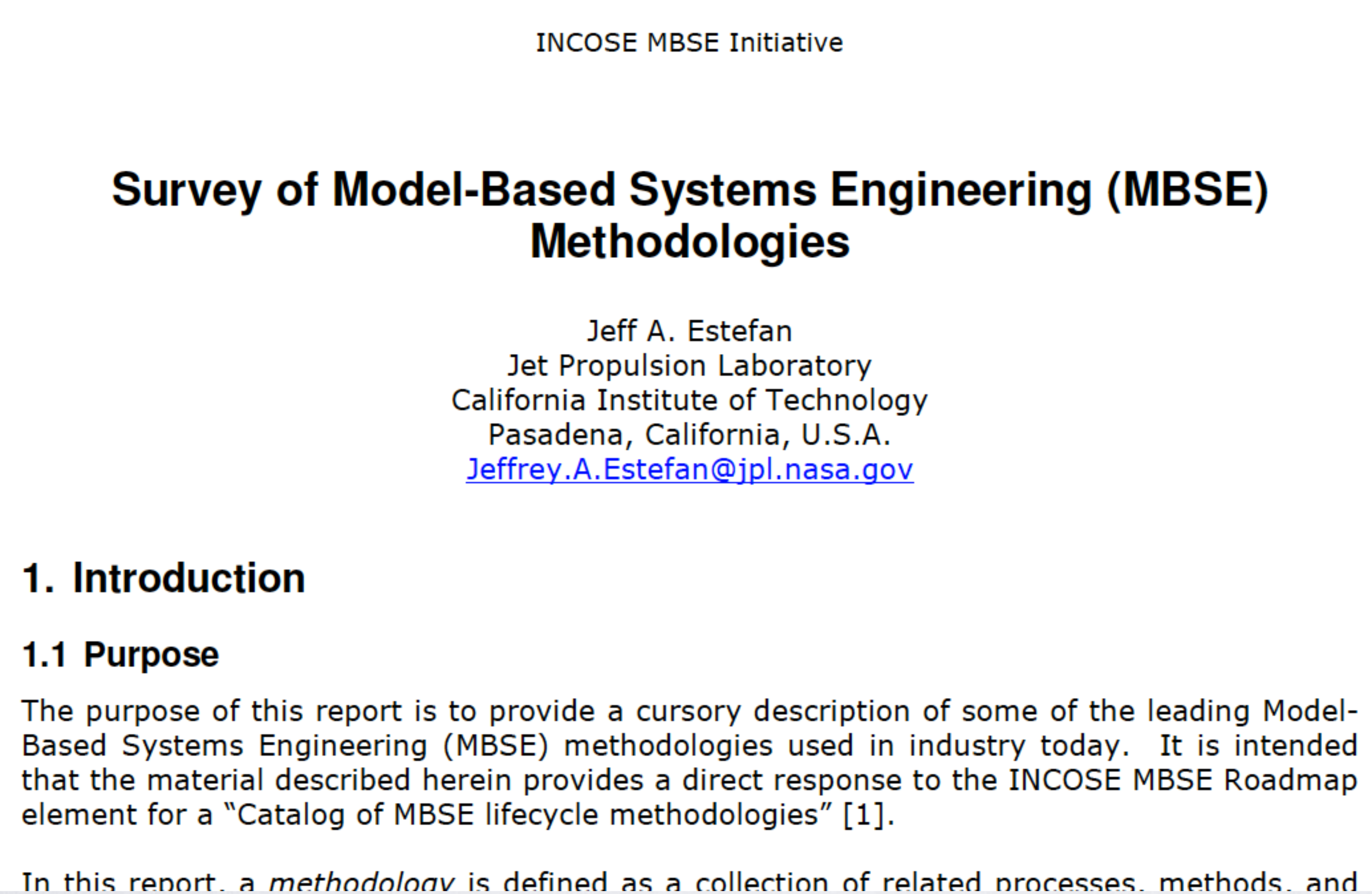
PMR 5020

Modelagem do Projeto de Sistemas

Aula 12: MBSE in the new design era

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



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

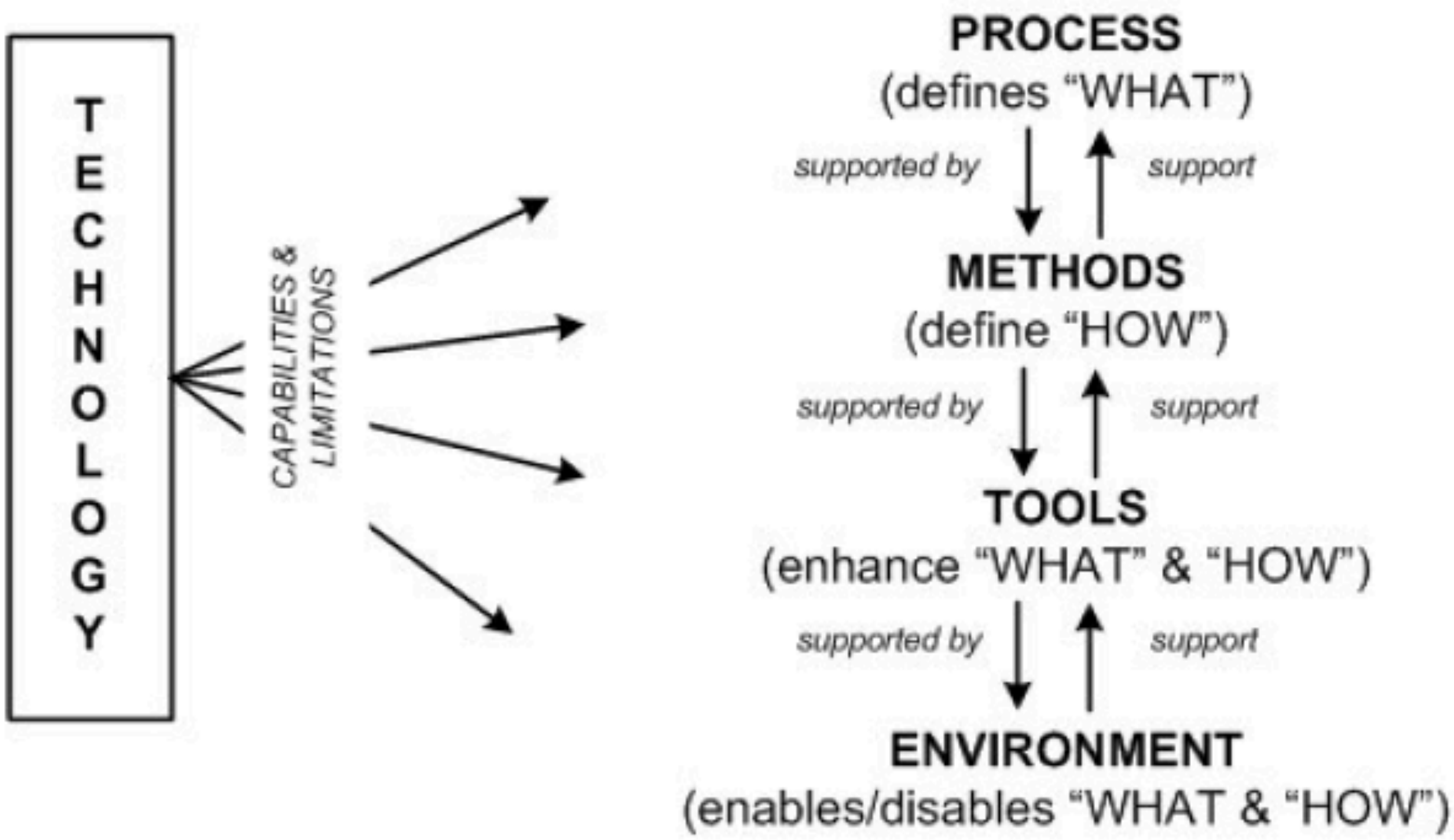
1.1 Purpose

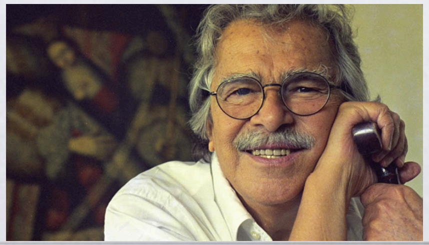
The purpose of this report is to provide a cursory description of some of the leading Model-Based Systems Engineering (MBSE) methodologies used in industry today. It is intended that the material described herein provides a direct response to the INCOSE MBSE Roadmap element for a "Catalog of MBSE lifecycle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

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

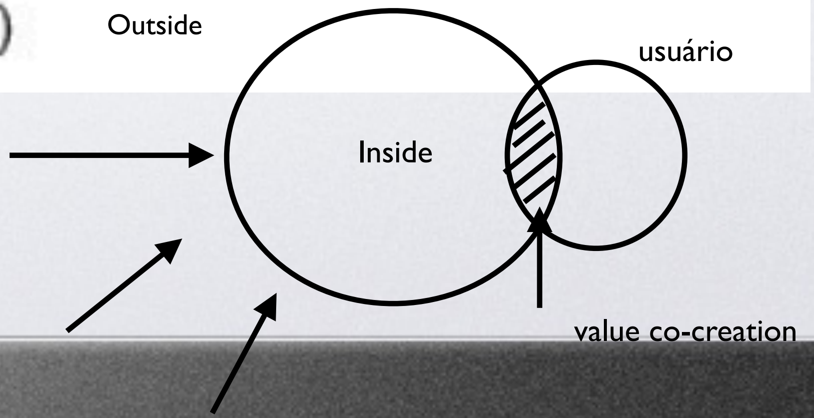
Principal Engineer at NASA's
Jet Propulsion Lab

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

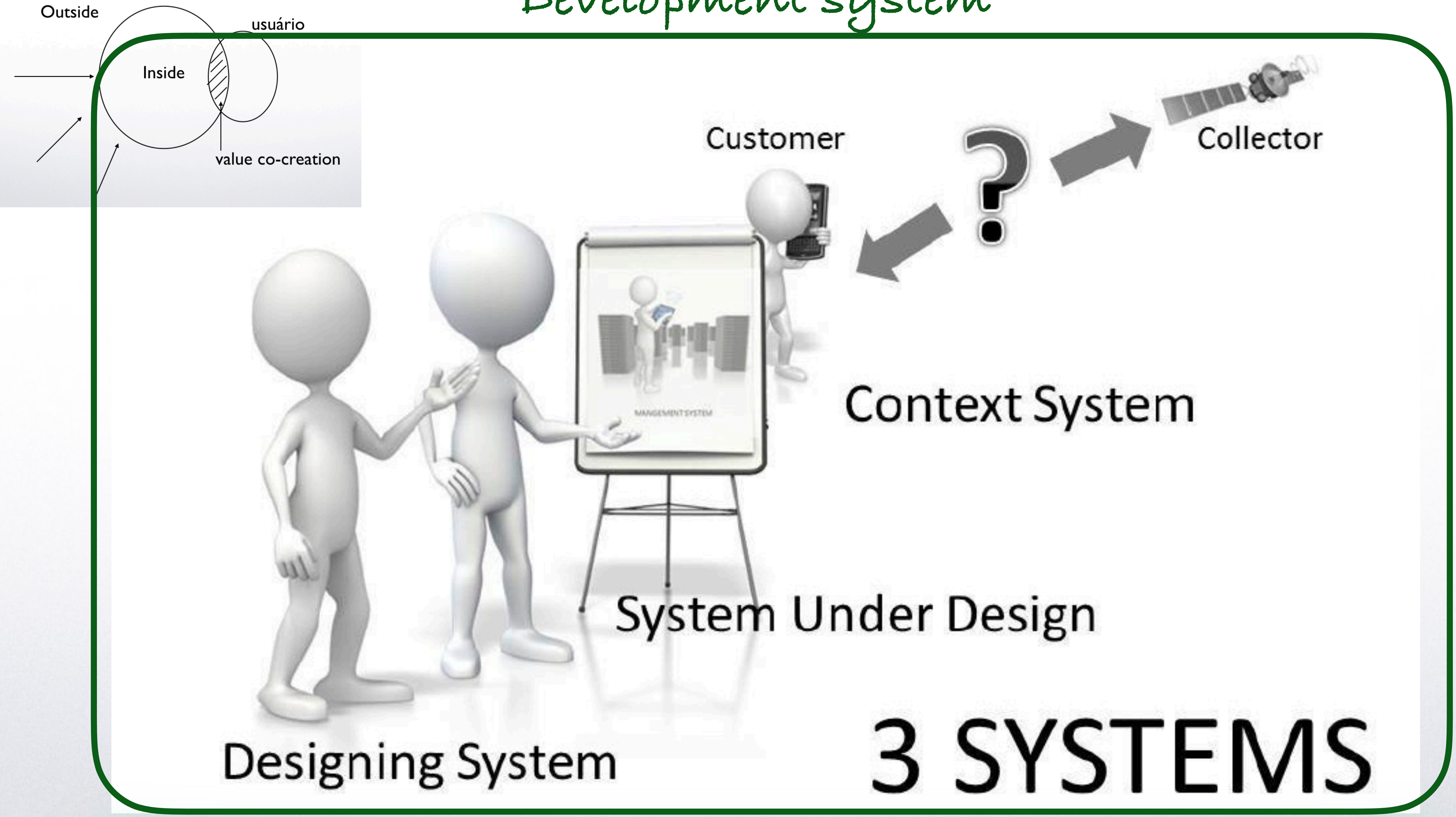




Darcy Ribeiro



Development system



Customer

Collector

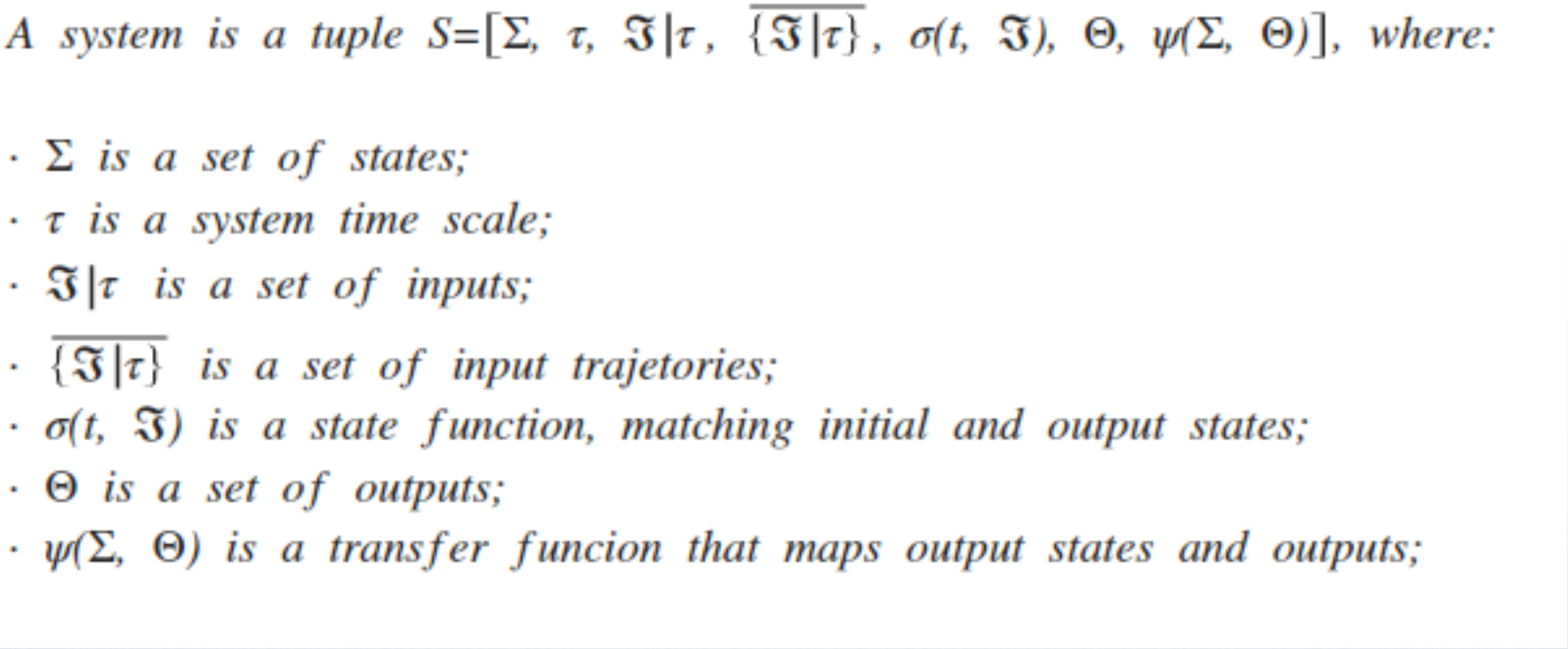
Context System

System Under Design

Designing System

3 SYSTEMS

A T3SD (tricotyledon theory of system design)



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

- Σ is a set of states;
- τ is a system time scale;
- $\Im|\tau$ is a set of inputs;
- $\overline{\{\Im|\tau\}}$ is a set of input trajetories;
- $\sigma(t,\ \Im)$ 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

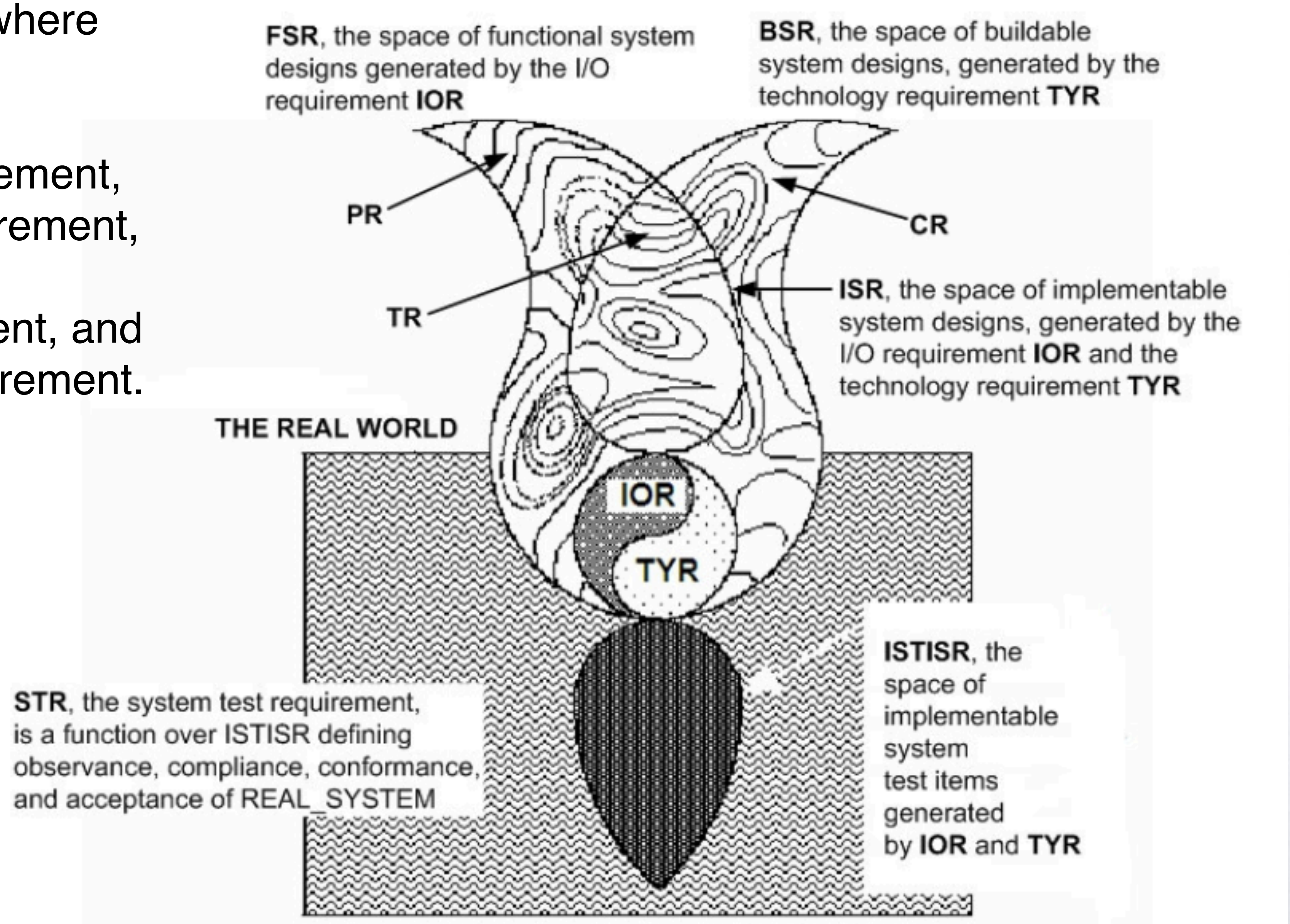
i) IOR is the I/O requirement,
ii) TYR is the technology requirement,
iii) PR is the performance requirement,
iv) CR is the cost requirement,
v) TR is the trade-off requirement, and
vi) STR is the system test requirement.

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.



FSR, the space of functional system designs generated by the I/O requirement **IOR**

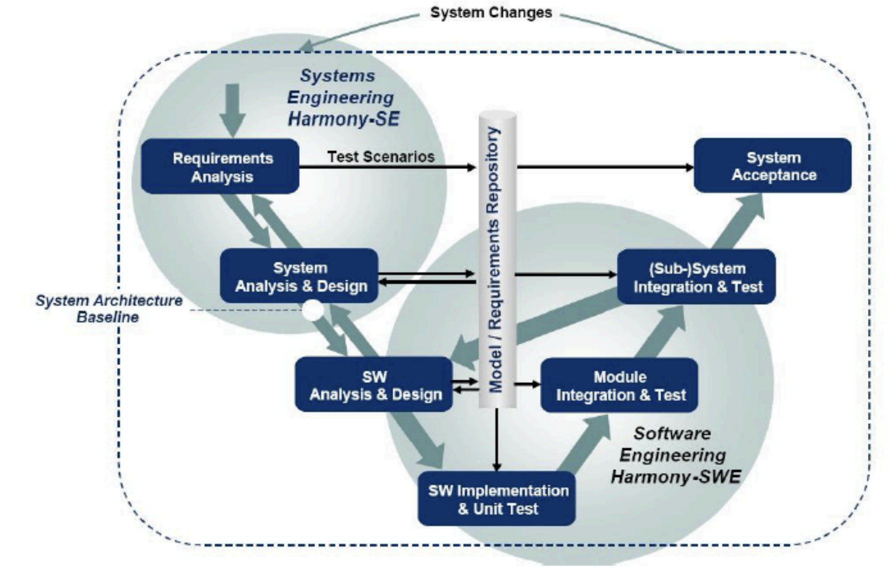
BSR, the space of buildable system designs, generated by the technology requirement **TYR**

ISR, the space of implementable system designs, generated by the I/O requirement **IOR** and the technology requirement **TYR**

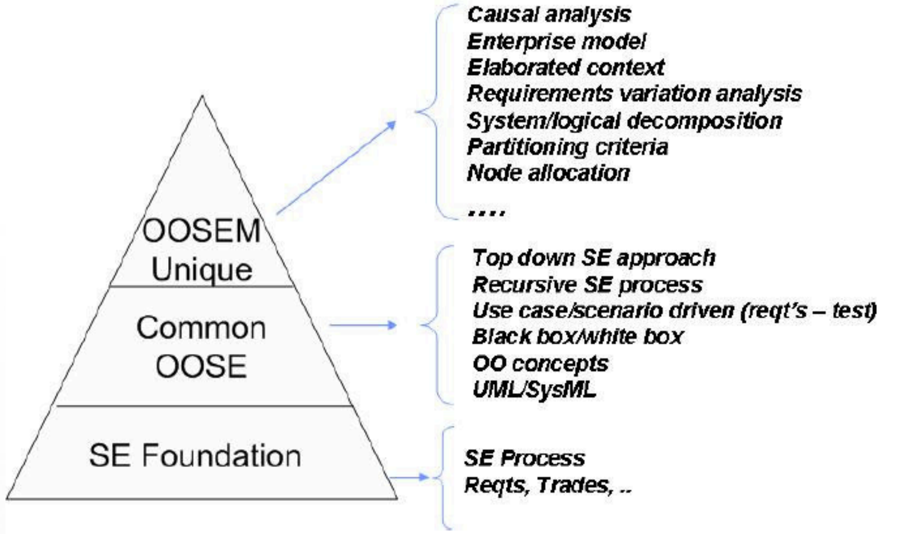
STR, the system test requirement, is a function over ISTISR defining observance, compliance, conformance, and acceptance of REAL_SYSTEM

ISTISR, the space of implementable system test items generated by **IOR** and **TYR**

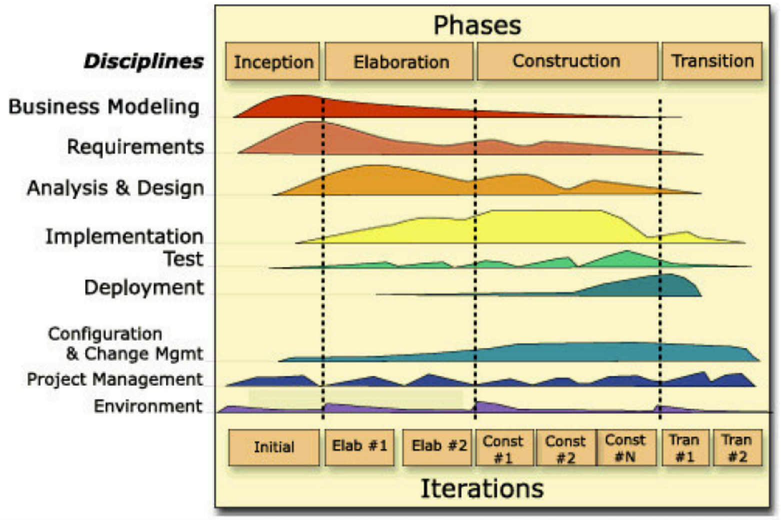
MBSE Methodologies


System Changes
Systems Engineering Harmony-SE
Requirements Analysis
Test Scenarios
System Acceptance
Model / Requirements Repository
System Analysis & Design
(Sub-)System Integration & Test
System Architecture Baseline
SW Analysis & Design
Module Integration & Test
Software Engineering Harmony-SWE
SW Implementation & Unit Test

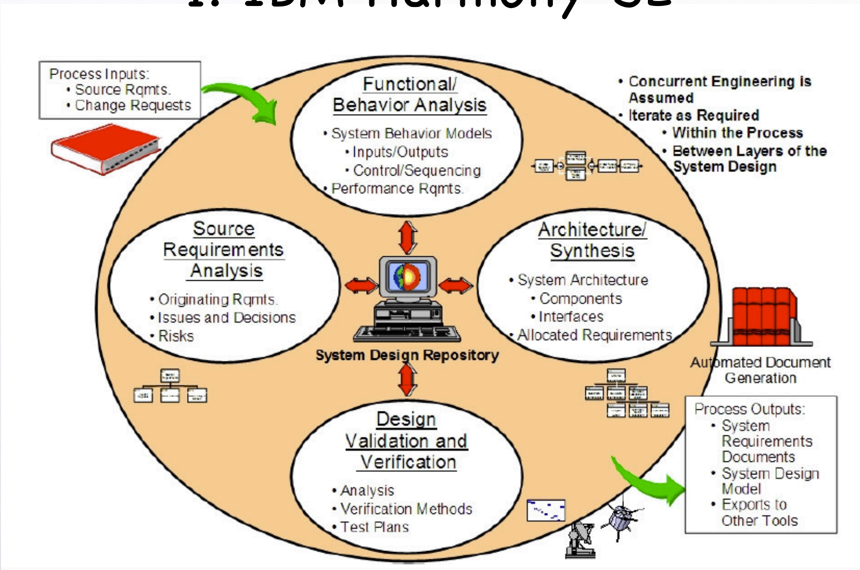
1. IBM Harmony-SE



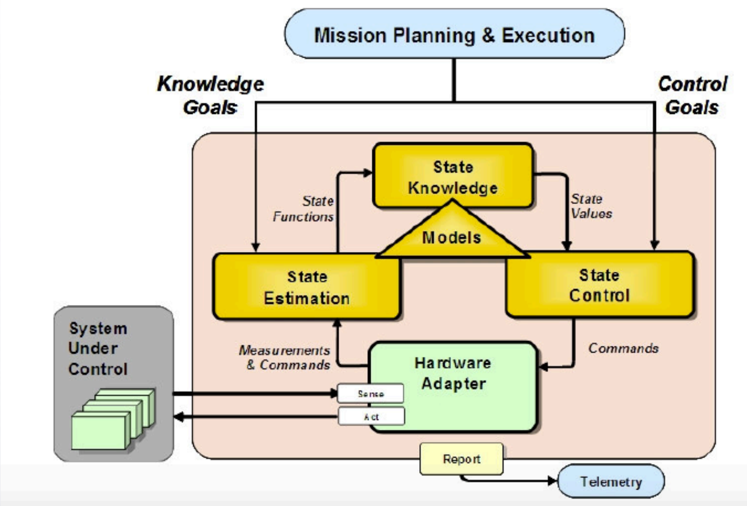
2. OOSEM



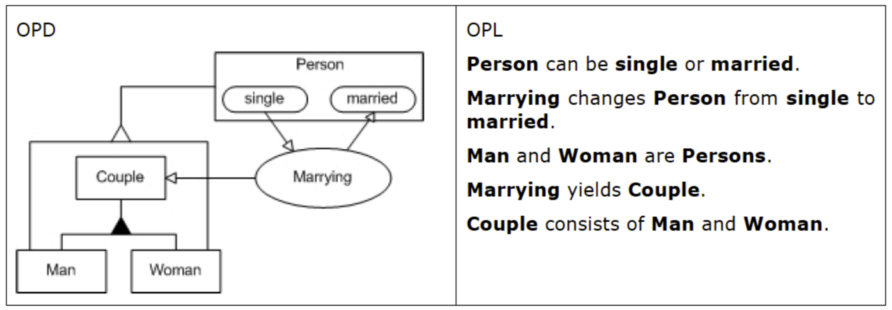
3. RUP



4. Vitech



5. JPL



OPL

Person can be **single** or **married**.

Marrying changes **Person** from **single** to **married**.

Man and **Woman** are **Persons**.

Marrying yields **Couple**.

Couple consists of **Man** and **Woman**.

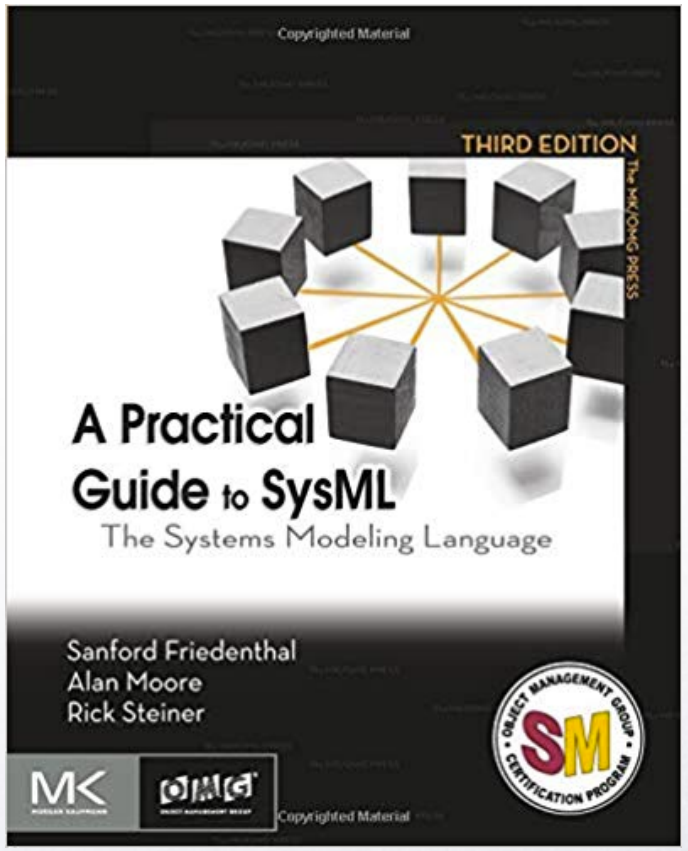
6. OPM (Object-Process Methodology)
ISO/PAS 19450 (Dovi Dori)

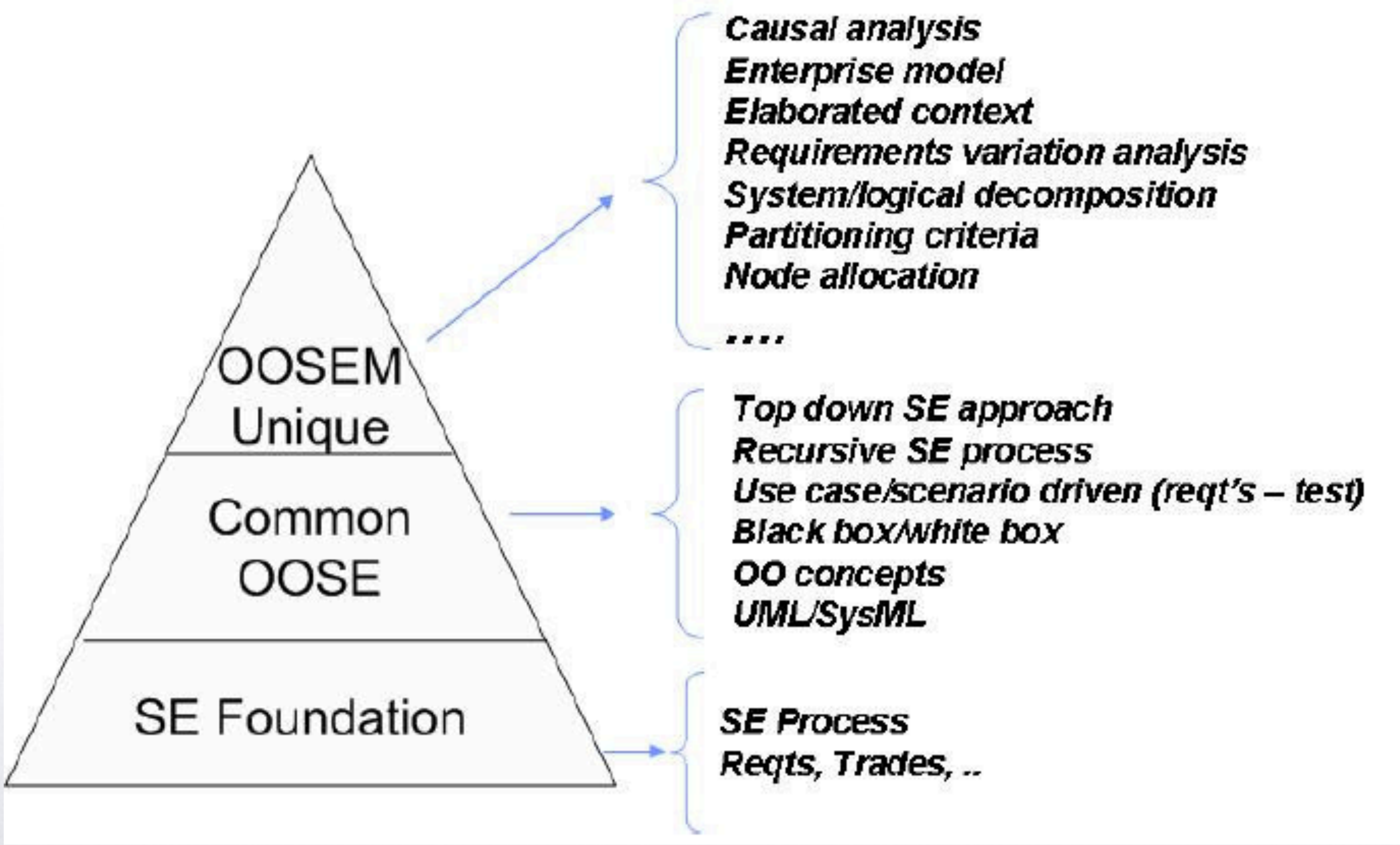
Method.	Proponent	Tool	Reference
IBM Harmony-SE	IBM	not specific	IBM Rhapsody
OOSEM	INCOSE	not specific	INCOSE/OMG
IBM Rational	IBM	RUP	IBM Rational
Vitech	Vitech	CORE	www.vitech.com
JPL	JPL	State DB	JPL Caltech
OPM	Dov Dori (1995)	OPCAT	www.opcat.com ISO/PAS 19450

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.





OOSEM Unique

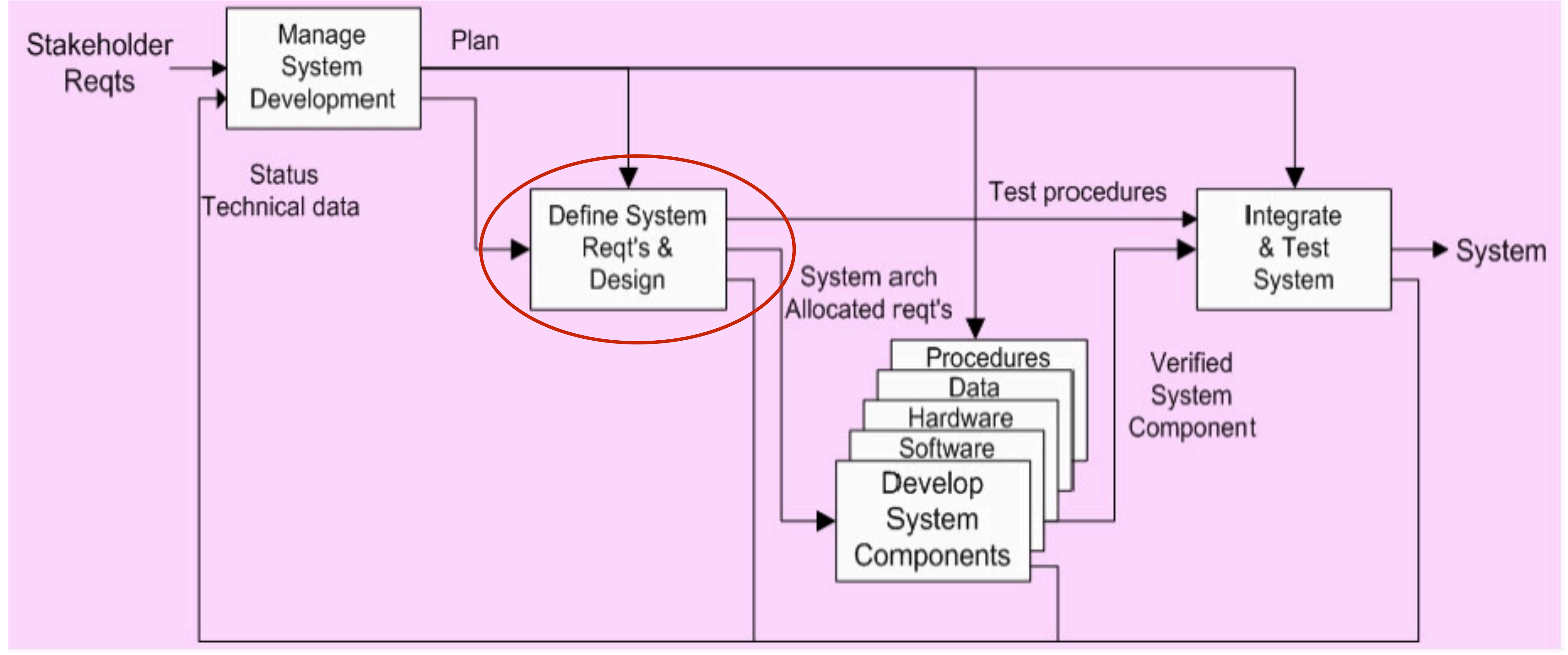
Common OOSE

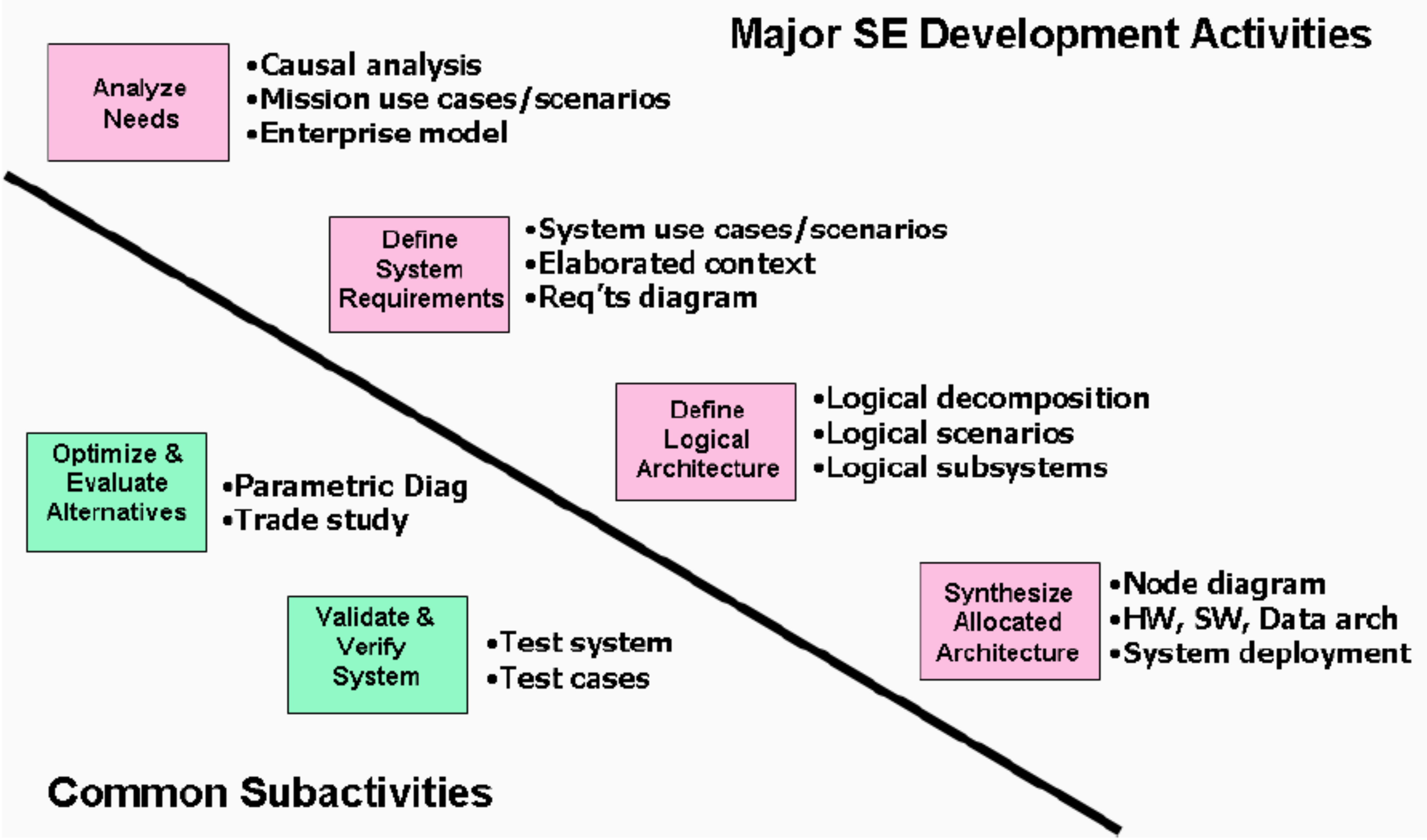
SE Foundation

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

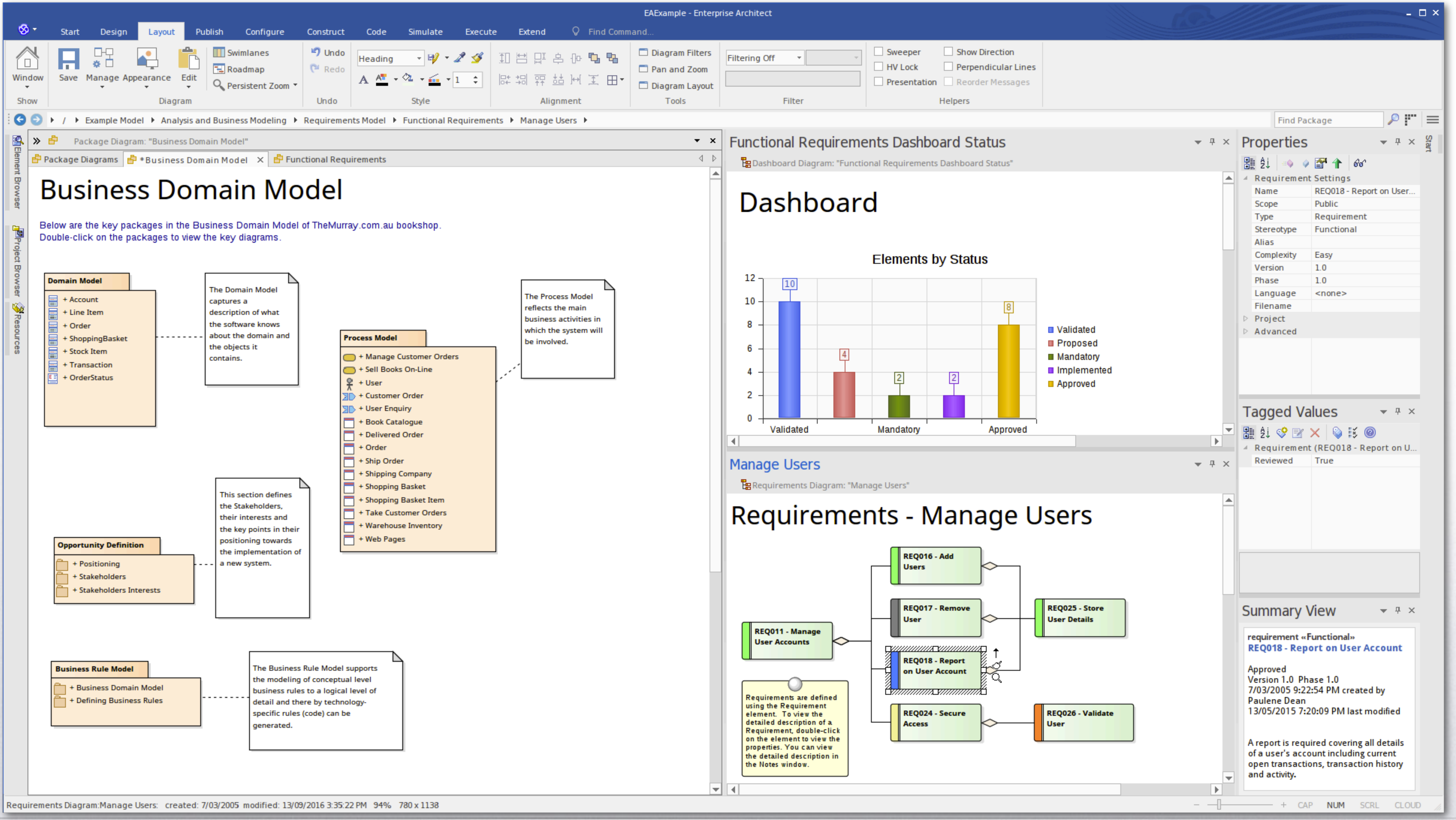
- •Parametric Diag
- •Trade study

Validate & Verify System

- •Test system
- •Test cases

Common Subactivities

Enterprise Architect (SparX)



Business Domain Model

Below are the key packages in the Business Domain Model of TheMurray.com.au bookshop.
Double-click on the packages to view the key diagrams.

The Domain Model captures a description of what the software knows about the domain and the objects it contains.

The Process Model reflects the main business activities in which the system will be involved.

This section defines the Stakeholders, their interests and the key points in their positioning towards the implementation of a new system.

The Business Rule Model supports the modeling of conceptual level business rules to a logical level of detail and there by technology-specific rules (code) can be generated.

Dashboard

Requirements - Manage Users

Requirements are defined using the Requirement element. To view the detailed description of a Requirement, double-click on the element to view the properties. You can view the detailed description in the Notes window.

Requirement Settings

Name	REQ018 - Report on User...
Scope	Public
Type	Requirement
Stereotype	Functional
Alias	
Complexity	Easy
Version	1.0
Phase	1.0
Language	<none>
Filename	

Project

Advanced

Requirement (REQ018 - Report on U...

Reviewed	True

requirement «Functional»
REQ018 - Report on User Account

Approved
Version 1.0 Phase 1.0
7/03/2005 9:22:54 PM created by Paulene Dean
13/05/2015 7:20:09 PM last modified

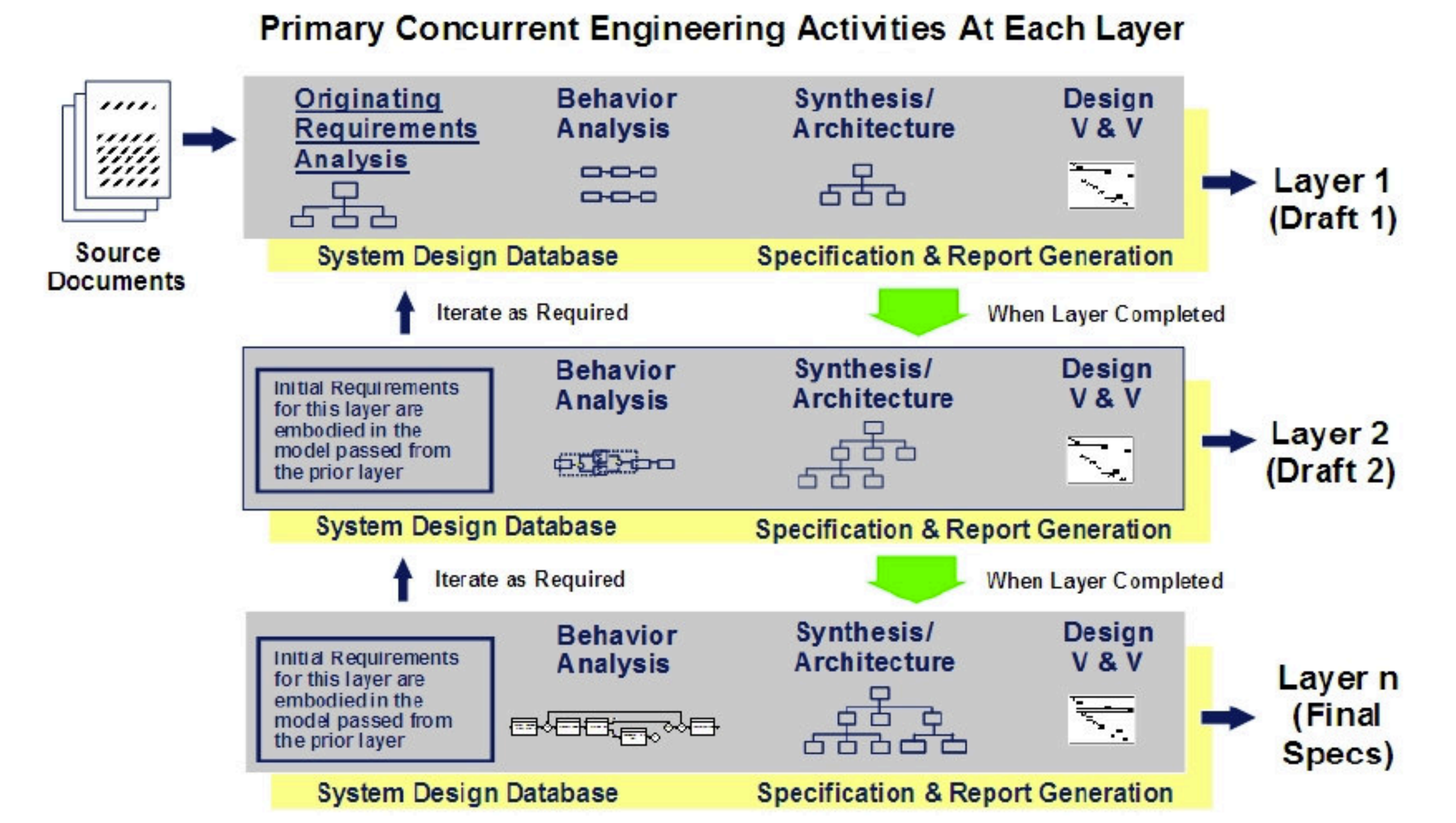
A report is required covering all details of a user's account including current open transactions, transaction history and activity.

Requirements Diagram:Manage Users: created: 7/03/2005 modified: 13/09/2016 3:35:22 PM 94% 780 x 1138

Prof. José Reinaldo Silva

Escola Politécnica da USP

Vitech
Insight through integration

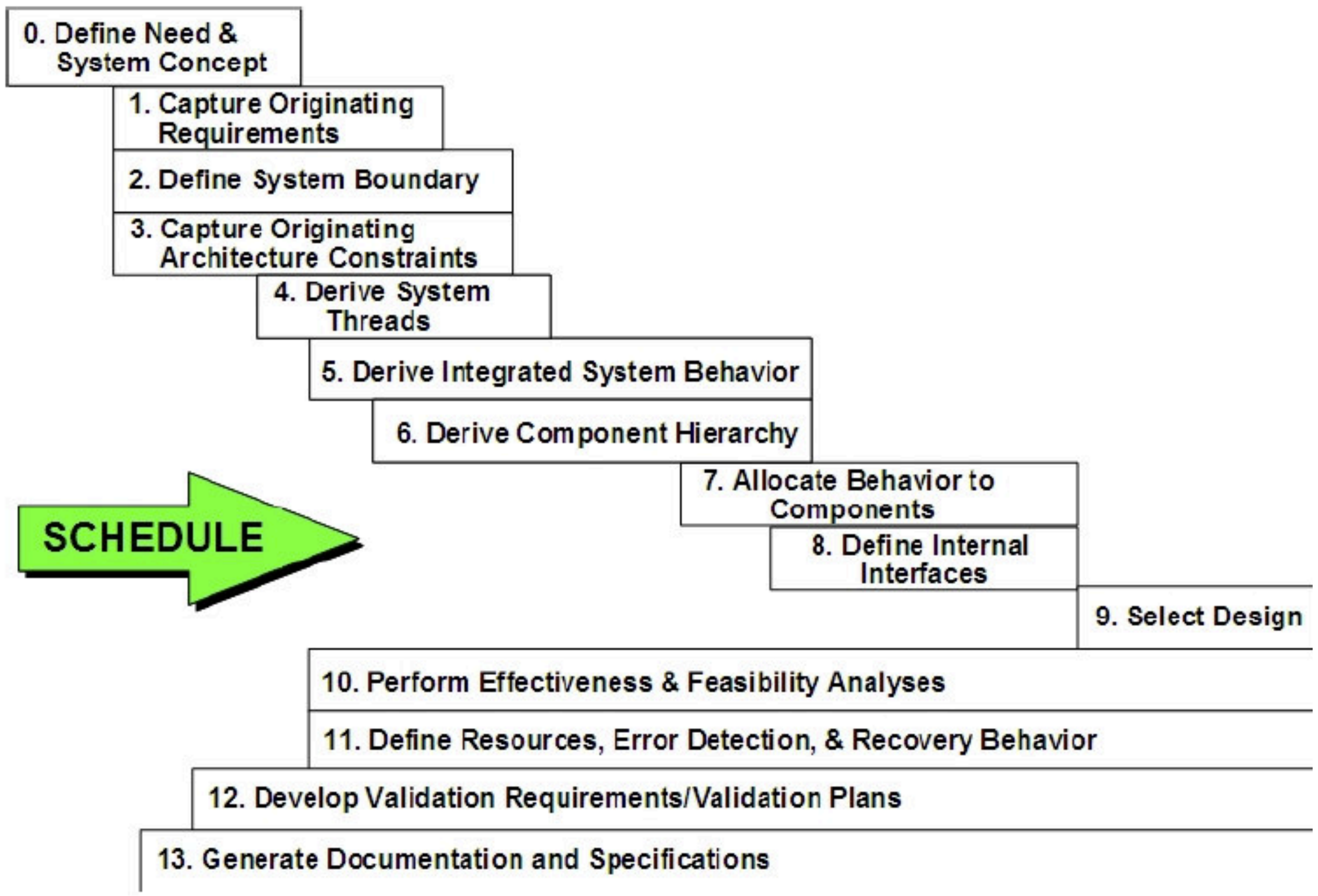


GENESYS

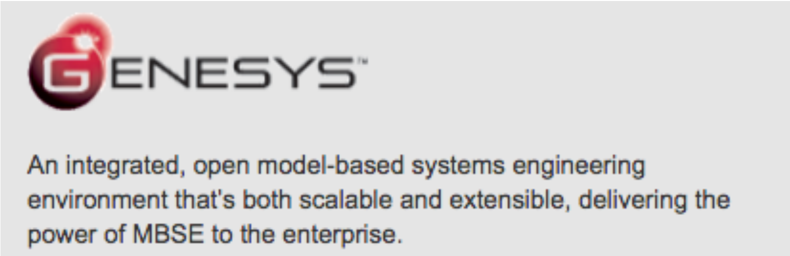
An integrated, open model-based systems engineering environment that's both scalable and extensible, delivering the power of MBSE to the enterprise.

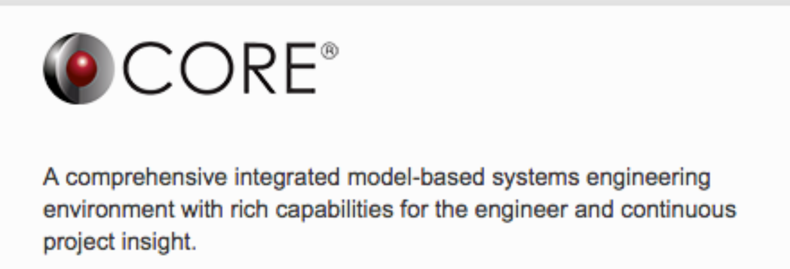
CORE®

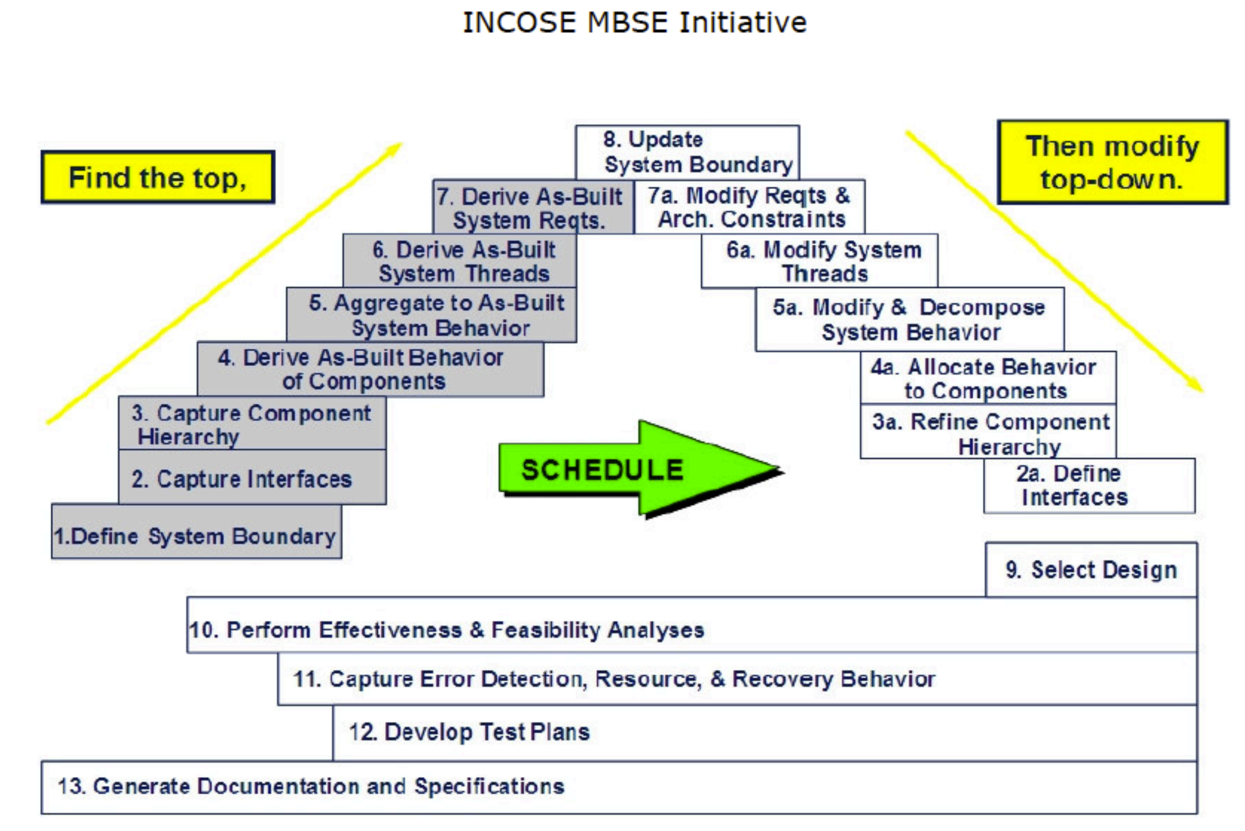
A comprehensive integrated model-based systems engineering environment with rich capabilities for the engineer and continuous project insight.


0. Define Need & System Concept
1. Capture Originating Requirements
2. Define System Boundary
3. Capture Originating Architecture Constraints
4. Derive System Threads
5. Derive Integrated System Behavior
6. Derive Component Hierarchy
7. Allocate Behavior to Components
8. Define Internal Interfaces
9. Select Design
10. Perform Effectiveness & Feasibility Analyses
11. Define Resources, Error Detection, & Recovery Behavior
12. Develop Validation Requirements/Validation Plans
13. Generate Documentation and Specifications
SCHEDULE

Top-down Model

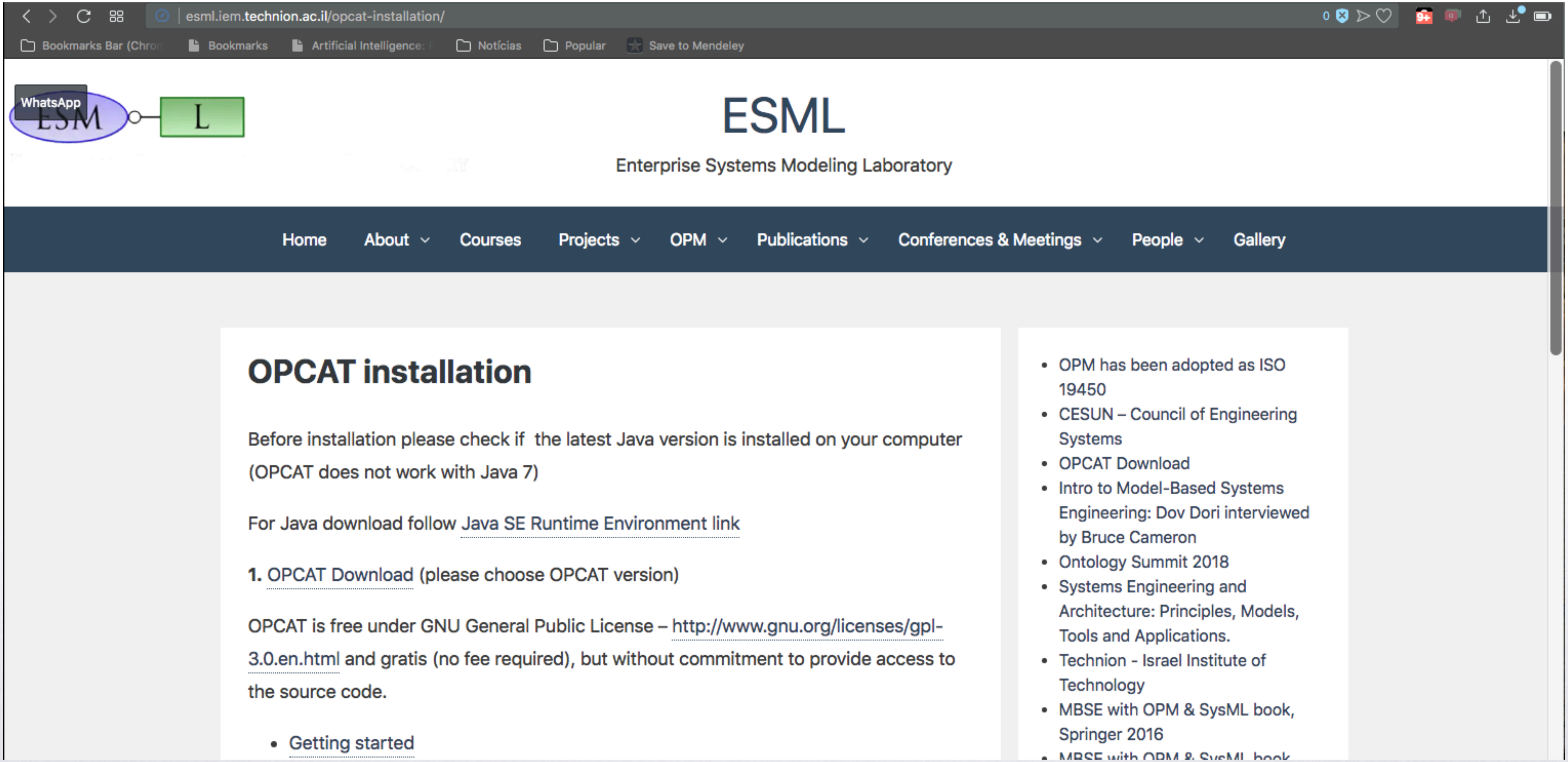

An integrated, open model-based systems engineering environment that's both scalable and extensible, delivering the power of MBSE to the enterprise.


A comprehensive integrated model-based systems engineering environment with rich capabilities for the engineer and continuous project insight.


INCOSE MBSE Initiative
Find the top,
Then modify top-down.
8. Update System Boundary
7. Derive As-Built System Reqts.
7a. Modify Reqts & Arch. Constraints
6. Derive As-Built System Threads
6a. Modify System Threads
5. Aggregate to As-Built System Behavior
5a. Modify & Decompose System Behavior
4. Derive As-Built Behavior of Components
4a. Allocate Behavior to Components
3. Capture Component Hierarchy
3a. Refine Component Hierarchy
2. Capture Interfaces
2a. Define Interfaces
1.Define System Boundary
SCHEDULE
9. Select Design
10. Perform Effectiveness & Feasibility Analyses
11. Capture Error Detection, Resource, & Recovery Behavior
12. Develop Test Plans
13. Generate Documentation and Specifications

Reverse Eng. Model

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



ESML

Enterprise Systems Modeling Laboratory

OPCAT installation

Before installation please check if the latest Java version is installed on your computer (OPCAT does not work with Java 7)

For Java download follow Java SE Runtime Environment link

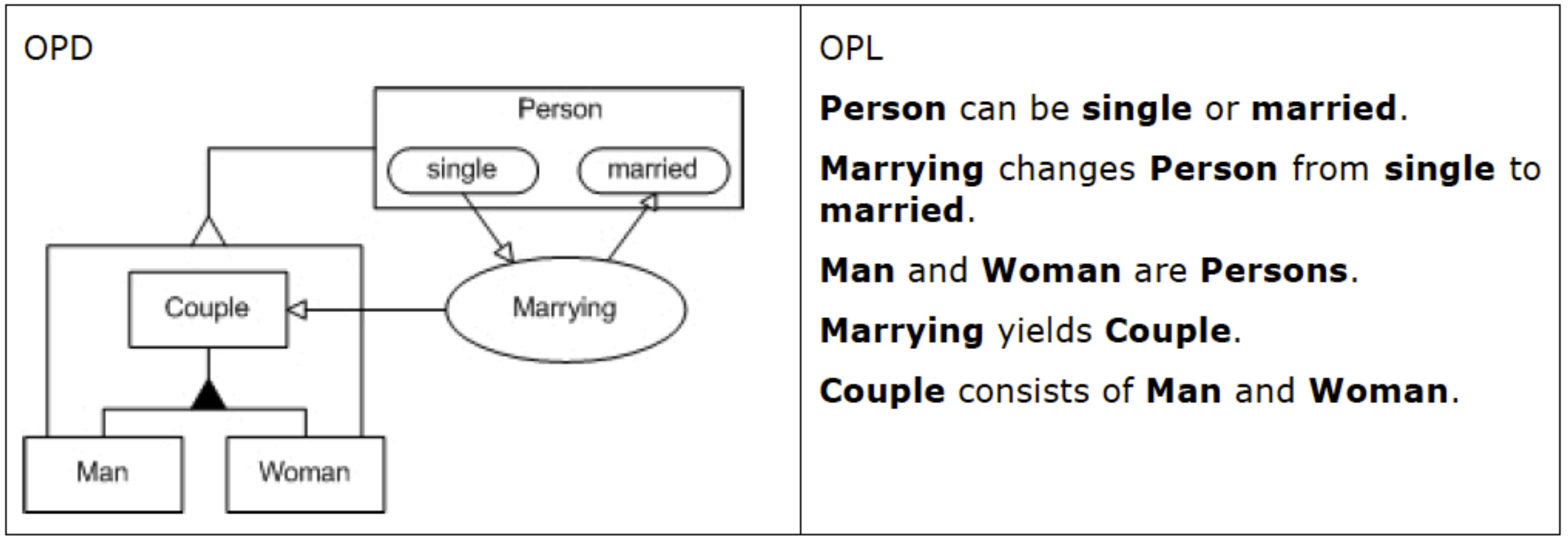
1. OPCAT Download (please choose OPCAT version)

OPCAT is free under GNU General Public License – http://www.gnu.org/licenses/gpl-3.0.en.html and gratis (no fee required), but without commitment to provide access to the source code.

- Getting started

- OPM has been adopted as ISO 19450
- CESUN – Council of Engineering Systems
- OPCAT Download
- Intro to Model-Based Systems Engineering: Dov Dori interviewed by Bruce Cameron
- Ontology Summit 2018
- Systems Engineering and Architecture: Principles, Models, Tools and Applications.
- Technion - Israel Institute of Technology
- MBSE with OPM & SysML book, Springer 2016
- MBSE with OPM & SysML book

- An **object** is a thing that exists or has the potential of existence, phys mentally.
- A **process** is a pattern of transformation that an object undergoes.
- A **state** is a situation an object can be at.



OPL

Person can be **single** or **married**.

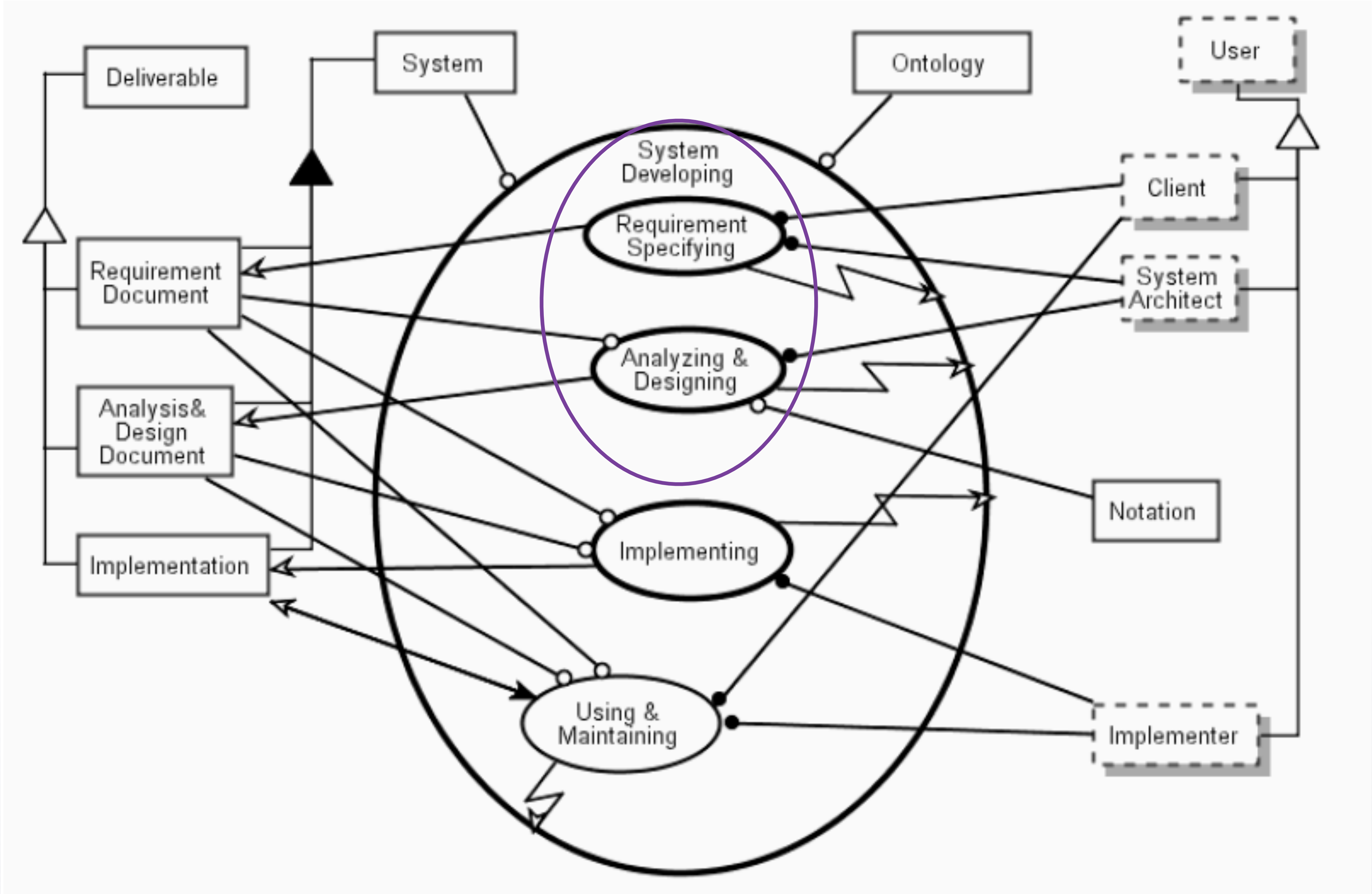
Marrying changes **Person** from **single** to **married**.

Man and **Woman** are **Persons**.

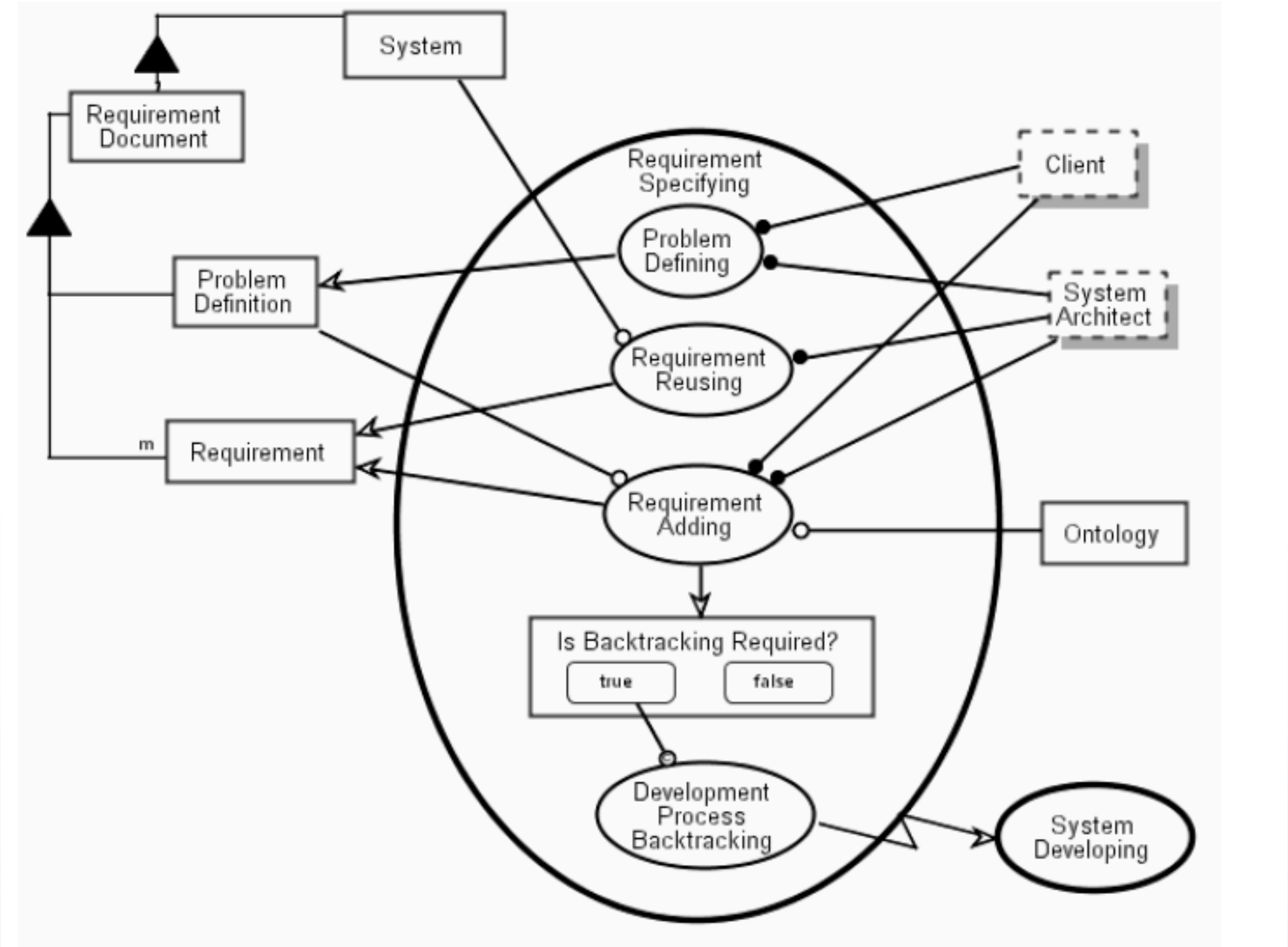
Marrying yields **Couple**.

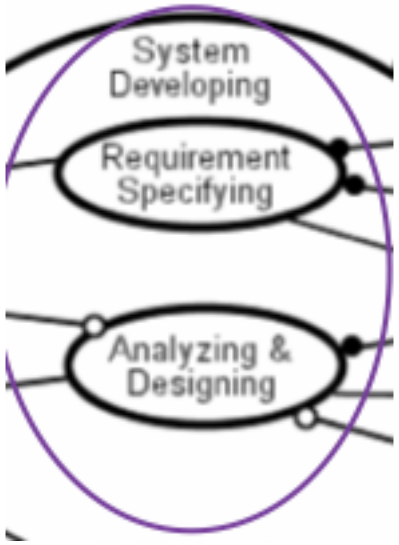
Couple consists of **Man** and **Woman**.

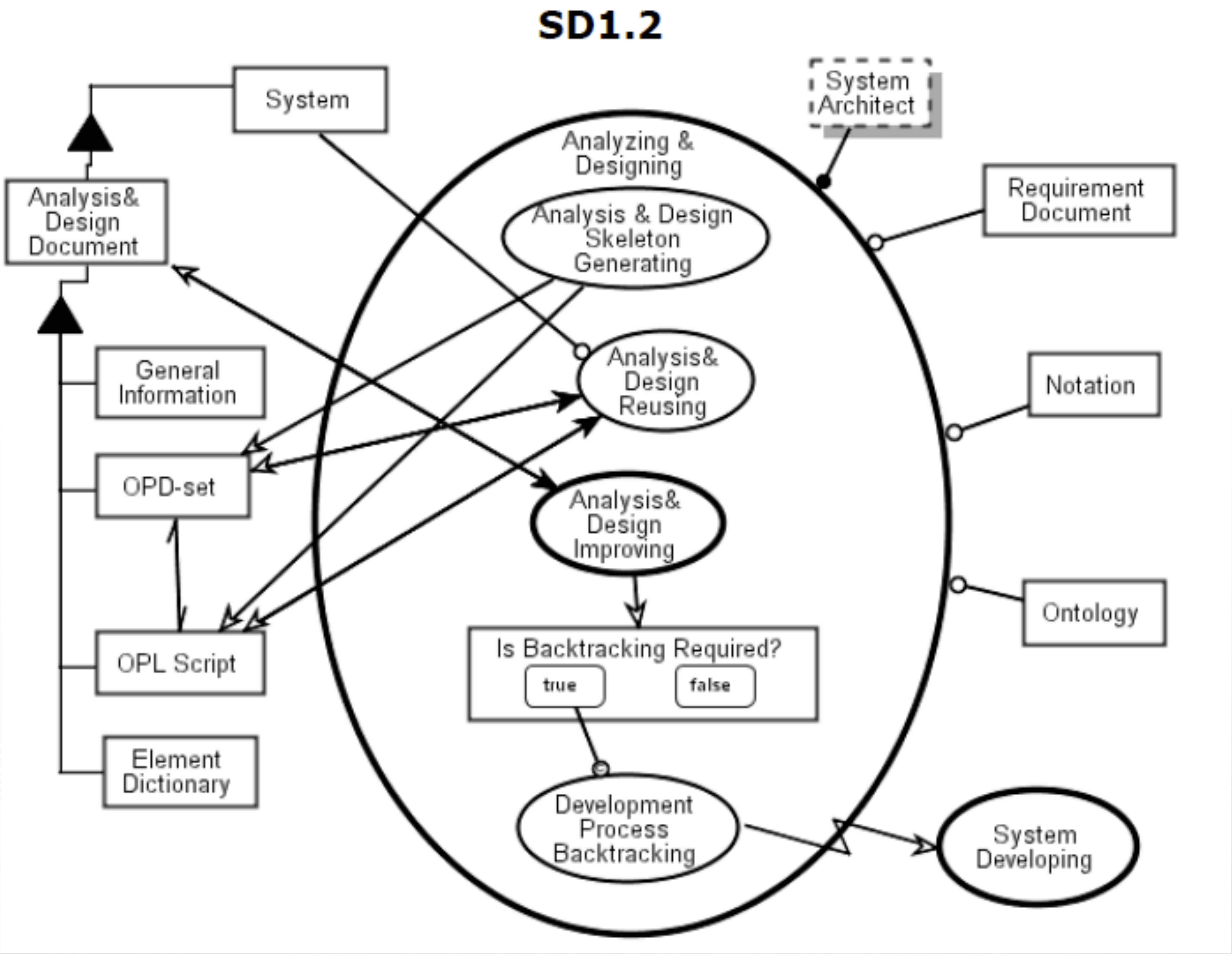
SD1

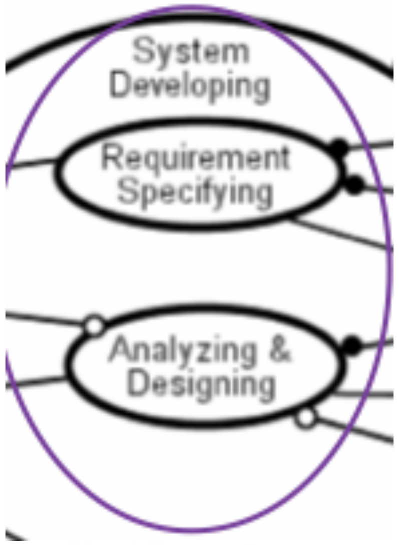

Deliverable
System
Ontology
User
System Developing
Requirement Specifying
Client
Requirement Document
System Architect
Analyzing & Designing
Analysis& Design Document
Notation
Implementing
Implementation
Using & Maintaining
Implementer

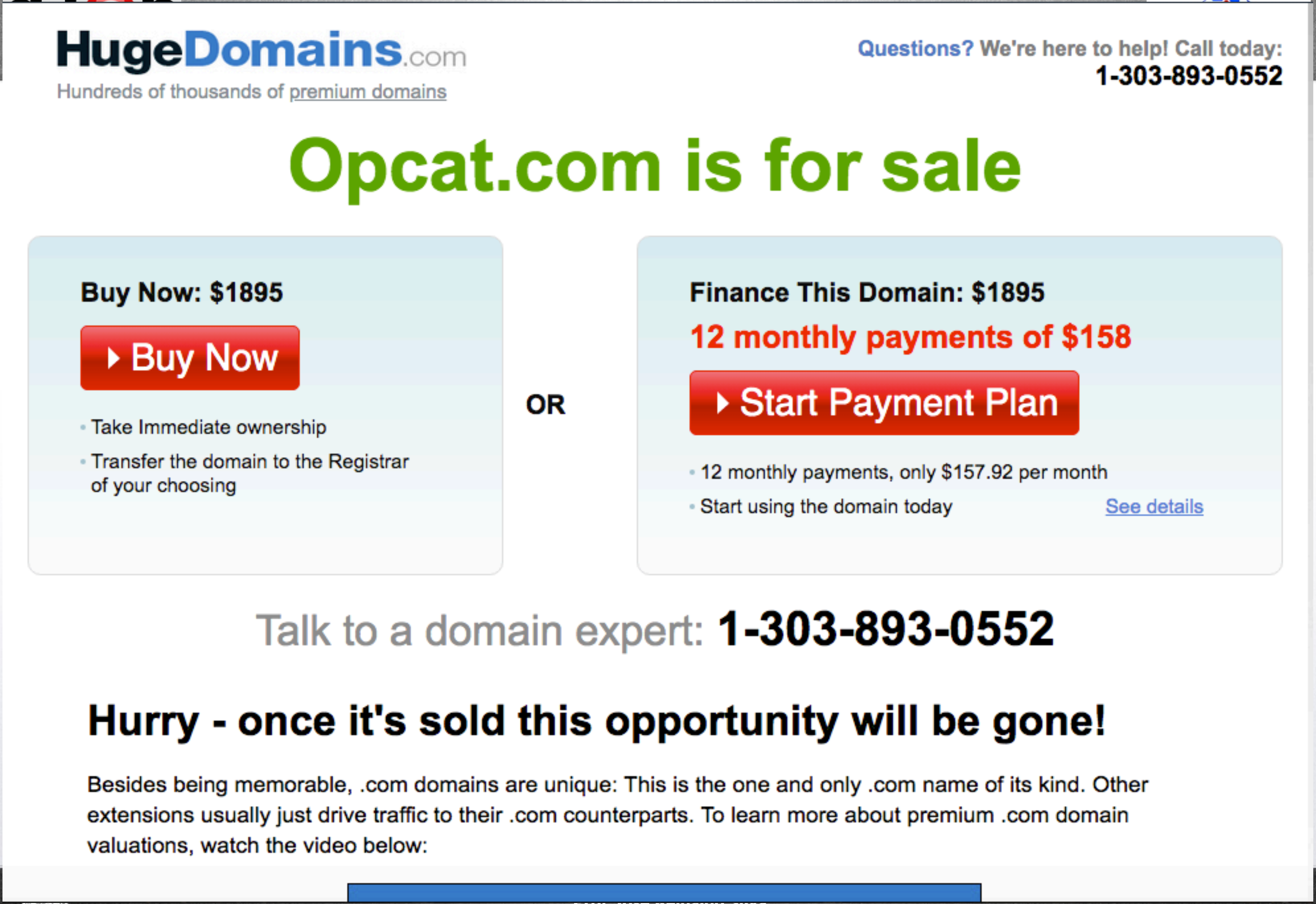
SD1.1


System
Requirement Document
Problem Definition
m
Requirement
Requirement Specifying
Problem Defining
Requirement Reusing
Requirement Adding
Client
System Architect
Ontology
Is Backtracking Required?
true
false
Development Process Backtracking
System Developing


System Developing
Requirement Specifying
Analyzing & Designing


SD1.2
System
System Architect
Analysis& Design Document
Analyzing & Designing
Analysis & Design Skeleton Generating
Analysis& Design Reusing
Analysis& Design Improving
Is Backtracking Required?
true
false
Development Process Backtracking
System Developing
Requirement Document
Notation
Ontology
General Information
OPD-set
OPL Script
Element Dictionary


System Developing
Requirement Specifying
Analyzing & Designing



HugeDomains.com

Hundreds of thousands of premium domains

Questions? We're here to help! Call today:
1-303-893-0552

Opcat.com is for sale

Buy Now: $1895

- Take Immediate ownership
- Transfer the domain to the Registrar of your choosing

OR

Finance This Domain: $1895

12 monthly payments of $158

- 12 monthly payments, only $157.92 per month
- Start using the domain today

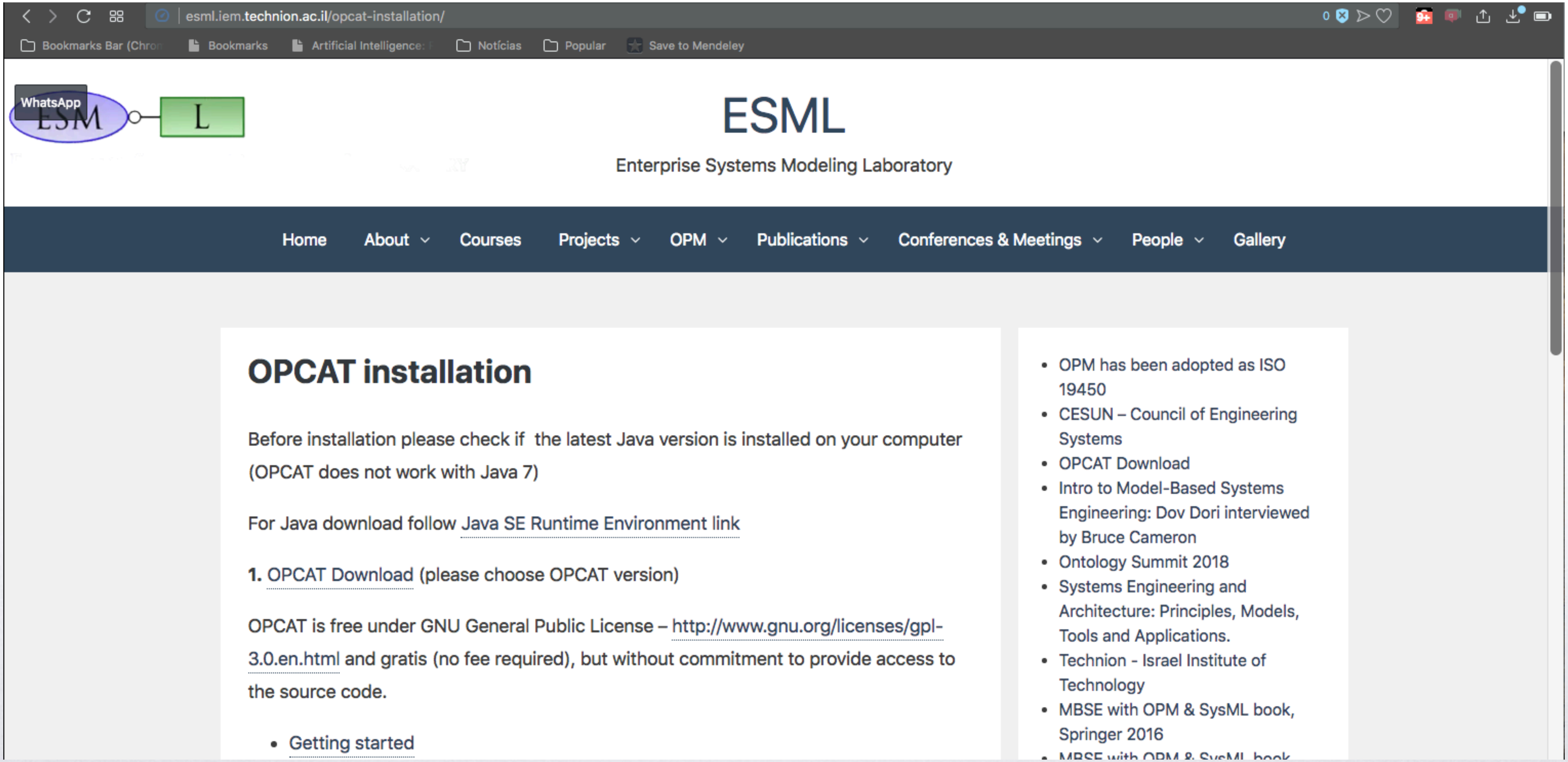
See details

Talk to a domain expert: **1-303-893-0552**

Hurry - once it's sold this opportunity will be gone!

Besides being memorable, .com domains are unique: This is the one and only .com name of its kind. Other extensions usually just drive traffic to their .com counterparts. To learn more about premium .com domain valuations, watch the video below:

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



ESML

Enterprise Systems Modeling Laboratory

OPCAT installation

Before installation please check if the latest Java version is installed on your computer (OPCAT does not work with Java 7)

For Java download follow Java SE Runtime Environment link

1. OPCAT Download (please choose OPCAT version)

OPCAT is free under GNU General Public License – http://www.gnu.org/licenses/gpl-3.0.en.html and gratis (no fee required), but without commitment to provide access to the source code.

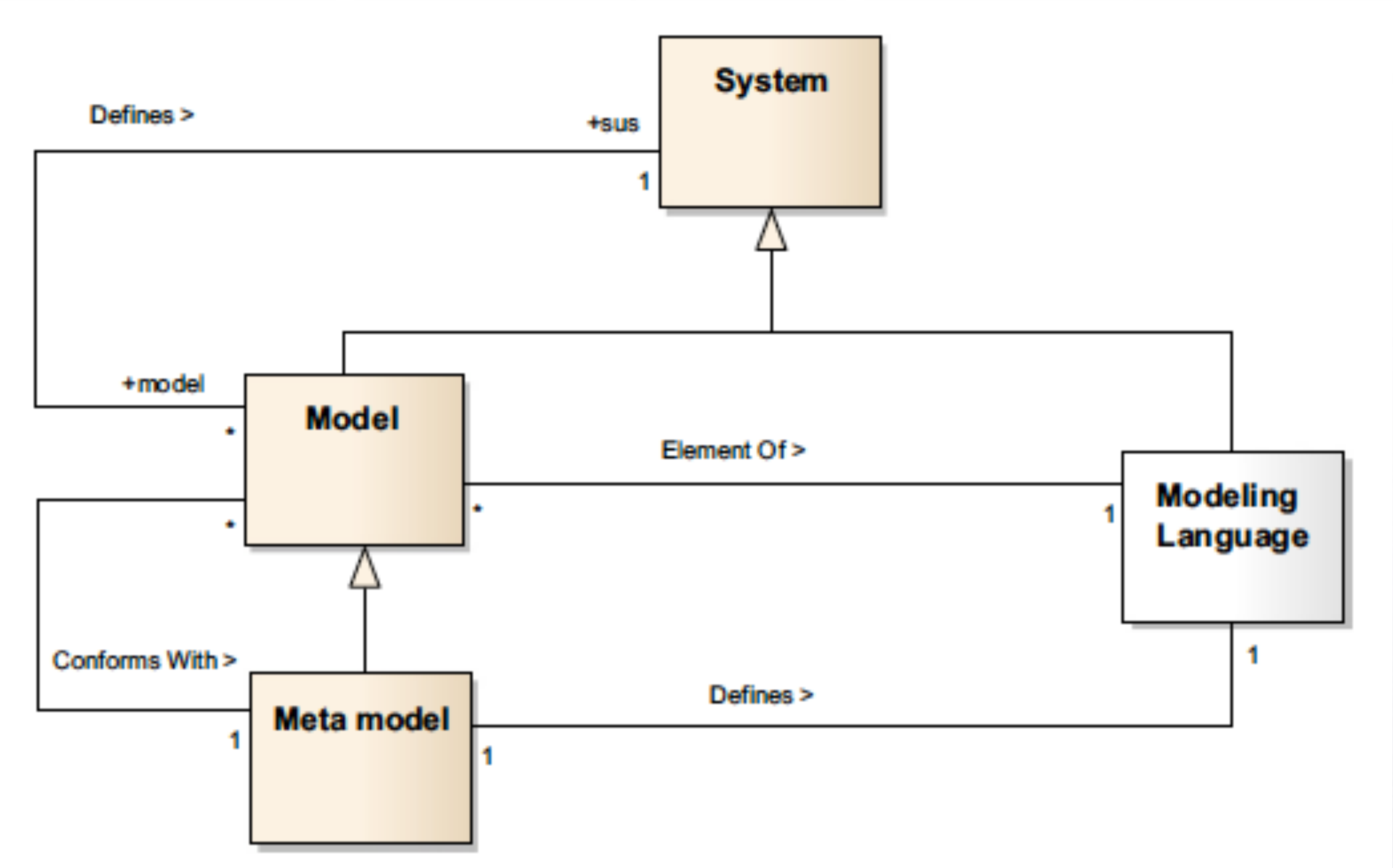
- Getting started

- OPM has been adopted as ISO 19450
- CESUN – Council of Engineering Systems
- OPCAT Download
- Intro to Model-Based Systems Engineering: Dov Dori interviewed by Bruce Cameron
- Ontology Summit 2018
- Systems Engineering and Architecture: Principles, Models, Tools and Applications.
- Technion - Israel Institute of Technology
- MBSE with OPM & SysML book, Springer 2016
- MBSE with OPM & SysML book

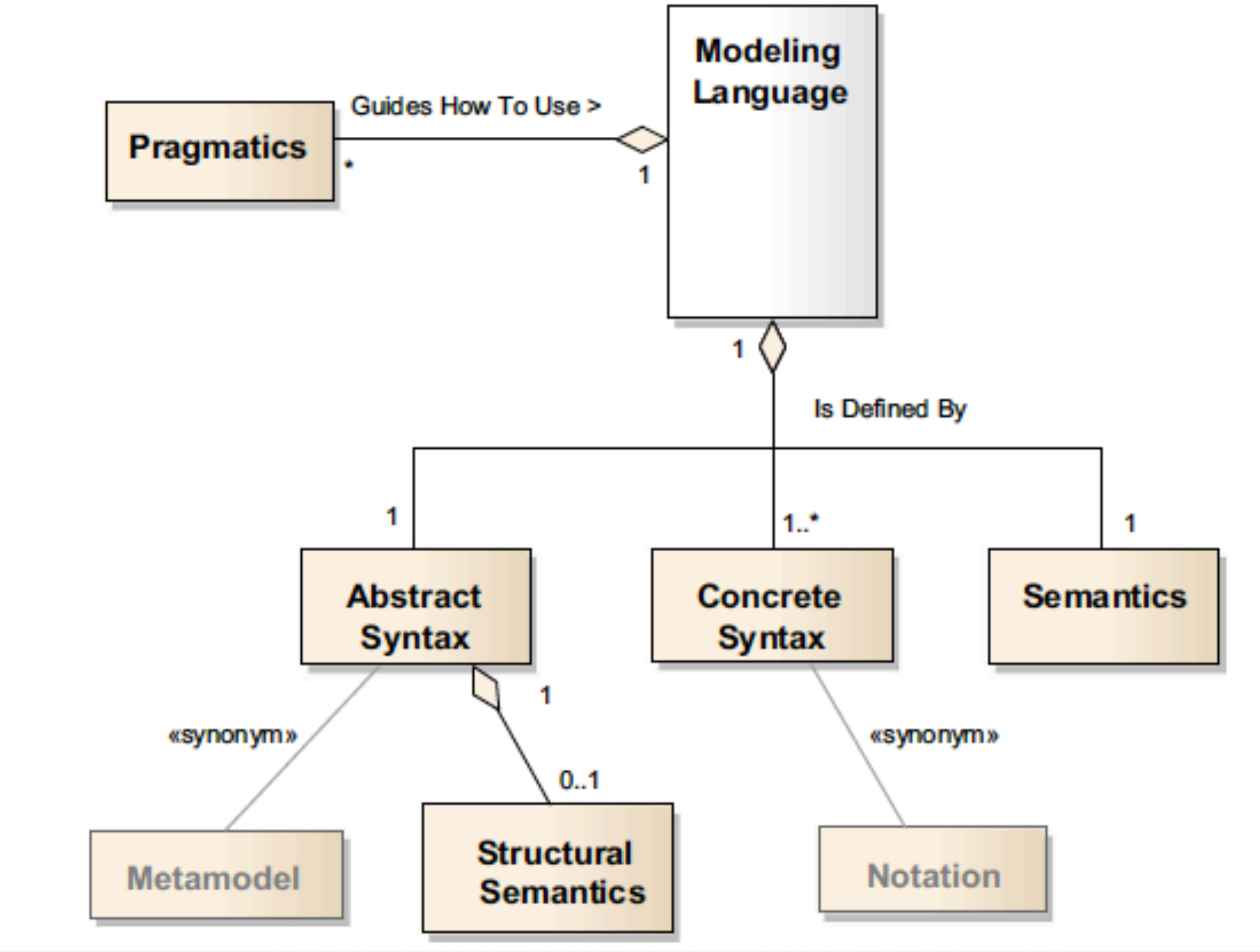
Language, model and meta-model

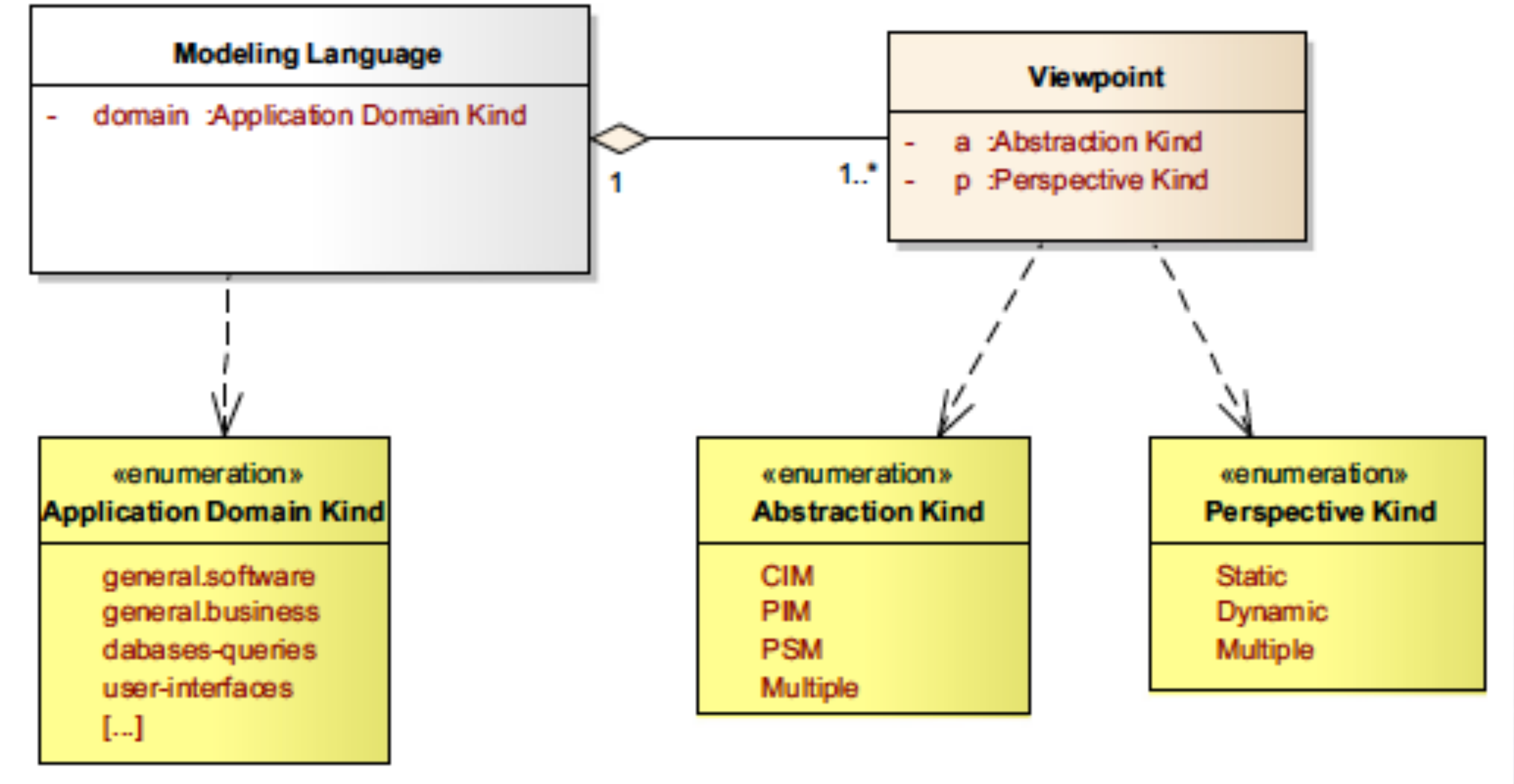
- System

  “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
[...]

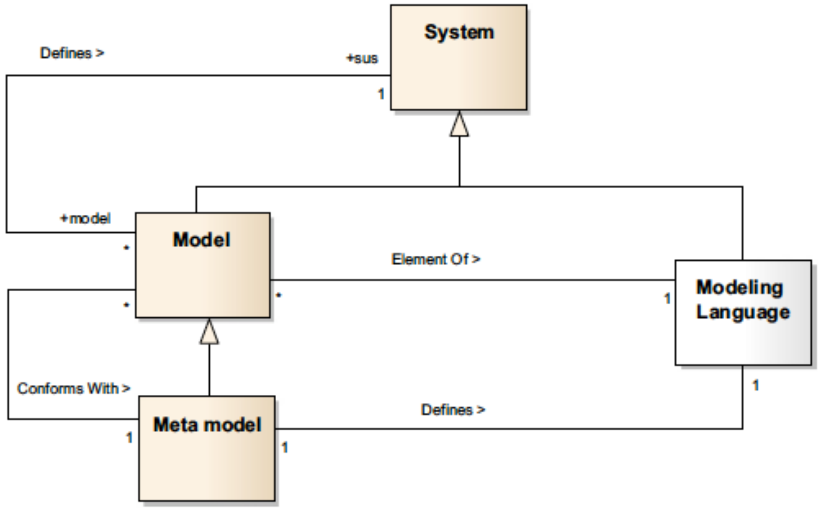
«enumeration»
Abstraction Kind

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

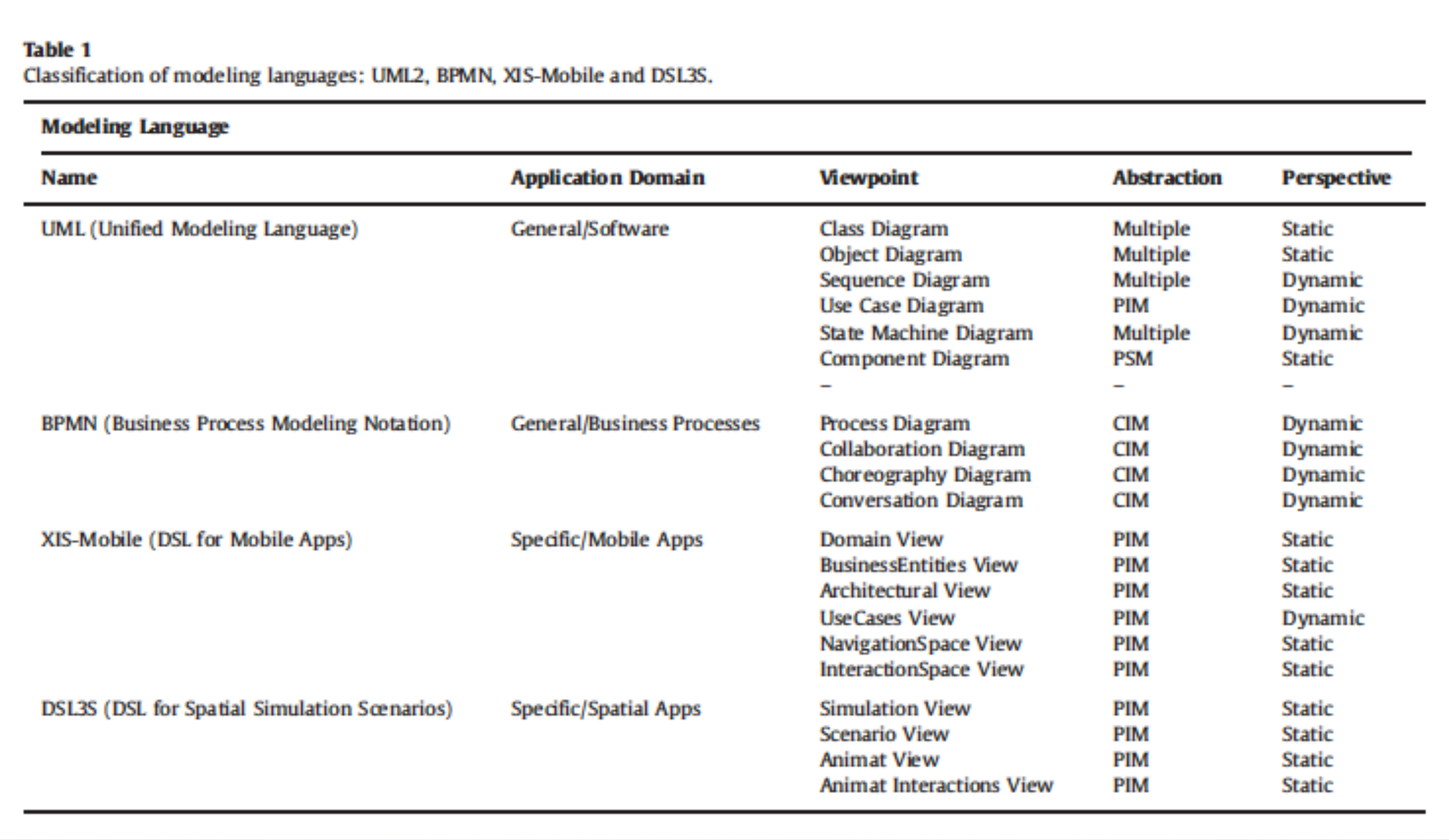
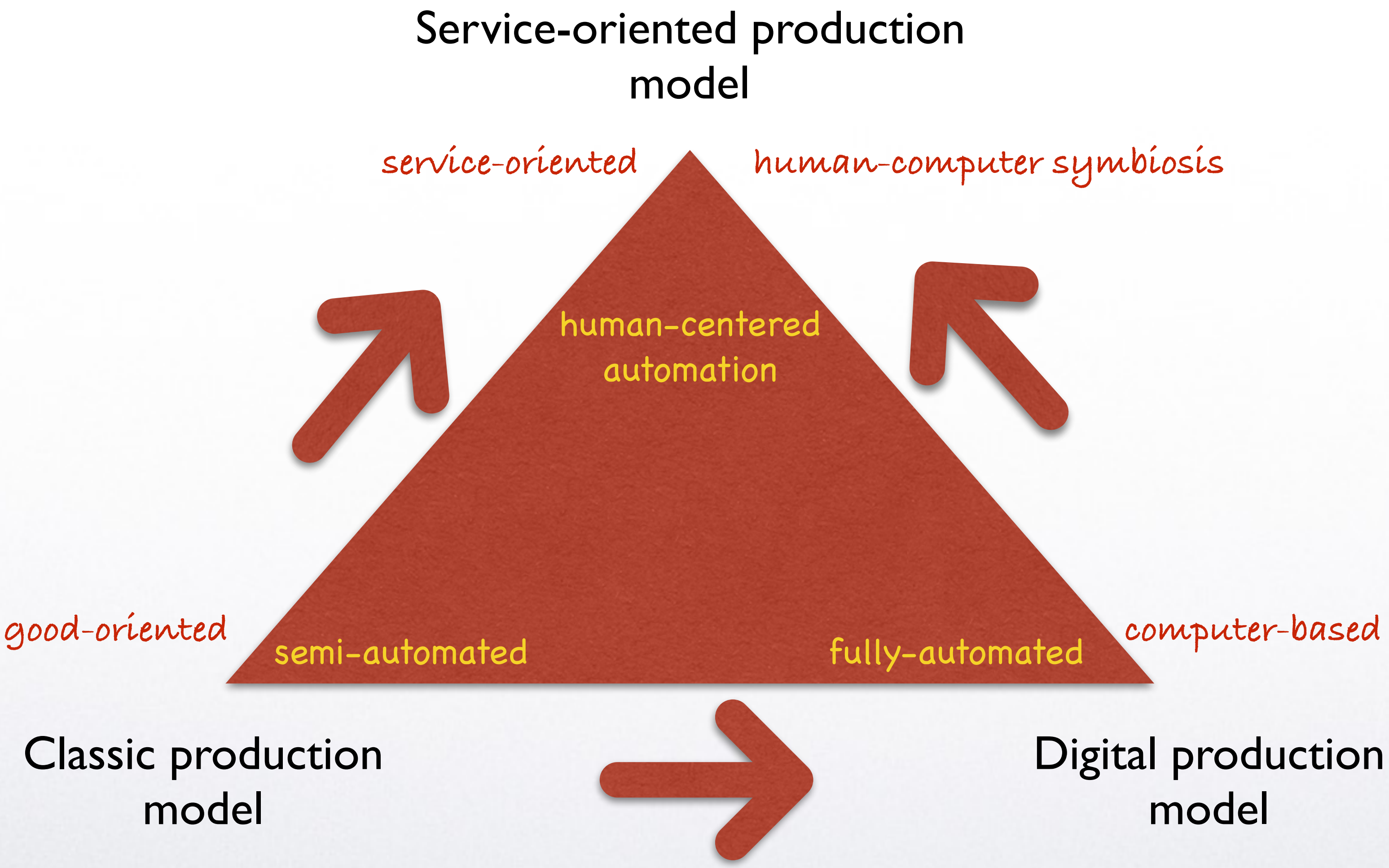


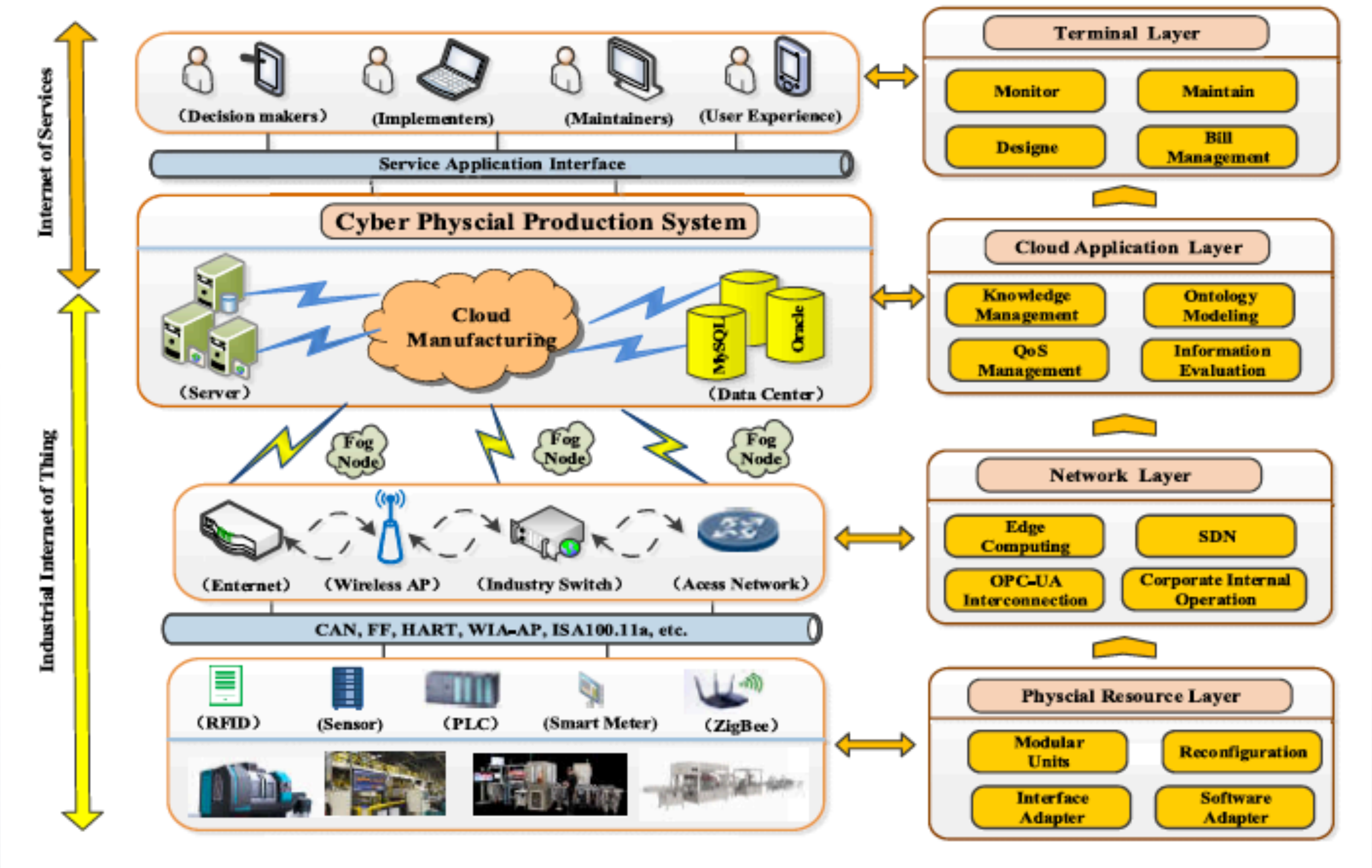
Table 1
Classification of modeling languages: UML2, BPMN, XIS-Mobile and DSL3S.

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

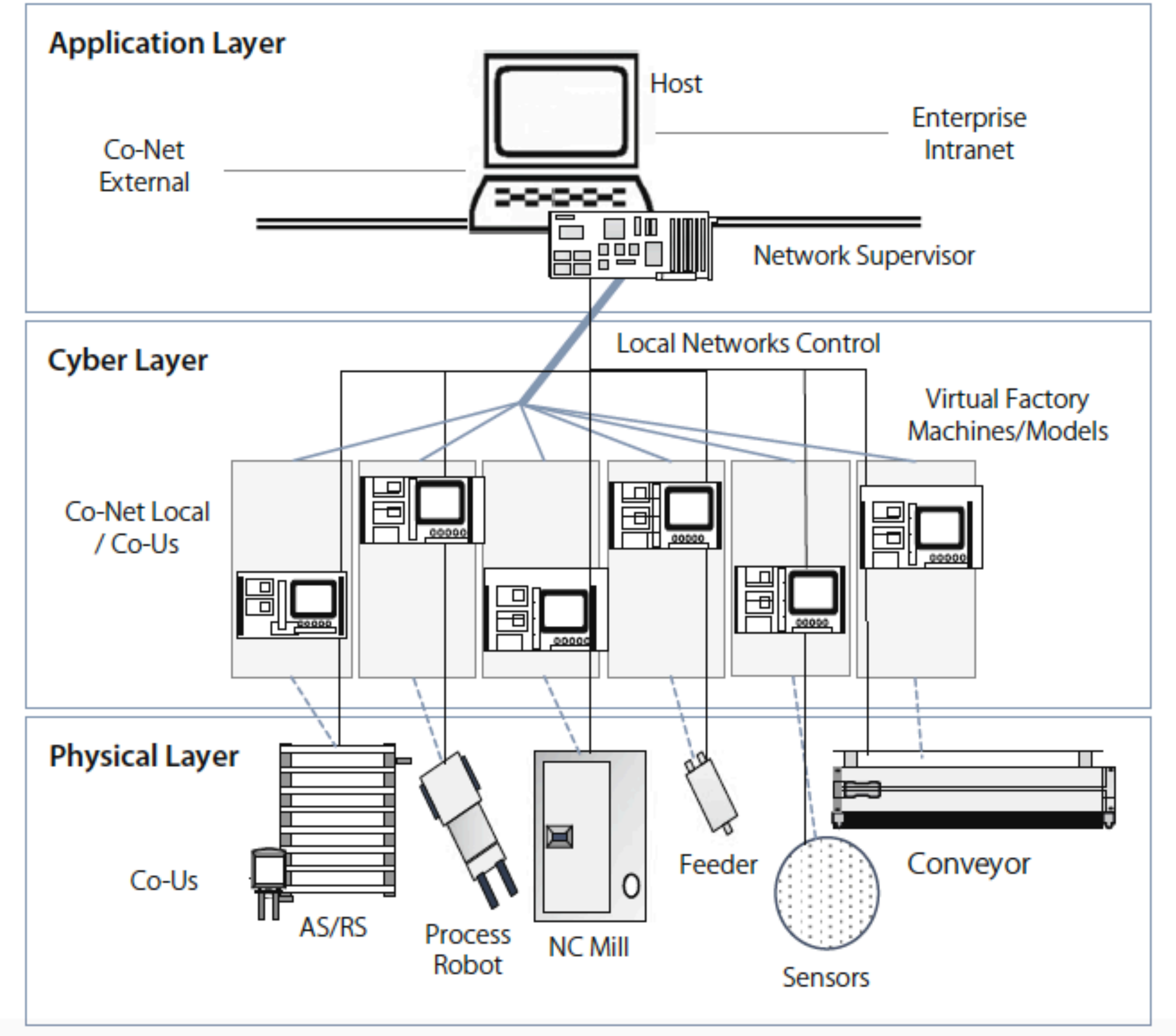
Prospective Applications



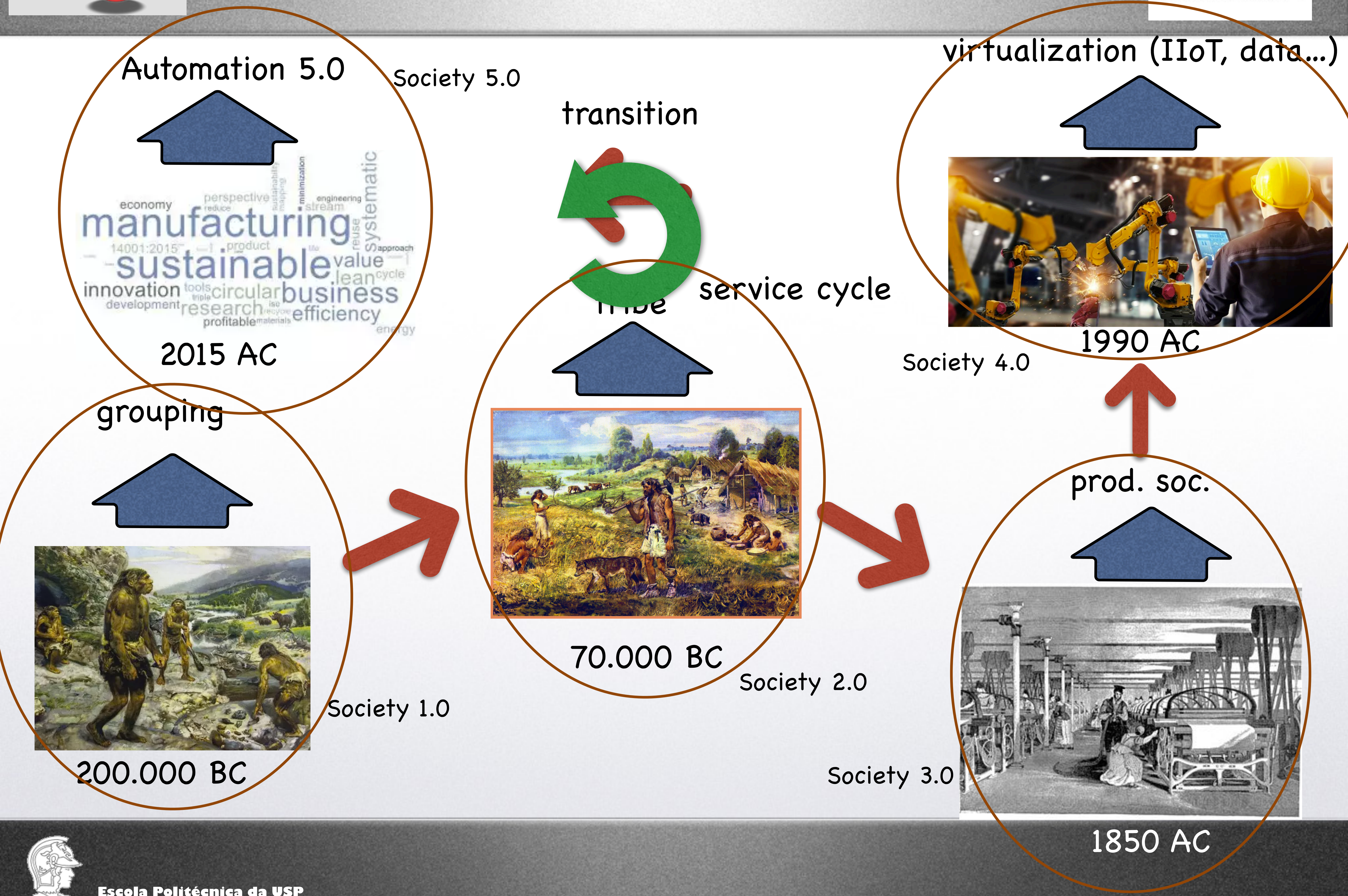
Industry 4.0 Architecture



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



grouping

200.000 BC

Society 1.0

Tribe

70.000 BC

Society 2.0

prod. soc.

1850 AC

Society 3.0

virtualization (IIoT, data...)

1990 AC

Society 4.0

Automation 5.0

2015 AC

Society 5.0

transition

service cycle

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.

AUCOTEC
Create Synergy – Connect Processes

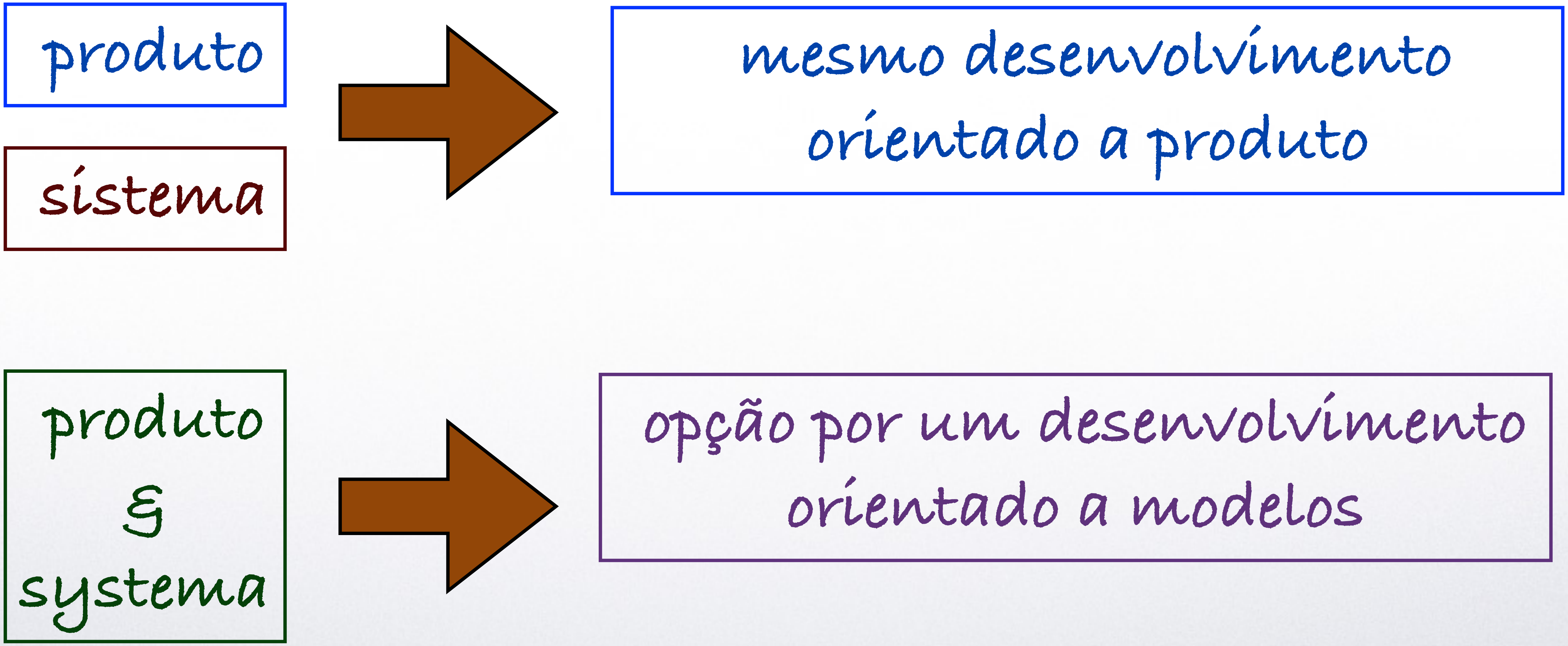
Top 10 Engineering Document and Data Management Challenges

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

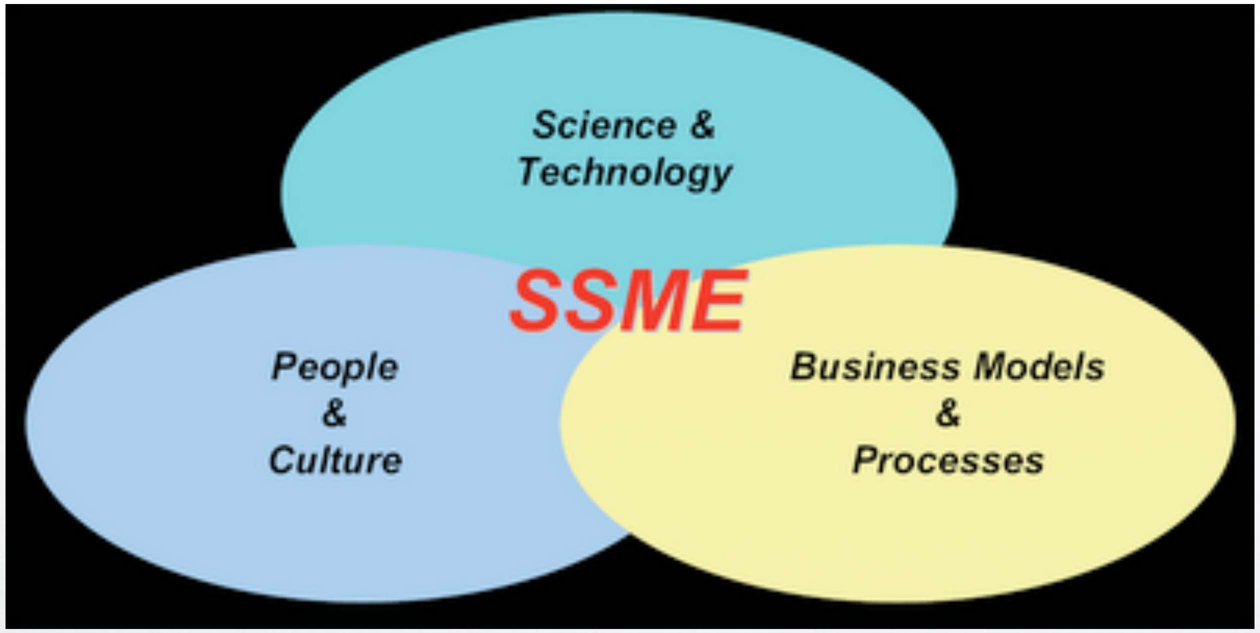
mesmo desenvolvimento orientado a produto

produto & systema

opção por um desenvolvimento orientado a modelos

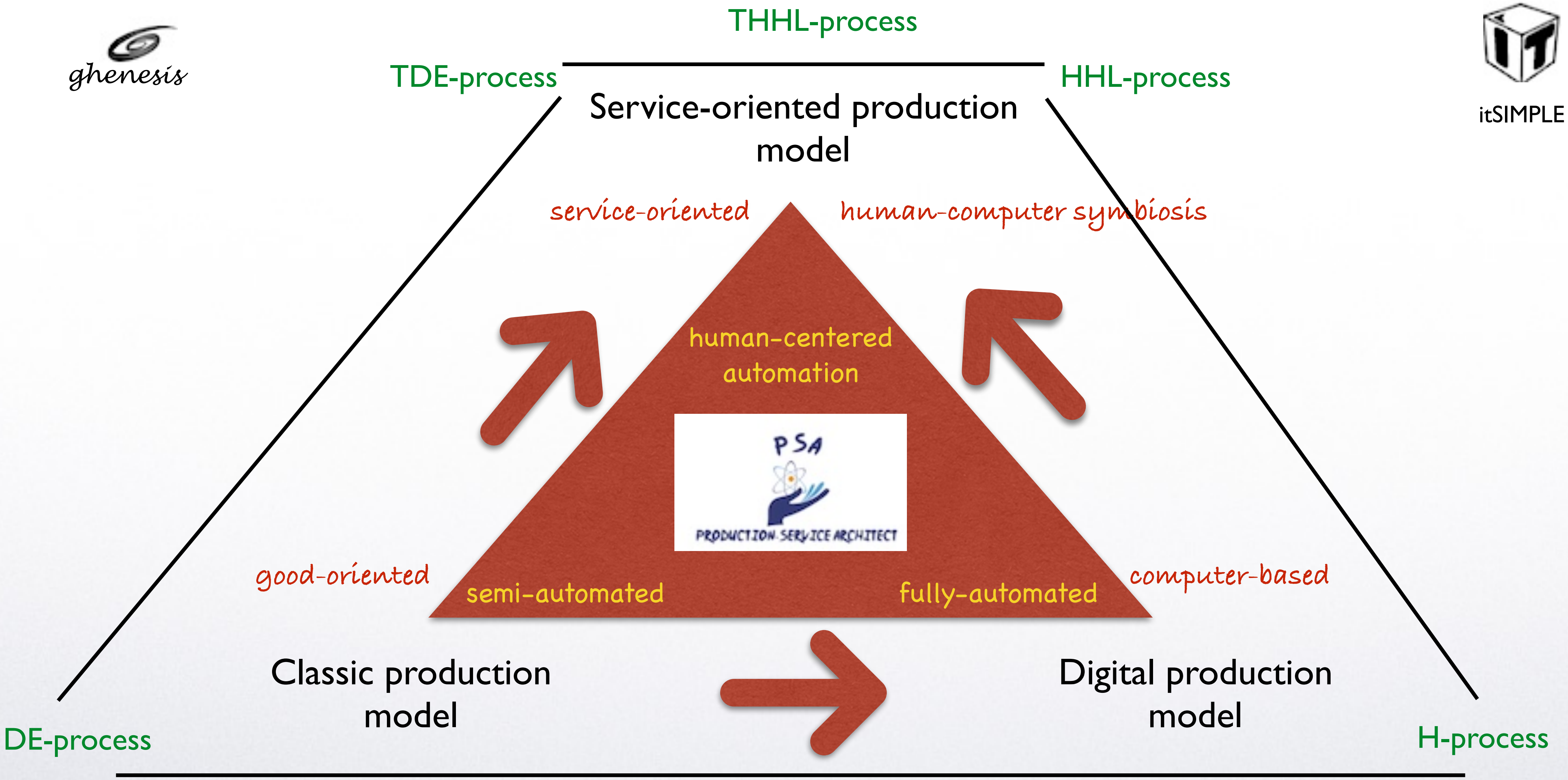
Product-service architecure

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.

Prospective Applications



THHL-process

TDE-process

HHL-process

Service-oriented production model

service-oriented

human-computer symbiosis

human-centered automation

PSA

PRODUCTION SERVICE ARCHITECT

good-oriented

semi-automated

fully-automated

computer-based

Classic production model

Digital production model

DE-process

H-process

itSIMPLE

Acabou o curso…

espero que tenham gostado!

Obrigado

Reinaldo