HDF5 Composite CAE Specification

Authors:	Johan Grape	johan.grape@vistagy.com
	Jim Koenig	jim.koenig@vistagy.com
	Oliver König	oliver.koenig@ansys.com
	Marc Wintermantel	marc.wintermantel@ansys.com
	Uche Mennel	uche.mennel@ansys.com

Table of Contents

Change Log 2

HDF5 Object Tree 4

Node Types in HDF5 Tree 5

Specification of Node Names 5

Specification of Node Attributes and Data 5

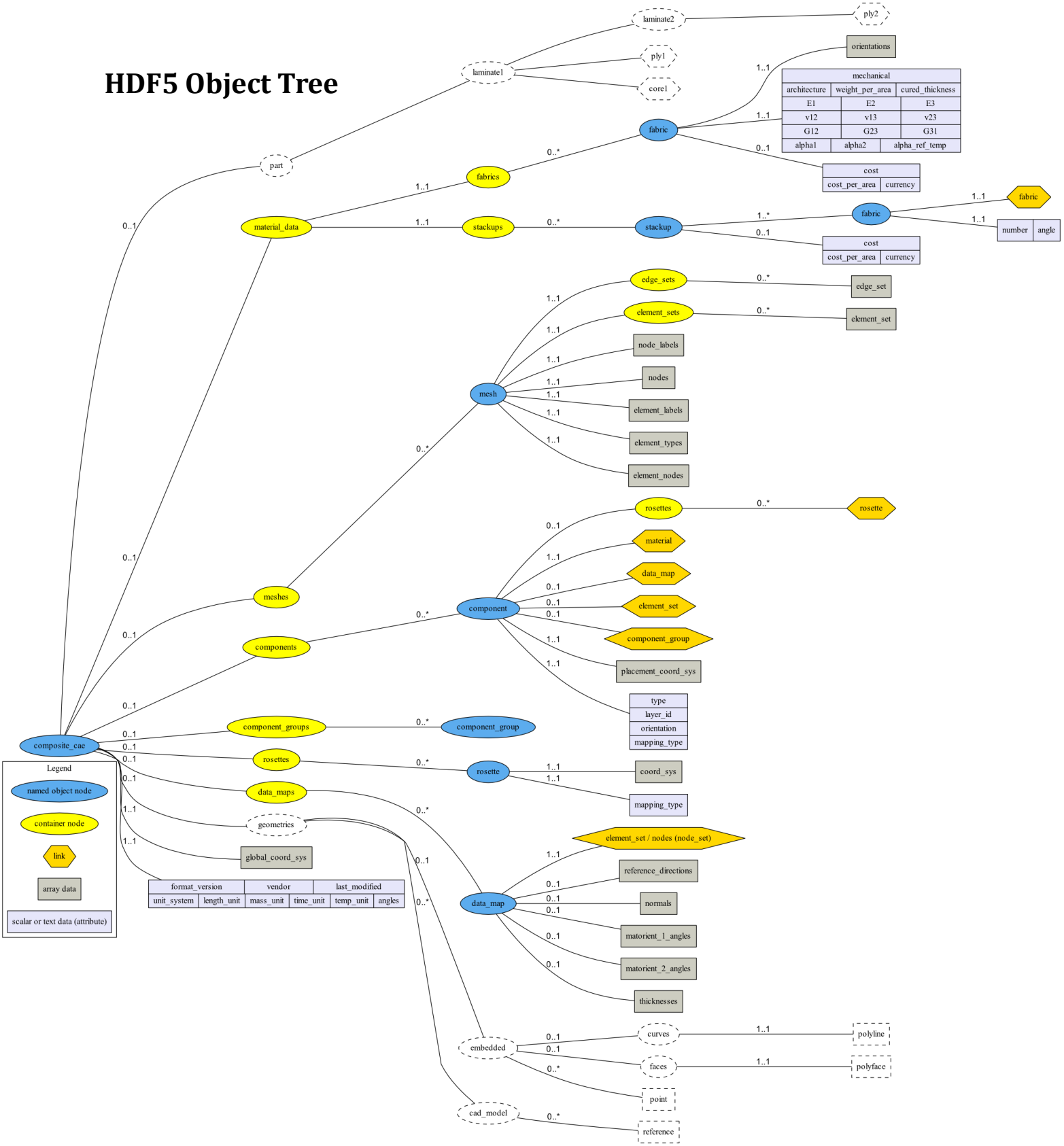
Incomplete Items for Further Discussions in 2011 10

Change Log

May 5, 2010	ACP – FiberSIM Interface Format Specification Rev 01	O.Koenig, M.Wintermantel
June 5, 2010	HDF5 Composite CAE Data File Specification Proposal version 1.1	J. Grape, J. Koenig
Aug 10, 2010	HDF5 Composite CAE Data File Specification Release Candidate 1 for stable version 1.0 Latest modifications from EVEN: • Modified stackup definition sligthly to take into account that links can not have attributes. • Proposing element type ID's to be used within mesh/elements array. Latest modifications taken from Vistagy's feedback of Aug 9 2010: • Added composite_cae/temp_unit attribute • Added the proposed thickness map data to the specification. This extension also leads to a name change of the “orientation_map” to the more general “data_map”. For consistency we also propose to change 'material_thickness' to 'thicknesses'.	O.Koenig, U.Mennel, M.Wintermantel
Aug 11, 2010	Minor updates in “Open Questions”, added Section “Specification of Node Names”	O.Koenig, U.Mennel
Aug 30, 2010	Upgrade specification to match all issues recently discussed with Vistagy via emails.	O.Koenig
Mar 17, 2011	Cleanup the document a little	O.Koenig
Oct 17 2014	Adding component_groups and component_group nodes	O.Koenig
June 28 2019	Adding some additional information	R. Roos

January 24, 2020	Corrected name for matorient_angles: “mat_orient1_angles” to “matorient_1_angles” and “mat_orient2_angles” to “matorient_2_angles”. Defined which fields in data_map are optional and specify what happens if optional nodes are missing. Clarified use of placement_coord_sys	J. von Rickenbach
September 15, 2020	Mentioned that global_coord_sys is currently not supported. Correct what happens when matorient_1_angles are not present	J. von Rickenbach

HDF5 Object Tree



Node Types in HDF5 Tree

Container Node
Group node which contains a list of object nodes of the same type.
Examples: fabrics, meshes, rosettes

Named Object Node
Node to specify an object. To avoid naming restrictions of HDF5 nodes, the external object name should be stored in the title attribute of the node.
The associated data of an object is stored in attributes of the object node or attached data nodes.
Examples: fabric, mesh, rosette

Link Node
Dependencies between the different nodes in the tree are stored using link nodes.

Data Node
Data nodes can either contain array data or scalar data in attributes.

Specification of Node Names

To provide smooth interoperability with most high-level HDF5 interfaces (e.g. PyTables), the HDF node names must match the (perl regexp) pattern '^[a-zA-Z_][a-zA-Z0-9_]*$'. The names on the Nodes must be unique within each group.

Specification of Node Attributes and Data

composite_cae
Attributes:

format_version	string	1.0
last_modified	datestring	“2010-08-23 13:31:13Z”
vendor	string	"VISTAGY, Inc."
unit_system	string	“SI”,”Imperial”
length_unit	string	"mm","in"
mass_unit	string	"g","lbf"
time_unit	string	"s"
temp_unit	string	“C”, “K”, “F”
angles	string	“deg”, “rad”

composite_cae->global_coord_sys
(4 by 3 array of double)

[origin x]	[origin y]	[origin z]
[dir1 x]	[dir1 y]	[dir1 z]
[dir2 x]	[dir2 y]	[dir2 z]
[dir3 x]	[dir3 y]	[dir3 z]

Defines the origin and directions of the origin for the entire model.
Currently not supported by Ansys

material_data->fabrics
fabric(named)

material_data->fabrics->fabric(named)->orientations
(array of double)
[0, 45,-45]

Notes:
Provides all directions of fibers in the fabric.

material_data->fabrics->fabric(named)->mechanical
Attributes:

architecture	string	uni,woven
weight_per_area	double	
cured_thickness	double	
E1	double	
E2	double	
E3	double	
G12	double	
G23	double	
G31	double	
v12	double	
v13	double	
v23	double	
alpha1	double	
alpha2	double	
alpha_ref_temp	double	

Note: To allow the specification of draping dependent material properties of woven fabrics in the future, we intend to introduce draping-shear-angle dependent look-up tables within this node. This would mean that optionally every material property can be defined in an array which defines the values of this property for given shear angles. This way we would once again be independent of possibly different implementations of material models by different vendors, eventually the look-up tables can be filled directly from material tests.

material_data->fabrics->fabric(named)->cost
Attributes:

currency	string	EUR,USD,etc
cost_per_area	double	

material_data->stackups->stackup(named)->cost
Attributes:

currency	string	EUR,USD,etc
cost_per_area	double	

material_data->stackups->stackup(named)->fabric

Attributes:

number	unsigned	Determines order of fabrics within stackup.
angle	double	Angle of fabric within stackup.

meshes->mesh(named)->node_labels
(n by 1 array of doubles)
[unique node id]

meshes->mesh(named)->nodes
(n by 3 array of doubles)
[node x] [node y] [node z]

meshes->mesh(named)->element_labels
(n by 1 array of doubles)
[unique element id]

meshes->mesh(named)->element_types
(n by 1 array of doubles)
[element type]
Currently the following element types are defined:

- 3: 3 node shell
- 4: 4 node shell
- 6: 6 node shell
- 8: 8 node shell

The list of element types might be expanded in the future, e.g. using 108 and 120 for 8 and 20 node bricks.

meshes->mesh(named)->element_nodes
(n by m VLArray of int)
[node 1] [node 2] … [node m]
Defines nodes of elements with node INDICES pointing directly to entries in the nodes/node_labels arrays.

NOTE: here a VLArray data set is used since the mesh can contain element of different types (refer to meshes->mesh(named)->element_types)

meshes->mesh(named)->edge_sets->edge_set(named)
(array of unsigned)
Array with node INDICES referencing the nodes in this set.

meshes->mesh(named)->element_sets->element_set(named)
(array of unsigned)
Array with element INDICES referencing the elements in this set.

components->component(named)
Attributes:

type	string	ply,core,etc.
layer_id	integer	aka step value. Global identifier increases through thickness.

orientation double nominal angle orientation of ply/core ribbon direction relative to direction vector
mapping_type string

NOTE: The mapping_type is optional and will be detailed later. Since ACP allows to use multiple rosettes to define the reference direction within a ply, different mapping types (interpolation algorithms) are defined, such as "minimum distance", "minimum angle", etc.

components->component(named)->placement_coord_sys
(4 by 3 array of double)

[origin x]	[origin y]	[origin z]
[dir1 x]	[dir1 y]	[dir1 z]
[dir2 x]	[dir2 y]	[dir2 z]
[normal x]	[normal y]	[normal z]

The placement_coord_sys is used for two things:

- The normal of the placement coordinate system is used to define the direction of the layup.
- The origin and dir1 are used to define the origin and the direction of the draping calculation

components->component(named)->material
HRef
Defines the ply material and refers to an object in "material_data->fabrics" or "material_data->stackups".

components->component(named)->element_set
HRef
Defines a link to an element set of a mesh and specifies the ply extent.

components->component(named)->data_map
HRef
Defines a link to the data map of this component.
The following restrictions and rules apply:

- If the data_map contains a thickness, a "material" of type "material_data->stackups" is not supported
- If "data_map->thickness" is not defined the thickness is computed from "cured_thickness" if material is a "fabric" or the sum of the "cured_thickness" for all the "fabric" if material is a "material_data->stackup"
- If matorient_1_angles is not present it set equal to "component->orientation":
- If matorient_2_angles is not present it is computed as:
  - Matorient_2_angles = matorient_1_angles + 90°

rosettes->rosette(named)
attributes:
mapping_type string

NOTE: The possible values of the rosette mapping_type are not yet specified.
FiberSIM defines "translational", "standard", and "rotational" mapping types.
ACP defines “parallel”, “radial”, “cylindrical” and “spherical” as standard rosette types.
The exact definition of this attribute has been postponed, because this is optional information to exchange, which it is not needed to map the orientation field of the plies correctly.

rosettes->rosette(named)->coord_sys
(4 by 3 array of double)
[origin x] [origin y] [origin z]
[dir1 x] [dir1 y] [dir1 z]
[dir2 x] [dir2 y] [dir2 z]
[normal x] [normal y] [normal z]

data_maps->data_map(named)->element_set/node/node_set
Defines a link to an element set of a mesh or a link to an edge_set of a mesh. This defines the positions where the orientations, thicknesses etc are given.

data_maps->data_map(named)->reference_directions
(n by 3 array of double)
Defines the reference directions absolutely with direction vectors.

data_maps->data_map(named)->normal (*Optional)*
(n by 3 array of double)
Defines the normal directions absolutely with direction vectors. If present they can be used to project the reference direction onto the plane defined by the normal.

data_maps->data_map(named)->matorient_1_angles (Optional)
(n by 1 array of double)
Defines the material orientation 1 angles with respect to the reference direction and the given normal.

data_maps->data_map(named)->matorient_2_angles (Optional)
(n by 1 array of double)
Defines the material orientation 2 angles with respect to the reference direction and the given normal.

data_maps->data_map(named)->thicknesses (*Optional)*
(n by 1 array of double)
Defines position dependent thickness values.

component_groups->component_group(named)
Component groups can be defined to store grouping information of components.

Incomplete Items for Further Discussions in 2011

Component boundaries, such as ply boundaries, will be assumed to derive from the element set associated with the component. Handling components that do not conform to element edges is deferred to a later version.

Geometry (embedded and reference to external) is deferred until later revision.

geometries

geometries->embedded

geometries->embedded->point(named) (array of 3 double) point coordinates
[point x]
[point y]
[point z]

geometries->embedded->curves

polyline(named) (n by 3 array of double) polyline xyz
[vertex x] [vertex y] [vertex z]

geometries->embedded->faces->polyface(named)

geometries->embedded->faces->polyface(named)->vertices
(x by 3 array of double)
[vertex x] [vertex y] [vertex z]
[...]

geometries->embedded->faces->polyface(named)->connectivity
(n by 3 array of int) triangle vertex connectivity
[vertex index 1] [vertex index 2] [vertex index 3]
[...]

geometries->cad_model
attributes:
type string catiav5,proe,iges,step_ap203,..
file_name string name of model
file_path string path (relative, absolute) to cad model

geometries->cad_model->reference(named)
string
attributes
type string curve,edge,etc.

part hierarchy
notes: part hierarchy to be discussed.

Unicode support

Since version 1.8.0 of HDF5, strings can be stored UTF-8 encoded. This could be used to support all kinds of object names generated by various modeling tools. It is therefore necessary to keep the (non conforming) object names as separate unicode attributes.