PMR 5020

Modelagem do Projeto de Sistemas

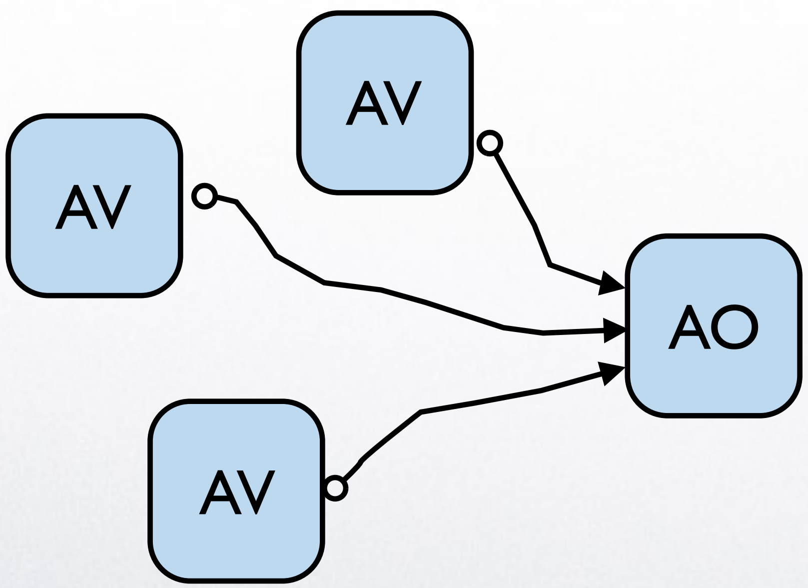
Aula 11: MBSE, features and methods

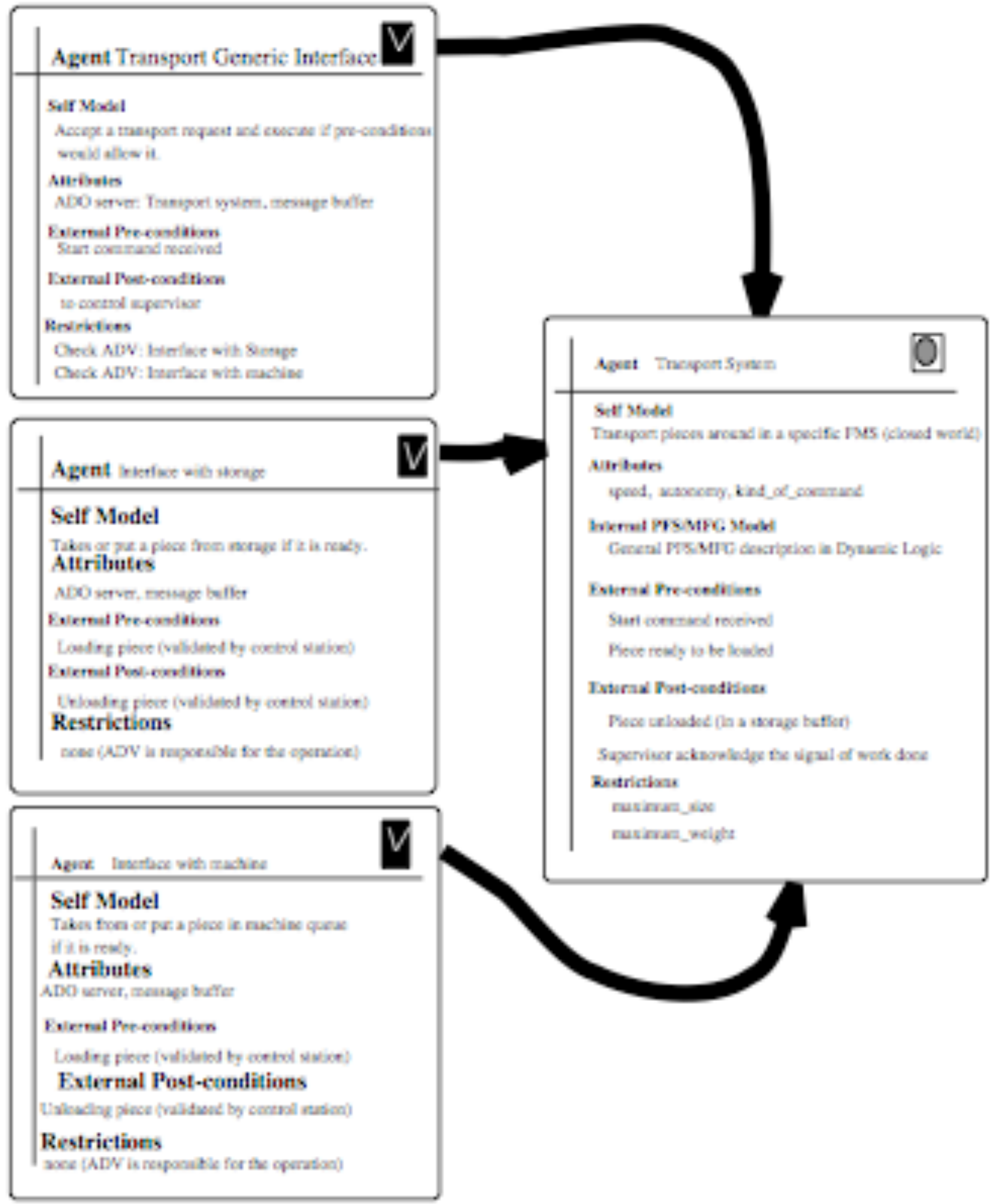
Prof. José Reinaldo Silva
reinaldo@poli.usp.br

The big challenge in today's class is to answer the question: how could be scale the concepts discussed so far to large and complex (automated) systems, indeed, how could we apply all this knowledge - claimed to be effective - to real life projects?

→ Model Driven Engineering

Object interface-service relationship





INCOSE MBSE Initiative

Survey of Model-Based Systems Engineering (MBSE) Methodologies

Jeff A. Estefan
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California, U.S.A.
Jeffrey.A.Estefan@jpl.nasa.gov

1. Introduction

1.1 Purpose

The purpose of this report is to provide a cursory description of some of the leading Model-Based Systems Engineering (MBSE) methodologies used in industry today. It is intended that the material described herein provides a direct response to the INCOSE MBSE Roadmap element for a "Catalog of MBSE lifecycle methodologies" [1].

In this report, a *methodology* is defined as a collection of related processes, methods, and

Principal Engineer at NASA's
Jet Propulsion Lab

Survey of Candidate Model-Based Engineering (MBSE) Methodologies, Rev. B, May 23, 2008 - INCOSE MBSE Initiative

Basic Definitions: process

- A Process (P) is a logical sequence of tasks performed to achieve a particular objective. A process defines “WHAT” is to be done, without specifying “HOW” each task is performed. The structure of a process provides several levels of aggregation to allow analysis and definition to be done at various levels of detail to support different decision-making needs.

Basic Definitions: methods

- A Method (M) consists of techniques for performing a task, in other words, it defines the "HOW" of each task. (In this context, the words "method," "technique," "practice," and "procedure" are often used interchangeably.) At any level, process tasks are performed using methods. However, each method is also a process itself, with a sequence of tasks to be performed for that particular method. In other words, the "HOW" at one level of abstraction becomes the "WHAT" at the next lower level.

Basic Definitions: tools

A Tool (T) is an instrument that, when applied to a particular method, can enhance the efficiency of the task; provided it is applied properly and by somebody with proper skills and training. The purpose of a tool should be to facilitate the accomplishment of the "HOWs." In a broader sense, a tool enhances the "WHAT" and the "HOW." Most tools used to support systems engineering are computer- or software-based, which also known as Computer Aided Engineering (CAE) tools.

Methodology is something else...

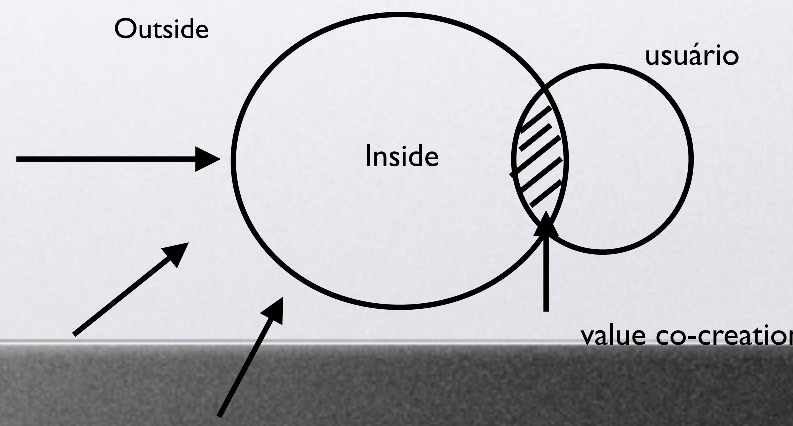
Based on these definitions, a methodology can be defined as a collection of related processes, methods, and tools. A methodology is essentially a "recipe" and can be thought of as the application of related processes, methods, and tools to a class of problems that all have something in common.

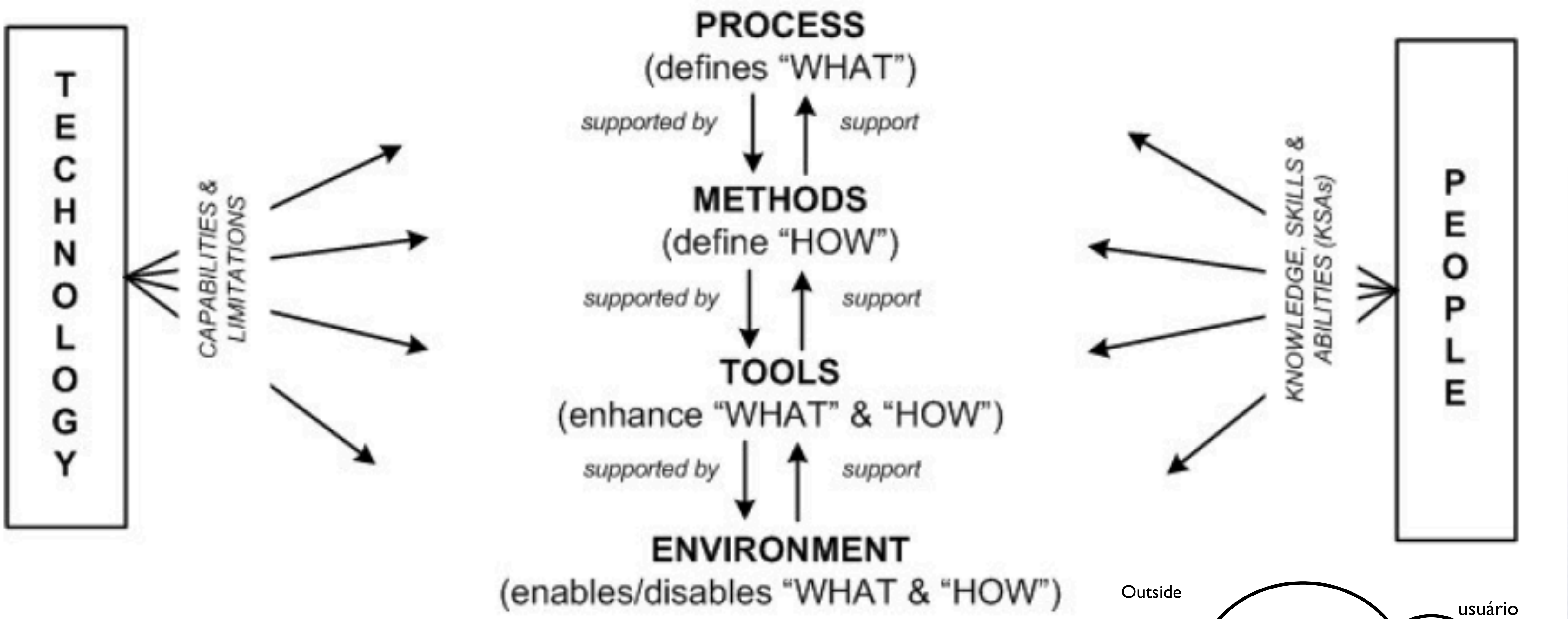
Bloomberg, Jason and Ronald Schmelzer, Service Orient or Be Doomed!, John Wiley & Sons: Hoboken, New Jersey, 2006.

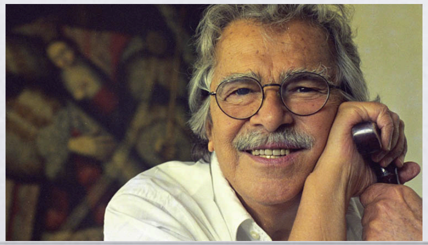
Everything works in an environment...

Associated with the above definitions for process, methods (and methodology), and tools is environment. An Environment (E) consists of the surroundings, the external objects, conditions, or factors that influence the actions of an object, individual person or group. These conditions can be social, cultural, personal, physical, organizational, or functional.

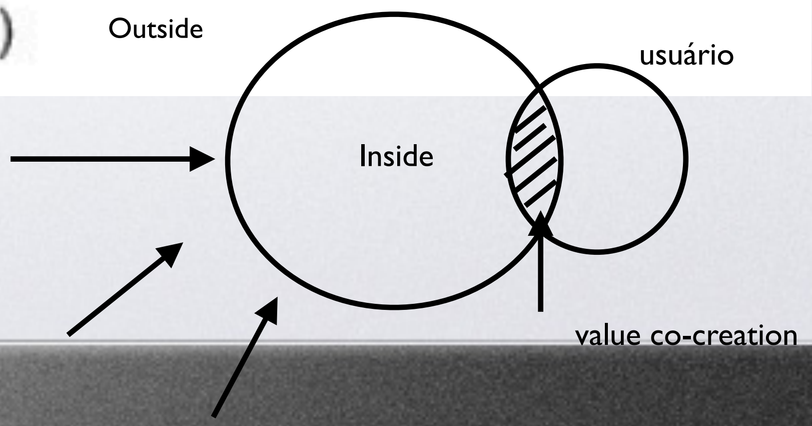
The purpose of a project environment should be to integrate and support the use of the tools and methods used on that project. An environment thus enables (or disables) the "WHAT" and the "HOW."

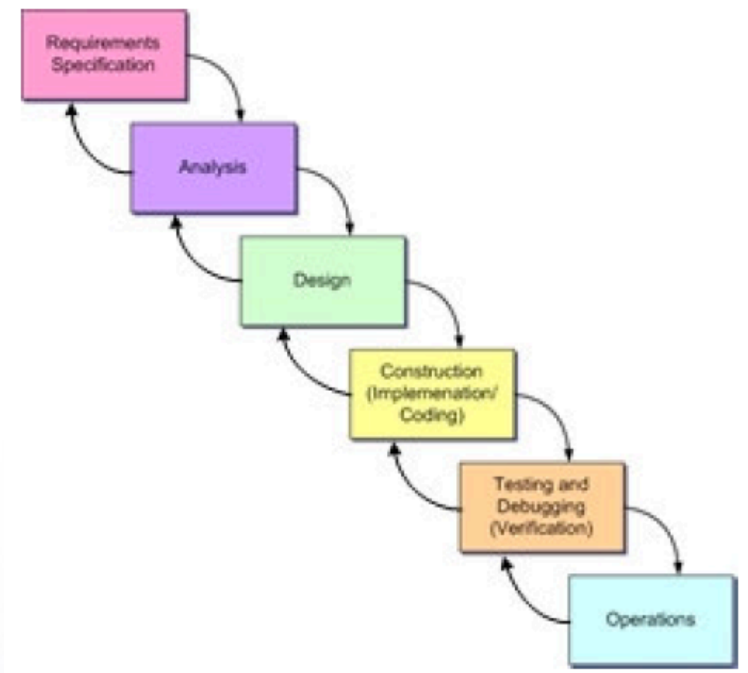


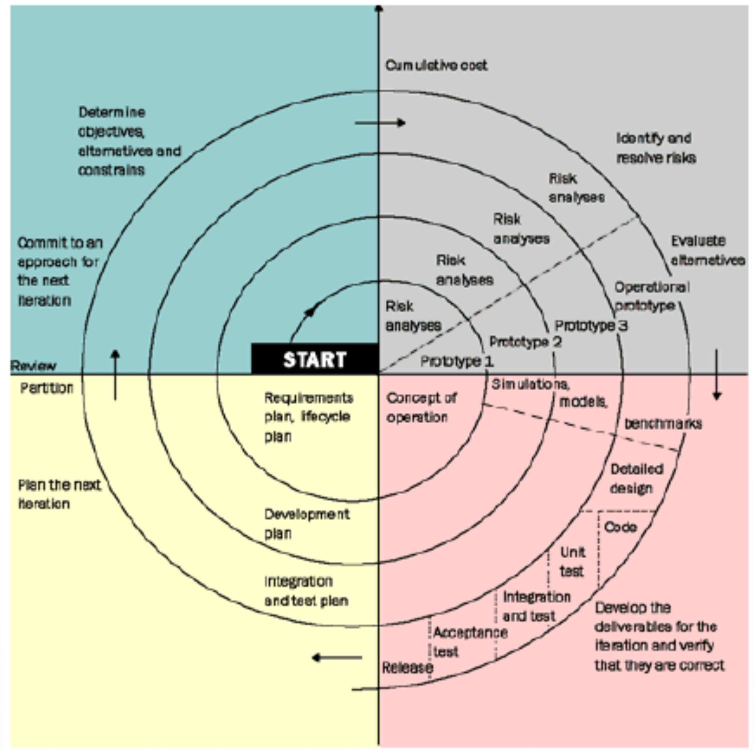


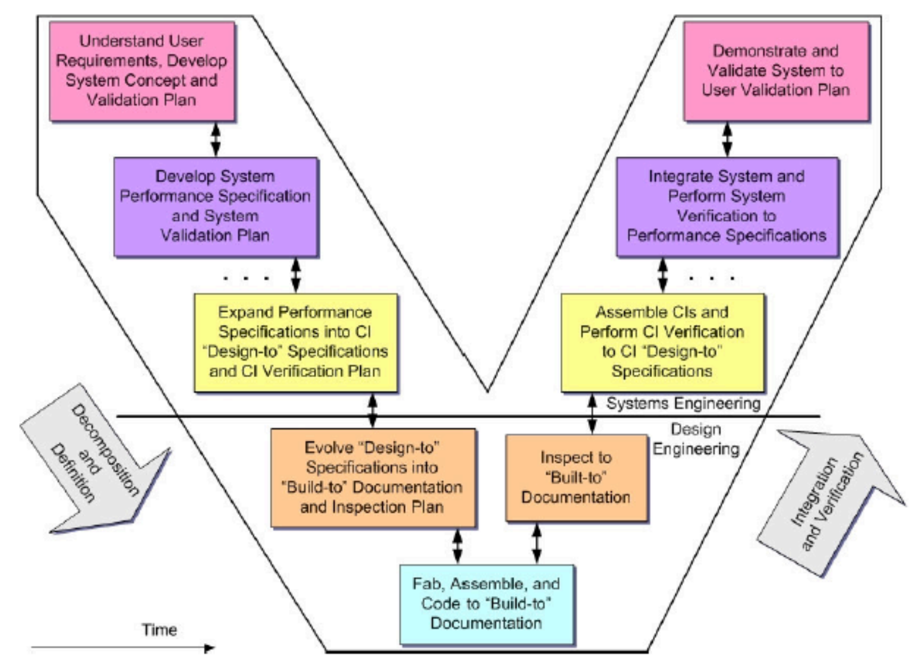


Darcy Ribeiro



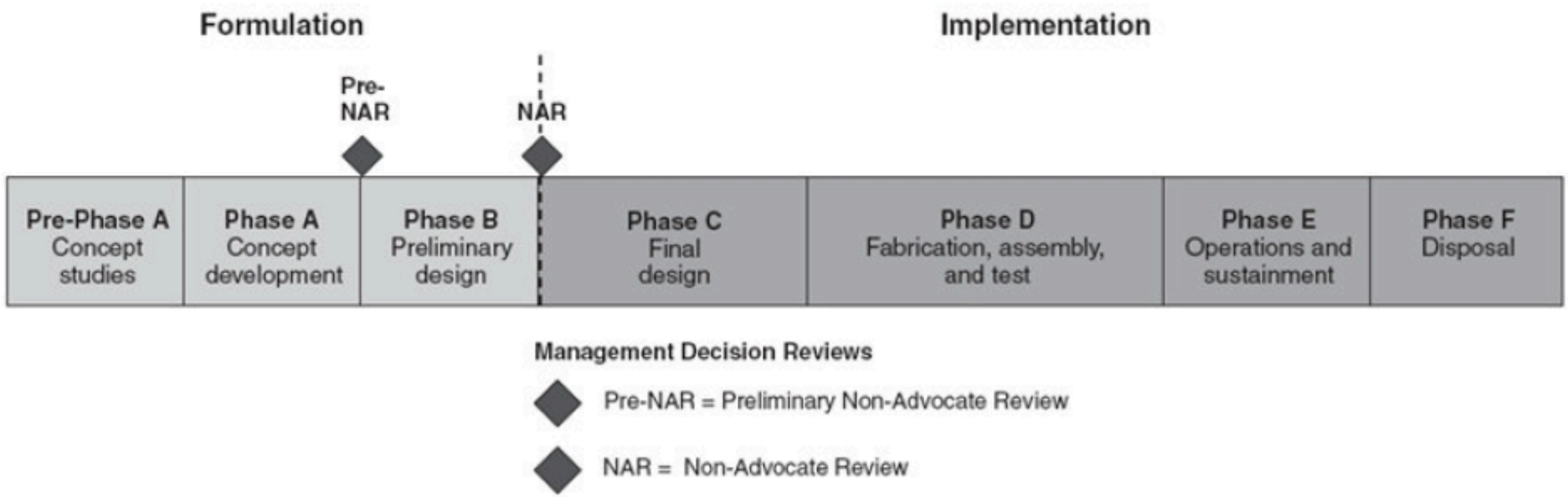






Seminal Lifecycle Development Models: (a) Waterfall, (b) Spiral, (c) "Vee"

NASA Project Lifecycle



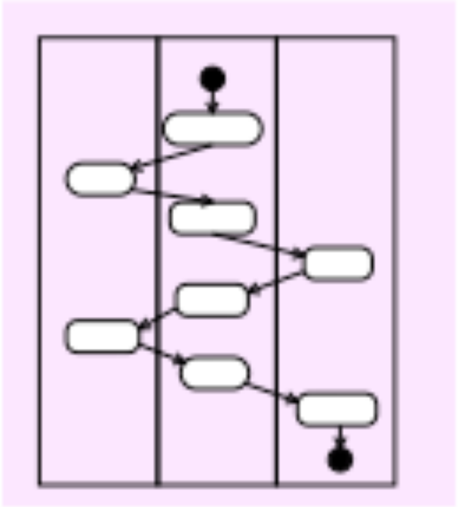
SE Practices for Describing Systems

Past

Text

- Specifications
- Interface requirements
- System design
- Test plans
- Analysis & Trade-off

Future



Model

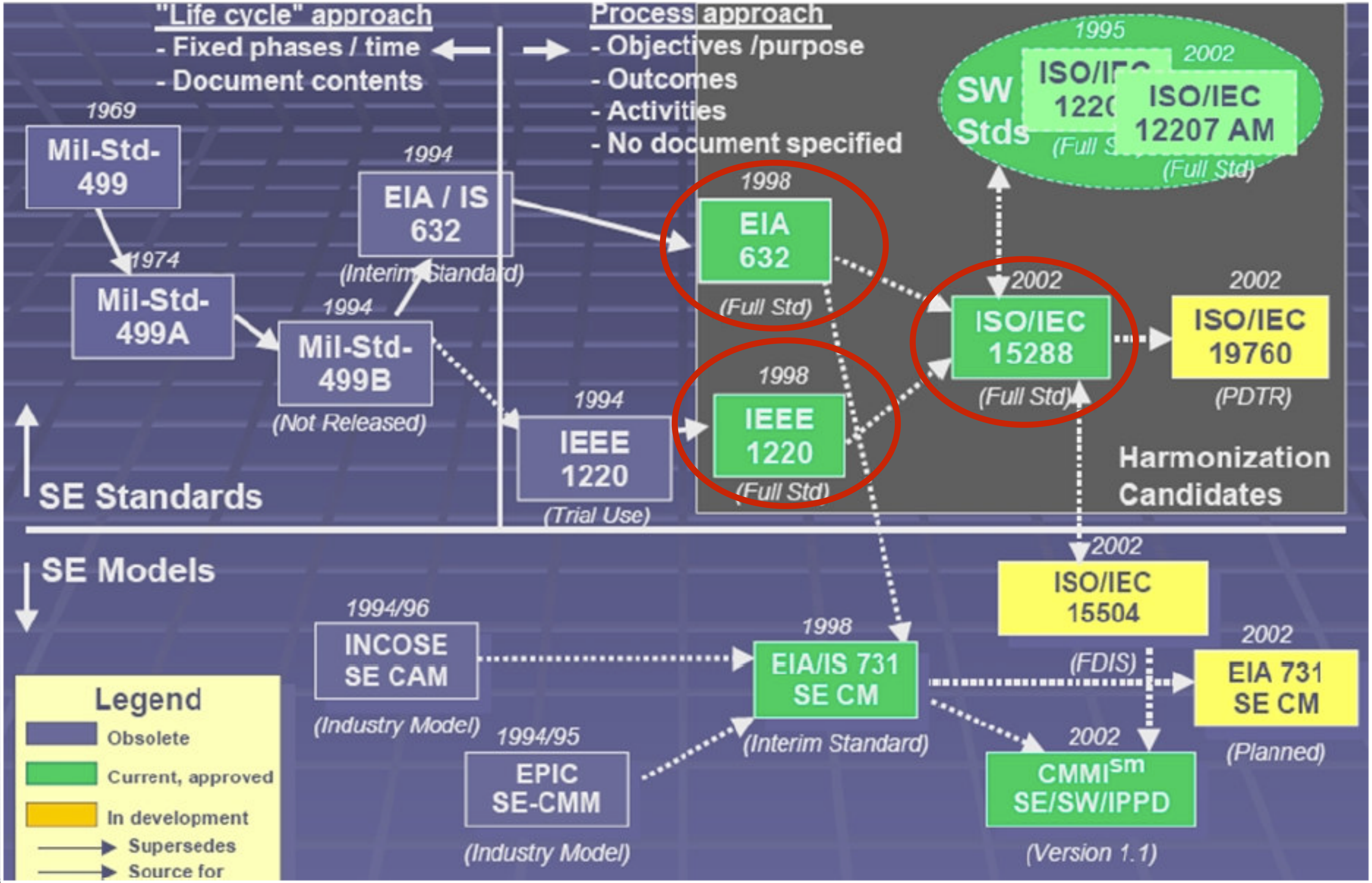
The System Engineering Approach

" Systems engineering (SE) is an interdisciplinary approach and means to enable the realization of successful systems. Successful systems must satisfy the needs of their customers, users and other stakeholders.

In the broad community, the term system “system,” may mean a collection of technical, natural or social elements, or a combination of all three.

SEBoK-2018 v. 1.9.1 release 16 October 2018

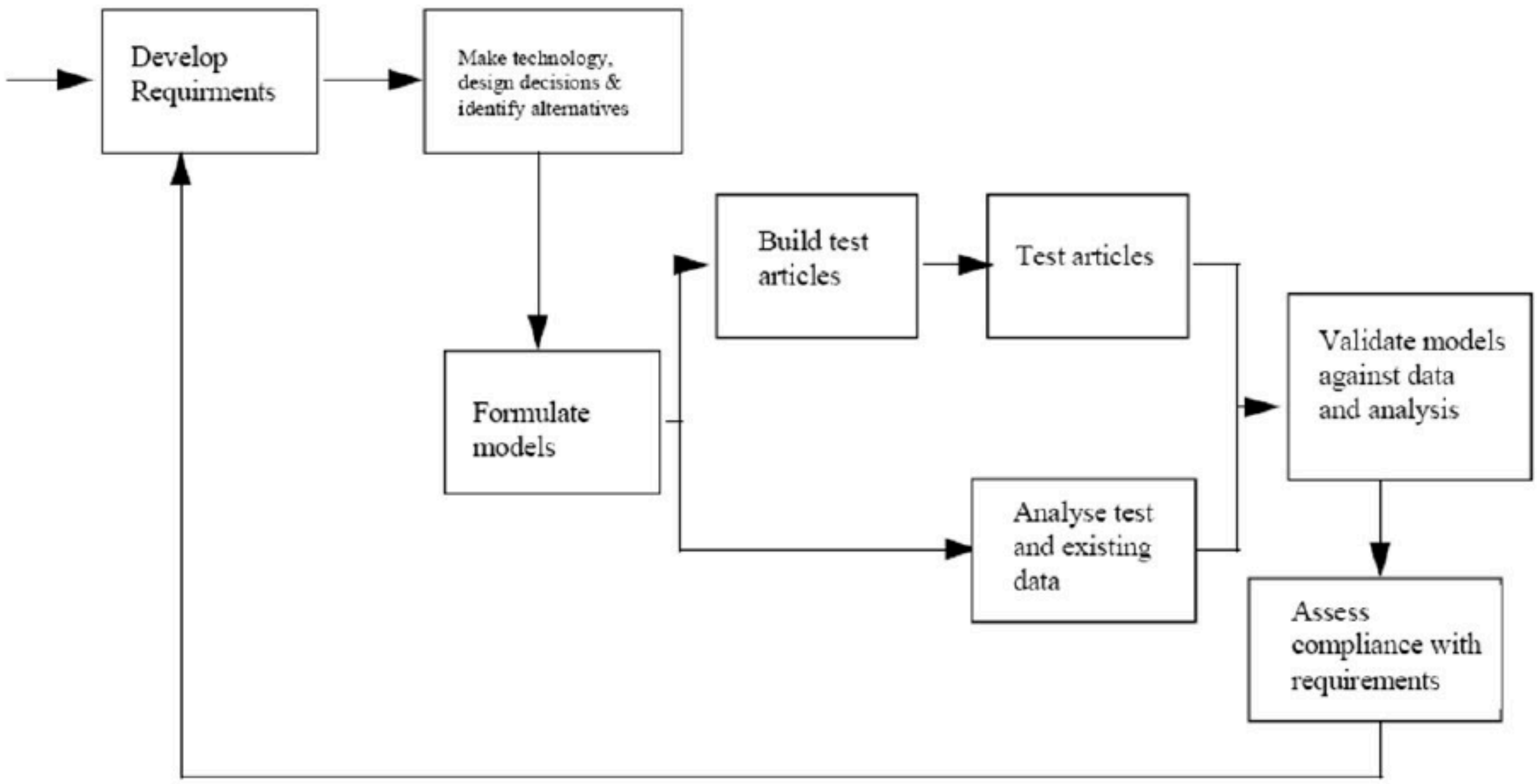
https://www.sebokwiki.org/wiki/Download_SEBoK_PDF

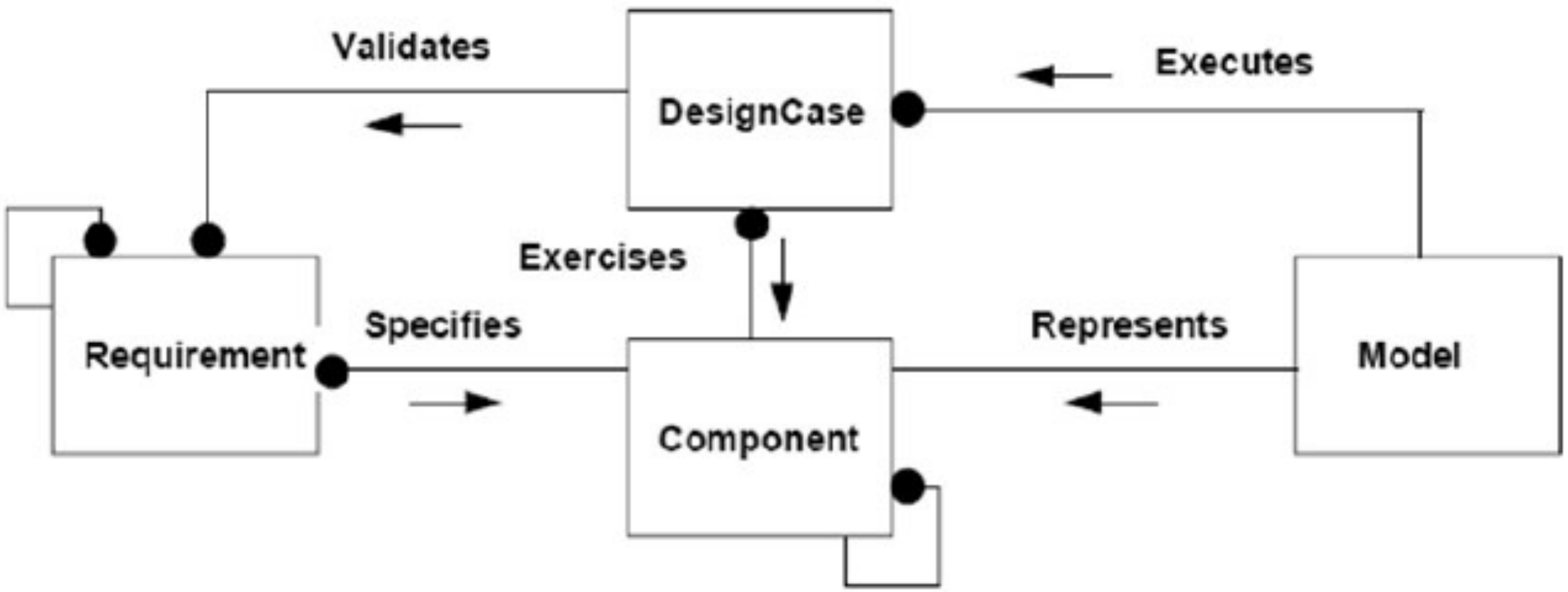

"Life cycle" approach
- Fixed phases / time
- Document contents
Process approach
- Objectives /purpose
- Outcomes
- Activities
- No document specified
1969
Mil-Std-499
1974
Mil-Std-499A
1994
Mil-Std-499B
(Not Released)
1994
EIA / IS 632
(Interim Standard)
1994
IEEE 1220
(Trial Use)
1998
EIA 632
(Full Std)
1998
IEEE 1220
(Full Std)
SW Stds
1995
2002
ISO/IEC 12207 AM
(Full Std)
2002
ISO/IEC 15288
(Full Std)
2002
ISO/IEC 19760
(PDTR)
Harmonization Candidates
SE Standards
SE Models
1994/96
INCOSE SE CAM
(Industry Model)
1994/95
EPIC SE-CMM
(Industry Model)
1998
EIA/IS 731 SE CM
(Interim Standard)
2002
ISO/IEC 15504
(FDIS)
2002
EIA 731 SE CM
(Planned)
2002
CMMI sm SE/SW/IPPD
(Version 1.1)
Legend
Obsolete
Current, approved
In development
Supersedes
Source for

The *purpose* of each major SE process model standard can be summarized as follows [17]:

- ISO/IEC 15288 – Establish a common framework for describing the lifecycle of systems.
- ANSI/EIA 632 – Provide an integrated set of fundamental processes to aid a developer in the engineering or re-engineering of a system.
- IEEE 1220 – Provide a standard for managing a system.

Model Driven System Design





- Requirements specify Components
- Requirements may be decomposed into other Requirements
- Components may be decomposed into other Components
- Design Alternates satisfy Requirements
- Design Alternates represent Components
- Models execute Design Alternates
- Models represent Components

Isso visto pela ótima das boas práticas de design, mas...

... e a busca por uma Teoria Geral do Design

A discussão sobre a formalização do processo de design começou em 1981com a proposta de Hiroyuki Yoshikawa, e foi logo em seguida ampliada por Tetsuo Tomiyama, seu orientado. A polemica perdura até hoje e varia de alegações ao arcabouço teórico, à abordagem conceitual, até a perspectiva de aplicação.

Hiroyuki Yoshikawa

Tetsuo Tomiyama

Fundamentação matemática para o MBSE

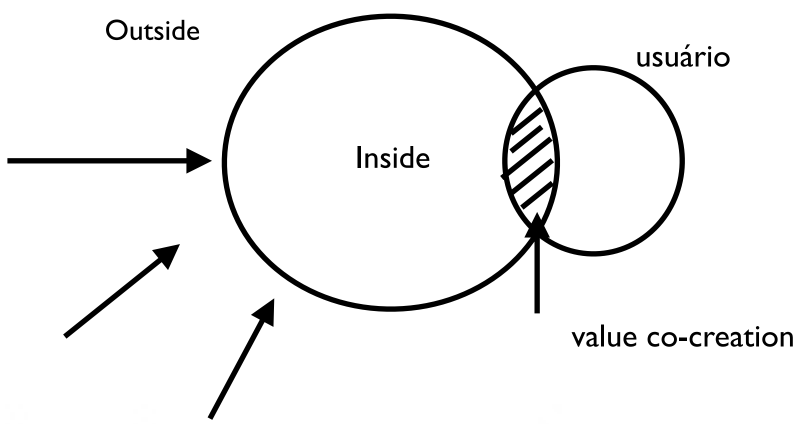
A. Wayne Wymore (T3SD)

1927-2011

Wymore, A. Wayne, A Mathematical Theory of Systems Engineering: The Elements , John Wiley & Sons: New York, NY, 1967.

Wymore, A. Wayne, Model-Based Systems Engineering , CRC Press, Inc.: Boca Raton, FL, 1993.

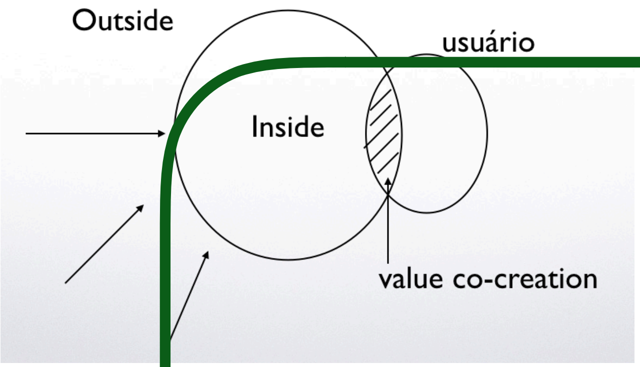
Wymore, A. Wayne, “Contributions to the Mathematical Foundations of Systems Science and Systems Engineering,” Systems Movement: Autobiographical Retrospectives, The University of Arizona, Tucson, AZ, 2004.

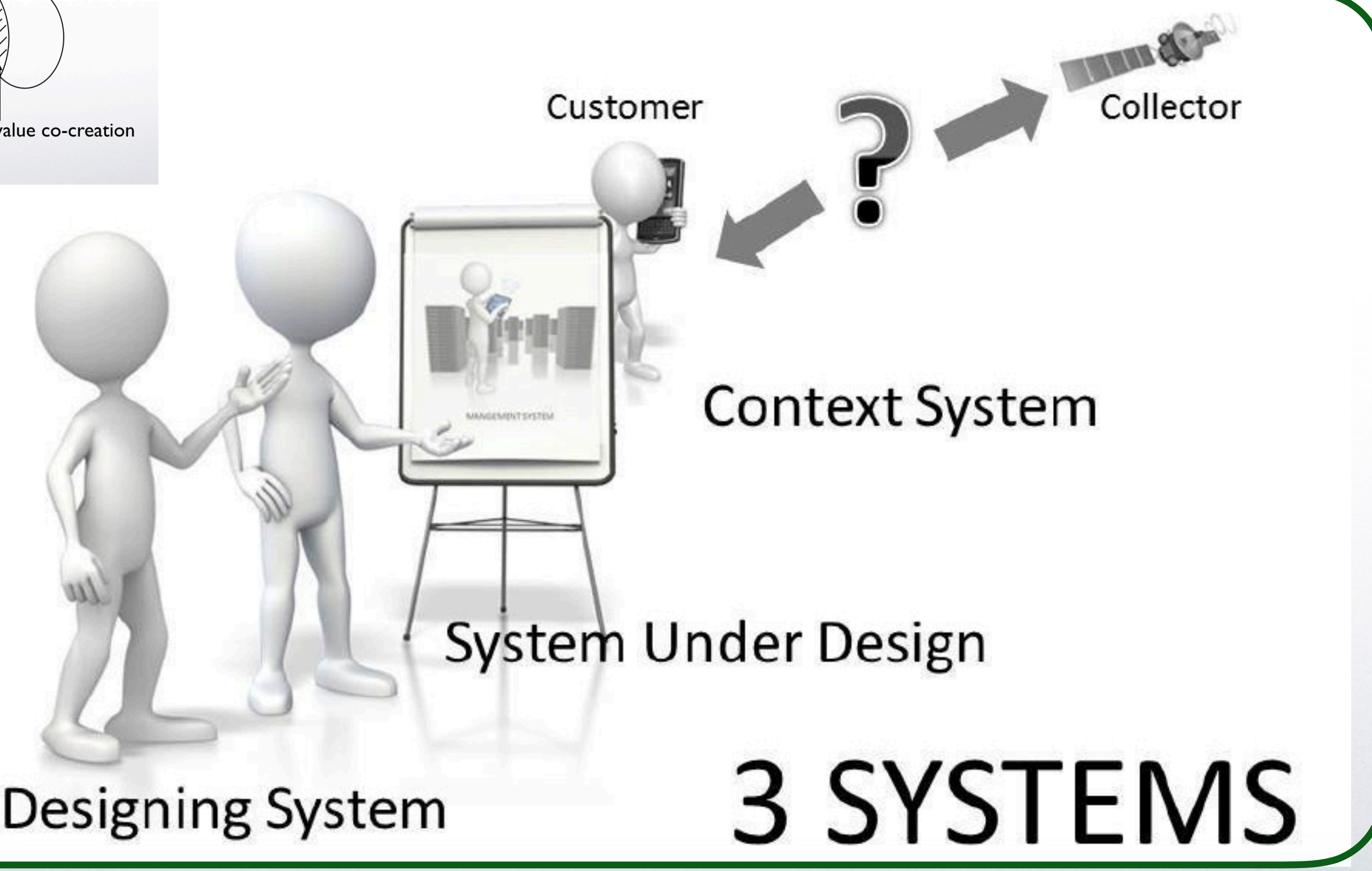


The basis of system theory and systems engineering, according to Wymore, is modeling, and the concept of “system” is human interpretation via senses (i.e., a mental model). A system model is a description that separates the perceived universe into two parts, the part “inside” the system and the part “outside” the system. From the “outside” the system receives inputs. To the “outside” the system delivers outputs. The “inside” of the system is described, initially, as states. The state of the system at any time is a function of its state at a previous time and the intervening inputs (including “noise”). System designs are system models. When modelers describe some part of reality as a “real” system, it means that their system mode “adequately” represents the reality. For the purpose of accurate communication, mathematical definitions of various classes of system models are postulated.

Wymore proposed a tricotyledon theory of system design as he named the specific mathematical system theory he developed to facilitate the process of system design. The proposed three basic spaces of system design are described shortly. .

Development system





A T3SD (tricotyledon theory of system design)

A system is a tuple $S=[\Sigma,\ \tau,\ \mathfrak{I}|\tau,\ \overline{\{\mathfrak{I}|\tau\}},\ \sigma(t,\ \mathfrak{I}),\ \Theta,\ \psi(\Sigma,\ \Theta)]$, where:

- Σ is a set of states;
- τ is a system time scale;
- $\mathfrak{I}|\tau$ is a set of inputs;
- $\overline{\{\mathfrak{I}|\tau\}}$ is a set of input trajetories;
- $\sigma(t,\ \mathfrak{I})$ is a state function, matching initial and output states;
- Θ is a set of outputs;
- $\psi(\Sigma,\ \Theta)$ is a transfer funcion that maps output states and outputs;

A key aspect that is elaborated on in the second part of the MBSE book is Wymore's introduction of T3SD and identification of the six core categories of system design requirements (SDR), which he defines as follows:

SDR = (IOR, TYR, PR, CR, TR, STR) where

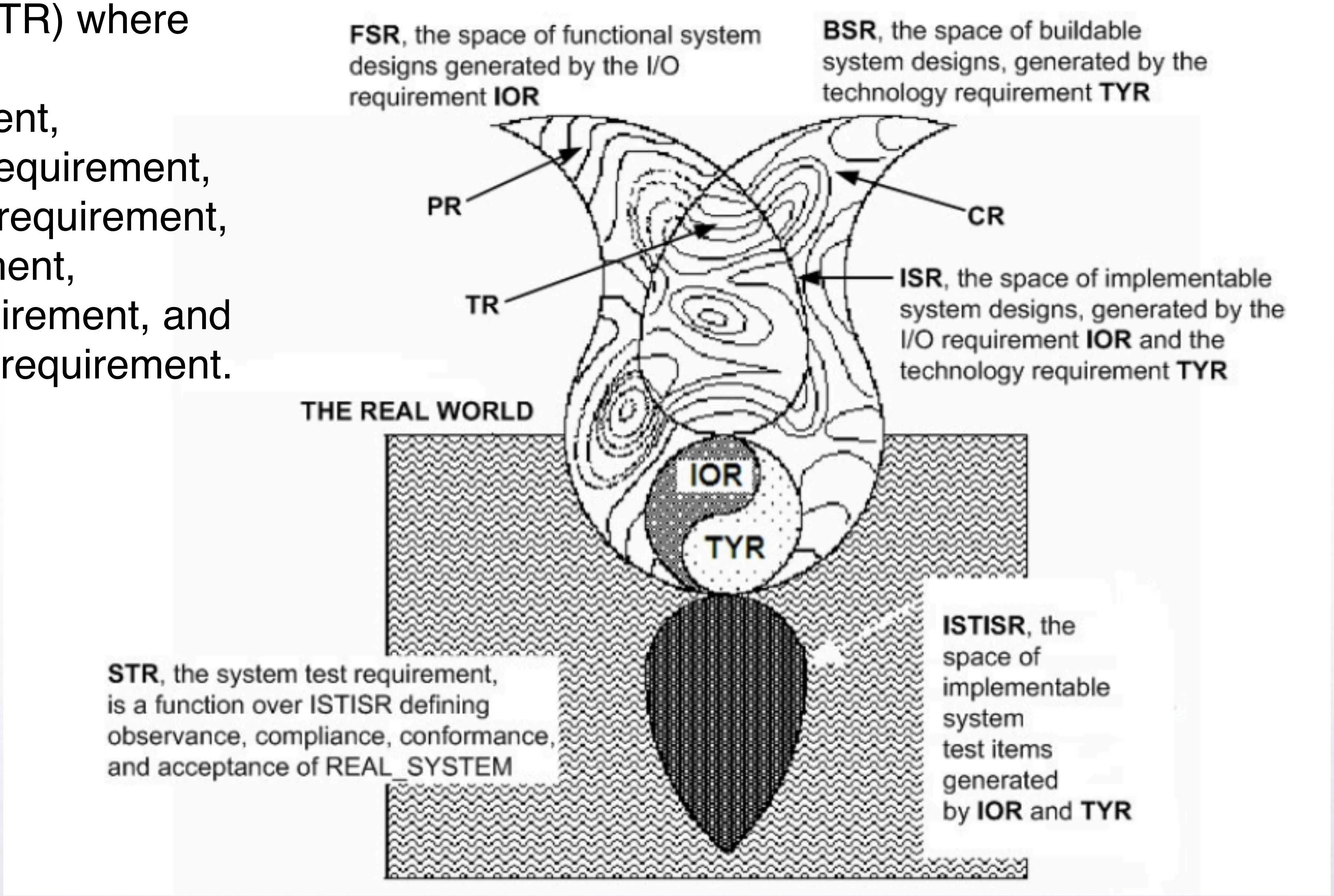
i) IOR is the I/O requirement,
ii) TYR is the technology requirement,
iii) PR is the performance requirement,
iv) CR is the cost requirement,
v) TR is the trade-off requirement, and
vi) STR is the system test requirement.

According to Wymore, across all projects, the system has to verify that these six conditions are met. This has direct relevance to the discipline of systems engineering. Systems engineering considers as many alternative implementable system designs as possible and selects the best with respect to the tradeoff requirement, finding the implementable systems design that is optimum with respect to the tradeoff requirement and most likely to pass the system test, if possible.

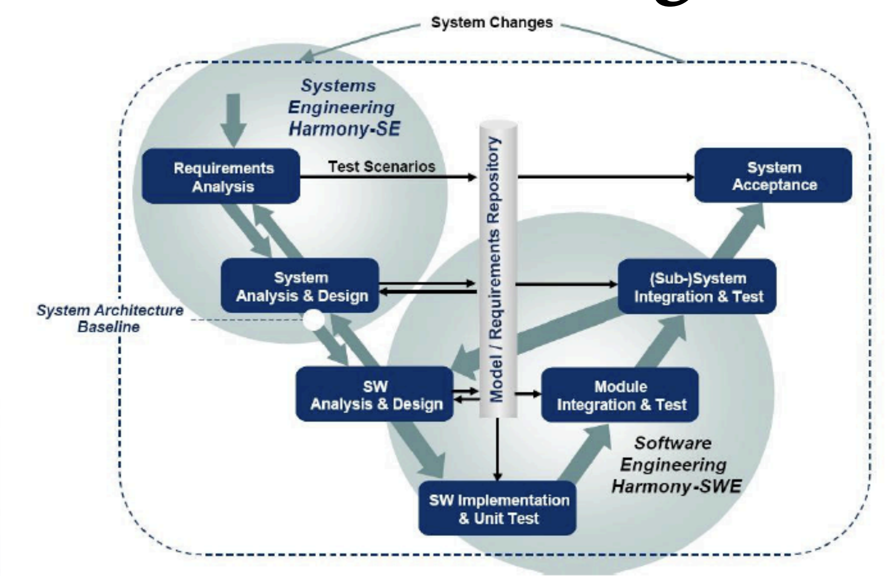
No Silver bullet

SDR = (IOR, TYR, PR, CR, TR, STR) where

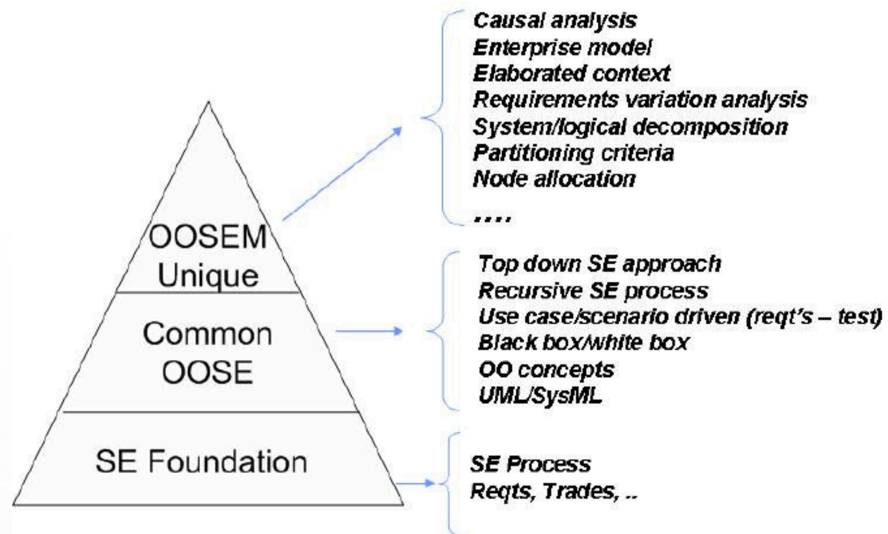
i) IOR is the I/O requirement,
ii) TYR is the technology requirement,
iii) PR is the performance requirement,
iv) CR is the cost requirement,
v) TR is the trade-off requirement, and
vi) STR is the system test requirement.



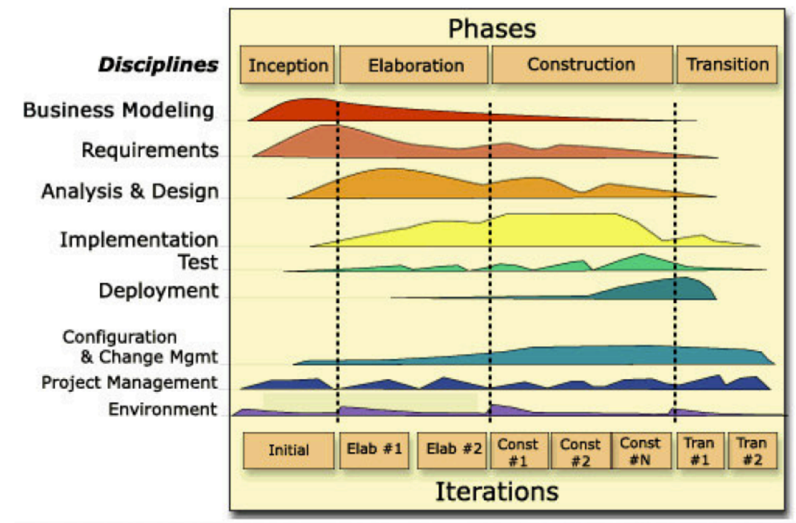
MBSE Methodologies



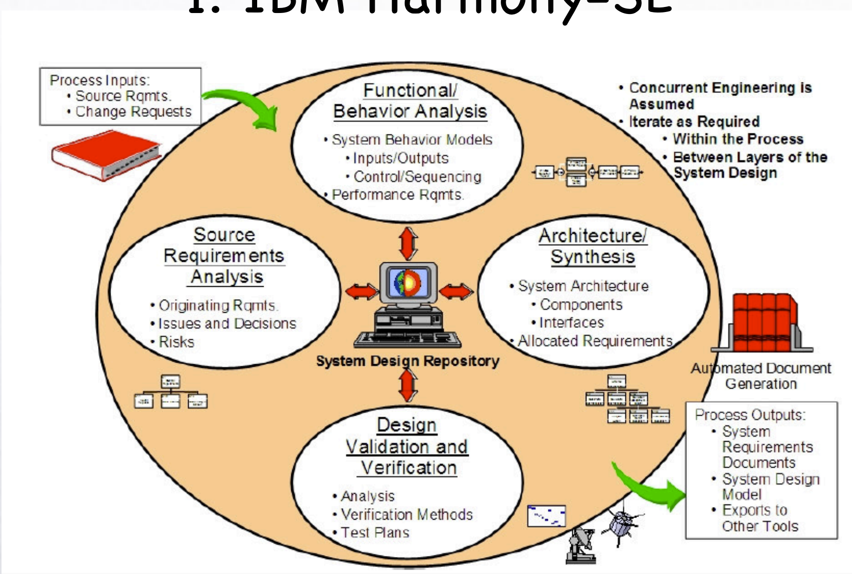
1. IBM Harmony-SE



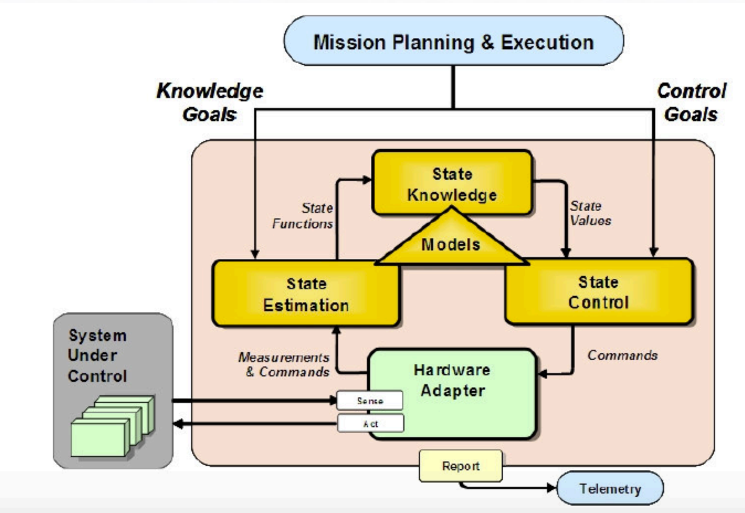
2. OOSEM



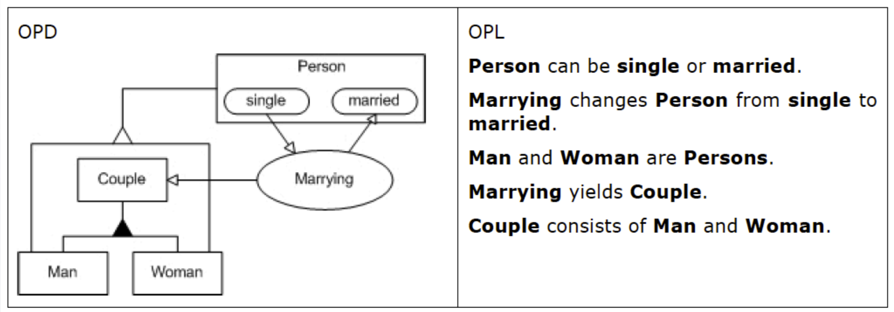
3. RUP



4. Vitech

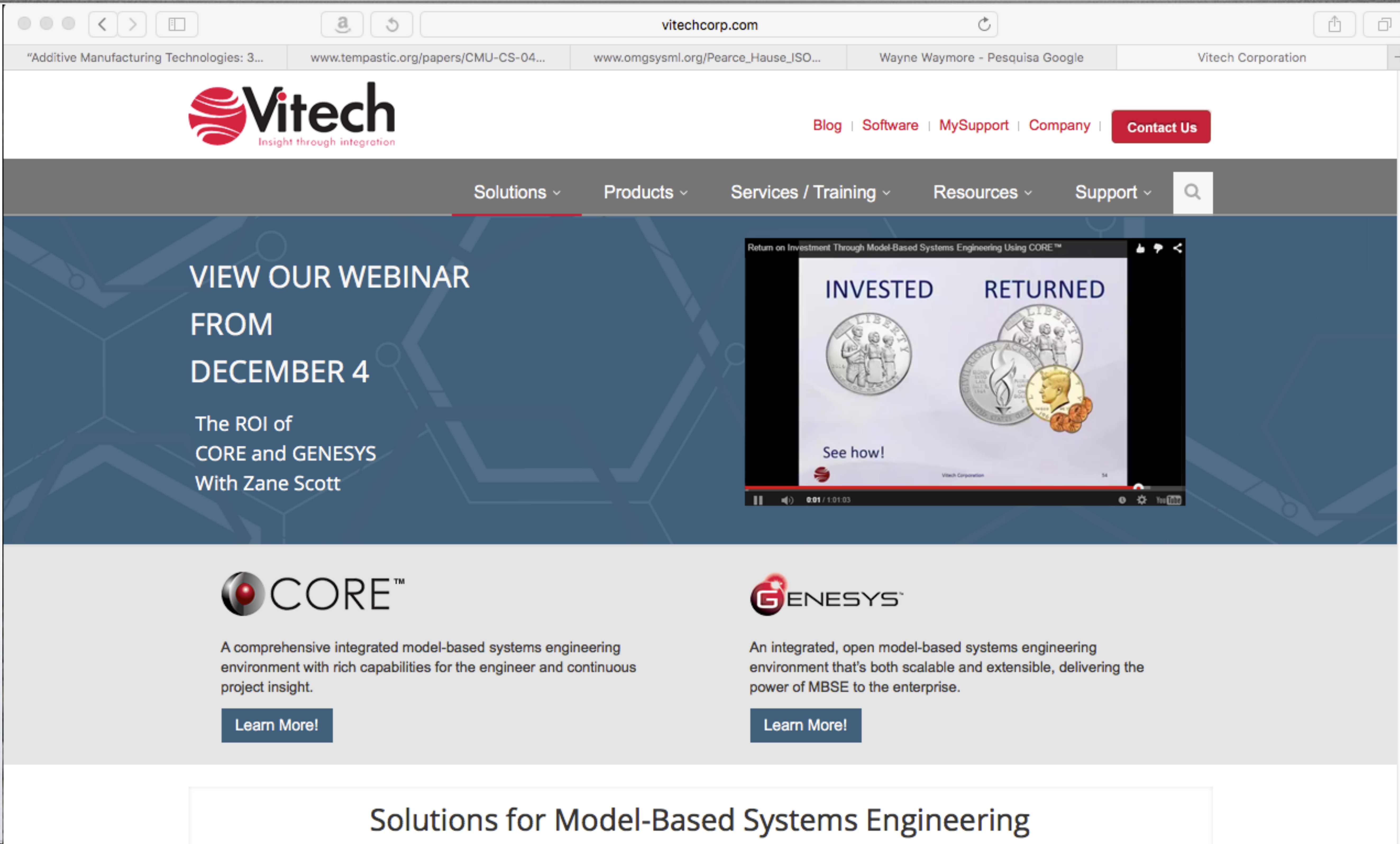


5. JPL

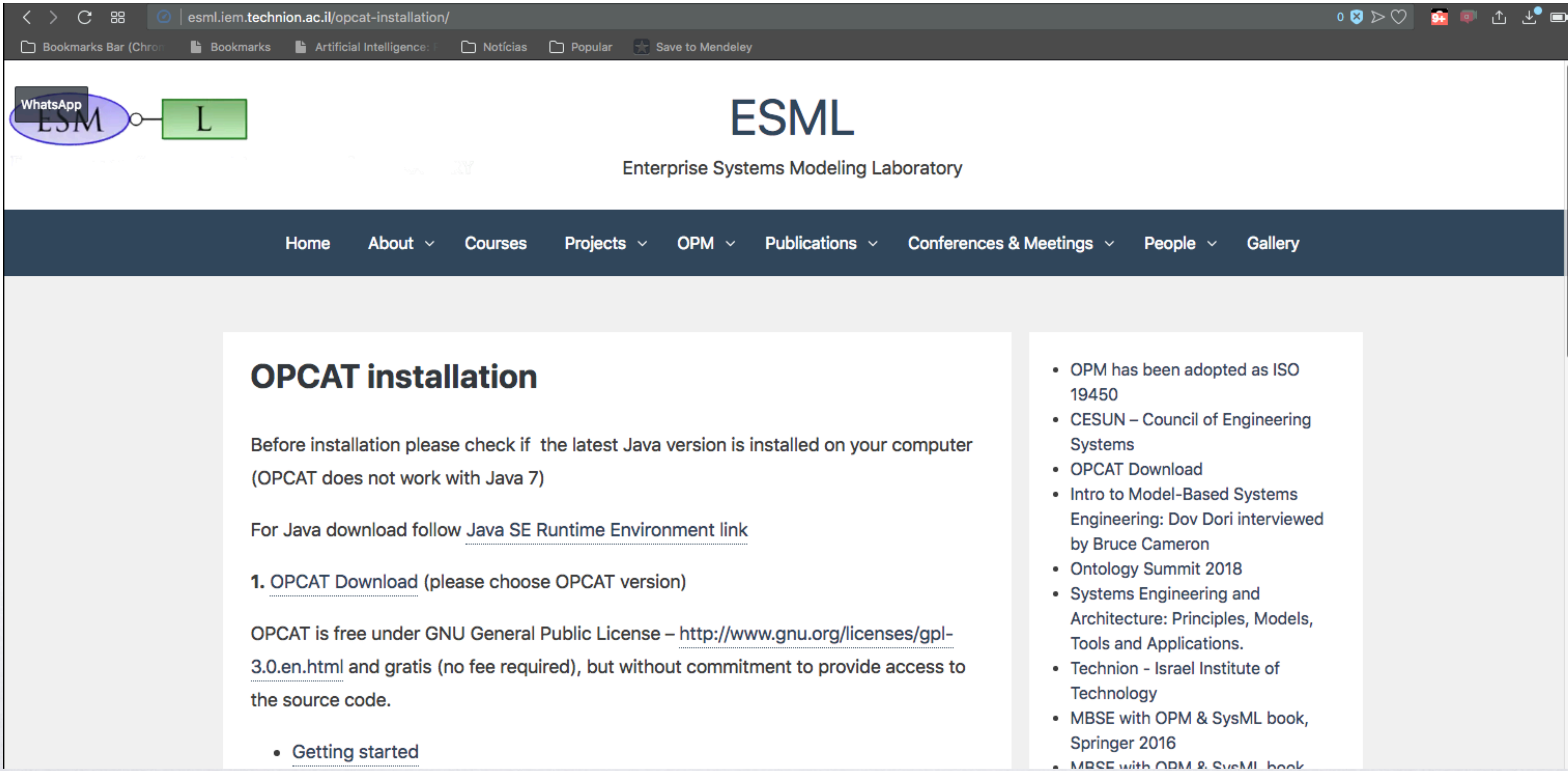


6. OPM (Object-Process Methodology)
ISO/PAS 19450 (Dovi Dori)

Method.	Proponent	Tool	Reference
IBM Harmony-SE	IBM	not specific	IBM Rhapsody
OOSEM	INCOSE	not specific	INCOSE/OMG
IBM Rational	IBM	RUP	IBM Rational
Vitech	Vitech	CORE	www.vitech.com
JPL	JPL	State DB	JPL Caltech
OPM	Dov Dori (1995)	OPCAT	www.opcat.com ISO/PAS 19450


vitechcorp.com
"Additive Manufacturing Technologies: 3...
www.tempastic.org/papers/CMU-CS-04...
www.omgsysml.org/Pearce_Hause_ISO...
Wayne Waymore - Pesquisa Google
Vitech Corporation
Vitech
Insight through integration
Blog | Software | MySupport | Company |
Contact Us
Solutions
Products
Services / Training
Resources
Support
VIEW OUR WEBINAR
FROM
DECEMBER 4
The ROI of
CORE and GENESYS
With Zane Scott
Return on Investment Through Model-Based Systems Engineering Using CORE™
INVESTED
RETURNED
See how!
0:01 / 1:01:03
CORE™
A comprehensive integrated model-based systems engineering environment with rich capabilities for the engineer and continuous project insight.
Learn More!
GENESYS
An integrated, open model-based systems engineering environment that's both scalable and extensible, delivering the power of MBSE to the enterprise.
Learn More!
Solutions for Model-Based Systems Engineering

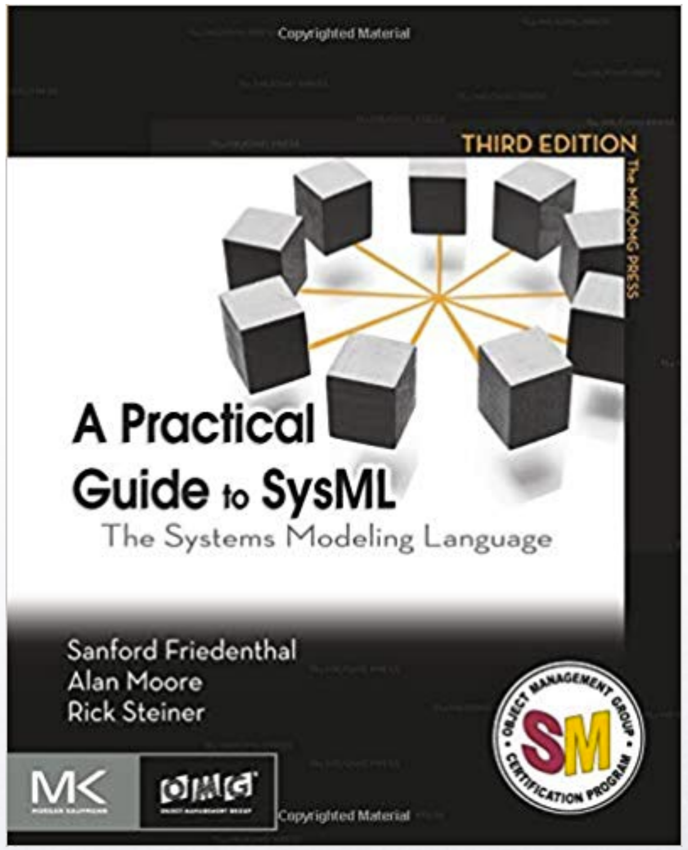
http://esml.iem.technion.ac.il/opcat-installation/



OOSEM - Object-oriented System Engineering Method

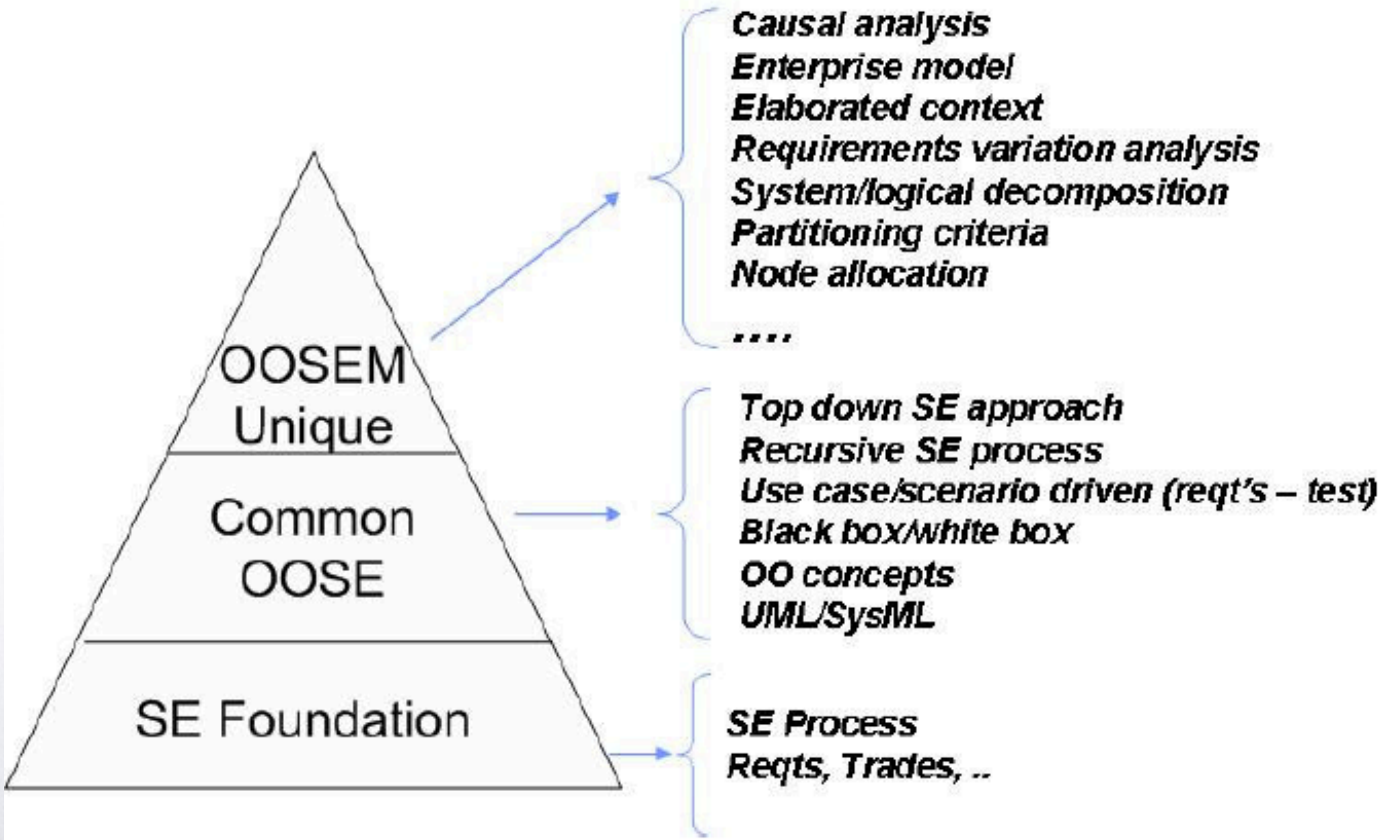
OOSEM appear in mid 1990’s in an attempt to reinforce object-oriented method to system design. It turns to an INCOSE chapter in 2000 and receive later the support of OMG.

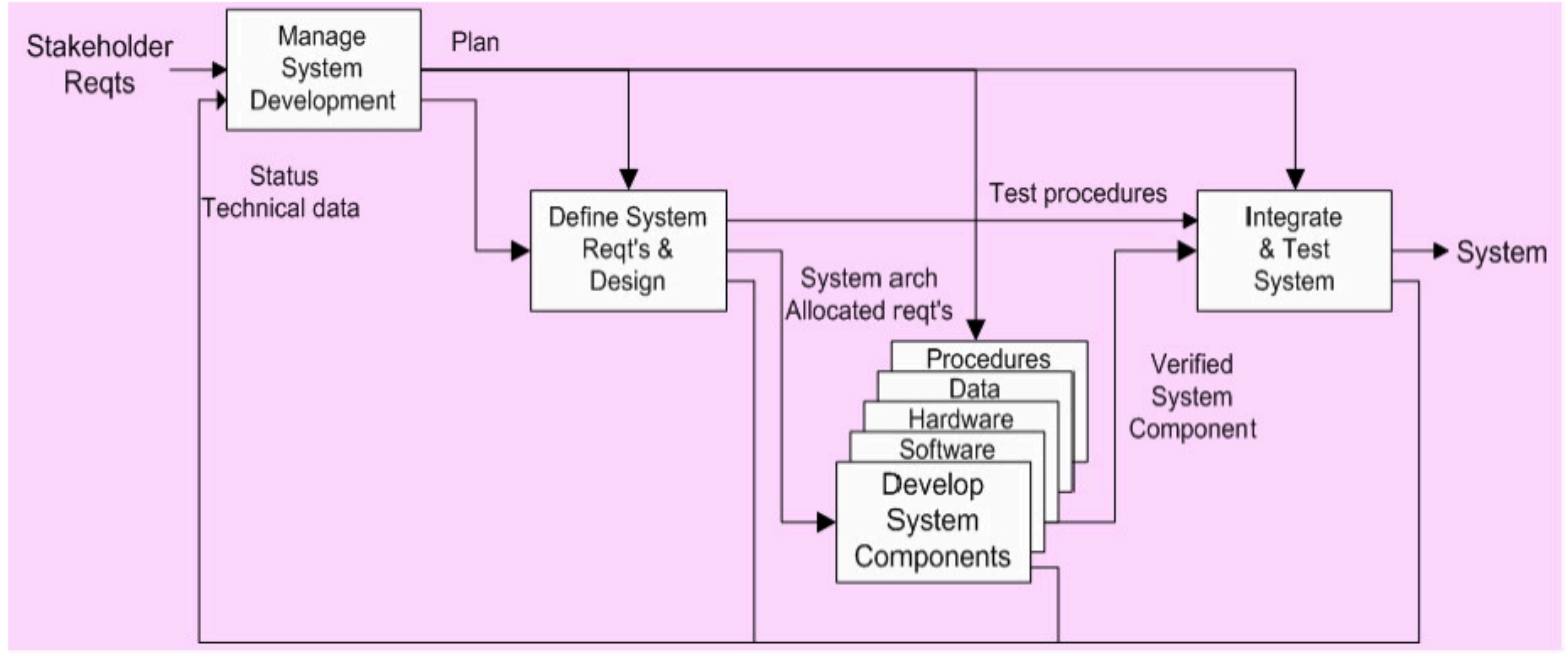
In 2012 Morgan Kauffman published a book by Friedenthal, Moore and Steiner with the title “A Practical Guide to SysML”, where OOSEM is detached as a method and SysML the specification language.

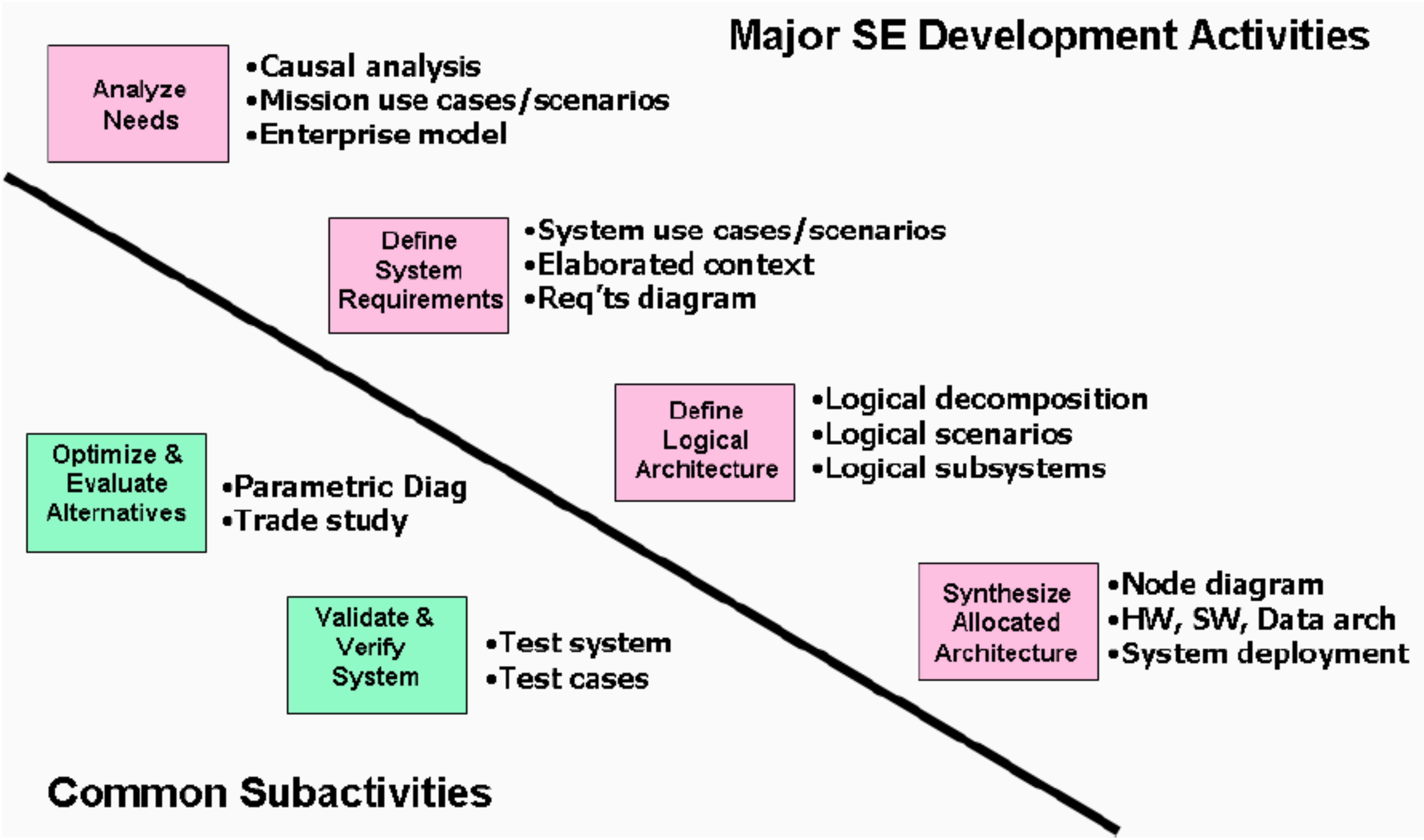


The OOSEM objectives are the following:

- Capture and analysis of requirements and design information to specify complex systems.
- Integration with object-oriented (OO) software, hardware, and other engineering methods.
- Support for system-level reuse and design evolution.


OOSEM
Unique
Common
OOSE
SE Foundation
Causal analysis
Enterprise model
Elaborated context
Requirements variation analysis
System/logical decomposition
Partitioning criteria
Node allocation
....
Top down SE approach
Recursive SE process
Use case/scenario driven (reqt's – test)
Black box/white box
OO concepts
UML/SysML
SE Process
Reqts, Trades, ..

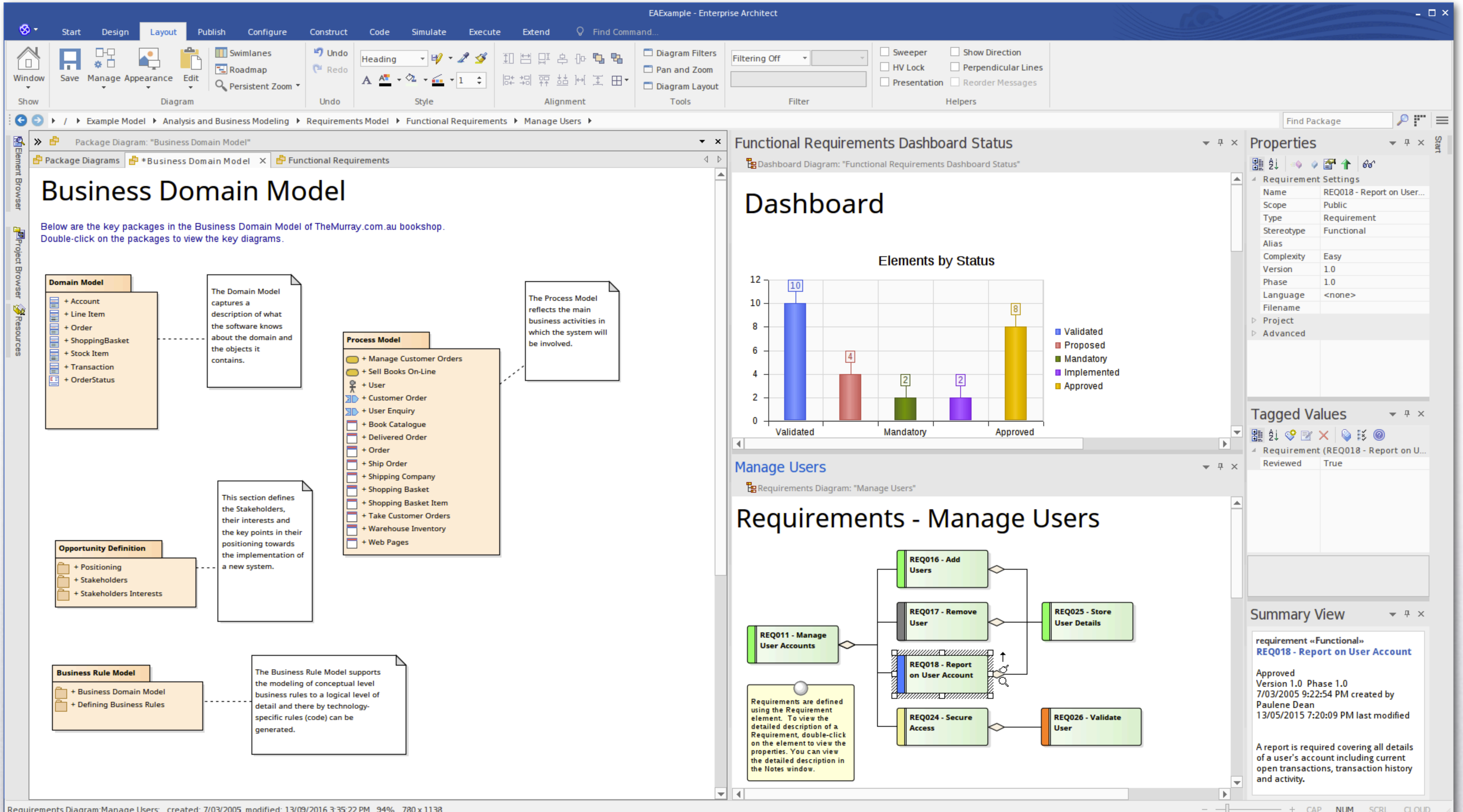

Stakeholder Reqts
Manage System Development
Plan
Status
Technical data
Define System Reqt's & Design
Test procedures
Integrate & Test System
System
System arch
Allocated reqt's
Procedures
Data
Hardware
Software
Develop System Components
Verified System Component


Major SE Development Activities
Analyze Needs
•Causal analysis
•Mission use cases/scenarios
•Enterprise model
Define System Requirements
•System use cases/scenarios
•Elaborated context
•Req'ts diagram
Define Logical Architecture
•Logical decomposition
•Logical scenarios
•Logical subsystems
Synthesize Allocated Architecture
•Node diagram
•HW, SW, Data arch
•System deployment
Optimize & Evaluate Alternatives
•Parametric Diag
•Trade study
Validate & Verify System
•Test system
•Test cases
Common Subactivities

Requirements Analysis

System-of-Interest (Level of Design)	OOSEM Black-Box Scenario	Corresponding OOSEM White-Box Scenario
Enterprise	Mission Scenario	System Scenario
System	System Scenario	Logical Scenario
Logical Subsystem (recursively)	Logical Scenario	Logical Scenario (recursively)

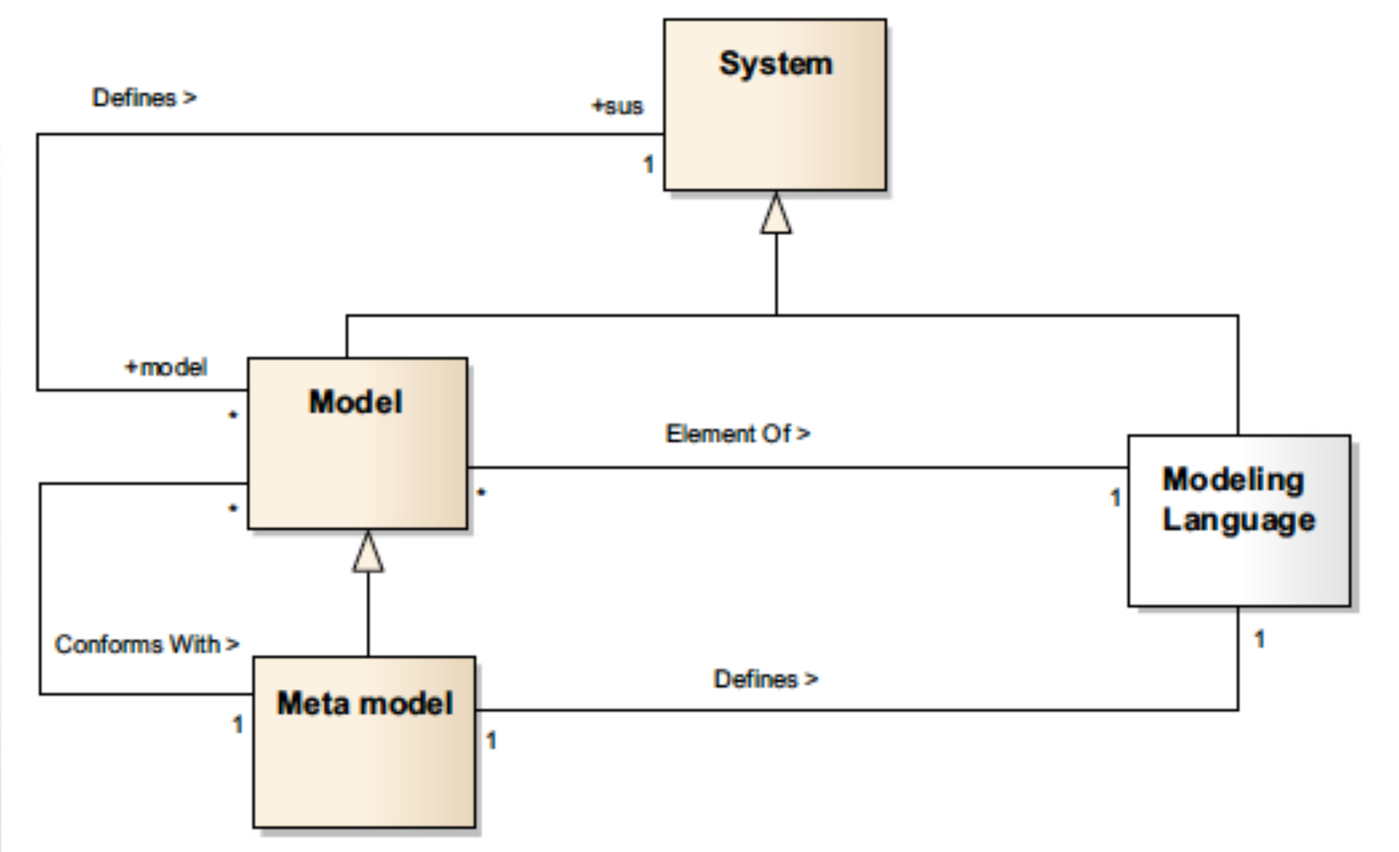
Enterprise Architect (SparX)



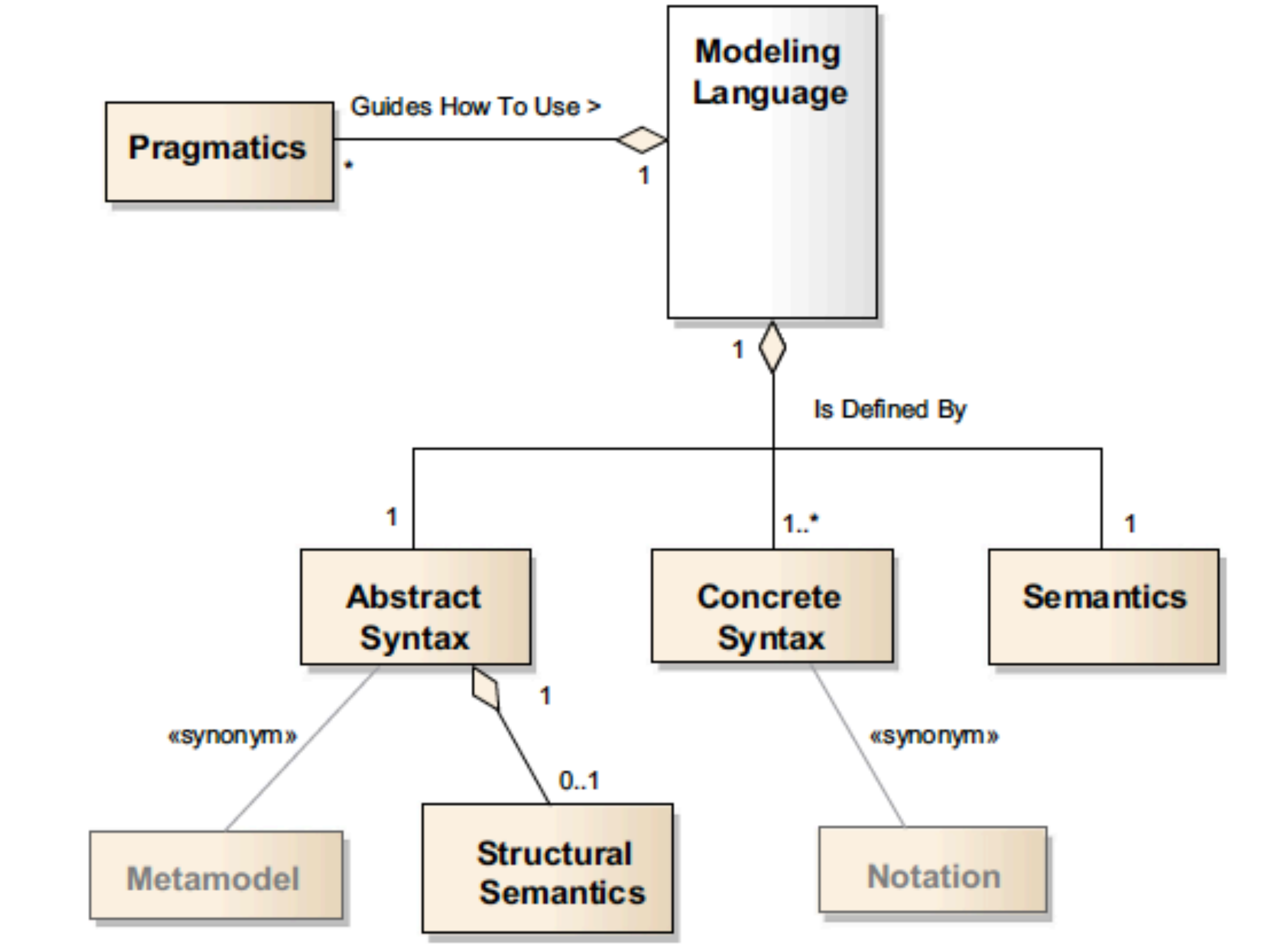
Language, model and meta-model

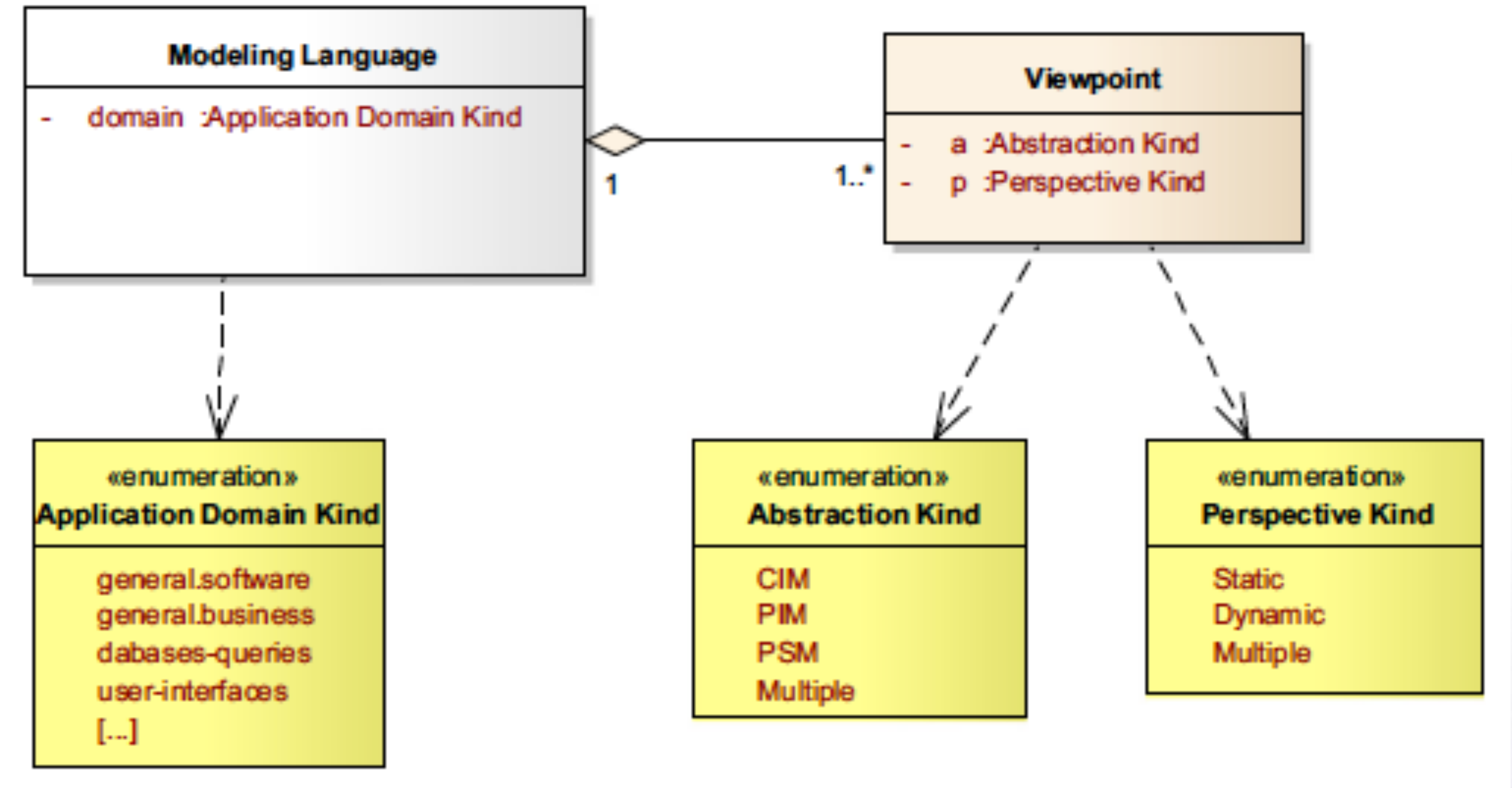
- System

  “a metamodel is a model that defines the structure of a modeling language”.

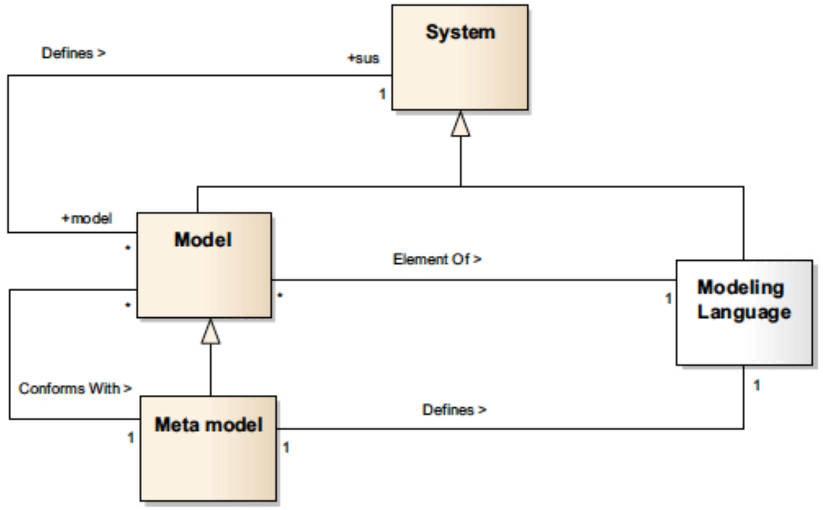


Going down to the concrete project




Modeling Language
- domain :Application Domain Kind
1
1..*
Viewpoint
- a :Abstraction Kind
- p :Perspective Kind
«enumeration»
Application Domain Kind
general.software
general.business
dabases-queries
user-interfaces
[...]
«enumeration»
Abstraction Kind
CIM
PIM
PSM
Multiple
«enumeration»
Perspective Kind
Static
Dynamic
Multiple

Classifying the modeling language



There are two kinds of modeling languages

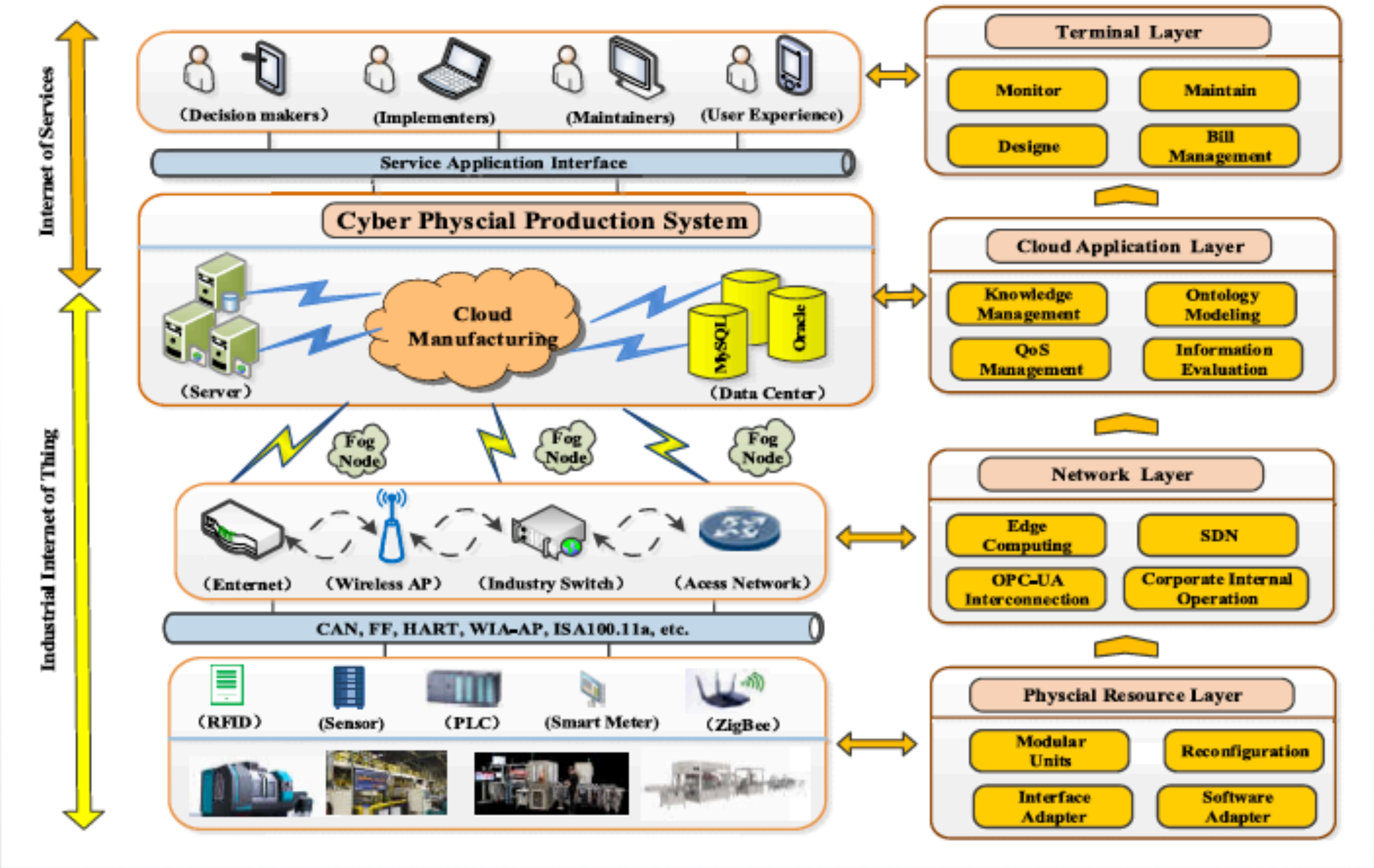
(1) General Purpose Languages

(2) Domain Specific Languages

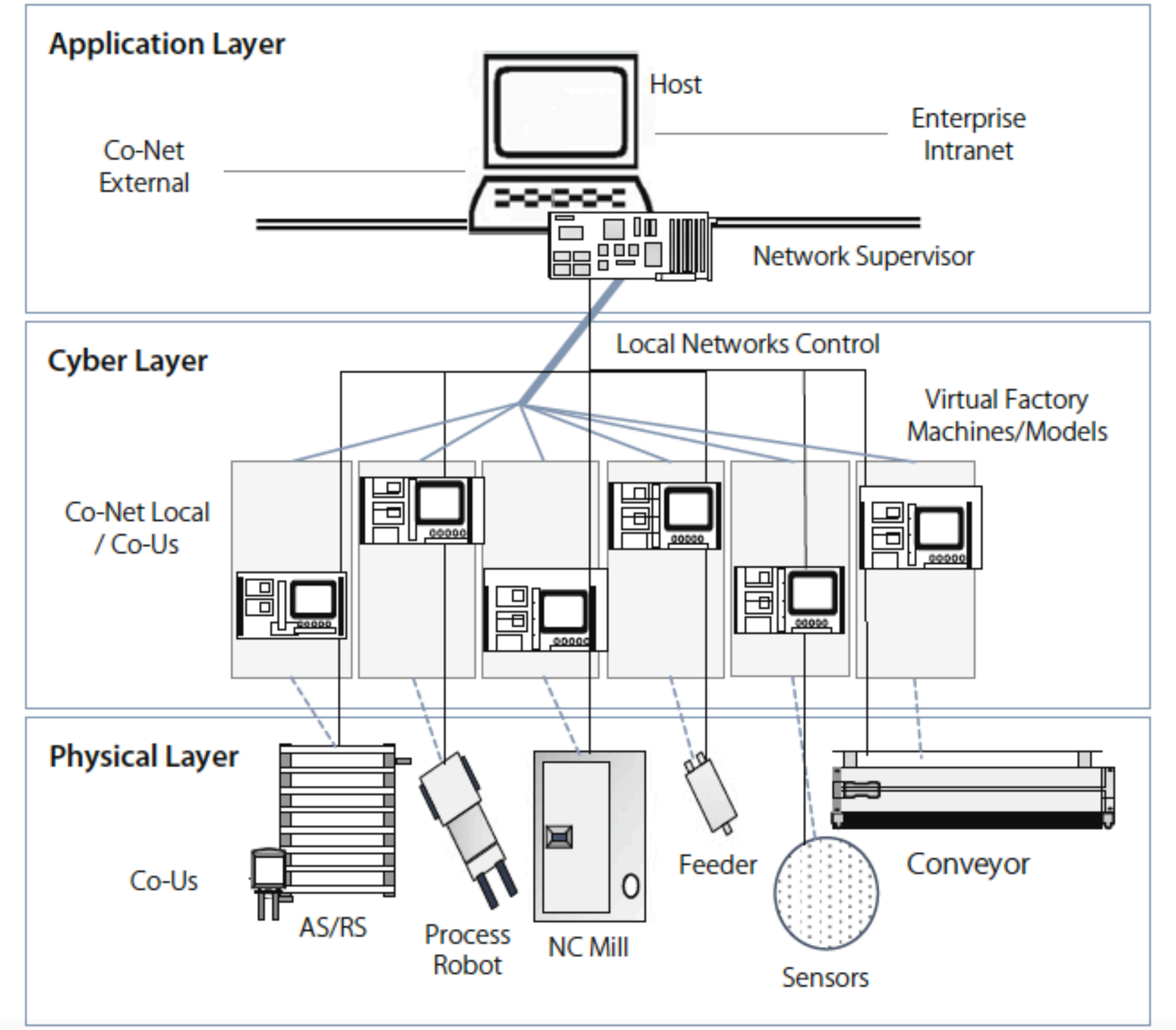
Table 1
Classification of modeling languages: UML2, BPMN, XIS-Mobile and DSL3S.

Modeling Language				
Name	**Application Domain**	**Viewpoint**	**Abstraction**	**Perspective**
UML (Unified Modeling Language)	General/Software	Class Diagram	Multiple	Static
		Object Diagram	Multiple	Static
		Sequence Diagram	Multiple	Dynamic
		Use Case Diagram	PIM	Dynamic
		State Machine Diagram	Multiple	Dynamic
		Component Diagram	PSM	Static
		–	–	–
BPMN (Business Process Modeling Notation)	General/Business Processes	Process Diagram	CIM	Dynamic
		Collaboration Diagram	CIM	Dynamic
		Choreography Diagram	CIM	Dynamic
		Conversation Diagram	CIM	Dynamic
XIS-Mobile (DSL for Mobile Apps)	Specific/Mobile Apps	Domain View	PIM	Static
		BusinessEntities View	PIM	Static
		Architectural View	PIM	Static
		UseCases View	PIM	Dynamic
		NavigationSpace View	PIM	Static
		InteractionSpace View	PIM	Static
DSL3S (DSL for Spatial Simulation Scenarios)	Specific/Spatial Apps	Simulation View	PIM	Static
		Scenario View	PIM	Static
		Animat View	PIM	Static
		Animat Interactions View	PIM	Static

Industry 4.0 Architecture



B. Chen et al.: Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges, IEEE Access, vol 6, March 9, 2018



Moghaddam, M., Nof, S.Y.; Best Matching Theory & Applications, ACES (Automation, Collaboration & E-Service) Series, Springer, 2017

Designing large Service Information Systems

Novo SIS

Sistemas de informação conjugam flexibilidade e capacidade de integração, fundamental para inovação e automação.[1]
Convergência entre sistemas de serviço e sistemas de informação. [2]

[1] Stair, R.; Reynolds, G. "*Information Systems*", 9th ed., *Course Technology*, 2010.
[2] Bardhan, I. ; Demirkan, H.; Kannan, P.; Kauffman, R.; Sougstad, R. "*An Interdisciplinary Perspective on IT Services Management and Service Science*". *Journal of Management Information Systems*, v. 26, n. 4, p. 13-64, 2010.

Top 10 Engineering Document and Data Management Challenges

1. Finding the right documents
2. Version control
3. Change management
4. Scalability and flexibility

One of our customers told us about a project that involved 290 spreadsheets that contained somewhere close to 8,000 wires. One spreadsheet alone had 1,000 instruments and 169 columns for data entry!

5. Multi-user collaboration
6. Multiple database
7. Backup and security
8. Management across the project life cycle
9. Compliance with various standards
10. Reinventing the wheel (reusability)

A mudança de paradigma

produto

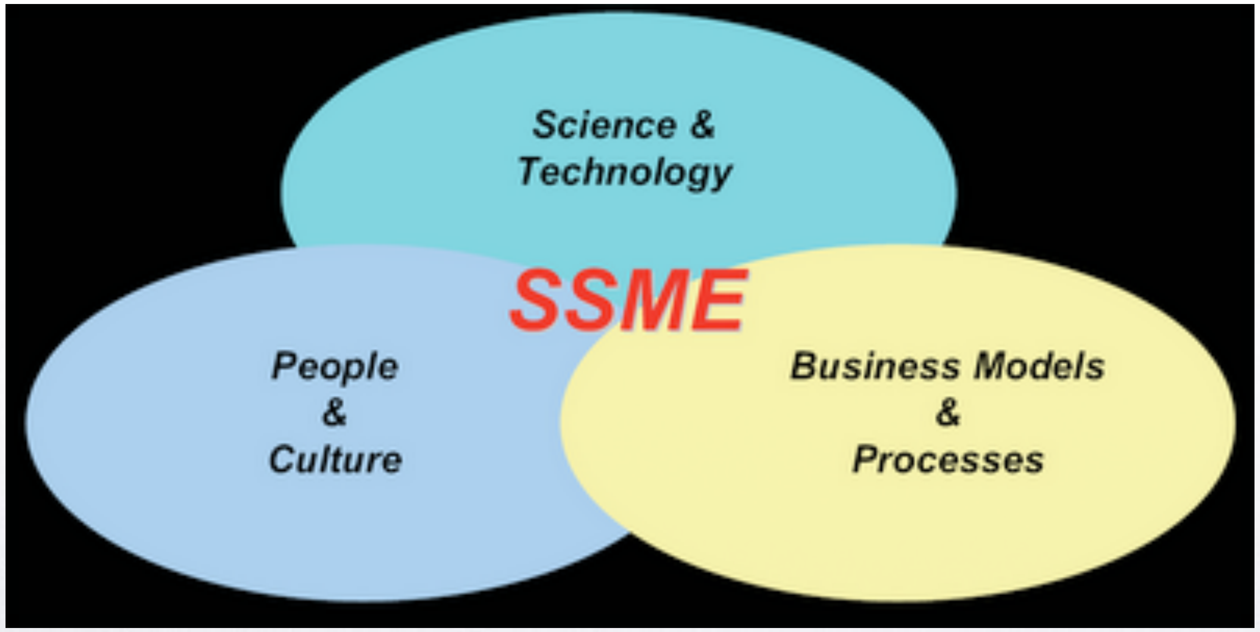
sistema

→ mesmo desenvolvimento orientado a produto

produto
&
systema

→ opção por um desenvolvimento orientado a modelos

Service Science, Management and Engineering



SSME is a new research field that aims to formalize and control the relationship between humans and (cognitive) information systems to establish a new paradigm of associative interaction.

Obrigado

Reinaldo