

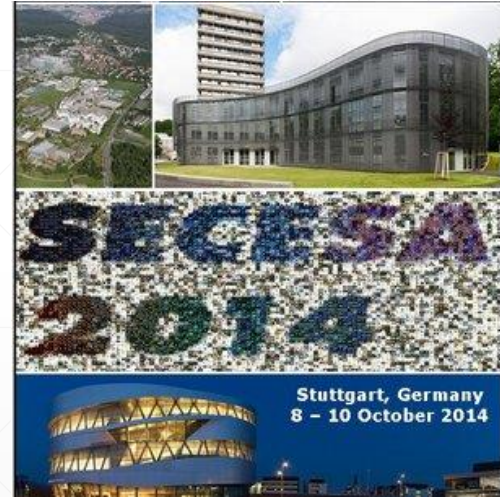
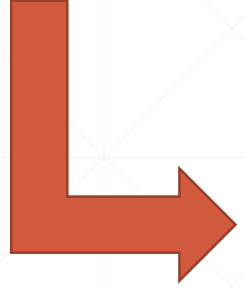
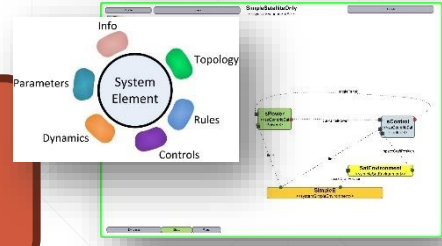




Ph.D. pathway...



Developing a MBx diagram to use CE into a AR Collaborative Environment



Integrating Knowledge with System Models: A Knowledge Based Engineering Approach
 Pathmeswaran Raju (1), Craig B. Chapman (2)

Postgraduation Course:
 Space Engineering and Technology

OPM



Developing AR Collaborative Environment to **MBCE using OPM**





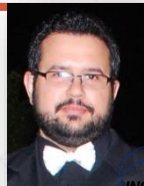
PÓS-GRADUAÇÃO

ENGENHARIA E TECNOLOGIA ESPACIAIS



A Model Based Concurrent Engineering Framework using ISO-19450 Standard

Christopher Shneider Cerqueira



Ana Maria Ambrosio



Claudio Kirner

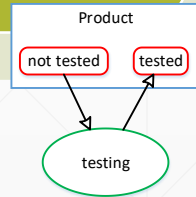




Agenda

[meta-presentation] [presentation of the presentation]

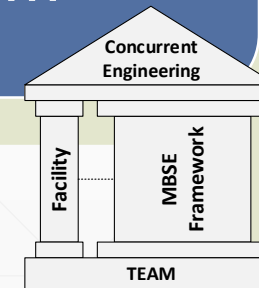
ISO-19450 –
Object Process
Methodology
(OPM)



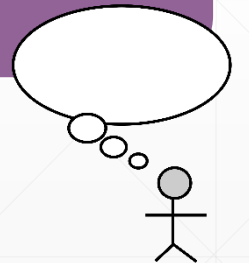
How OPM fits
in CE??



MBCE
proposal using
OPM



Final
Considerations





ISO-19450 - Object Process Methodology (OPM)



OPM

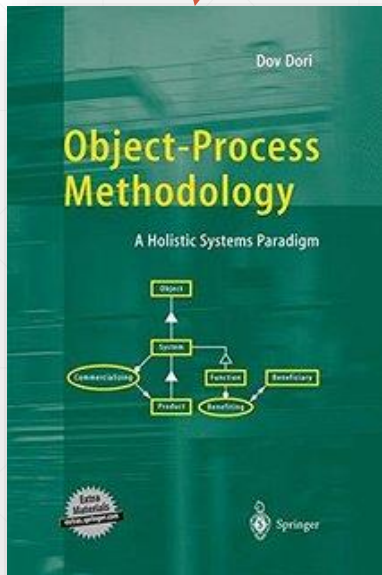
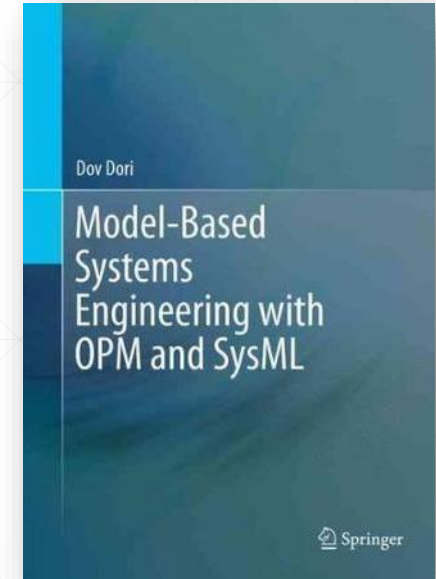
Prof. Dov Dori



Created in 2002

- A Single Diagram – Maps Behaviour and Structure
- 2 Building Blocks and 10 basic relations
- Designed to “Systemic View” and “Concept Design”
- Simulation Ready

improving and showing it applicability



- ~130 Pages standard
- Published in late 2015
- Intended to “Automation Systems and Integration”
- Has the “power” of a ISO seal.

ISO/PAS 19450:2015

Automation systems and integration -- Object-Process Methodology
(Only available in English)

Abstract

[Preview ISO/PAS 19450:2015](#)

ISO/PAS 19450:2015 specifies Object-Process Methodology (OPM) with detail sufficient for enabling practitioners to utilise the concepts, semantics, and syntax of Object-Process Methodology as a modelling paradigm and language for producing conceptual models at various extents of detail, and for enabling tool vendors to provide application modelling products to aid those practitioners.

While ISO/PAS 19450:2015 presents some examples for the use of Object-Process Methodology to improve clarity, it does not attempt to provide a complete reference for all the possible applications of Object-Process Methodology.

FORMAT LANGUAGE

PDF English

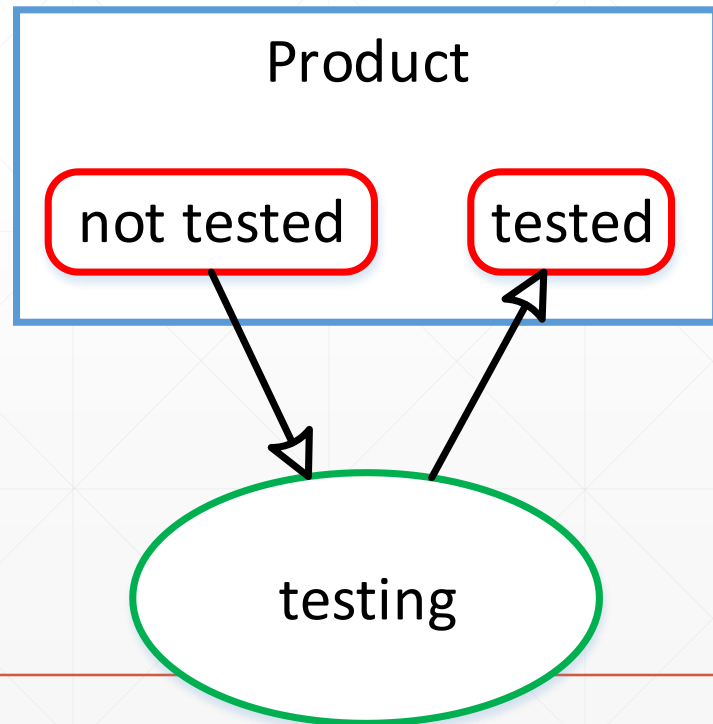
HTML English

CHF 198 [Add to basket](#)



OPM has only two building blocks (things):

1. **Objects with states**
2. **Processes**



All the other OPM elements are relations (links) between things.

Structural

- Relational
- Aggregation
- Exhibition
- Specialization
- Instantiation

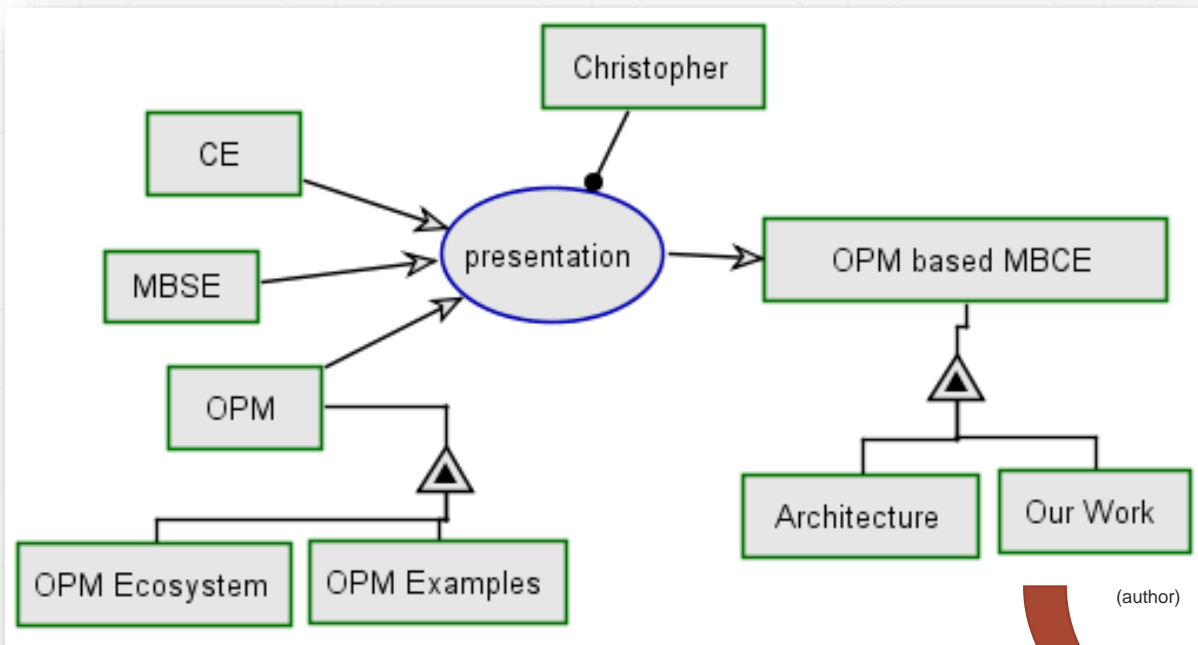
Procedural

- Transforming
 - Consume/Create/Effect
- Enabling
 - Agent/Instrument
- Control
 - Events/Conditions
 - Invocations/Exceptions
 - Multiplicity
 - Logical



OPM has two simultaneous cognitive channels: visual-OPD and textual-OPL

Diagram (OPD)



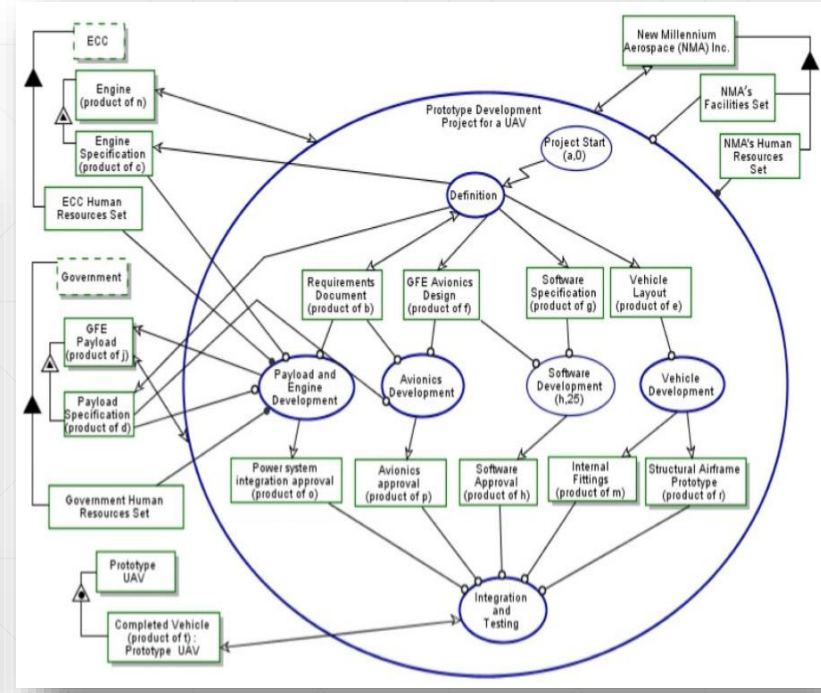
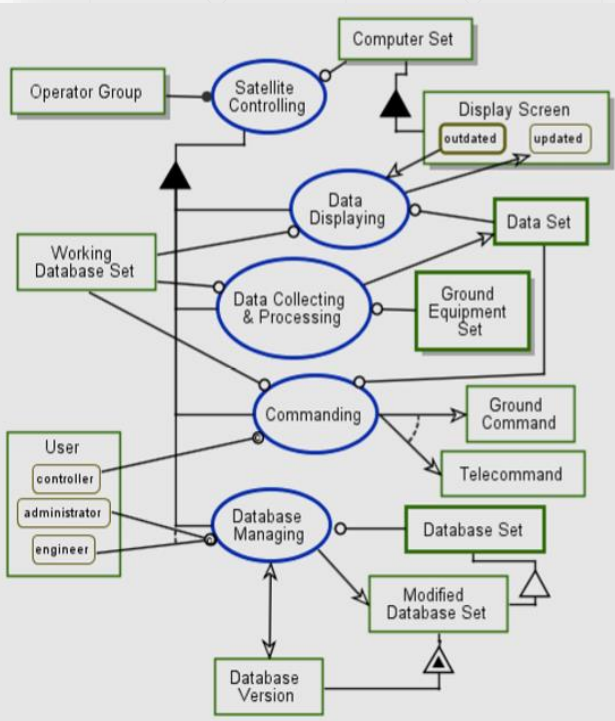
Textual (OPL) – auto generated

- Christopher handles presentation.
- OPM exhibits OPM Examples and OPM Ecosystem.
- OPM based MBCE exhibits Architecture and Our Work.
- presentation consumes OPM, CE, and MBSE.
- presentation yields OPM based MBCE.



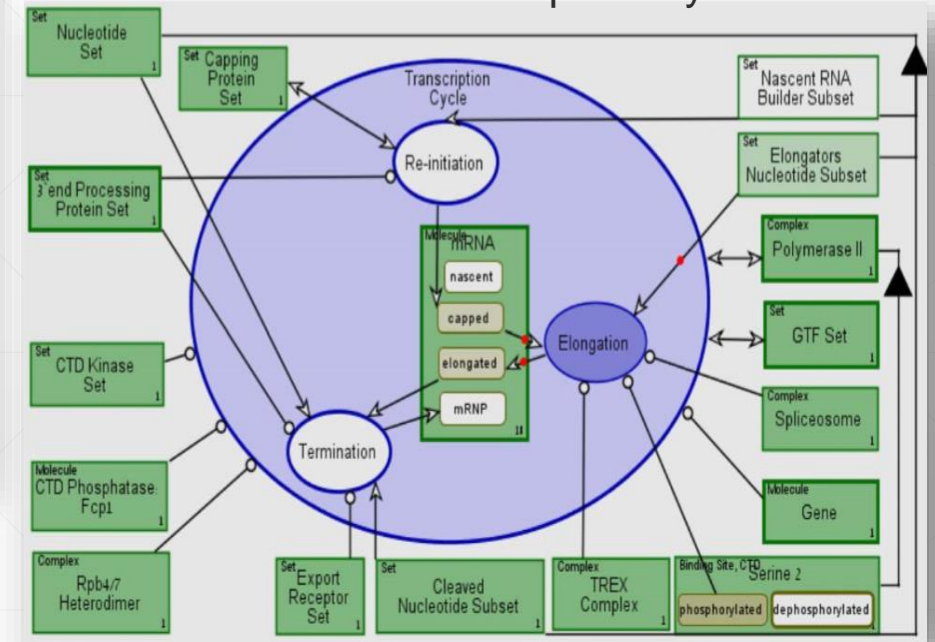
3 Use Examples:

Model-Based Guidelines for User-Centric **Satellite Control Software Development**



Project Management vs. Systems Engineering **Management:** A Practitioners' View on Integrating the Project and Product Domains

Conceptual Model-Based Systems **Biology:** Mapping Knowledge and Discovering Gaps in the mRNA Transcription Cycle

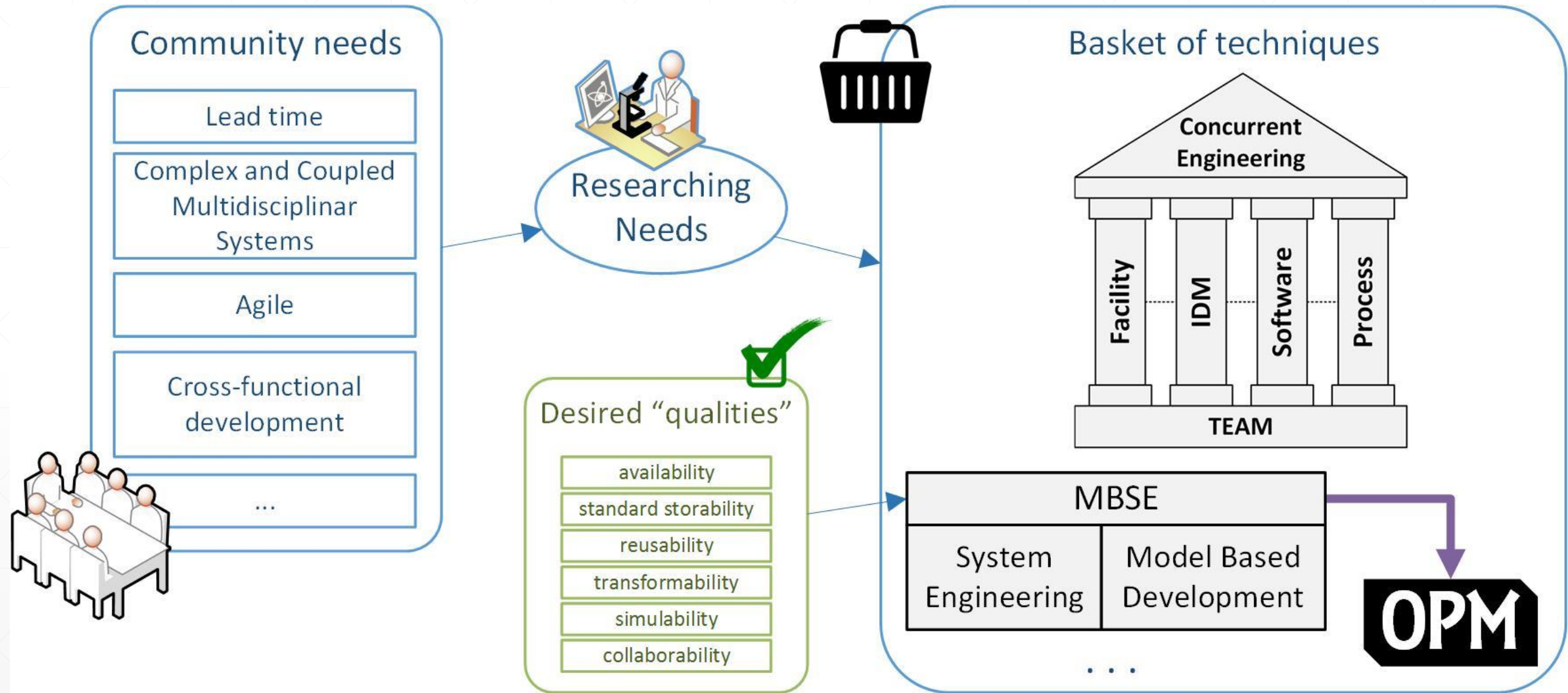




How OPM fits in CE??

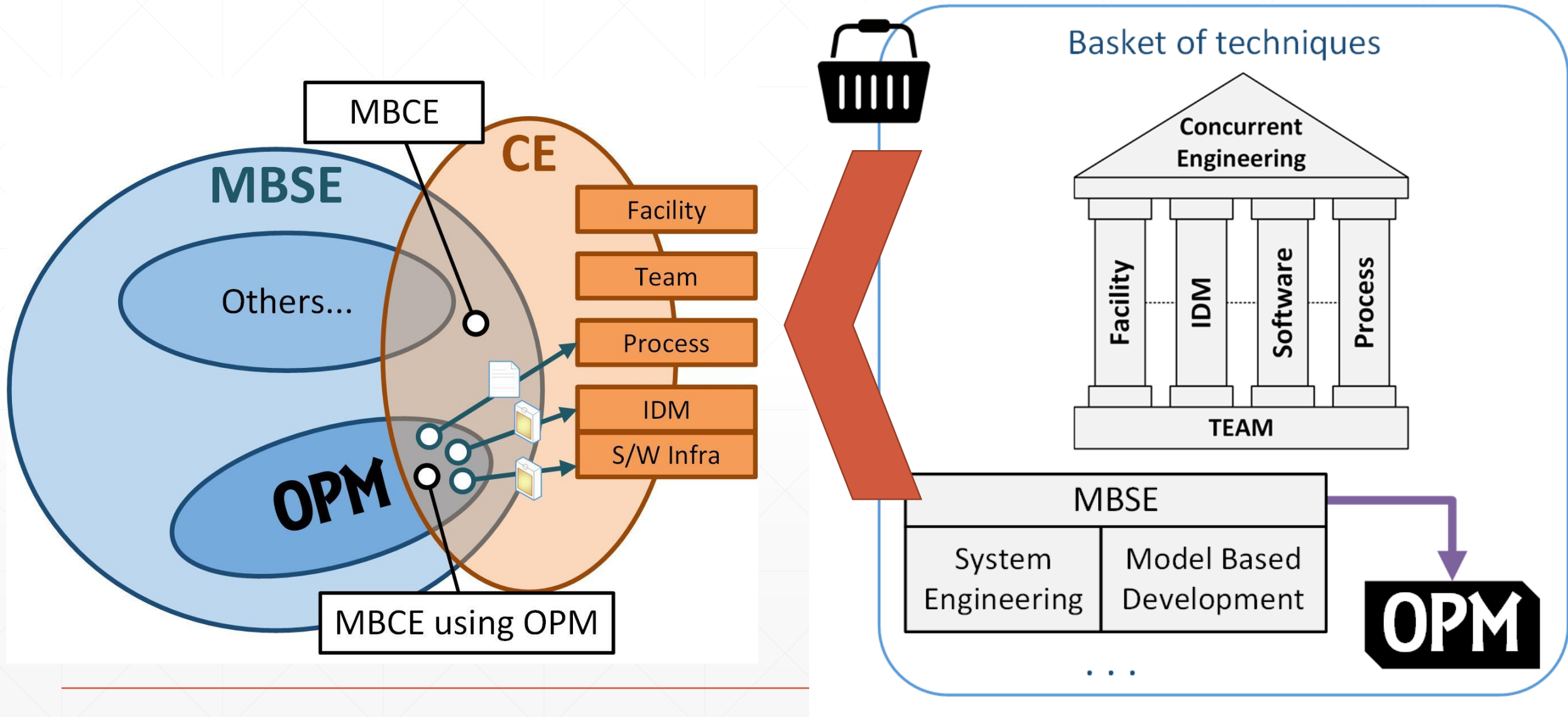


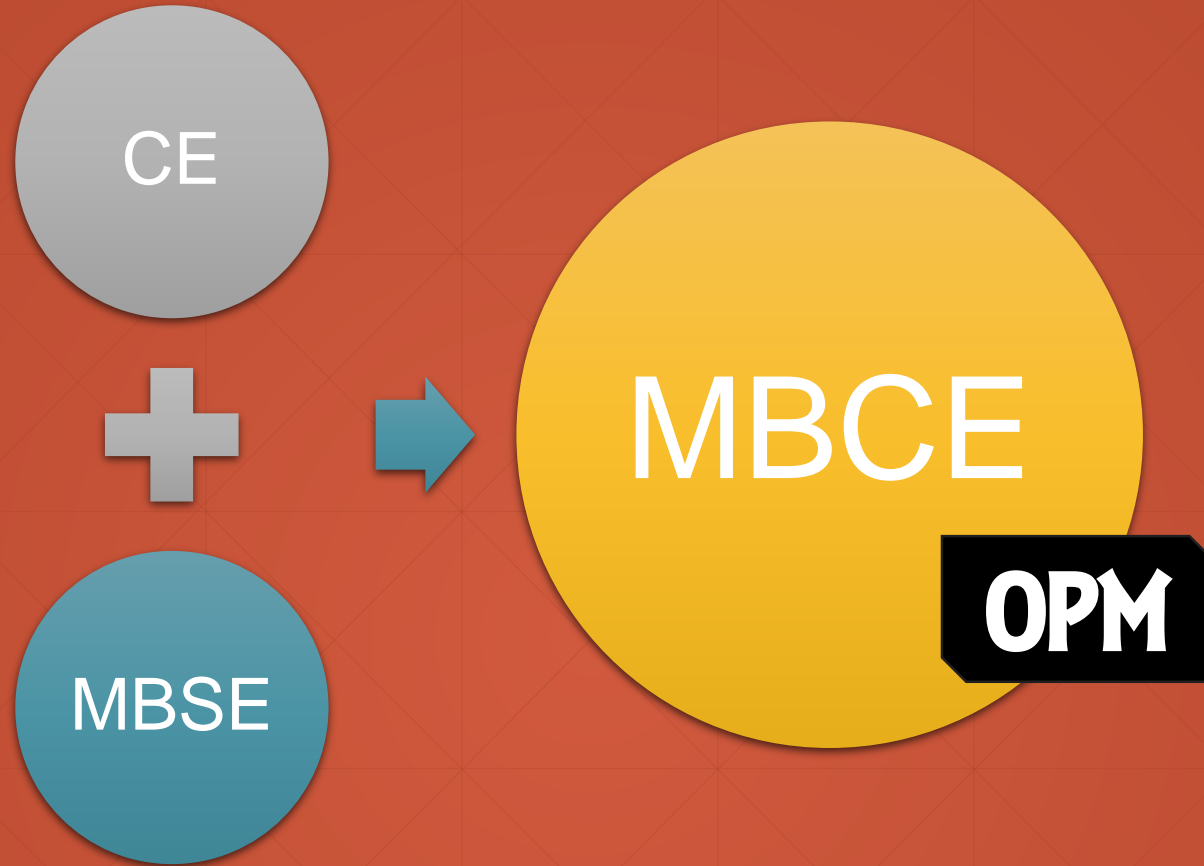
briefly ... Our proposal blends CE with MBSE using OPM





briefly ... Our proposal blends CE with MBSE using OPM

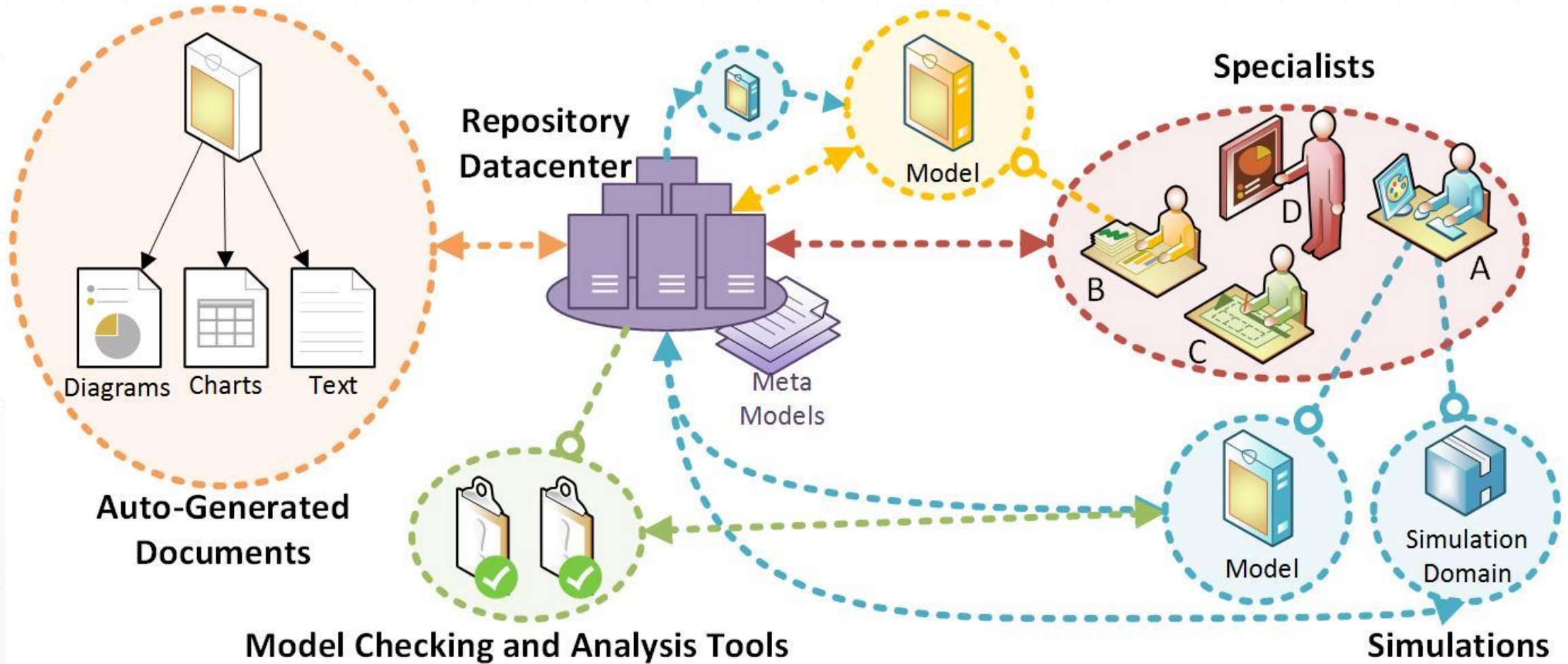




MBCE proposal using OPM



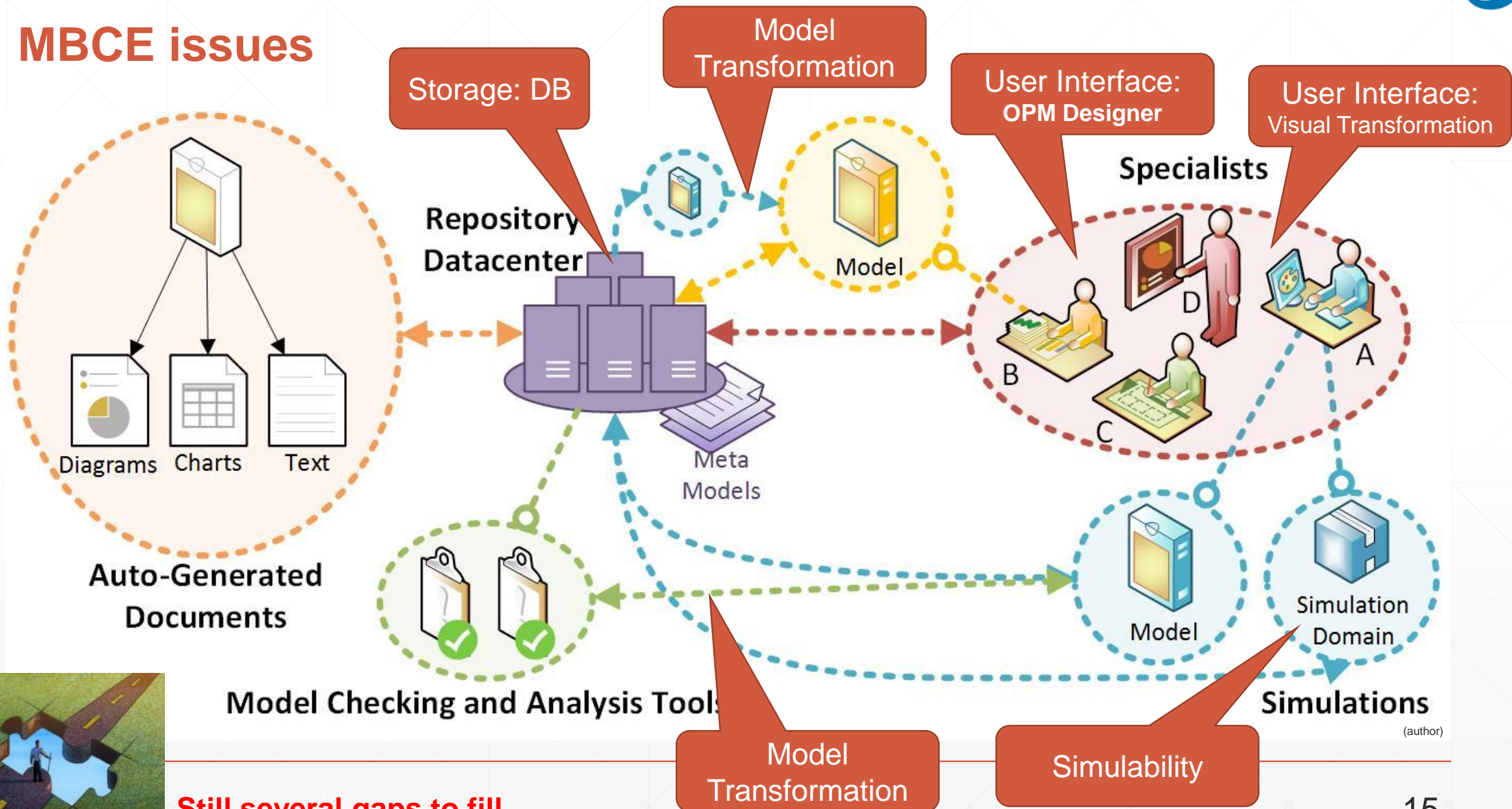
MBCE architecture



(author)



MBCE issues



Still several gaps to fill

(author)



Storage: Database – using graph approach

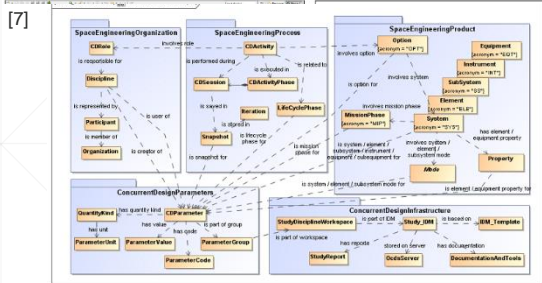
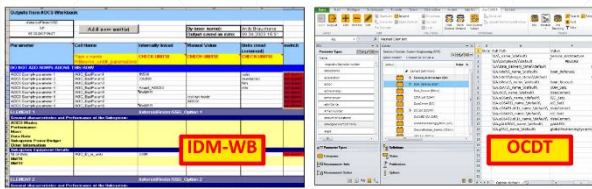
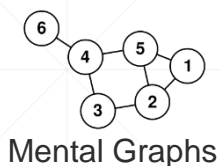


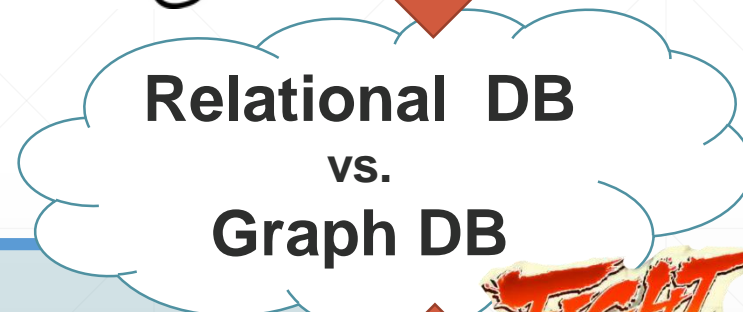
Figure 4-3 SEIM main information object types and relationships [ECSS-E-TM-10-23]



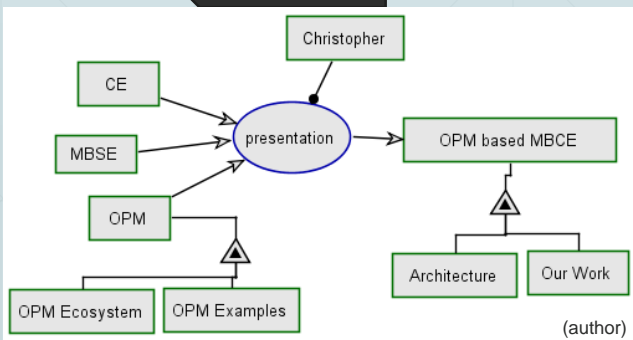
Parameter Relationship



PostgreSQL



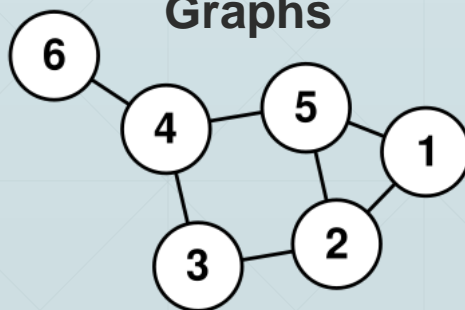
OPM



(author)



Graphs

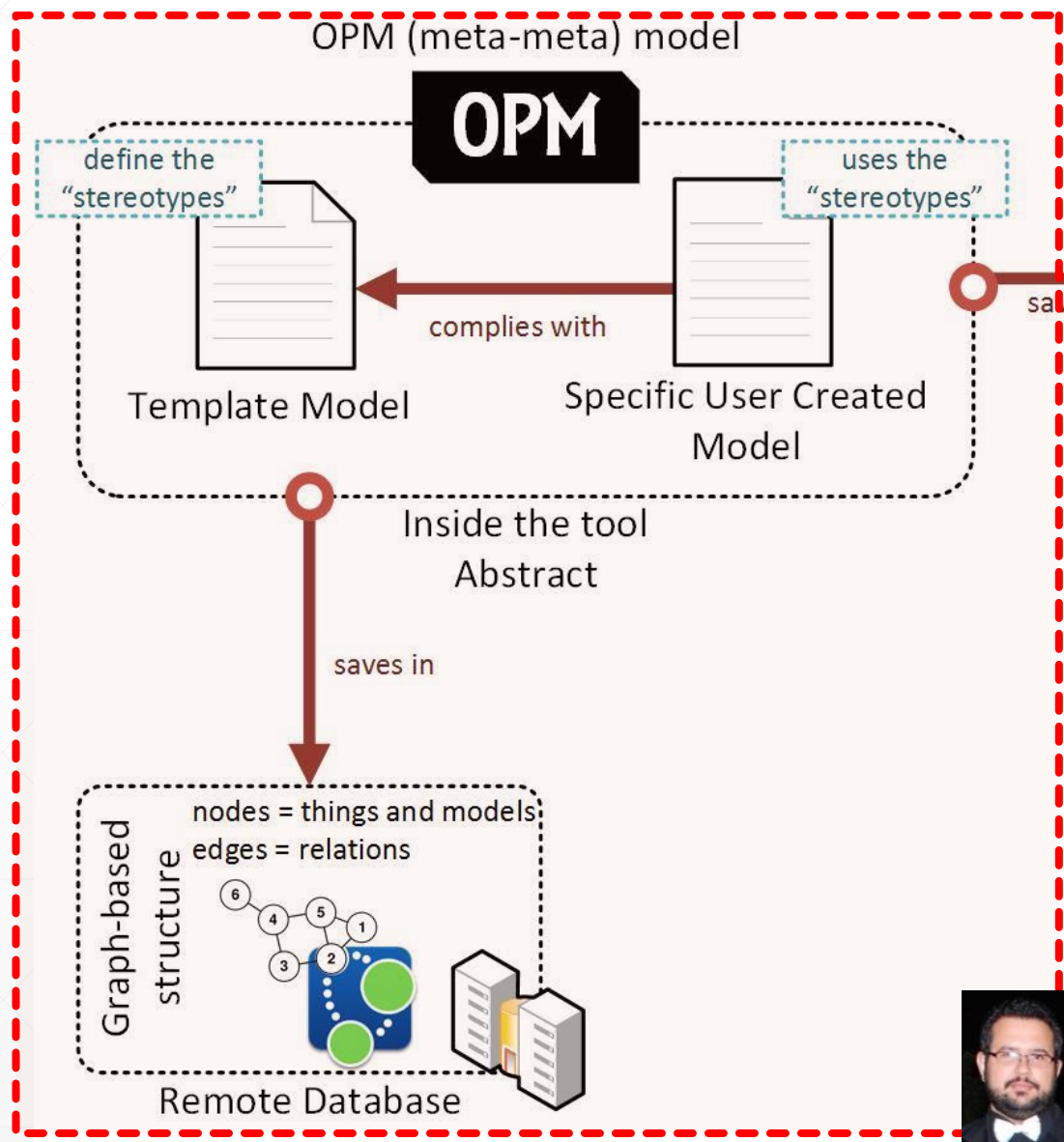


Neo4J

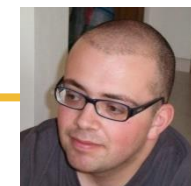




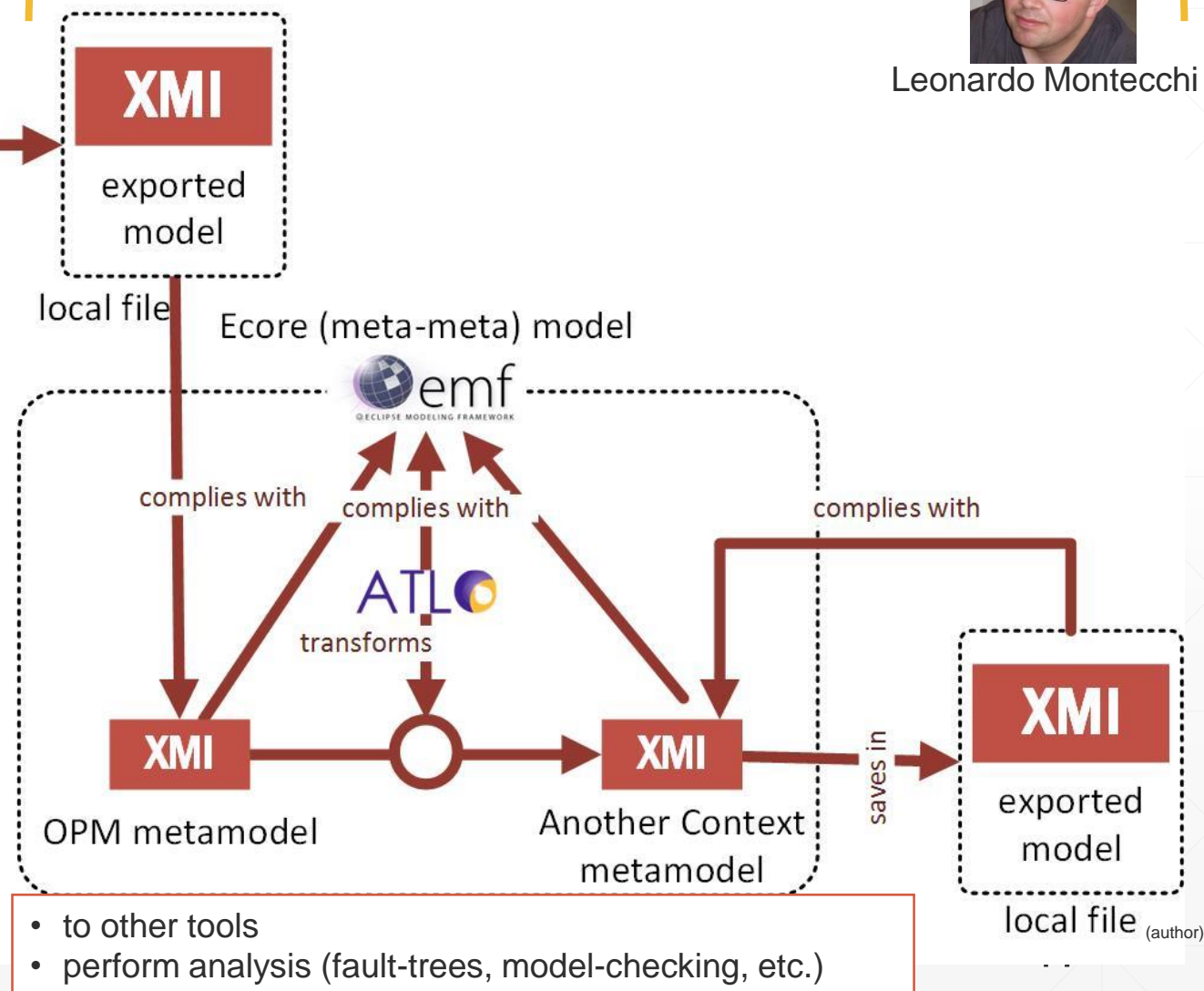
Model Transformation



further collaboration with the [Università degli Studi di Firenze](#)

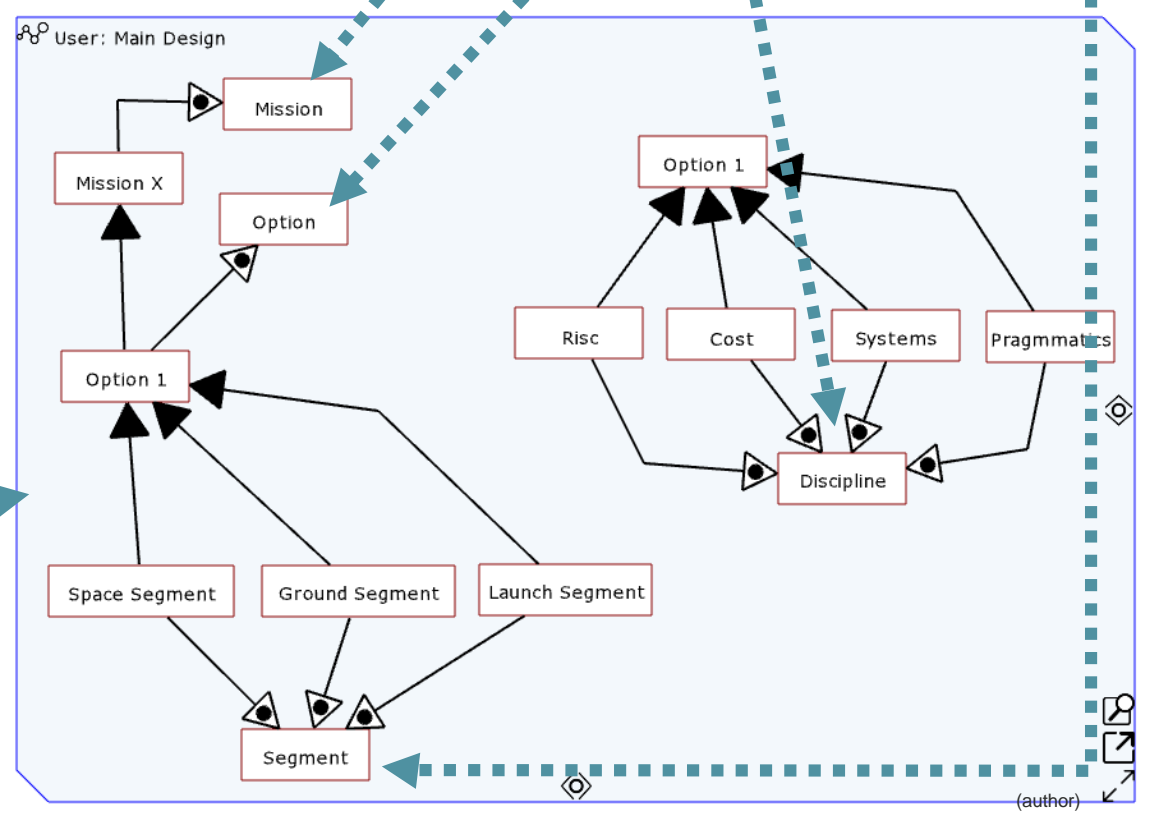
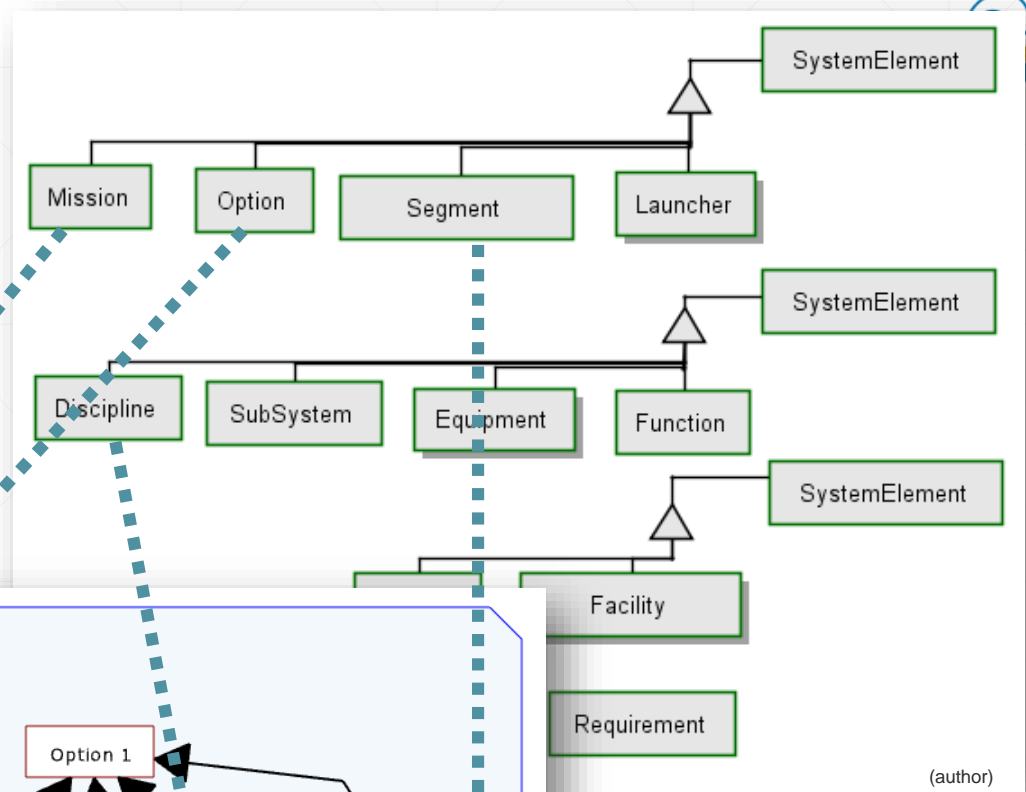
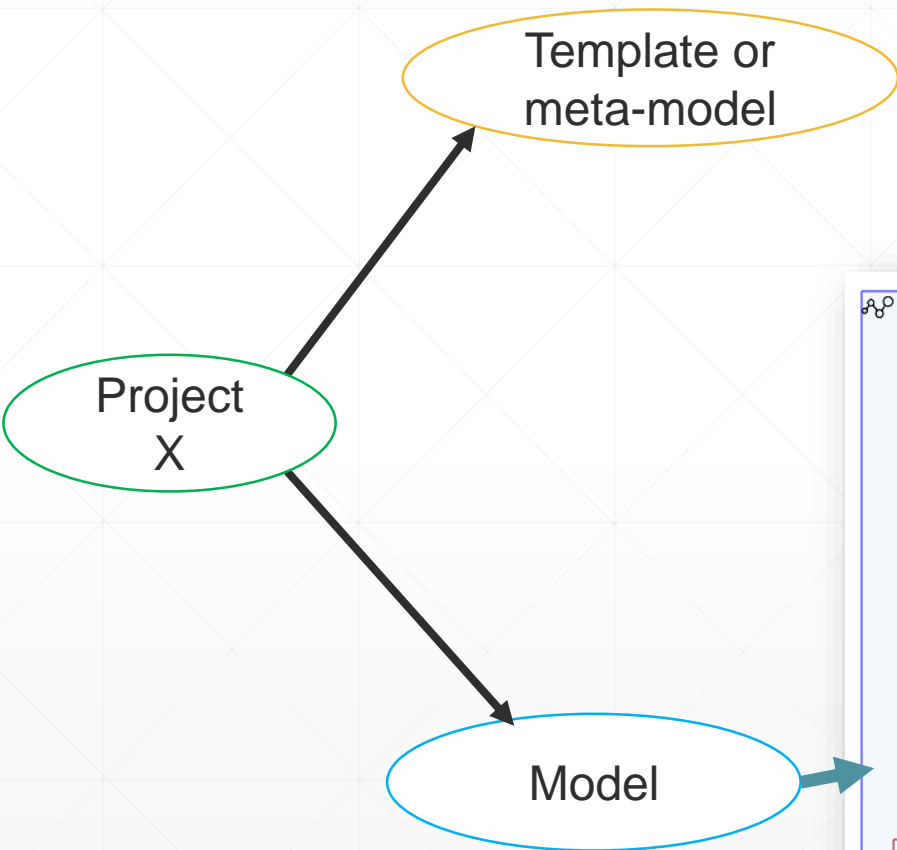


Leonardo Montecchi





Model Transformation - OPM <<Template>> vs. Model



(author)

(author)



User Interface: OPM Designer

- *Document-centric model-based* user-interface tool
 “works directly into the final document”



- Textual/Parametric
- Web/**Local** based



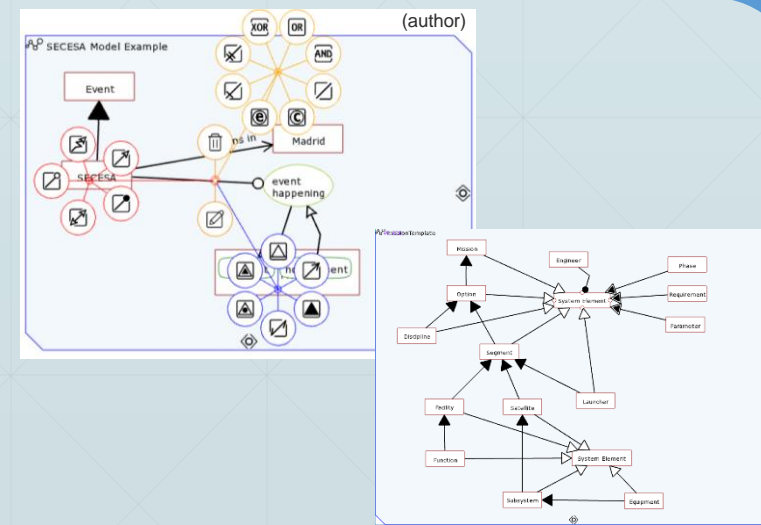
[7]

- *Model-centric model-based* user-interface tool
 “works into models”



- Diagrammatic
- Runs transformations using plain Java (*further collaboration*)
- **OPM Editor** - Web/Local based in JavaScript

- **OPCat** is a Prof. Dori (nowadays only) available **free option**.





User Interface: Visual Transformation

SECESA

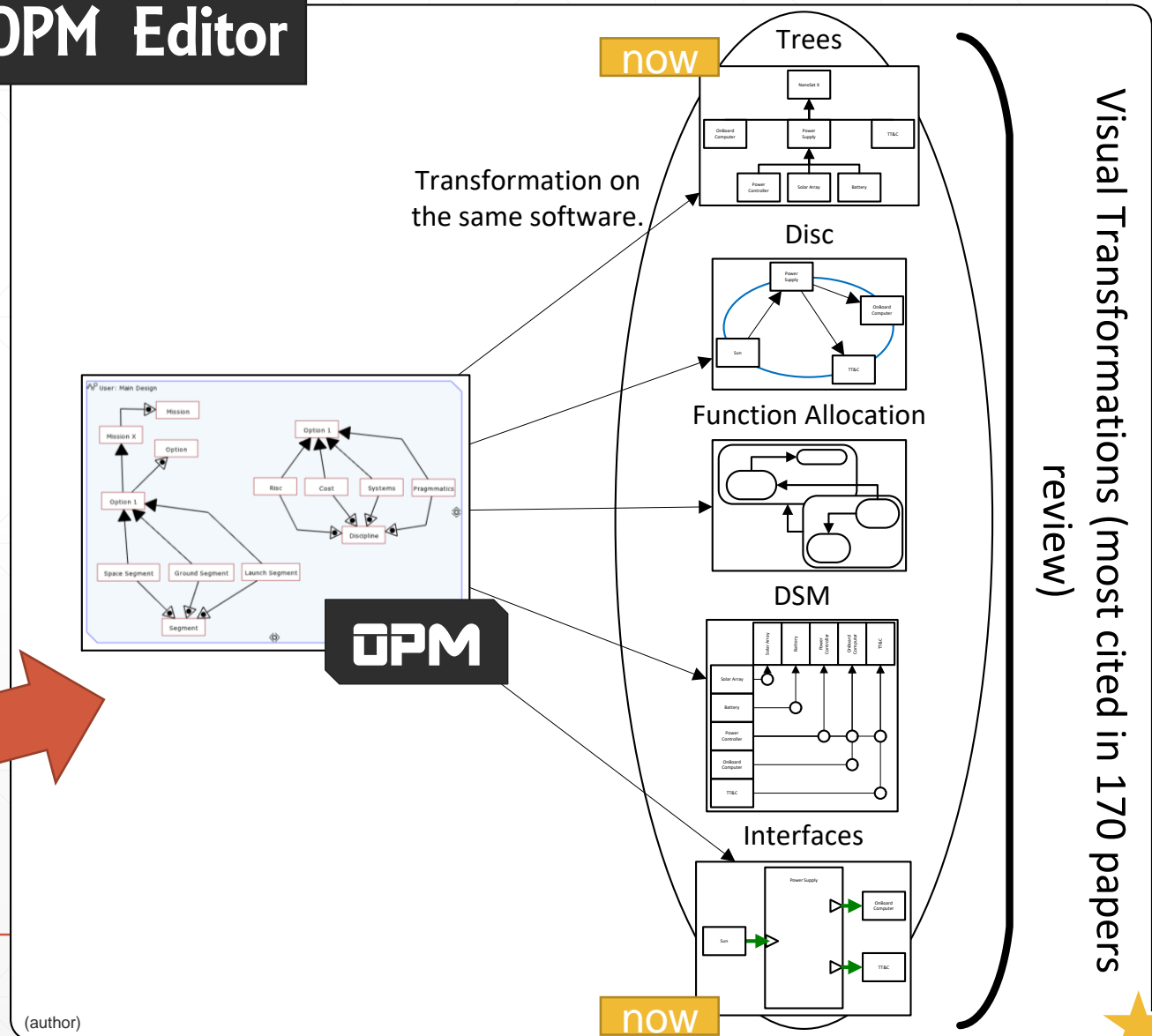
Year	# works	# cite UI	# exclusive about UI
2010	48	28	3
2012	56	24	4
2014	66	29	5
Total	170	81	12

the most cited visual representations:

- Trees,
- Tables,
- Discs,
- Block,
- FSM,
- DSM,
- Interfaces,
- 3D,
- Etc..



OPM Editor



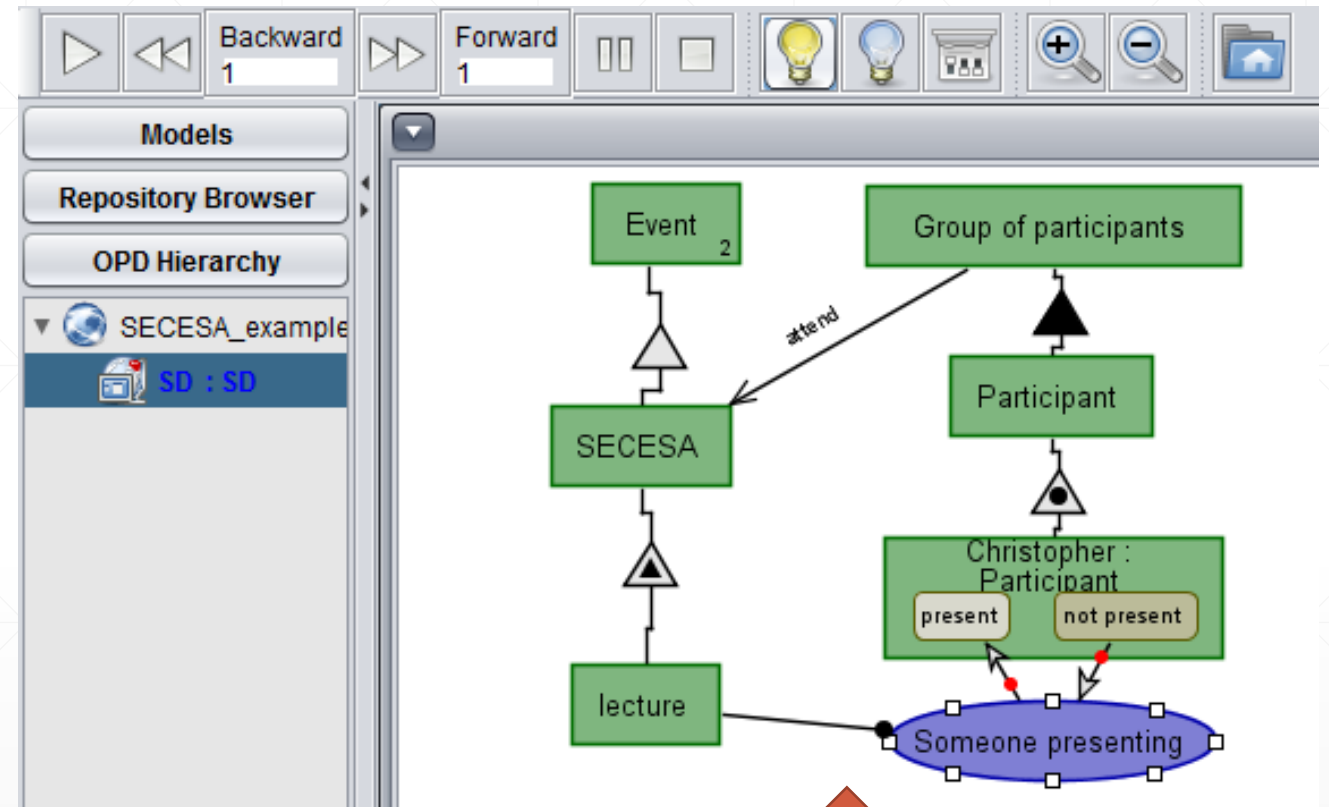
Visual Transformations (most cited in 170 papers review)





Simulation

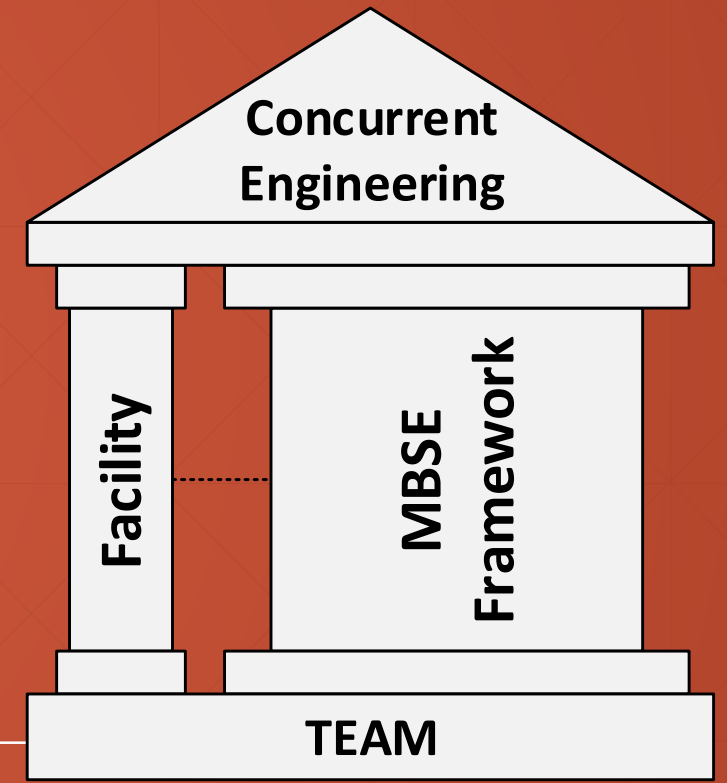
- OPM is simulation ready
- Event-based simulation
- Main simulation activities are:
 - (i) enable processes,
 - (ii) transform objects, and
 - (iii) change object states.



- Figure with the **OPCat** Simulation Controls



Final Considerations



(author)

“Model Based Concurrent Engineering”



Final Considerations

 OPM just turned to be a ISO standard. (15-Dec-2015)

 Being looked by aerospace companies (**EMBRAER/BOEING**)

 Being researched to Concurrent Engineering (**Cambridge/INPE**)

- Knowledge Based Systems

 Simpler to implement than any UML/SysML specification.

- Has direct compatibility with SysML models (**back/forth**)
- Being considered as a *de facto* substitute for SysML (**if it does not get reformulated**)

 ▪ Dual Channel → textual + visual

- Extra third Channel → simulation (**Active Processing**)

 ▪ More understandable than any UML-like visual representation to non-computer specialists.

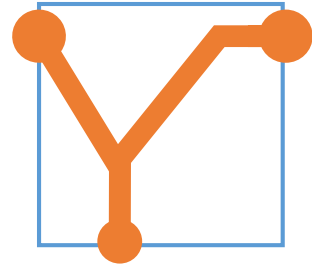
 It will be applied into the CubeSat Mission Definition of the INPE's MSc/PhD program



Questions to think:



- How **easy** (or how **hard**) is to shift from doc-centric to (**OPM**) model-centric?
- Can **OPM** be *the most relevant* visual concept representation in CE?
- As it turned ISO, will **OPM** be an usual (space) engineering “*practice*”?
- Is **OPM** ready to **fulfill all CE needs**?



**Thank you!
Questions?**

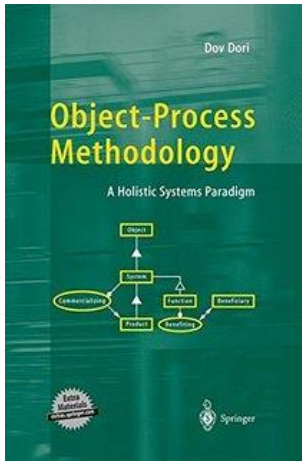


**christophercerqueira@gmail.com
<http://cscerqueira.com.br>**

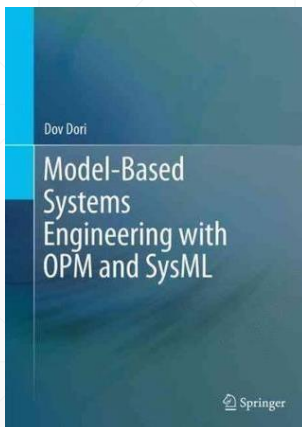
- **Prof. Ph.D. Ana Ambrosio - ana.ambrosio@inpe.br**
- **Prof. Ph.D. Claudio Kirner - ckirner@gmail.com**



For further reading



- Object-Process Methodology – A Holistic System Paradigm, Dov Dori



- Model-Based Systems Engineering with OPM and SysML

ISO/PAS 19450:2015

Automation systems and integration -- Object-Process Methodology

(Only available in English)

Abstract

[Preview ISO/PAS 19450:2015](#)

ISO/PAS 19450:2015 specifies Object-Process Methodology (OPM) with detail sufficient for enabling practitioners to utilise the concepts, semantics, and syntax of Object-Process Methodology as a modelling paradigm and language for producing conceptual models at various extents of detail, and for enabling tool vendors to provide application modelling products to aid those practitioners.

While ISO/PAS 19450:2015 presents some examples for the use of Object-Process Methodology to improve clarity, it does not attempt to provide a complete reference for all the possible applications of Object-Process Methodology.

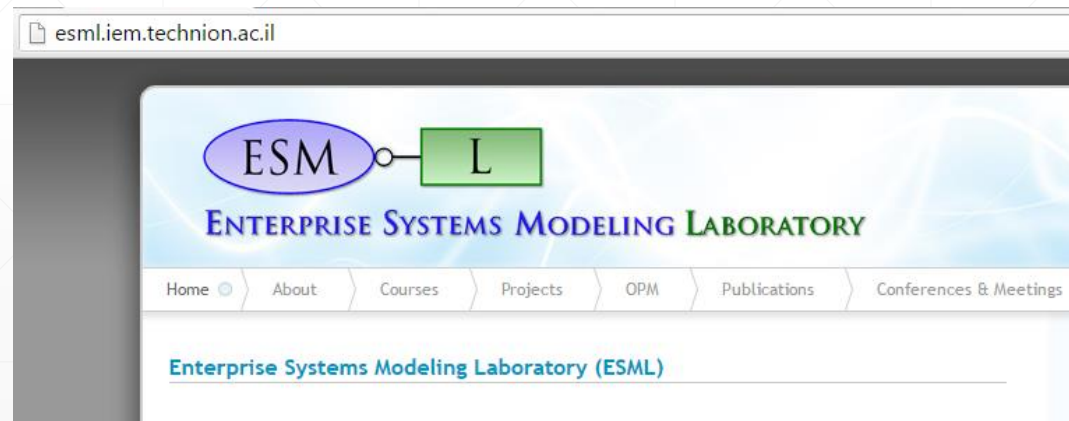
ISO/PAS 1950

FORMAT ? LANGUAGE

PDF English

PAPER English

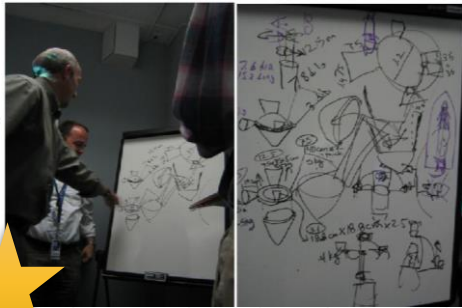
CHF 198 [Add to basket](#)



All the free content at: <http://esml.iem.technion.ac.il/>



CE and Models – (model classification)

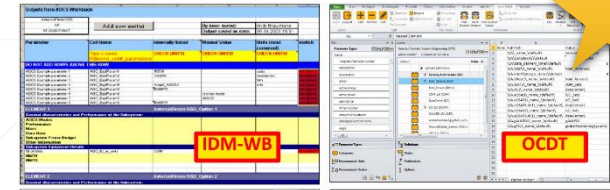


[TeamX]

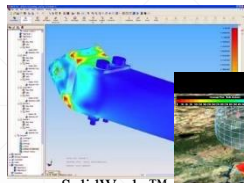
(i) **free** - free descriptions of the systems with no formalism in it,



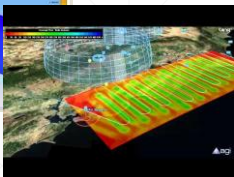
(ii) **loose** - free descriptions of the systems with a fair formalism in software interfaces,



[7]



SolidWorks™



STK™

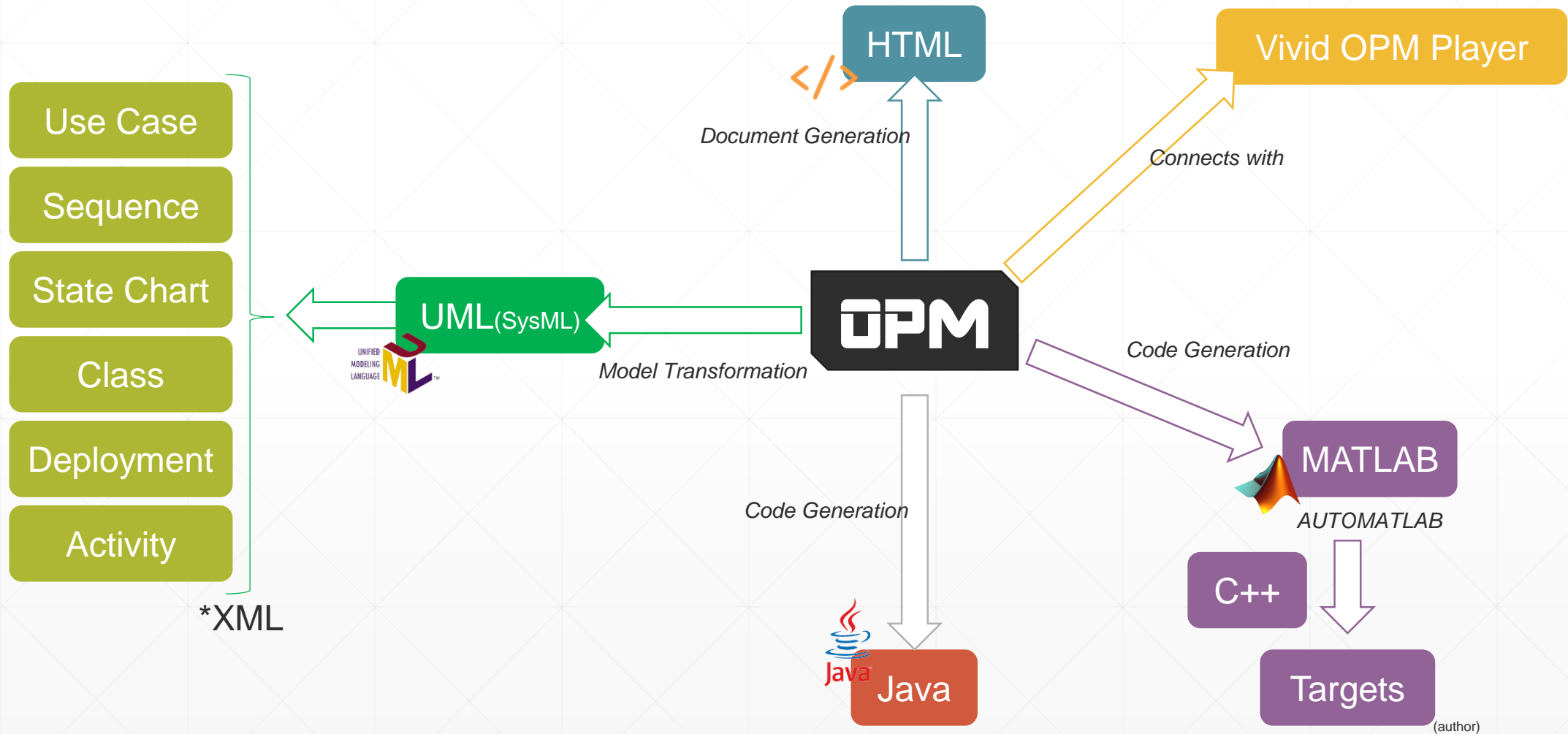
(iii) **domain specific** - described using a certain domain specific language, using the symbols and grammar available by the domains software used by the specialist; and

(iv) **tool independent** - described using a certain domain specific language that is a specialization of a higher-level meta language, and it is easily accessible by other tools.





OPM Ecosystem (that I found)





In practice

JS

Specialist Environment

GUI - Graph Views

Alternative Views

Tree View

DSM View

Interface View

Function View

Disc View

OPM View

GUI - Controls and Comm

Get

Save

Remove

Browse

GUI - Local Saving and Transformation

Save

Load

XMI

exported model

local file

{JSON}

Server