ENSTA ParisTech

26th annual INCOSE international symposium

Edinburgh, UK

July 18 - 21, 2016

Quantifying System Complexity in Design Phase Using Higraph-Based Models

Hycham Aboutaleb & Bruno Monsuez

hycham.abou-taleb@ensta-paristech.fr

Introduction: Historical Context

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

Complexity

- **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: Historical Context

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

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

Complexity

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

Repeatability

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

Introduction: Historical Context

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

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

Complexity

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

Repeatability

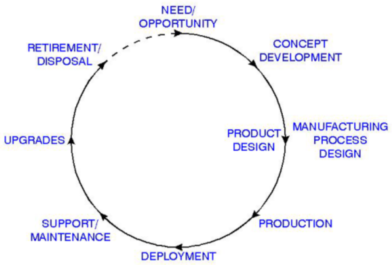
- **Modern Times**:
  Produced systems are market-driven and follow a set of technical and non technical processes to handle complexity.

Heterogeneity

Introduction: Industrial Context

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

- **System Life Cycle**:
  Framework for meeting the stakeholders needs by defining lifecycle stages and using decision gates to determine readiness to move from one stage to the next.



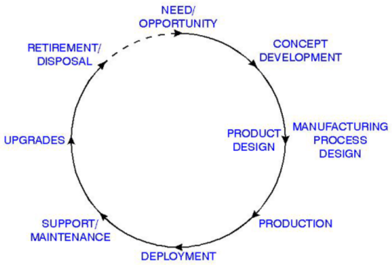
Life Cycle

- **Chaos Report**:
  Many errors are introduced early in design phase. System failures are due to errors in early phases.

- **System Costs**:
  80% of costs are committed early in design phase: when 20% of the actual cost has been accrued, 80% of the total LCC has already been determined.

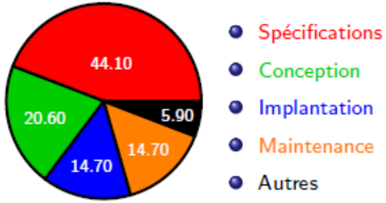
Introduction: Industrial Context

- **System Life Cycle**:
  Framework for meeting the stakeholders needs by defining lifecycle stages and using decision gates to determine readiness to move from one stage to the next.



Life Cycle

- **Chaos Report**:
  Many errors are introduced early in design phase. System failures are due to errors in early phases.

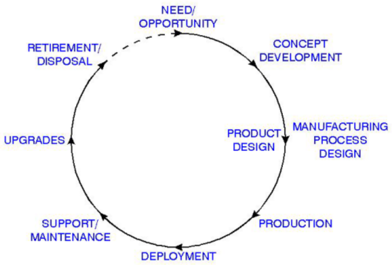


System Failure

- **System Costs**:
  80% of costs are committed early in design phase: when 20% of the actual cost has been accrued, 80% of the total LCC has already been determined.

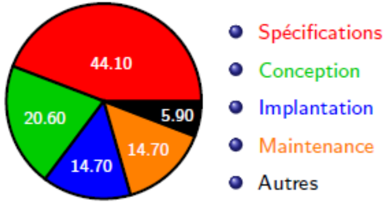
Introduction: Industrial Context

- **System Life Cycle**:
  Framework for meeting the stakeholders needs by defining lifecycle stages and using decision gates to determine readiness to move from one stage to the next.



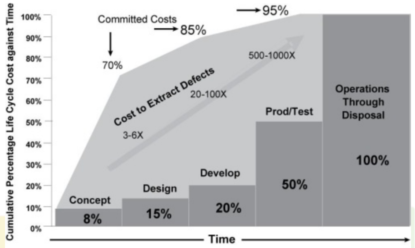
Life Cycle

- **Chaos Report**:
  Many errors are introduced early in design phase. System failures are due to errors in early phases.



System Failure

- **System Costs**:
  80% of costs are committed early in design phase: when 20% of the actual cost has been accrued, 80% of the total LCC has already been determined.



Project Failure

Hycham Aboutaleb

Introduction: Scope

Complexity

Life Cycle

Repeatability

System Engineering

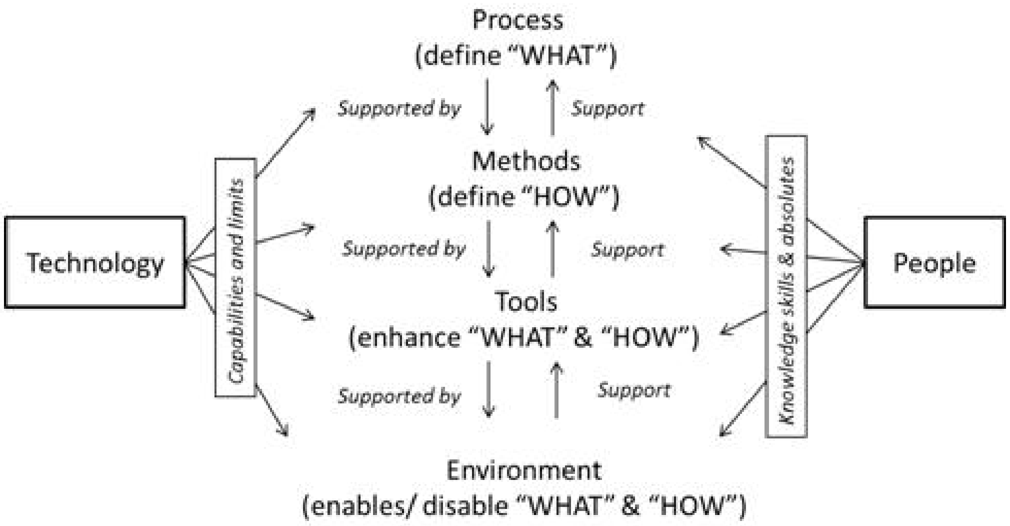
Project Failure

Heterogeneity

System Failure

System Engineering aims to decrease the probability of **system failure** and **project failure** during **system life cycle** by handling system **complexity** and **heterogeneity** using **repeatable** framework and processes.

System Engineering Framework



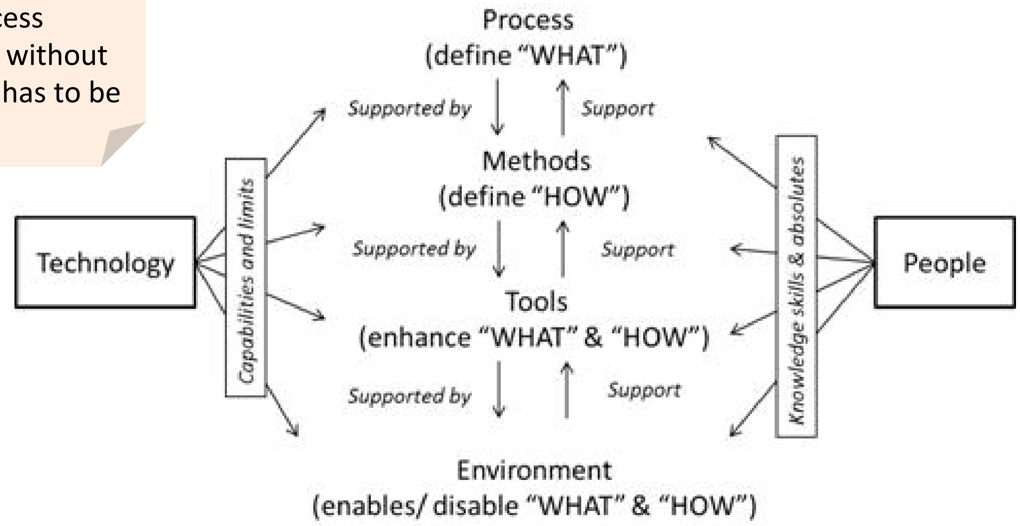
Credits: Martin, James N., *Systems Engineering Guidebook: A Process for Developing Systems and Products*, CRC Press, Inc.: Boca Raton, FL, 1996.

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.

.



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

System Engineering Framework

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

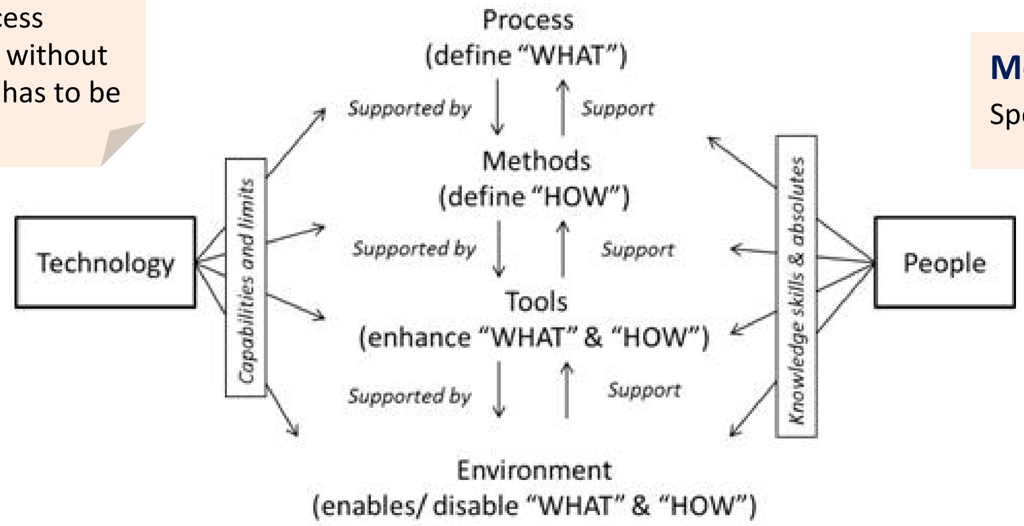
Process:

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



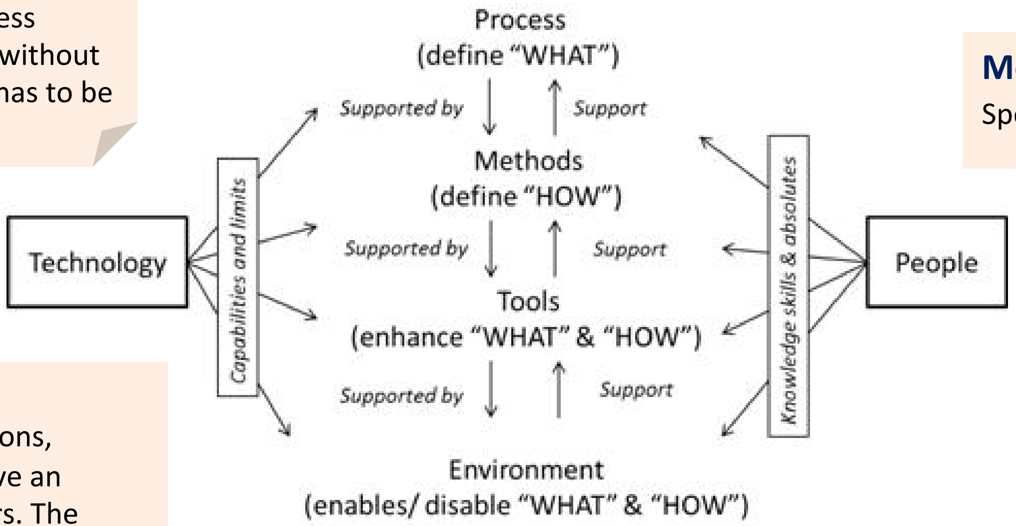
Credits: Martin, James N., *Systems Engineering Guidebook: A Process for Developing Systems and Products*, CRC Press, Inc.: Boca Raton, FL, 1996.

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

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.

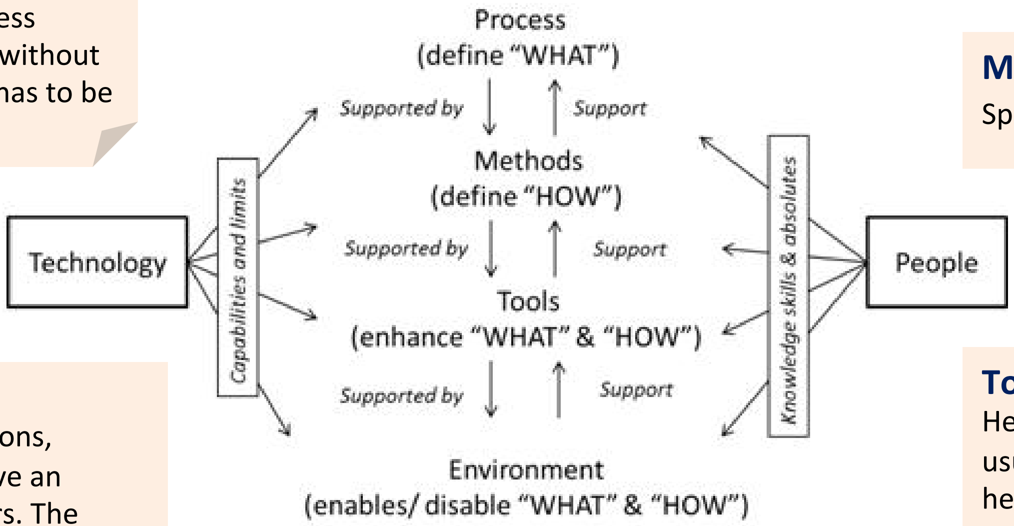
Credits: Martin, James N., *Systems Engineering Guidebook: A Process for Developing Systems and Products*, CRC Press, Inc.: Boca Raton, FL, 1996.

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

Environment:

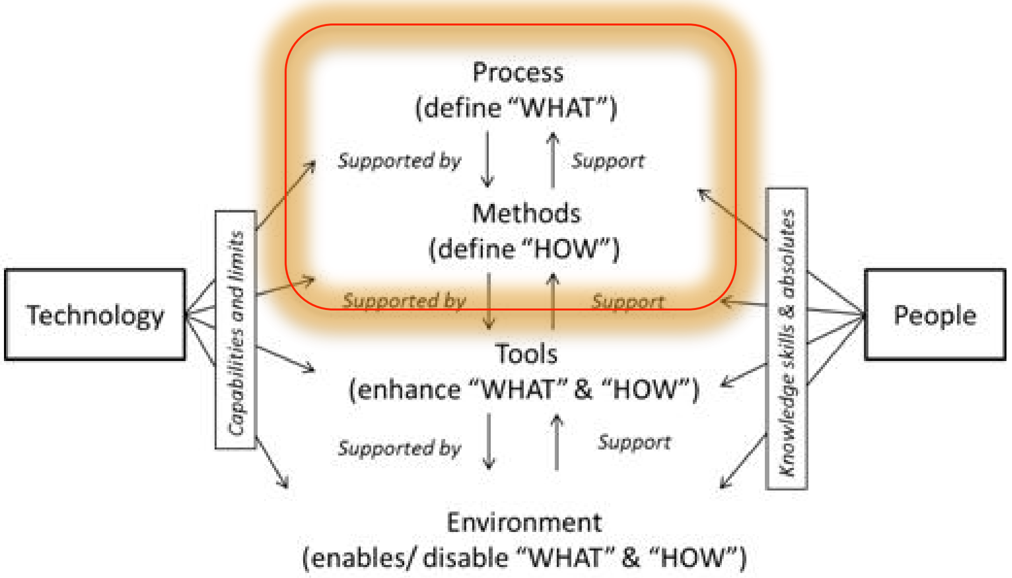
Consists of external conditions, systems, or factors that have an influence on systems, actors. The purpose of environment is to put in practice the use of tools and methods of a project.

Tool:

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

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

Modelling and Representation



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

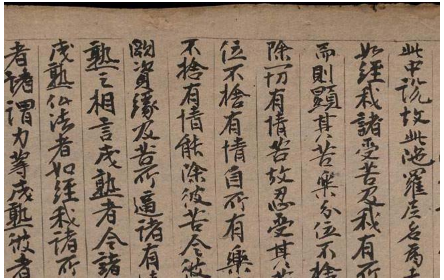
Modelling Needs

- 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

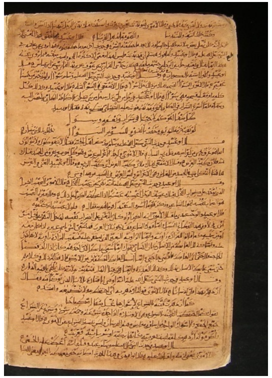
- Modelling Needs:
  - **Trustworthiness**: how close the model is to the reality?
  - **Understandability**: is the model perceived and understood the same way by people?
  - **Usefulness**: does the model help to get the desired results?

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

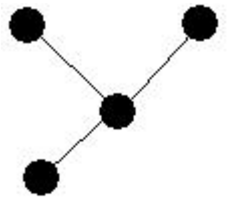


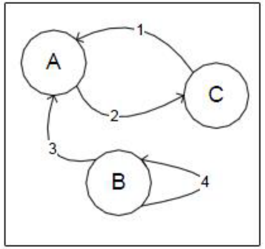
Graphical representations

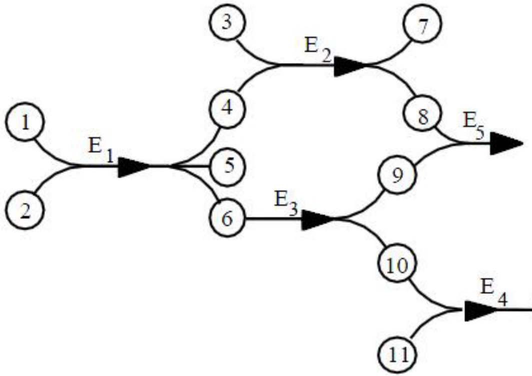
- Convey multiple features of a system simultaneously.
- Expressions correspond to the components of a model describing the system.
- Preserve explicitly the information about the topological and geometric relations among the elements of the system.

Graph-Based Models

- Graphs have been naturally used to represent and model problems since the emergence of computer science.
- Graph-based models give a visual and intuitive representation, as well as with required accuracy.
- They are a well-suited means to describe in a natural way all kinds of systems, where nodes describe system entities and edges describe relations between them.







System Model 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

System Model 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

Orthogonality

System Model 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

Orthogonality

-**Segmentation**: decomposition of a complex system into smaller parts that can be studied in isolation

Depth

Higraph-Based Models

- Graphs have been naturally used to represent and model systems.

- A higraph is a graph extended to include notions of depth and orthogonality:

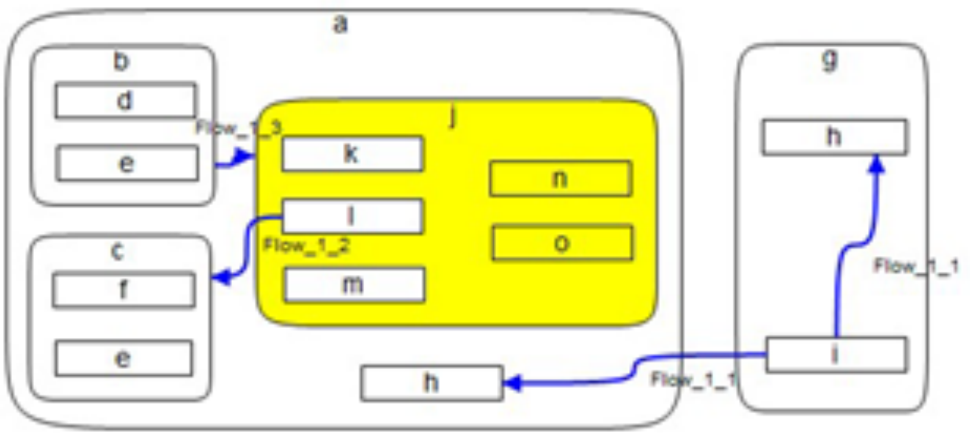
  Higraph = Graph + Depth + Orthogonality

- **Definition** (Higraph).

A higraph is a quadruble $H = (B, E, \rho, \Pi)$ where:

- B is the set of blobs (or nodes);
- E is the set of edges;
- ρ is the hierarchy function. It assigns to each blob b its set of sub-blobs $\rho(b)$;
- Π is the orthogonality (or partitioning function) defined as , associating with each blob some equivalence relation $\Pi(b)$ on the set of sub-blobs, $\rho(b)$.

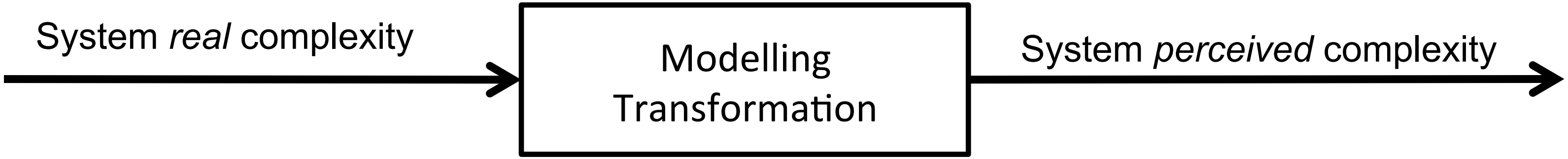
Higraph Formalism



- $\rho(a) = \{b, c, h, j\}$
- $\rho(g) = \{h, i\}$
- $\rho(b) = \{d, e\}$
- $\rho^2(a) = \{d, e, f, e, k, l, m, n, n, o\}$
- $\rho^{-1}(b) = \{a\}$
- $\rho^{-1}(h) = \{a, g\}$

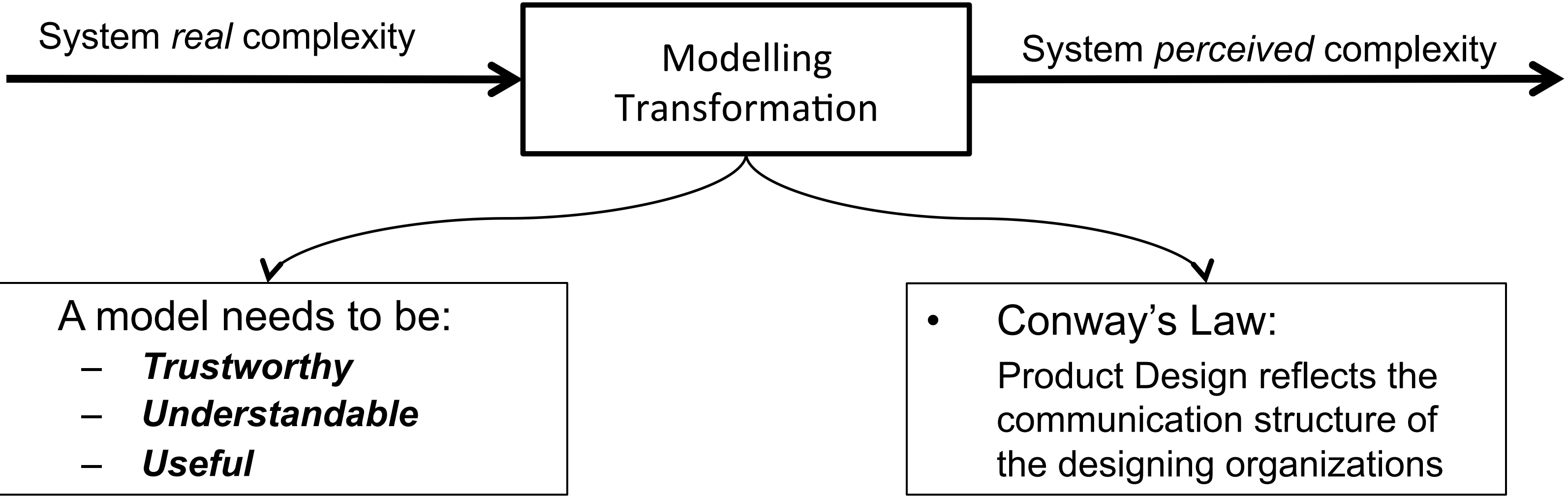
- $\Pi_w(a) = \{b, c, h\}$
- $\Pi_w(g) = \{h, i\}$
- $\Pi_y(a) = \{j\}$
- $\Pi_w(j) = \{k, l, m\}$
- $\Pi_y(j) = \{n, o\}$

System Complexity

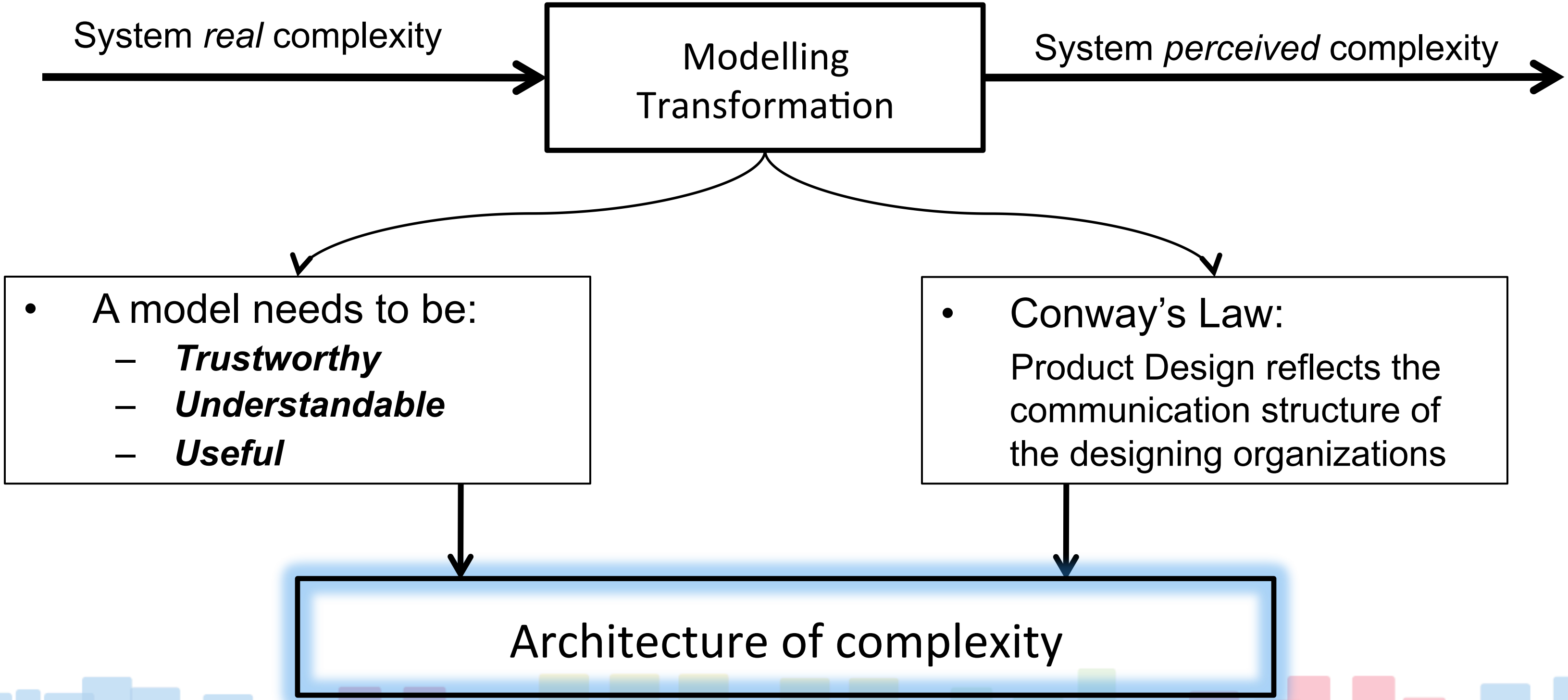


ENSTA ParisTech
26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

System Complexity



System Complexity



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

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.

Complexity Metrics

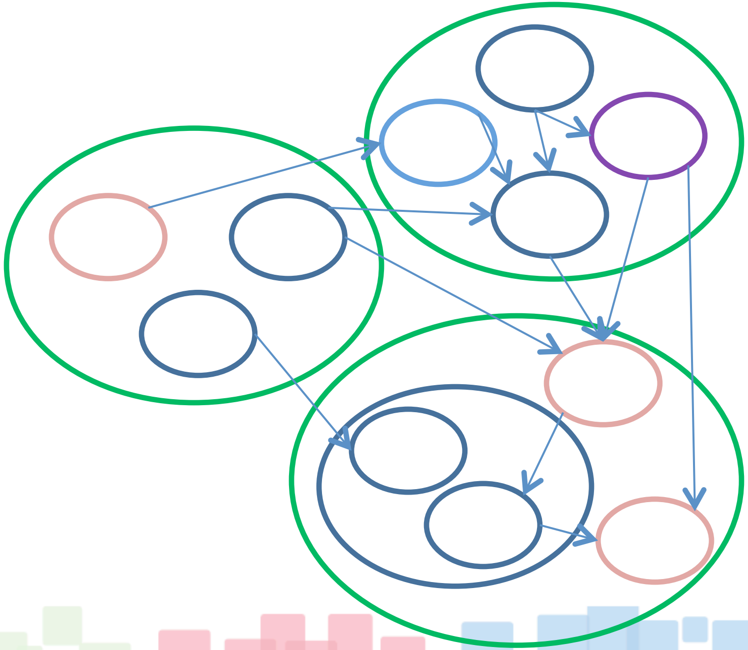
- Direct Metrics:
  - Size
  - Depth
  - Width
- Indirect Metrics:
  - Density
  - Type Variety
  - Interface Load
- Structural Complexity

Direct Metrics

- **Size:**
  The most obvious and useful attributes of a model is its size. The most intuitive way is to take into account the number of nodes and the number of edges.

- **Depth:**
  The depth of a higraph-based model is the highest number of levels between the top node and the lowest level node.

- **Width:**
  The width of a higraph-based model is the highest number of nodes at any one level.



Direct Metrics

- **Size:**
  The most obvious and useful attributes of a model is its size. The most intuitive way is to take into account the number of nodes and the number of edges.

- **Depth:**
  The depth of a higraph-based model is the highest number of levels between the top node and the lowest level node.

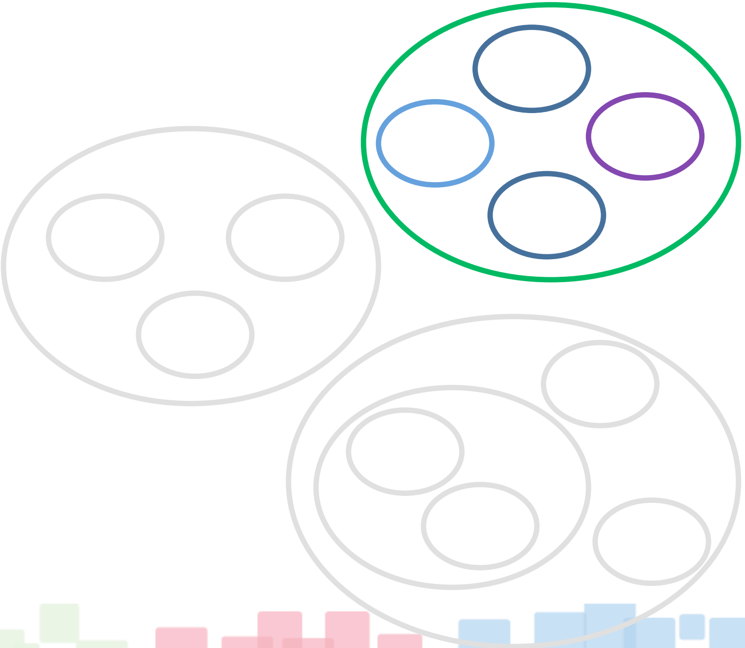
- **Width:**
  The width of a higraph-based model is the highest number of nodes at any one level.

Direct Metrics

- **Size:**
  The most obvious and useful attributes of a model is its size. The most intuitive way is to take into account the number of nodes and the number of edges.

- **Depth:**
  The depth of a higraph-based model is the highest number of levels between the top node and the lowest level node.

- **Width:**
  The width of a higraph-based model is the highest number of nodes at any one level.

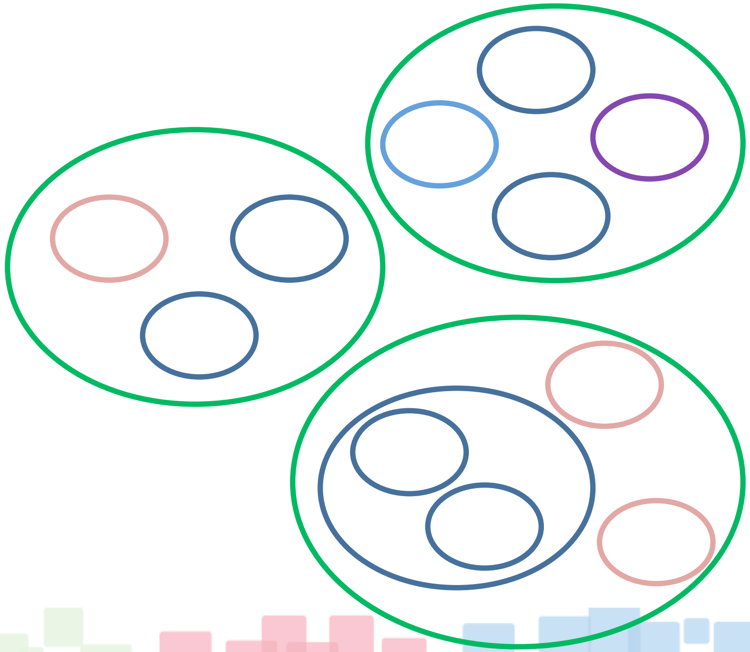


Indirect Metrics

- **Density:**
  It measures the node constituents to the number of nested components. This metric is used to identify the density of nested elements.

- **Type Variety:**
  The number of types in a set of elements is a good indicator of variety if all the types are of equal importance, which is usually not the case.

- **Interface Load:**
  This index measures the average number of interface inputs into an element and the average number of interface outputs of an element.



Indirect Metrics

- **Density:**
  It measures the node constituents to the number of nested components. This metric is used to identify the density of nested elements.

- **Type Variety:**
  The number of types in a set of elements is a good indicator of variety if all the types are of equal importance, which is usually not the case.

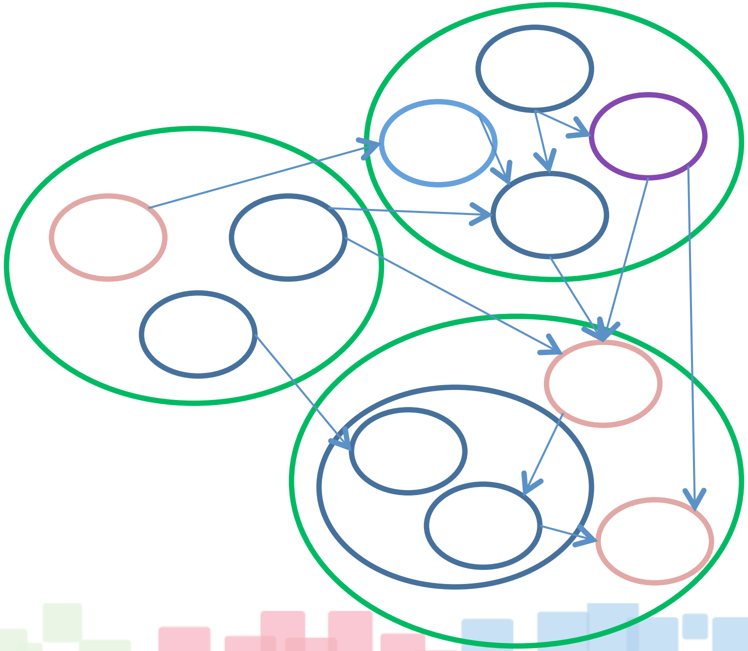
- **Interface Load:**
  This index measures the average number of interface inputs into an element and the average number of interface outputs of an element.

Indirect Metrics

- **Density:**
  It measures the node constituents to the number of nested components. This metric is used to identify the density of nested elements.

- **Type Variety:**
  The number of types in a set of elements is a good indicator of variety if all the types are of equal importance, which is usually not the case.

- **Interface Load:**
  This index measures the average number of interface inputs into an element and the average number of interface outputs of an element.



Structural Complexity

- A higraph model *M* entropy intuitively depends on the number of blobs, the number of edges, the hierarchy and the orthogonality.

$$M = (B, E, \rho, \Pi)$$

- We get the total entropy of the model higraph *M* as follows:

$$H = H_B + H_E + H_\rho + H_\Pi$$

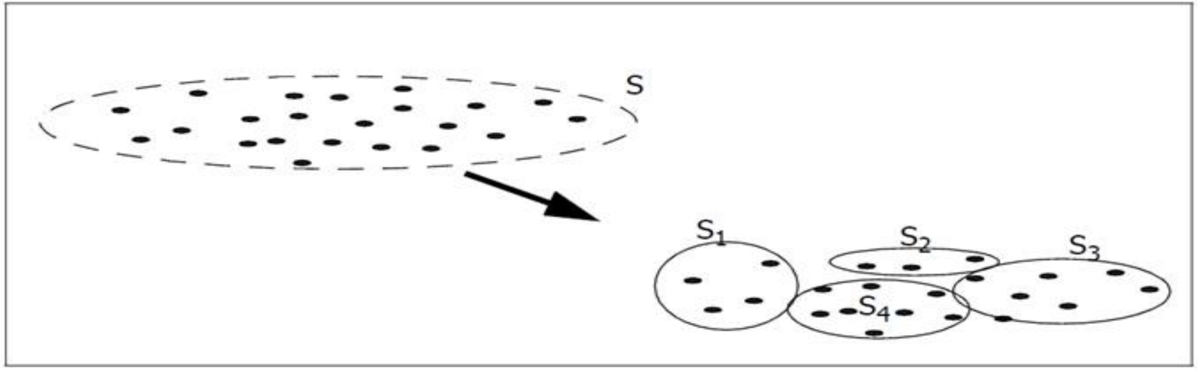
- Indicator of the complexity: Shannon's entropy

Shannon's Entropy

- Shannon's Entropy:
  For a set *S* with N_S states

$$H(S) = -\sum p_i \log p_i$$

- Design principle:
  A complex problem is decomposed into a set of smaller problems with smaller complexity. Besides, the global complexity is the same.



Hycham Aboutaleb

Structural Complexity (1)

- We get the total entropy of the model higraph *M* as follows:

$$H = H_B + H_E + H_\rho + H_\Pi$$

- The first term relates to the complexity due to the number of nodes.

- $H_B = H(B) = -log(1/|B|) = log(|B|)$

Structural Complexity (2)

- We get the total entropy of the model higraph *M* as follows:

$$H = H_B + H_E + H_\rho + H_\Pi$$

- The second term relates to the complexity due to the number of edges. It takes into account the head and the tail of the edge.

- $H_E = H(E) = -2log(1/|E|) = 2log(|E|)$

Structural Complexity (3)

- We get the total entropy of the model higraph *M* as follows:

$$H = H_B + H_E + H_\rho + H_\Pi$$

- The third term relates to the number of hierarchical relationships between the elements of the model. It takes into account parent and child relationship.

- $H_\rho = -2log(1/N) = 2log(\sum \rho(x)),\ \ for\ x \in M$

Hycham Aboutaleb

Structural Complexity (4)

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

- We get the total entropy of the model higraph *M* as follows:

$$H = H_B + H_E + H_\rho + H_\Pi$$

- The fourth term relates to the complexity due to the orthogonality.
- Let *g* a morphism that associates to each element *x* of the Model Higraph *M* to its type, with M_Π, the Model Type Higraph. B_Π is the corresponding set of blobs. <u>There is no edge and no type in the Type Higraph</u>.

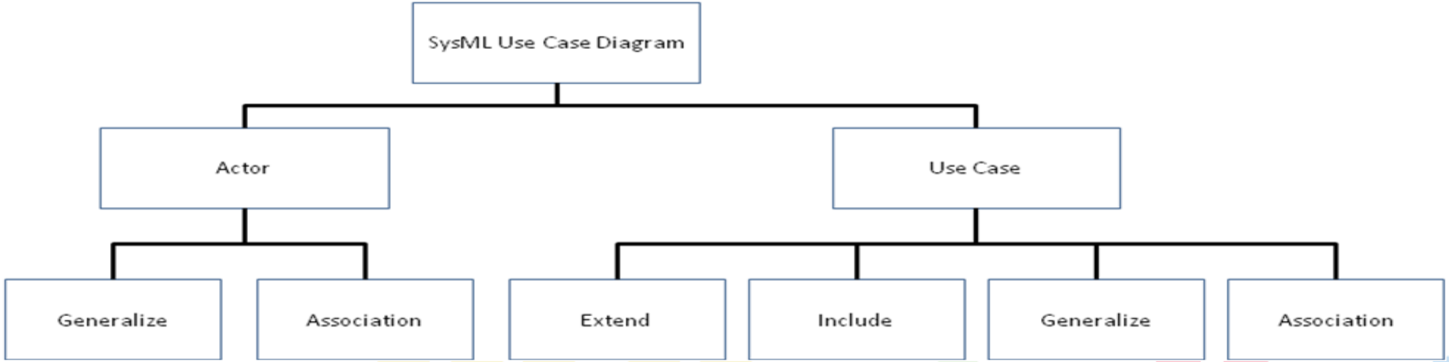
- $H_\Pi = log(|B_\Pi|) + 2log(\sum \rho(x)), \;\; for \; x \in M_\Pi$

Example: Use Case Diagram

- SysML Use Case Diagram is a higraph $H_{UC} = (B;E;\rho;\Pi)$.

- Each node *n* belongs to one and only one of the following types: *Actor, Use Case*

- Each edge e belongs to one and only one of the following types: (*Associate, Generalize, Extend, Include*).



Example: Use Case Diagram

- 3 elements of type *Actor* and 3 elements of type *Use Case*:

$$H_B = \log_2 6$$

- 3 relations of type *Association:*

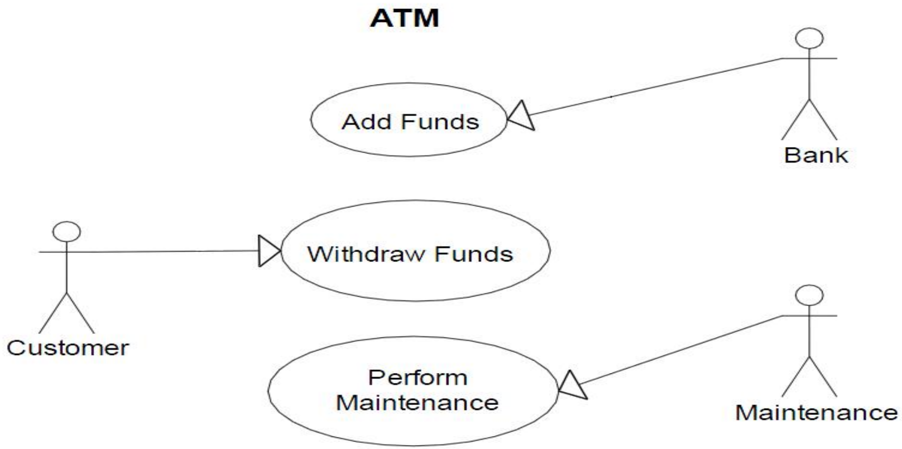
$$H_E = 2\log_2 3 = \log_2 9$$

- We have no hierarchy:

$$H_\rho = 0$$

- Type Higraph:

$$H_\Pi = H(M_\Pi) = \log_2 216$$



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.
- Its value depends on the amount of details, elements and relationships between them, as well as the number of hierarchy levels.
- Smaller sets mean less complexity.
- The choice of aggregation allows dealing with subsets separately to handle this complexity.
- The complexity measurement is a relevant metric to compare different architectures for the same system.

ENSTA ParisTech

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

Questions?

hycham.abou-taleb@ensta-paristech.fr