# Quels besoins en topologie pour la représentation des objets au cours du processus de développement de produits ?

***Jean-Claude Léon***, *Leila de Floriani, Franck Hetroy*

# Presentation outline

- Context
  - Product Development Process (PDP)
  - Multiple representations of objects
- Shape diversity in a PDP
  - Categories of object shapes
  - Information attached to a shape boundary
- Modeller capabilities
  - Are volume models only volumes ?
  - Modelling surfaces
- Some topology issues to improve a PDP

# Presentation outline

- Shapes in 3D Euclidean space
  - Contribution to a classification of non-manifold singularities from transformations of 2-manifolds
    - Context
    - Definitions
    - Classification based on 1-cycles
- Shape boundary decomposition
  - Contribution to the description of non-manifold decompositions,
- Conclusion – Open issues

# Context - PDP

- Products in the field of mechanical engineering involving 3D digital models
- Product Development Process (PDP)
  - A connected set of steps where:
    - a product and its components are shaped,
    - behaviours of product components are simulated:
      - Structural behaviour in mechanics,
      - Computational fluid dynamics,
      - Thermal exchanges,
      - Electromagnetics,
      - …,
    - design solutions are assessed,
    - Manufacturing processes are simulated:
      - Machining,
      - Assembly,
      - Injection molding,
      - …

# Context - PDP

- These steps are performed by groups of people in a company:
  - The engineering office,
  - Process planning department,
  - Departments named after a subset of the product,
  - …
- A PDP contains two complementary aspects, at least:
  - digital model processing,
  - Human based decision making and information processing

# Context – Multiple representations

- The various PDP steps can focus differently on the same component or sub-system:
  - different models of the same component / sub-system are generated,
  - Simulation objectives (time, accuracy, cost) are guiding the model definition,
  - Often, different models of component / sub-system mean different shapes of the same component / sub-system, i.e. geometry and topology differ from one shape to another but they all stand for the same component / sub-system,

- Having different shapes of the same component / sub-system fits into the concept of their multiple representations

# Context – Multiple representations

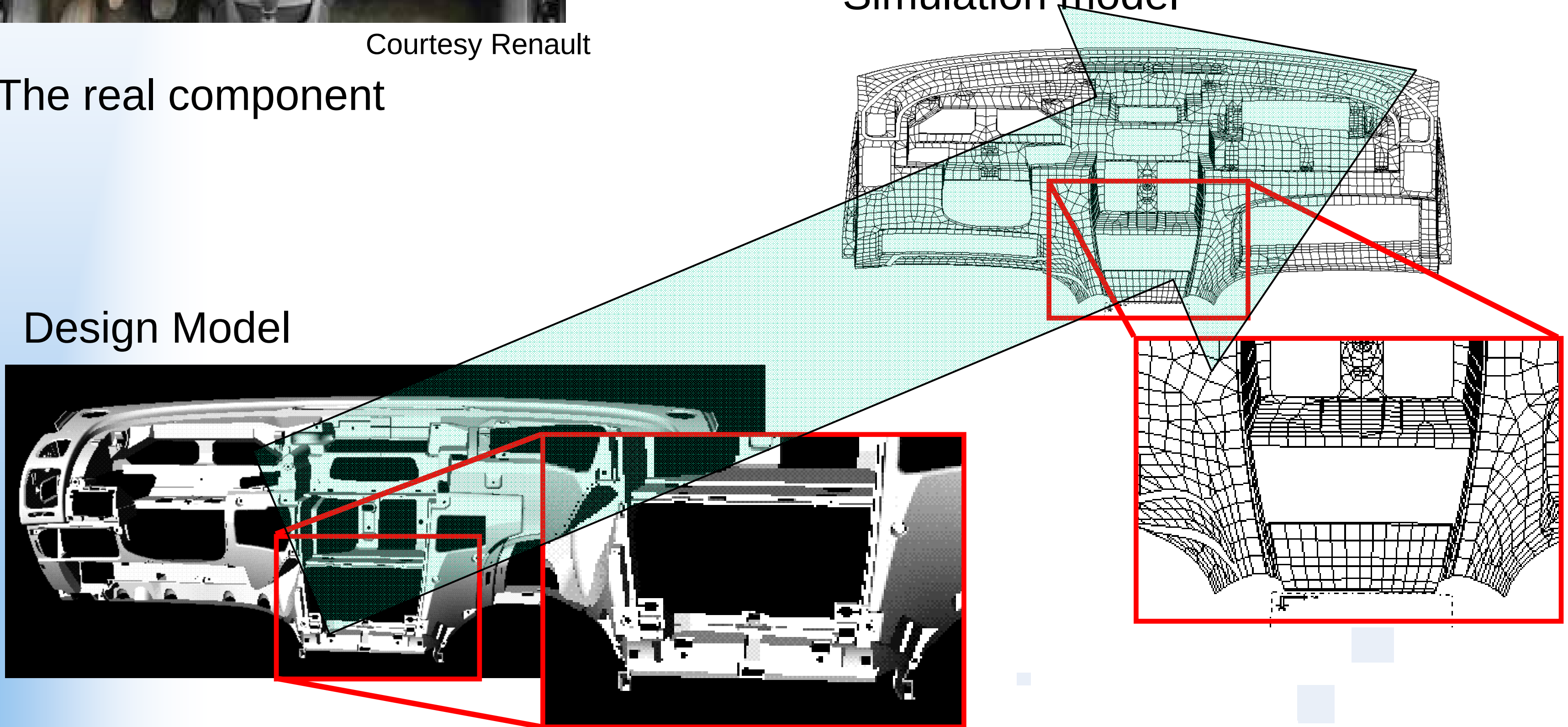
Courtesy Renault

The real component

Other
Representations
...

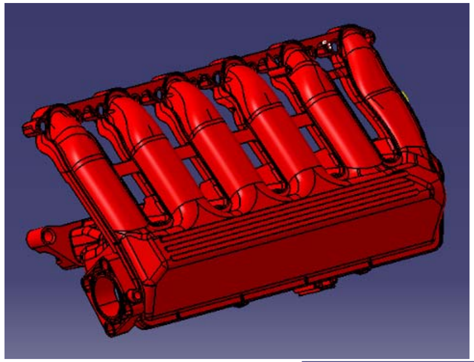
Simulation model

Design Model



# Context – Multiple representations

A car engine fuel inlet component as defined by the engineering office

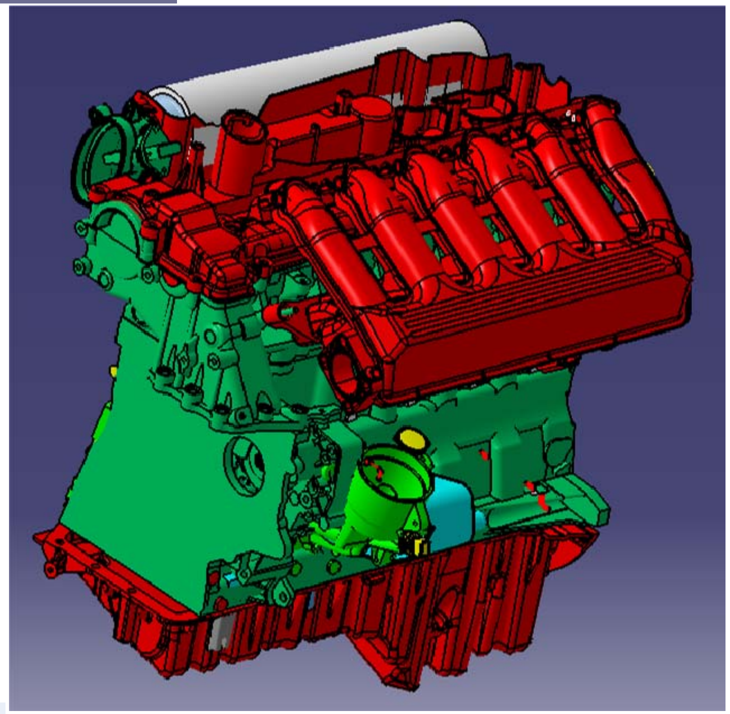


Courtesy BMW

Design Model

The same component inserted in the engine digital mock-up

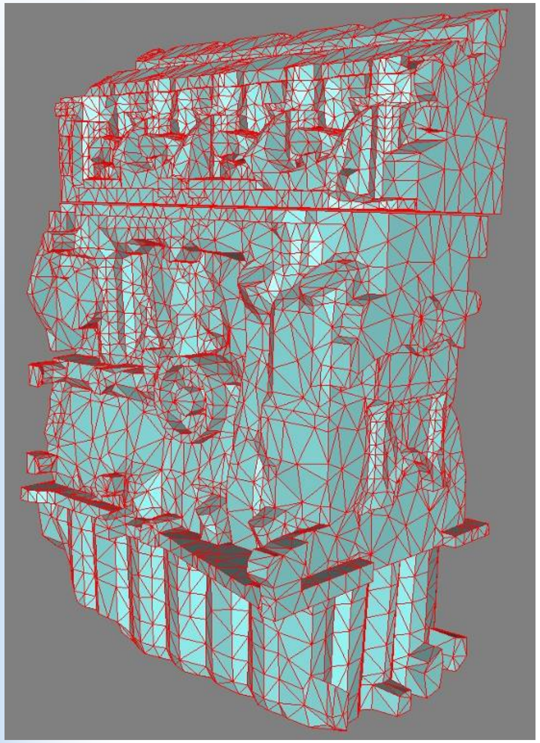
Design Model



Courtesy BMW

# Context – Multiple representations

A car engine considered as single object for thermal simulation

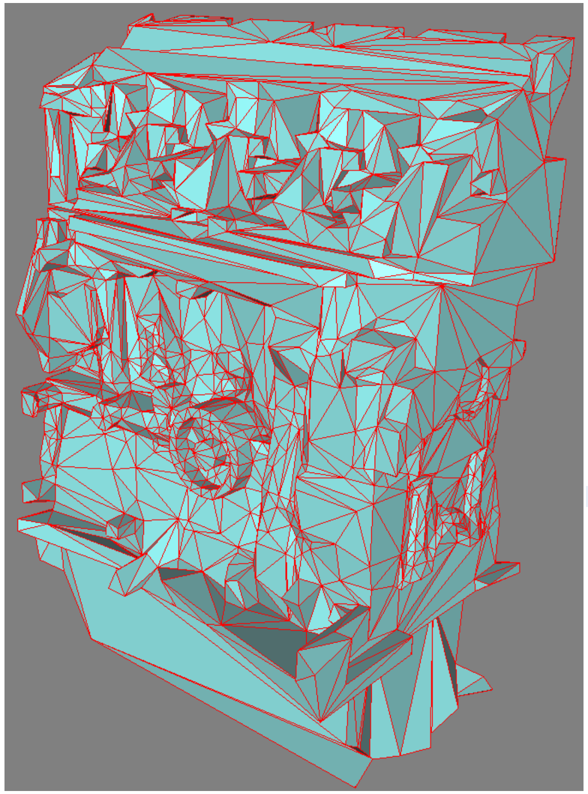


Courtesy Metravib

Finite Element (FE) mesh of the car engine

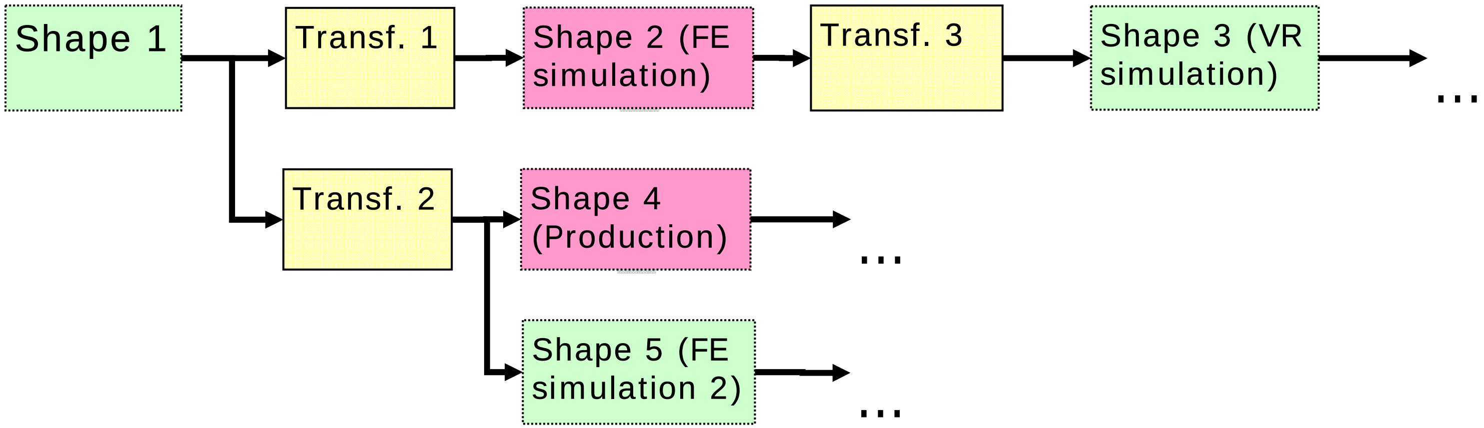
Geometric model as basis for the FE mesh generation

A car engine considered as single object for acoustic simulation

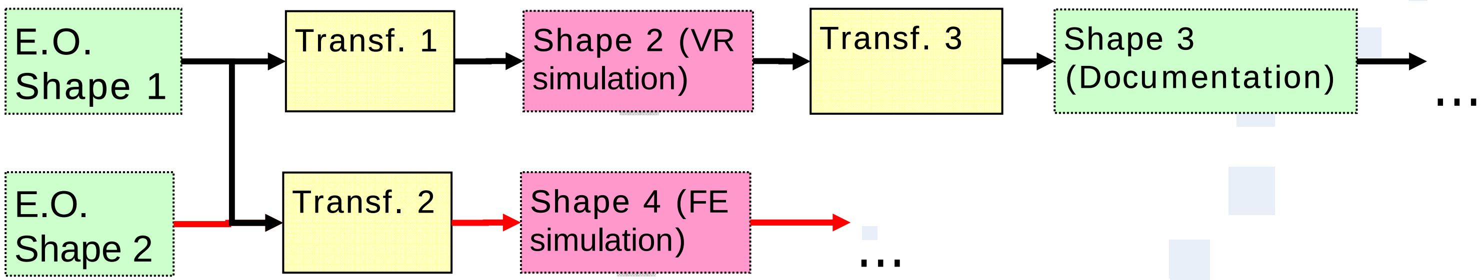


# Context – Multiple representations

- Most of the time, representations are derived from each other in accordance to a PDP structure,



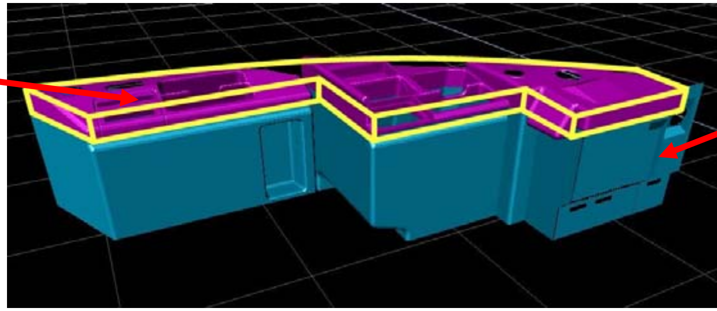

- Currently, the digital shape produced by the Engineering Office (EO) is often at the root of the PDP



# Context – Multiple representations

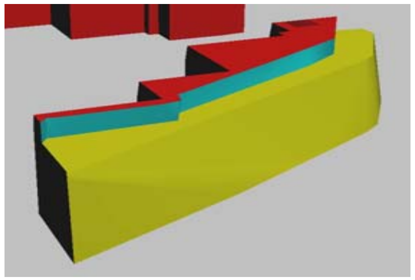
- Example of multiple shapes transformed into a single one

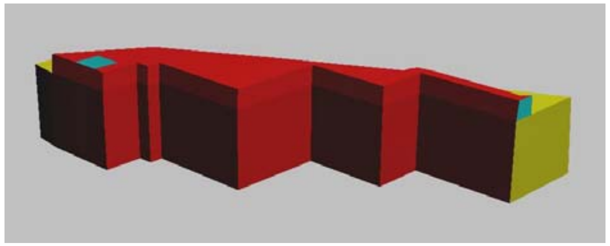
E.O. Shape 1



E.O. Shape 2

Courtesy EADS IW





Shape for thermal simulation

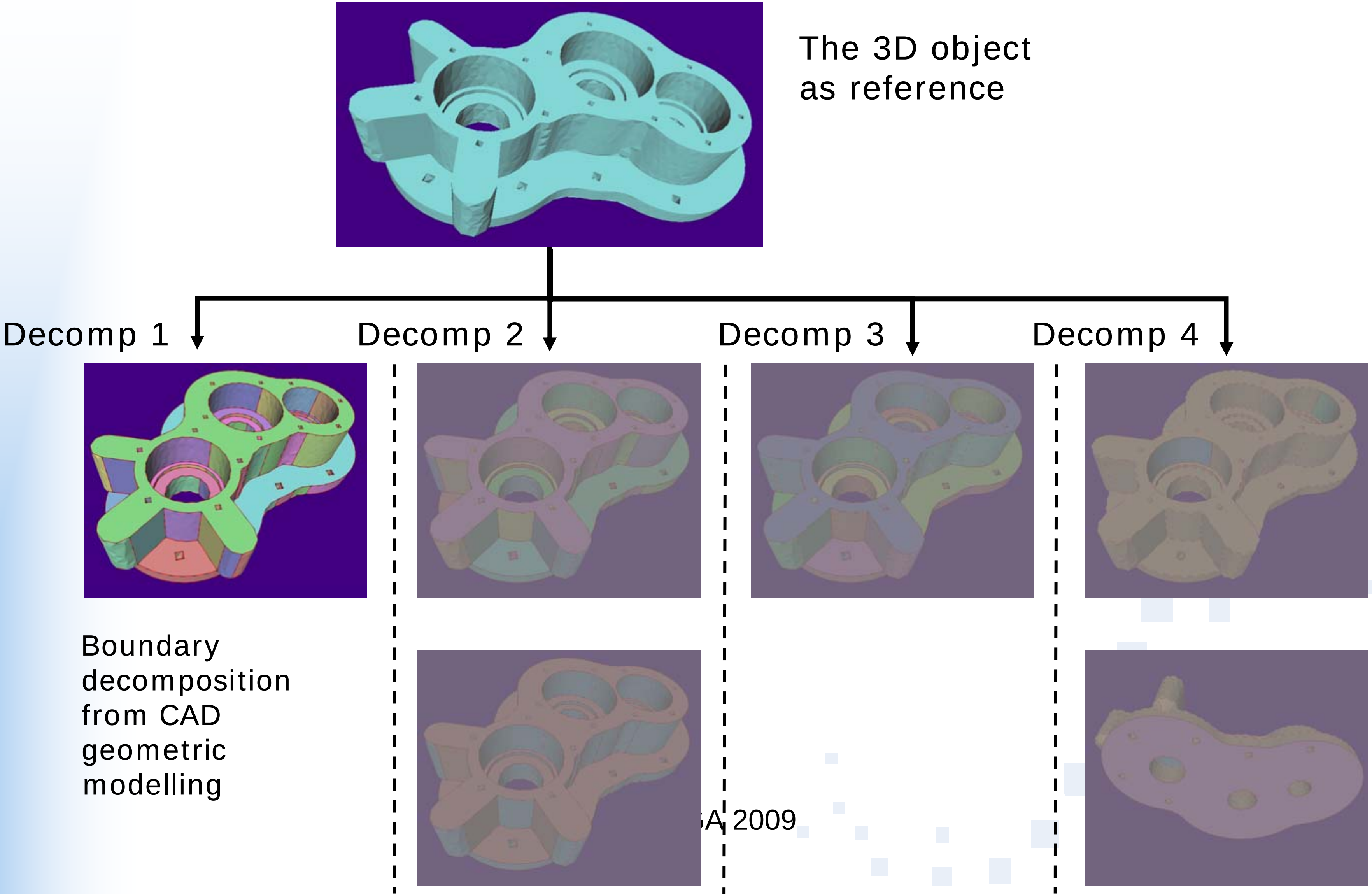
# Context – Multiple representations

- Currently, these multiple representations are often independent shapes in the sense that:
  - A modification on an upstream shape cannot be automatically propagated to its downstream ones,
  - Shape transformations are generated with different software (COTS),
  - Shape transformations are performed with the help of a human being who adds data: they cannot be automated,
  - Shape transformations must incorporate constraints of geometric modellers to match their software configuration and capabilities

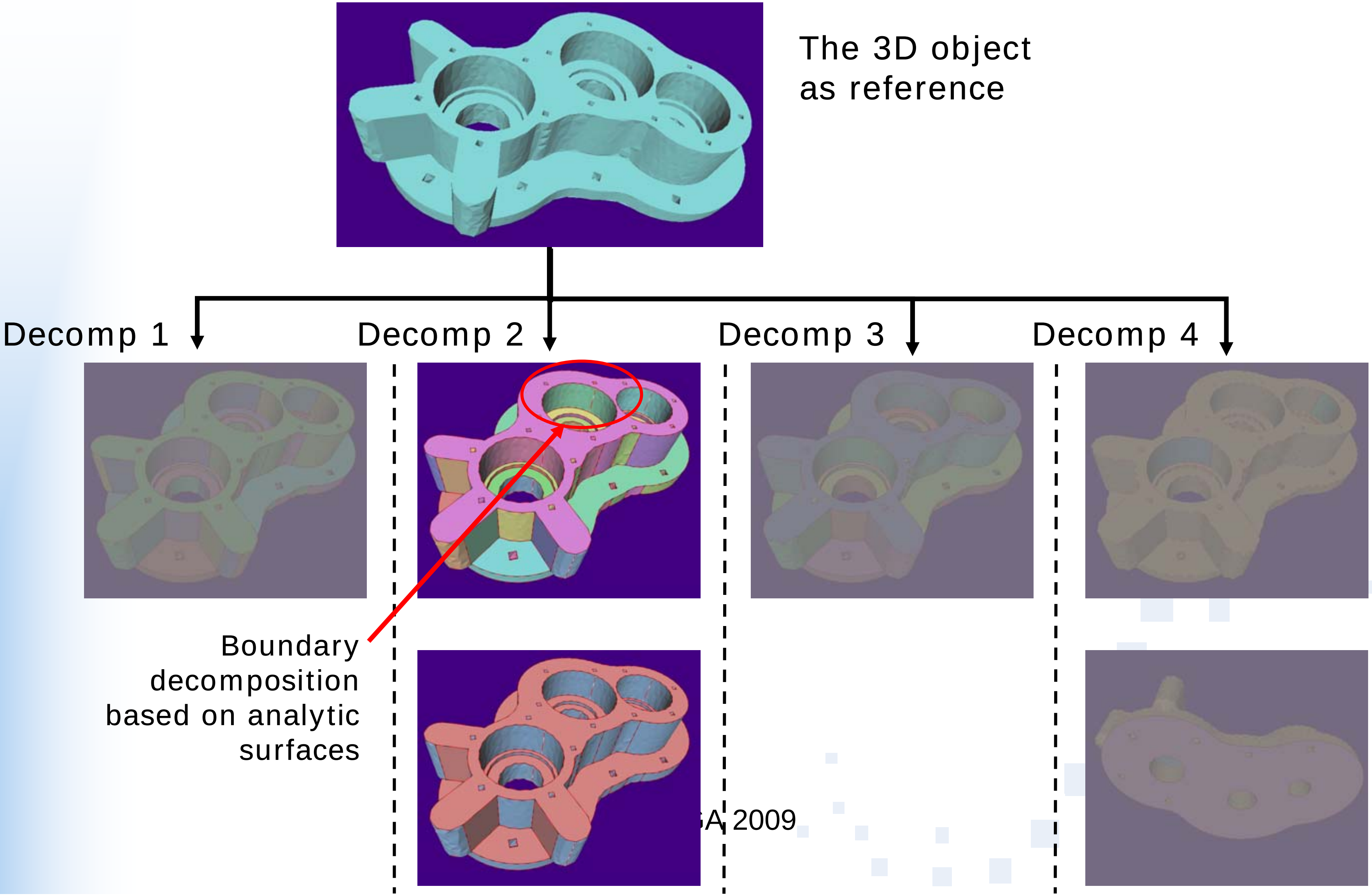
# Context – Multiple representations

- Shapes are not restricted to 3D objects in a PDP,
- Because of its usage, a shape boundary is subdivided into sub domains where different informations are located,
- Shape boundary decomposition depends on the PDP step considered. This decomposition is often referred to as shape feature,
- Shape interior decomposition may be needed for specific applications, e.g. to express an object decomposition into several materials,
- Similarly, multiple 3D shapes for a unique object, its boundary decomposition is not unique but guided by the PDP needs at a given step.

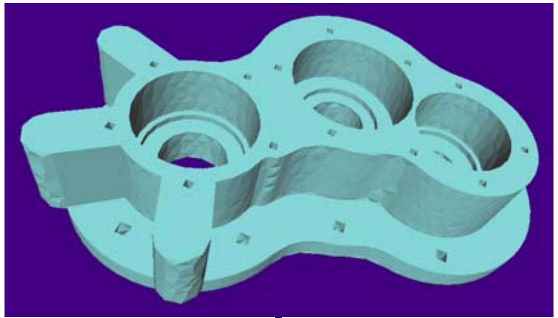
# Context – Multiple representations



# Context – Multiple representations

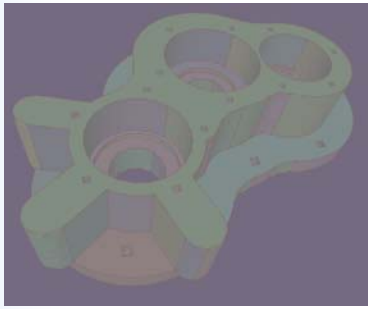


# Context – Multiple representations

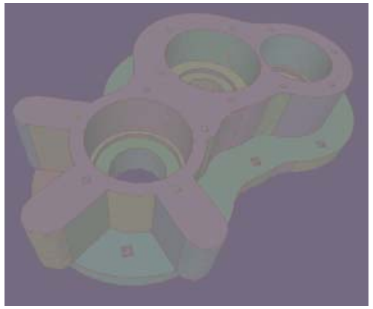


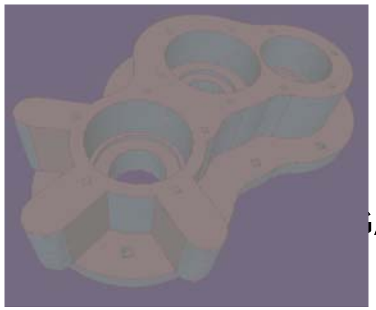
The 3D object as reference

Decomp 1

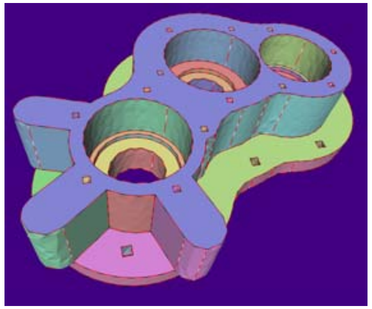


Decomp 2



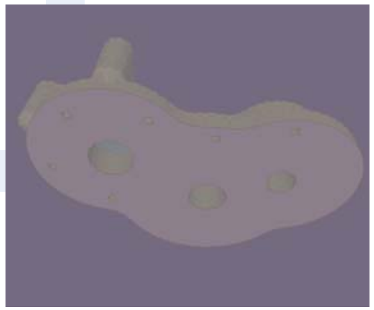


Decomp 3

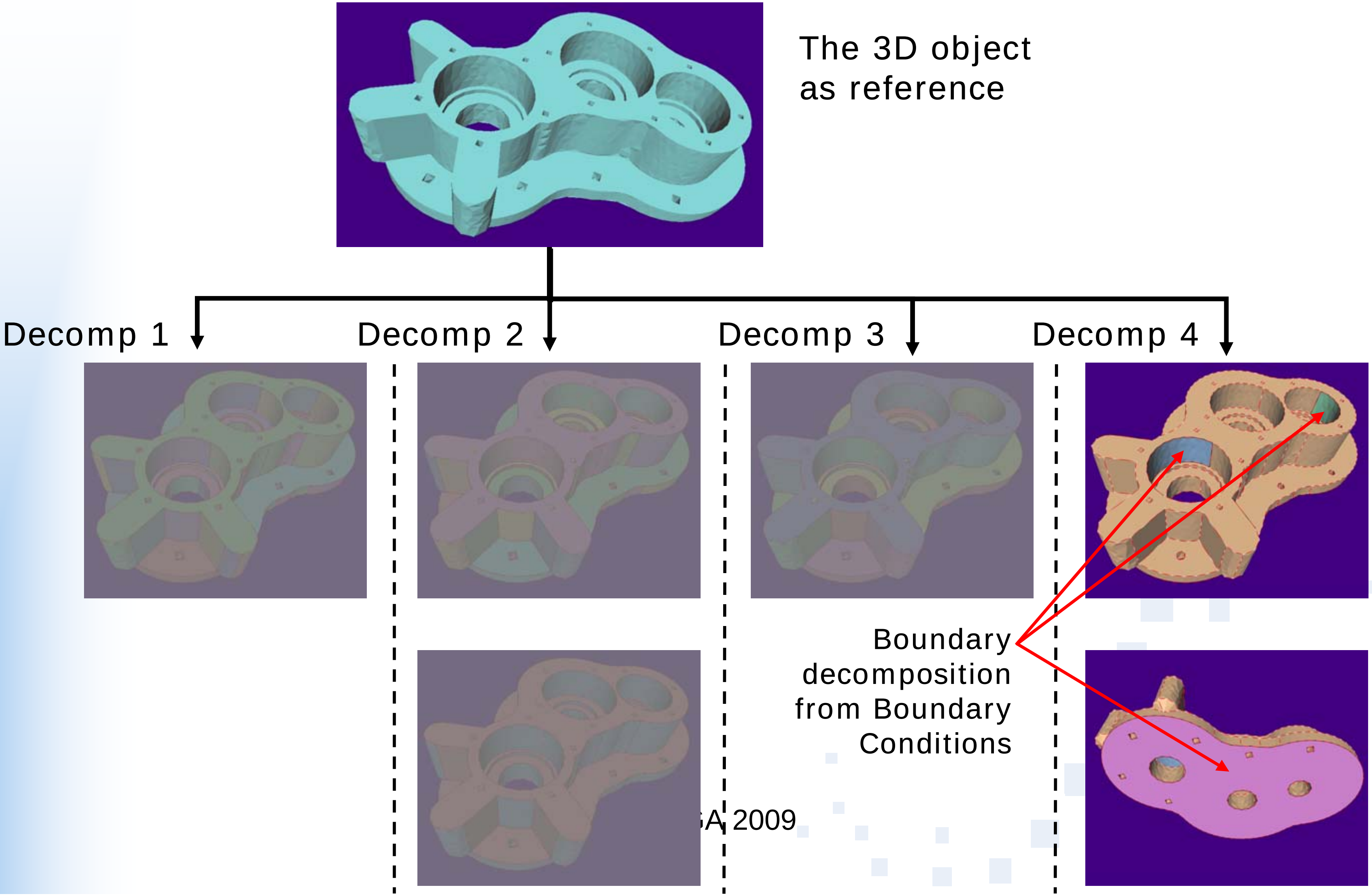


Boundary decomposition based on smooth edges

Decomp 4



# Context – Multiple representations



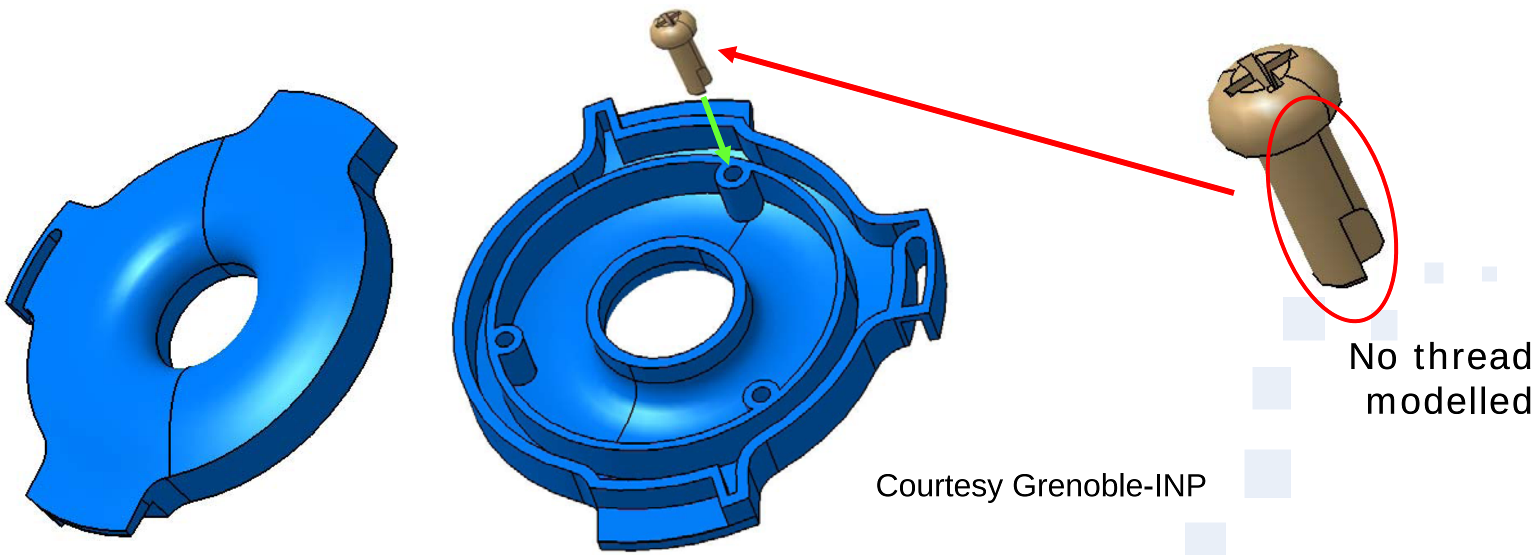
# Context – Multiple representations

- When shape transformations are operated during a PDP, they can couple:
  - Shape transformations in 3D,
  - Shape boundary decomposition transformation,

  To:
  - Conform to the requirements of some PDP step, e.g. FE for structural mechanics behaviour,
  - Form a transformation intrinsic to a PDP step, i.e. it does not rely on data combined across multiple PDP steps and it should be chronology independent.

# Shape diversity in a PDP shape categories

- Shape categories reflect diverse needs to describe objects in a PDP:
  - Volumes are mandatory to describe a physical object:
    - They are widespread in a PDP to produce representations close to real components / sub-systems,
    - Often, they are modelled through their boundary as 2-manifolds

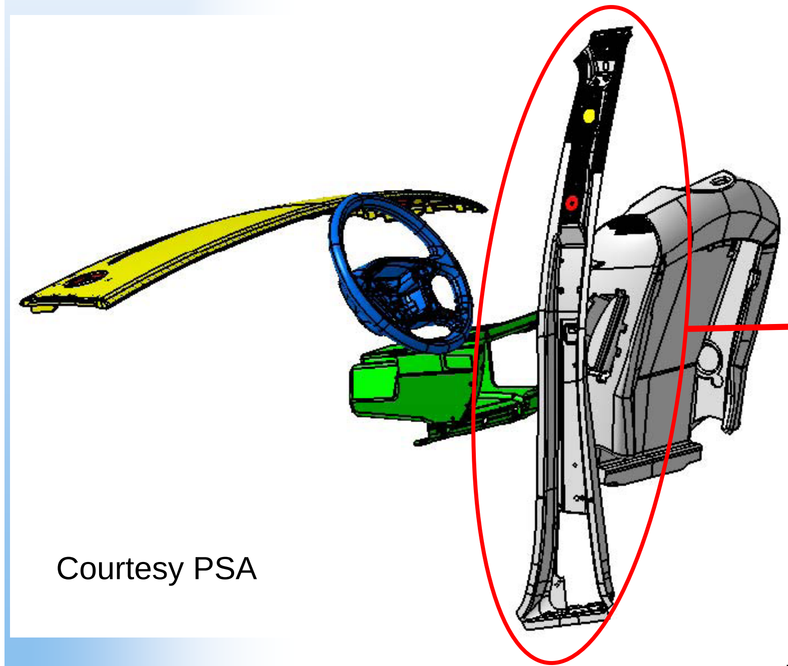



Courtesy Grenoble-INP

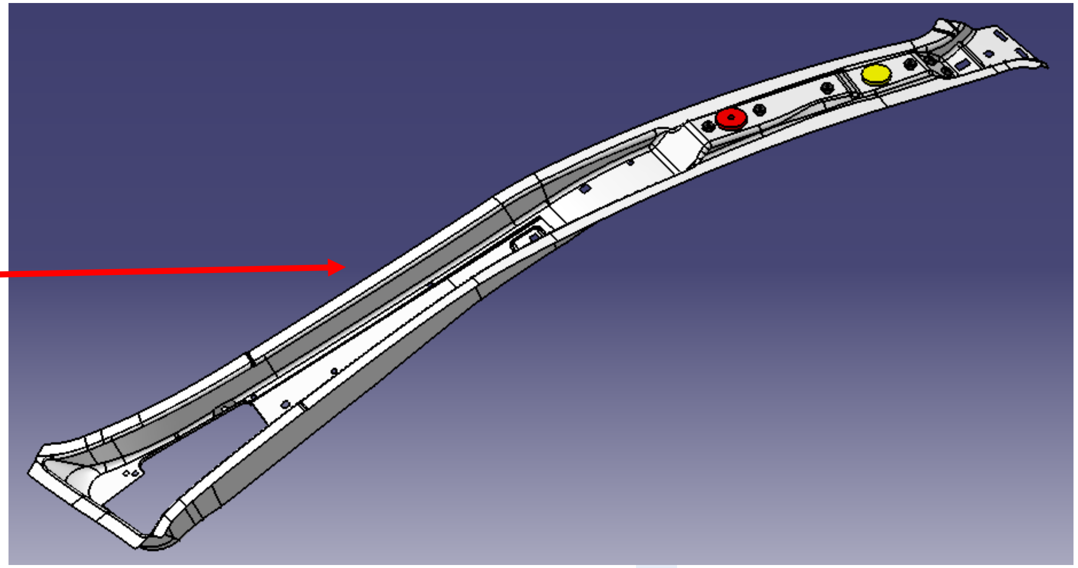
J-C Léon – JGA 2009

# Shape diversity in a PDP
# shape categories

– surfaces are very common to describe free-form objects:
  - They are often used to describe thin shells. They can be regarded as an abstraction (idealization or manifold dimension reduction) of thin shells,
  - Their category is designated as 2-manifolds with boundary(ies)

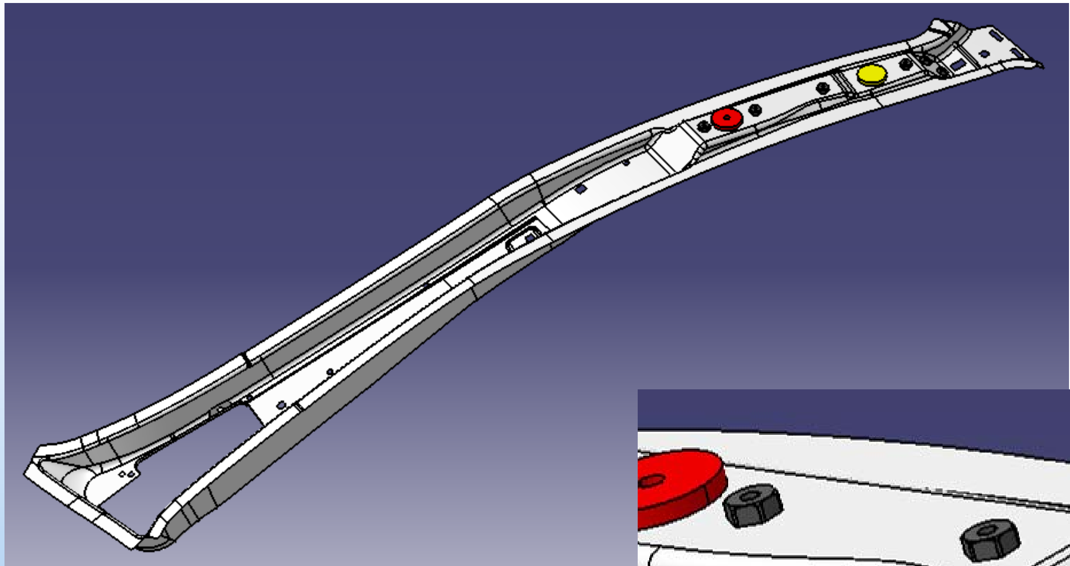


Courtesy PSA

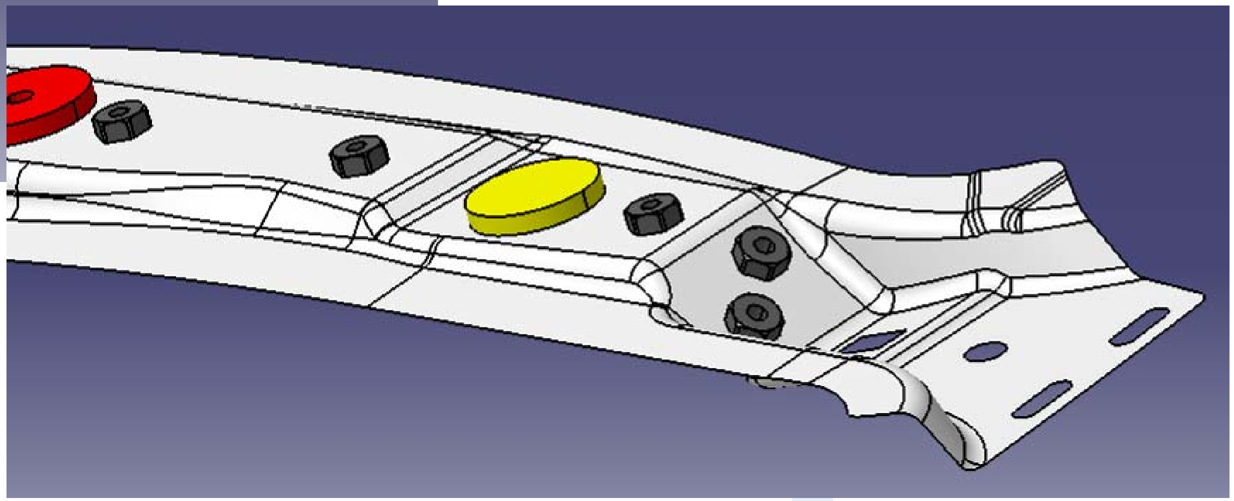


# Shape diversity in a PDP
# shape categories

- Example of shell type object idealized as 2-manifold with boundaries

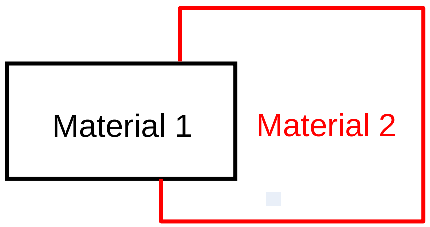


Courtesy PSA



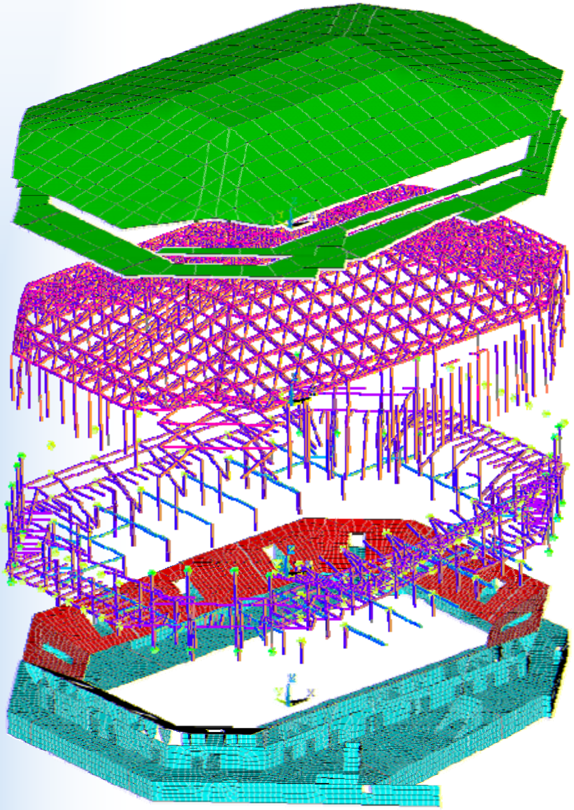
# Shape diversity in a PDP
# shape categories

– ‘combinations’ of curves, surfaces, volumes are very common to describe models:
  - often they reflect idealizations of component sub-domains for simulation purposes (local manifold dimension reduction),
  - Their category is designated as non-manifold because:
    – Curves, surfaces, volumes are connected together and form non-manifold singularities in the object,
    – The object is defined with several cells, which can be used to define several adjacent domains, e.g. different materials. The models are often designated as ‘cellular models’.

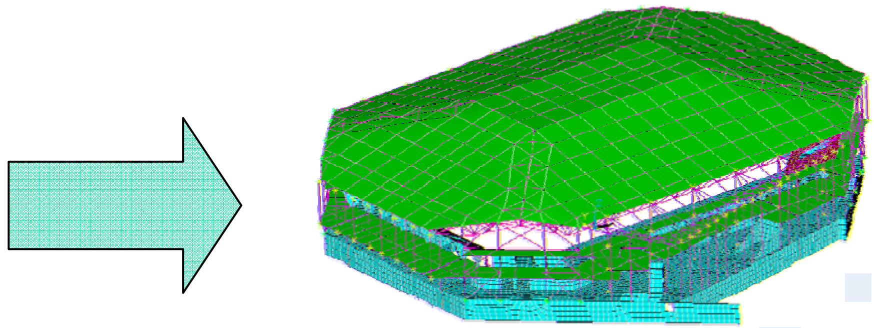


# Shape diversity in a PDP
# shape categories

An example of simulation model defined as a non-manifold model. Lines and surfaces may be connected or not in accordance with mechanical hypotheses.



Courtesy D Lovinfosse



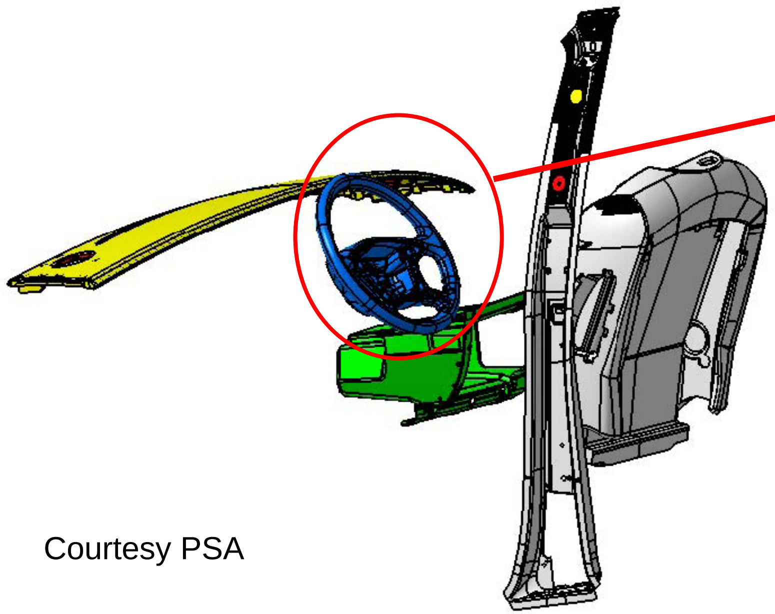
A FE simulation model of a civil engineering structure (shell, plate, beam FE elements)

G-SCOP

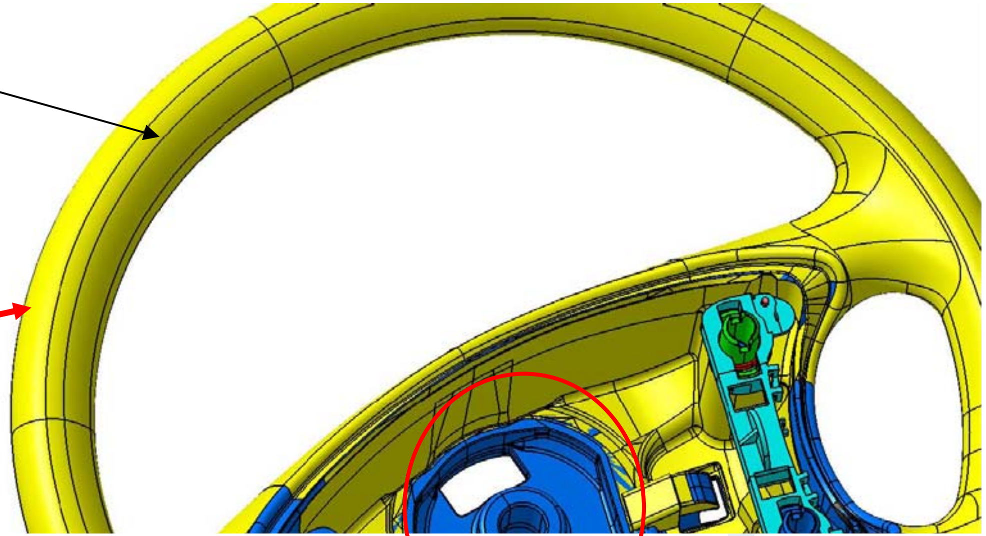
# Shape diversity in a PDP
# shape categories

An example of object defined as a cellular model.

Plastic sub-domain



Courtesy PSA



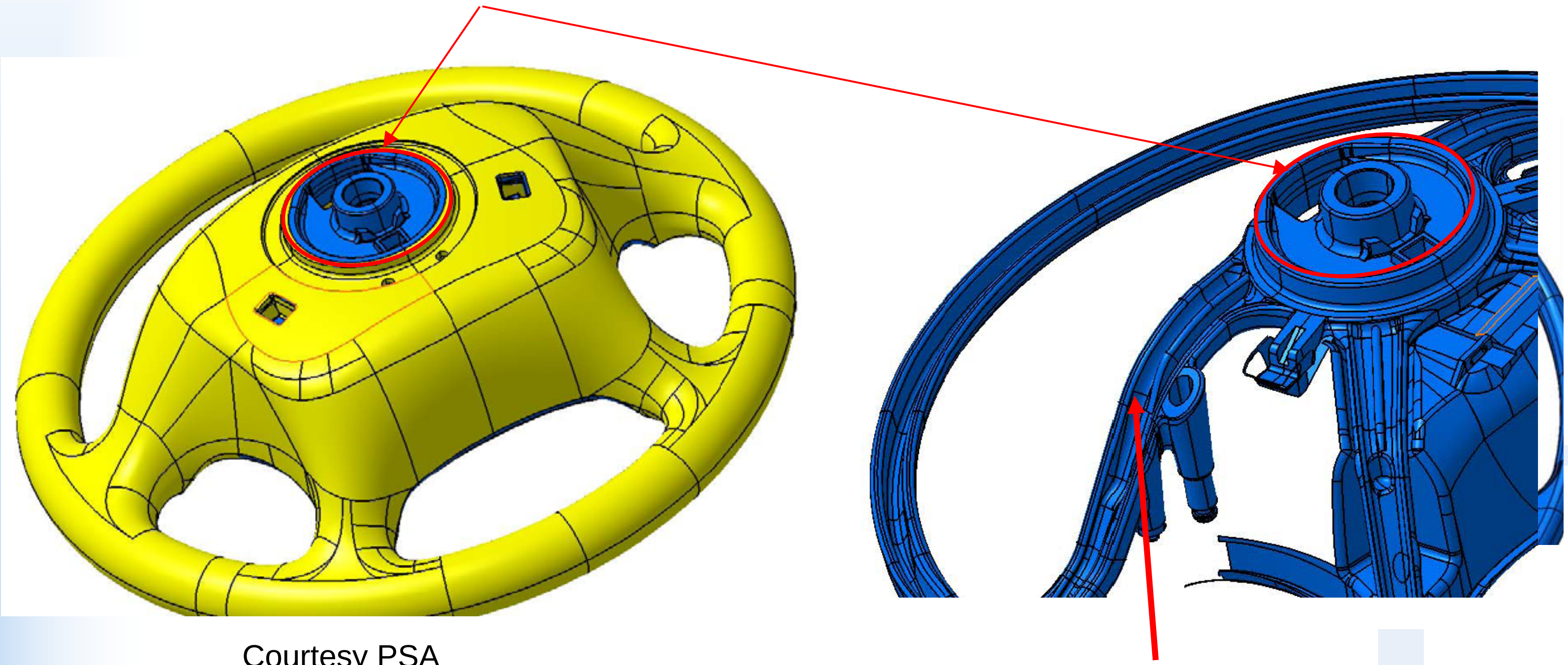
Courtesy PSA

Detail of the steering wheel highlighting different materials.

Steel sub-domain

# Shape diversity in a PDP shape categories

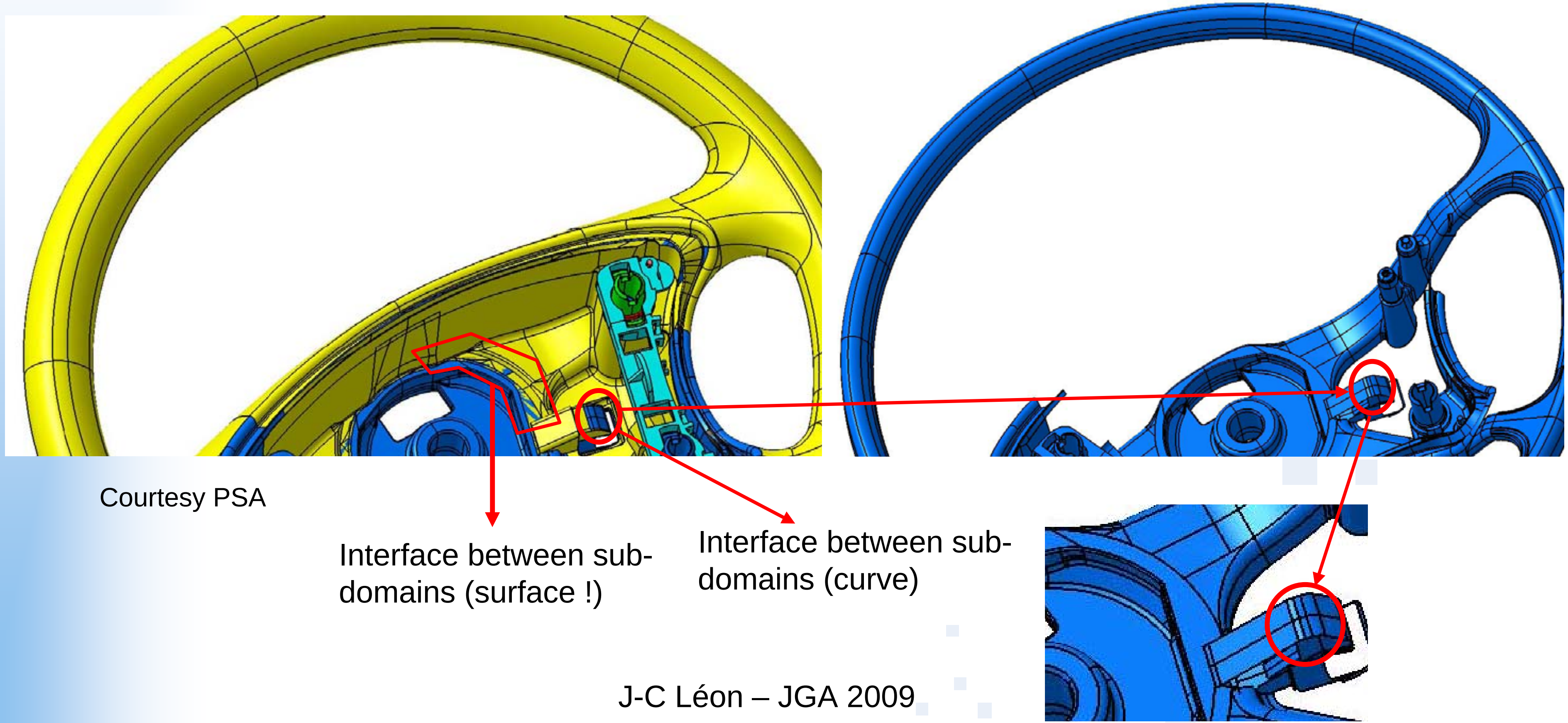
- Example of interface between cells: non-manifold singularities (steel/plastic interfaces) along curves




Courtesy PSA

# Shape diversity in a PDP shape categories

- Example of interface between cells: non-manifold singularities (steel/plastic interfaces) along surfaces and curves

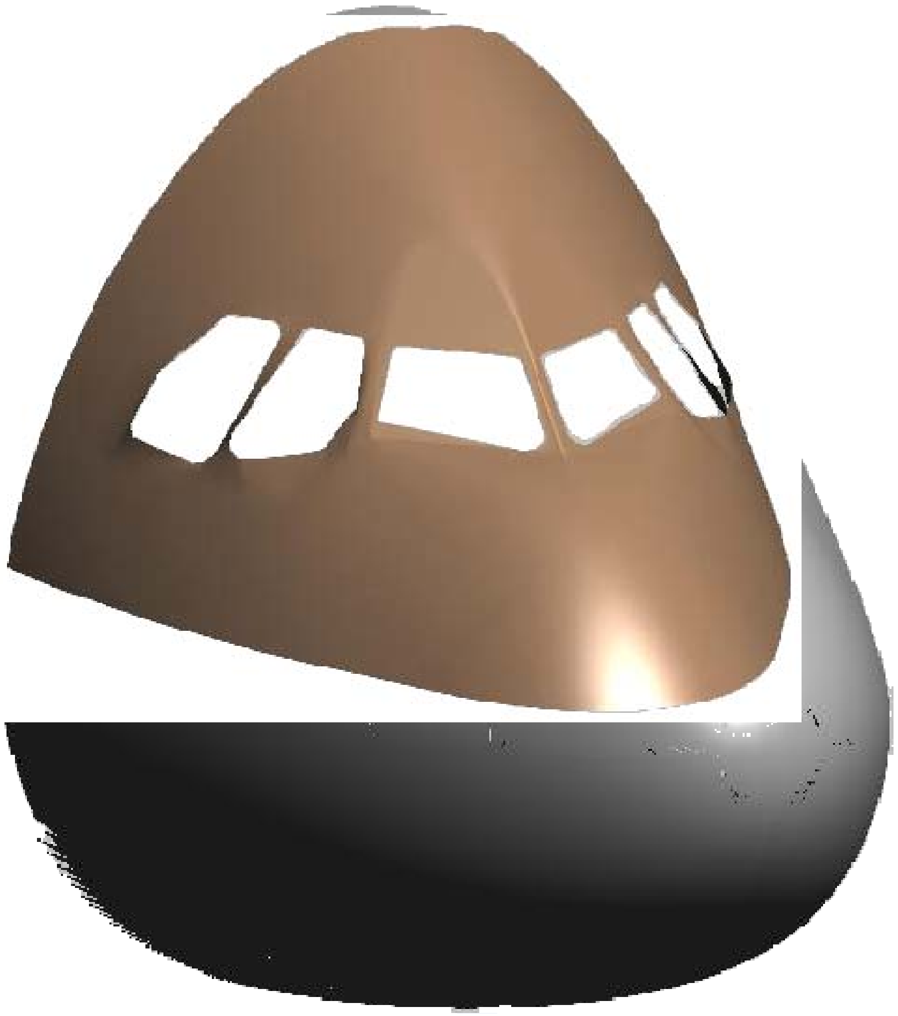



Courtesy PSA

# Shape diversity in a PDP
# shape categories

- Often cellular models are needed for simulation purposes. Cells are required to express different materials, different types of FE.

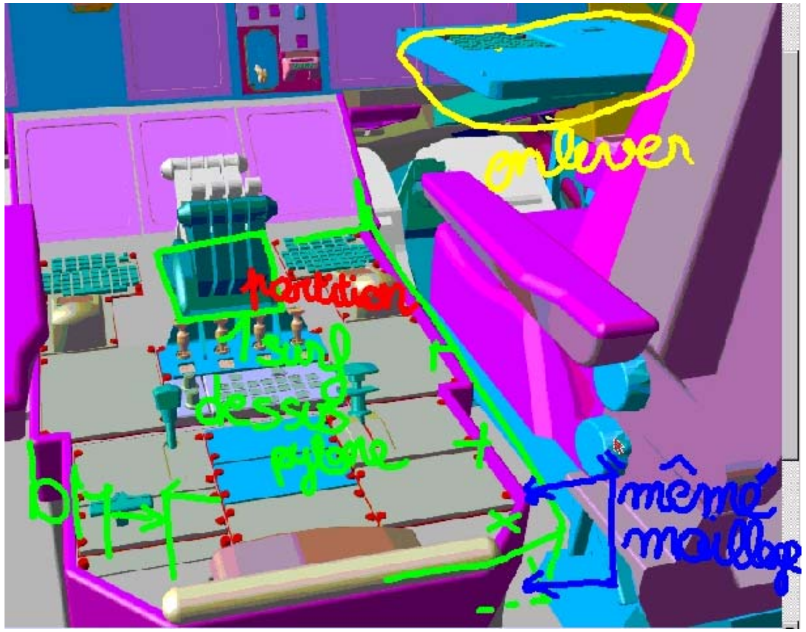
  Assemblies (sets of volumes) often end up as cellular models.



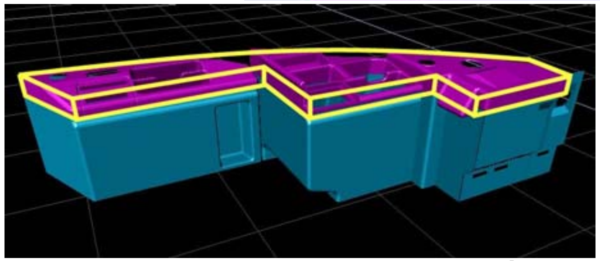
Thermal model of A380 cockpit
(courtesy Airbus France EEI)

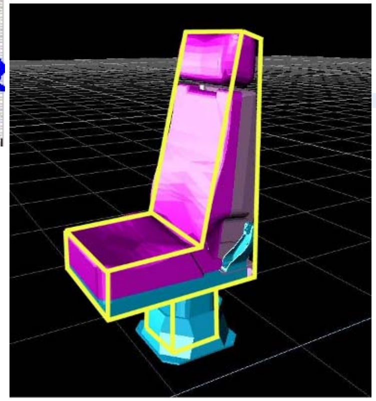
# Shape diversity in a PDP
# shape categories

- Example of the numerous components forming the assembly representing a subset of the cockpit



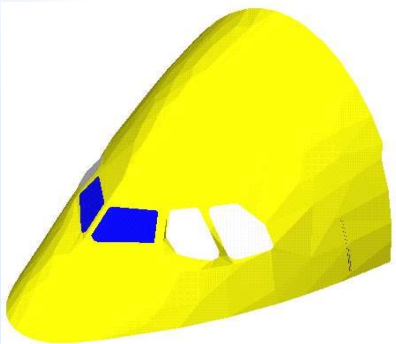
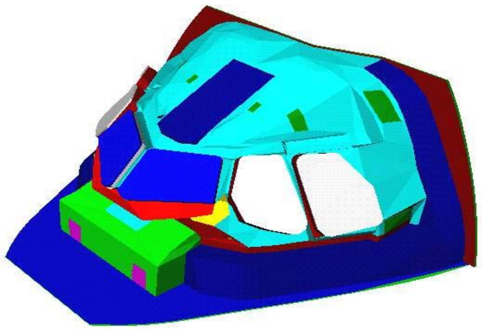
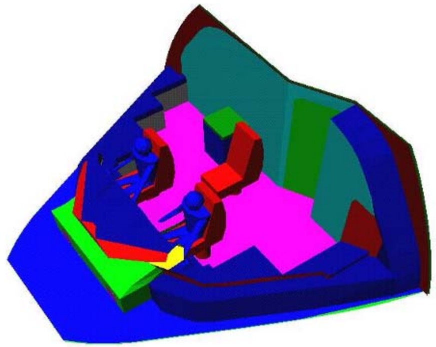

Courtesy Airbus France EEI, EADS IW



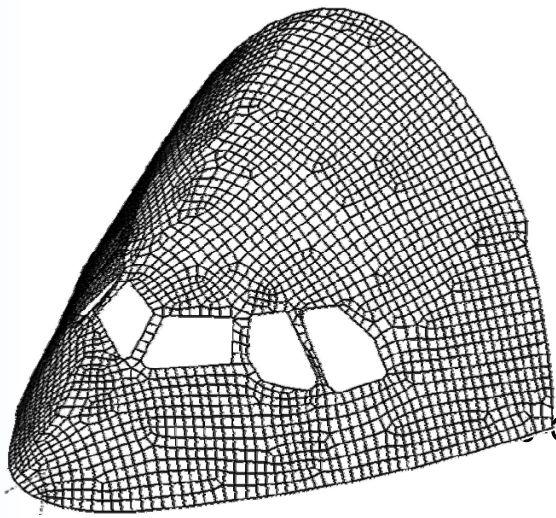
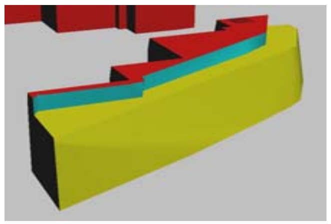
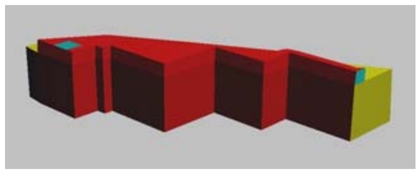
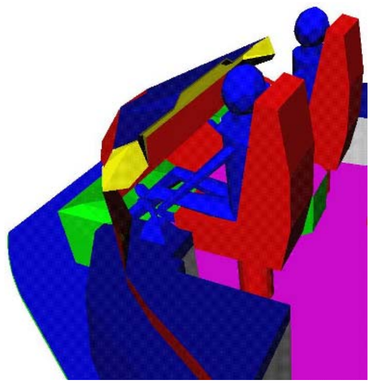


# Shape diversity in a PDP
# shape categories

- Assembly transformed into a cellular model for thermal simulation purposes. Cells are further decomposed according to material requirements (here colors).

Simpoly®

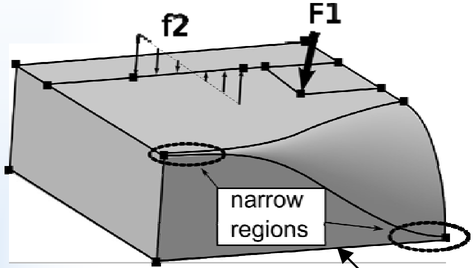





# Shape diversity in a PDP shape boundary decomposition

- At various PDP steps, information attached to a shape boundary becomes critical and its layout must be structured,
- Currently, shape boundary decomposition is often bounded by modeller constrains rather than PDP needs,
- Topological properties are also of interest to characterize the information laid out on a shape boundary,
- Configurations of boundary decomposition are examplified with in the context of FE model preparation phase to:
  – Produce an appropriate set of FE mesh generation constraints,
  – Describe appropriate shape features and boundary conditions.

# Shape diversity in a PDP
# shape boundary decomposition

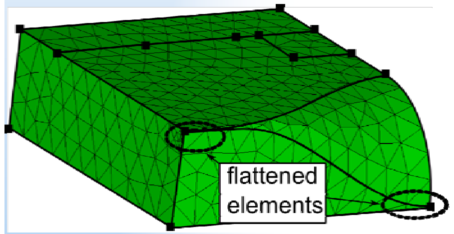
- In a PDP, FE model preparation is preliminary to FE behaviour simulation and involves boundary decomposition transformation



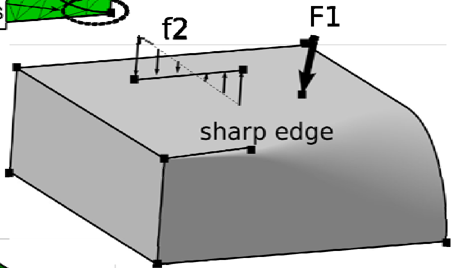

Simple component: geometric model with boundary conditionsc

Model paving based on geometric modelling requirements

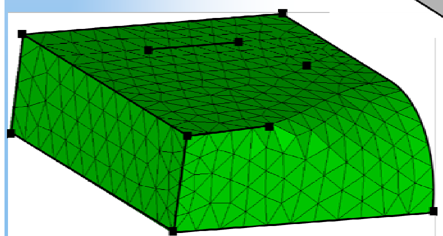
Initial CAD model




FE mesh generated from geometric model paving



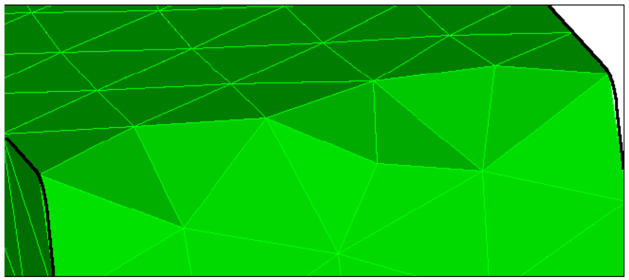

Component boundary decomposition based on FE simulation requirements



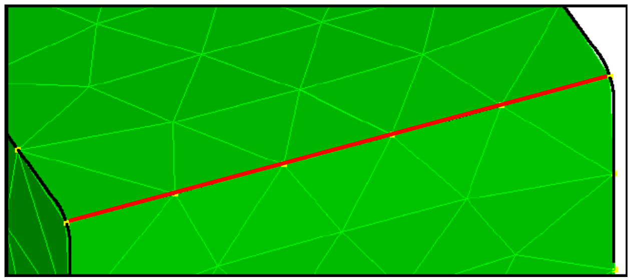
FE mesh generated from FE simulation requirements

# Shape diversity in a PDP
# shape boundary decomposition

- Boundary decomposition for FE simulation is essentially governed by:
  - Specification of boundary conditions,
  - Monitoring the domain discretization: a mean to monitor locally the deviation between a FE mesh and the input model, e.g. prescribing a line,



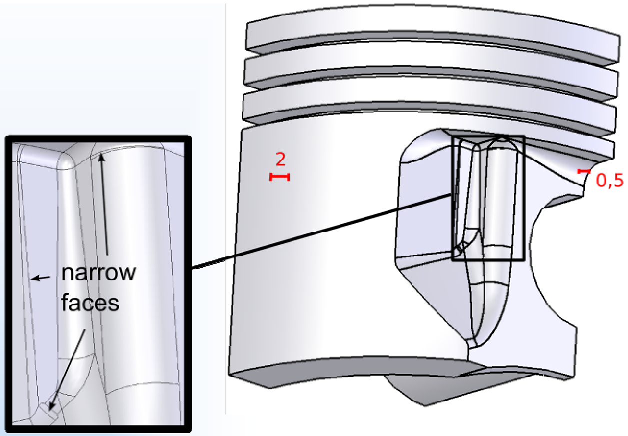
Meshing a smooth blending area with large FEs and discretization deviation attached to the surface only



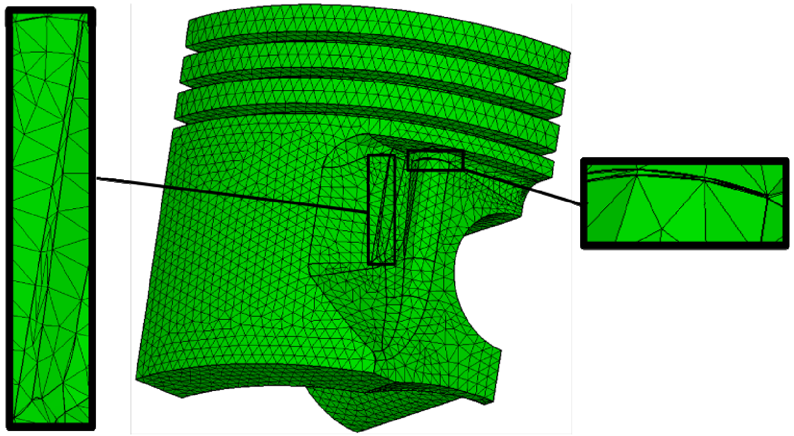
Meshing a smooth blending area with large FEs and discretization deviation locally monitored by the ‘red line’

# Shape diversity in a PDP
# shape boundary decomposition

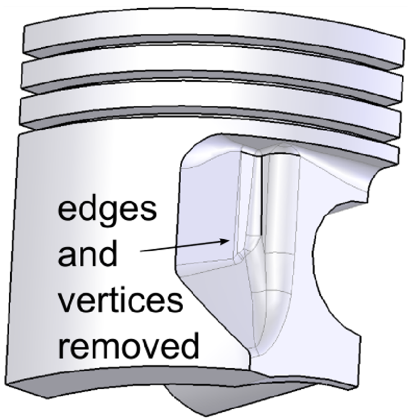
Initial model with boundary decomposition obtained from CAD modelling

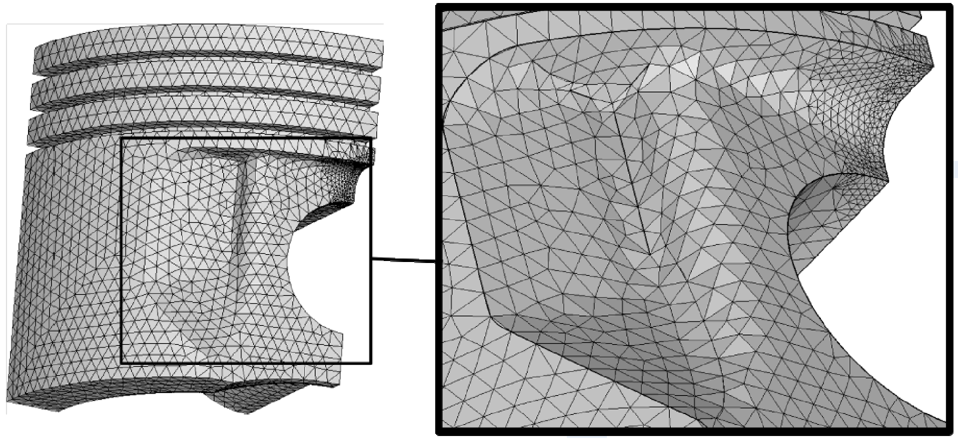



flattened and distorded elements are caused by the faces which are smaller than the target size



Boundary decomposition compatible with FE mesh size requirements



# Shape diversity in a PDP Synthesis

- Shape diversity during a PDP is related to PDP step requirements and shape transformations between successive PDP steps,
- Improving a PDP can be achieved with more powerful:
  - shape modelling,
  - shape transformation,

  capabilities to produce representations intrinsic to each PDP step,
- Non-manifold shapes are widespread in a PDP,
- Modelling shapes and performing shape transformations use local and global topological properties,
- Global topological properties related to shapes are basic building blocks of shape modelling and transformation processes

# Industrial modeller capabilities

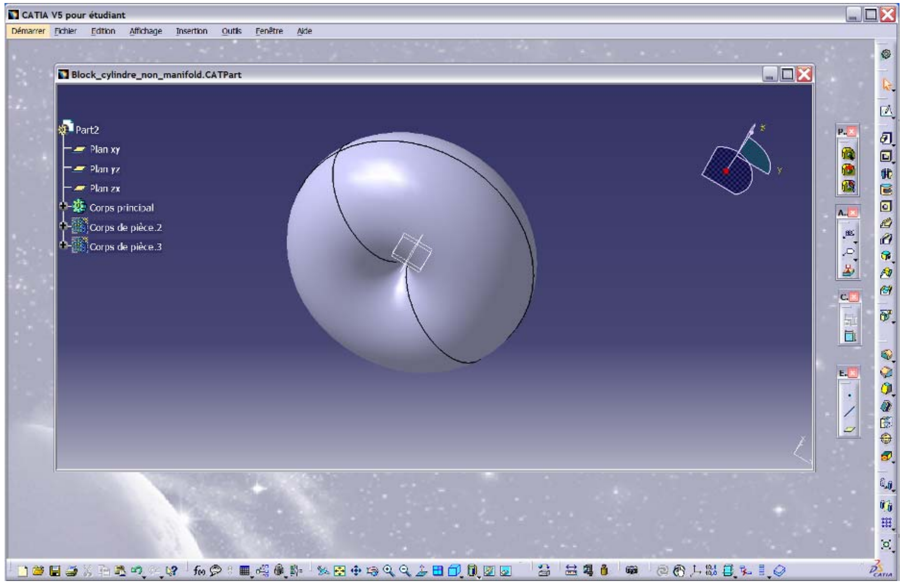
- To support PDP steps, current software incorporates:
  - Volume modellers,
  - Surface and wireframe modellers,
  - Connecting volumes, surfaces, lines as a restricted set of non-manifold connections.

- Boundary decomposition of objects according to ‘cell decomposition’

- Consequently, ‘modeling noise’ is created across a PDP

# Industrial modeller capabilities

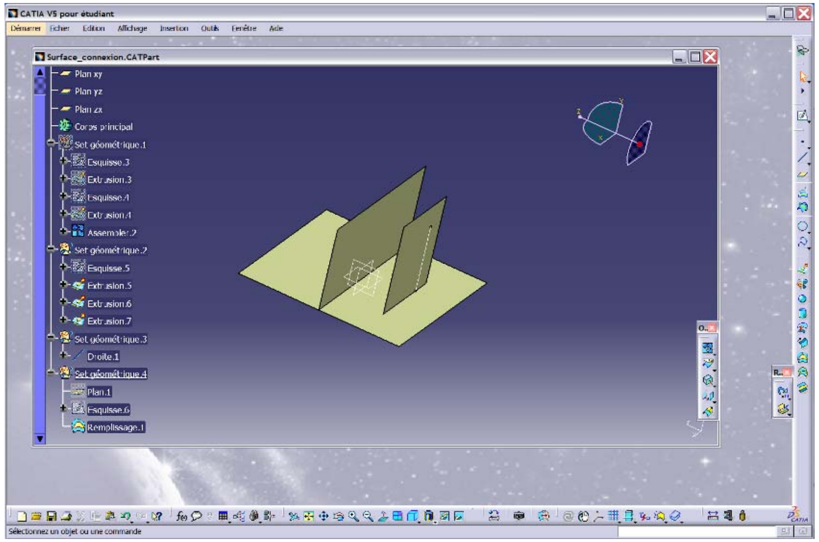
- Are volume modellers able to robustly distinguish volumes from other objects ?
  - Simple demos catiaV5:
    - Torus and degenerated torus,
    - Tangent hole, contact shells
    - Volumes becoming non-manifold objects (structure of a volume modeller)



G-SCOP

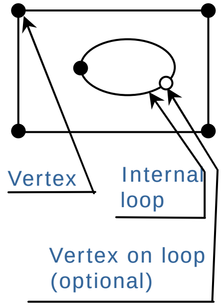
# Industrial modeller capabilities

- Are surface modellers able to robustly distinguish surfaces from other objects ?
  - Simple demos catiaV5:
    - Connection between two surfaces,
    - Connection between three surfaces,
    - Connection between line and surfaces, internal connection

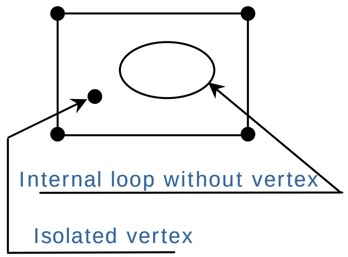


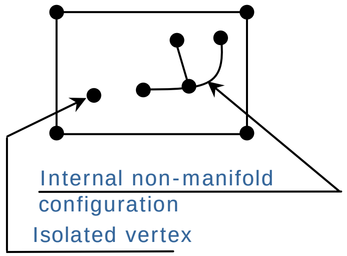
# Industrial modeller capabilities

- Paving constraints incorporated in geometric modellers for 2-manifolds with or without boundaries




- Some paving constraints needed at some PDP step



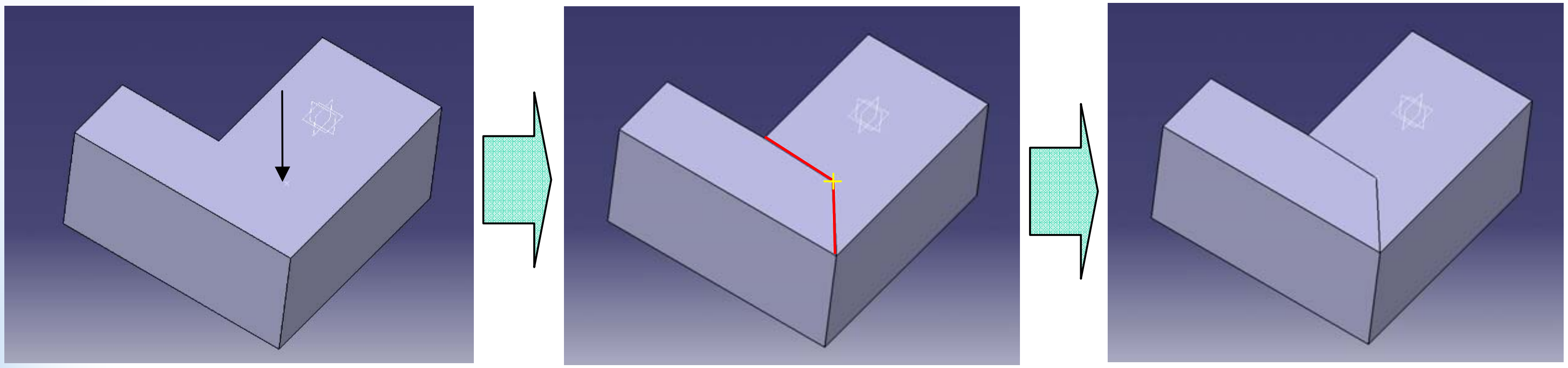



# Industrial modeller capabilities
# Synthesis

- Modelling non-manifold shapes is a real need in a PDP but scarcely available in industrial modellers,
- Shape decomposition transformations rely on low level functions and are tedious operations,
- Shape transformations incorporating manifold dimension reductions cannot be guided by global topological properties,
- Multiple shape transformations or shape decomposition transformations between PDP steps generate some ‘modelling noise’, reducing PDP integration. (See following example)

# Industrial modeller capabilities
# Synthesis

- To generate appropriate boundary decomposition, subdivisions are performed to add further extraneous constraints,



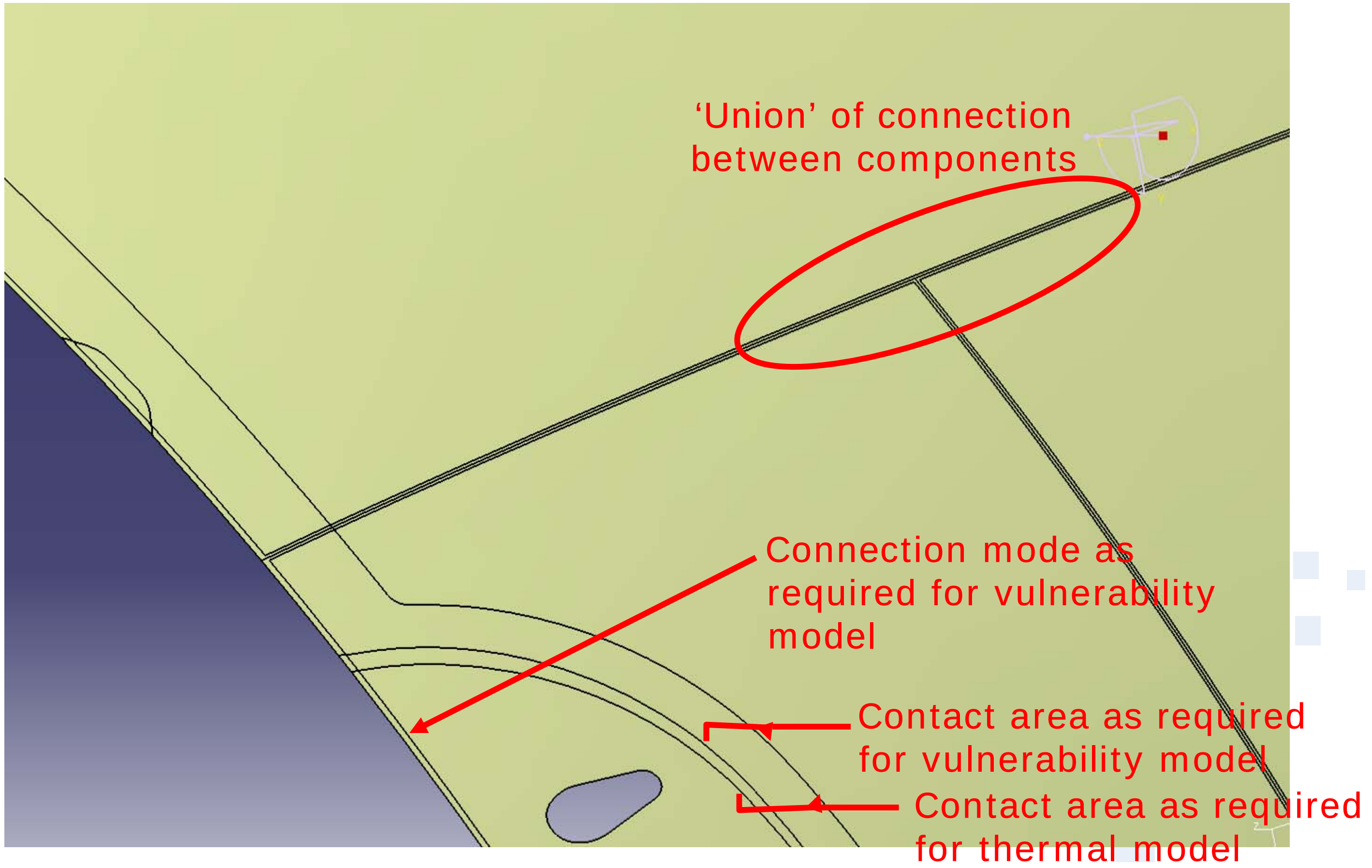
Desired model

Model built to incorporate the desired constraints

- Processing the model at a next PDP step may require to merge again the faces to keep only the desired decomposition, e.g. FE mesh generation. Decomposition constraints generate ‘modelling noise’

# Industrial modeller capabilities
# Synthesis

- Illustration of the multiple decompositions needed for different simulations of assemblies




Courtesy Airbus France, EADS IW

# Industrial modeller capabilities
# Synthesis

- Often, shape boundary transformations between PDP steps are performed interactively with low level functions (curve split/merge, surface split/merge),

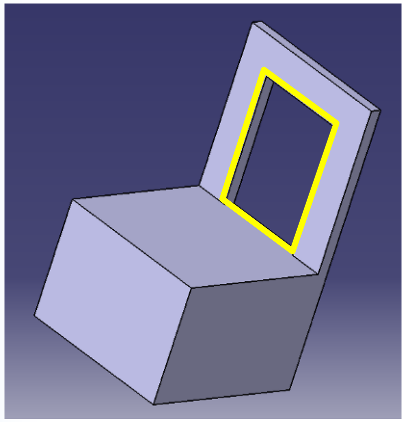
- Shape transformations or idealizations (manifold dimension reductions) are hardly supported by current modellers,

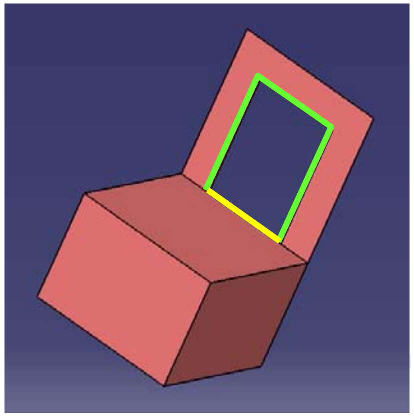
# Topology issues

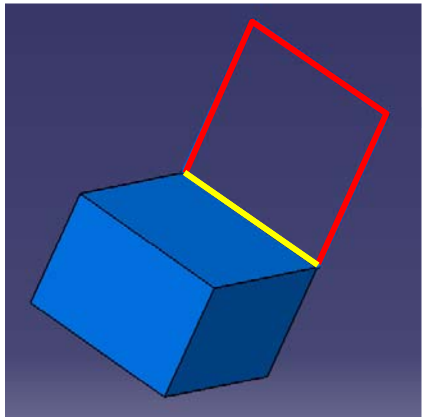
- Need for global topological properties to characterize non-manifold objects,
- Shape generation is strongly based on global topological properties, e.g. volumes,
- Need for establishing a connection between global topological properties and shape parameters,
- Shape boundary decomposition must describe intrinsically the vertices, edges and faces needed at each PDP step,
- Generating a shape taxonomy acts as a first use of topological properties for non-manifold objects

# Shapes in 3D Euclidean space

- A simple example showing different categories of non-manifold objects though sharing a common shape feature:
  1) a unique volume domain having a 'handle' or hole,
  2) a volume and a surface domains having a 'handle' or hole,
  3) a volume and a line domains having a 'handle' or hole,
- These three objects have different constitutive domains because they are manifolds of different dimensions (volume, surface, line) but they share an invariant topological feature: a handle or hole,

1) 

2) 

3) 

J-C Léon – JGA 2009

# Shapes in 3D Euclidean space

- As a first stage, simplicial complexes are addressed only, though CAD models often used in a PDP are not covered yet,

- Euler-Poincaré theorem:

  $$v - e + f = \beta_0 - \beta_1 + \beta_2$$

  is applicable, where $\beta_0$, $\beta_1$, $\beta_2$ are the Betti numbers characterizing independent equivalence classes, but the shape meaning of $\beta_1$ and the location of 1-cycles are not obvious,

- With 2-manifold objects, the meaning of 1-cycles is related to through holes:

  $$v - e + f = 2(s - h)$$

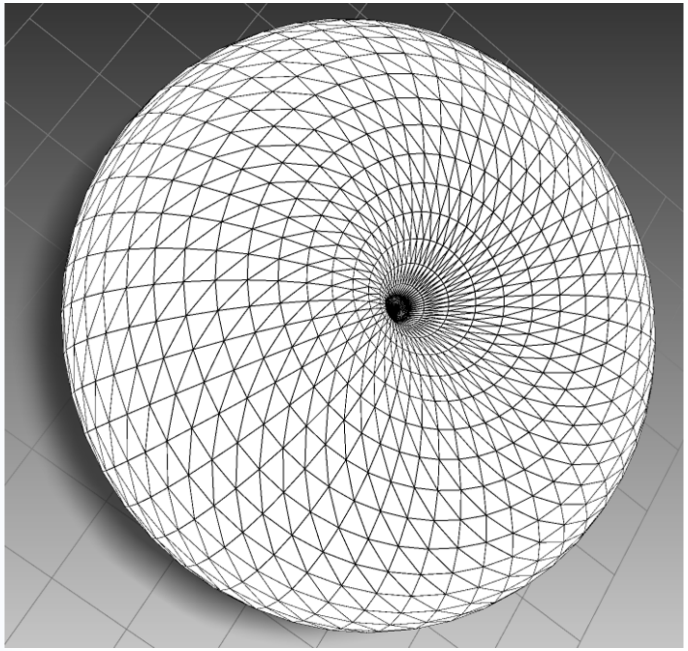
# Shapes in 3D Euclidean space

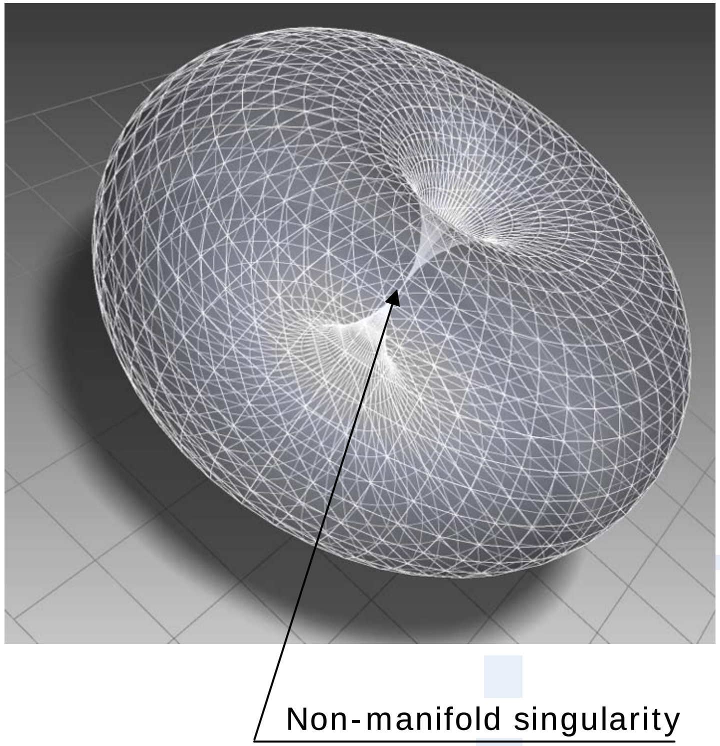
- Current objectives are set on interpreting 1-cycles from a shape point of view, ➡ possible meaning of $\beta_1$ for non-manifold objects

- Starting point is a MC-decomposition of non-manifold models [de Floriani et al. 06],
- This decomposition is unique, based on the definition of a manifold and obtained with a front propagation process,
- Here, focus is set on non-manifold objects composed of MC-components based on 2-manifold definition
- The MC-components obtained may contained non-manifold singularities at locations of the front closing simplices, i.e. vertices and edges (0 and 1-simplices)

# Shapes in 3D Euclidean space

- An elementary example of MC-component containing a non-manifold singularity





# Shapes in 3D Euclidean space

- Distinguishing MC-components, $D_i$, containing non-manifold singularities from others, $D_j$, leads to two concepts:
  - Boundary non-manifold configuration $\partial D_j$ ,
  - Internal non-manifold configuration $iD_i$ ,

- Definitions
  - An internal connection $iD_i$ takes place along a set of 0 and 1-simplices forming a 1-simplicial complex $C_i$ with:

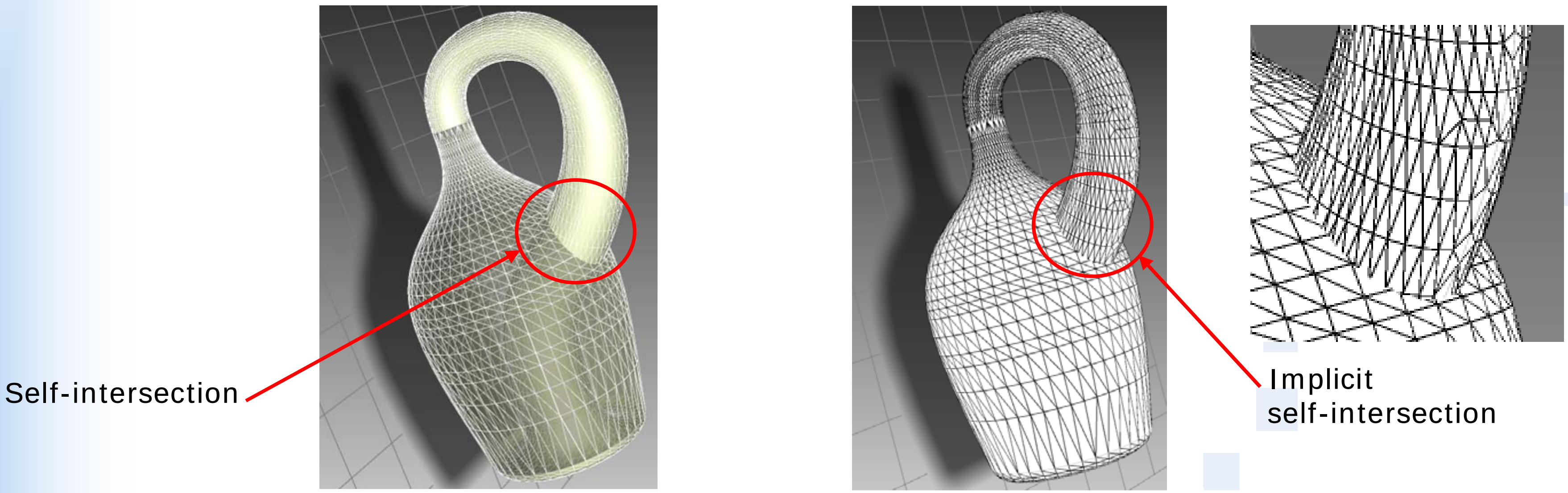
$$C_i \subset \left(D_i - \partial D_i\right)$$

Where $C_i$ has no 1-cycle.

- A boundary connection takes place along $\partial D_j$ . Here, $D_j$ is a 2-manifold with boundary.

# Shapes in 3D Euclidean space

- Objects considered can be either embedded or immersed in 3D Euclidean space,
- Objects immersed but not embedded contain self-intersections,
- Self-intersections are loosely connected to shapes whereas they reflect non-manifold configurations



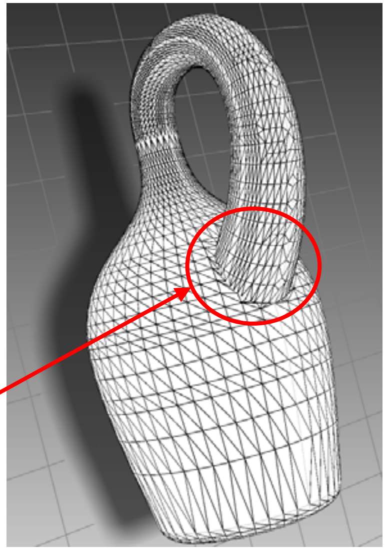
# Shapes in 3D Euclidean space

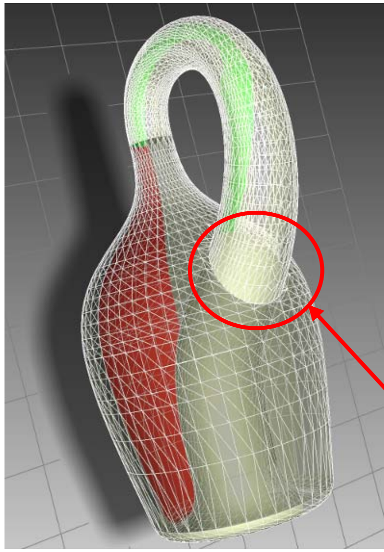
- Combinatorial topology strictly addresses 2-manifolds, hence a need to define self-intersections
- Definitions
  - Implicit self-intersection of MC-component $D_i$ is such that no topological entity (0 and 1-simplex) is located along the self-intersection (see previous example)
  - Explicit self-intersection of a 2-simplicial complex is such that every point and segment of the self-intersection matches a 0 or 1-simplex of this complex (see following example).
    - As a result, 2-manifolds immersed in $E^3$ have (explicit) self-intersections. In $E^3$ they can addressed as non-manifold objects.

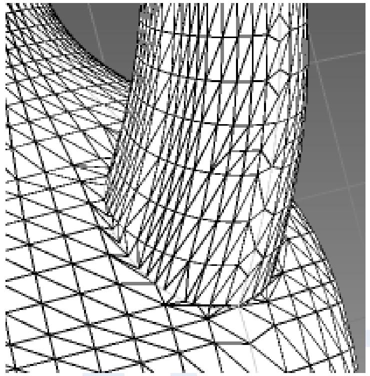
# Shapes in 3D Euclidean space

- Orientation of immersed non-manifold objects is characterized by a transition function Θ at each non-manifold connection
- If a non-manifold connection reduces to a 0-simplex, it reduces to a cone apex and Θ is undefined there

Self-intersection



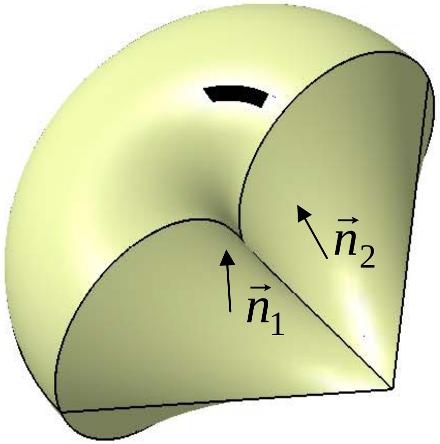




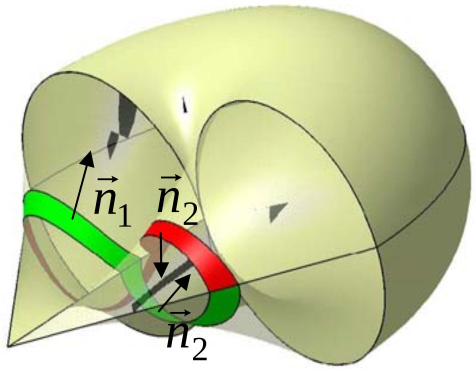
Explicit self-intersection

# Shapes in 3D Euclidean space

- An example of transition functions for two non-manifold objects




A pinched object




Cross-cap with a Moebius strip

- A strip is identical to a closed manifold 1-path, i.e. an alternating sequence of 1 and 2-simplices lying on the non-manifold object,
- A strip is characterized by its Twist: number of half turns generated when glueing its two extremities (a Moebius strip has a twist T=1)

# Shapes in 3D Euclidean space

- Identify and structure non-manifold configurations to give them shape meanings,
- Currently two working axes:
  - Define a basis for 1-cycles $\beta_1 \rightarrow f(c_1, c_2, \ldots)$ where $c_i$ are classes of 1-cycles and $c_i$ are independent of each other,
  - Characterize the categories of strips according to their twist number,
- Different non-manifold classes of objects are generated from 2-manifolds either embedded or immersed in $E^3$ through transformations up to non-manifold configurations,
- Purpose is to characterize objects that are needed for a given PDP as well as those that must be rejected depending on the application context,

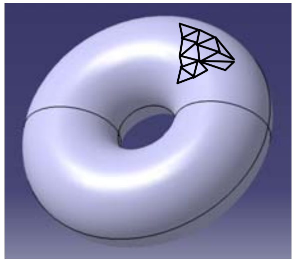
# Shapes in 3D Euclidean space

- Subset of 1-cycles categories in 2-manifolds with singularities :
  - Through holes,
  - Pinched configurations,
  - Squeezed configurations,
  - Twisted pinched configurations,
  - Surface holes,
  - Twisted holes,
  - Stitched holes,
- Start with a contribution to a taxonomy to set up classes of objects prior to establish the link with a global topological invariant,

# Shapes in 3D Euclidean space classification

- Each category is characterized by:
  - Location of non-manifold singularities: either $\partial D_j$ or $iD_i$ ,
  - Twist parameter T of the strip generated from a 1-cycle, equivalent to a transition function configuration $\Theta_i$ ,
  - A neighborhood around the non-manifold singularity that uniquely defines it

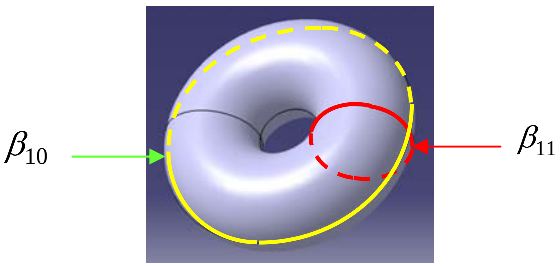
- Illustrations are based on NURBS representations to simplify images



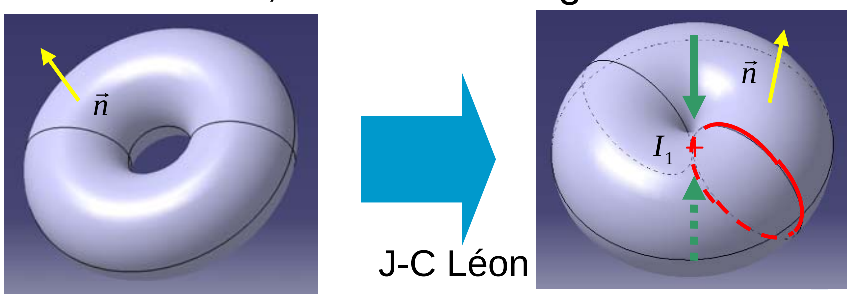
- Objects are obtain from a modeller but they are defined ‘visually’, the modeller cannot model them as effective non-manifold models

# Shapes in 3D Euclidean space classification

- Configurations with closed MC-components, i.e. class $(iD_j, iD_j)$,
- 1-cycles defining a through hole, Twist=0, all vertex neighborhood homeomorphic to 1 disk:
  - Two 1-cycles defining a through hole in a 2-manifold MC-component (reference configuration)

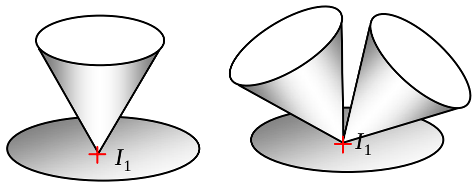


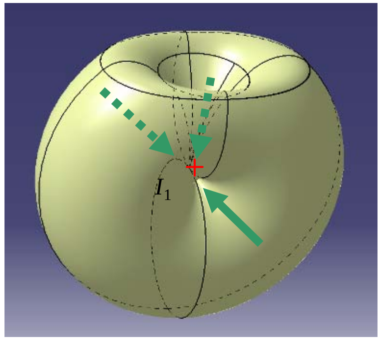

- 1-cycles defining a 'pinched' configuration:
  - One 1-cycle in a 2-manifold MC-component, pinched at point: Twist undefined, one vertex neighborhood homeomorphic to 2 distinct disks



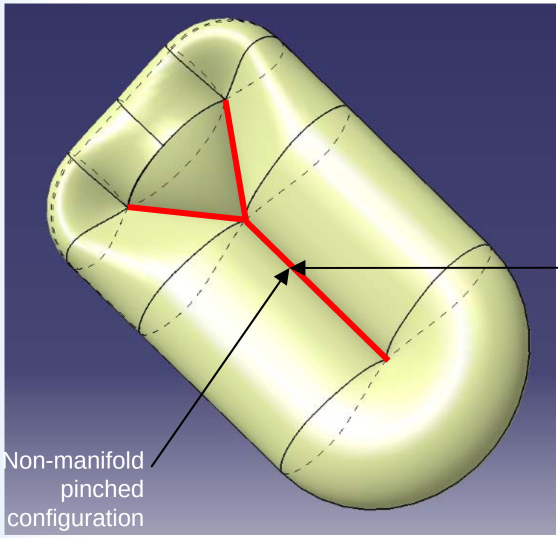
# Shapes in 3D Euclidean space classification

- 1-cycles defining a ‘pinched’ configuration
  - Class ‘pinched’ configuration is assigned a parameter, the arity $a_p$ (nb of disks connected together)




  - Class ‘pinched’ incorporates non-manifold connections without 1-cycles (in the description of the connection) as a generalization of this configuration

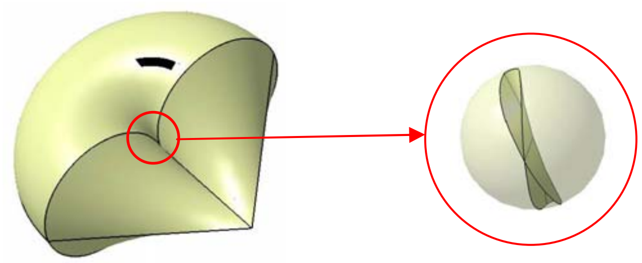



Example with Twist = 0

Non-manifold connection without 1-cycle

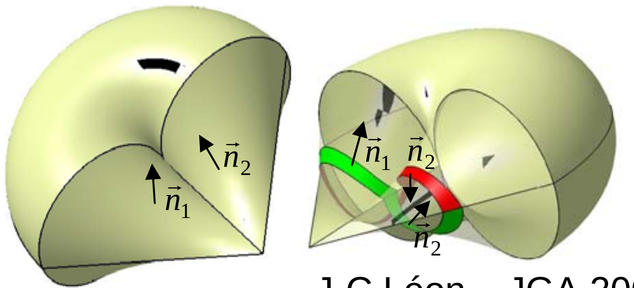
# Shapes in 3D Euclidean space classification

- ‘pinched’ configuration without generation of 1-cycle
  - Class ‘pinched’ with a neighborhood of non-manifold configuration homeomorphic to 2 tangent disks (or cones), Twist = 0,



Non-manifold singularity neighbourhood homeomorphic to two tangent cones

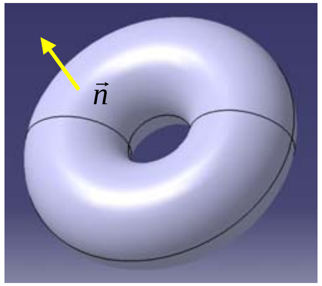
- 1-cycles defining a ‘pinched’ configuration
  - Class ‘pinched’, neighborhood of non-manifold configuration with 2 tangent cones, Twist =1,

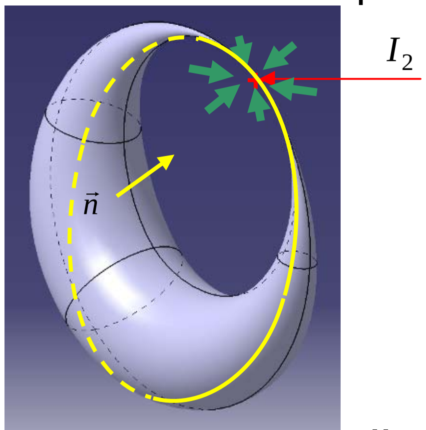



J-C Léon – JGA 2009

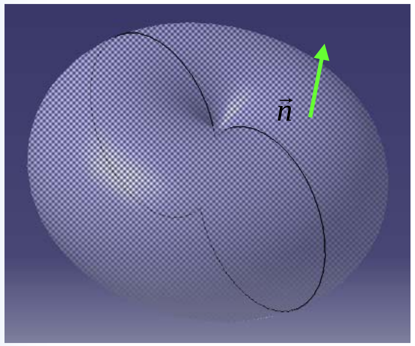
# Shapes in 3D Euclidean space classification

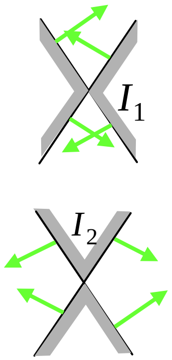
- 1-cycles defining a ‘squeezed’ configuration
  - One 1-cycle in a MC-component. Here also the concept of arity $a_s$ exists but it appears only with transformation of $a_s$- torii

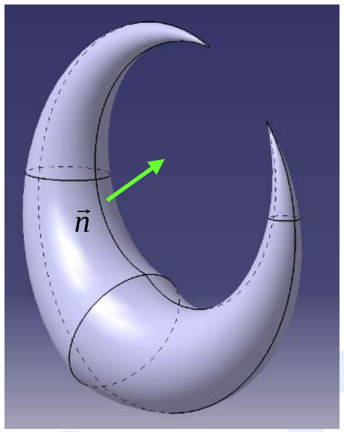



- Squeezed and pinched configurations are topologically equivalent




A reference orientation must be used to categorize the different configurations

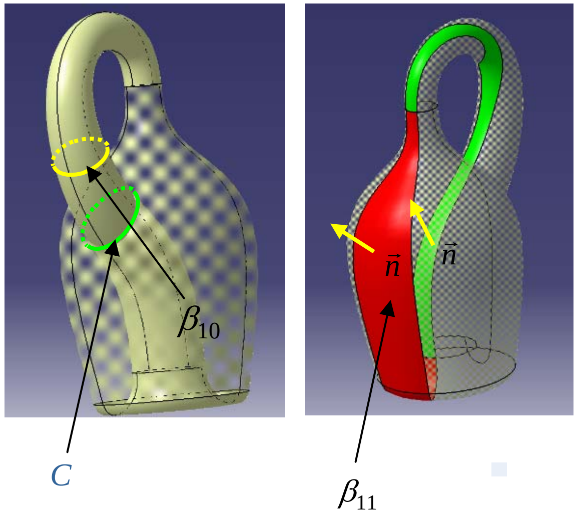
# Shapes in 3D Euclidean space classification

- Squeezed and pinched configurations at point
  - Non-manifold singularity neighborhood homeomorphic to 2 distinct disks,
  - Twist undefined,
  - Their distinction needs a common reference orientation,

- Similarly to pinched configurations, squeezed ones can be generated with non-manifold singularities along a 1-complex with Twist=0 or 1

# Shapes in 3D Euclidean space classification

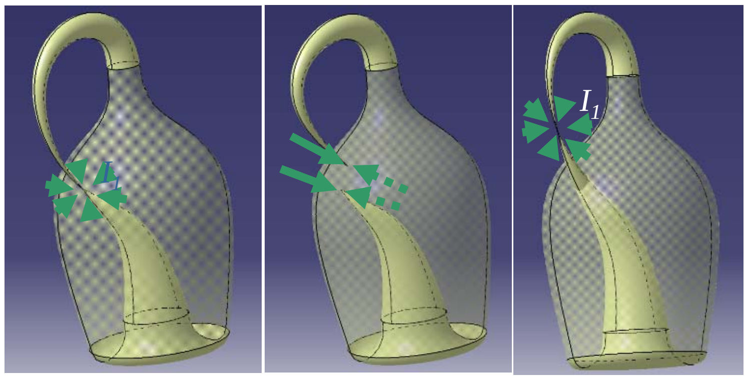
- Combination of 1-cycles with Twist = 0 and 1, explicit self-intersection
  - Klein bottle as reference for transformation, Twist = 0 ($\beta_{10}$), Twist = 1 ($\beta_{11}$),
  - Explicit self-intersection leads to 2 MC-components,
  - Configuration with MC-components with boundary, i.e.$(\partial D_i, \partial D_j)$.



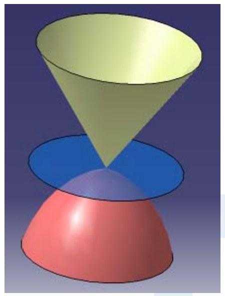

J-C Léon – JGA 2009

# Shapes in 3D Euclidean space classification

- Squeezed and pinched configuration:
  - One 1-cycle,
  - Self-intersection reduced at a point,
  - Neighborhood of non-manifold configuration, homeomorphic to 3 disks with 2 nested ones,
  - Twist undefined at singularity,
  - Configuration with MC-components $(iD_j, iD_j)$,



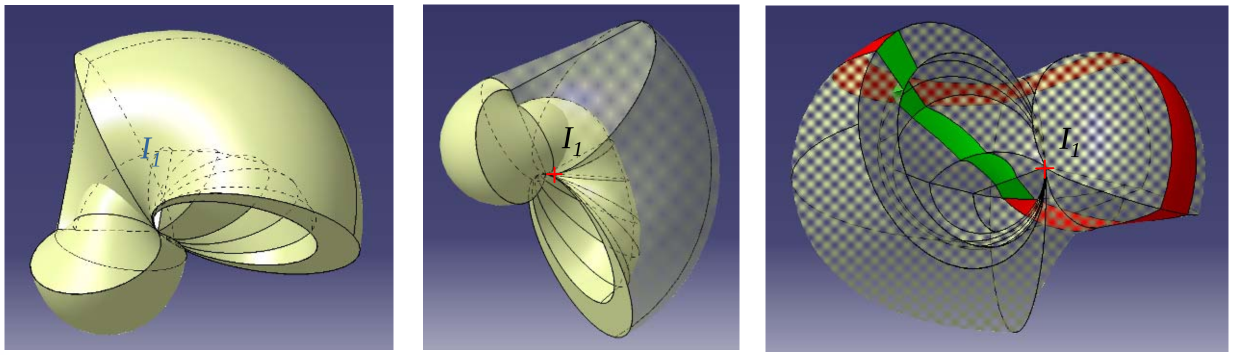

J-C Léon – JGA 2009



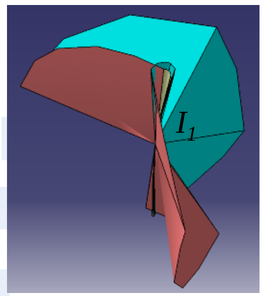
Non-manifold singularity neighbourhood homeomorphic to three cones with two nested ones

# Shapes in 3D Euclidean space classification

- Twisted and pinched configuration:
  - Configuration with several MC-components with boundary, i.e. $(\partial D_i, \partial D_j)$,
  - A 1–cycle with Twist = 1,
  - A point neighborhood with a non separable structure

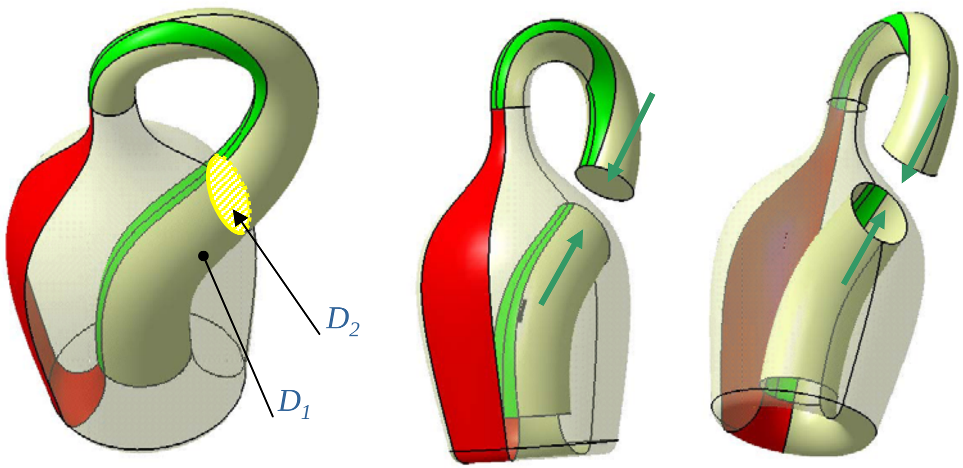



Non-manifold singularity neighbourhood



# Shapes in 3D Euclidean space classification

- Illustration of the transition function variants:
  - Klein bottle, a transition with Twist = 0 ($\beta_{10}$), Twist = 1 ($\beta_{11}$), a 2-manifold immersed in $E^3$,
  - A torus with stitched protrusion, Twist = 0 ( $\beta_{10}$ and $\beta_{11}$ ), a non-manifold object embedded in $E^3$



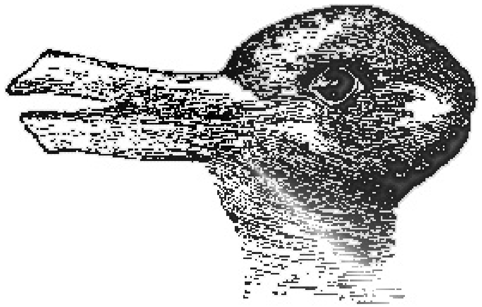

Klein bottle

Torus with stitched protrusion

J-C Léon – JGA 2009

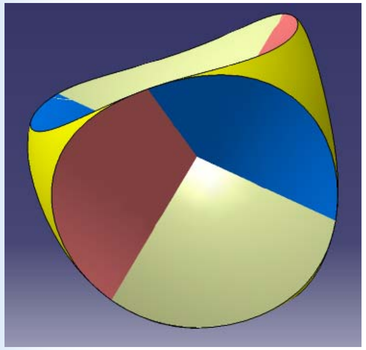
# Shapes in 3D Euclidean space classification

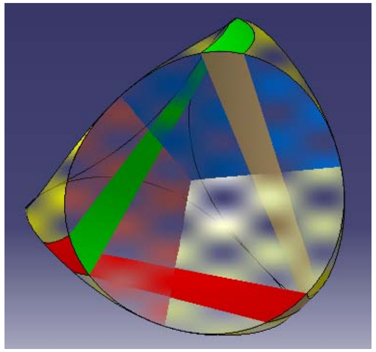
- Shape variants generated with different transition functions act as 3D optical illusion !!!

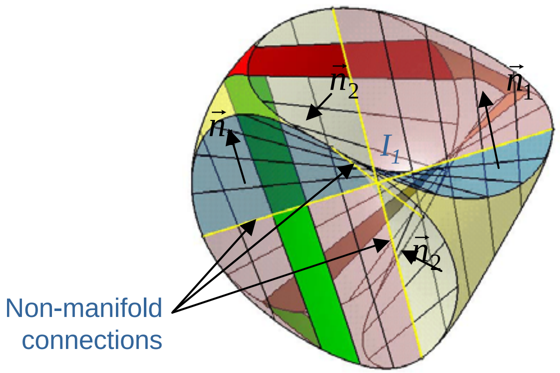


# Shapes in 3D Euclidean space classification

- On the basis of strips, shape variants can be derived for each object with non-manifold singularities:
  - Projective plane,
  - Non-manifold object with Twist = 1,
  - Explicit self-intersection, configuration with 4 MC-components of type $(iD_i, iD_j)$
  - Non-manifold singularity forming a 1-simplicial complex without 1-cycle

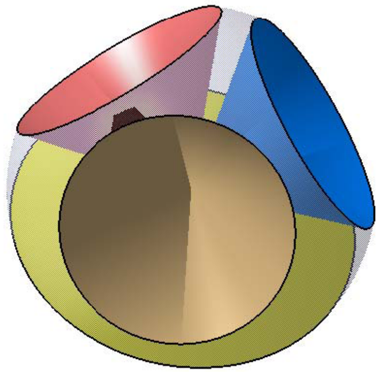


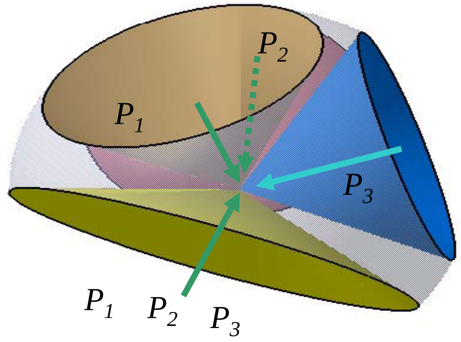




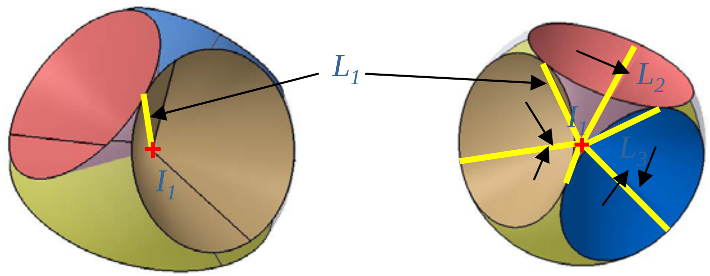
# Shapes in 3D Euclidean space classification

- Non-manifold object homeomorphic to the projective plane and generated from a sphere pinched 3 times,





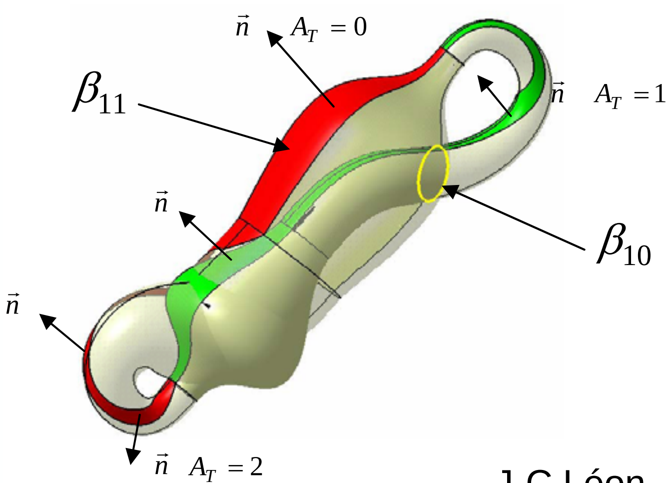

- With 4 cones made tangent generating a neighborhood homeomorphic to 4 tangent cones at one non-manifold vertex,

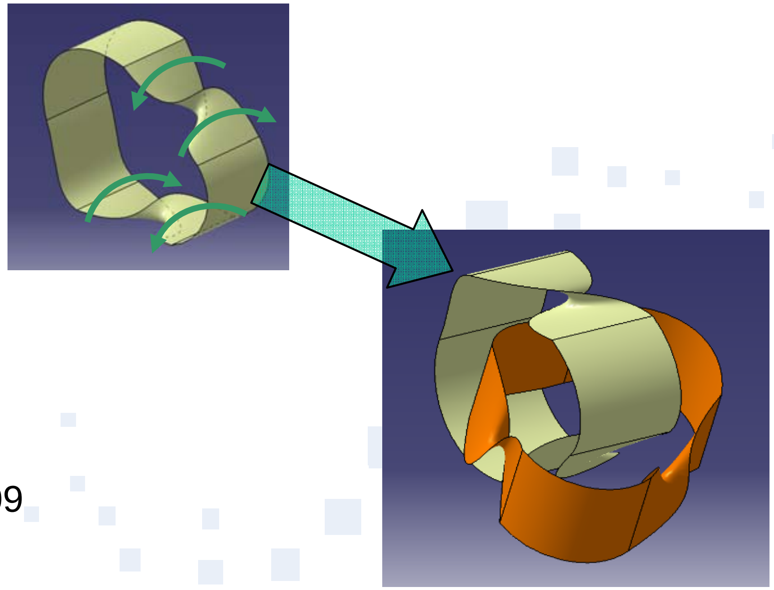



- Then, applying the same transition function as the projective plane.

# Shapes in 3D Euclidean space classification

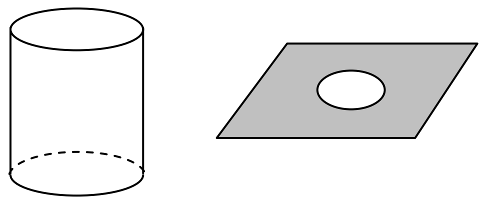
- Example of generalization of twisted strips as contribution to the classification:
  - Non-manifold object with Twist = 0 ($\beta_{10}$) and Twist = 2 ($\beta_{11}$),
  - Explicit self-intersections,
  - All neighborhoods of non-manifold configurations, homeomorphic to 2 disks,
  - Configuration with several MC-components with boundary, i.e. $(\partial D_i, \partial D_j)$



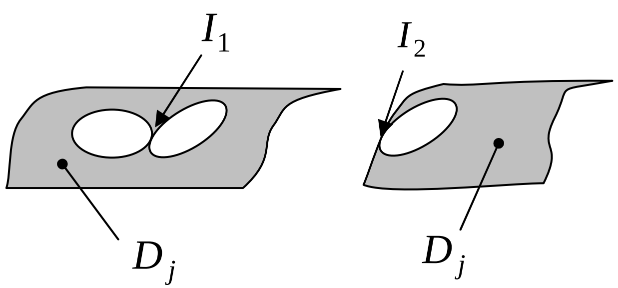



# Shapes in 3D Euclidean space classification

- Configurations with open sub-domains, i.e. class $(\partial D_j, \partial D_j)$
- 1-cycles defining surface holes
  - One 1-cycle defining a surface hole (reference configuration)



  - Non-manifold configurations still containing independent 1-cycles (stitched surface holes)

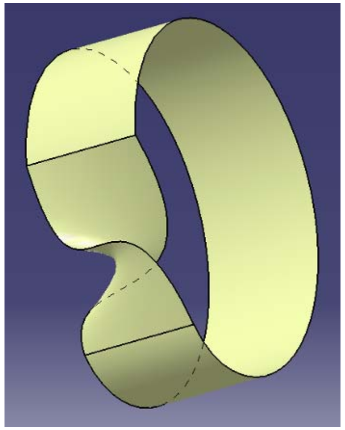



neighborhood of non-manifold configurations, homeomorphic to 2 distinct half disks only,

Twist undefined with connections at vertices only.

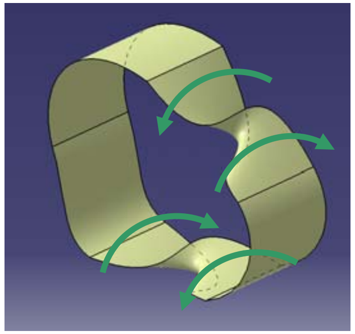
# Shapes in 3D Euclidean space classification

- Open surfaces with twisted holes (non-orientable surfaces: Moebius strip). Here again, the arity $A_T$ characterizes the number of half turns $(k\pi)$

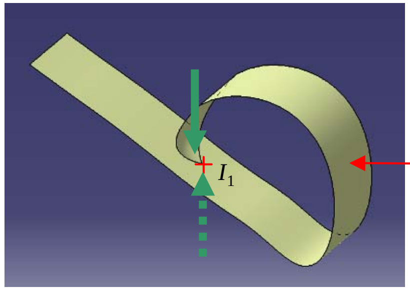


Moebius strip:
$A_T = 1$

Twisted hole:
$A_T = 2$



- Open surfaces with stitched holes, class i.e. $\left(iD_j, \partial D_j\right)$



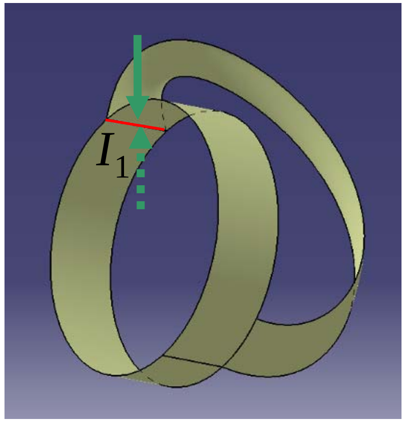

$D_j$

neighborhood of non-manifold configurations, homeomorphic to 3 or more distinct half disks,

Twist undefined with connections at vertices only.

# Shapes in 3D Euclidean space classification

- Open surfaces with stitched holes
  Stitched configurations can take place along different classes of non-manifold connections, possibly non-manifold themselves without 1-cycles




Non-orientable strip:
Twist =1

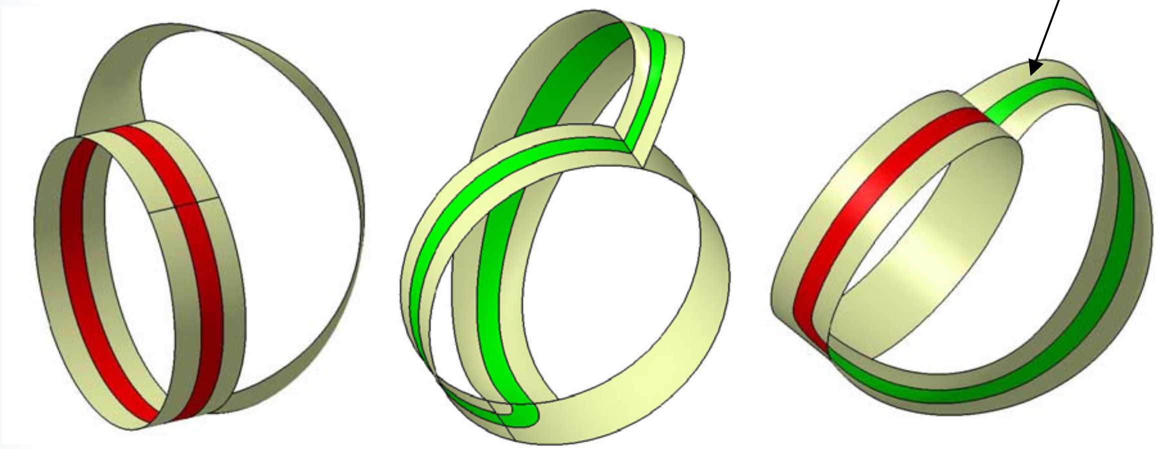


Illustration of the three categories of 1-cycles and the influence of the non-manifold connection

# Shapes in 3D Euclidean space synthesis

- Classification is partial up to now, mainly covering ‘real world’ configurations,
- Synthesis of the proposed classification

Non-manifold configurations

| One connection configuration at a time | | $(iD_j, \partial D_j)$ | $(iD_j, iD_j)$ | $(\partial D_j, \partial D_j)$ | $(\partial D_i, \partial D_j)$ | Manifold |
|---|---|---|---|---|---|---|
| All strips Twist = 0 | Orientable sub-domain | Stitched hole | Pinched Conf.<br>Squeezed hole | Surface hole | | Line hole<br>Through hole |
| Twist = 1<br>Twist = 1 and Twist = 0 | Non-orientable | | Pinched Conf. | | Twisted pinched | Twisted hole |
| Twist = 2<br>Twist = 2 and other | | … | … | … | general Characterization | … |

# Shapes in 3D Euclidean space synthesis

- The classification produces a decomposition of $\beta_1$ into a basis of manifold/non-manifold configurations. All the configurations summarized in the previous table are independent of each other,
- The classification must incorporate the orientation (twist) to characterize the whole range of proposed shapes,
- Toward the connection with the global topological invariant ?

$$v - e + f = \beta_0 - \left(N_h + N_{sh} + N_p + N_{ph} + N_{th} + N_{st} + \ldots + N_C\right) + \beta_2$$

Through holes

Surface holes

Pinched config.

Squeezed holes

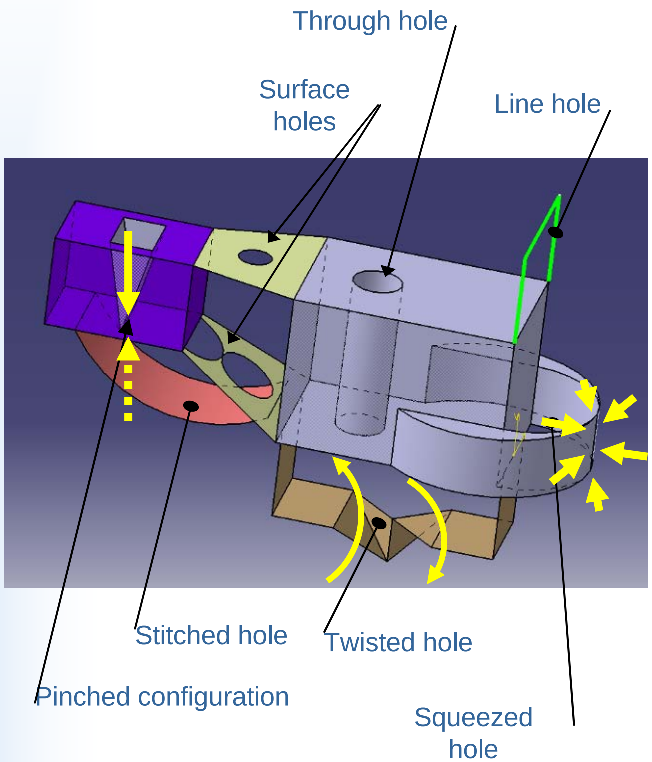
Twisted holes

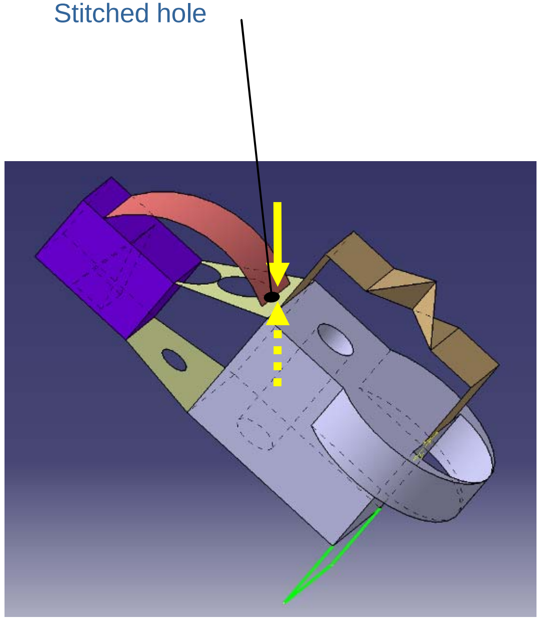
Stitched holes

Holes obtained with multiply connected MC-components

# Synthesis example of non-manifold object

A more complex example incorporating a set of MC-components




A non-manifold object -> An object with 1 through hole, 1 line hole, 1 squeezed hole, 1 twisted hole, 1 pinched configuration, 3 surface holes, 1 stitched hole

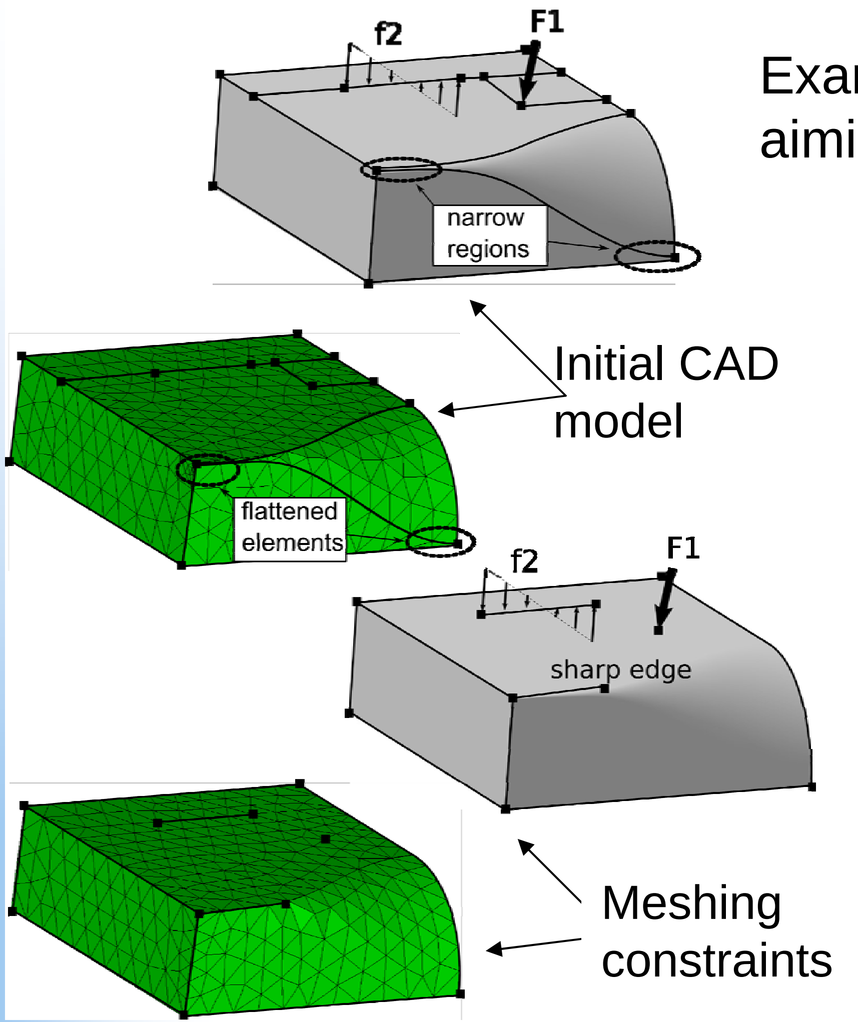
# General synthesis

- A first level of classification of non-manifold models and connection between shape parameters and 1-cycles,
- A first structure to characterize the types of non-manifold connections with interior and boundary,
- Rather exhaustive classification of connections with all strips having Twist = 0,
- Need to further characterize the twist,
- Need to refine the interaction between connection types and twist to improve the classification,
- Generalize the classification to connections among multiple MC-components,

# Shape boundary decomposition

- The purpose is to be able to produce arbitrary boundary decomposition to obtain a general framework for a wide range of PDP steps,
- To meet the application requirements to attach information to an object boundary,
- To transform an object boundary decomposition into another from one PDP step to another,
- To contribute to a more explicit representation of a design intent,
- Proposed approach based on hypergraphs (work of G. Foucault in partnership with J-C Cuillière and V François)

# Shape boundary decomposition

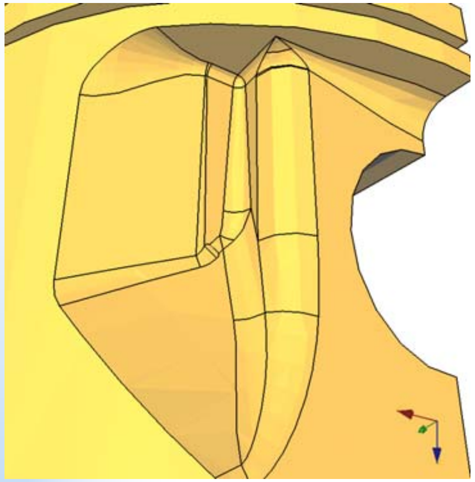


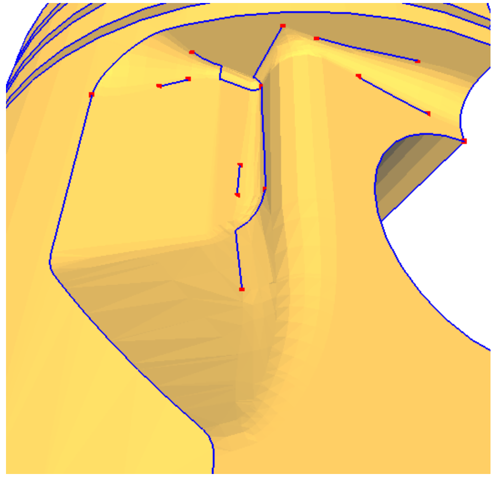

Example based on meshing constraints aiming at:

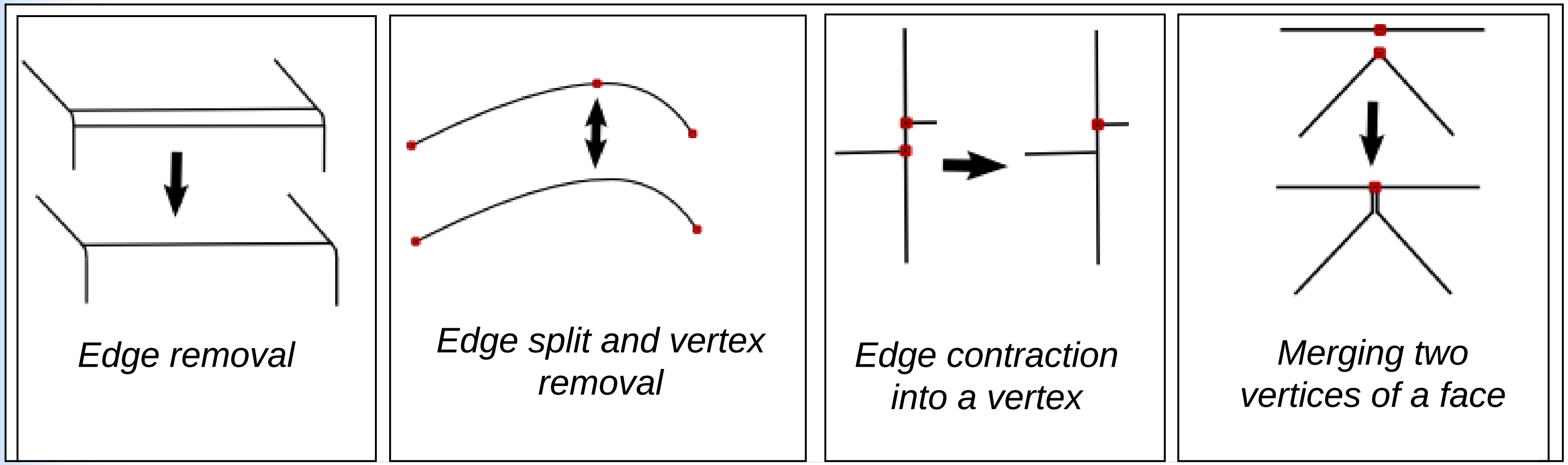
- Representing the boundary condition domains,
- Representing the relevant shape features, e.g. high-curvature locations,
- Ensuring the compatibility between the size of *Mesh entities* and the FE mesh sizes prescribed

# Shape boundary decomposition transformation

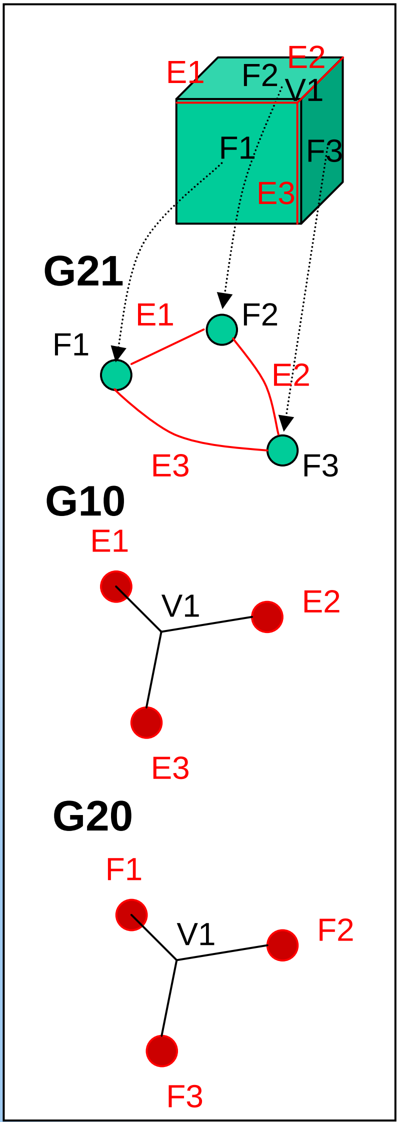
- Non-manifold boundary decomposition applied to FE meshing constraints







# Shape boundary decomposition transformation




- Hypergraphs form the description of non-manifold boundary decomposition:

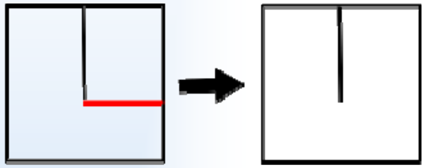
- Face-edge adjacency hypergraph (G21):
- Each vertex defines a MC-Face
- The arcs define the MC-edges linking the MC-faces

- Edge-vertex adjacency hypergraph (G10):
- Each vertex defines a MC-edge,
- The arcs define the MC-vertices linking the MC-edges

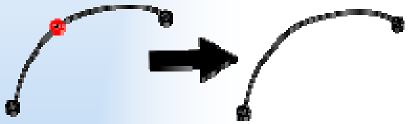
- Face-vertex adjacency hypergraph (G20):
- Each vertex defines a MC-Face,
- The arcs define the MC-vertices linking the MC-faces

# Shape boundary decomposition transformation

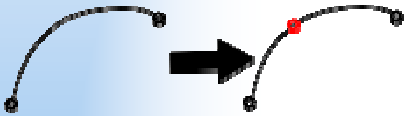
- Operators modifying the adjacency graphs of the MCT



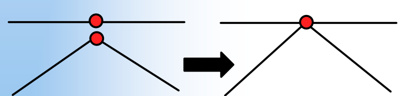
- *MC-edge removal = arc* contraction in G21, vertex removal in G10



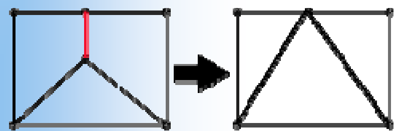
- *MC-vertex removal = arc* contraction in G10, MC-edge merging



- *MC-edge splitting*



- *MC-vertex merging of a MC-face*



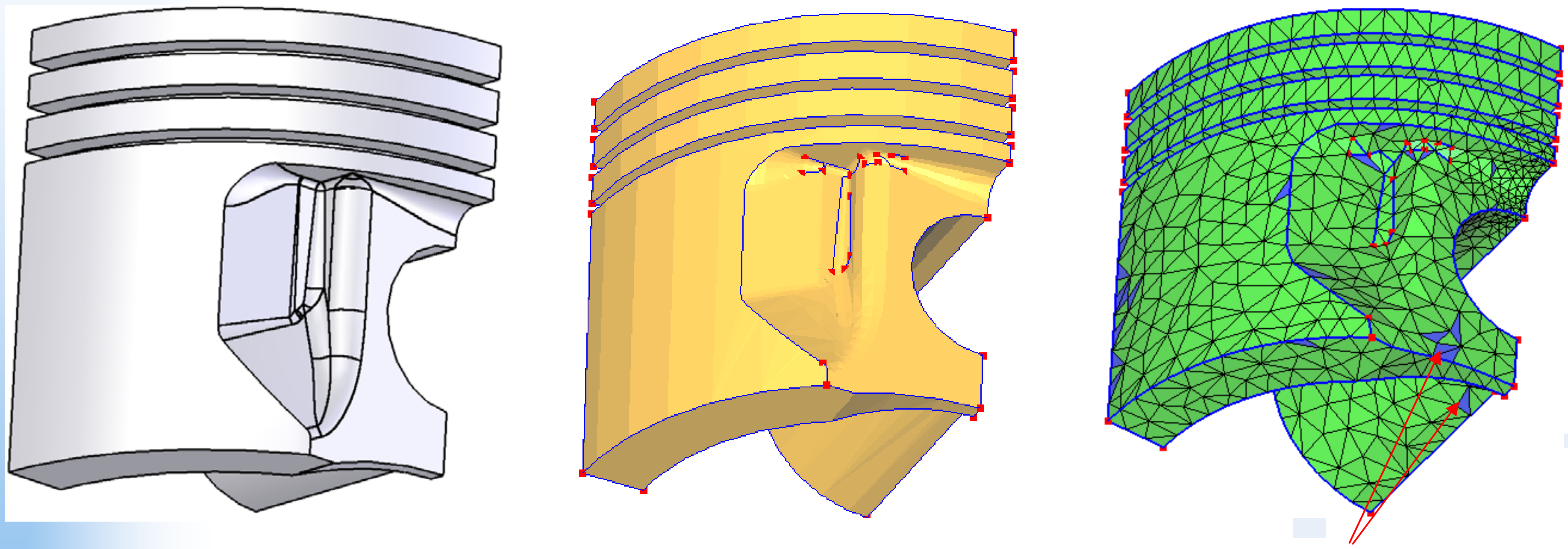
- *MC-edge contraction on its extremity*

# Criteria for MCT generation

- MCT adaptation criteria:
  - Topological conformity: edge extremities, …
  - Preservation of BCs domains
  - FE mesh map of sizes constraints,
  - High curvature constraints
- MC-edge removal criteria:
  - Local width of MC-faces,
  - Angle between faces at MC-edges,
- MC-vertex removal criteria:
  - MC-edge length,
  - Curvature around MC-vertices

# Examples of MCT generation

variable H(x,y,z)



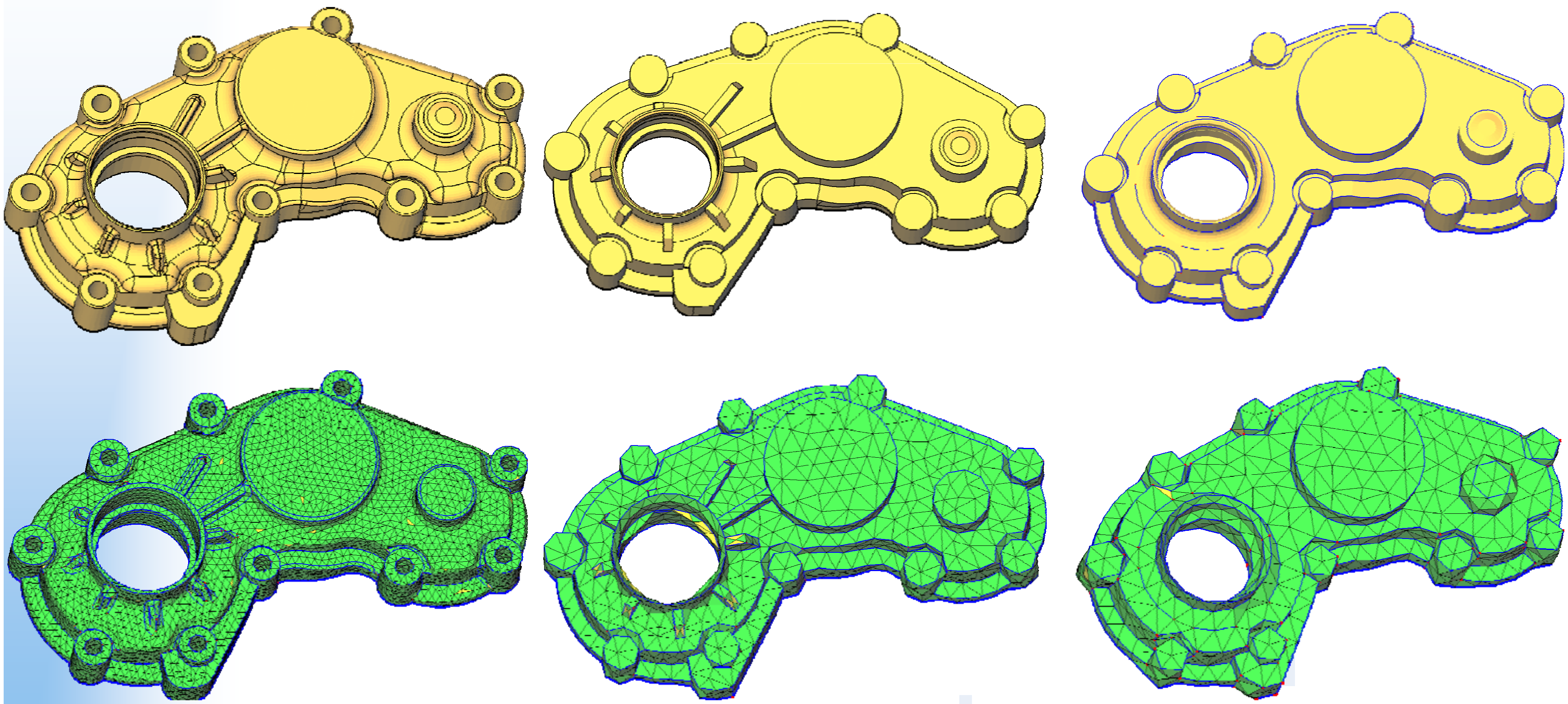

CPU time (P4@1.6 GHz) : nearly 2 minutes

# Examples of MCT generation

FE size = 4 mm

FE size = 11 mm

FE size = 15 mm



# Conclusions & Future Work

- Observation and synthesis of product development processes have led to a set of shape classes
- Topological requirements have emerged from the shape classes and their boundary decomposition to produce two major classes of requirements
- A first proposal of classification has been described that concentrates on 1-cycles and their influence on shapes in 3D Euclidean space,
- Shape boundary decomposition with non-manifold singularities has been initiated for FE applications,

- Investigate the concept of twist as a global topological parameter and contributor to a taxonomy of non-manifold models,
- Set up the hypergraph model to describe the topology of non-manifold models;
- Structure the proposed taxonomy with the minimum set of parameters and characterize each of them

- And much more … !