Integrating Finite Element Analysis with Systems Engineering Models

- KONEKSYS – Jerome Szarazi, Axel Reichwein

July 26, 2016

Koneksys

This work was performed under the following financial assistance award NIST Grant 2014-NIST-MSE-01 from U.S. Department of Commerce, National Institute of Standards and Technology (U.S. NIST contact: Conrad Bock)

Outline

- Introduction and motivation
- Challenges in FEA standardization
- New proposed FE mathematics description
- Validation
- Next steps and summary

Outline

- Introduction and motivation
- Challenges in FEA standardization
- New proposed FE mathematics description
- Validation
- Next steps and summary

Koneksys

Integration between Systems Engineering and FEA

PRODUCT

Requirement

Behaviour

Systems Engineering

Structure

Parametric

Software engineering

Electrical engineering

Mechanical engineering

Manufacturing engineering

FEA

Motivation: Communication and archiving

Cross-disciplinary communication

Defining concepts

Are my requirements validated?

What do you want me to simulate?

Archiving and reuse

What has been simulated?

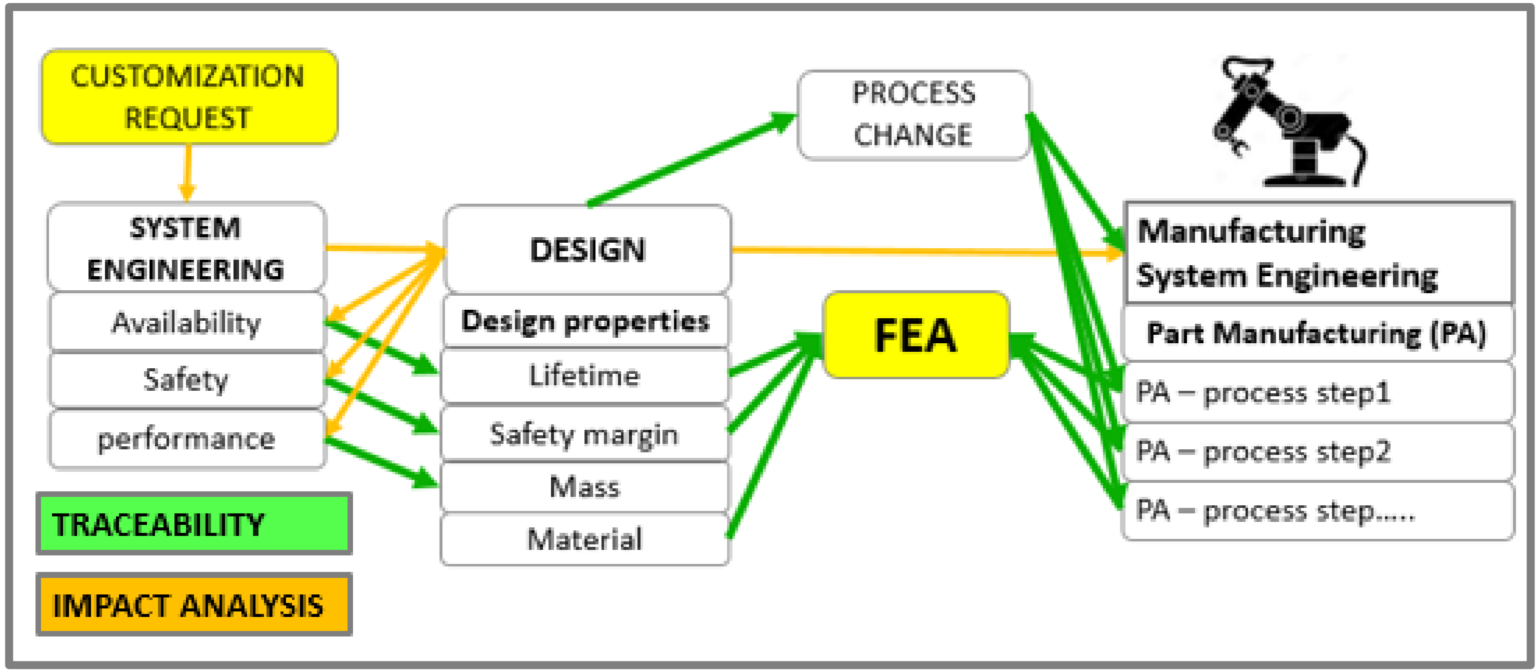
Can I reuse the simulation?

Koneksys

Motivation: traceability and impact analysis

Requirement traceability

Impact analysis

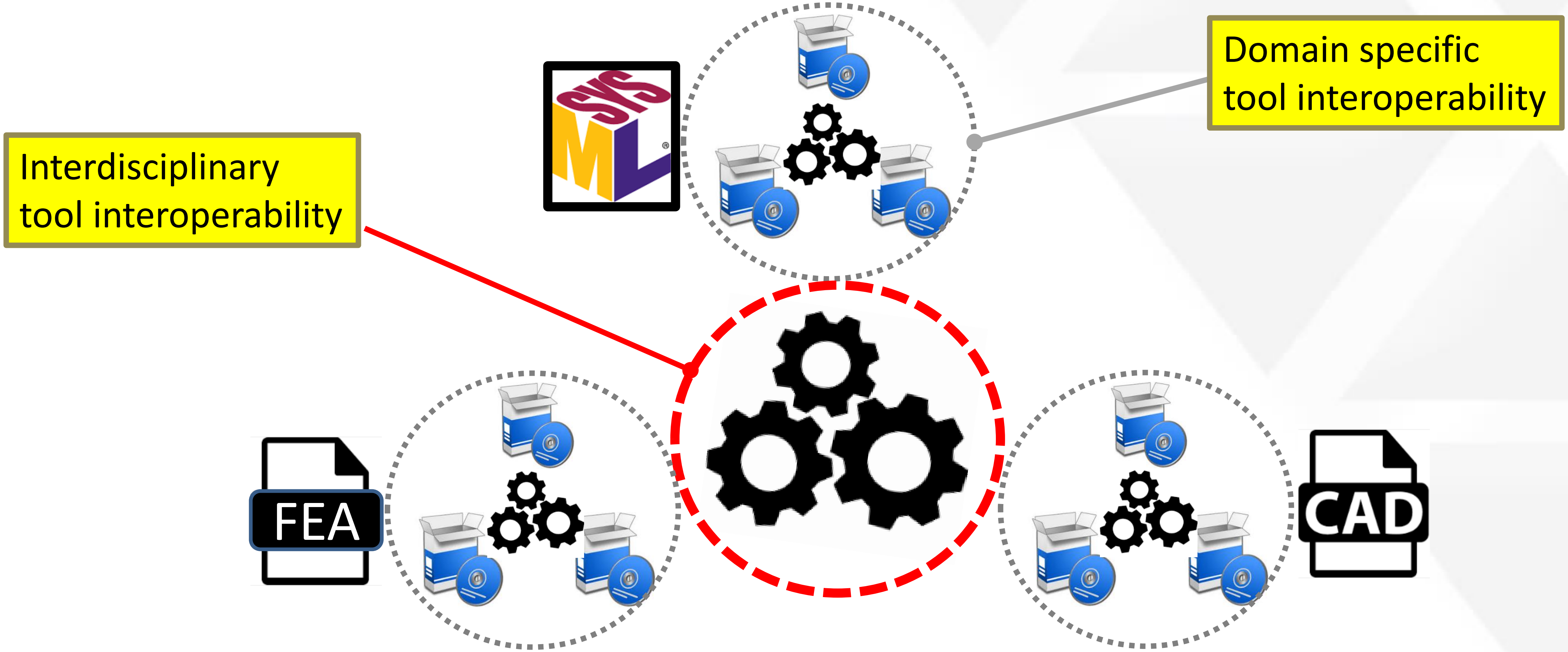


e.g. Customization request

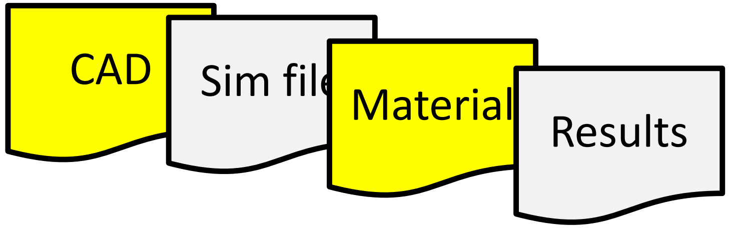
e.g. Cost reduction program…

Koneksys

Motivation: Tool interoperability

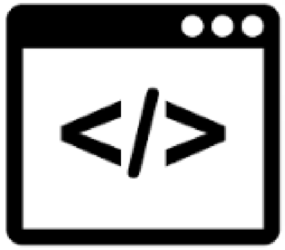


The challenge

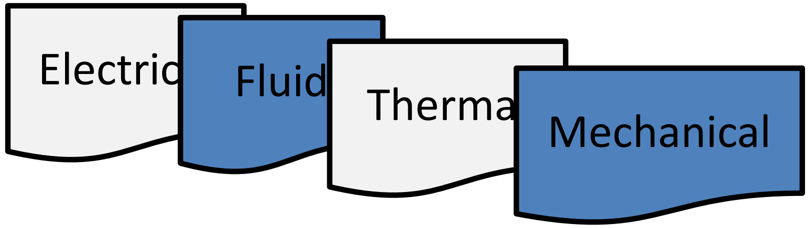


Many Artifacts

Many Vendors



Custom code



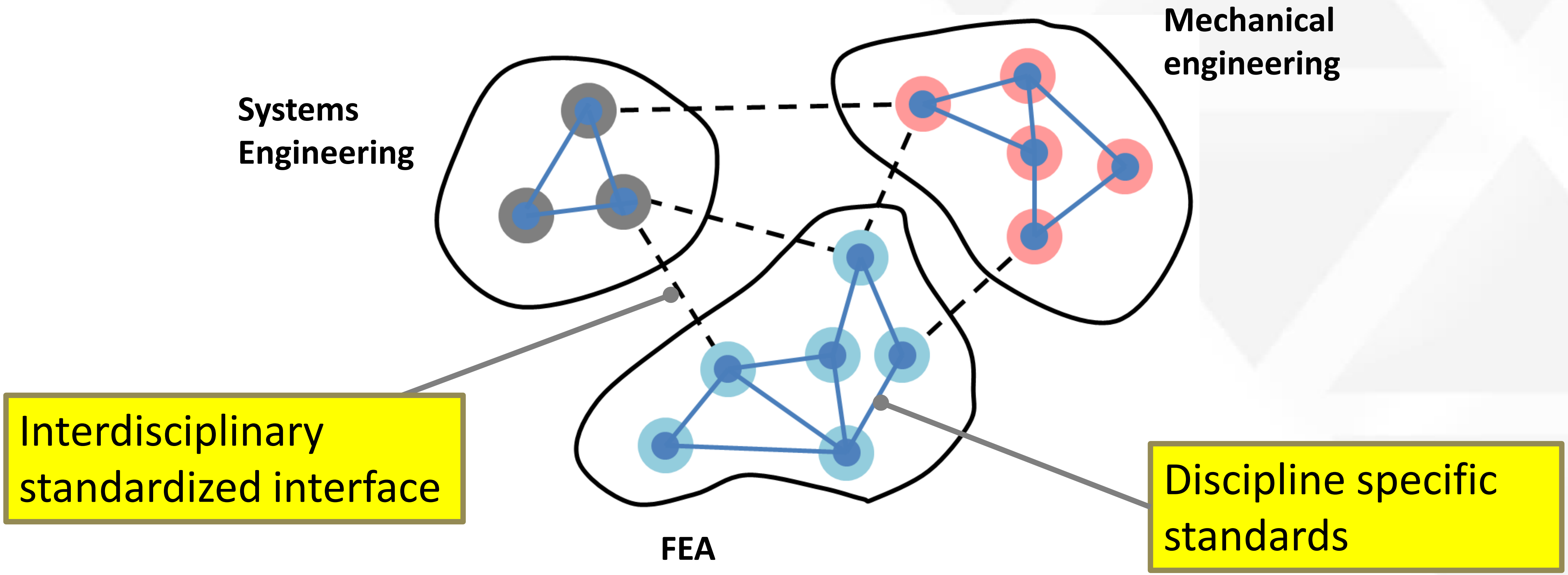
Multiphysics

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

FEA is complex

Koneksys

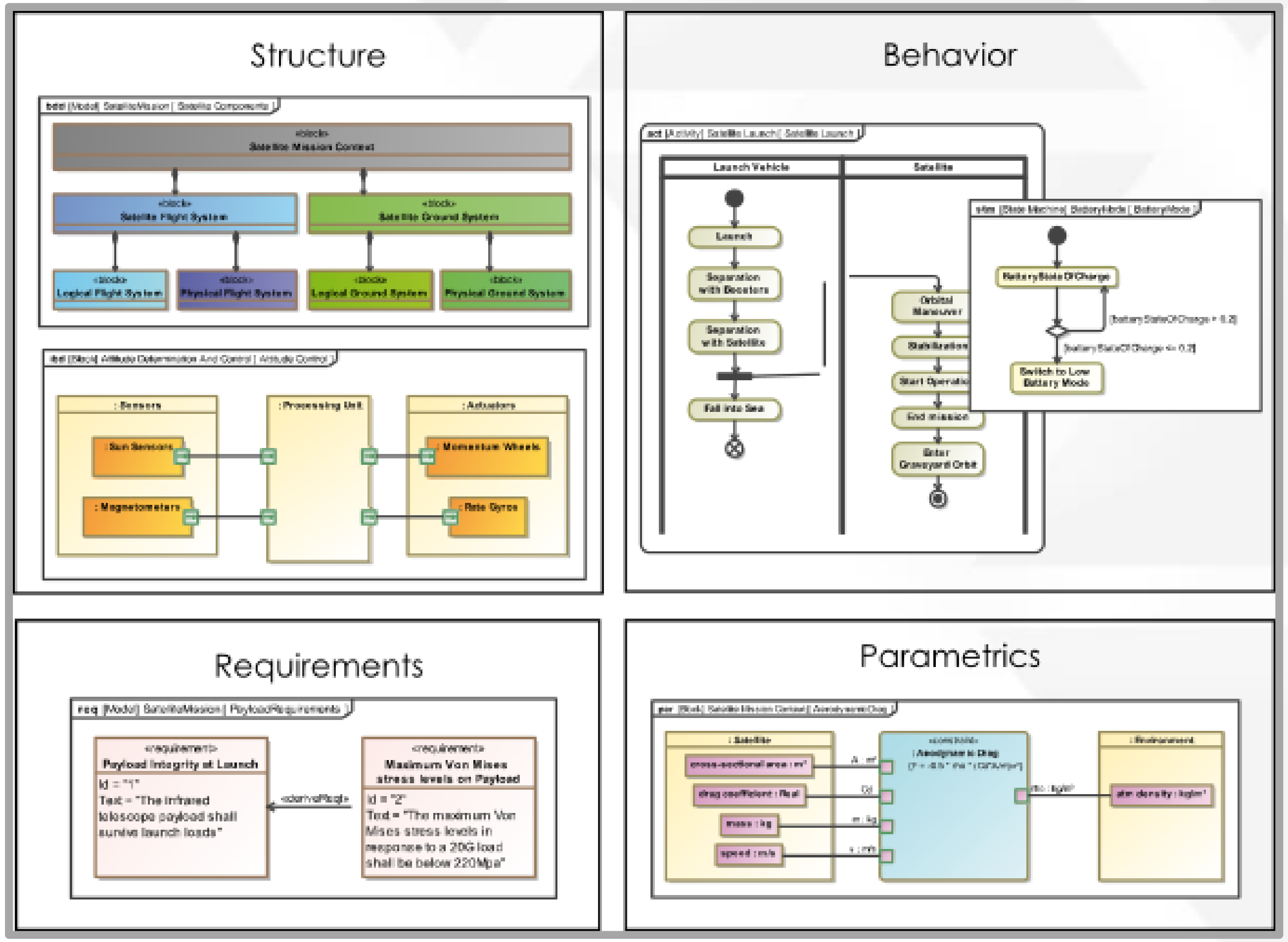
The requirement to success



Koneksys

SysML – Standard for Systems Engineering

- SysML: Systems Modeling Language
- Defined by the OMG as standard in 2007
- Widely adopted for Model-Based Systems Engineering (MBSE)
- Current version: 1.4 (2015)



Koneksys

FEA-related Standards

No useful standards

Standards supporting FEA

- STEP-AP209-ED1
- STEP-AP209-ED2
- IGES V6
- Finite element module
- Finite element shape analysis
- Finite element shape analysis
- JT 9.5 ISO/FDI S 14306

Incomplete
Informal

Many data formats

Common format mesh and visualization format

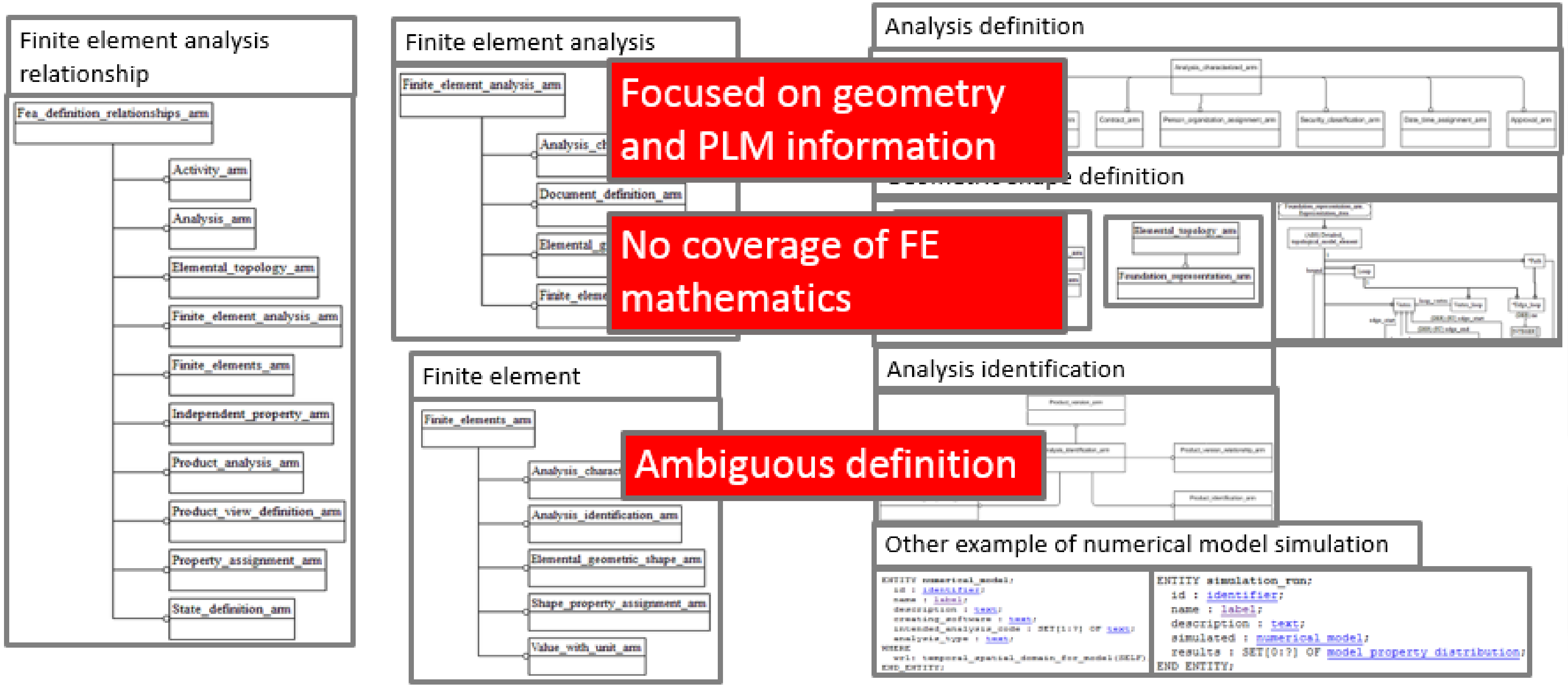
- HDF5
- UNV
- Gmsh
- VTK

No standard for mesh data

Workflow support

- Pro-Step Sim PDM 4.2 (xDM for PDM)
- FEDES
- CAE-Pronet

AP209 (v2014)-based FEA model description



Ref: ISO 10303-209:2014(E) - Application protocol: Multidisciplinary analysis and design

Koneksys

Impact of missing FEA standard

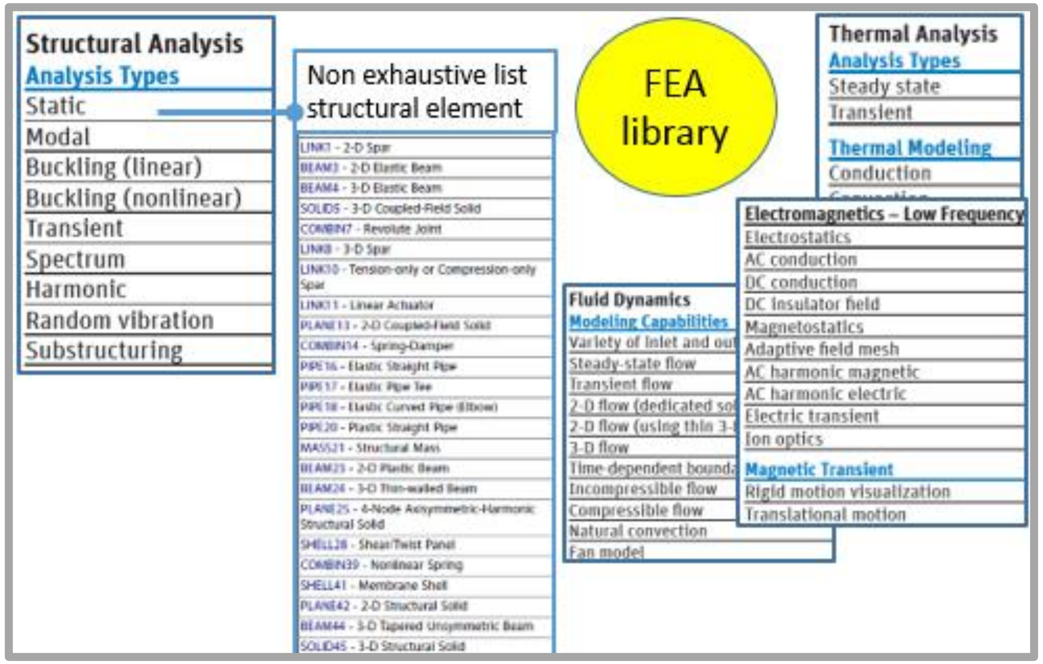
- Interoperability is compromised
- Impact on reusability (custom code)
- communication between system and FEA engineers is not efficient
- No open-standard

Outline

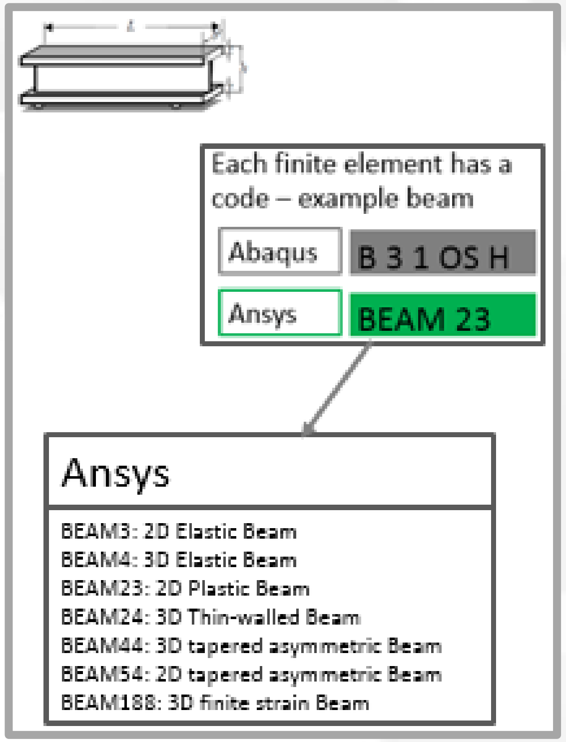
- Introduction and motivation
- Challenges in FEA standardization
- New proposed FE mathematics description
- Validation
- Next steps and summary

Koneksys

Challenge 1: Capturing model information



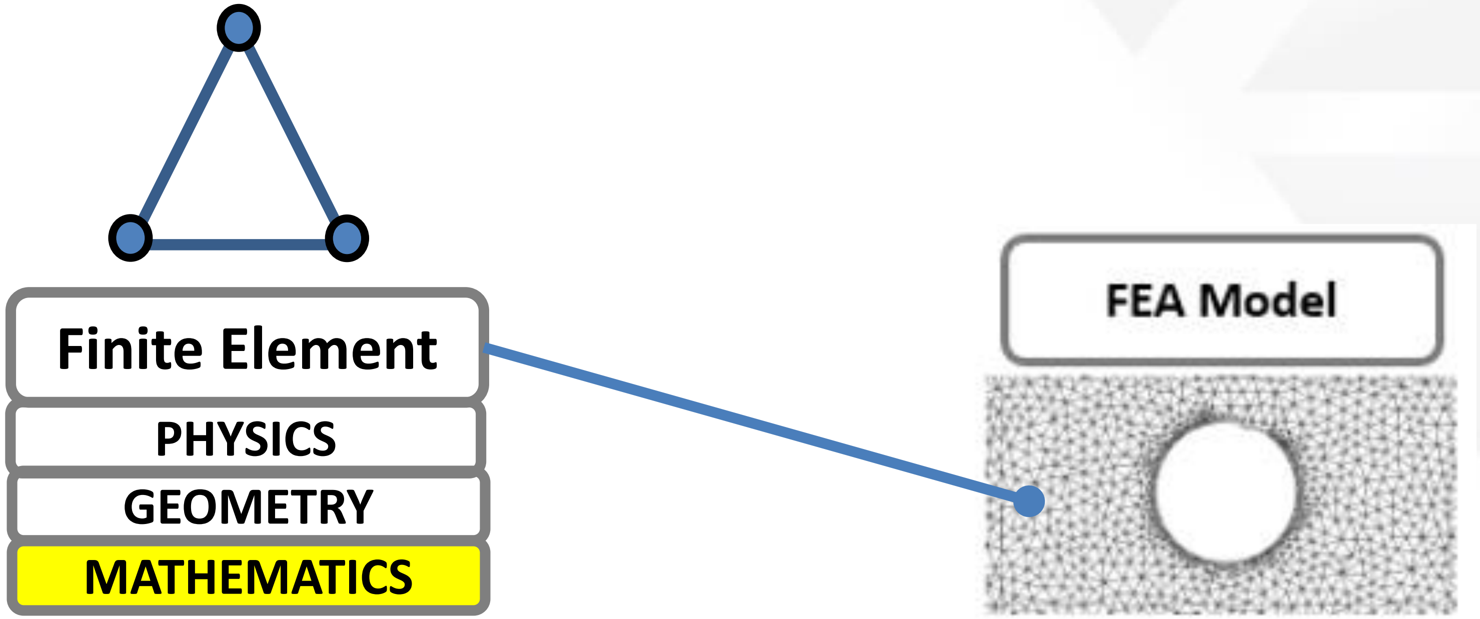
Ref: Ansys capabilities overview



Problem of encoding one model

Ambiguity

The method: decomposition and reuse

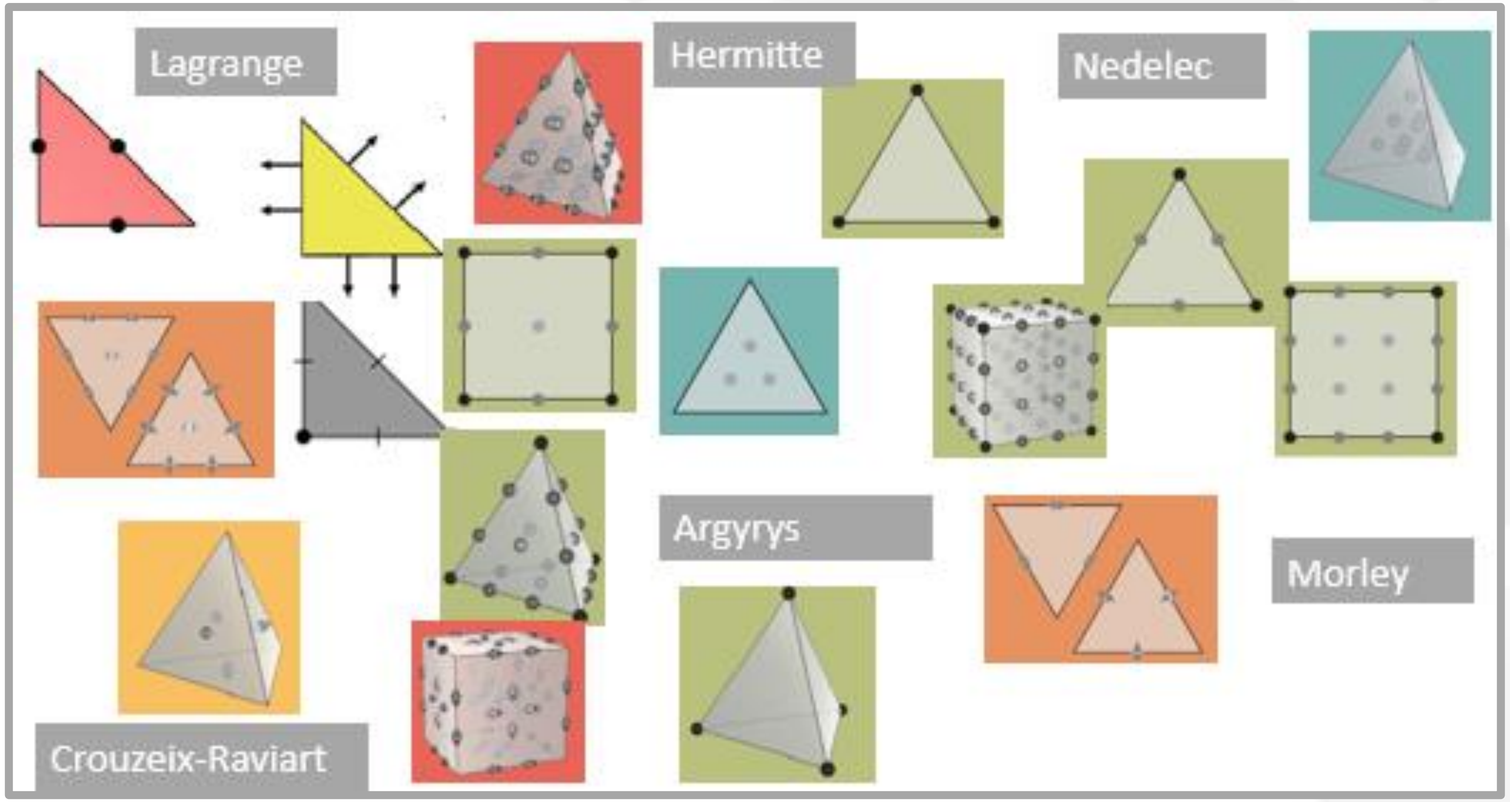


Start with the definition of finite element mathematics

Challenge 2: Describing finite element mathematics

Literature names are not descriptive

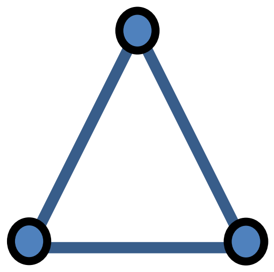
Difficult to create an ontology



Logg A. et Al., Automated solution of differential equation by the finite element method, 2012, Springer
Logg A., Arnold D., periodic table of finite elements, 2014, Siam News

Removing ambiguity?

1 finite element



Many Names

Linear simplex
Linear triangle
Linear Lagrange element…

1 Reference

Outline

- Introduction and motivation
- Challenges in FEA standardization
- New proposed FE mathematics description
- Validation
- Next steps and summary

Ciarlet's definition of FE mathematics

FE maths fully defined as a triple

Geometry

+

PE: Point evaluation
FD: First derivative
SD: second derivative....

Functional requirement or degrees of freedom

+

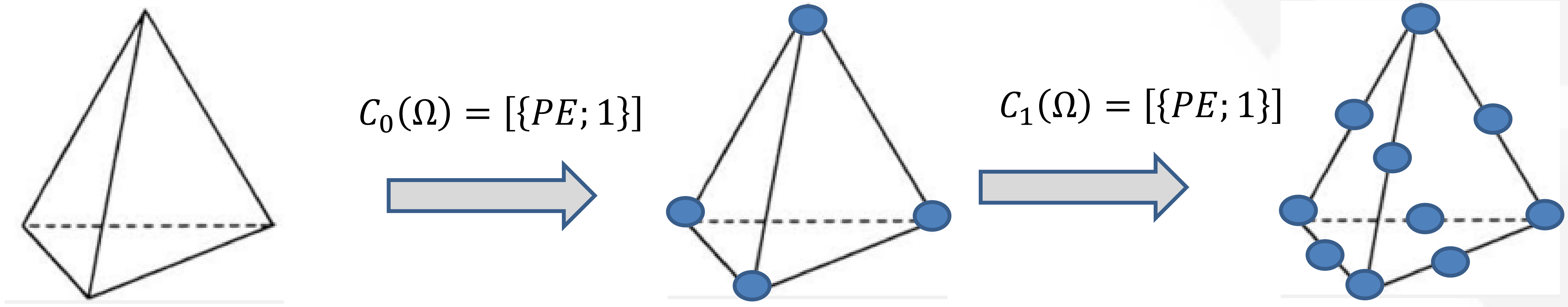
$ax^2 + bx + c$

Polynomial

New FE mathematics description: Assigning requirements to the geometry

Name: Element XY

- Tetrahedron
- $C_1(\Omega) = [\{PE; 1\}]$
- $C_0(\Omega) = [\{PE; 1\}]$
- Polynomial ref.

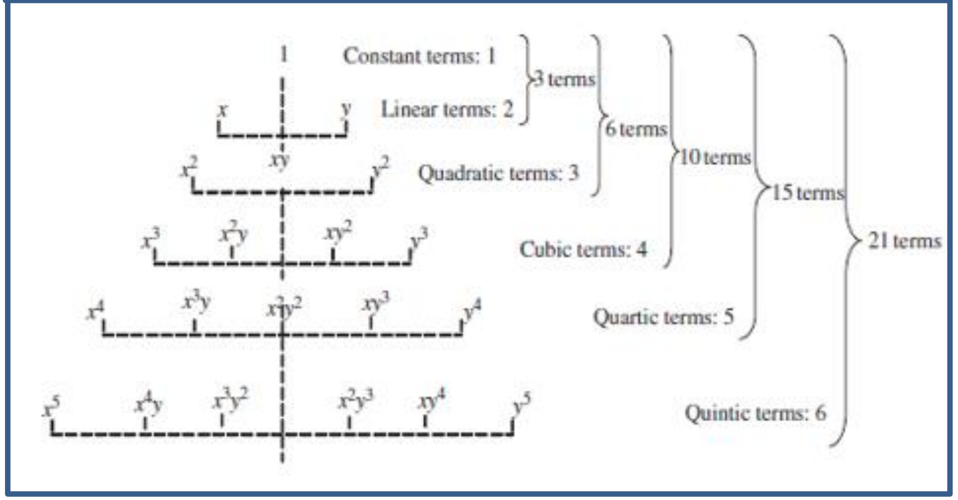


Tetrahedron

Polynomial basis dictionary

A polynomial is composed of monomials

Pascal triangle of **2-dimensional** monomials



Monomials can be ordered in a dictionary

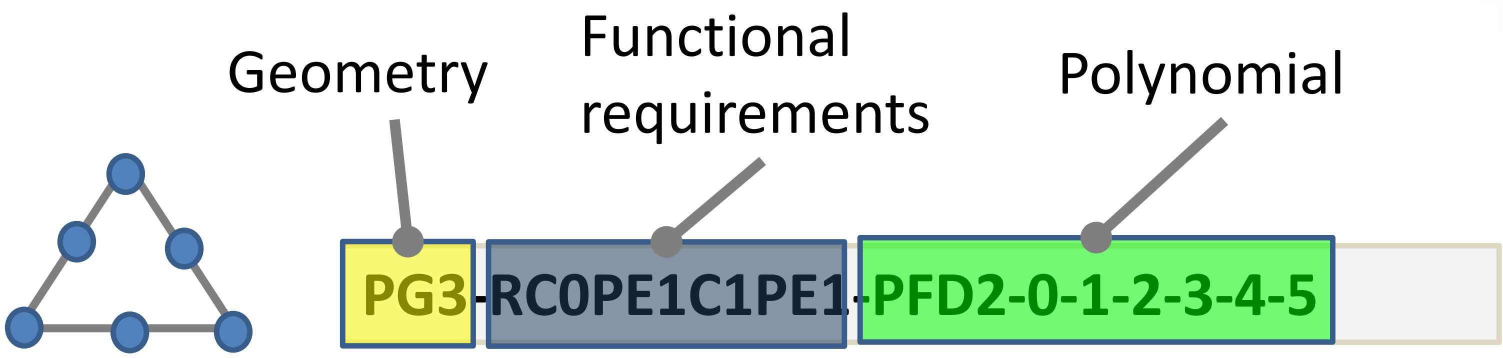
Graded Lexicographic ordering

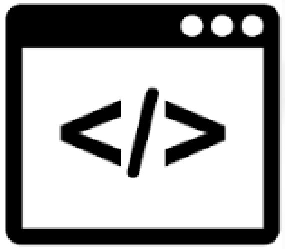
$$1 < y < x < y^2 < xy < x^2 < y^3 < x^2y \ldots$$

0 1 2 3 4 5 6 7

Koneksys

Encoding FE mathematics



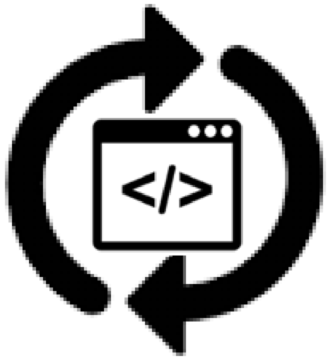


Custom code

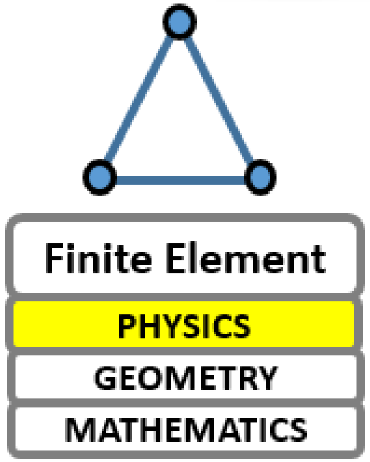
This specification provides non ambiguous information for code implementation

Reusing the new FE description for physics

- line
- $C_0(\Omega) = [\{PE; 1\}]$
- D1-0-1



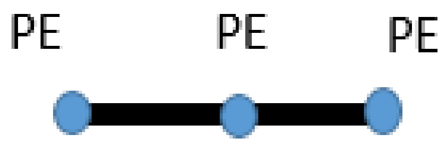
REUSE



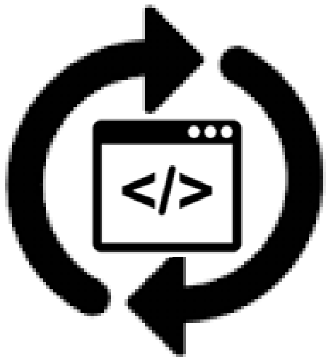
- line
- $C_0(\Omega) = [\{PE; 1\}]$
- D1-0-1
- Temperature
- Type: Scalar

- line
- $C_0(\Omega) = [\{PE; 1\}]$
- D1-0-1
- Displacement
- Type: Vector

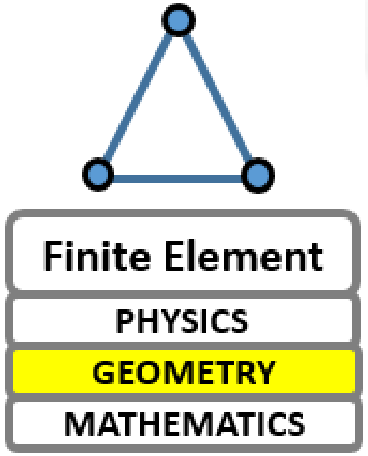
Reusing the new FE specification for geometry description

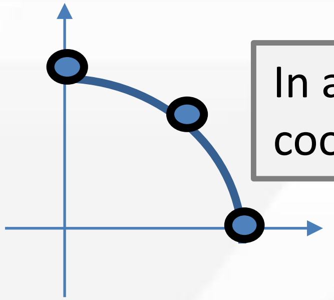


- Line
- $C_1(\Omega) = [\{PE; 1\}]$
- $C_0(\Omega) = [\{PE; 1\}\ ]$
- D1-1-2-3



REUSE



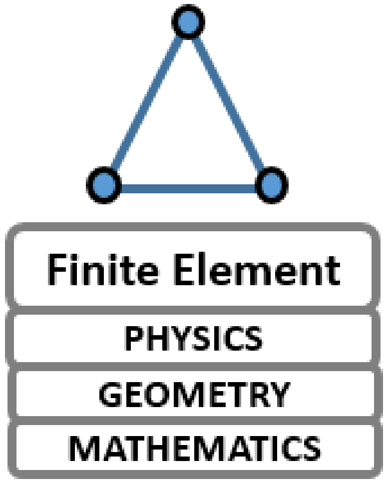


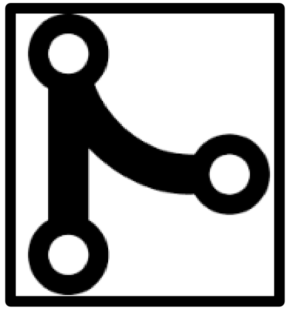
In a Cartesian coord. system

- Line
- $C_1(\Omega) = [\{PE; 1\}]$
- $C_0(\Omega) = [\{PE; 1\}\ ]$
- D1-1-2-3

- Dimension: 2
- Type: Cartesian

Merging information to describe parametric finite elements





MERGE

PHYSICS

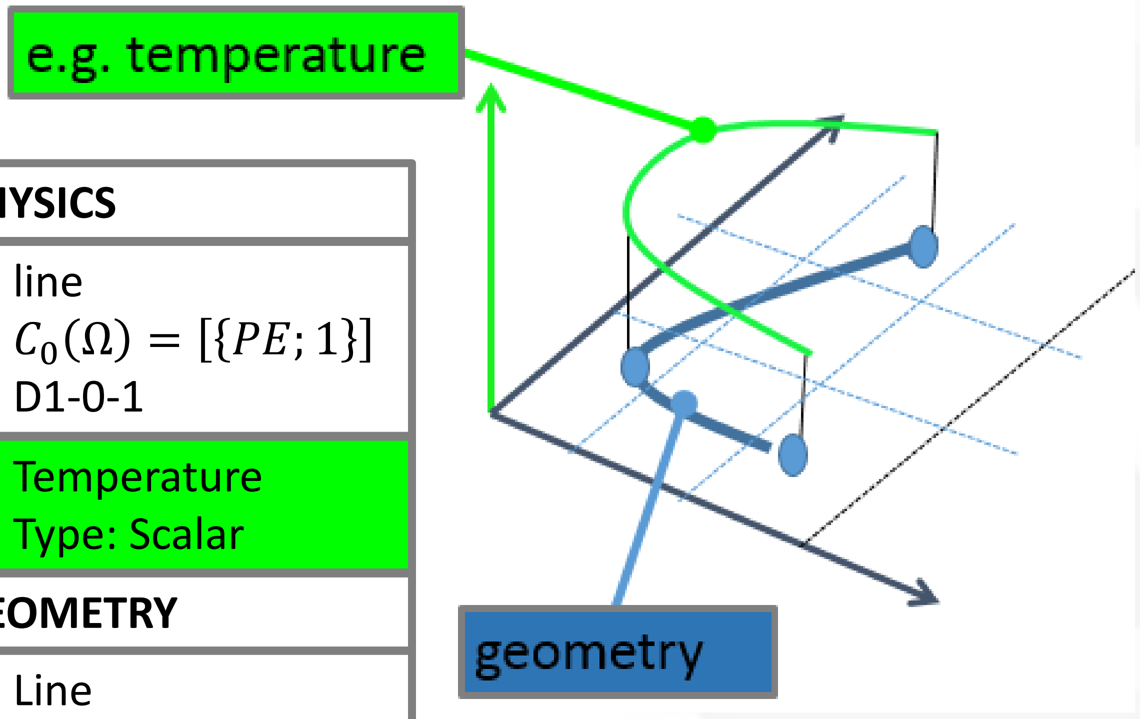
- line
- $C_0(\Omega) = [\{PE; 1\}]$
- D1-0-1

- Temperature
- Type: Scalar

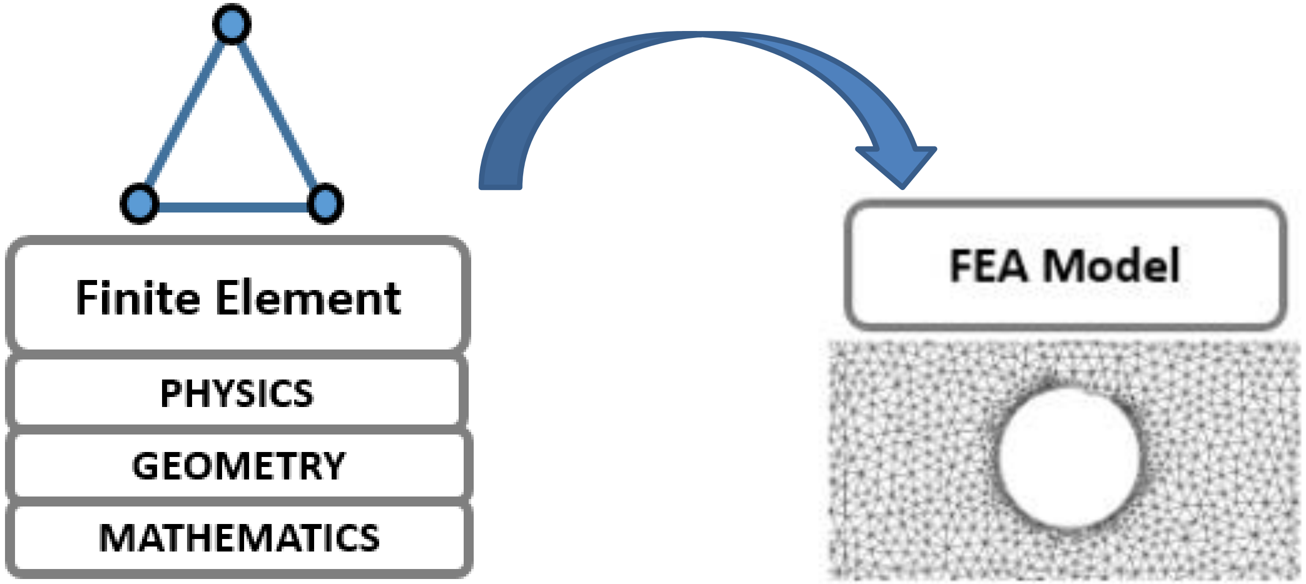
GEOMETRY

- Line
- $C_1(\Omega) = [\{PE; 1\}]$
- $C_0(\Omega) = [\{PE; 1\}\ ]$
- D1-1-2-3

- Dimension: 2
- Type: Cartesian



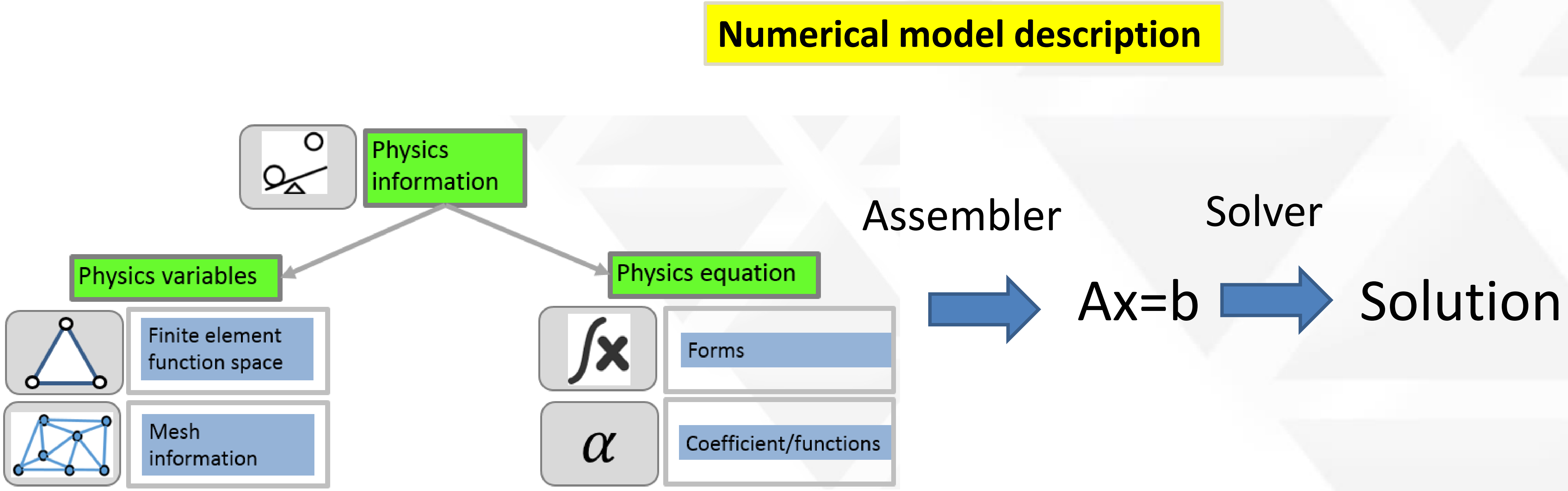
Next step of our work: Specifying the FEA model



Use the same principle:
Decomposition for reusability

Many physics use the same computational model…to be continued…

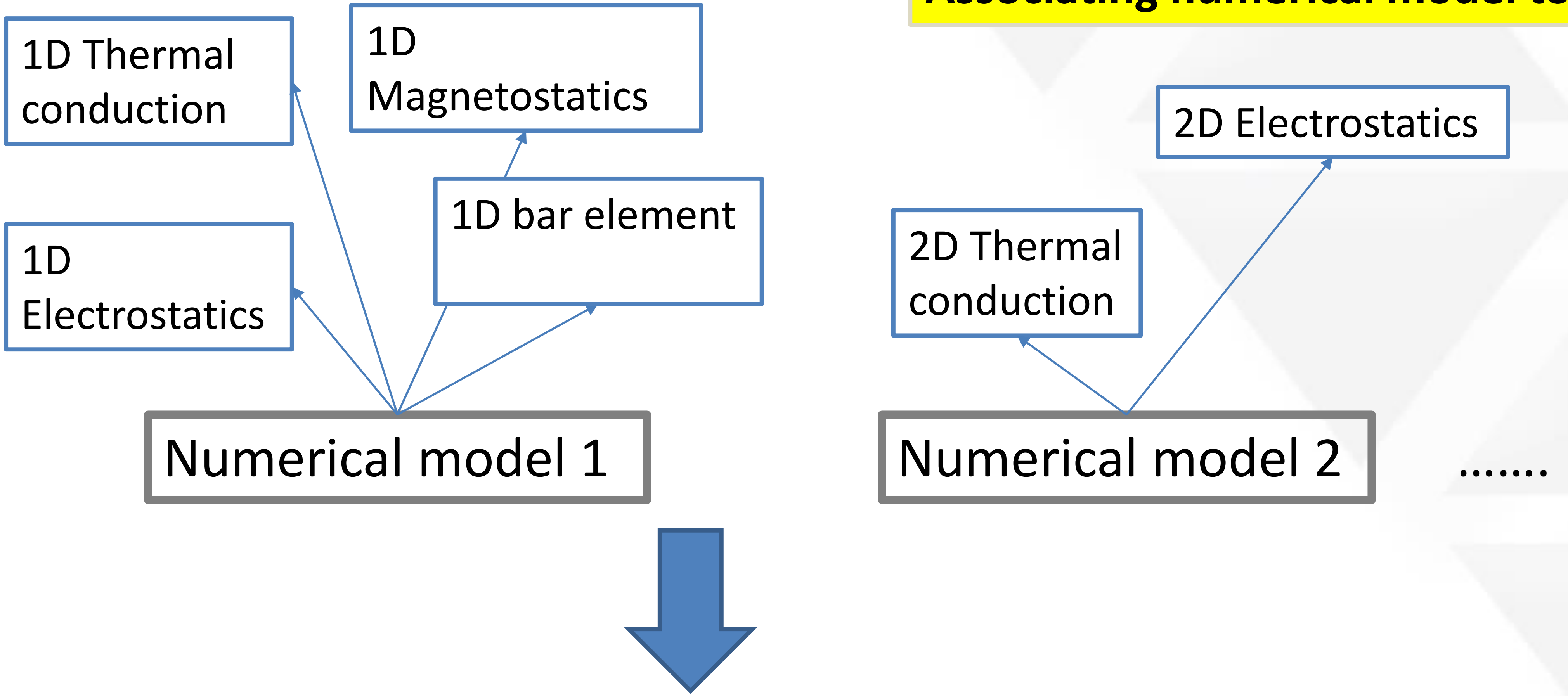
Unifying assembly process



Ref: M. S. Alnæs, A. Logg, K.-A. Mardal, O. Skavhaug, and H. P. Langtangen (2008) 'Unified Framework for Finite Element Assembly'.

Problem classification

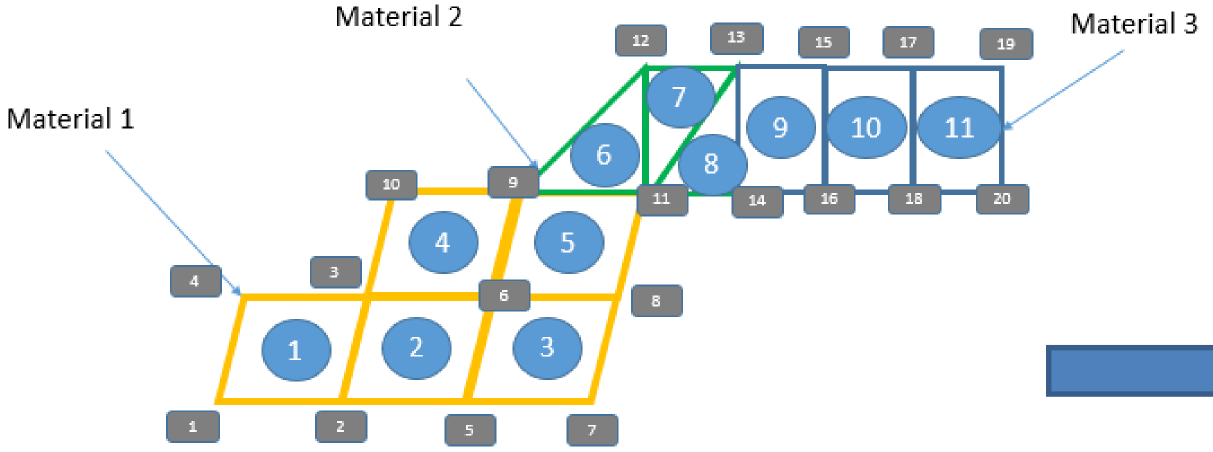
Associating numerical model to physics

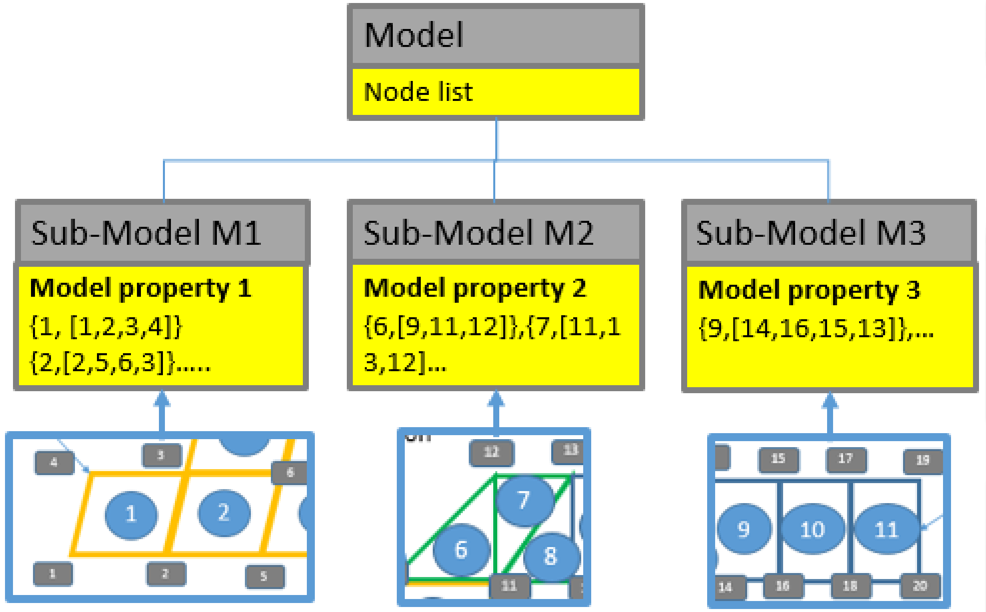


Model reuse

model input knowledge

Using SysML for model description: Example of domain sub-classing





Example domain sub-classing by material properties

Remove ambiguity of software specific vocabulary

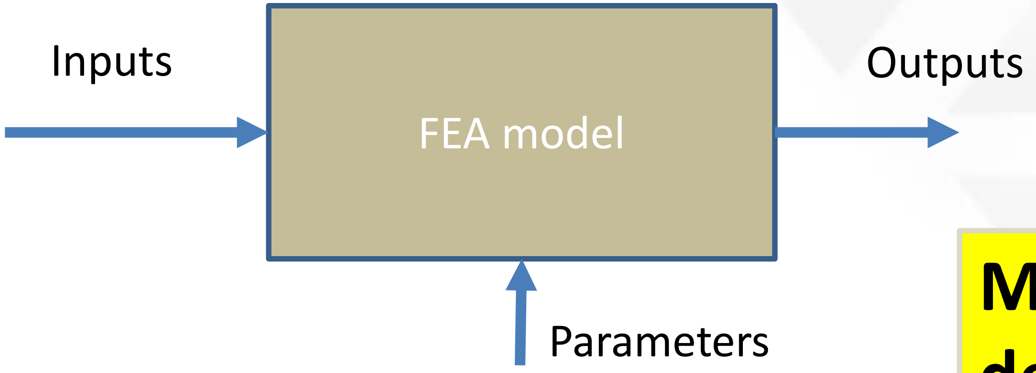
Example of Dirichlet boundary conditions for axial elastostatic problems

1D problem – vector is a scalar
2 combinations
u=0 (**fixed, pin, wall**) or u=free

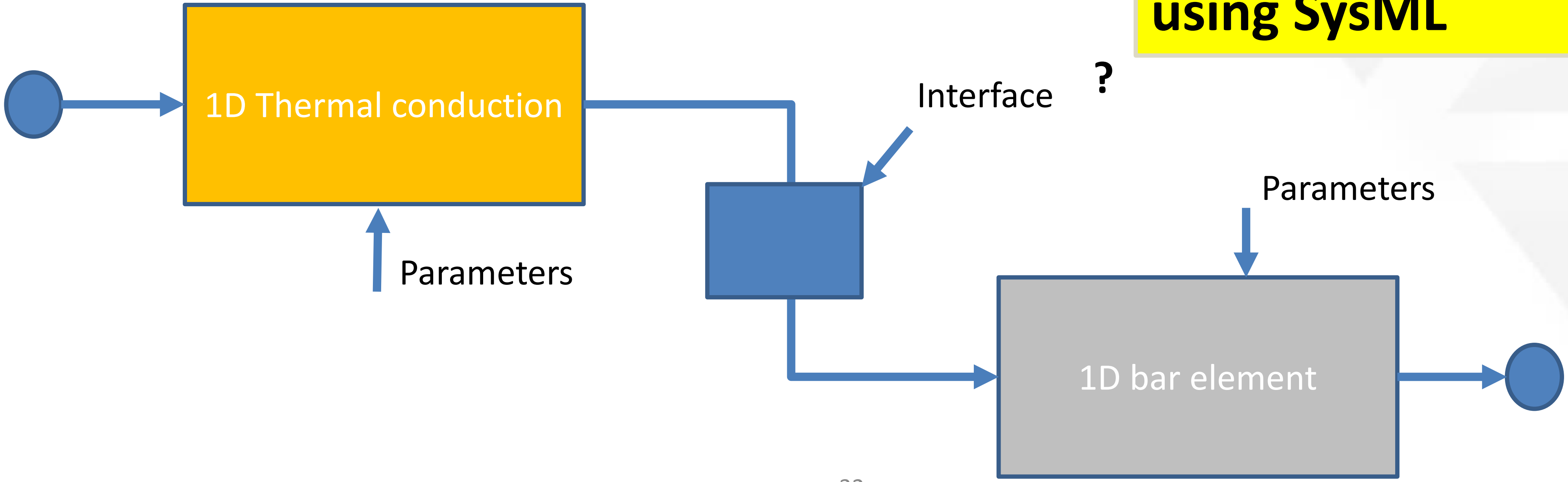
2D problem – vector has 2 components.
4 combinations
ux=0 and uy=0 *(**vocab: fixed, pin, wall**…)*
ux=free and uy=0 *(**vocab: roller, guide…**)*
ux=0 and uy=free *(**vocab: roller, guide…**)*
ux=free and uy=free *(**vocab: planar…**)*

Reconcile vocabulary of input deck files

Model reuse and connectivity



Midterm goal: block based description of FEA problem using SysML



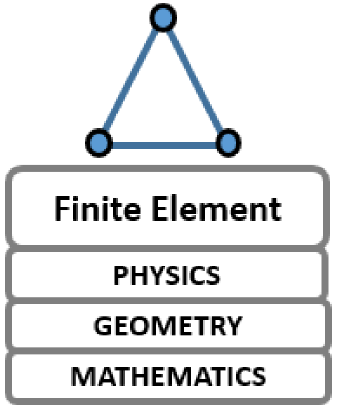
Outline

- Introduction and motivation
- Challenges in FEA standardization
- New proposed FE mathematics description
- Validation
- Next steps and summary

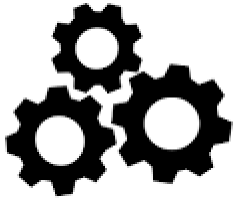
Python code to validate FE mathematics spec

- Model FEA specification in SysML
- FEA code to test the new proposed FEA spec
- FEA implemented in Python using object oriented concepts
- Using symbolic equations for interoperability
- integration with open source FreeCAD
- Code available on GitHub: https://github.com/koneksys/KFE/

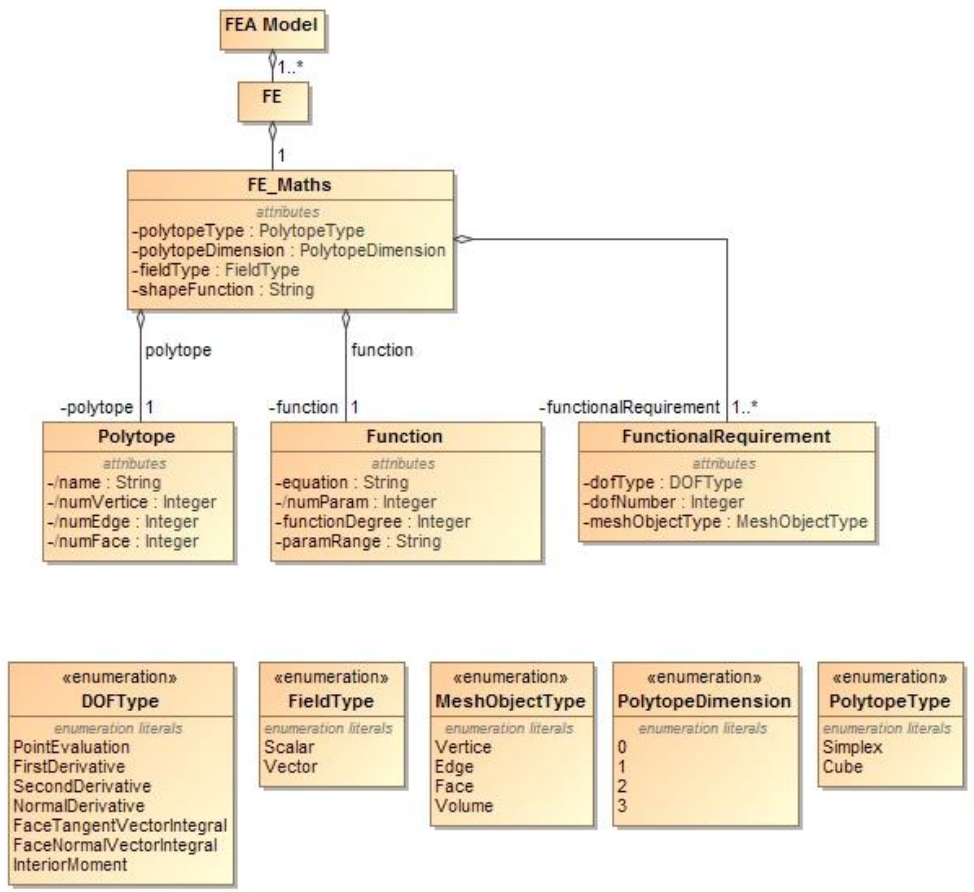
Translating FE description into SysML



New FE-description



SysML model



Code interoperability

Code uses symbolic equation for interoperability

Symbolic equation program (i.e. Sympy)

- HTML specification
- Text
- Equation image
- Plot image

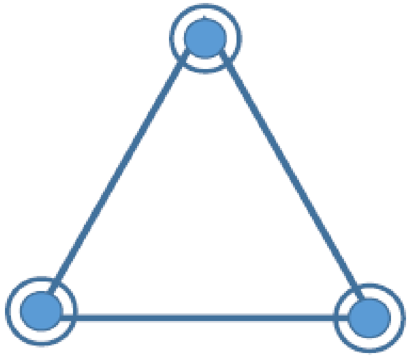
MathML info

Code generator

Koneksys

Information aggregation

Aggregation of finite element information in a single python object



Name: T1 element

- triangle
- $C_1(\Omega) = [\emptyset]$
- $C_0(\Omega) = [\{PE; 1\}, \{FD; 1\}]$

```
▼ self = {Femesh} <__main__.Femesh object at 0x04B38090>
  ▼ verticelist = {list} [<__main__.Vertice object at 0x04B38030>, <__main__.Vertice object
      __len__ = {int} 3
    ▼ 0 = {Vertice} <__main__.Vertice object at 0x04B38030>
      ▶ coordinates = {list} [0, 0]
      ▶ funreq = {list} [<Doftype.pointevaluation: 1>, <Doftype.firstderivative: 2>]
      ▶ index = {list} [0]
    ▼ 1 = {Vertice} <__main__.Vertice object at 0x04B380B0>
      ▶ coordinates = {list} [0, 1]
      ▶ funreq = {list} [<Doftype.pointevaluation: 1>, <Doftype.firstderivative: 2>]
      ▶ index = {list} [1]
    ▼ 2 = {Vertice} <__main__.Vertice object at 0x04B380D0>
      ▶ coordinates = {list} [1, 1]
      ▶ funreq = {list} [<Doftype.pointevaluation: 1>, <Doftype.firstderivative: 2>]
      ▶ index = {list} [2]
```

Outline

- Introduction and motivation
- Challenges in FEA standardization
- New proposed FEA description
- Validation
- Summary

Koneksys

Summary

- Benefits of new FE mathematics specification based on algebraic topology:
  - Covering FE mathematics
  - Understandable to engineers who are not mathematicians
  - Simple and precise definition of a finite element
  - Covering information for implementing FE mathematics in FEA code
  - Can describe more FE elements than with descriptions based on Ciarlet/periodic table

- New FE mathematics specification will benefit integration between systems engineering and FEA
  - Traceability
  - Consistency/Synchronization
  - Reuse

Thank You!

Koneksys
Jerome Szarazi
t: +44(0)7736732512
e: jerome.szarazi@koneksys.com