ITEA 3

ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands

T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

Deliverable 6.2: Exchange format definition

COMPAS

Compact modelling of high-tech systems for health management and optimization **a**long the **s**upply chain

---

Lead beneficiary:	Infineon Technology AG
WP. no, title	WP 6, Standardization and IP protection
Contributing task(s)	T6.2 Document exchange format
Dissemination level	Public
Delivery date	M34 (M35)
Status	Draft

Leading Authors		
Name	*Beneficiary name*	*Contact email*
Dr. Martin Niessner	Infineon Technologies AG	martin.niessner@infineon.com
Dr. Daming Lou	MSC Software	daming.lou@hexagon.com
Reviewers		
Name	*Beneficiary Name*	*Contact email*
Dr. Michiel van Soestbergen	NXP Semiconductors	michiel.van.soestbergen@nxp.com
Diana Manvelyan	Siemens	diana.manvelyan@siemens.com

Table of contents

Table of contents 2
Executive Summary 4
1. Requirements concluded from WP1 and WP6 5
2. Decision for FMI Version 3.0 as exchange format 6
3. Workflow for thermo-mechanical compact model generation, exchange and re-integration 7
4. Keywords.. 7
5. Example 10

List of abbreviations and acronyms

Abbreviations	
Dx.y	Deliverable x.y of the COMPAS project
FEM	Finite Element Model / Method
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
JEDEC	JEDEC Solid State Technology Association
JEP	JEDEC Publication
ROM	Reduced Order Model
Tx.y	Task x.y of the COMPAS project
WPx	Work package x of the COMPAS project
XML	Extensible Markup Language

Partner acronyms			
Atlas	Atlas Technologies B.V.	MSC	MSC Software
FhG-ENAS	Fraunhofer ENAS	NXP	NXP Semiconductors Netherlands
Eesy	Eesy-Innovation	Reden	Reden B.V.
IFAG	Infineon	SIEM	Siemens Munich
JADE	JADE Hochschule	SISW	Siemens Industry Software
KU Leuven	Katholieke Universiteit Leuven	TUD	Delft University of Technology
MCE	MicroConsult Engineering	Tue	Eindhoven University of Technology

Executive Summary

The prerequisite for a seamless exchange of the thermo-mechanical compact models developed in the COMPAS project, a.k.a. reduced-order models (ROMs), along the supply chain is a standardized exchange format.

The previous deliverables of the COMPAS project D1.1, D1.3 and D6.1 stated that the preservation of geometry information is a must-have for the thermo-mechanical ROMs, because this is necessary for re-integrating the ROMs into 3D environments. Furthermore, the exchange format needs to be able to transport the system matrices of the ROM. Finally, since software packages from different vendors are used along the supply chain, the format needs to be independent of any vendor.

In D1.3 it was initially concluded that none of the existing exchange formats and standards fulfil the requirements of the ROMs developed in the COMPAS project. However, recently the new 3.0 version of the xml-based Functional Mock-up Interface (FMI) was released, which significantly extends the capabilities of this format and covers the requirements of the thermo-mechanical compact models developed in the COMPAS project.

Consequently, in order to avoid defining an entirely new format, an exchange format was developed using FMI 3.0 as a framework and is presented in this document. With the FMI/FMU-based exchange format, the thermo-mechanical compact models are also eligible for referencing in the digital description of electronics parts using the “JEP30: PartModel Guidelines” of the JEDEC standardization organization.

This document first revisits the requirements of the thermo-mechanical ROMs developed in the COMPAS project, then briefly discusses the recent extensions of the FMI standard available with version 3.0, outlines the overall workflow of model generation, exchange, and re-integration, documents the keywords used with the exchange format and concludes by giving an example.

1. Requirements concluded from WP1 and WP6

Especially the deliverables D1.1 and D1.3 of WP1 and D6.1 from WP6 summarize the requirements regarding the thermo-mechanical compact models developed in COMPAS project as well as for the later exchange of compact models. Two essential requirements which significantly impact the exchange format are revisited here:

- **Preservation of geometry information:** The partners of the COMPAS project which are active in the semiconductor industry intend to generate thermo-mechanical compact models of their packaged products, which can then be passed along the supply chain. The users of the compact models may then place those “component-level” (or “package-level”) compact models on different locations on their PCB, e.g. at the centre of the PCB, at the perimeter of the PCB, in the vicinity of other components or next to connectors. The aforementioned users characteristically perform their simulations not on a system-level which is abstract from the actual geometry, but 3D FEM simulations are used, which take the 3D geometry information into account. In order to enable 3D simulations with compact models of packages, they need to be coupled to the 3D FEM model of the PCB (see Figure 1). Consequently, an abstract compact model without any geometry information is not sufficient, but the geometry, i.e. dimensional information of component needs to be preserved in and transferred along with the compact model.

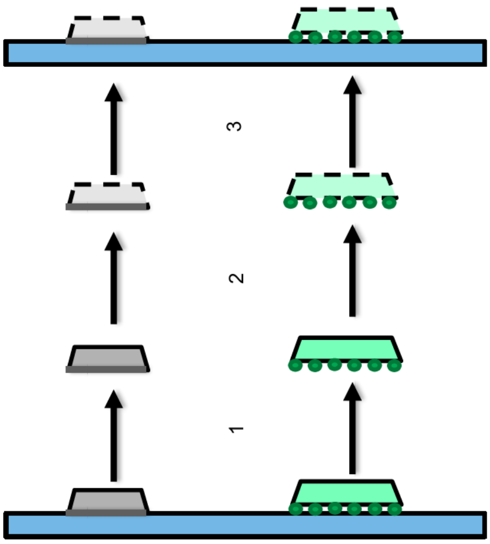

Figure 1: Diagram taken from D1.1 illustrating the decoupling of packages from the PCB (1), reducing them (2) and coupling them back (3)

- **Interchangeability:** 3D FEM simulation tools from different vendors are used along the supply chain. For example, the partners in the COMPAS project use tools from 4 different software vendors. Consequently, interchangeability and the preservation of a high quality of data along the supply chain is essential. Being able to extend or build on existing formats and standards which are already known to different vendors can be considered as advantageous.

In summary, the following information needs to be conveyed by the exchange format:

1. Coordinates (geometry information of the nodes of the compact model for integration into the FEM model)
2. Connectivity (nodes number which are considered as the interface for coupling with other parts of the structure)
3. In-Output (nodes number and the physical units which are considered as the input and the output)
4. System matrix (system matrices of the compact model)

2. Decision for FMI Version 3.0 as exchange format

In task T1.3 of WP1, several software exchange formats and standards were reviewed, from which D1.3 documents in more detail the findings w.r.t. VMAP, Functional Mock-up Interface (FMI) Version 2.0 and Delphi. D1.3 initially concluded that none of the existing formats and standards would suffice the requirements of the compact models developed in the COMPAS project. Especially the preservation of the geometry information as well as the potentially high complexity of the presumably non-linear thermo-mechanical reduced-order models are two differentiators which are not covered by the existing formats and standards.

However, after deliverable D1.3 was finalized, the xml-based Functional Mock-up Interface (FMI) Version 3.0 was released, which significantly extends the capabilities of this format. Especially the "extension of variables to arrays" (see https://fmi-standard.org/docs/3.0/), allows now to convey information such as matrices of complex compact models and geometry information. The FMI organization itself argues that "with FMI 3.0, virtual electronic control units (vECUs) can be exported as FMUs in a more natural way" (see https://fmi-standard.org/docs/3.0/).

Since the recently released update of FMI fulfils the requirements for the thermo-mechanical compact models developed in the COMPAS project, the COMPAS consortium decided to build on the FMI 3.0 format instead of defining an entirely new format. By using the FMI format, the expectation is that it will be also easier to load the compact models into the simulation tools because already existing FMI/FMU interfaces can be re-used and, in case necessary, extended.

For the electronic components investigated here, an additional target is to include the resulting thermo-mechanical compact FMI/FMU models in the digital description of the part using the "JEP30: PartModel Guidelines" (https://www.jedec.org/category/technology-focus-area/jep30) of the JEDEC standardization organization, especially the "JEP30-T100A PartModel Thermal Guidelines for Electronic-Device Packages" and "JEP30-T101 XML Requirements". These guidelines have recently been extended in March 2023, so that additional external models, e.g. given in FMI/FMU format, can be referenced by links which can be included in the digital description.

3. Workflow for thermo-mechanical compact model generation, exchange and re-integration

The overall workflow for the generation, exchange (via FMI 3.0) and re-integration of thermo-mechanical compact models is depicted in Figure 2. Superelements are speciality elements and were identified in WP4 of the COMPAS project as means suitable for bringing back the thermo-mechanical compact models and reduced-order models (ROM) into the 3D FEM full order model (FOM) simulation environment. An overview of different methods for the extraction of ROMs from FOMs can be found in the public deliverable D2.6 of WP2 of the COMPAS project.

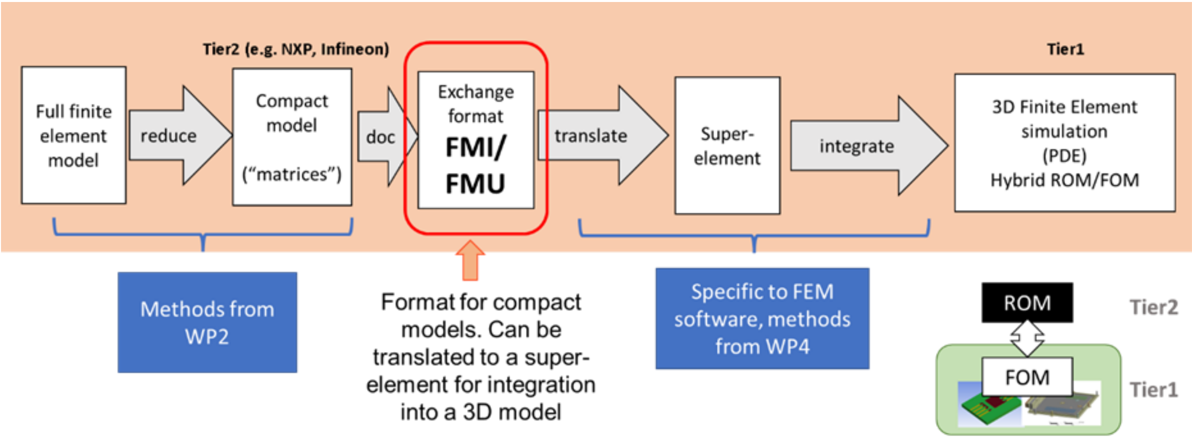

Figure 2: Overall workflow for the generation, exchange and re-integration of thermo-mechanical compact models

4. Keywords

Table 1 and Table 2 show the names, format and content description of the variables defined in the exchange format. The parameter variables listed in the tables are used for exchanging information, and the input variables need to be set before obtaining the output variables.
Detailed information about the FMI 3.0 standard is available at https://fmi-standard.org/docs/3.0/.

Variable type	Name	Format	Description
Parameter	phy_stru	Integer: 0 → no 1 → yes	Flag the ROM structural model; other types of physics, such as thermal with the keyword "phy_thm", can be supported in the future.
Parameter	time_dep	Integer: 0 → static 10 → dynamic	Flag to indicate whether inertia loads are considered.
Parameter	num_interf	Integer (> 0)	Number of interface nodes.
Parameter	boundary_size	Integer (> 0)	Number of boundary DOFs (number of interface nodes times the number of DOFs per node).
Parameter	para_xxx	Float	xxx parameter(s).
Parameter	sym_stiff	Integer: 0 → nonsymmetric 1 → symmetric	Flag to indicate if the stiffness matrix is symmetric or non-symmetric. • For non-symmetric option, the str_stiff returns a full stiffness matrix • For symmetric option, str_stif returns the lower triangle part of the stiffness matrix
Parameter	sym_mass	Integer: 0 → nonsymmetric 1 → symmetric 2 → lumped mass	Flag to indicate if the mass matrix is symmetric, non-symmetric or lumped (diagonal matrix). • For non-symmetric option, the str_stif returns a full stiffness matrix • For symmetric option, str_stif returns the lower triangle part of the stiffness matrix
Parameter	analysis_dim	Integer: 0 → 3-D 1 → planar 2 → axisymmetric	Flag indicating the model dimension.
Parameter	set_geoinfo	Float	Spatial location of the interface nodes in the global Cartesian XYZ space.
Parameter	temp_ref	Float	Reference(initial) temperature of the ROM.

Parameter	unit_system	Integer
		1 → default unit system
			The default unit system: SI + millimeter + Newton
			• Length → Millimeter (mm) // Time → Second (s)
			• Mass → 1000 kg (10^3 kg or t) // Force → Newton (N)
			• Temperature → kelvin (K) // Electric current → Amper (A)//
			• Amount of substance → mol (mol) //Luminous intensity → Candela (CD)

Table 1: Keywords and names for the variable type "parameter" used with the exchange format

Variable type	Name	Format	Description
Input	`temp_state`	Float	State temperature as for thermal loads (load vectors).
Input	`pmor_para_xxx`	Float	Parameter is reserved for deploying the parameterized model order reduction techniques (pMOR).
Output	`str_stif`	Float	Stiffness matrix: • full matrix if sym_stif=0 lower triangle part if sym_stif=1
Output	`str_mass`	Float	Mass matrix: • full matrix if sym_mass=0 • lower triangle part if sym_mass=1 Only the diagonal part of the matrix if sym_mass=2
Output	`thm_load`	Float	Thermal load (thermal load vector) at the boundary nodes.

Table 2: Keywords and names for the variable types "input" and "output" used with the exchange format

When the thermo-mechanical reduced order model (ROM) is being defined in the exchange format using the keywords mentioned in the tables, the following points should be noted:

- The `set_geoinfo` keyword returns a set of geometry location of the interface nodes, with their coordinates ordered as X, Y, and Z (where Z equals 0.0 for planar and axisymmetric models); so, the dimension of the set is 3*`num_interf`.
- The `thm_load` keyword returns an array of the load vector which corresponds to the interface nodes, and the dimension of this array is equal to the `boundary_size`.
- The `str_stif` keyword returns the stiffness matrix in a flatten format (1D array)
    - If `sym_stif` = 0, the matrix has dimension of (1*boundary_size^2)
    - If `sym_stif` = 1, the matrix has dimension of (1*0.5*(boundary_size+1)*boundary_size)
- The `str_mass` keyword returns the mass matrix in a flatten format (1D array)
    - If `sym_mass` = 0, the matrix has dimension of (1*boundary_size^2)
        - e,g., $\text{Mass} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \longrightarrow [A_{11}, A_{12}, A_{21}, A_{22}]$
    - If `sym_mass` = 1, the matrix has dimension of (1*0.5*(boundary_size+1)*boundary_size)
        - e,g., $\text{Mass} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} \longrightarrow [A_{11}, A_{12}^T, A_{22}]$
    - If `sym_mass` = 2, the matrix has dimension of (1*boundary_size)
        - e,g., $\text{Mass} = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \longrightarrow [A_{11}, A_{22}]$
- The ordering of interface nodes in `set_geoinfo`, `thm_load`, `str_stif` and `str_mass` must be identical.
- The `para_xxx` denotes the xxx variable, which can either be a single floating-point value or a range of floating-point values, with minimum and maximal values specified in the XML file.
- The `temp_state` represent the state temperature, which can either be a single floating-point value or a range of floating-point values, with minimum and maximal values specified in the XML file.
- The `unit_system` denotes the unit system defined in the FMU. The details are given in the description.

5. Example

In this section, we demonstrate how a ROM can be converted into a FMU by using the exchange format proposed in the previous chapter. We use the example of the model of the PG-TDSON-8 package studied across the COMPAS project shown in Figure 3.

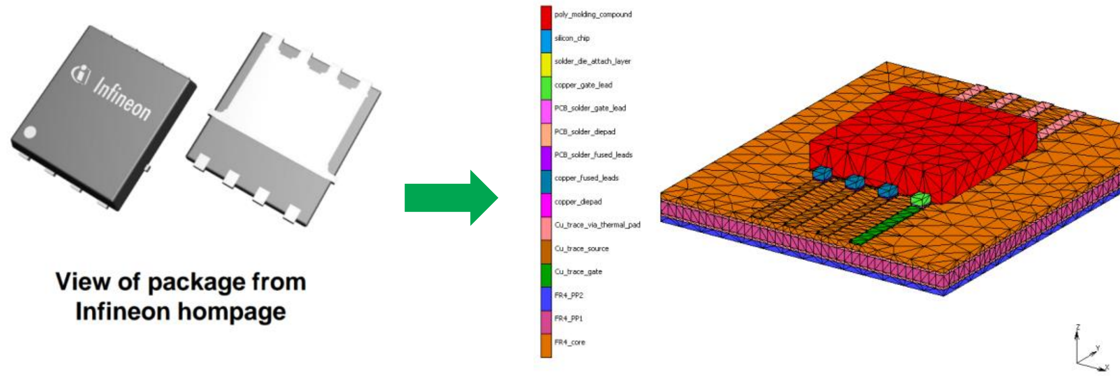

Figure 3: View of the hardware component (left) and of the modelled component on a PCB (right)

As shown in more detail in the WP5 deliverables of the COMPAS project, the package component FOM shown in Figure 4 will be reduced to a ROM. Then, a FMU will be created based on the ROM and assembled with the PCB board. Finally, simulations of the full setup will be performed using a hybrid ROM-FOM model, employing the FMU.

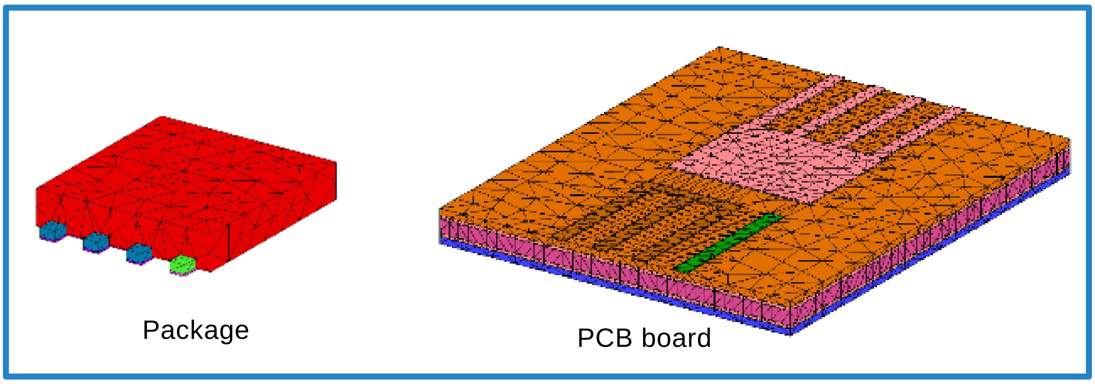

Figure 4: Overview of the package and the PCB board.

It should be noted that the materials in the package FOM have temperature-dependent properties. For example, Young's modulus and the thermal expansion coefficient are functions of temperature for these materials. Consequently, the structural behavior of the package will vary under different temperature conditions due to these temperature-dependent material properties.

Model order reduction techniques (see public deliverable D2.6 of WP2 of the COMPAS project) were applied to the package model for generating a corresponding reduced order model (ROM). In this ROM, temperature is an input parameter. The outputs of the ROM are the load vectors (representing reaction forces induced by temperature) and the stiffness matrix (see Figure 5).

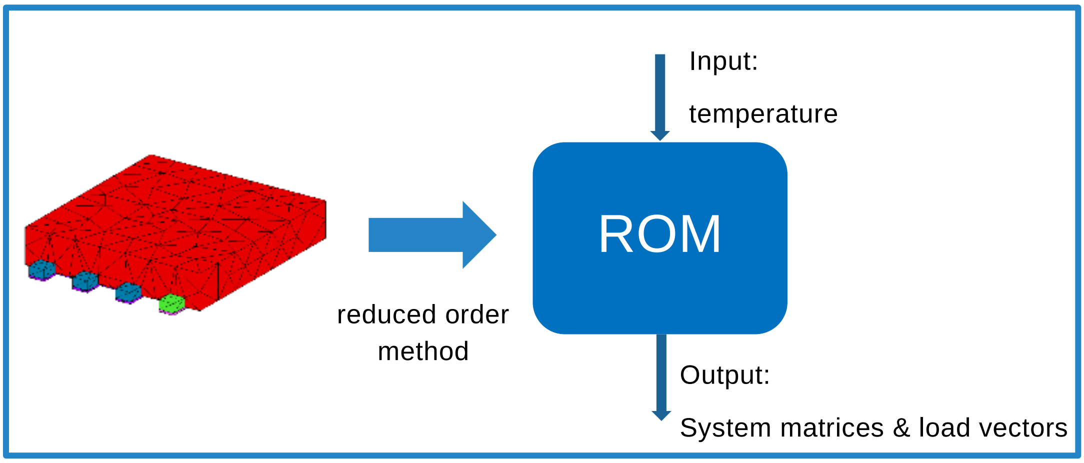

Figure 5: Process from FOM to ROM as FMU

Using the specifications provided in the tables of the previous chapter, a functional mock-up unit (FMU) can be generated from the reduced order model.
The XML file within the FMU begins with the XML version and encoding line. Under the <fmiModelDescription> section, the top-level details are provided as follows (see also except shown in Figure 6):

- The FMI version is specified in the required fmiVersion tag per the FMI standard.
- The model name of the FMU.
- The compatibility token used to check compatibility with the FMU implementation.
- The tools used to generate the FMU.
- The date and time when the FMU was created.

```
<?xml version="1.0" encoding="UTF-8"?>
<fmiModelDescription
    fmiVersion="3.0"
    modelName="package_COMPAS"
    instantiationToken="{8a972d5a-fb29-4bcd-91c5-0aad1b66d3be}"
    generationTool="MSC Software"
    generationDateAndTime="2023-11-17T14:47:41" >
<ModelExchange modelIdentifier="package_COMPAS" needsExecutionTool='true' canGetAndSetFMUState='true'
    canSerializeFMUState='true' providesDirectionalDerivatives='true' providesAdjointDerivatives='false'/>
<LogCategories>
<ModelVariables>
<ModelStructure>
</fmiModelDescription>
```

Figure 6: Sample excerpt of top level line of the XML file defining the FMU

The parameters, the inputs and the outputs are detailed in the <ModelVariables> section (see also excerpt shown in Figure 7):

- Parameters include the `boundary_size`, `time_dep`, `analysis_type`, `sym_stiff`, `num_interf`, `time`, `unit_system`, `temp_ref` and the `set_geoinfo`,
- Inputs contain the `temp_state`,
- Outputs contain the `thm_load` and the `str_stif`,


```xml
<LogCategories>
<ModelVariables>
    <UInt64 name="boundary_size" valueReference="101" causality="parameter" variability="fixed"
        start="426" description="Number of Boundary Grid dofs" />
    <UInt64 name="time_dep" valueReference="102" causality="parameter" variability="fixed"
        start="0" description="" />
    <UInt64 name="analysis_type" valueReference="103" causality="parameter" variability="fixed"
        start="0" description="Analysis Type" />
    <UInt64 name="sym_stiff" valueReference="104" causality="parameter" variability="fixed"
        start="1" description="stiffness matrix symmetricity" />
    <UInt64 name="num_interf" valueReference="105" causality="parameter" variability="fixed"
        start="142" description="number of interface nodes" />
    <UInt64 name="analysis_dim" valueReference="106" causality="parameter" variability="fixed"
        start="0" description="model dimension" />
    <Float64 name="time" valueReference="0" causality="independent" variability="continuous" description="Simulation time"/>
    <Float64 name="set_geoinfo" valueReference="107" causality="independent" variability="continuous"
    description="Spatial location of the interface nodes in the global Cartesian XYZ space."/>
    <UInt64 name="unit_system" valueReference="108" causality="parameter" variability="fixed"
        start="1" description="unit system" />
    <Float64 name="temp_ref" valueReference="109" causality="independent" variability="continuous"
    description="Reference(initial) temperature of the ROM."/>
    <Float64 name="temp_state" valueReference="1" description="temp_state" causality="parameter"
        variability="tunable" start="-40" min="-40" max="125"></Float64>
    <Float64 name="thm_load" valueReference="7" initial="calculated" causality="output"
        description="output of thermal loads" variability="continuous">
        <Dimension start="426" />
    </Float64>
    <Float64 name="str_stif" valueReference="8" initial="calculated" causality="output"
        description="output of stiffness matrix" variability="continuous">
        <Dimension start="90951" />
    </Float64>
</ModelVariables>
<ModelStructure>
    <Output valueReference="7" />
    <Output valueReference="8" />
    <InitialUnknown valueReference="7" dependencies="1" dependenciesKind="dependent"/>
    <InitialUnknown valueReference="8" dependencies="1" dependenciesKind="dependent"/>
</ModelStructure>
</fmiModelDescription>
```


Figure 7: Sample excerpt of the model variables and the structure of the FMU

This FMU can then be imported into a 3D simulation environment and connected together with the FOM of the PCB to perform hybrid ROM-FOM simulations of the assembled setup. In Figure 8, the full order model (FOM) and the assembled model using the FMU are both evaluated at a temperature of 40°C. The results show close agreement, indicating that the FMU accurately represents the behavior of the full order package model.

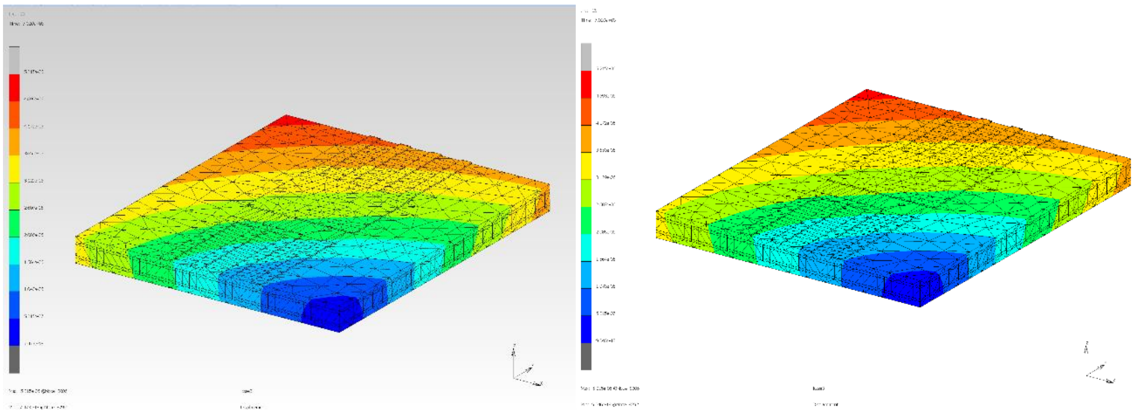

Figure 8: Result comparison of the PCB displacement at 40°C of the FOM only simulation (left) and the hybrid ROM-FOM simulation (right) using a ROM of the component imported to the 3D simulation environment as an FMU