DEVELOPMENT OF A HIERARCHICAL DATA FORMAT FOR MODELING, SIMULATION AND POSTPROCESSING IN STRUCTURAL MECHANICS AND ITS ECOSYSTEM

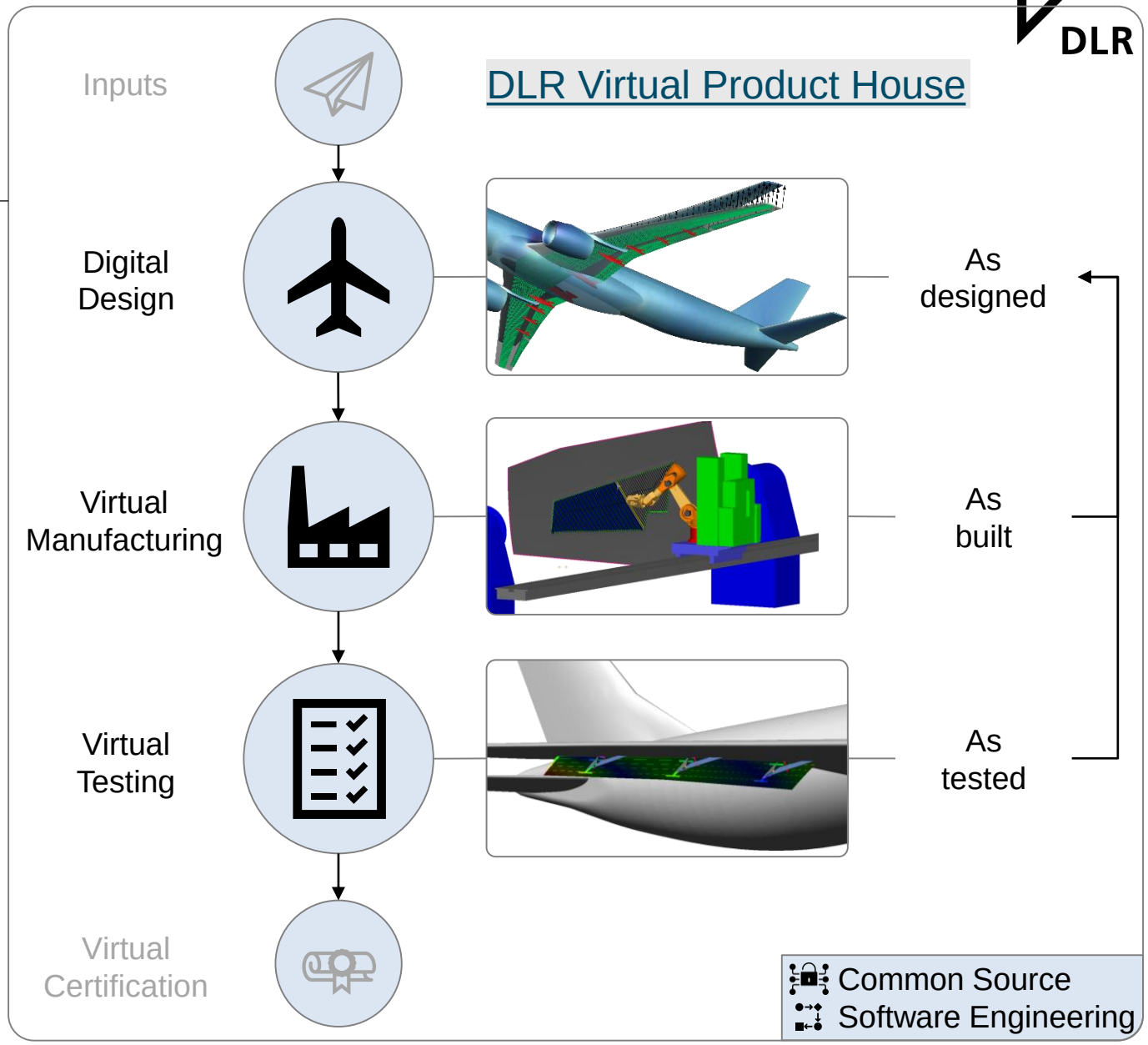
NAFEMS World Congress
18.05.2023
Martin Rädel, Jean Lefèvre, Andreas Schuster

DLR

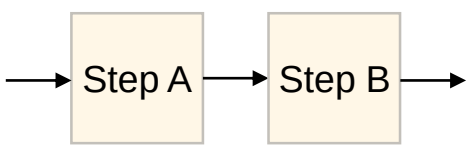
MOTIVATION

Task

- Virtual product development
  - End-to-end process
  - Multiple simulation steps
  - Multiple partners
- Methods:
  - Numerical (e.g. FEM, PD)
  - Semi-analytical (e.g. Ritz)
  - Analytical
- Different:
  - Solvers
  - Tool ecosystems
- Common ground: data

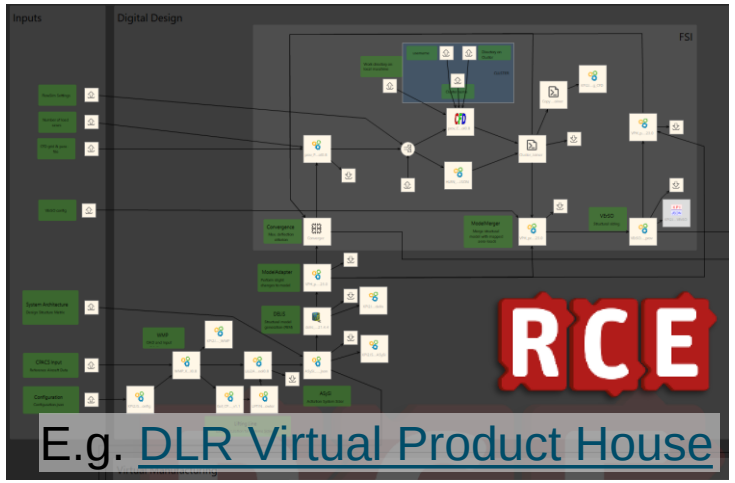


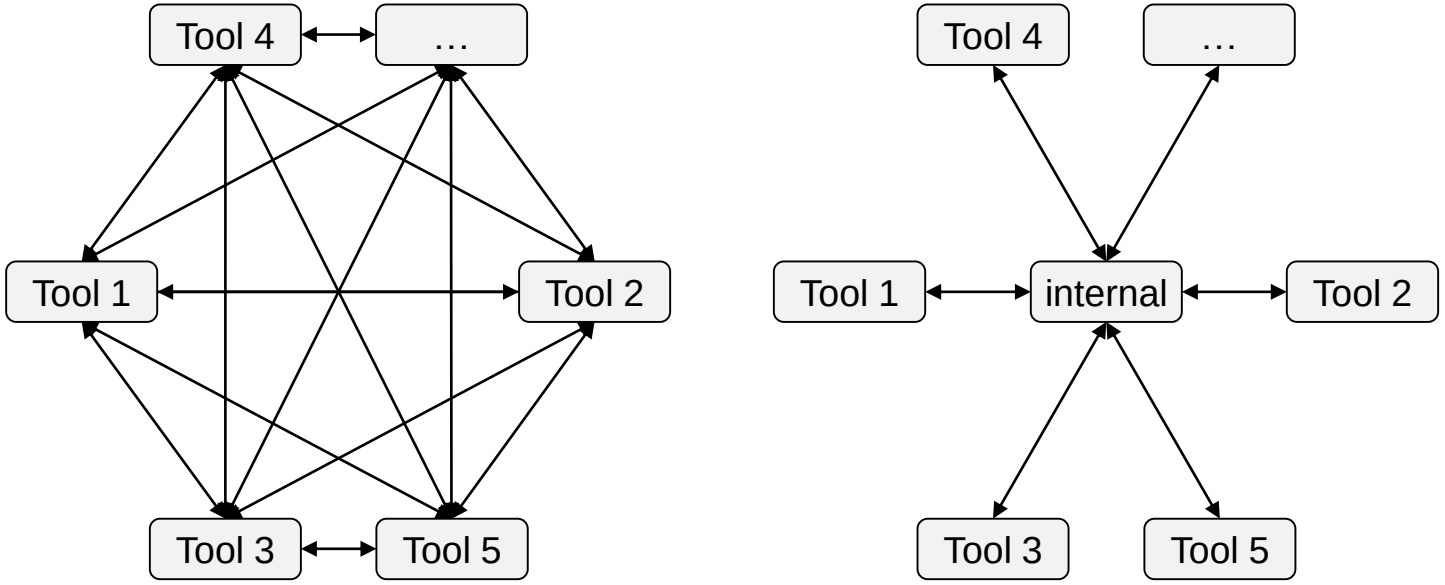
Task
Example: Aircraft moveable

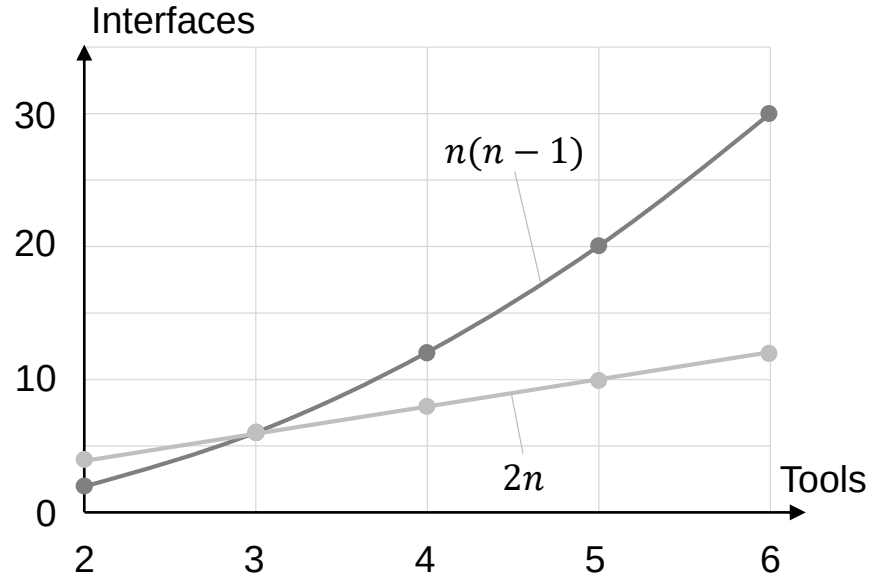


- Each tile
  - Function: Assessment → Process
  - Data: Input, Output → I & O

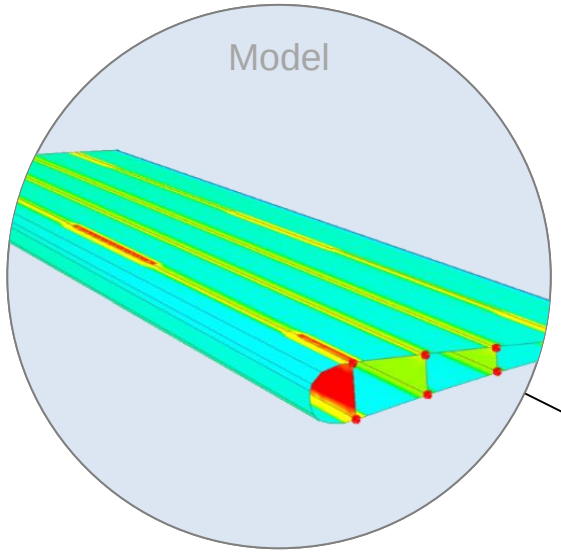
  } IPO
- Basically same information, vendor-specific formats



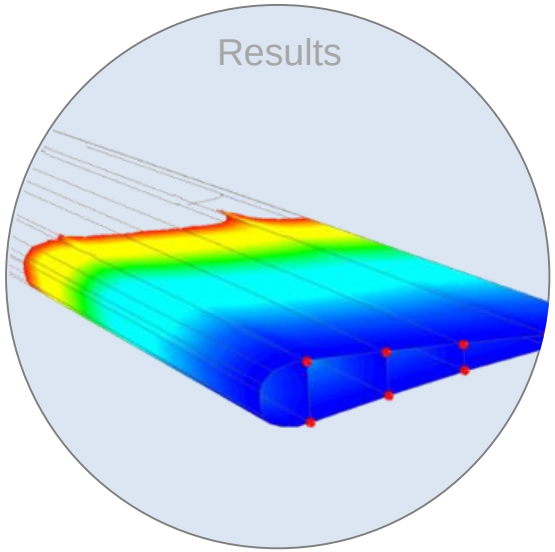




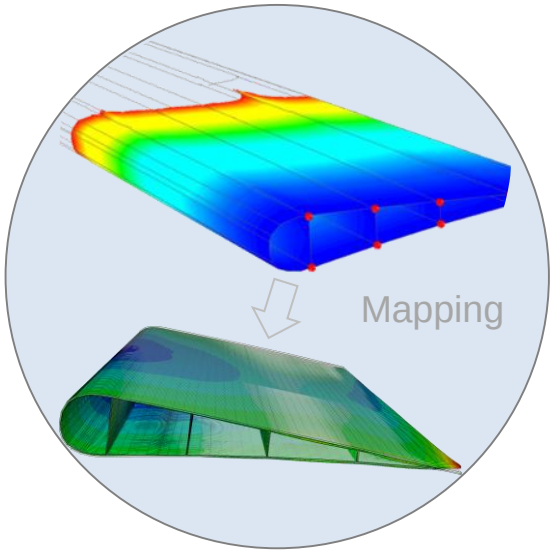
Task
Steps



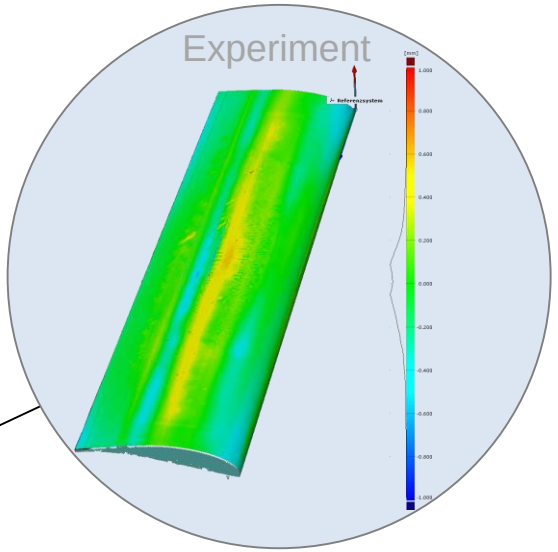
(Parametric) model generation
→ Input



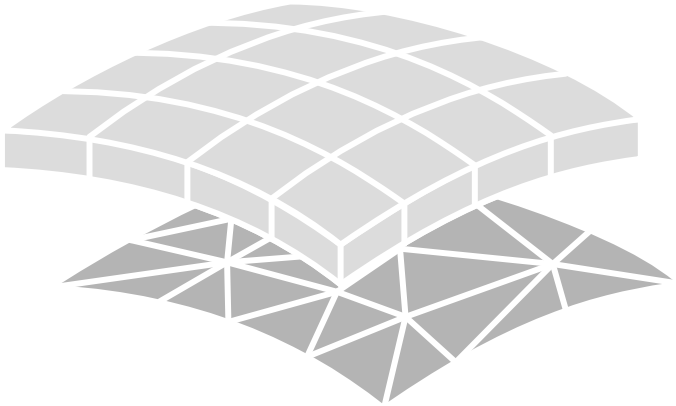
Postprocessing
→ Output



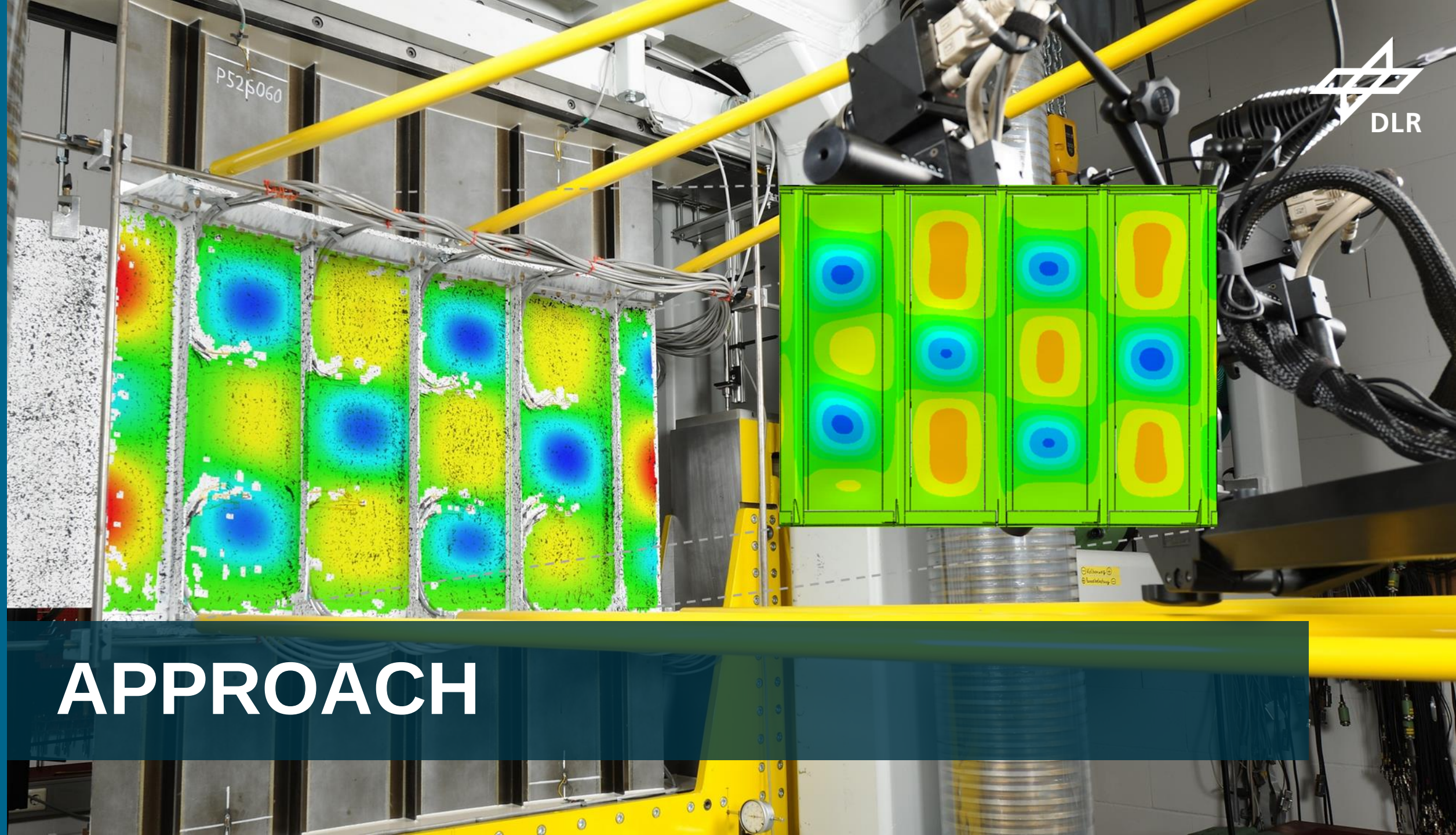
State transfer
→ Next input



Spatial & temporal measurements



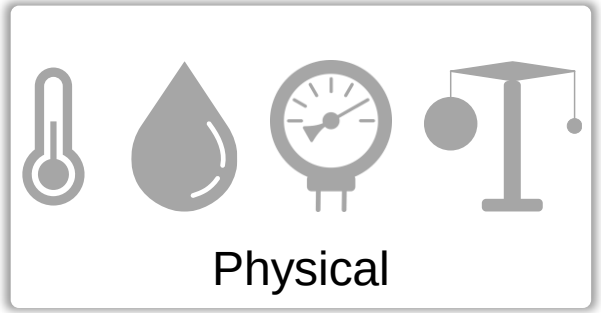
- Tool/solver-agnostic
- Standardized
- Programming-language independent

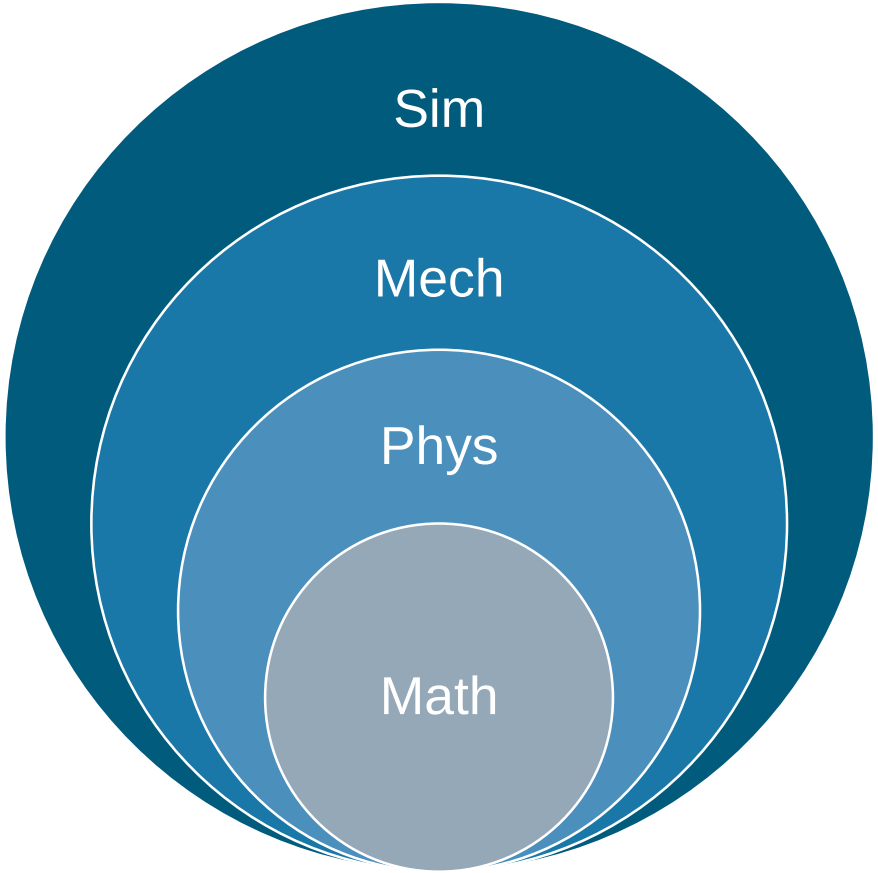


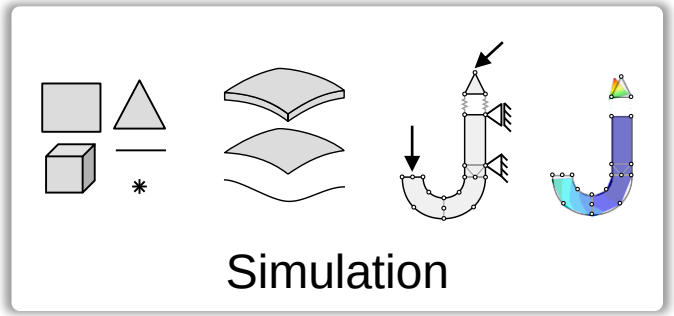
APPROACH

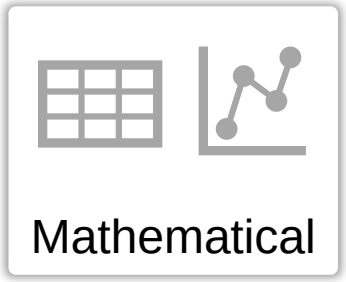
Approach
Hypothesis

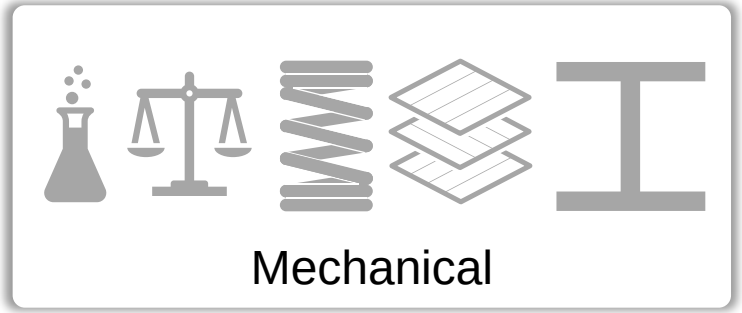
- Modular language vocabulary → based on level of assumptions & theories
- Reuse between levels





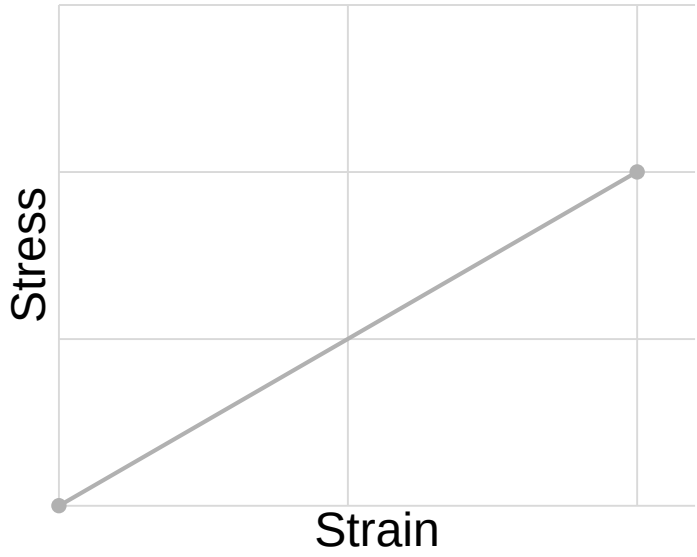




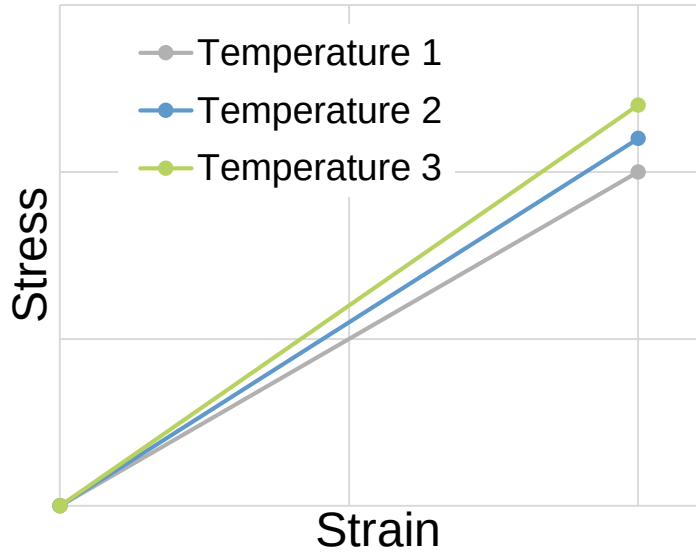


Approach
Syntax

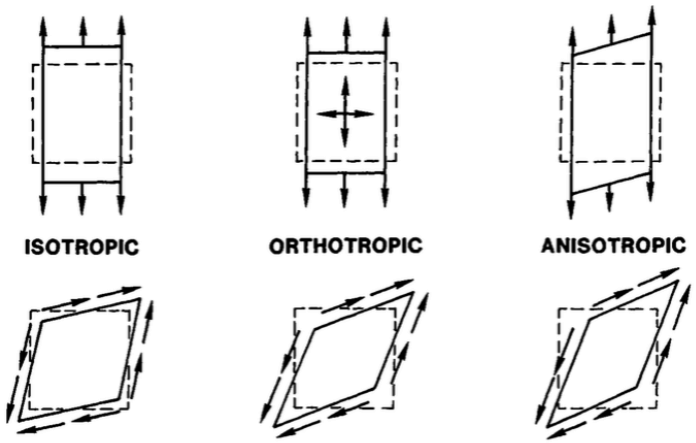
- Base entities
  - Description:
    - Entity
    - Model
    - Value
  - Example
    - ElasticityEvolution
    - Expandable subtypes
- Compound entities
  - Consist of base entities
  - Only entity, value



→ ConstantModel

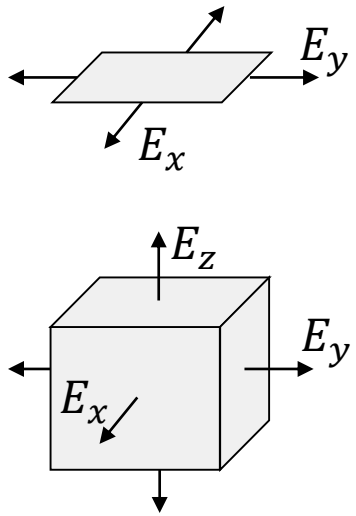


→ ConditionDependentModel



DOI: 10.13140/RG.2.2.31723.59682

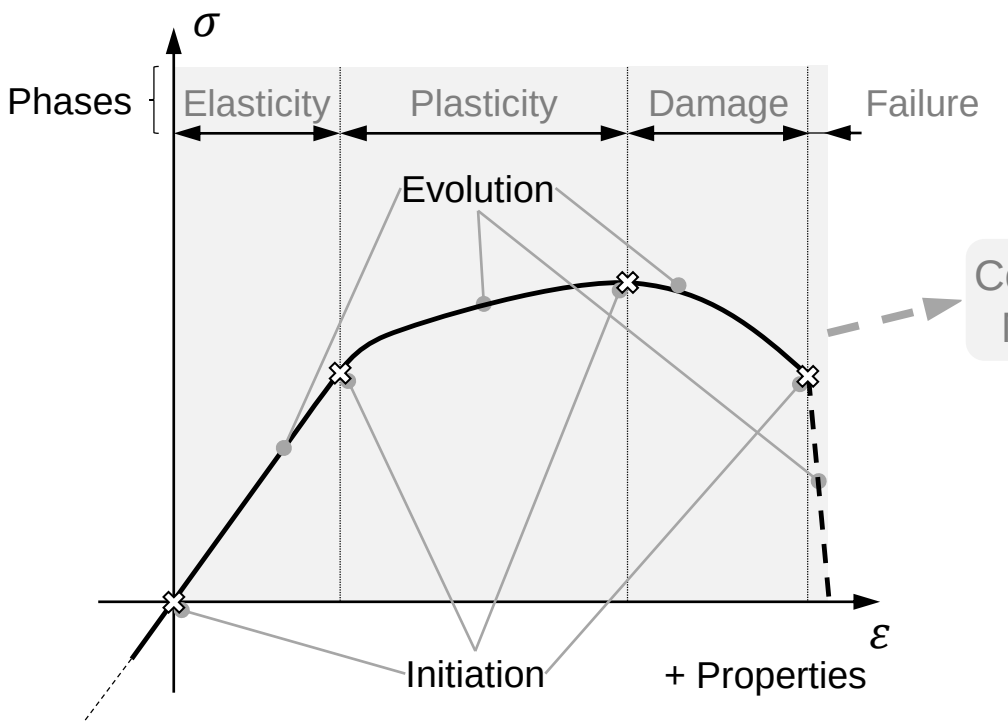
Behaviours

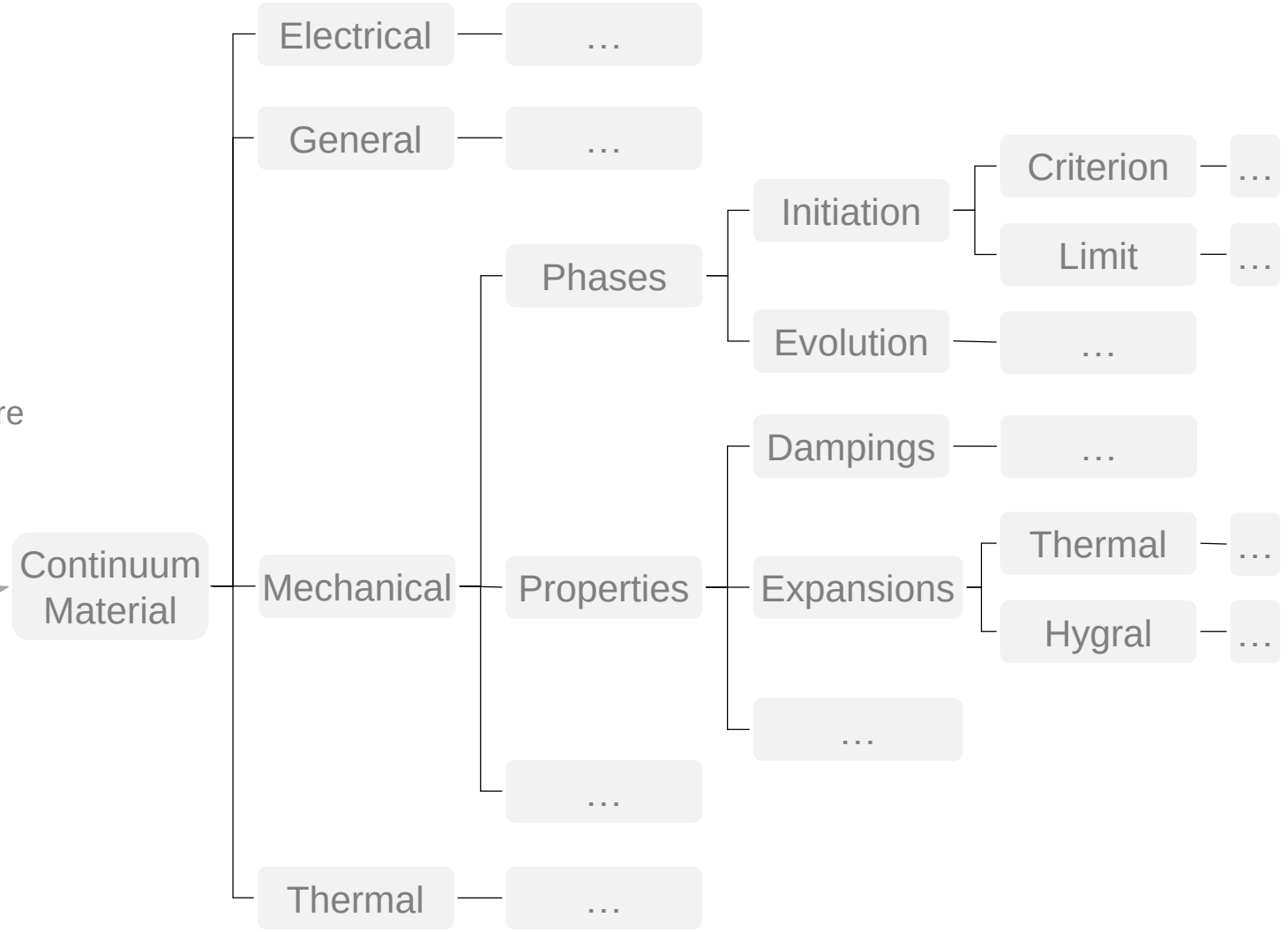


Idealizations

Approach
Hierarchy development

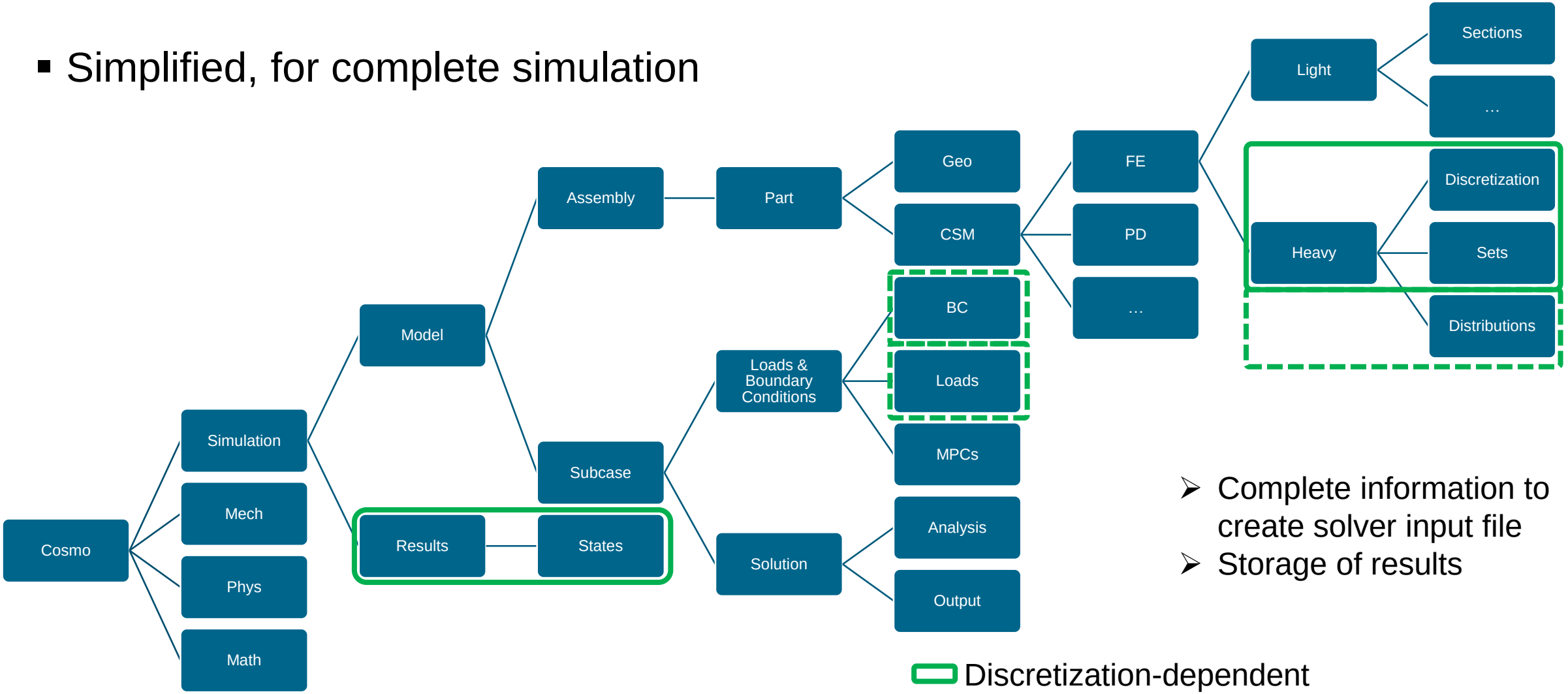
- Goal: expandable, modular
- Example:
  - Entity: Material
  - Value: ContinuumMaterial





Approach
Hierarchy development

- Simplified, for complete simulation

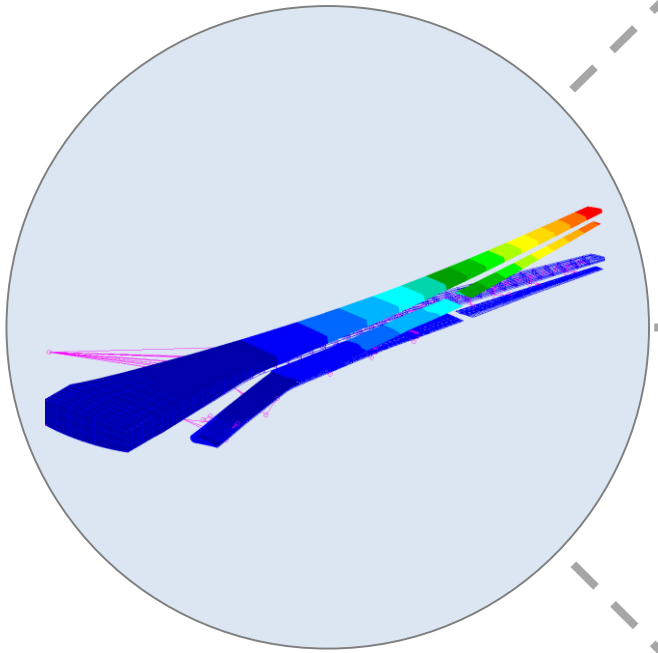


- Complete information to create solver input file
- Storage of results

Discretization-dependent

Discretization-dependent, type-dependent

Approach
Hierarchy realization



Numerical implementation data

- Nodes
- Elements
- …

- Results

Heavy data

VMAP

References

Connecting data

- Sections
- Loads & BC
- …

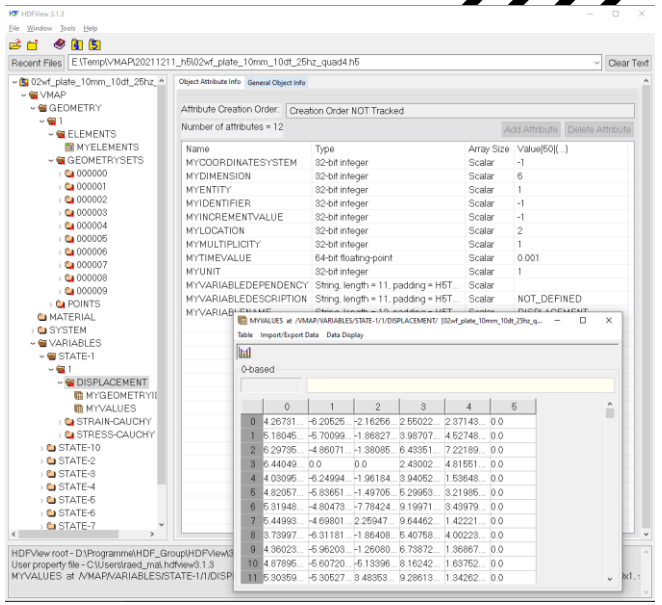
- Solution

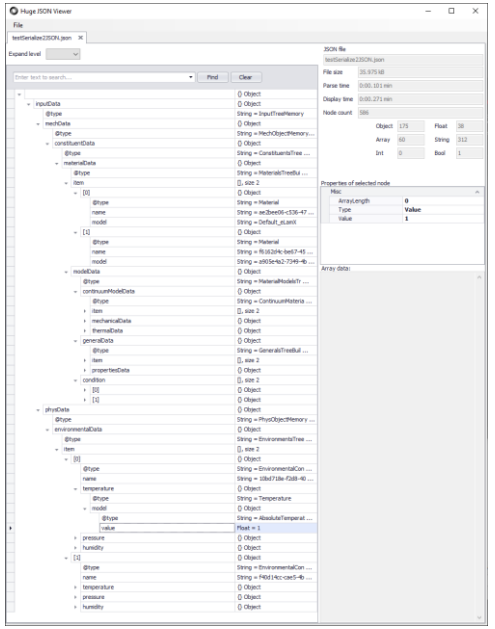
Light data

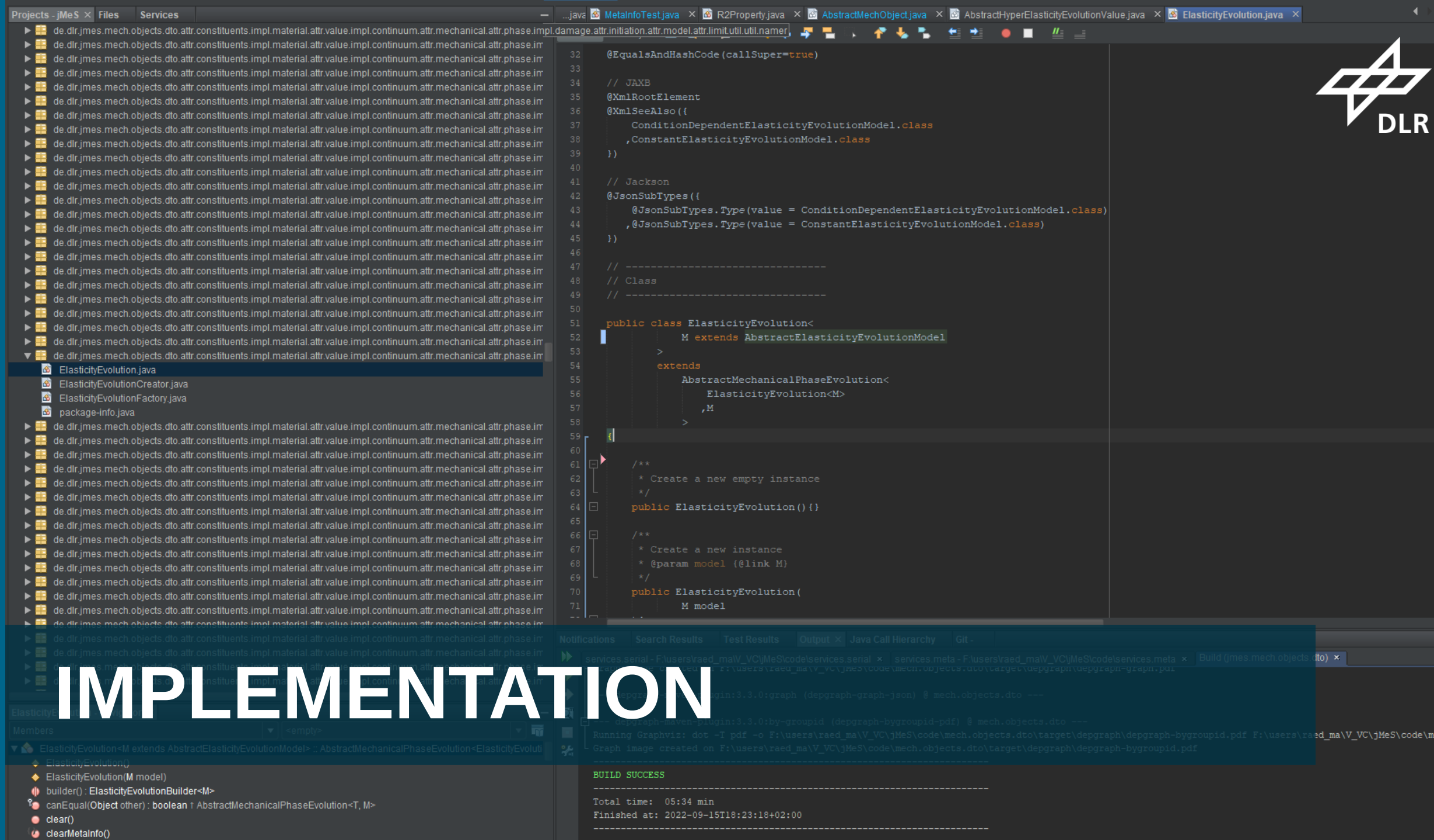
cosmo

Math, physics & mech. idealization data

- Materials
- Composites
- Crosssections
- …







IMPLEMENTATION

Implementation
Cosmo standard creation

- Hierarchical, object-oriented data structure
  - Implementation in Java-based backend jMeS (Java Mechanics Suite)
  - Usage of JAXB, Jackson annotations across hierarchy to denote
    - Abstraction
    - Attribute types
- Annotations understood by de-/serialization libraries
  - E.g. Jackson, MOXy, …
  - Allows creation of schema, e.g. XML
  - Creation of XML, JSON, YAML, …

Abstract superclass

```
// JAXB
@XmlTransient
@XmlAccessorType(XmlAccessType.FIELD)

// Jackson
@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.PROPERTY)
public abstract class AbstractMechObjectProperty<
            T extends AbstractMechObjectProperty<T,M>
           ,M extends AbstractMechObjectModel
        >
```

Implementation

```
// JAXB
@XmlRootElement
@XmlSeeAlso({
    ConditionDependentElasticityEvolutionModel.class
   ,ConstantElasticityEvolutionModel.class
})

// Jackson
@JsonSubTypes({
    @JsonSubTypes.Type(value = ConditionDependentElasticityEvolutionModel.class)
   ,@JsonSubTypes.Type(value = ConstantElasticityEvolutionModel.class)
})
public class ElasticityEvolution<
```

Attributes

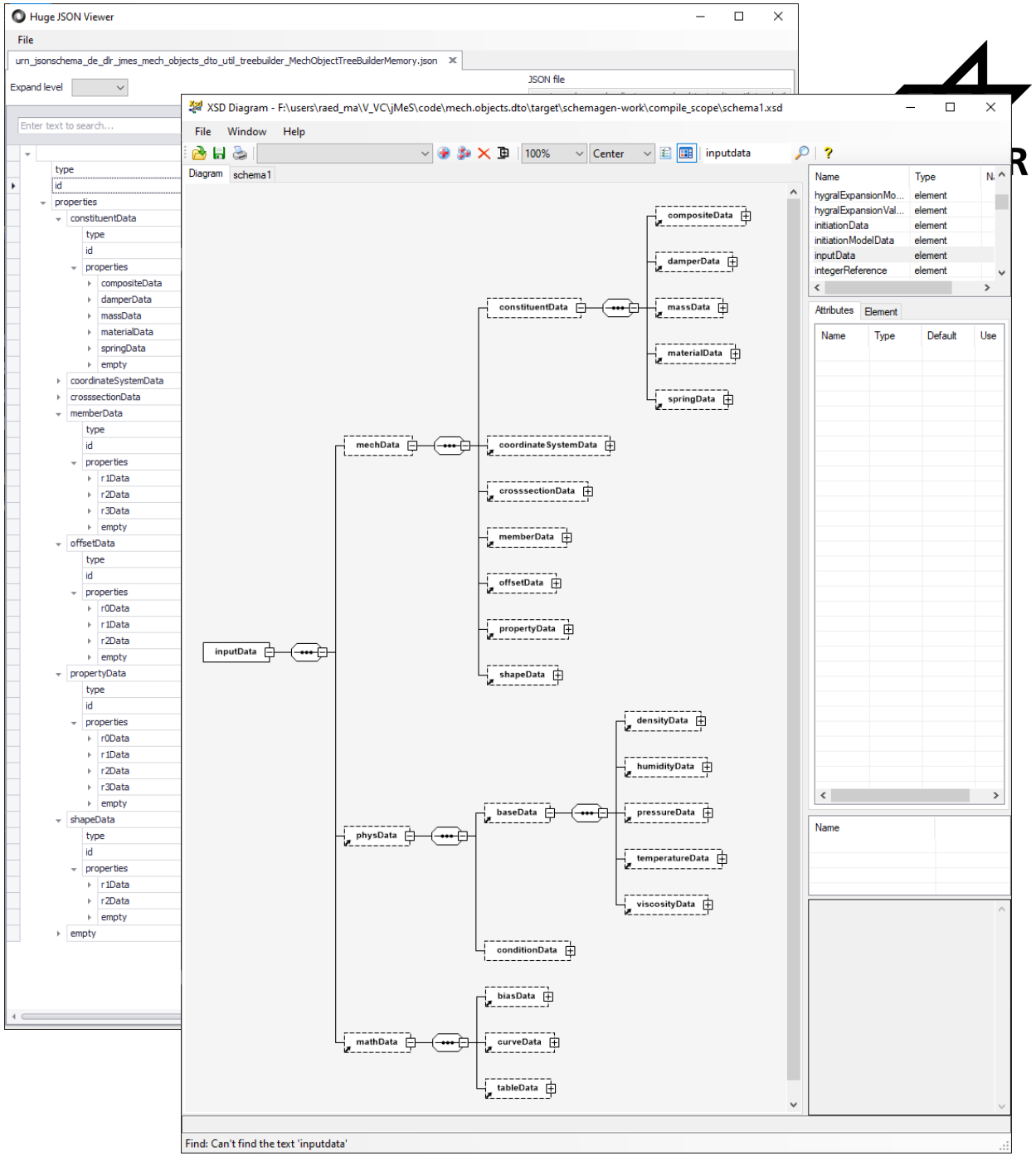
```
public abstract class AbstractMechObjectProperty<
            T extends AbstractMechObjectProperty<T,M>
           ,M extends AbstractMechObjectModel
        >
    @XmlAttribute
    @XmlID
    private String name = DEFAULT_NAME;

    @XmlIDREF
    protected M model;

    protected MetaInfo<? extends AbstractMetaInfoValue> metaInfo;
```

Implementation
Cosmo standard definition

- Automatic XSD & JSON schema generation during code compilation
- Automatic translation to code:
  - C++: codesynthesis
  - Python: generateDS
  - Java: Jackson
  - …
- Result:
  - ✓Tool/solver-agnostic
  - ✓Standardized
  - ✓Programming-language independent



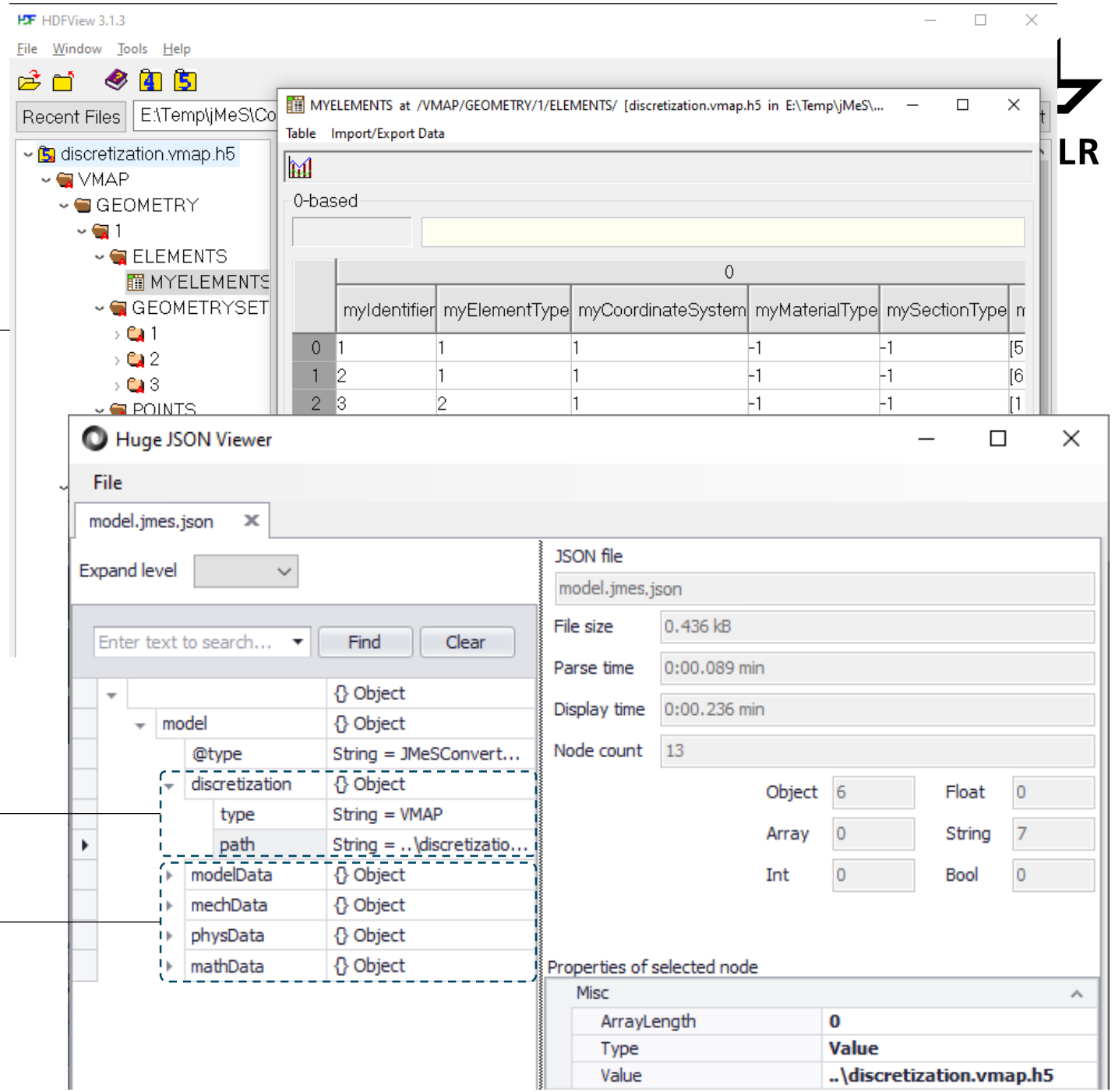
Implementation
Examples

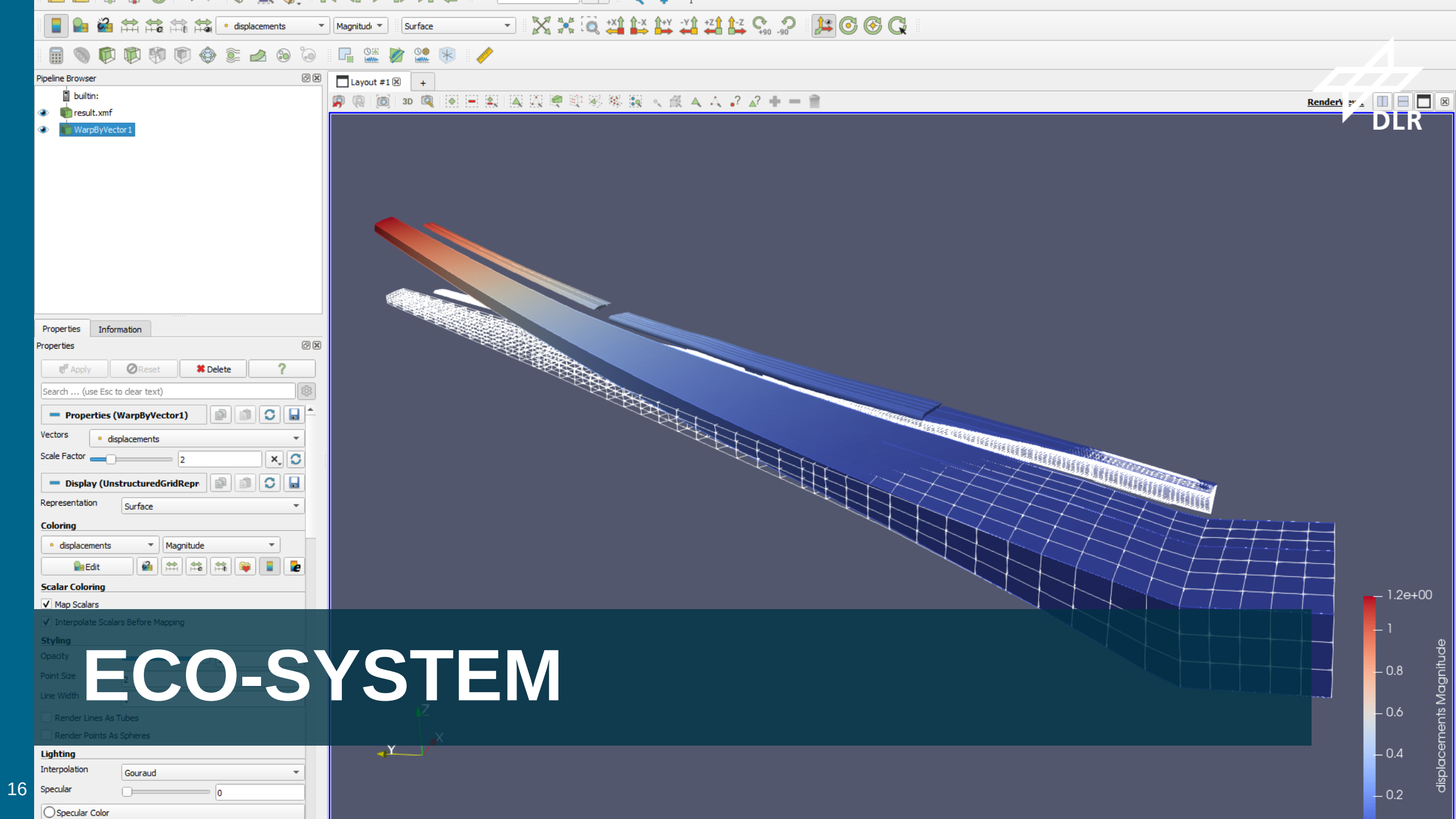
- Implementation

Link to heavy data file

Light data

- Meta data
- Solver-specific extensions
  - As meta information
  - E.g.
    - Element type managers
    - Solution parameters

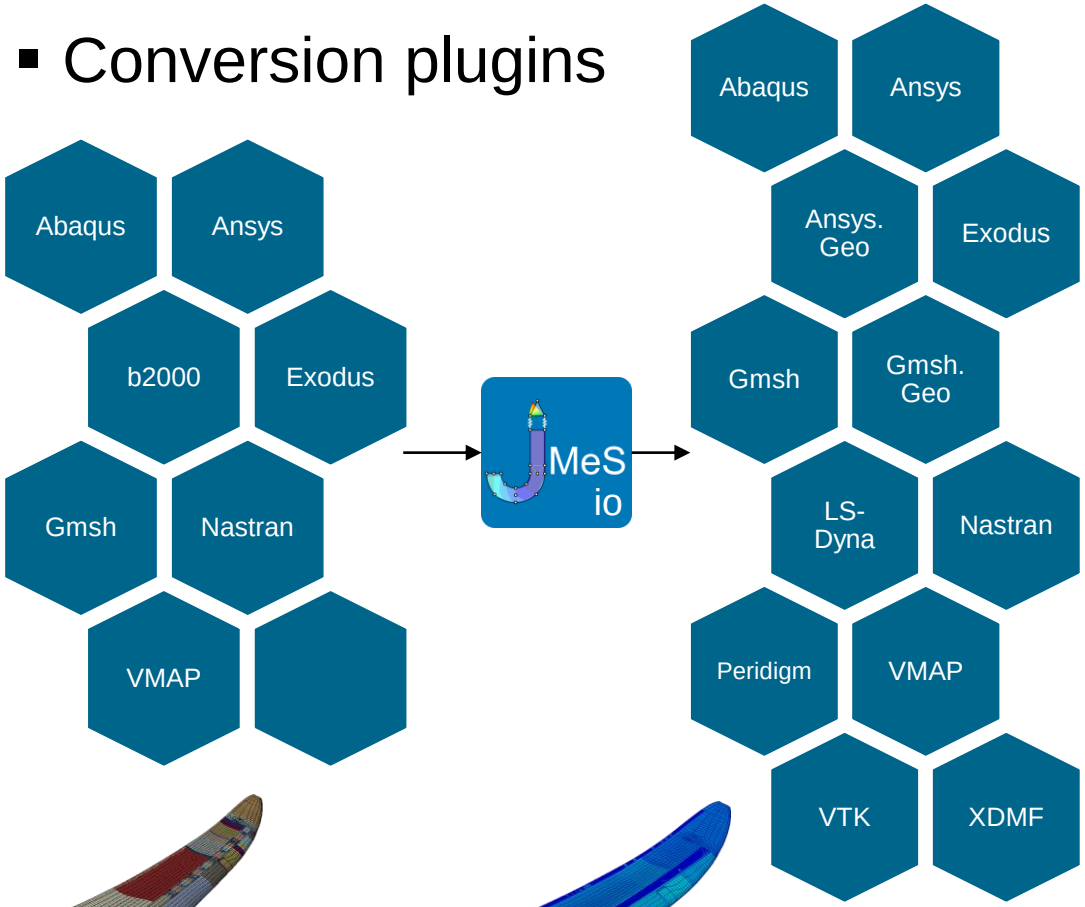


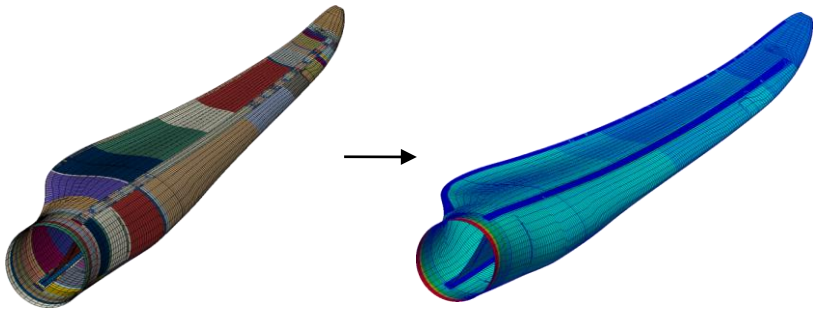

displacements
Magnitud
Surface
Pipeline Browser
builtin:
result.xmf
WarpByVector1
Layout #1
Properties
Information
Apply
Reset
Delete
Search ... (use Esc to clear text)
Properties (WarpByVector1)
Vectors
displacements
Scale Factor
2
Display (UnstructuredGridRepr
Representation
Surface
Coloring
displacements
Magnitude
Edit
Scalar Coloring
Map Scalars
Interpolate Scalars Before Mapping
Styling
Opacity
Point Size
Line Width
Render Lines As Tubes
Render Points As Spheres
Lighting
Interpolation
Gouraud
Specular
0
Specular Color
DLR
1.2e+00
1
0.8
0.6
0.4
0.2
displacements Magnitude

ECO-SYSTEM

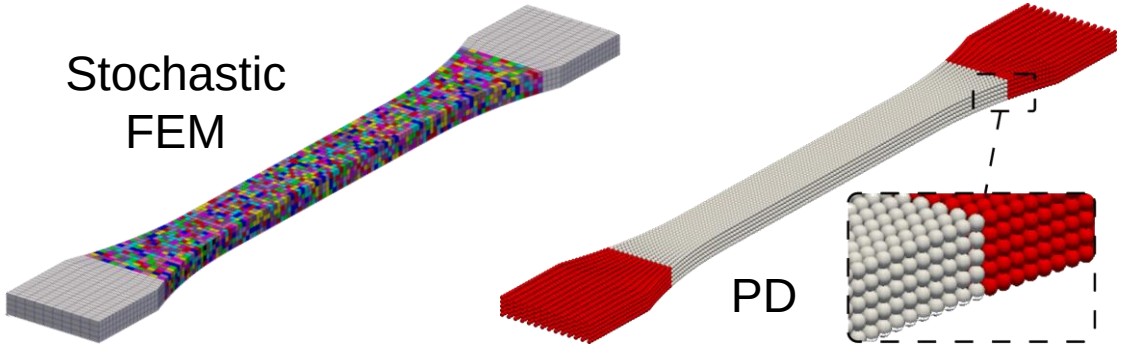
Eco-system
Modeling

- Conversion plugins

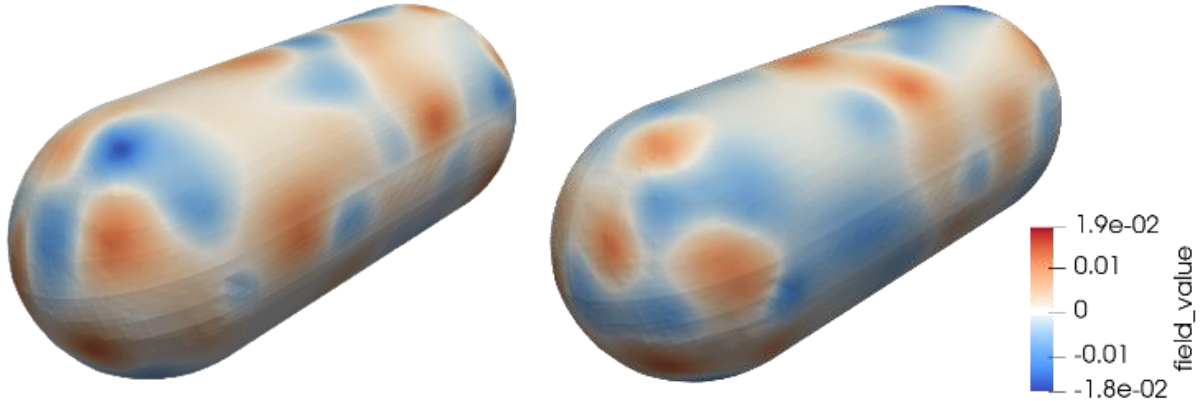




- Idealization & theory-independent model generation

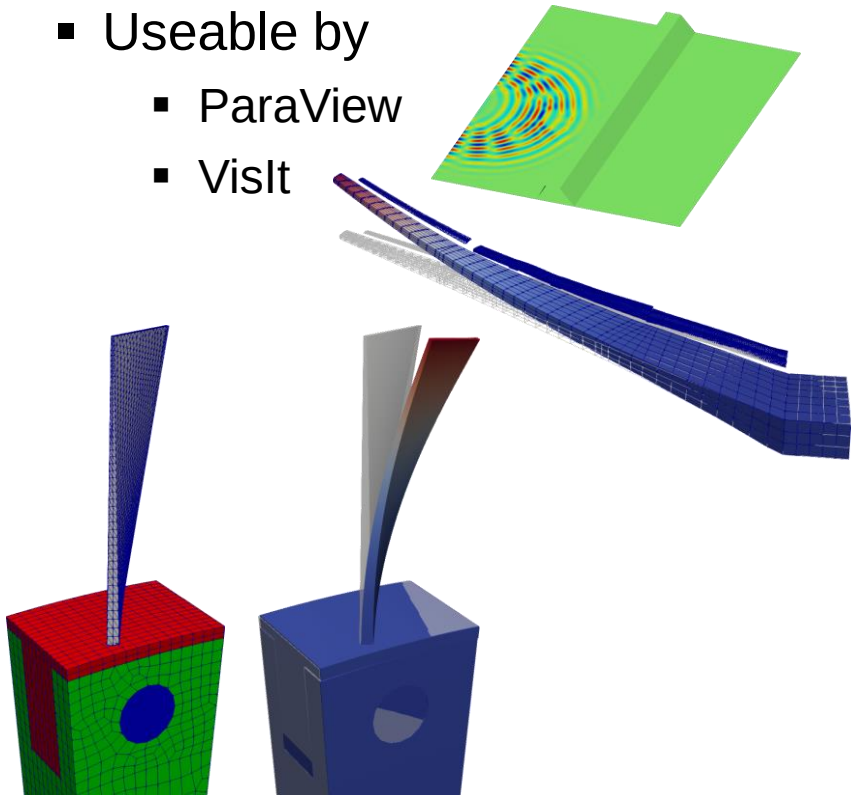


- Probabilistic modeling & manipulation



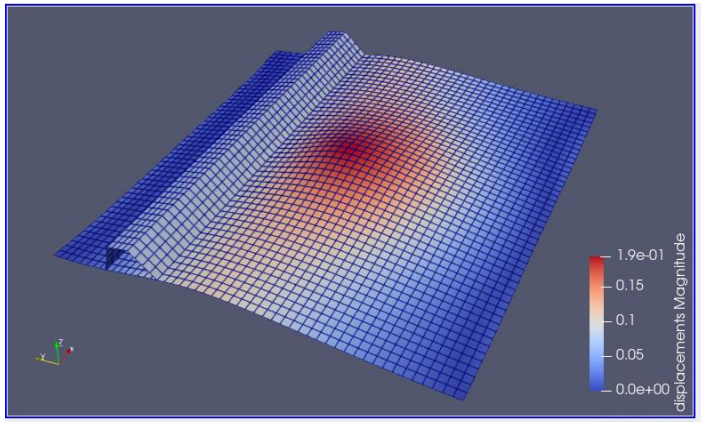
Eco-system
Results & postprocessing

- Visualization: XDMF export
  - Useable by
    - ParaView
    - VisIt



- Postprocessing lib MCODAC

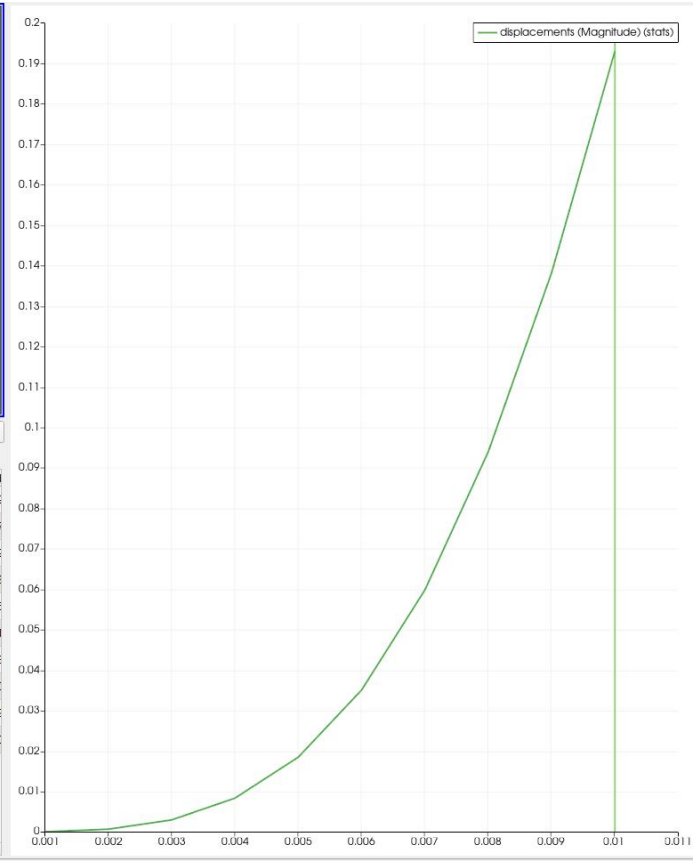
- Automated reporting, e.g. via pvpython

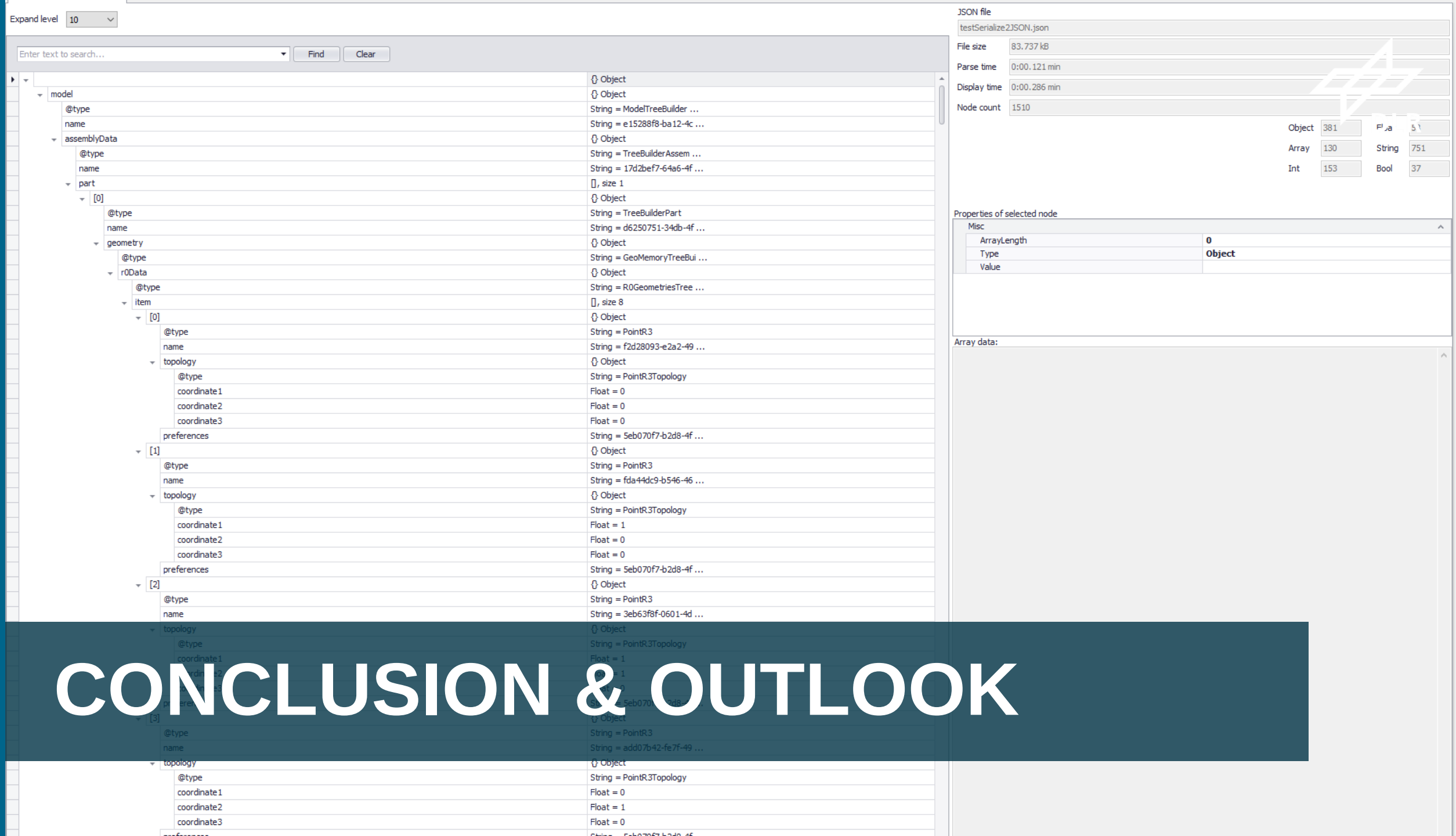


SpreadSheetView2

Showing: PlotSelectionOverTime1 Attribute: Row Data Precision: 6

	Block Name	Row ID	N	Time	avg(X)	avg(Y)	avg(Z)	avg(displacements (0))	avg(displacements (1))	avg(displacements (2))	avg(d
0	stats	0	1	0.001	250	240	0.0185639	8.32518e-08	-1.42928e-06	7.42557e-05	7.4269
1	stats	1	1	0.002	250	239.997	0.177728	5.07389e-07	-1.09788e-05	0.000710913	0.0007
2	stats	2	1	0.003	250	239.992	0.749688	-1.0978e-08	-3.38598e-05	0.00299875	0.0029
3	stats	3	1	0.004	249.998	239.983	2.09672	-8.9602e-06	-7.05869e-05	0.00838689	0.0083
4	stats	4	1	0.005	249.989	239.969	4.62704	-4.39711e-05	-0.00012289	0.0185082	0.0185
5	stats	5	1	0.006	249.969	239.954	8.77877	-0.000125061	-0.000186723	0.0351151	0.0351
6	stats	6	1	0.007	249.937	239.943	14.9661	-0.000254121	-0.000227641	0.0598646	0.0598
7	stats	7	1	0.008	249.899	239.957	23.4999	-0.000405452	-0.000171531	0.0939994	0.0940
8	stats	8	1	0.009	249.865	240.017	34.5641	-0.000539408	6.5739e-05	0.138256	0.1382
9	stats	9	1	0.01	249.847	240.138	48.2696	-0.000611975	0.000551454	0.193079	0.1930

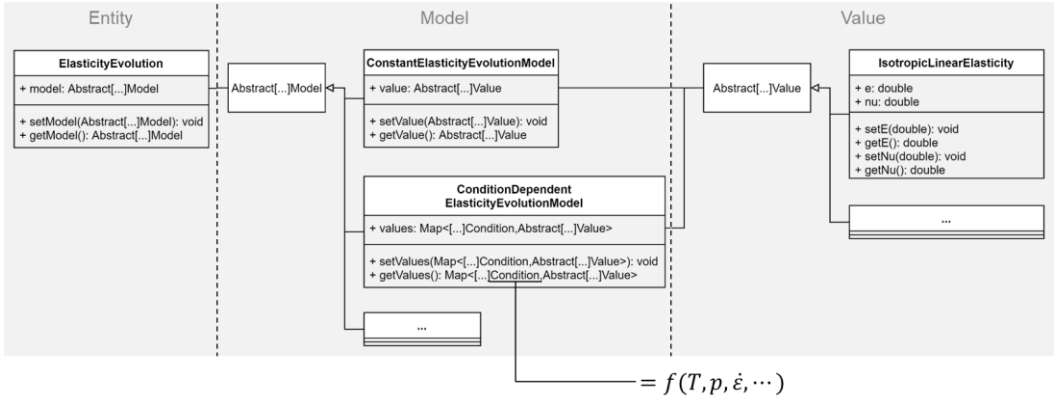


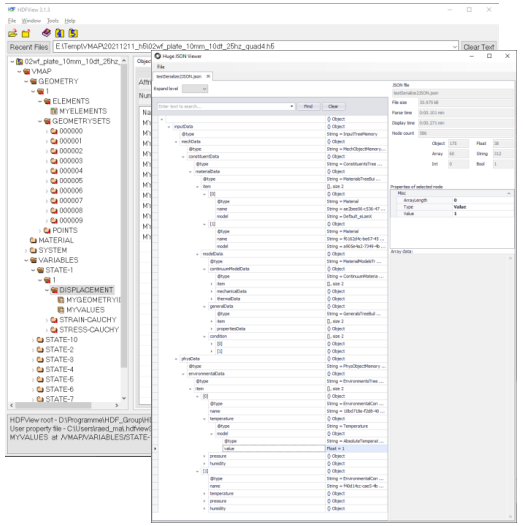


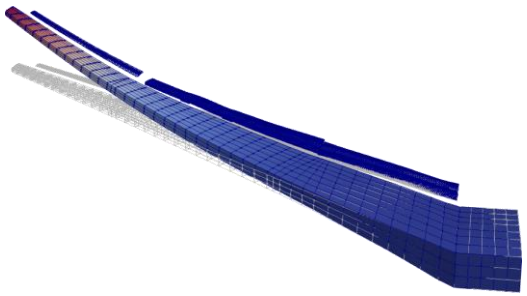
CONCLUSION & OUTLOOK

Conclusion & outlook

- Hierarchy with modular syntax possible
  - Grind: E.g. mech.object.dto project → 2891 classes, interfaces & enums
- Used as input/output for several DLR tools
  - Analytical methods
  - Model generators
  - FEM data handling & postprocessing
- Interaction with commercial & open source solvers demonstrated
- State:
  - By no means feature-complete
  - Hierarchy converging, not yet free of refactoring
  - On our way to open source







https://gitlab.com/jmes_project

THANK YOU FOR YOUR ATTENTION

Contact

Martin Rädel

German Aerospace Center
Institute of Lightweight Systems
Department of Structural Mechanics

Lilienthalplatz 7
38108 Braunschweig

☎ +49 (0)531 295-2048
🖷 +49 (0)531 295-2232
✉ martin.raedel@dlr.de
💻 www.dlr.de/sy

Virtual Product House

Cornelius-Edzard-Straße 15
28199 Bremen

💻 www.dlr.de/vph

European Union
Investing in Bremen's Future
European Regional
Development Fund

FUNDED BY ERDF