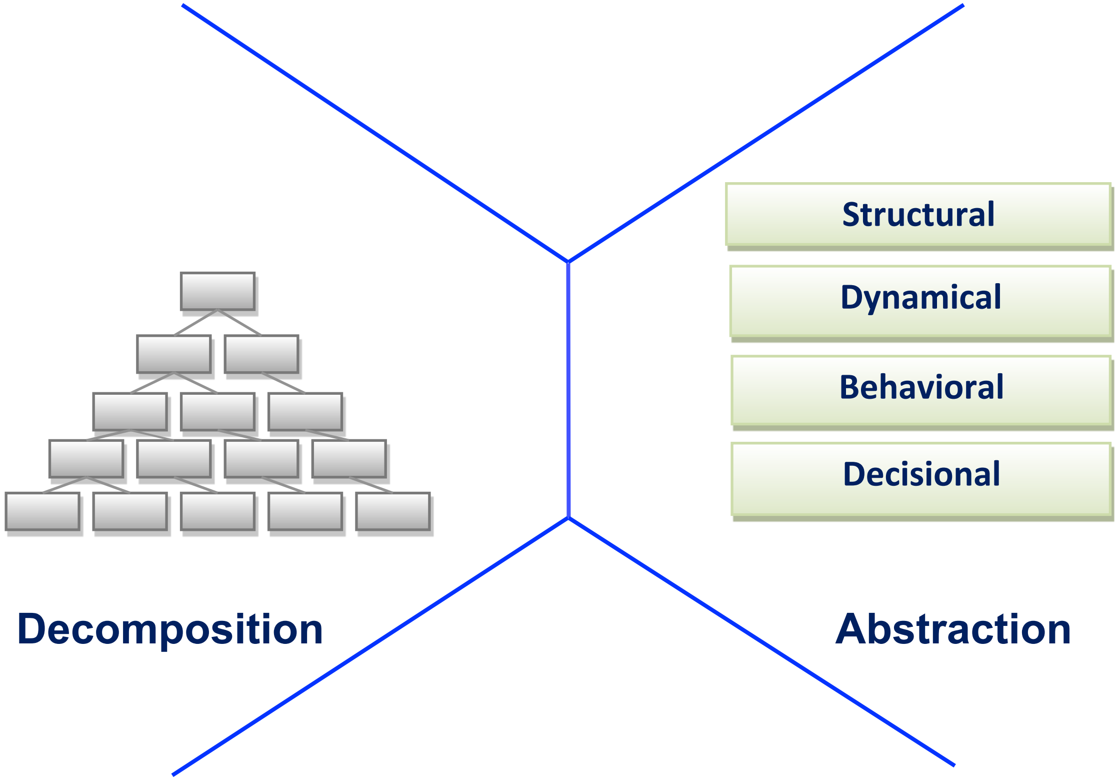
Characterizes changes over time, as well as time-based properties such as adaptability of the system.

Behavioral layer:

Relates to the model of the emergent behaviors resulting from the complex interconnections in order to understand how the systems will perform.

Decisional layer:

Relates to the decisions to be taken, as well as the actions expected to be performed by the system.



Structural

Dynamical

Behavioral

Decisional

Decomposition

Abstraction

Complex Systems Characterization

Structural layer :

Characterizes the form of the system physical components and their interrelationships. Theses structural aspects include both vertical and horizontal dependencies, such as hierarchy or coupling of the component systems.

Dynamical layer :

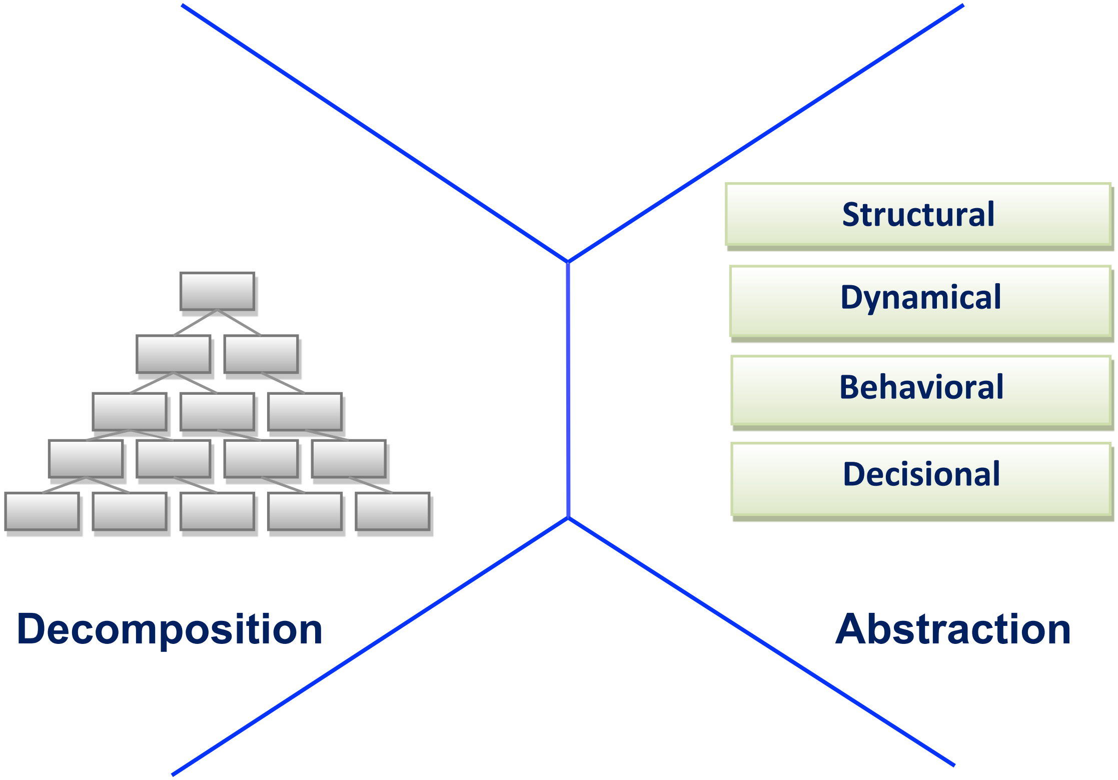
Characterizes changes over time, as well as time-based properties such as adaptability of the system.

Behavioral layer:

Relates to the model of the emergent behaviors resulting from the complex interconnections in order to understand how the systems will perform.

Decisional layer:

Relates to the decisions to be taken, as well as the actions expected to be performed by the system.



Complex Systems Characterization

Structural layer :

Characterizes the form of the system physical components and their interrelationships. Theses structural aspects include both vertical and horizontal dependencies, such as hierarchy or coupling of the component systems.

Dynamical layer :

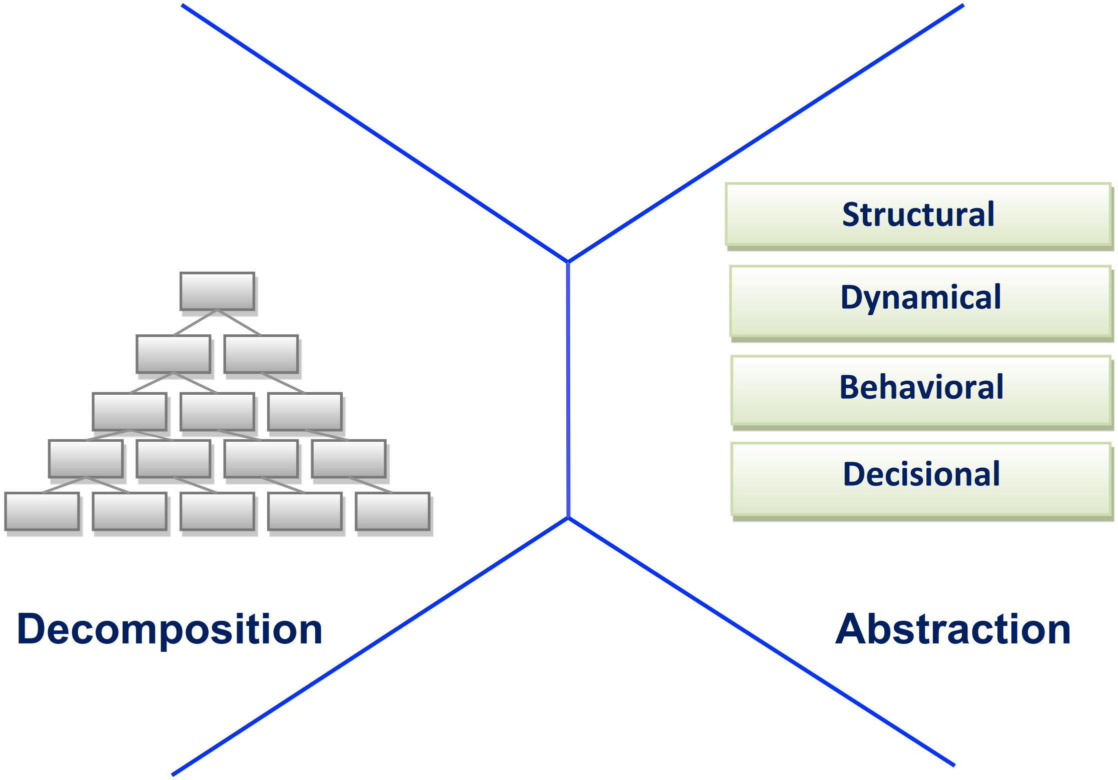
Characterizes changes over time, as well as time-based properties such as adaptability of the system.

Behavioral layer:

Relates to the model of the emergent behaviors resulting from the complex interconnections in order to understand how the systems will perform.

Decisional layer:

Relates to the decisions to be taken, as well as the actions expected to be performed by the system.



Complex Systems Characterization

Structural layer :

Characterizes the form of the system physical components and their interrelationships. Theses structural aspects include both vertical and horizontal dependencies, such as hierarchy or coupling of the component systems.

Dynamical layer :

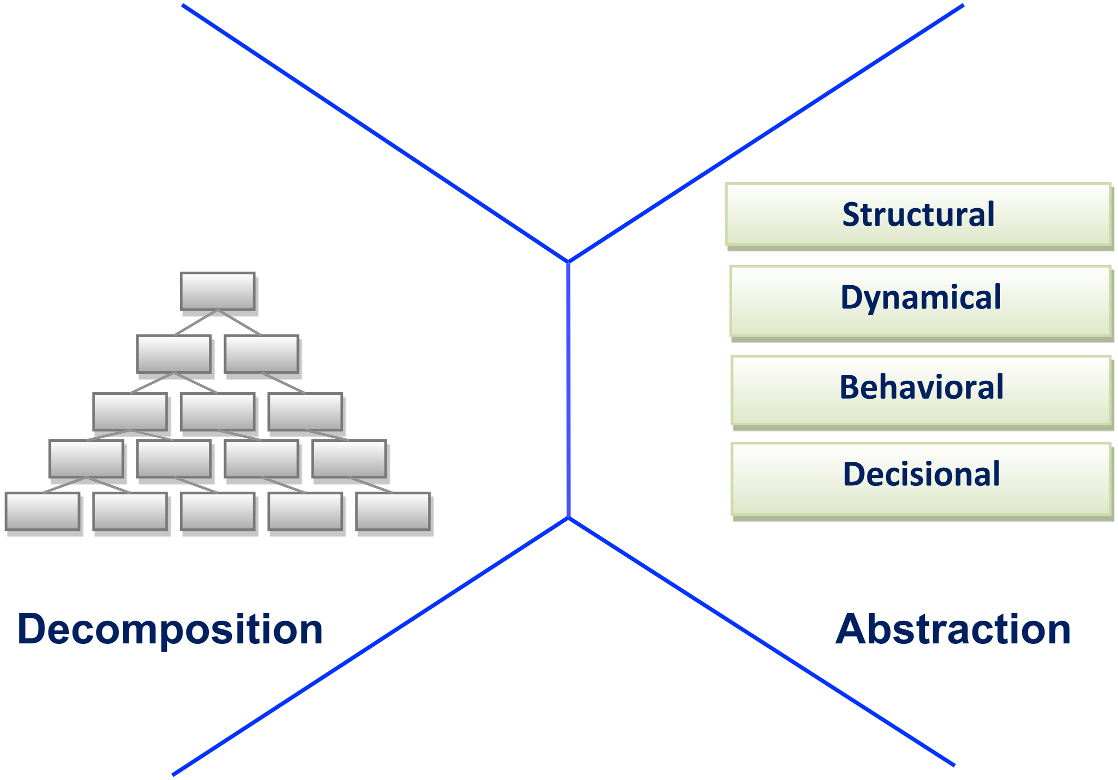
Characterizes changes over time, as well as time-based properties such as adaptability of the system.

Behavioral layer:

Relates to the model of the emergent behaviors resulting from the complex interconnections in order to understand how the systems will perform.

Decisional layer:

Relates to the decisions to be taken, as well as the actions expected to be performed by the system.



Structural

Dynamical

Behavioral

Decisional

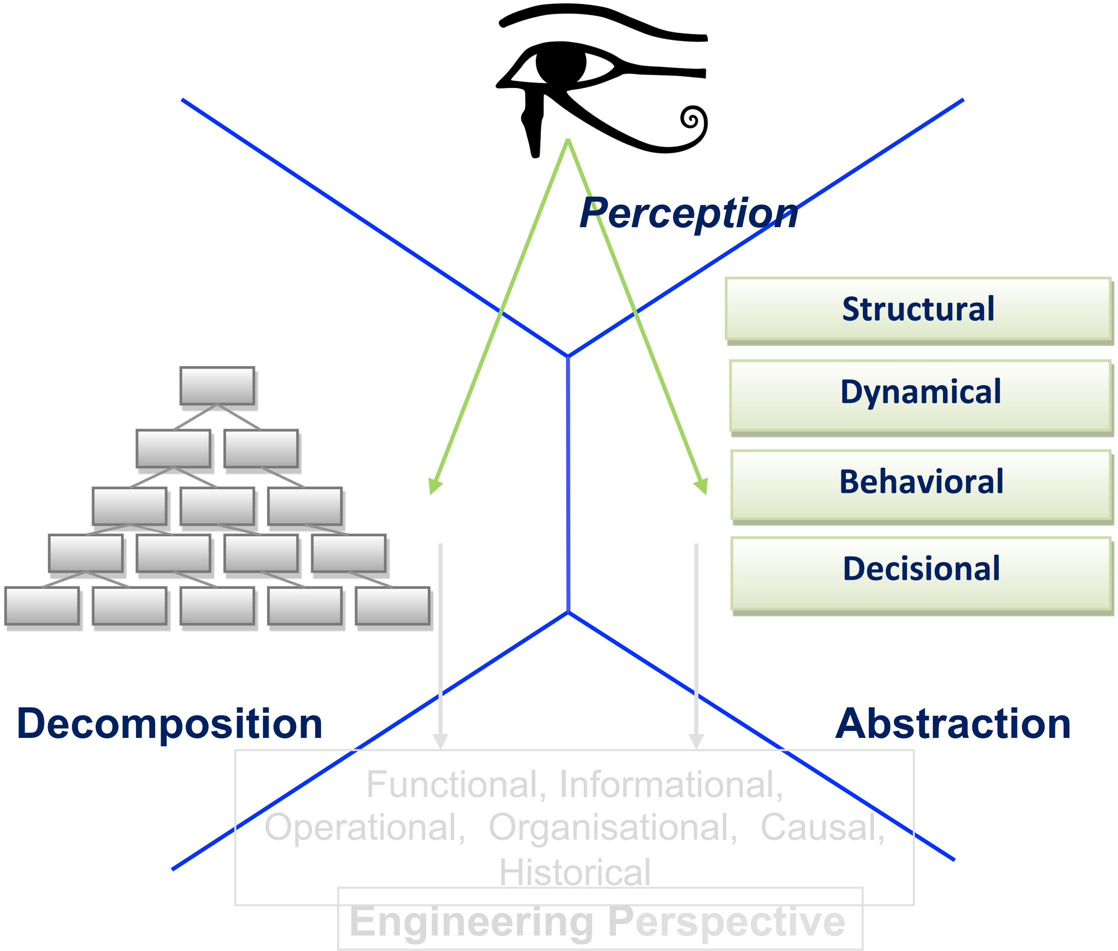
Decomposition

Abstraction

System Engineering Paradigm

Perception:

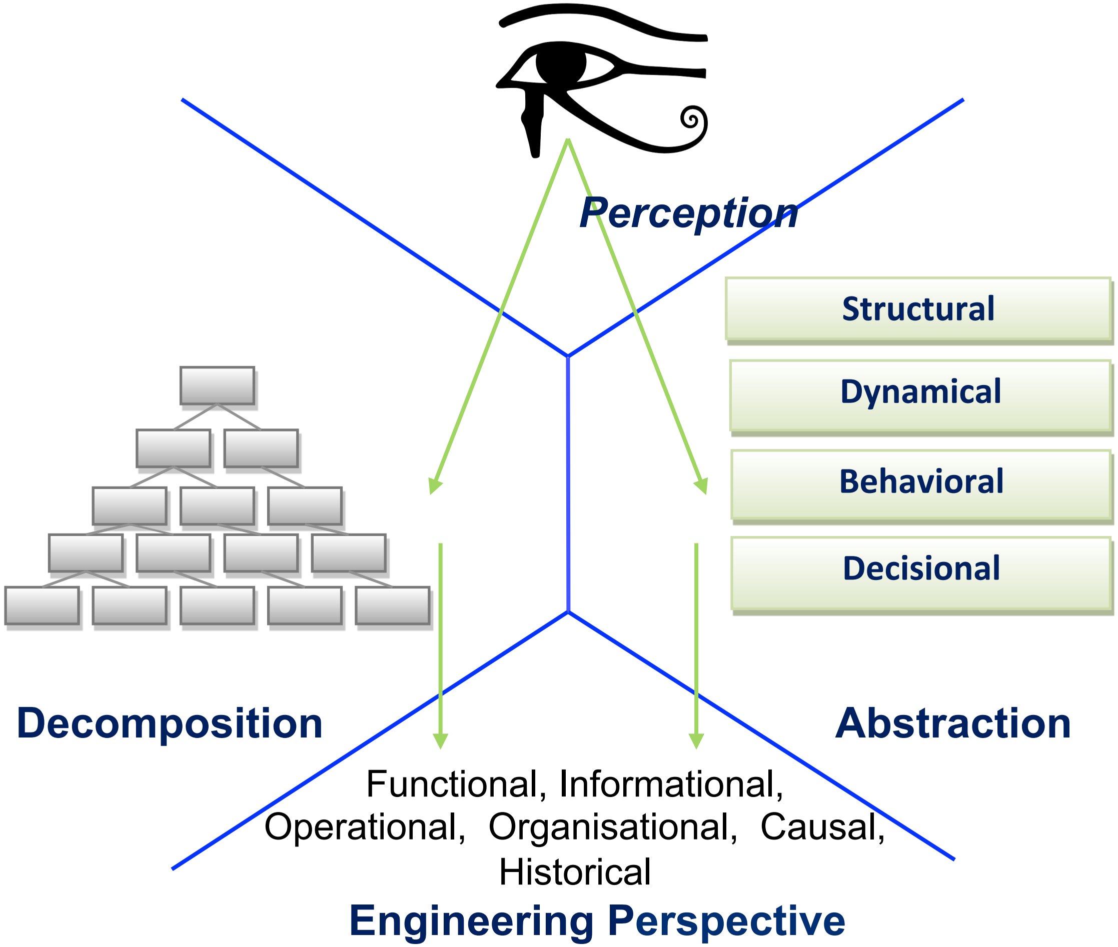
The point of view of each stakeholder or actor that limits or filters the available information, it allows building different models or representations of the problem.



System Engineering Paradigm

Engineering Perspective:

What is needed to be taken into account to design the system. There are technical aspects (which are the technical processes) as well as non-technical ones



Perception

Structural

Dynamical

Behavioral

Decisional

Decomposition

Abstraction

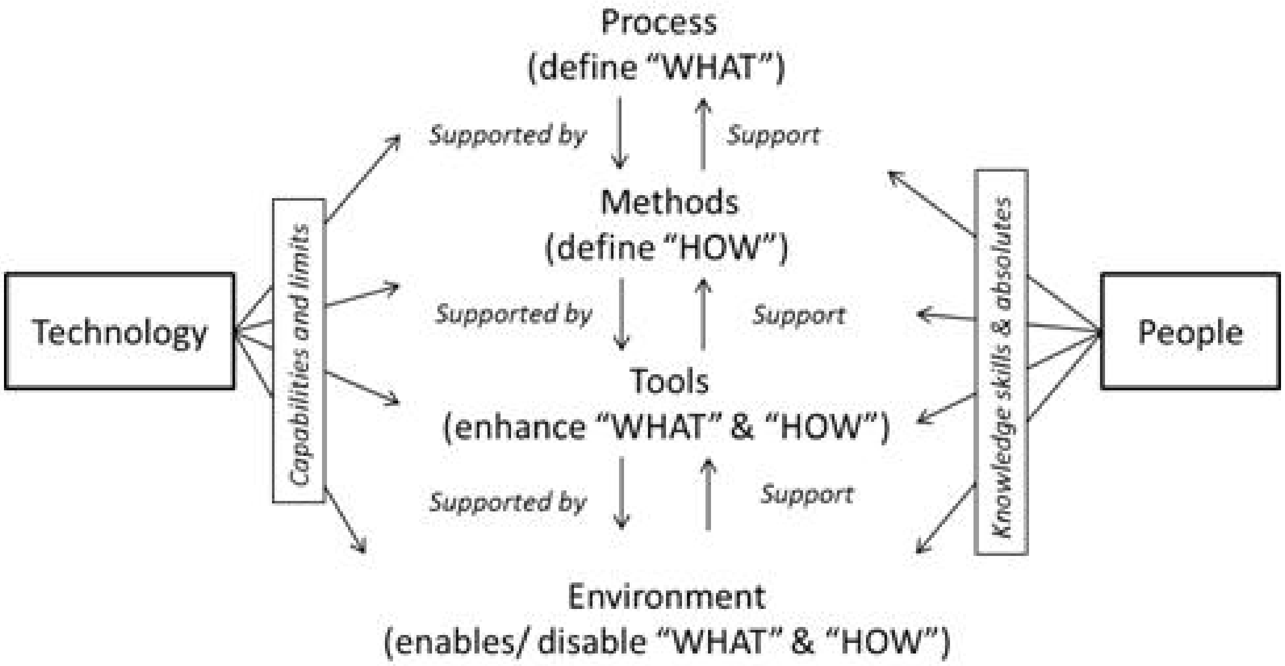
Functional, Informational, Operational, Organisational, Causal, Historical

Engineering Perspective

System Engineering Framework

- Many Model-Based System Engineering (MBSE) methodologies emerged after the introduction of system engineering in the industry.

- To better understand the key features of different methodologies: Need of a framework and a terminology



Martin, James N., *Systems Engineering Guidebook: A Process for Developing Systems and Products*, CRC Press, Inc.: Boca Raton, FL, 1996.

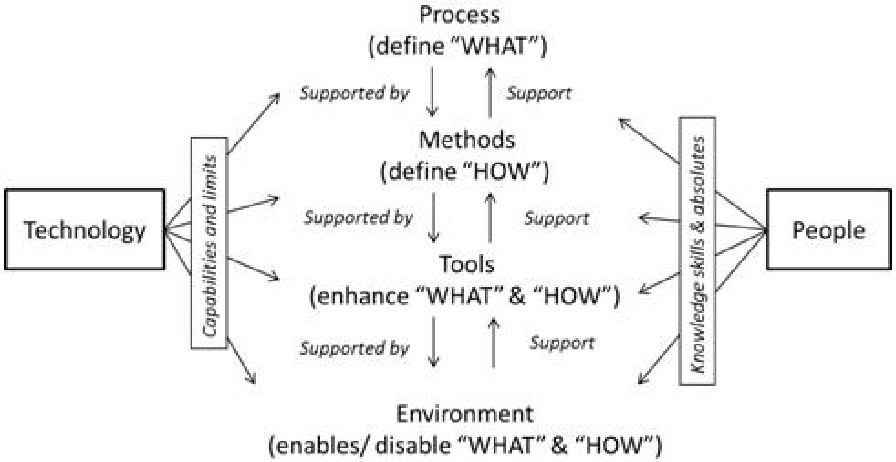
System Engineering Framework

- **Process**: sequence of tasks aiming to achieve a particular objective. Process defines what is to be done without defining how each activity has to be performed.

- **Method**: specifies *how* to perform each task.

- **Tool**: helps to accomplish of *how*. It usually supports a language that helps applying the method.

- **Environment**: consists of external conditions, systems, or factors that have an influence on systems, actors. The purpose of environment is to put in practice the use of tools and methods of a project.



Process
(define "WHAT")
Supported by
Support
Methods
(define "HOW")
Supported by
Support
Tools
(enhance "WHAT" & "HOW")
Supported by
Support
Environment
(enables/ disable "WHAT" & "HOW")
Technology
Capabilities and limits
Knowledge skills & absolutes
People

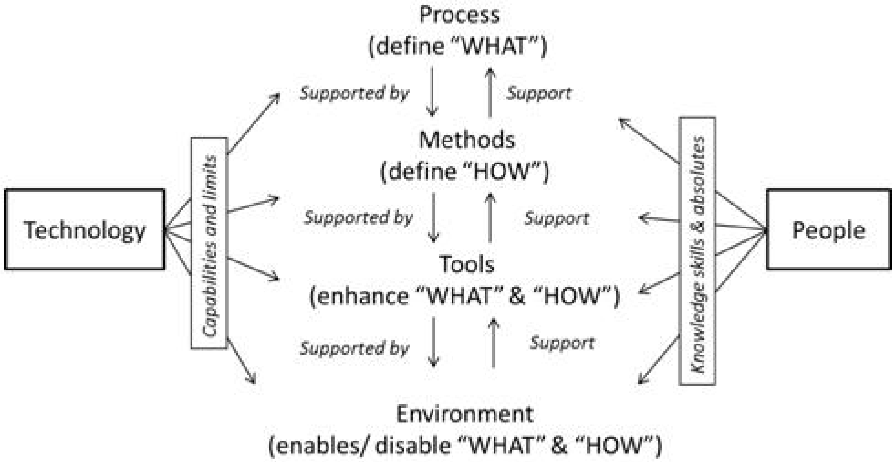
System Engineering Framework

- **Process**: sequence of tasks aiming to achieve a particular objective. Process defines what is to be done without defining how each activity has to be performed.

- **Method***:* specifies *how* to perform each task.

- **Tool**: helps to accomplish of *how*. It usually supports a language that helps applying the method.

- **Environment**: consists of external conditions, systems, or factors that have an influence on systems, actors. The purpose of environment is to put in practice the use of tools and methods of a project.



25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

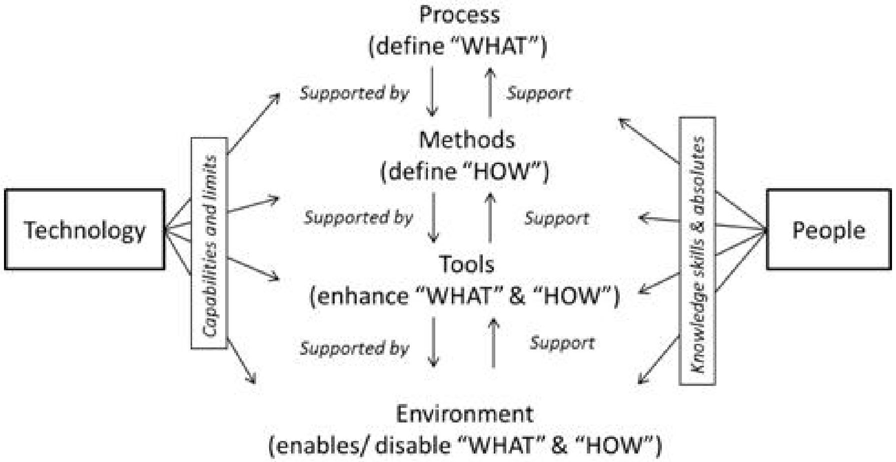
System Engineering Framework

- **Process**: sequence of tasks aiming to achieve a particular objective. Process defines what is to be done without defining how each activity has to be performed.

- **Method**: specifies *how* to perform each task.

- **Tool**: helps to accomplish of *how*. It usually supports a language that helps applying the method.

- **Environment**: consists of external conditions, systems, or factors that have an influence on systems, actors. The purpose of environment is to put in practice the use of tools and methods of a project.



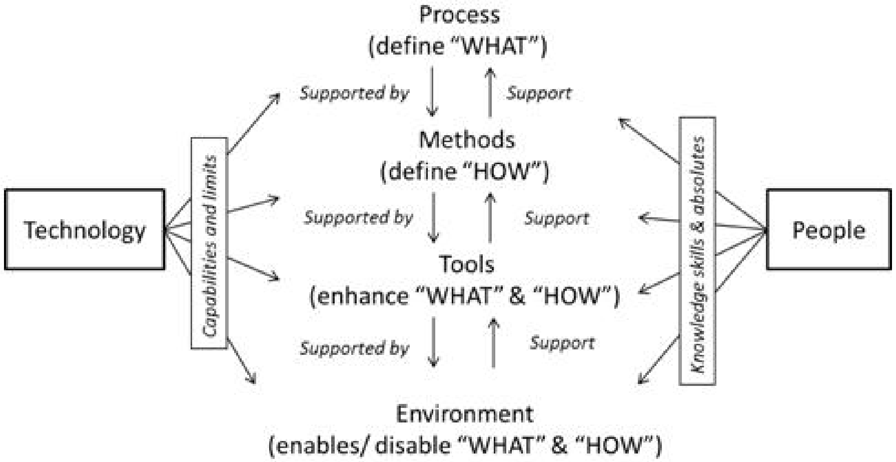
System Engineering Framework

- **Process**: sequence of tasks aiming to achieve a particular objective. Process defines what is to be done without defining how each activity has to be performed.

- **Method**: specifies *how* to perform each task.

- **Tool**: helps to accomplish of *how*. It usually supports a language that helps applying the method.

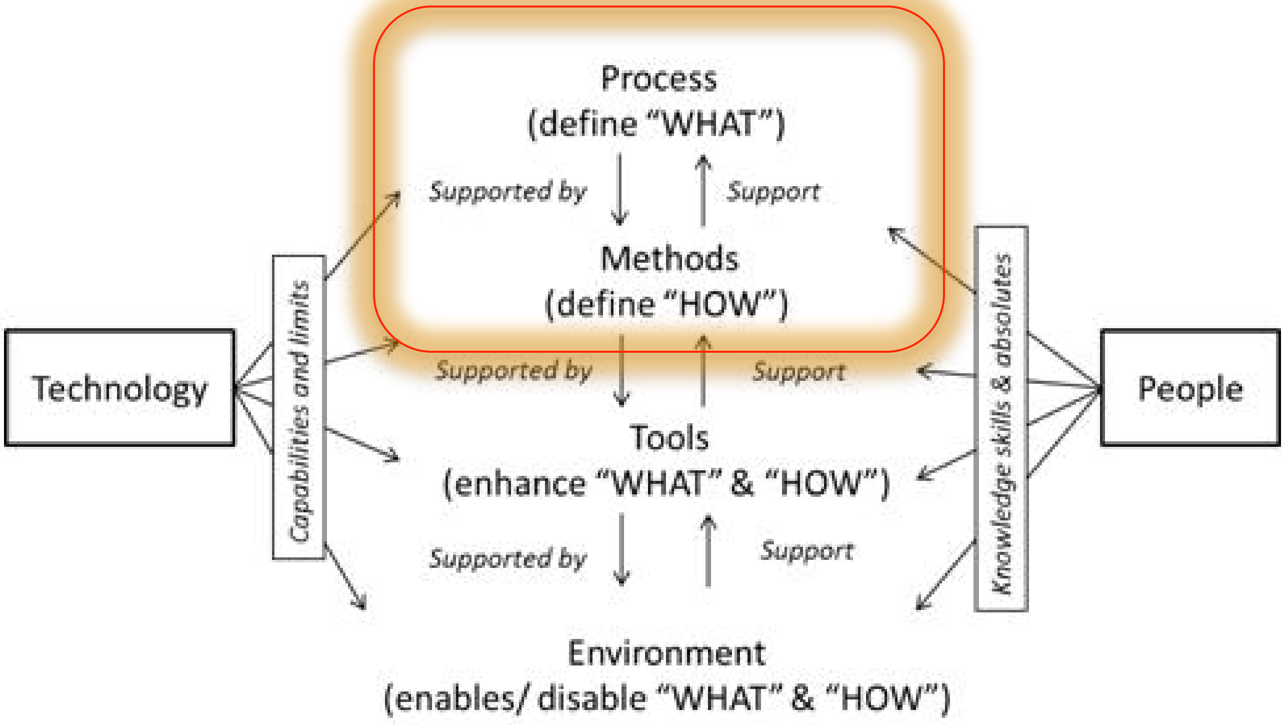
- **Environment**: consists of external conditions, systems, or factors that have an influence on systems, actors. The purpose of environment is to put in practice the use of tools and methods of a project.



25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

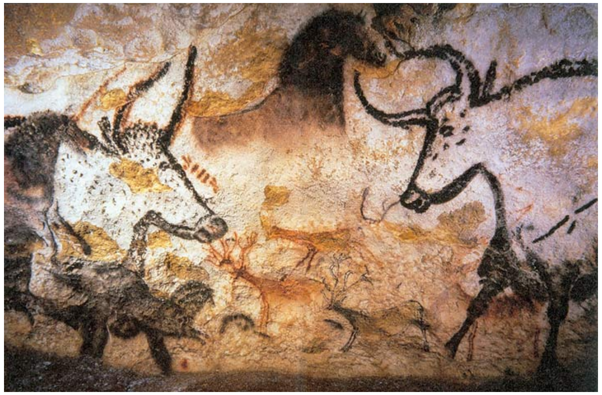
Modelling and Representation

Modelling



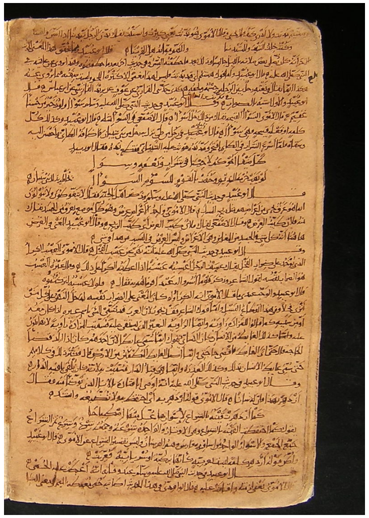
Graphical representations

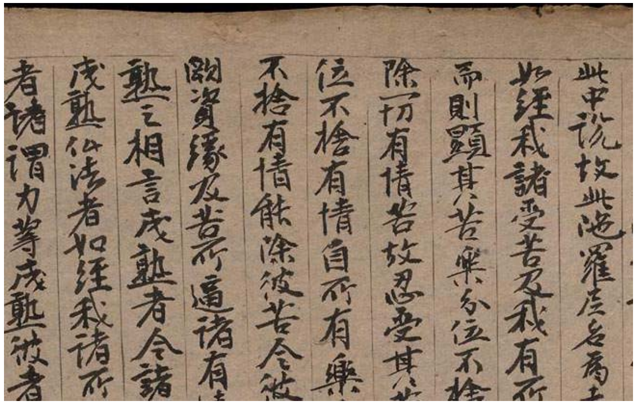
- Convey multiple features of a system simultaneously.
- Expressions correspond to the components of a model describing the system.
- Preserve explicitly the information about the topological and geometric relations among the elements of the system.



Natural Language

- Natural language is used everyday.
- Limitations:
  - not an effective way to describe a system behavior or structure
  - time consuming and needs huge efforts to keep track of the versions
  - might be ambiguous





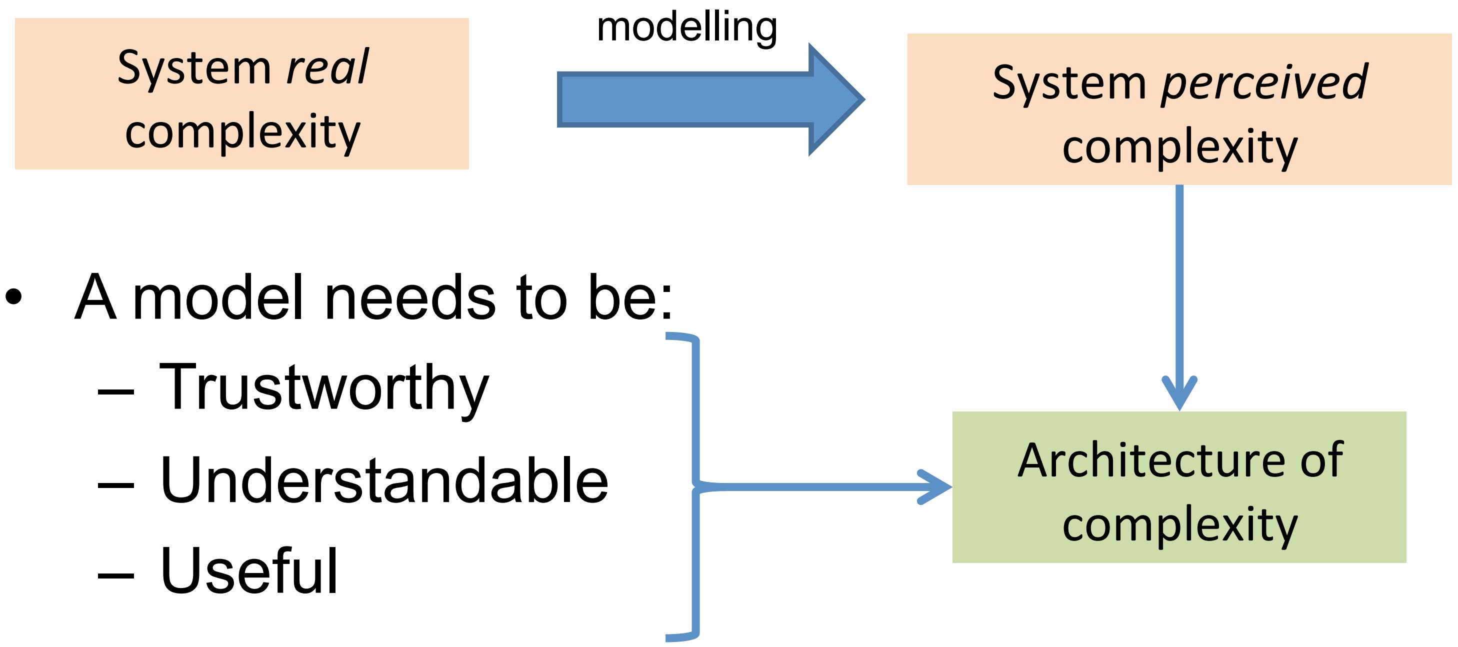
此中說故此陀羅尼名為
如經我諸受苦及我有所
而則顯其苦樂分位不捨
除一切有情苦故忍受其
位不捨有情自所有樂
不捨有情能除彼苦令彼
謂資緣及苦所逼諸有
熟之相言成熟者令諸
成熟佛法者如經我諸所
者諸謂力等成熟彼者

Modelling Needs

- Why Modelling?
  - to help designing the system of interest
  - to address all the aspects of the system of interest
  - to share the knowledge between people involved in the process

- Modelling Needs:
  - **Trustworthiness**: how close the model is to the reality?
  - **Understandability**: is the model perceived and understood the same way by people?
  - **Usefulness**: does the model help to get the desired results?

System Complexity and Hierarchy



System *real* complexity

modelling

System *perceived* complexity

Architecture of complexity

- A model needs to be:
  - Trustworthy
  - Understandable
  - Useful

System Complexity and Hierarchy

- Two ways how to organize hierarchically a set:
  - Grouping : group items based on similar properties or characteristics.
  - Encapsulation: encapsulate many elements within a single element of a higher level.

- A model should be the result of a simplification strategy consisting in:
  - Conceptual chunking: captures the essence of the problem-at-hand and reduces the complexity
  - Segmentation: decomposition of a complex system into smaller parts that can be studied in isolation

Modelling Requirements

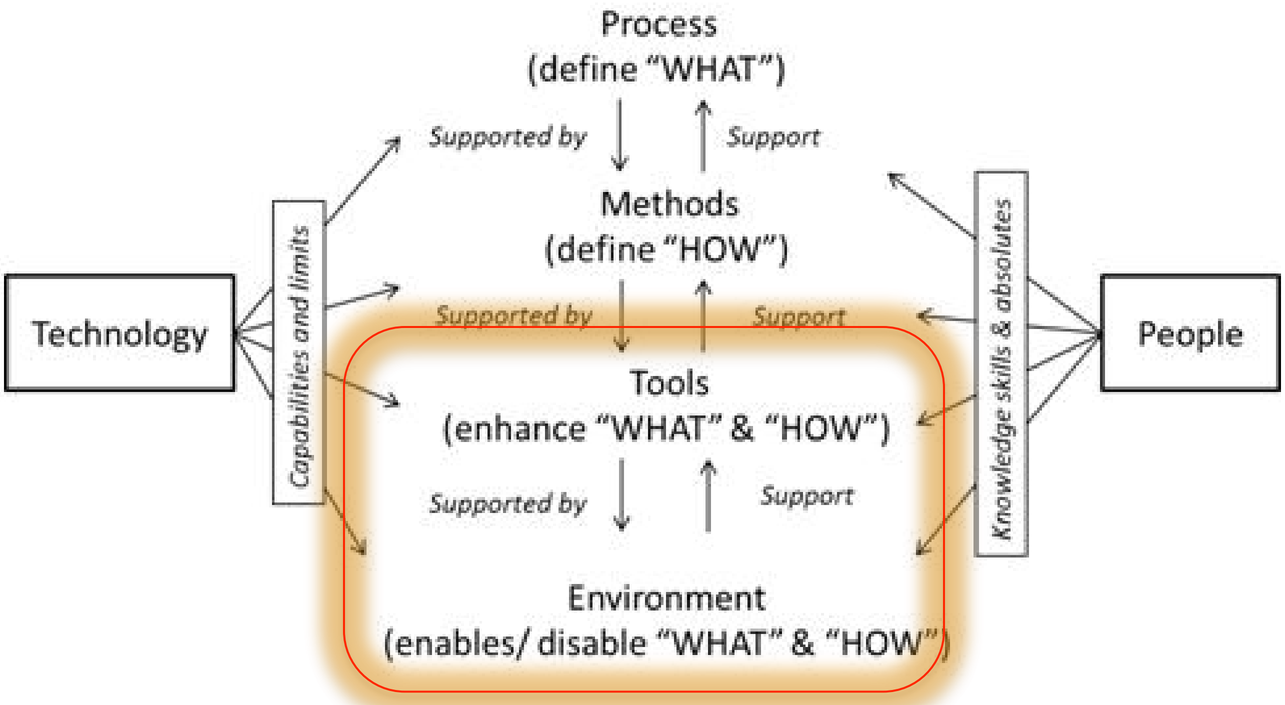
The modeling approach must be powerful enough to express all relevant properties of a system. It should meet the following requirements:

- Analogy
- Utility
- Stability
- Projection
- Modularity
- Compositionality
- Abstraction
- Refinement

Modelling Requirements

Requirement	Description
Analogy	A model shall correspond to conceptual models we know or at least avoid conceptual clash.
Utility	A model shall serve a useful well-defined goal.
Stability	A model shall be usable uniformly in many different contexts without any modification.
Projection	A model shall support diverse and integrated views on the system under development.
Modularity	A model shall have a hierarchical organization of its composing elements.
Compositionality	A model shall allow deducing the properties of a system from the properties of its subsystems.
Abstraction	A model shall enable capturing properties of a system that are needed to understand one of its aspects without paying attention to the details or the other aspects of the system.
Refinement	A model shall start with high-granular descriptions and allow to incrementally refining them into more detailed ones.

System Modelling Language



Modelling Language

System Modelling Language

- To model a complex system, it is necessary to address all the aspects.
- Usually, models focus on structure and behavior. It is likely that both model categories need to be combined and used at different points in the system development process.
- Based on literature and industrial experience, a set of modelling language requirements have been identified

Modelling Language Requirements

1. Simplicity
2. Visual, graphical language
3. Visual Flexibility
4. Semantic Flexibility
5. Domain and User Specific
6. Semantic Preciseness
7. Customizability
8. Compositionality
9. Multiple Views
10. Integrability
11. Extensibility
12. Textual Property
13. Versioning
14. Completeness Checking Property

Modelling Language Requirements

Requirement	Description
Simplicity	Modeling language shall be simple, with few basic concepts.
Graphical Language	Modeling language shall allow a visual, graphical depiction of the model, giving both an overview of the whole system and details about its parts.
Visual Flexibility	Modeling language shall allow modifying the graphical properties of the elements represented in the model if necessary
Semantic Flexibility	Modeling language semantics shall be able to evolve.
Domain/User Specificity	Modeling language shall be domain/user specific.
Semantic Preciseness	Models generated with language shall be precise and unambiguous
Customizability	The language shall allow contextualization and personalization for both semantic and graphical aspects

Modelling Language Requirements

Requirement	Description
Compositionality	Modeling language shall allow composition of models from parts.
Multiple views	Modeling language shall allow multiple perspectives to coexist and evolve.
Integrability	Different viewpoints shall be able to be represented together, not as disintegrated model views.
Extensibility	Modeling language shall be extensible so that users can have their own metamodel from an existing one.
Textual property	Modeling language shall have a user interface that integrates textual attributes.
Versioning	Modeling language shall allow capturing history of events and model changes.
Completeness Check Property	Modeling language shall be able to identify incomplete models.

System Complexity Measurement

- State of the art:
  - Large complex systems development projects are not repeatable, making comparative studies hard to perform.
  - There is no widely used system model complexity measure.

- Intuitive reasons that make complexity measuring relevant and worthy:
  - Cost
  - System Development Management
  - Quality Assessment

Complexity Measurement Requirements

Requirement	Description
Order	If the measurement value of an element *A* greater than the measurement value of an element *B,* then the measurement value of the element *B* is less than the measurement value of an element *A.*
Uniqueness	Measuring the same property twice in same conditions shall give the same value: the measurement value of an element *A* cannot be greater (or less) than itself.
Numerical Value	The measurement value shall be a mapping from an observed relation system or element to a numerical relation system.
Meaningfulness	The measurement value shall be understandable and its truth shall not depend on transformations on allowable scales, i.e. if the scale is changed the meaning shall be the same.

Conclusion

- To handle the complexity, it is necessary to architect the model.
- Hierarchy is the most intuitive way to address this issue. Two main types of hierarchy have been defined in that purpose.
- Requirements have been defined for system modelling and have been refined to get modelling language requirements.
- System engineers are still in need of a modelling language that is simple and intuitive to support many tasks in system engineering and architectural reasoning.
- These requirements should help to choose the tools that are to be used in System Engineering, or when designing such tools.
- The complexity measurement is a relevant metric to compare different architectures for the same system.

Questions?

hycham.abou-taleb@ensta-paristech.fr