# Towards a Formal Specification of Multi-Paradigm Modelling

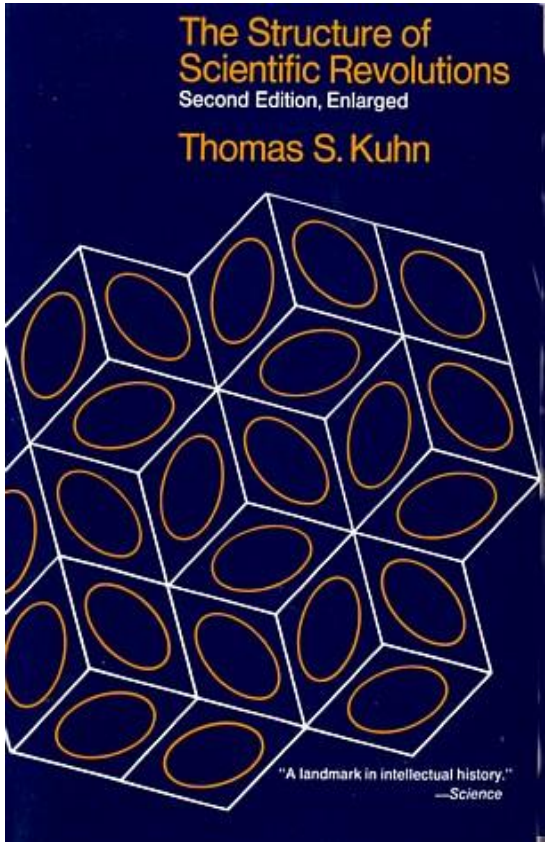
Moussa AMRANI, Dominique BLOUIN, Robert HEINRICH, Arend RENSINK, Hans VANGHELUWE, Andreas WORTMANN

MPM4CPS Workshop. 15 September 2019
Co-located with the MoDELS Conference, Munich, Germany

# Simulation in Europe

**ESPRIT Basic Research Working Group 8467**
**Simulation for the Future: New Concepts, Tools and Applications**

Keywords:
simulation technologies, multi-paradigm modelling, solvers, standards, interoperability, industrial deployment, demonstrators, user-simulator interfaces




# Simulation for the Future: Progress of the Esprit Basic Research Working Group 8467

Hans L. Vangheluwe, Ghislain C. Vansteenkiste
Applied Mathematics, Biometrics and Process Control
University of Gent
Coupure 653
B-9000 Gent
Belgium

Eugène J.H. Kerckhoffs *
Applied Mathematics and Informatics
Delft University of Technology
Julianalaan 132
NL-2628 BL Delft
The Netherlands

tel: +32 (9) 264 59 37
fax: +32 (9) 223 49 41
e-mail: SiE@hobbes.rug.ac.be

## Abstract

The Esprit Basic Research Working Group 8467 on "Simulation for the Future: new concepts, tools and applications", with the acronym SiE-WG (Simulation in Europe - Working Group), started its work on December 1, 1993. It was an initiative of the SiE-SIG (Special Interest Group), currently consisting of some 150 industrial and academic members. The SiE-SIG acts as a platform and validating forum for SiE-WG results. The –now concluded– first phase of the SiE-WG activities was one of meta-research and focusing. From the vast arena of modelling and simulation, specific topics were selected which are deemed crucial for the future. Research into these topics now receives sponsoring of the SiE-WG. Action clusters of particular relevance to European industry and to the end-user were defined. Within these clusters concerted research now takes place (with SiE sponsoring meetings).

The history of SiE is presented as well as the methods used to come to concrete Basic Research actions. The conclusions of the focussing phase of SiE's activities are presented in detail.

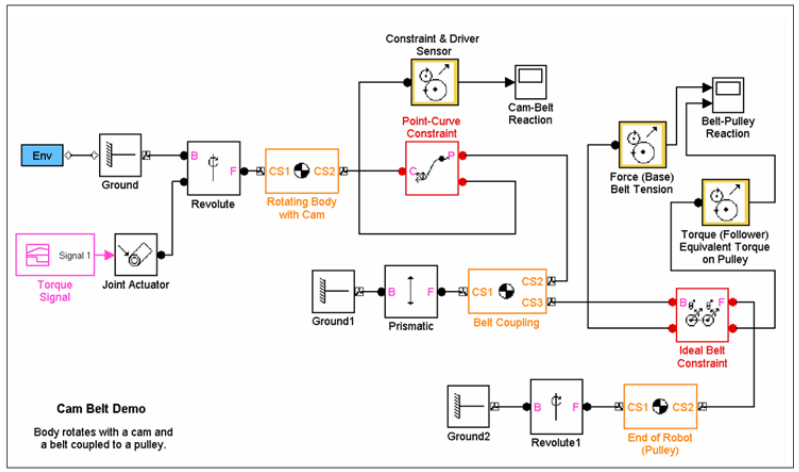
## Special Interest Group SiE-SIG

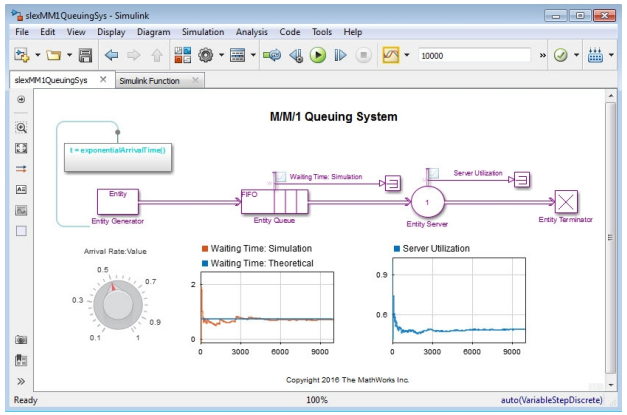
In January 1992, the Special Interest Group "Simulation in Europe" (SiE-SIG) was established. This was a direct result of the increasing need for computer simulation, its multi-disciplinary aspects and a joint industry/academia interest. One of the aims of SiE-SIG's establishment was to draw the attention of the Esprit officials to introducing simulation in Esprit Workprogrammes and encouraging its members to submit Esprit project proposals. On the basis of SIG discussions "Simulation Policy Guidelines" were formulated:

1. Improve the modelling and simulation process
   (a) Modelling:
   Redefine "modelling" in a broader perspective than currently used and exploit this as a basis for new modelling and simulation methodologies (i.e., *multi-paradigm systems*).
   Focus on generic (e.g., object-oriented) component modelling and supporting representations to enhance re-usability and portability of existing and new simulation models.
   (b) Techniques:
   Adapt Software Engineering and Artificial Intelligence methods and tools (e.g., formal verification, re-use, version management and decision support) to modelling and simulation problems. Merge results in integrated methods and tools (e.g., *multi-language software systems*).
   (c) Life-cycle:
   Attention to the full Modelling/Simulation Experimentation/Validation life-cycle. Explicit description and prescription of this (possibly concurrent) life-cycle to improve quality of the end-products (software and/or hardware).
2. Open new application areas
   (a) Include new peripheral devices and novel algorithms into simulators. Enter new application areas (for example, in the medical sector).
   (b) Exploit *highly parallel* hardware architectures to simulate *multi-component* systems by directly mapping model structure onto hardware structure.
3. Provide user-simulator interfaces
   (a) Provide a common basis for independent development of simulators and user-interfaces by means of Open Systems (e.g., PCTE).

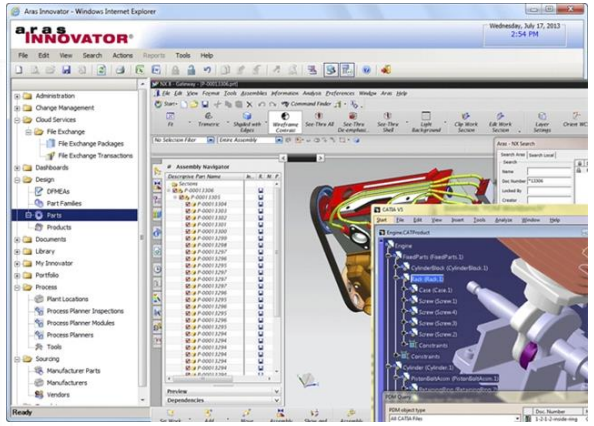
*Currently part-time professor at the University of Gent

# Multi-Formalism

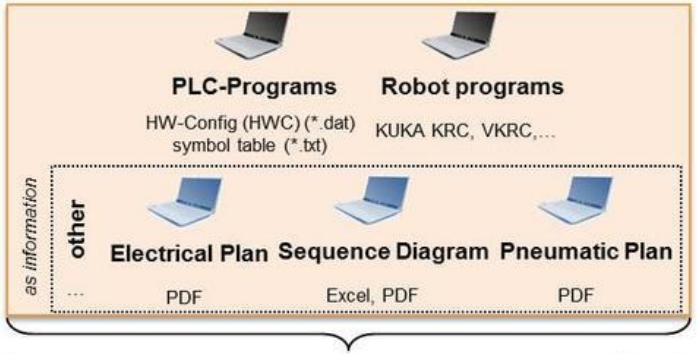


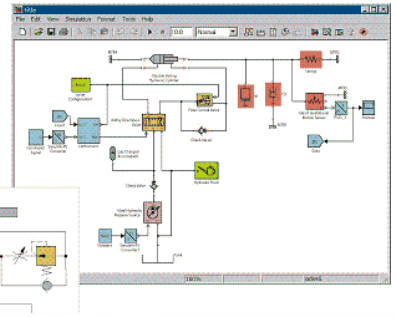



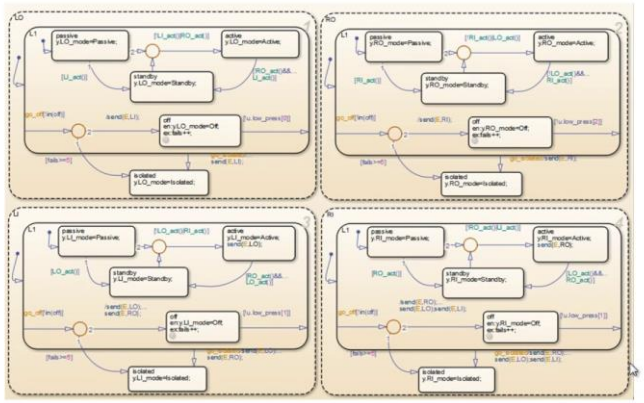

# PLM

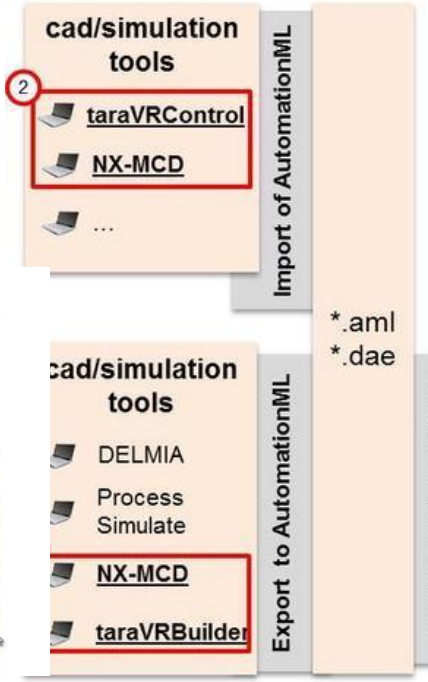



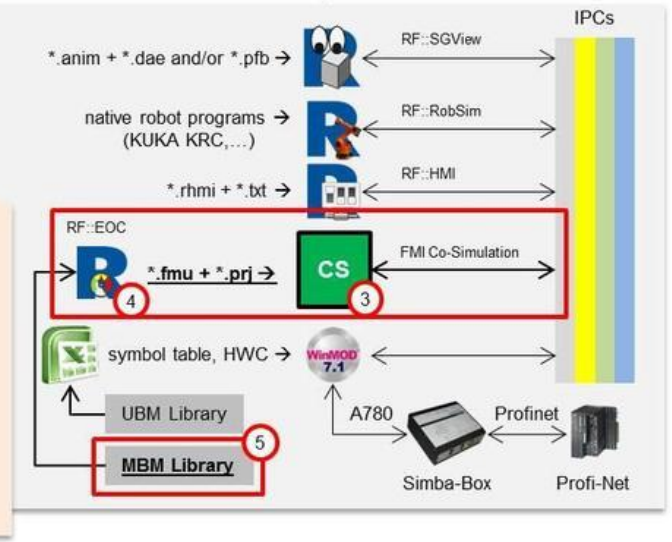

# toolchains







# 2019 Bellairs CAMPaM workshop


Home

Participants

Bellairs

Schedule

Manage


Welcome to the home page of the **16th Bellairs CAMPaM workshop**.

The workshop aims to further the state-of-the-art in **Computer Automated Multi-Paradigm Modelling** (CAMPaM) - a name (though Hans preferred "Aided" to "Automated") coined by Pieter Mosterman and Hans Vangheluwe in the late '90s - as well as to define future directions of this emerging research area by bringing together world experts in the field for an intense one-week workshop.

The workshop will be held **Friday 26 April** (arrival) - **Friday 3 May** (departure) **2019** at McGill University's Bellairs campus. The actual workshop starts on Saturday morning and continues for 5 full days (until Wednesday evening). Although it is possible to depart on Thursday, most participants leave on Friday to do some sightseeing on Thursday (in particular, to visit Crane Beach).
The workshop takes the Dagstuhl seminar format --bring a critical mass of top researchers together in a relatively remote location and soon new ideas will flow-- one step further: the Bellairs facilities are relatively primitive (and cheap) and there are no distractions such as typically found in hotels.

Organizers:


- **Hans Vangheluwe**, Department of Mathematics and Computer Science, University of Antwerp, Belgium, and School of Computer Science, McGill University, Canada.
- **Clark Verbrugge**, School of Computer Science, McGill University, Canada.


## Workshop Subject

### Computer Automated Multi-Paradigm Modelling (CAMPaM)

CAMPaM acknowledges that explicit **modelling** is the central activity in and main enabler for the **analysis and design of complex systems**. Because of the heterogeneous nature of for example embedded systems and the many implementation technologies, Multi-Paradigm Modelling is a critical enabler for holistic design approaches (such as mechatronics), to avoid overdesign and to support system integration. Multi-paradigm techniques have been successfully applied in the field of software architectures, control system design, model integrated computing, and tool interoperability. Fifteen CAMPaM workshops at Bellairs '04, '05, '06, '07, '08, '09, '10, '11, '12, '13, '14, '15, '16, '17, '18, many Multi-Paradigm Modelling (MPM) conference sessions and MoDELS '06 (Genoa), '07 (Nashville), '09 (Denver), '10 (Oslo), '11 (Wellington), '12 (Innsbruck), '13 (Miami), '14 (Valencia), '15 (Ottawa) workshops have been held. A special issue of the ACM Transactions on Modeling and Computer Simulation (TOMACS) was devoted to CAMPaM, and COST Action IC1404 "Multi-Paradigm Modelling for Cyber-Physical Systems" (MPM4CPS) worked 2014 - 2019 on MPM solutions for the design of complex, Cyber-Physical Systems. See also the (very outdated) CAMPaM page for more related material.

# MPM4CPS

cost
EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY



## Welcome to COST Action IC1404

## **Multi-Paradigm Modelling for Cyber-Physical Systems** (MPM4CPS)

Pisa Training School 2018

COST is supported by the EU Framework Programme Horizon 2020

### ANNOUNCEMENTS

Do you feel you can contribute to MPM4CPS? Become a collaborator! Fill the form.

The reports of the completed 2015 - 2018 STSMs can now be consulted here.

# Science-y Definition of Paradigm

From Merriam-Webster [1], and my own (paper) dictionary:

1. An **example**, a **pattern**; an outstandingly clear or **typical example** or **archetype** [1]
2. A **framework** containing the **basic assumptions**, **ways of thinking**, and **methodology** that are commonly accepted by members of a **scientific community**, and such a cognitive framework shared by members of any discipline or group [1]
   A philosophical and **theoretical framework** of a scientific school or discipline within which **theories**, **laws**, and **generalizations** and the experiments performed in support of them are formulated

(Some) *universally recognized scientific* achievements that, for a time, provide model problems and solutions for a community of practitioners, i.e., [2]

- *what* is to be observed and scrutinized
- the kind of *questions* that are supposed to be asked and probed for answers in relation to this subject
- *how* these questions are to be structured
- *what* predictions made by the primary theory within the discipline
- *how* the results of scientific investigations should be interpreted
- *how* an experiment is to be conducted, and *what* equipment is available to conduct the experiment

[1] https://www.merriam-webster.com/dictionary/paradigm
[2] Kuhn, Thomas S. (1996). The Structure of Scientific Revolutions.

# Object-Orientation on GPLs

| [OO] Object Orientation [extracted from 1] | |
|---|---|
| [OO1] | Shall possess the concepts of **Class** and **Object** |
| [OO2] | Objects shall possess a **state** and a **set of capabilities / operations** |
| [OO3] | Shall possess an **inheritance** mechanism |
| [OO4] | Inheritance shall allow **operation reuse** |



[1] P. Wegner, “Dimensions of Object-Based Language Design,” SIGPLAN Notices, vol. 22, no. 12, pp. 168–182, Dec. 1987.

# Object-Orientation on GPLs

| [OO_WEG] Object Orientation [extracted from 1] | |
|---|---|
| **[OO1]** | Shall possess the concepts of **Class** and **Object** |
| **[OO2]** | Objects shall possess a **state** and a **set of capabilities / operations** |
| **[OO3]** | Shall possess an **inheritance** mechanism |
| **[OO4]** | Inheritance shall allow **operation reuse** |


| [OO_Me] Object Orientation with Multiple Inheritance and … | |
|---|---|
| [OO1] | Shall possess the concepts of **Class** and **Object** |
| [OO2] | Objects shall possess a **state** and a **set of capabilities / operations** |
| [OO3'] | Shall possess a **multiple inheritance** mechanism |
| [OO4] | Inheritance shall allow **operation reuse** |
| **[OO5]** | Shall use **polymorphism** with subtyping |
| **[OO6]** | **…** |


[1] P. Wegner, "Dimensions of Object-Based Language Design," SIGPLAN Notices, vol. 22, no. 12, pp. 168–182, Dec. 1987.

# ”Agile” Development

1. Our highest priority is to satisfy the customer through **early and continuous delivery** of valuable software.
2. Welcome **changing requirements**, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver **working software frequently**, from a **couple of weeks to a couple of months**, with a preference to the **shorter timescale**.

…

7. **Working software** is the primary measure of progress.

| [Agile_Me] Agile Development | |
|---|---|
| **[Agile1]** | Shall follow a development lifecycle |
| **[Agile2]** | The full lifecycle, called macro, contains itself smaller, micro-lifecycles |
| **[Agile3]** | Each micro-cycle produces a working software |
| **[Agile4]** | Each micro-cycle last between 2 weeks and 4 months |
| **[Agile5]** | **…** |

[1] Kent Beck et al. (2001). "Principles behind the Agile Manifesto". Agile Alliance. https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

# Our notion of Paradigm

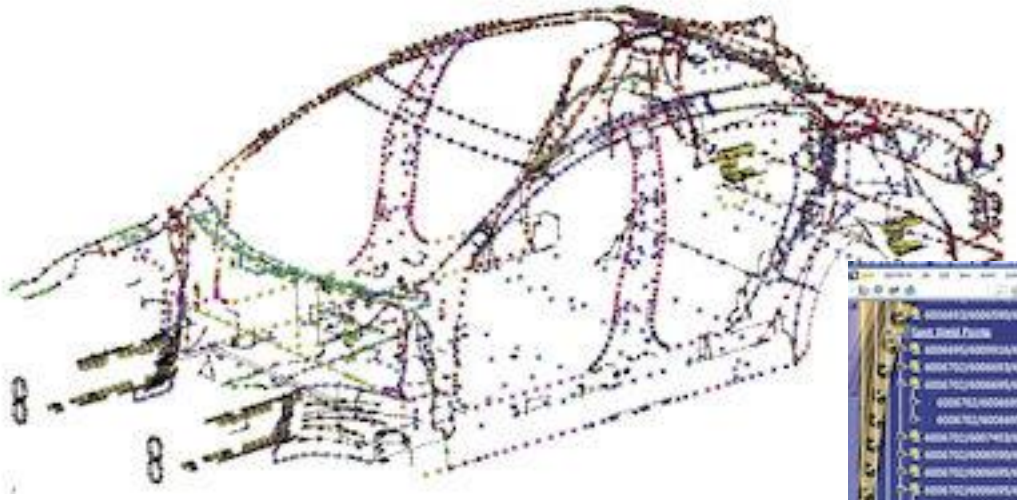
- A **paradigmatic** structu $PS \in \mathbb{PS}$ that includes
  - All **languages** manipulated;
  - All **processes** describing usage scenarios
  - All **relationships** between both

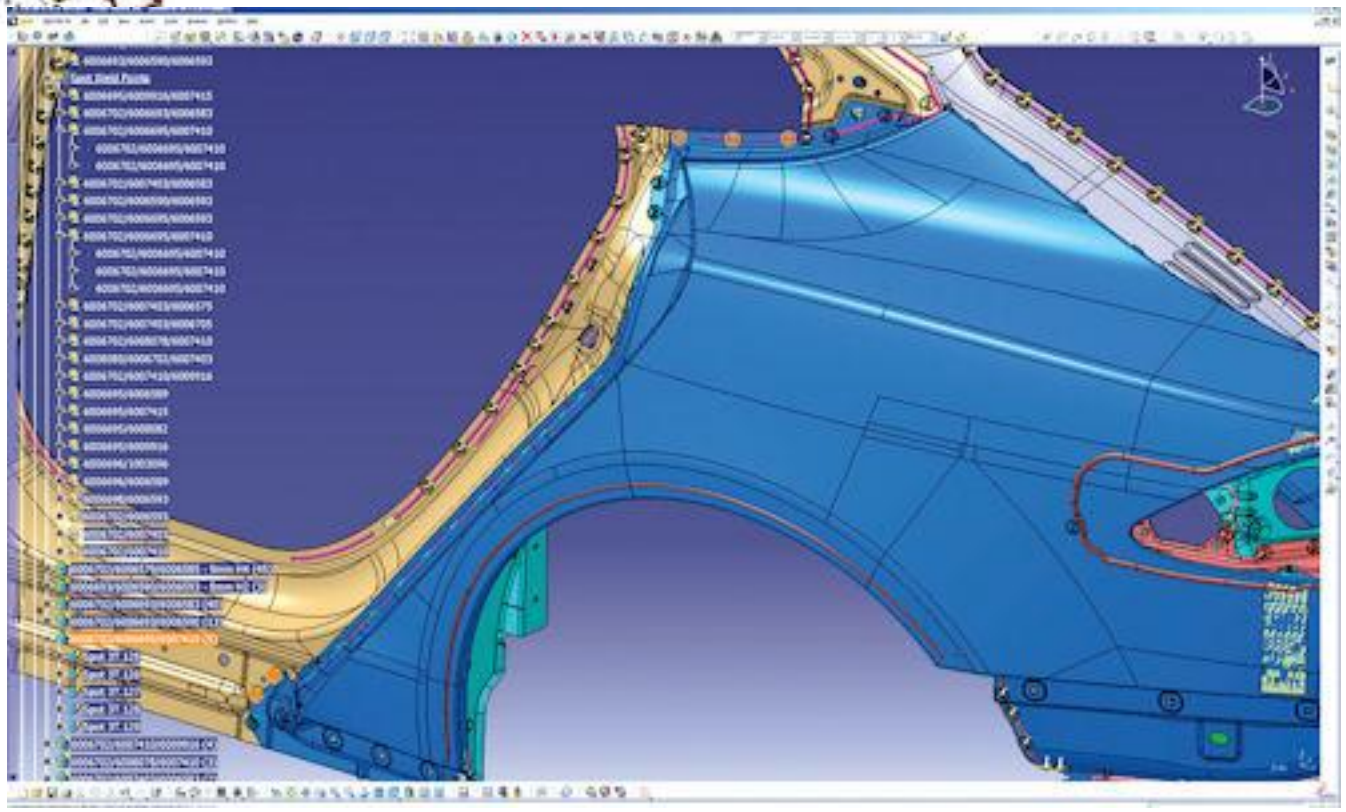
- A function $\boldsymbol{\iota} \in [\mathbf{ParadigmName} \mapsto \boldsymbol{\Pi}]$ mapping paradigm names to **(paradigmatic) properties** $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ characterising the paradigms at hand

- A **checking procedure** $\pi \vdash \mathrm{PS}$
  - May be automatic (for some parts)
  - But may well include human checking

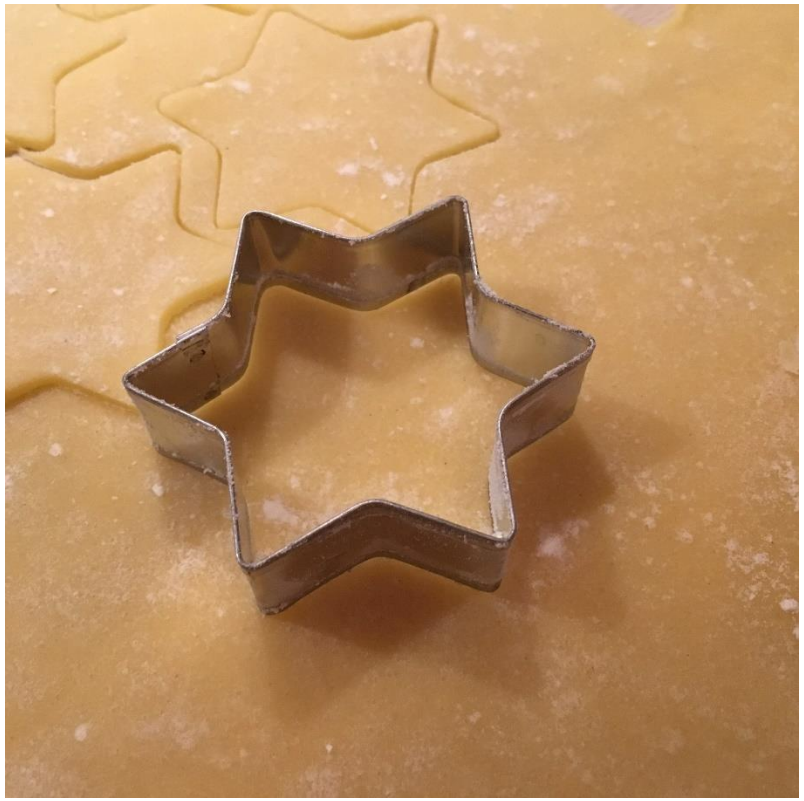
- PS **embodies** (or follows, or qualifies as) paradigm $p$ iff $\boldsymbol{\iota}(\boldsymbol{p}) \vdash \mathbf{PS}$

# Computer-Aided Design

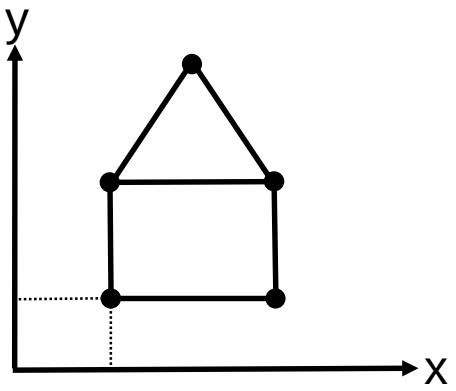




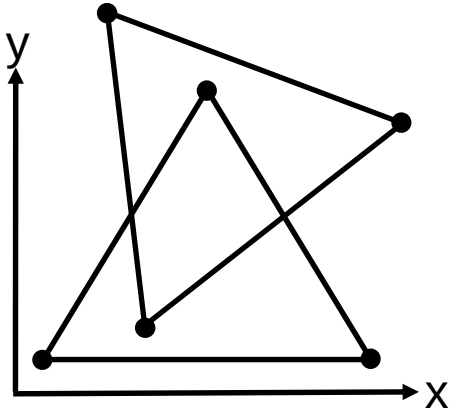
# Cookie Computer-Aided Design & Manufacturing



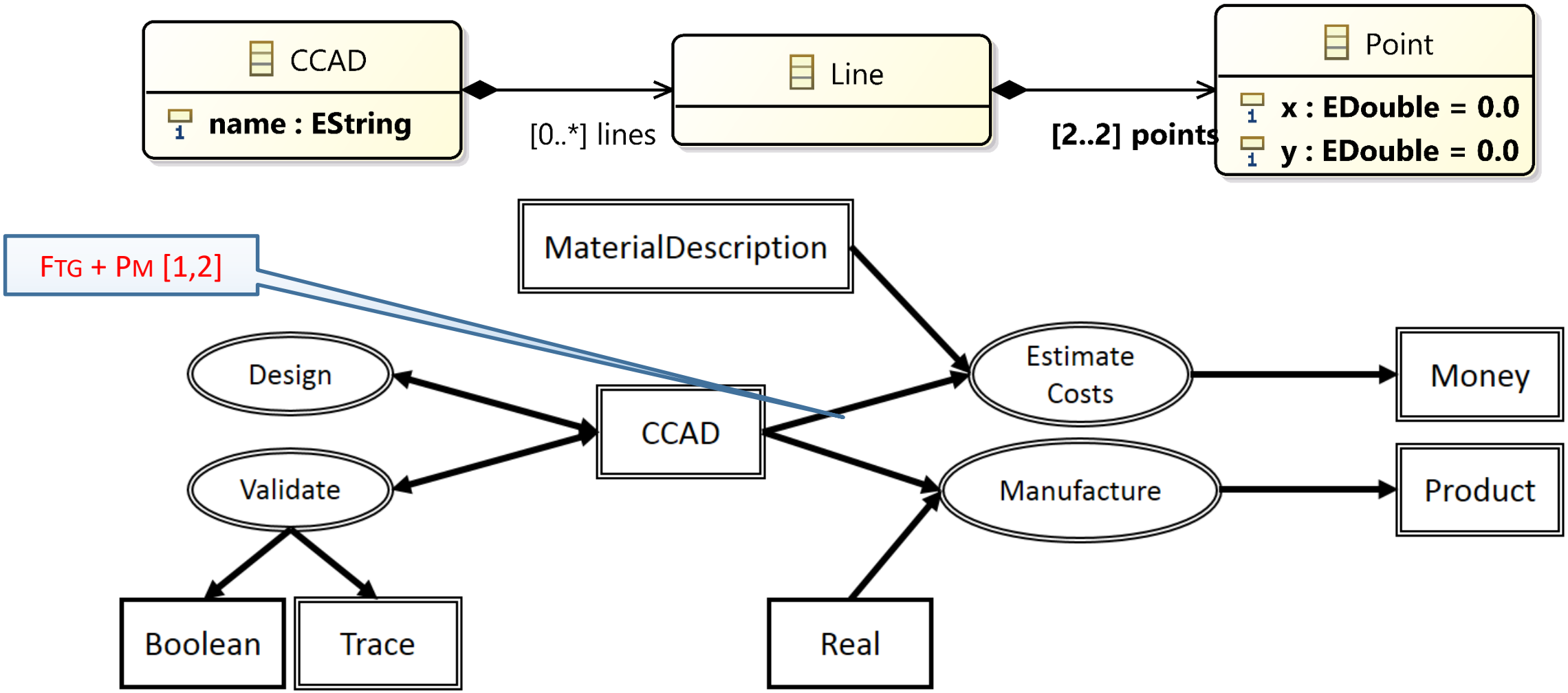
Valid Cookie Cutter
- shared lines
- welded parts?




Invalid Cookie Cutter
- crossing lines
- “floating” points



# CookieCAD MM + Processes



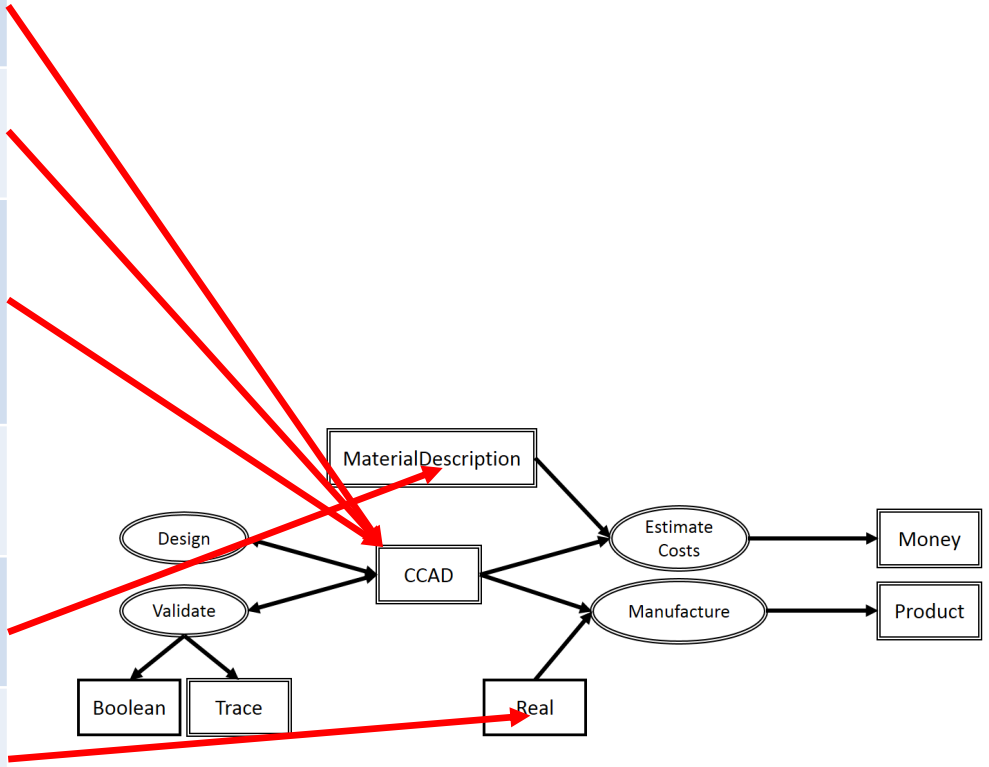


[1] L. Lúcio, S. Mustafiz, J. Denil, B. Meyers, and H. Vangheluwe. (2012). "The Formalism Transformation Graph As A Guide To Model-Driven Engineering,", Tech. Rep. McGill University.

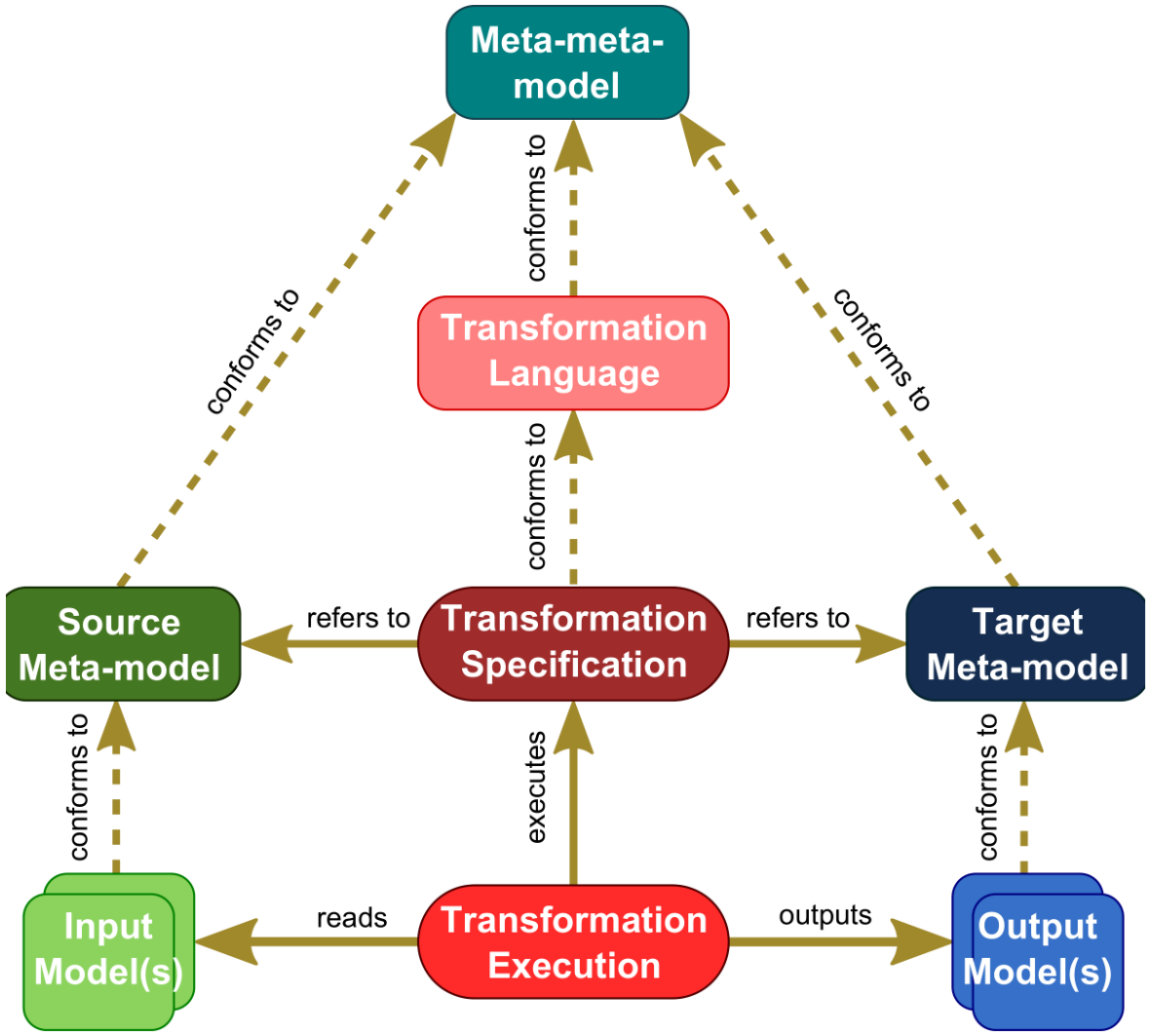
[2] L. Lúcio, S. Mustafiz, J. Denil, H. Vangheluwe, and M. Jukss, (2013). FTG+PM: An Integrated Framework For Investigating Model Transformation Chains. In International SDL Forum, pp. 182–202.

# CookieCAD Properties

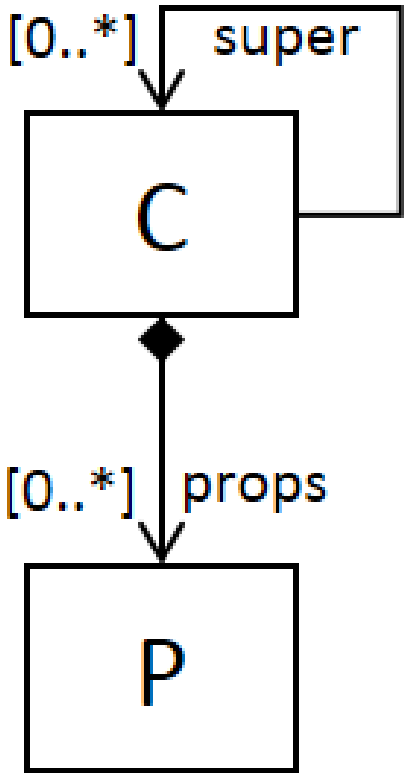
| [CookieCAD] CookieCAD Design & Manufacturing | |
|---|---|
| **[CookieCAD1]** | Shall define points and lines in 2D |
| **[CookieCAD2]** | A Shape is defined by a valid set of lines |
| **[CookieCAD3]** | A valid shape ensures the following properties<br>• It does not contain crossing edges<br>• … |
| **[CookieCADN]** | The material shall have an acceptable strain factor |
| **[CookieCADN+1]** | A strain factor shall be comprised between …. |
| **[CookieCADN+m]** | An acceptable thickness shall be defined for manufacturing |
| **…** | … |



# Enabler #1: Transformations



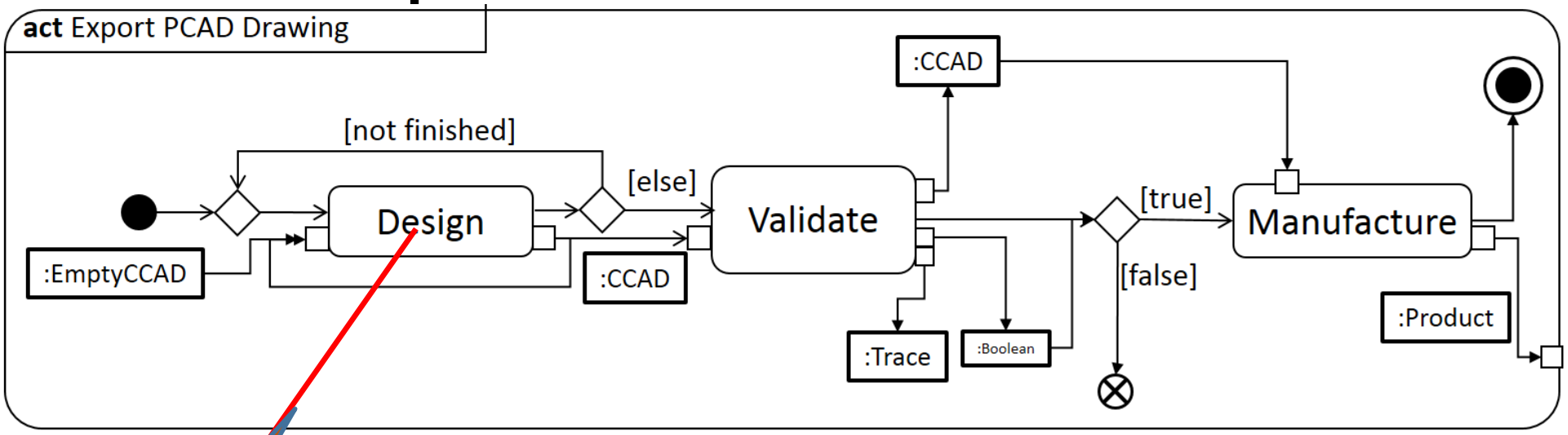
# Enabler #2: “Template” MM

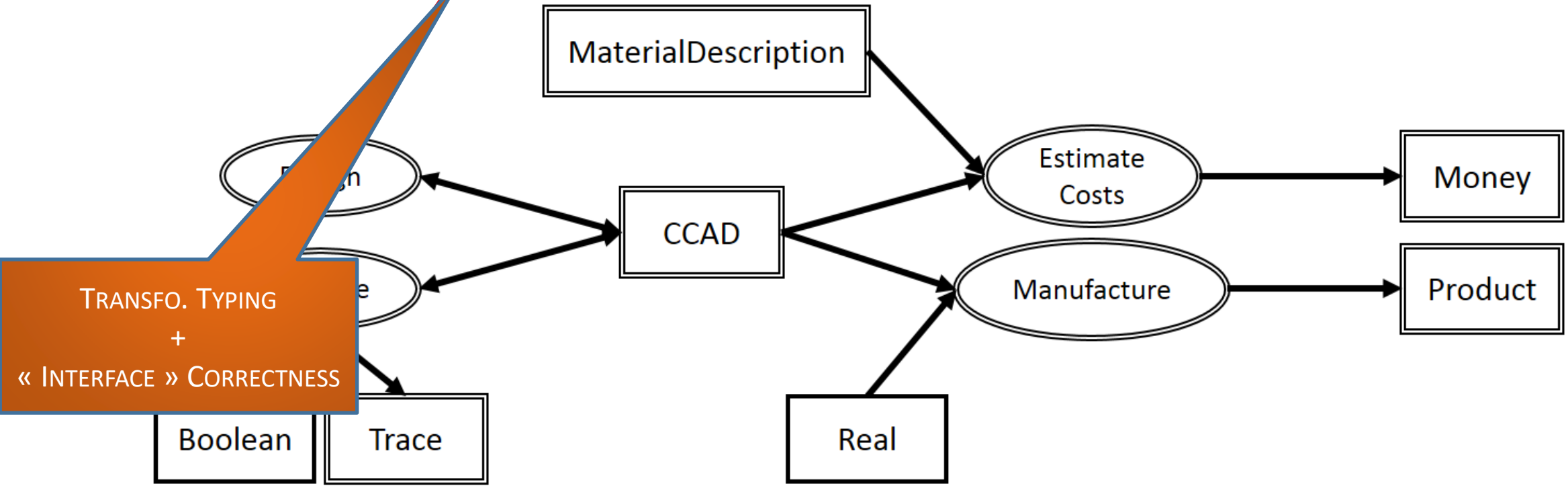



In Java, the **structural part** of the inheritance prop (OO1 – OO3) would correspond to the following:

- `C` would match to `Class` or `Interface`
- `P` would match to `Member`
- `super[0..*]` would match to `extends` or `implements`, and multiplicity would be restricted to `1`

# Enabler #3: “Template” Transformations




TRANSFO. TYPING
+
« INTERFACE » CORRECTNESS

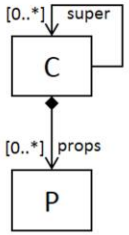
# Conclusion

## CS Paradigm

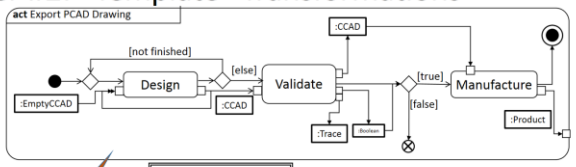
- A **paradigmatic** structure $PS \in \mathbb{PS}$ that includes
  - All **languages** manipulated;
  - All **processes** describing usage scenarios
  - All **relationships** between both
- A **set of (paradigmatic) properties** $\pi \in \Pi$
  capturing the characteristic properties describing the paradigm
- A **checking procedure** $\pi \vdash PS$
  - May be automatic (for some parts)
  - But may well include human checking

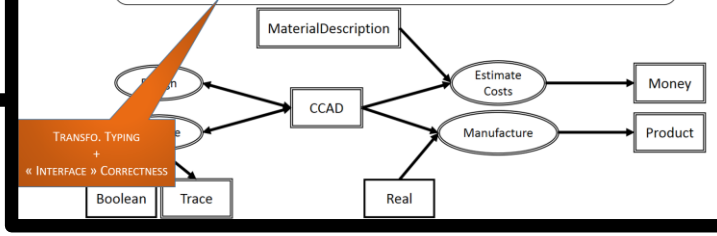
## Enabler #1: Transformations

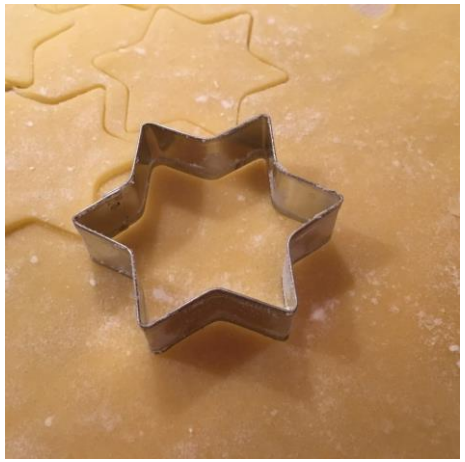

## Enabler #2: "Template" MM




## Enabler #2: "Template" Transformations



# Towards Multi-Paradigm Modelling for CPS

# Towards Multi-Paradigm Modelling for CPS

- Relate Paradigms through **relations**
  (last, left-out component of the paradigmatic structure)
  - Come with their own sets of paradigmatic properties
  - Add “sanity-check” (paradigmatic) properties

- Apply math. framework to real-life case studies
  - Study FTG+PM produced during previous COST Workshops
  - Illustrate with self-contained Use Cases (PowerWindows? HPI Lab Robotics? Other?)