Knowledge Centric Systems Engineering

Dr. Juan Llorens

Technical Director - Asociación Española de Ingeniería de Sistemas (AEIS) – INCOSE
Professor at Informatics Department - Universidad Carlos III de Madrid – Spain
CTO – The Reuse Company (TRC)

Juan.Llorens@uc3m.es

http://www.linkedin.com/pub/juan-llorens/b/857/632

https://www.researchgate.net/profile/Juan_Llorens/

South European Systems Engineering Tour (SESE 2014)
Zurich 1st September - Paris 23rd September - Madrid 24th September 2014

Systems Engineering in a Nutshell

What is Systems Engineering

Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems.

It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal.

SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.

Source INCOSE

Applied to:

- Complex Systems
- Systems developed by collaboration of multiple engineering disciplines

Why Systems Engineering

- Attempting to minimize what we already know that happens:

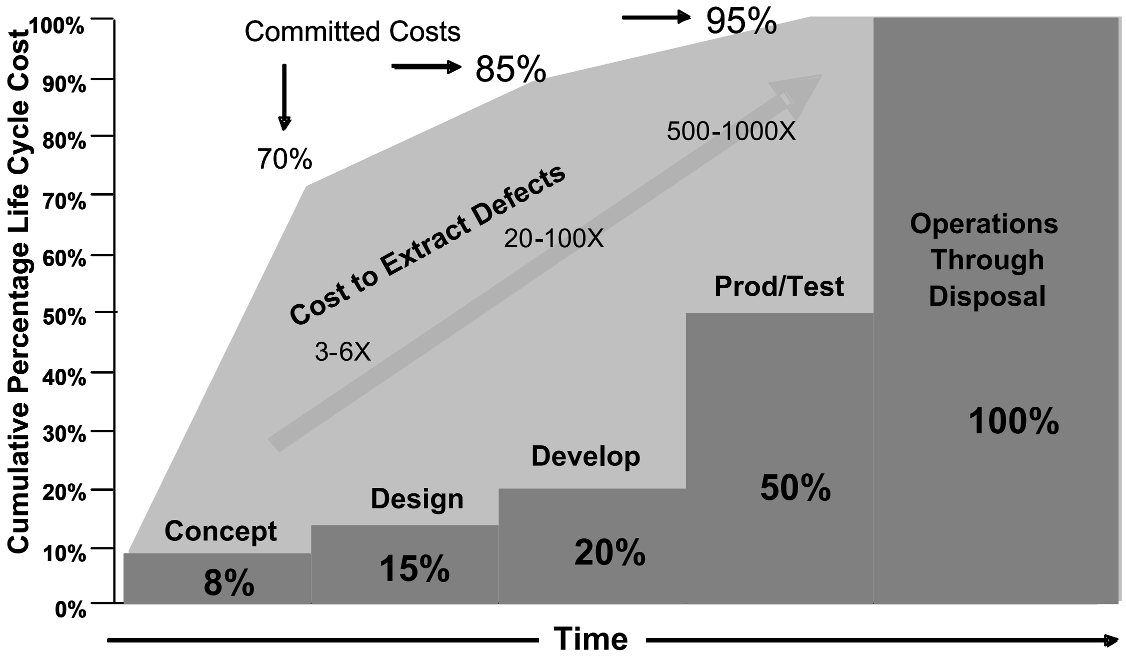


Figure 2-4 Committed Life Cycle Cost against Time (INCOSE HB)

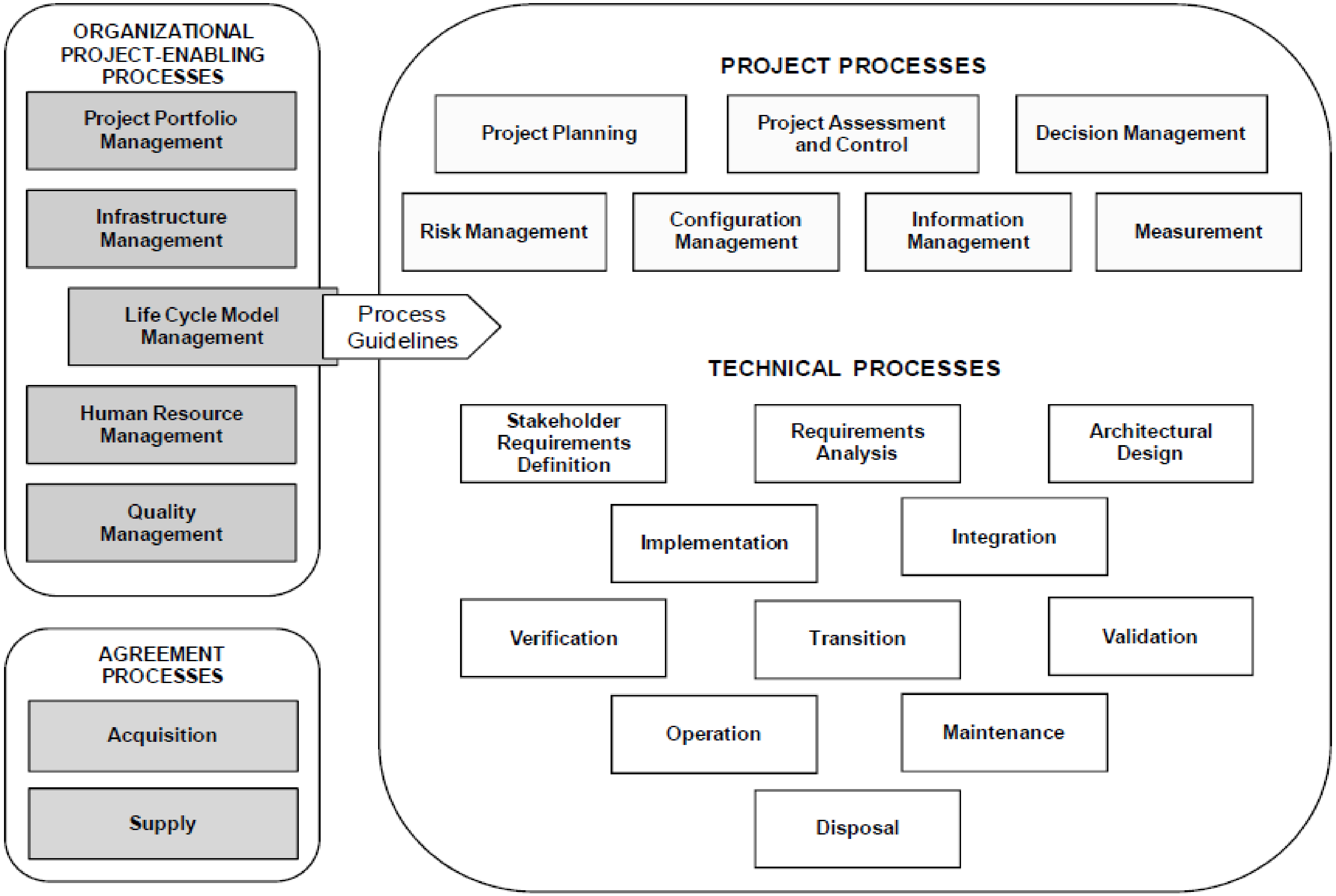
- Minimize defects, increase quality, reduce cost, reduce TTM, etc.

How Systems Engineering : A Life Cycle

LIFE-CYCLE STAGES	PURPOSE	DECISION GATES
EXPLORATORY RESEARCH	*Identify stakeholders' needs* *Explore ideas and technologies*	*Decision Options* – *Proceed with next stage* – *Proceed and respond to action items* – *Continue this stage* – *Return to preceding stage* – *Put a hold on project activity* – *Terminate project.*
CONCEPT	*Refine stakeholders' needs* *Explore feasible concepts* *Propose viable solutions*	
DEVELOPMENT	*Refine system requirements* *Create solution description* *Build system* *Verify and validate system*	
PRODUCTION	*Produce systems* *Inspect and verify*	
UTILIZATION	*Operate system to satisfy users' needs*	
SUPPORT	*Provide sustained system capability*	
RETIREMENT	*Store, archive, or dispose of the system*	

Source: INCOSE's Systems Engineering Handbook

How Systems Engineering : Processes and Activities



How Systems Engineering : The V-Model

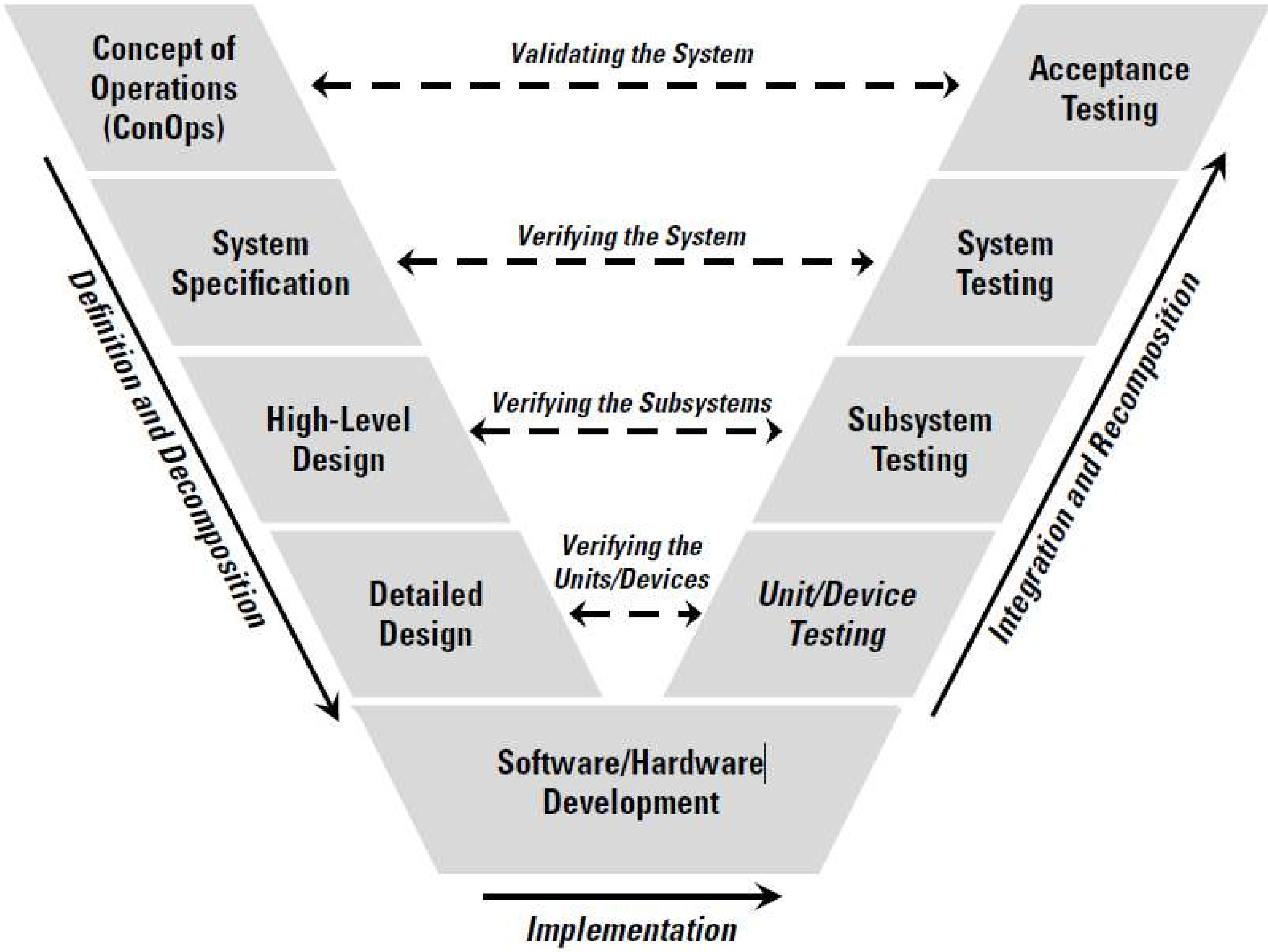
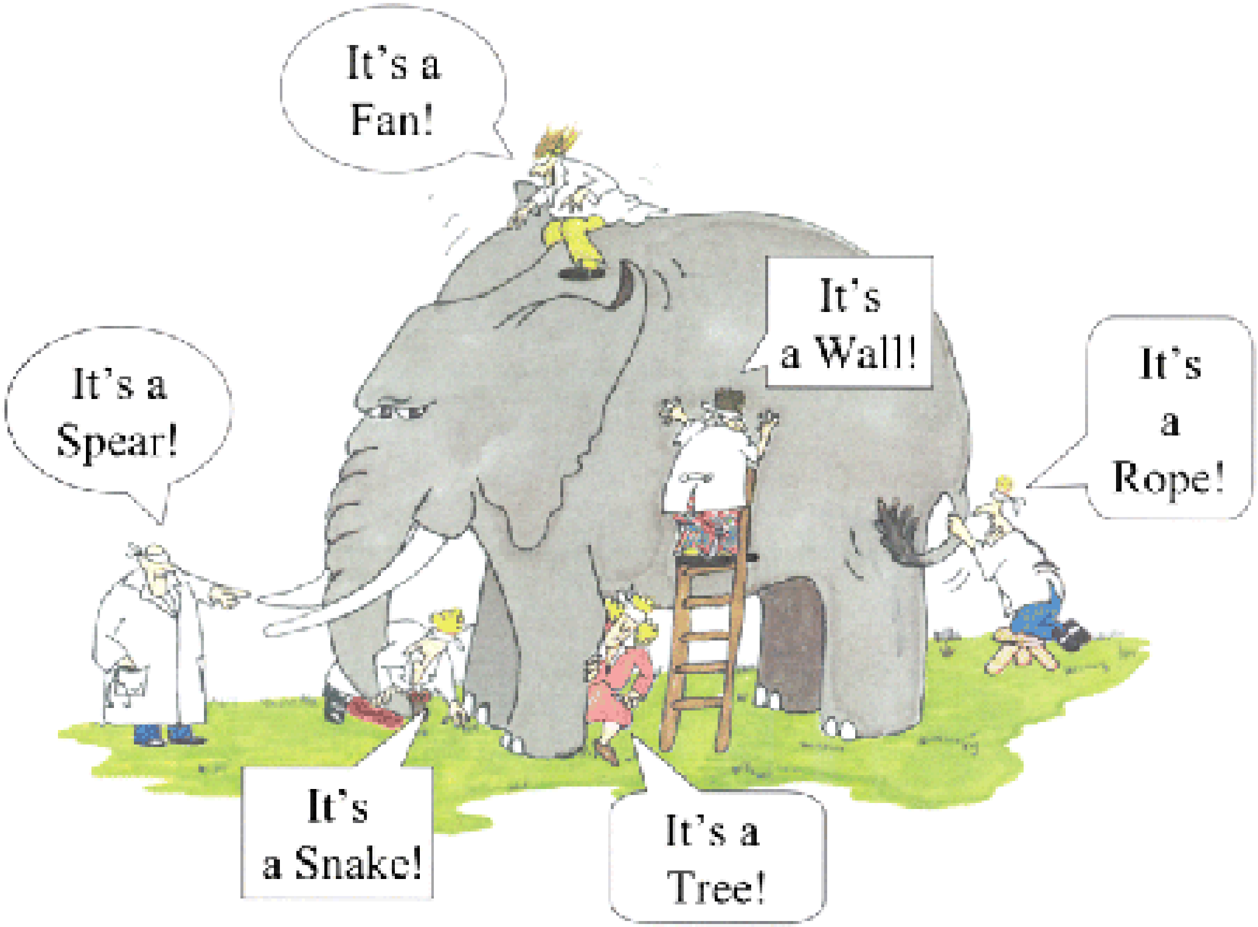


Figure 2-2 The V-model **Source: SE for dummies**

How Systems Engineering: System thinking



https://suifaijohnmak.wordpress.com/2009/03/19/

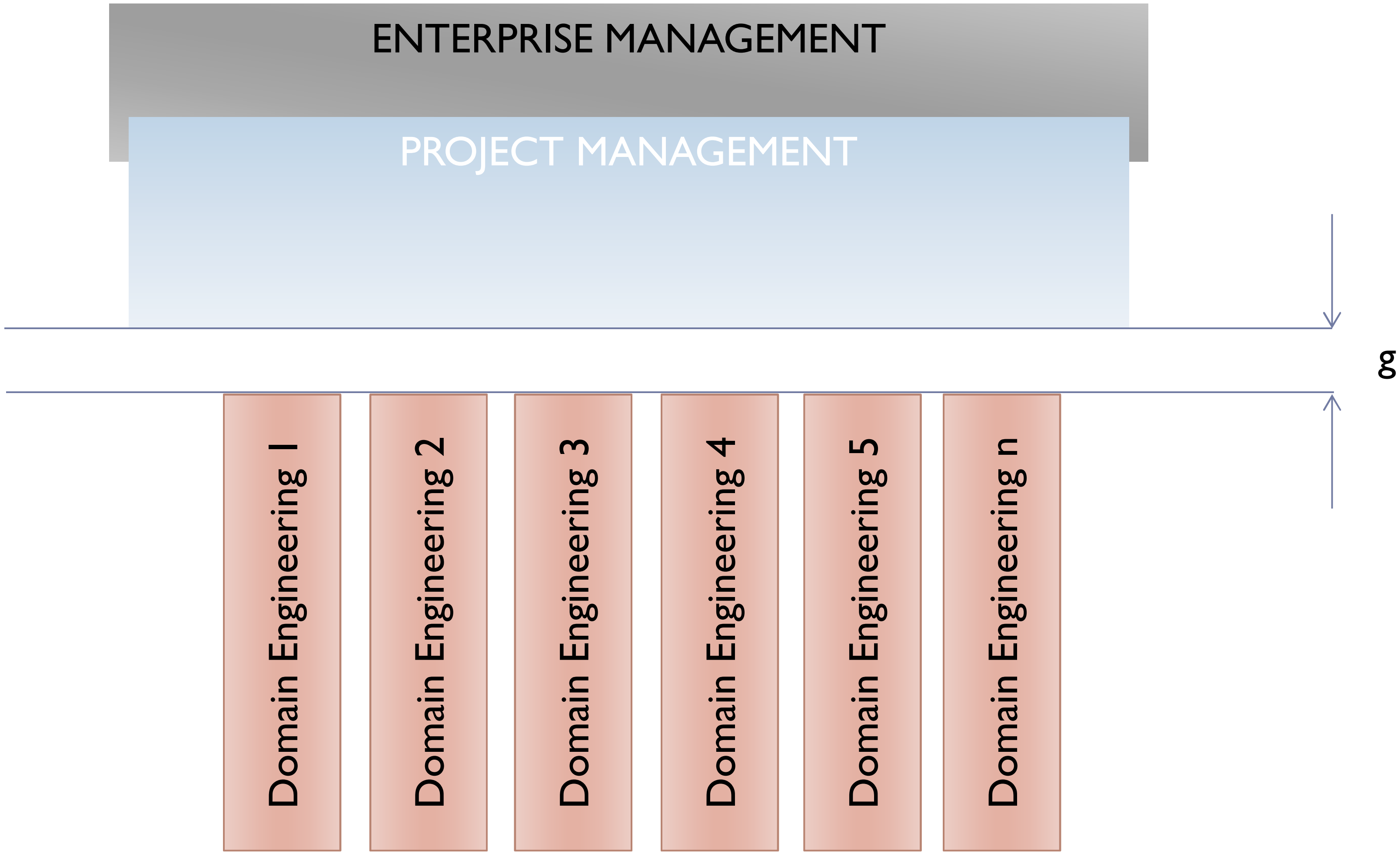
How Systems Engineering: STANDARDS

International Organization for Standardization

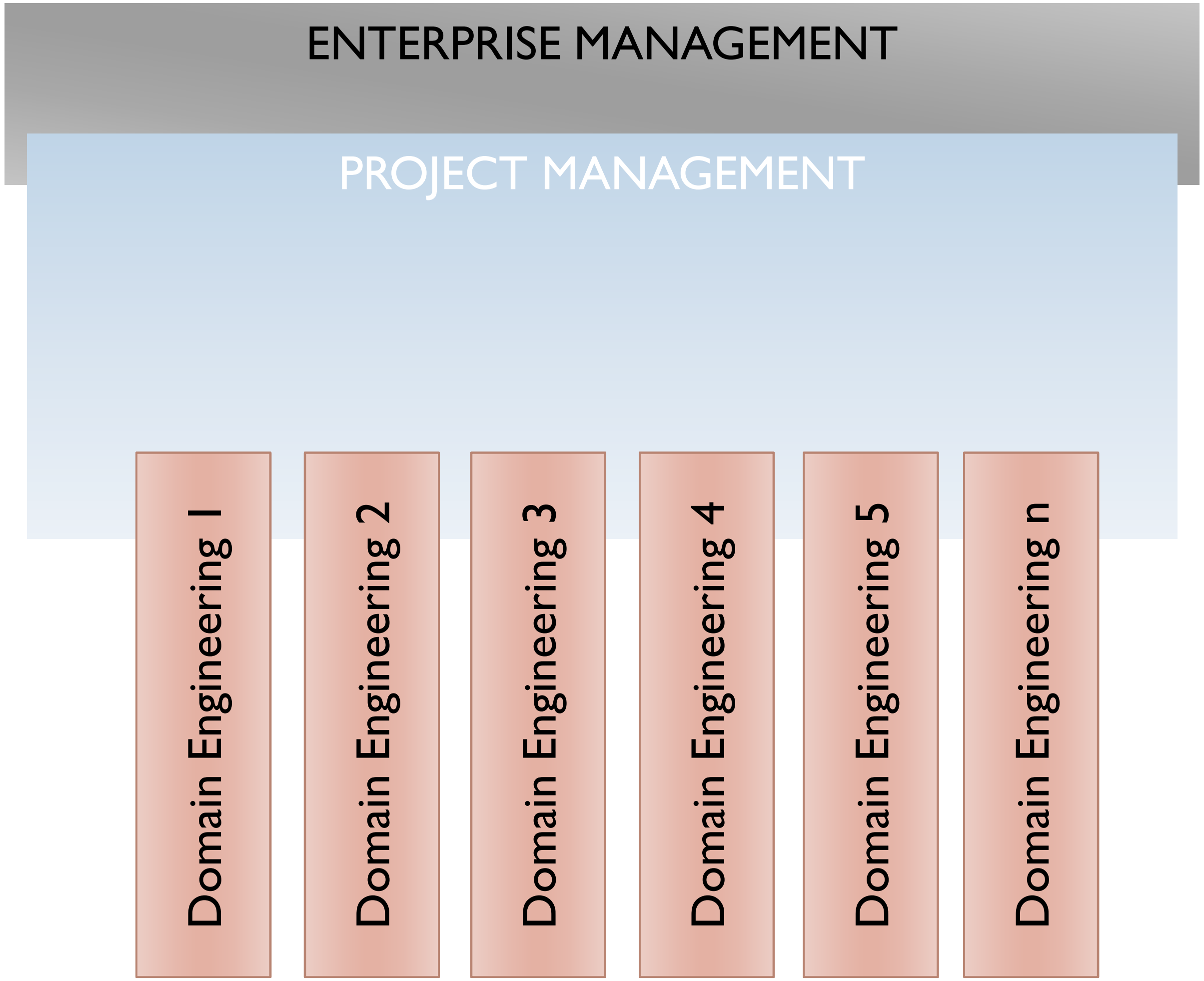
INCOSE International Council on Systems Engineering

ISO- IEC 15288

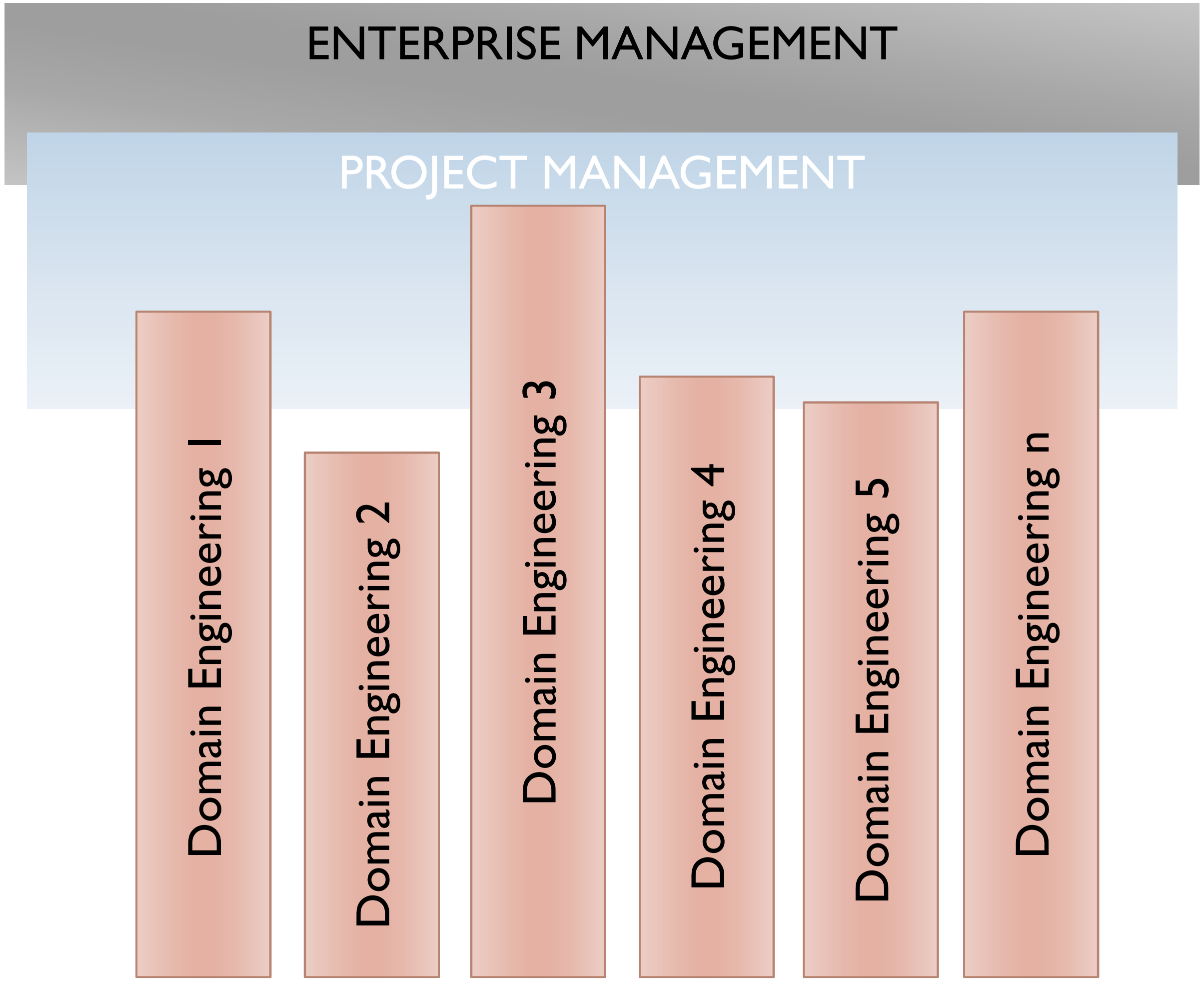
Where Systems Engineering



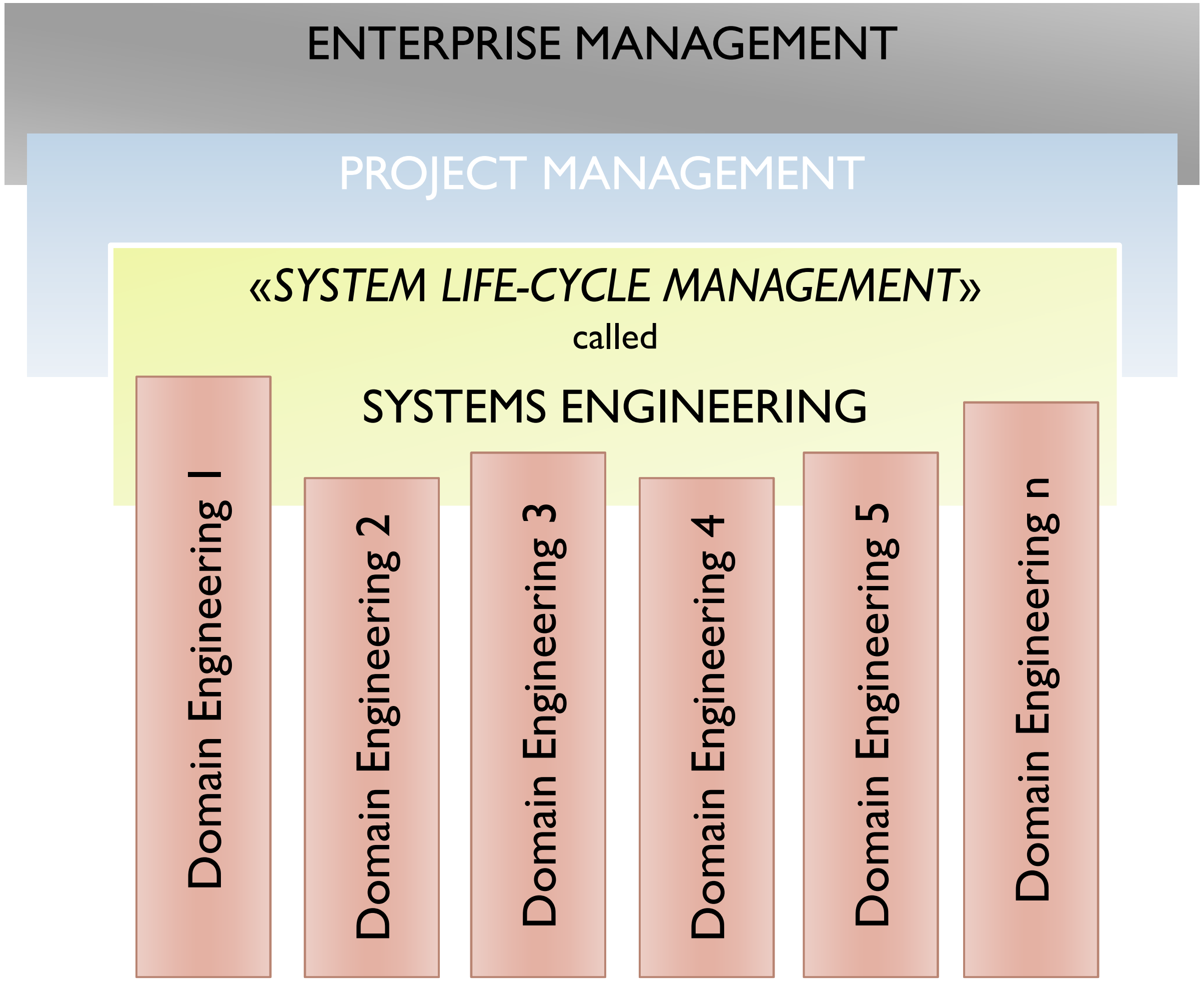
Where Systems Engineering : Classical solution



Where Systems Engineering : Other classical solution



Where Systems Engineering : A good solution



Systems Engineering : In practice

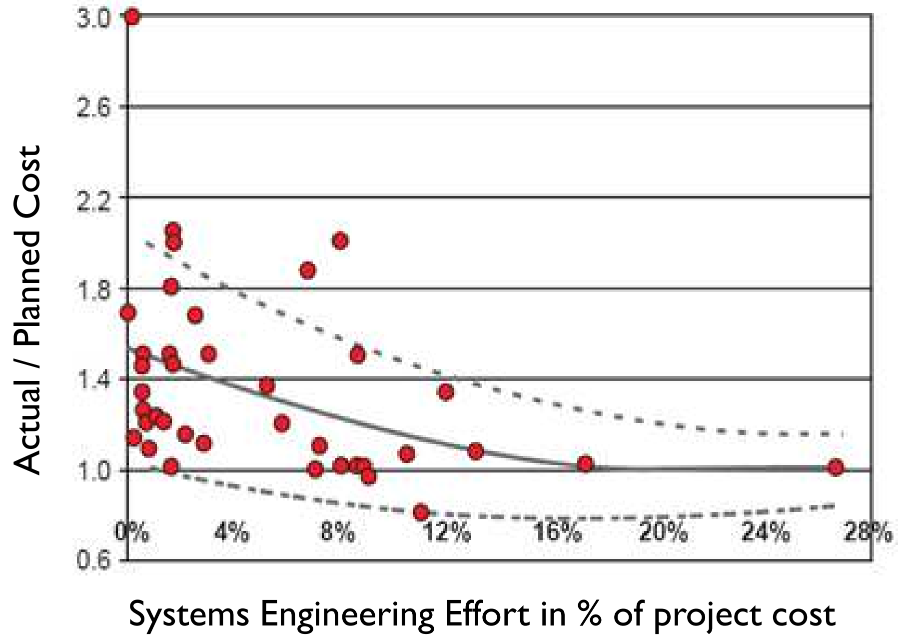


Figure 2-7 Cost and schedule overruns correlated with systems engineering effort (INCOSE HB)

http://www.incose.org/SECOE/0103/0103results.htm

Cost and schedule overruns lessen with increasing SE effort.
Variance also lessens with increasing SE effort.

Fundamentals of Knowledge Centric Systems Engineering (KCSE)

Modern Challenges in Systems Engineering

If you are novice in Systems Engineering… your challenge is …

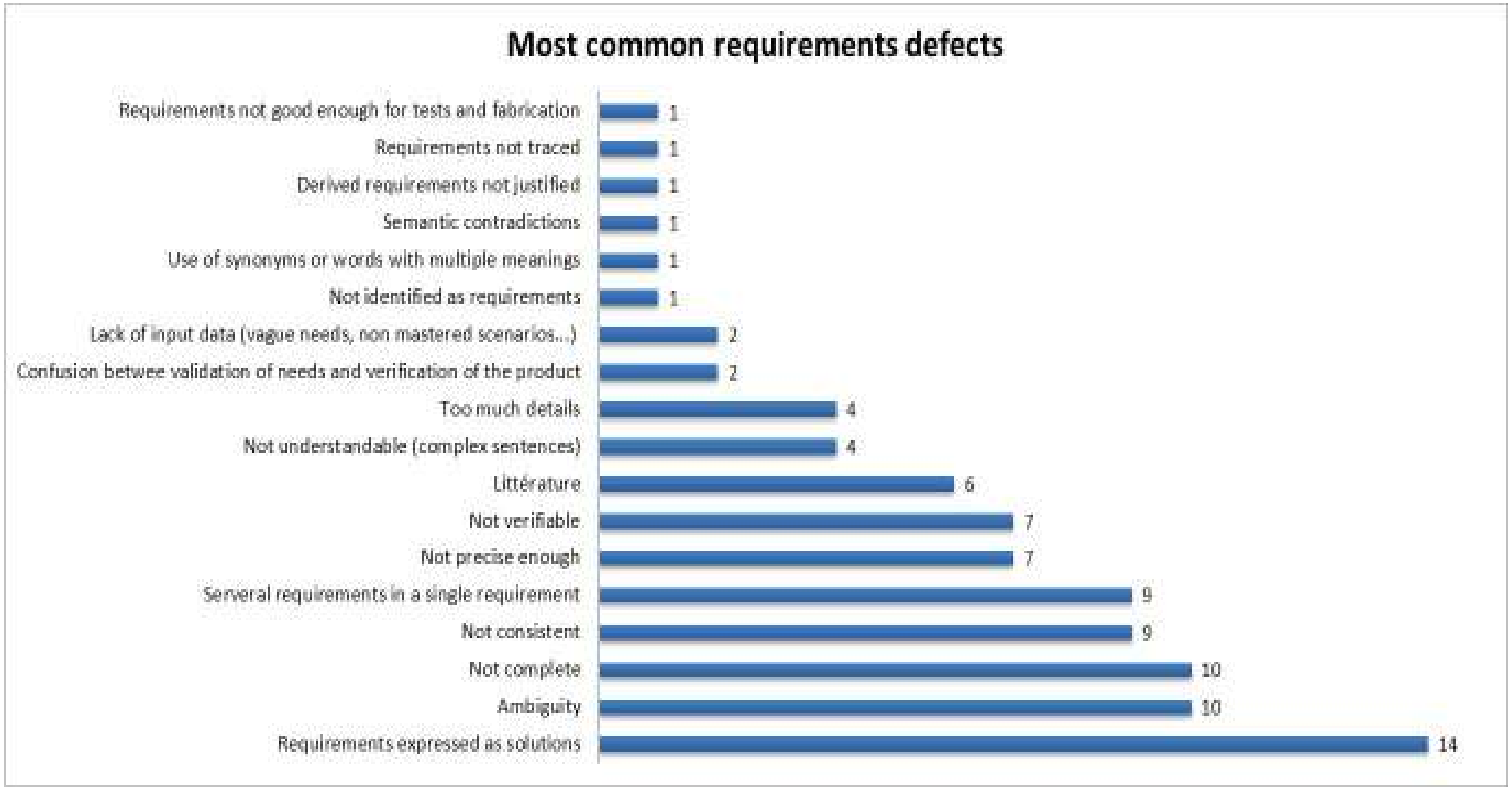
Systems Engineering itself !

If you already apply Systems Engineering, probably you would like to deal with issues around:

- Improve Quality Issues
- Promote Interoperability
- Offer Systems Engineering work-products Reuse
- Enhance the Authoring concept
- Identify and state integral and universal Traceability
- Move from Document Driven to Model Based SE
- ….

- At the end => improve Decision Support Systems (DSS)

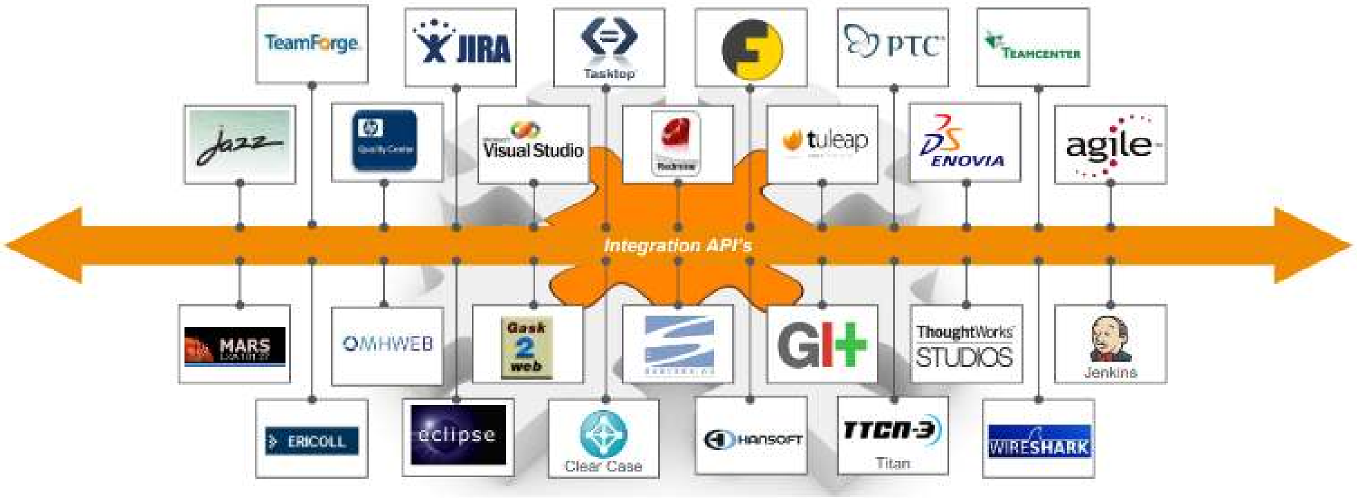
Requirements Quality: The Problem



(source: Gauthier Fanmuy - the RAMP project: - AFIS)

Interoperability

The Problem…



Mats Berglund (Eriksson)

http://www.ices.kth.se/upload/events/13/84404189f85d41a6a7d1cafd0db4ee80.pdf

Reuse : The Problem

SE Authoring : The Problem

- What is this?
- How a computer can guide me around my own knowledge?

http://grammar.ccc.commnet.edu/grammar/composition/computer.htm

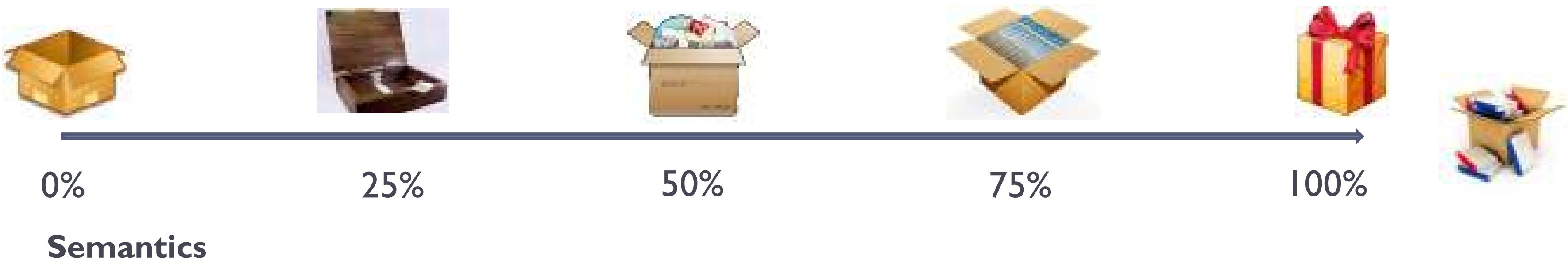
- …. By modeling, representing and reasoning around your own knowledge!

Knowledge Needs : Practical examples

- How should I write proper performance requirements?
  - It could be possible if .... My organization stores specific requirements patterns
- How can my models be aware of the existing requirements?
  - It could be possible if .... System architects can get access to requirements terminology when they are modeling
- Can an authoring technology advice of inconsistency problems in my model?
  - It could be possible if .... The complete requirements specification is formalized inside a repository
- Can I look for similar physical models when defining a simulation case?
  - It could be possible if .... All the physical models are stored inside a repository

Need of knowledge for better Systems Engineering

- The "smarter" we need systems engineering to be, the more dependent on "semantic" knowledge must it be.



- Knowledge must be represented within a knowledge structure (KOS)
  - from internal representations to glossaries, to ...., to ontologies)

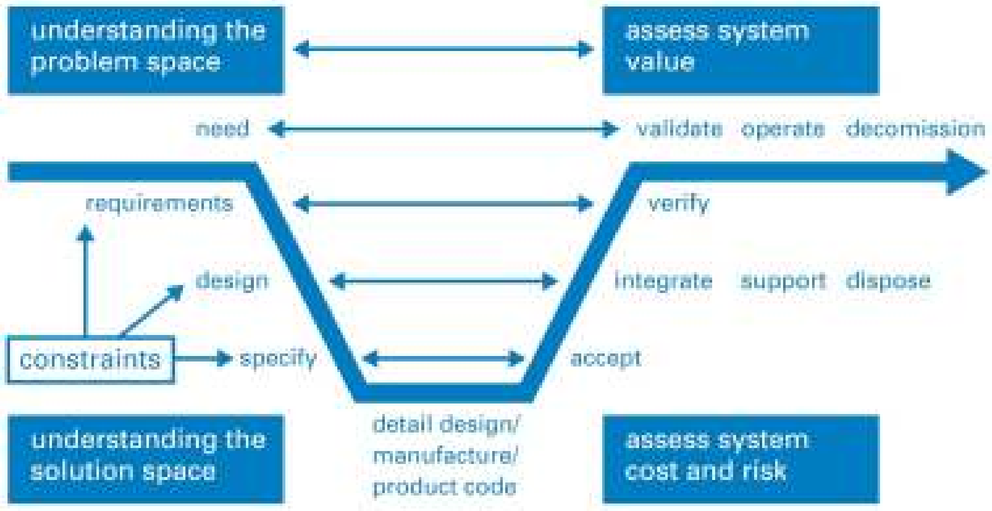
- The selection of the knowledge structure allows different possibilities to the organization

Knowledge Management today: an IT issue

Semantic web in a nutshell

- **Common and shared data model**
  - **Graph** (subject, object, predicate)
  - **RDF** with different serialization formats
  - Implicit **Multilinguism** suport
- **Knowledge Representation & Management**
  - Ontologies
  - **OWL** (Ontology Web Language)
  - **Logic** formalism: DL, F-Logic, etc.
  - **Reasoning**
  - Expert systems
  - **Standards: Query Languages, Vocabularies,** Datasets, ...

Knowledge Organization in Systems Engineering



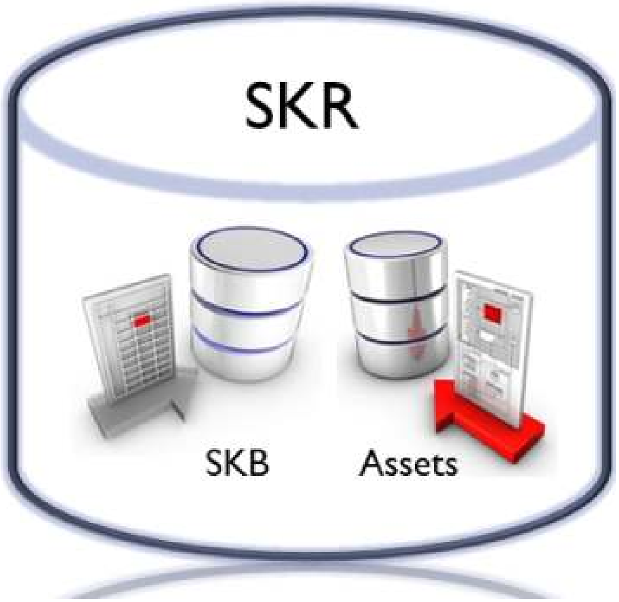
(source: INCOSE –UK chapter)

Organize your knowledge

in a System Knowledge Repository

Knowledge Organization in Systems Engineering

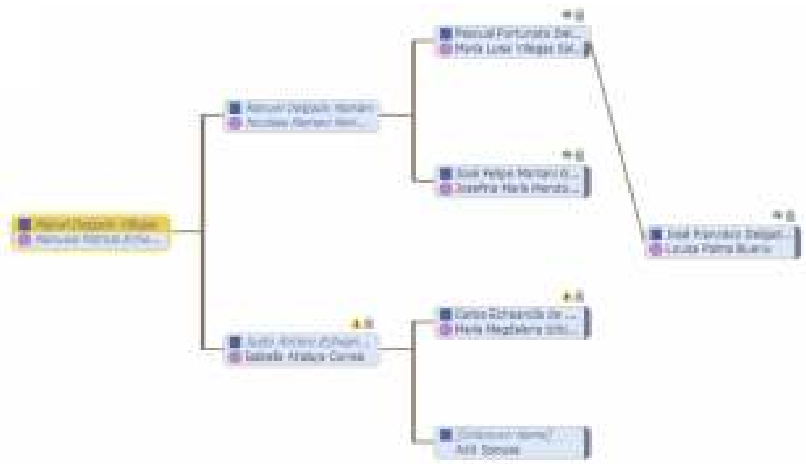
- System Knowledge Repository (SKR)
  - Allows representing, storing, managing and retrieving
    - Relevant knowledge around the System and its domain (including the SE Process)
    - Digital content (Assets) regarding a particular System
- The SKR is formed by
  - SKB – System Knowledge Base
  - SAS – System Assets Store



System Knowledge Base

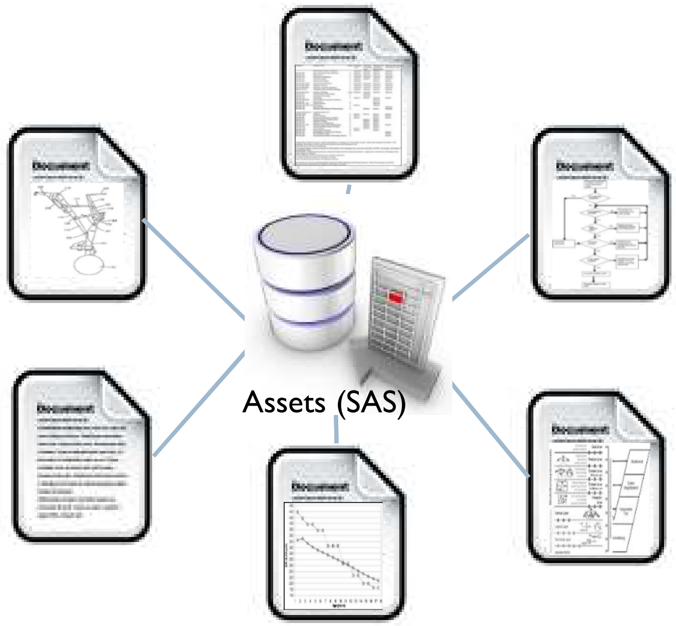
- SKB
  - Supports the complete representation of system (engineering) knowledge for the application of semantic services around the system life cycle (Including SE).
  - Knowledge is organized around the System Conceptual model

System Conceptual Model (SCM)

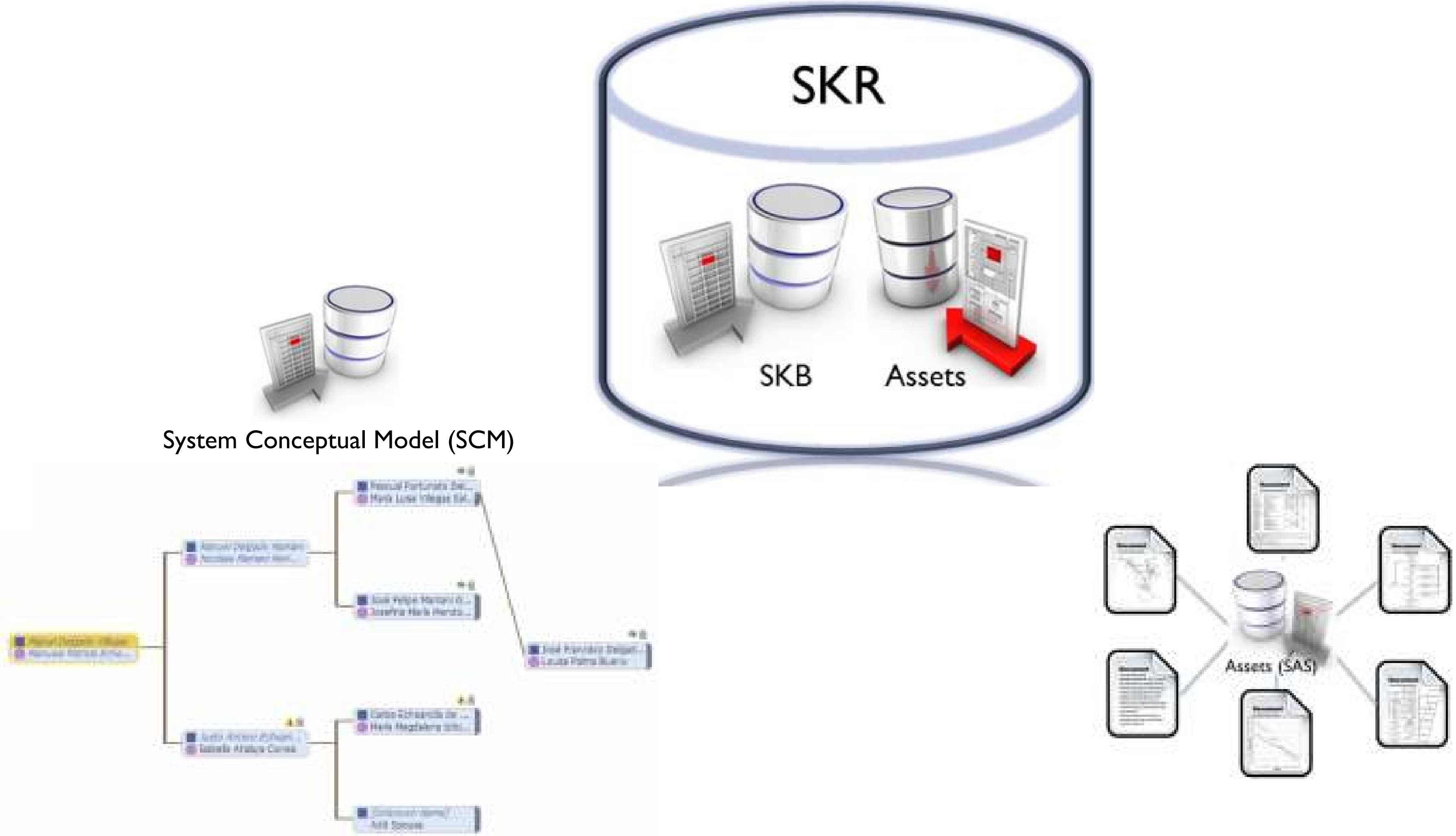


System Assets Store (SAS)

- SAS
  - Manages a formal representation of the System Assets: Requirements, Models, etc.
  - Is the base for offering services around these assets
    - Reuse
    - Traceability
    - MDE, TDD, etc.



SKR: Structure

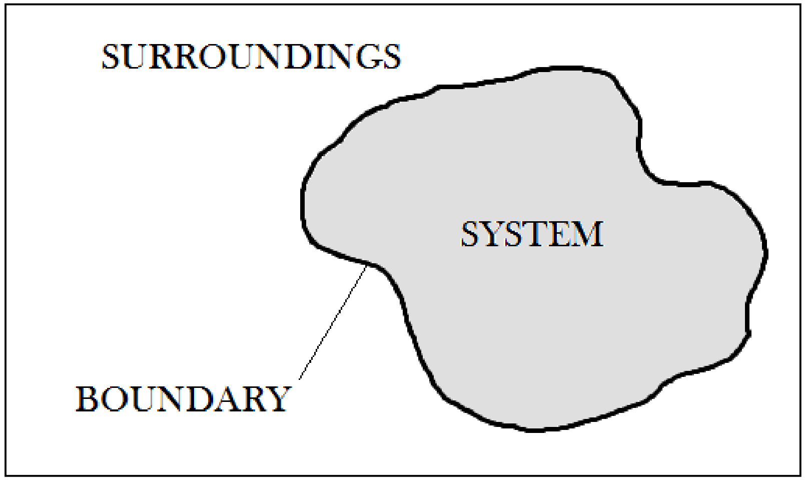


System Knowledge Base:

What is a system

Interacting objects organized to achieve one or more stated purposes [INCOSE & ISO 15288]

- A System is never alone. It is affected by its surroundings, and interacts with them through an interface (boundary).



System universe

this "ecosystem" can be called System universe

System Knowledge Base:

What is a System knowledge Base

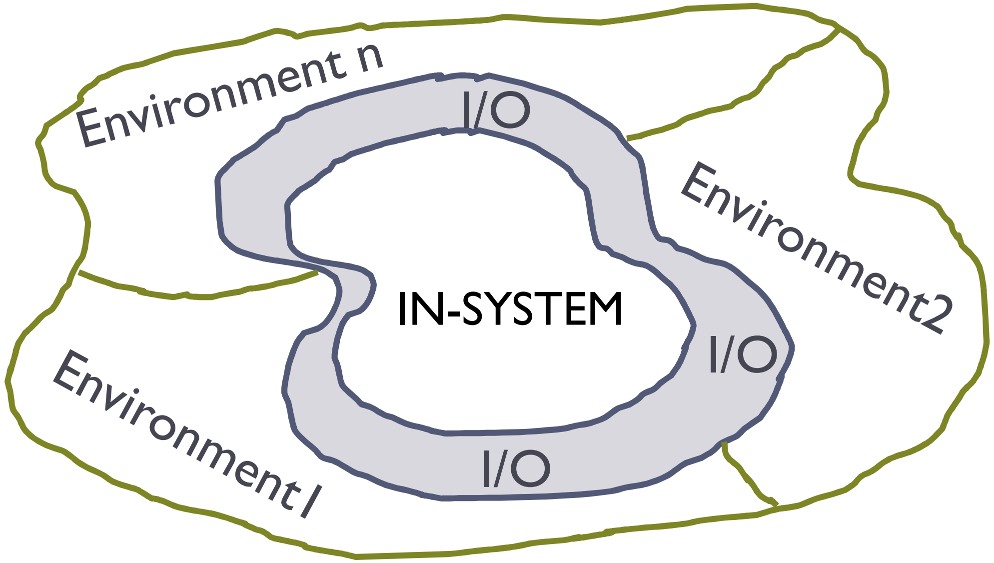
- The SKB represents a conceptual model of the system universe (SCM)
- Everything regarding the System universe can/should/must be stored in the SKB
- Whatever is included in the SKB must be considered an axiom or "ground truth".
- The SKB can be developed with different levels of accuracy and completeness.
  - The precision of the conceptual model affects system engineering processes.
  - For example:
    - A very generic conceptual model => Requirements define the system
    - A very detailed conceptual model => Requirements must fulfill the system model.

- If a SKB exists => Systems Engineering should be "REUSE-Intensive".

The System Knowledge Base (SKB) can be represented as an Ontology

System Knowledge Base:

Knowledge Organization in a System Conceptual Model



- The SCM must represent the System's universe
  - The (In)System
  - The Boundary (Interface between the system and the surroundings)
  - The Surroundings (Different environments)

System Knowledge Base:

System Conceptual Model (SCM)

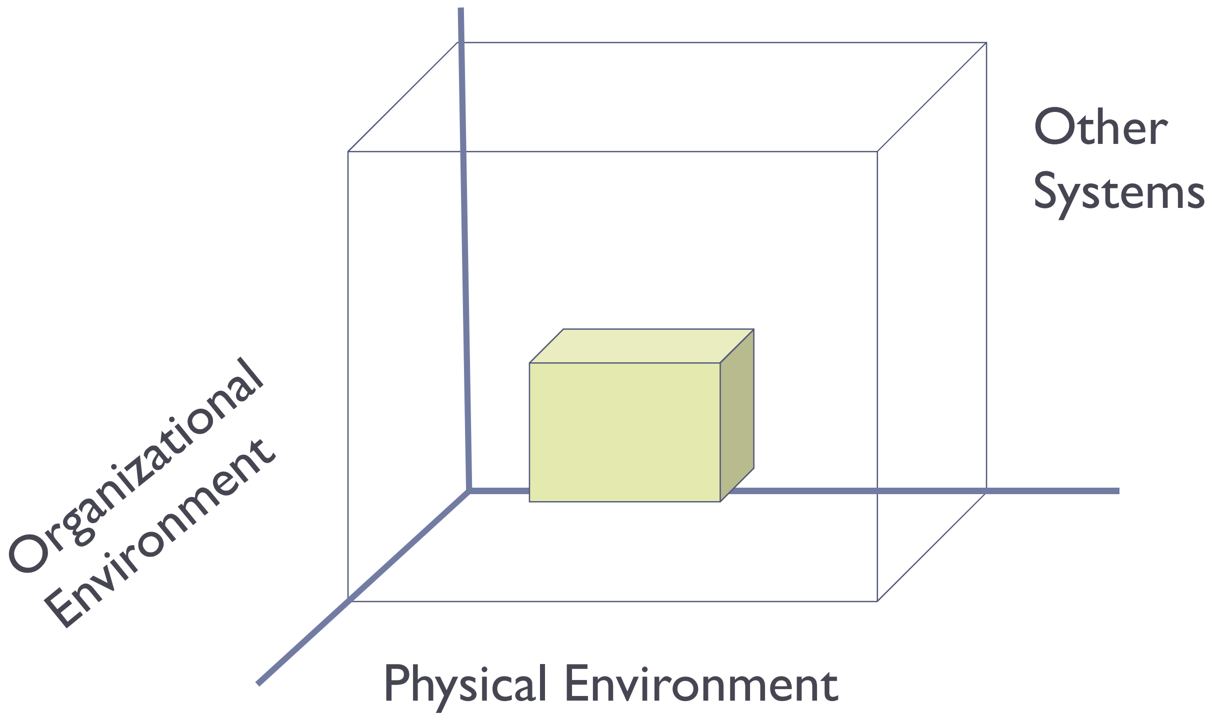
- In-System knowledge
  - Classification knowledge: (Abstraction management)
  - Functional knowledge: (Capacities management)
  - Structural (Logical and physical) knowledge: (Complexity management)
  - Dynamic Knowledge : (Collaboration management)
  - Conditions, Restrictions, Assumptions and Constraints (CRAC)
  - Properties
- Boundary knowledge:
  - Interface knowledge and management
- Environments knowledge
  - Physical Environment knowledge: ()
  - Organizational Environment knowledge: ()
  - Other Systems knowledge ()

- Everything at conceptual level!

System Knowledge Base (SKB)

Environments knowledge

- Multidimensional perspective

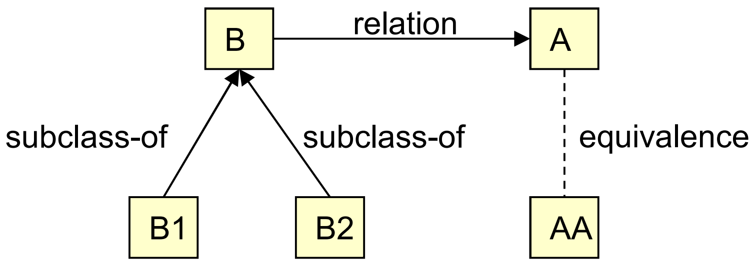


System Knowledge Base: Ontology

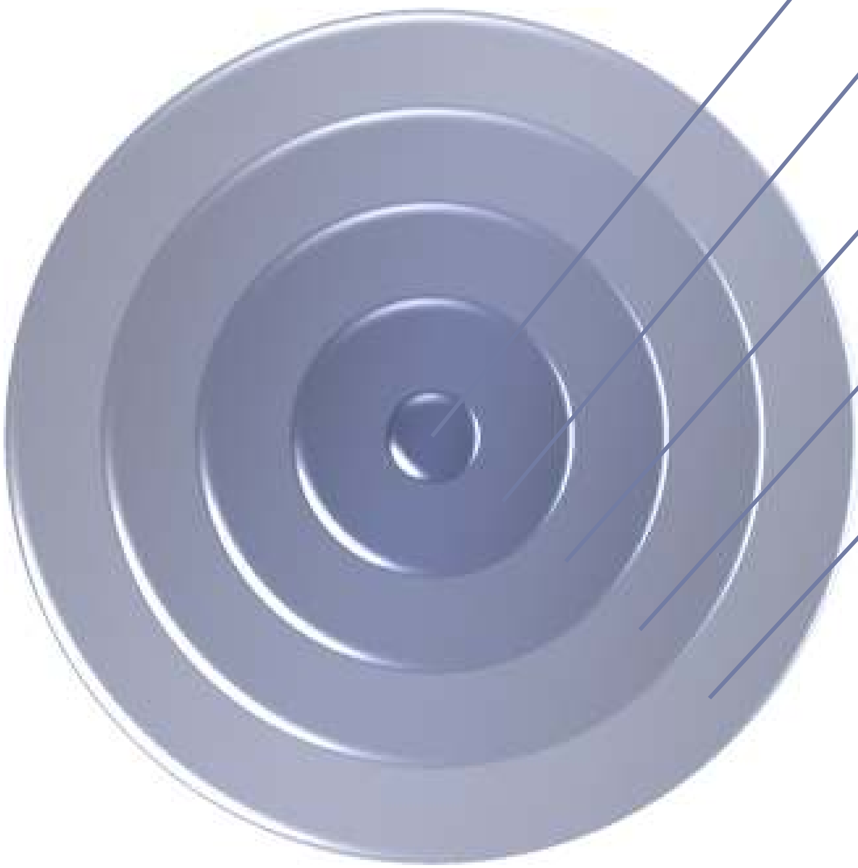
What is an ontology

- An ontology is a "specification of a conceptualization" [Gruber 1995].
  - Specification: formal and declarative representation
  - Conceptualization: abstract, simplified view of the world

- An ontology contains facts of the domain:
  - Concept: represents an entity in the domain with name and textual definition
  - Relation: labeled directed connection between concepts
  - Axiom: formal relationship between two concepts, e.g. subclass or equivalence



Practical Case: Ontology for Requirements Quality Mgmt.



Terminology layer: valid terms, forbidden terms, other NL terms, Syntactic clustering types, everything as concepts

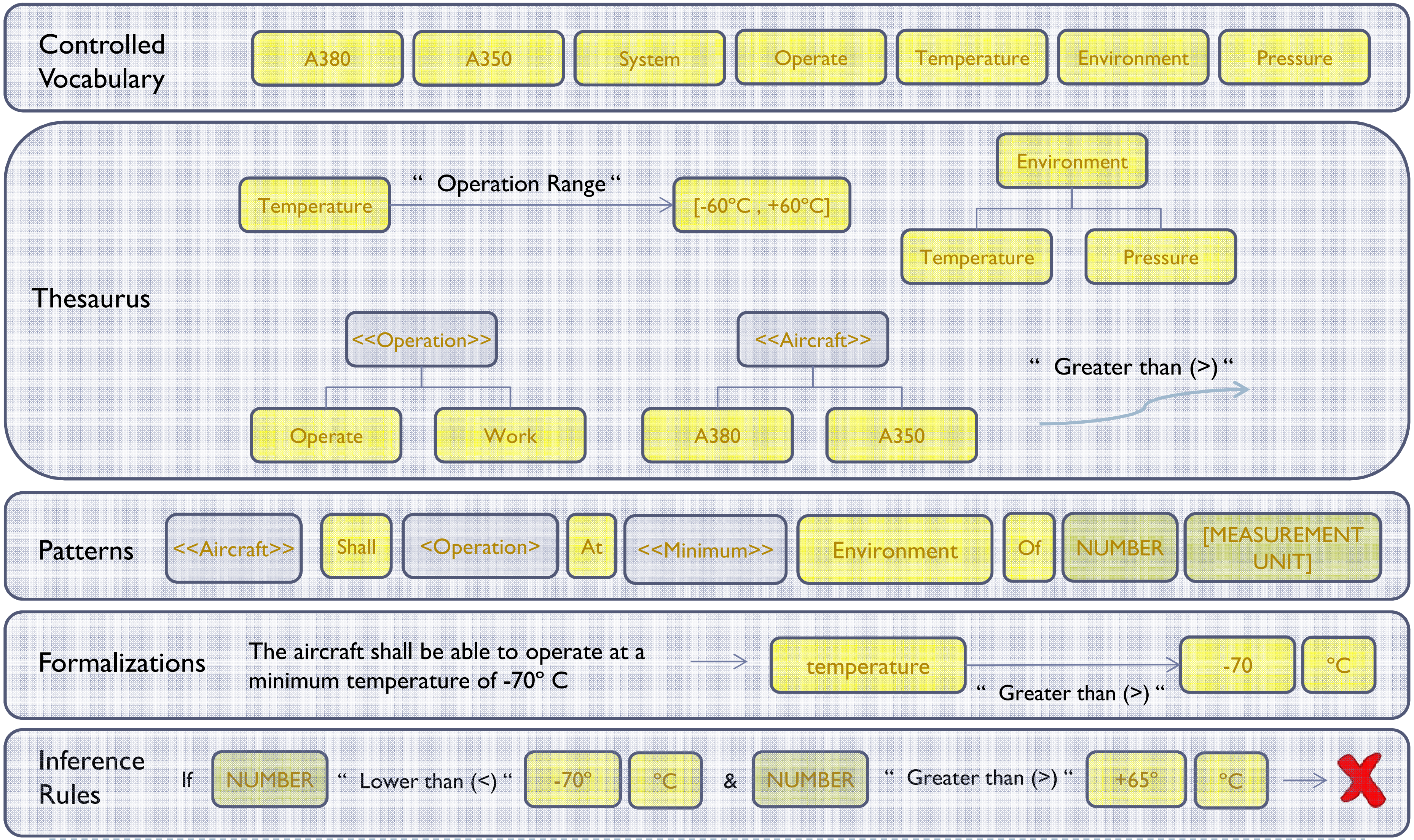
Thesaurus layer: relationships among concepts (hierarchies, associations, synonyms…), PBS, FBS, Etc.

Patterns layer: Matching Patterns

Formalization layer: **S**emantic formalization

Inference layer: for decision making (e.g. consistency, completeness)

Ontology : Example



Conclusions

- Knowledge is necessary when trying to solve complex problems in SE
- This knowledge can/must be stored and represented inside an ontology and used for improving Systems Engineering practices
- If knowledge is good enough => it should be reusable

Knowledge Centric Systems Engineering

Dr. Juan Llorens

Technical Director - Asociación Española de Ingeniería de Sistemas (AEIS) – INCOSE
Professor at Informatics Department - Universidad Carlos III de Madrid – Spain
CTO – The Reuse Company (TRC)

Juan.Llorens@uc3m.es

http://www.linkedin.com/pub/juan-llorens/b/857/632

https://www.researchgate.net/profile/Juan_Llorens/

South European Systems Engineering Tour (SESE 2014)
Zurich 1st September - Paris 23rd September - Madrid 24th September 2014