www.cadanda.com Computer-Aided Design AND APPLICATIONS

Managing Equivalent Representations of Design and Analysis Models

Christopher M. Tierney, Declan C. Nolan, Trevor T. Robinson and Cecil G. Armstrong

Queens University Belfast, ctierney06@qub.ac.uk, d.nolan@qub.ac.uk, t.robinson@qub.ac.uk, c.armstrong@qub.ac.uk

ABSTRACT

There is a requirement for better integration between design and analysis tools, which is difficult due to their different objectives, separate data representations and workflows. Currently, substantial effort is required to produce a suitable analysis model from design geometry. Robust links are required between these different representations to enable analysis attributes to be transferred between different design and analysis packages for models at various levels of fidelity.

This paper describes a novel approach for integrating design and analysis models by identifying and managing the relationships between the different representations. Three key technologies, Cellular Modeling, Virtual Topology and Equivalencing, have been employed to achieve effective simulation model management. These technologies and their implementation are discussed in detail. Prototype automated tools are introduced demonstrating how multiple simulation models can be linked and maintained to facilitate seamless integration throughout the design cycle.

Keywords: CAD-CAE integration, cellular modeling, virtual topology, equivalencing

1. INTRODUCTION

Pressure to design cost effective, superior products in less time has resulted in the increasing use of computational engineering analysis throughout product development cycles. An effective design process therefore requires seamless integration between Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) tools, as design decisions based on analysis results drive the design process. Many analysis iterations are used to provide optimal design decisions requiring frequent transfer between different packages. One fundamental issue regarding seamless CAD – CAE integration is the difference between design and analysis geometries. Fully featured manufacturing detailed models are used for design and simplified versions of the design model are required for different types of analysis, at different stages of the design cycle.

The processing of geometric models for computational analysis requires bi-directional links between the different geometric and analysis models in order to completely integrate the component representations in the CAD and CAE tools [1]. In the CAD-centric process, detailed CAD models containing many specific geometric features required for assembly; manufacturing etc. must be simplified to create affordable analysis models. Idealization involves approximating certain regions of a model with lower detail in order to create analysis models that are more computationally efficient [2]. Thakur et al. [19] surveys the multiple research efforts focusing on automated CAD model simplification to generate the desired simplified analysis geometry, but most of these techniques do not create the necessary associativity between the different representations.

There are many commercial and in-house design and analysis packages available to engineers. Organizations often use specialist functionality from a variety of packages to achieve the best possible solution to complex problems. The use of different packages results in multiple simulation models for the same part, with no robust link between them. Additionally, issues like tolerance mismatches, loss of feature information, along with varying underlying representations between packages can result in loss of model integrity. These problems are enhanced in large collaborative projects with multiple partners interacting at various stages of the design cycle, analyzing different physics domains, with models at various levels of detail and abstraction. Therefore, seamless integration must not only overcome the differences between design and analysis geometry representations but also the problems introduced through the use of different CAD and CAE tools.

Several attempts have been made to achieve CAD-CAE integration. Sypkens Smit and Bronsvoort [18] discuss the integration of an analysis view as part of multiple-view feature modeling. Lee [10] describes the use of a single non-manifold master model to store all form features along with their appropriate idealizations allowing models to be created at various levels of detail and abstraction. Shephard et al. [17] describes a simulation model manager to construct analysis models, without actually linking the models. Gujarathi and Ma [8] introduced a common data model (CDM) as a way to integrate CAD and CAE models at a parametric level.

While each approach makes an important contribution, these approaches do not provide a generic solution able to utilize design or analysis geometry at any stage of a CAD or CAE centric design process, whilst still providing the bi-directional associativity necessary for successful CAD-CAE integration. In this work the associativity between all model representations is stored, maintained and re-used in a robust manner, allowing simulation models to be re-built without compromising the integrity of a model. It is demonstrated how three core technologies named Cellular Modeling, Virtual Topology and Equivalencing are used to manage and manipulate the topology of geometric design and analysis models in a coherent, integrated fashion, independent of underlying CAD or CAE systems.

2. THREE CORE TECHNOLOGIES FOR SIMULATION MANAGEMENT

2.1. Cellular Modeling

Cellular representations are non-manifold geometric representations of both positive and negative spatial regions [4]. In manifold representations any point on the boundary of a solid region has a neighborhood homeomorphic to a 2-dimensional disk [20]. Geometric regions that are not manifold are referred to as non-manifold. For example, in a manifold representation a face can only bound one body and an edge can bound a maximum of two faces. In a non-manifold representation a face can bound two solid regions and an edge may bound any number of faces. A merged set [7], which is similar to a cellular model, is used to store the single boundary representation of all primitives. The merged set contains a description of all primitives, their interactions and information specifying the origin of entities in relation to the topological entities of the original primitives.

Here the concept of Cellular Modeling is expanded upon, with the entire design space being partitioned into cells of specific analysis significance. The non-manifold cellular model consists of both structural and fluid cells, Fig. 1 (a). For example, cells may be created to allow different analysis attributes (e.g. meshing styles) to be applied to different regions, or for defeaturing purposes cells could represent features essential for design but superfluous for all but the most detailed analysis. Cells can be utilized differently in different analyses. For example, consider a cellular model consisting of a cell representing a small "hole" in a body, Fig. 1 (b). In a preliminary global structural analysis it may be appropriate to consider the cell as solid material. Merging this cell with its adjacent structural cells means that the finite element mesh does not have to respect the small feature. In a fluid analysis it may be essential to consider the same cell as fluid and merge it with its adjacent fluid domain.

In this work it is demonstrated how queries on the topology of a non-manifold cellular model can be used to identify interfaces between interacting cells. Every cell in the cellular decomposition contains information about its originating cells. The interface and individual cell information can be used to link multiple analysis models with the design model.

2.2. Virtual Topology

Due to the difference in requirements between design and analysis models, it is often necessary to modify the design model in order to produce suitable analysis models. Sometimes additional topology is required, either to identify the subset of a face where loading is to be applied, or for applying contact between two parts. In other circumstances analysis models can be created by removing features that are not of interest for a particular analysis. For example, sliver faces or design features like fillets and chamfers may have no effect on the analysis of interest, but may

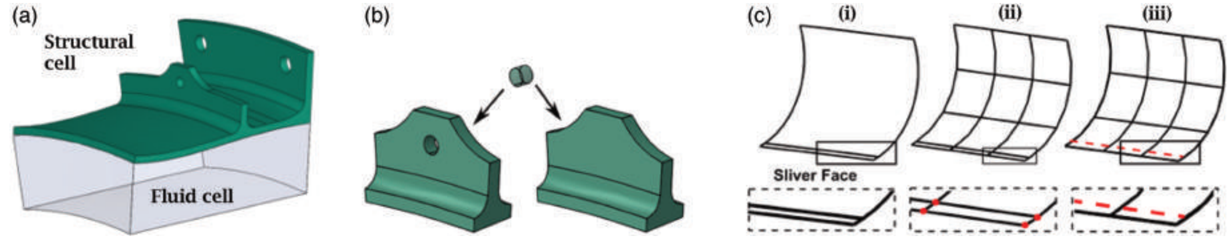

Fig. 1: (a) Cellular model with structural (dark) and fluid (light) cells, (b) Subtractive cell considered fluid or structural depending on the application, (c) Meshing a sliver face.

increase the number of nodes in the analysis mesh, and decrease its efficiency. Direct modification of a model to remove such features is a complicated procedure, which can produce many undesirable results and break the ties with the original model [3]. The user interaction required to directly modify the CAD geometry counteracts the desired automation of the analysis process. The level of success is also heavily dependent on the scale of change and the skill of the engineer implementing it.

Sheffer et al. [16] introduced the concept of Virtual Topology to remove the need for direct geometry editing. Virtual Topology makes use of real topological entities to create virtual entities required for analysis purposes. Virtual superset entities reference multiple entities that are merged together, while virtual subsets reference a section of an entity that has been split into multiple parts. In this work virtual subsets and supersets are used to create multiple representations from the same cellular model without affecting the underlying CAD geometry. Virtual entities can still reference the original geometry.

Mesh generation is one application for Virtual Topology. When generating a mesh on a model, nodes are distributed along all of the bounding entities in the model. If some of the bounding faces in the model are small (so-called sliver faces Fig. 1 (c) (i)) in comparison to the target element size, the mesh must respect the sliver face, creating elements which can be poor quality and inefficient in the analysis, Fig. 1 (c) (ii), where the element nodes for the sliver face are in red. One solution is to merge sliver faces with adjacent larger faces creating superset entities, Fig. 1 (c) (iii). When merging two entities using Virtual Topology, the common edge is effectively ignored (red dashed line Fig. 1 (c) (iii)), therefore nodes are not placed on the common edge between the two faces, or its bounding vertices, and poor mesh quality is avoided.

Some CAE packages already have Virtual Topology capabilities. The issue is that existing CAE packages make Virtual Topology decisions internally, and do not report the details of the operation to the user. In this work Virtual Topology is recorded in an open, transparent way, allowing it to be used by the analyst and be available from multiple packages. This capability allows design and analysis models to be linked to one another and use the same Virtual Topology, without the constraint of the analysis details having to conform to the original CAD model topology.

2.3. Equivalencing

Throughout the design evolution many different representations of the same object are required for different applications. For the example in Fig. 2 it is shown how a given cell in a model may be represented as a 3D cell, 3D cell with small features suppressed, 3D mesh or mesh of 2D shell elements with a thickness attribute. Here these representations are considered equivalent as they all represent the same cell.

Different types of equivalencing are required to successfully link the different models that may be used during the design cycle. One example is equivalencing an idealized simulation model to its more detailed representation. This equivalencing information can then be exploited for different purposes, like mapping results from an efficient dimensionally reduced analysis back to the detailed geometry.

Other forms of equivalencing include linking a non-manifold model to its equivalent manifold representation. Non-manifold entities can be equivalent to multiple manifold entities, requiring the relationship to be created and maintained between the different entities. In this work Virtual Topology is viewed as a form of equivalencing, where entities created using virtual supersets and subsets are linked to their equivalent entities in another representation. Equivalence relationships are not well defined, or are not defined at all, in current CAD/CAE packages. Equivalencing is required to maintain consistent models at different levels of detail and dimensionality in different software packages.

3. SIMULATION MODEL MANAGEMENT

The integration of design and analysis models is a challenge. Analysts use a number of specialist tools

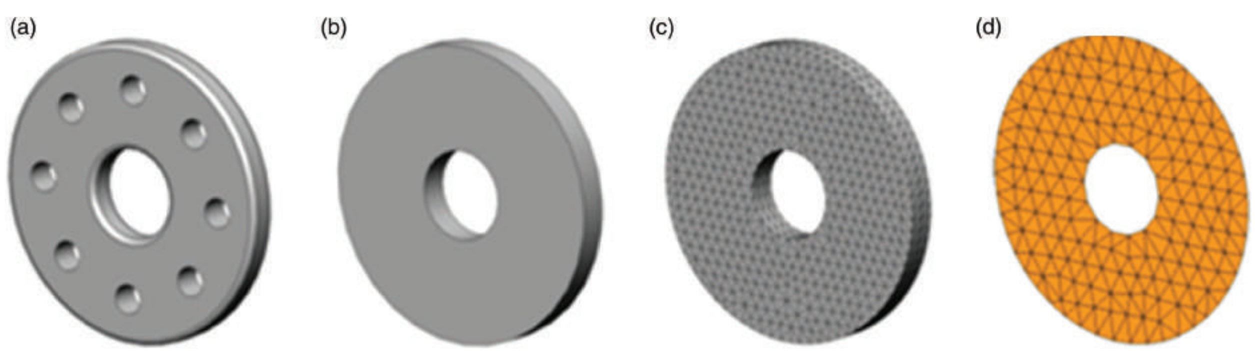

Fig. 2: Equivalent representations: (a) Detailed design representation, (b) Abstract analysis model, (c) 3D solid mesh on abstract model, (d) 2D shell mesh applied to mid-surface.

System	Typical Task	Non-manifold	Name Attributes		Accuracy (mm)
			Body	Topology	
Siemens NX	Design/Analysis	No	Yes	Yes	10^{-8}
CADFix	Decomposition / Geometry repair	Yes	Yes	Yes	10^{-12}
Parasolid	Interrogation	Yes	Yes	Yes	10^{-8}
Abaqus	Mesh Generation	Yes	Yes	No	10^{-6}

Tab. 1: Attributes of the design and simulation tools used to demonstrate simulation integration.

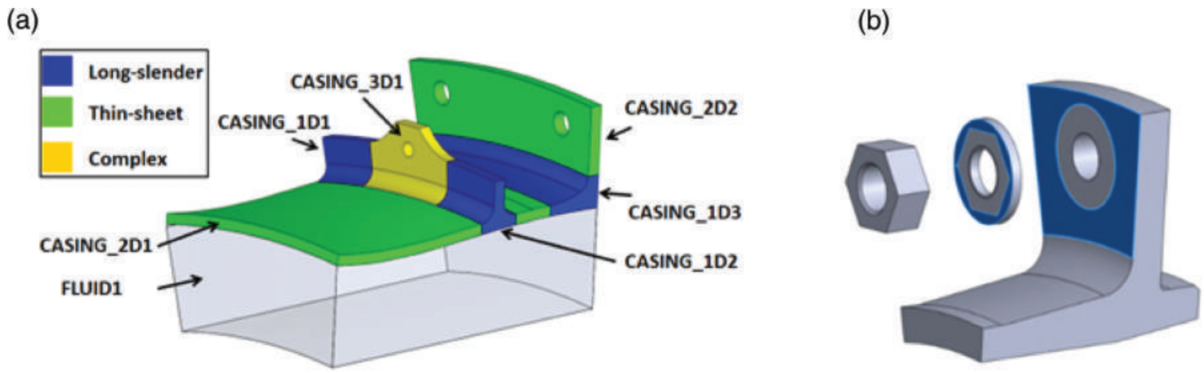

Fig. 3: (a) Decomposed cellular model, (b) Automatic imprinting by Boolean Union operation.

from different vendors for different tasks during the creation and processing of an analysis model. A summary of some key aspects of a number of commercially available design and analysis tools used in this work is shown in Tab. 1. Each tool is used for a specific task in the examples presented in this section. Whilst packages like Siemens NX contain design and analysis tools within one environment the underling architectures of these tools may be different.

Whilst the procedures described in this work can be used with any CAD and CAE tools, the simplified analysis model used for demonstration in this work was created by a CAD model simplification tool used to automatically decompose a design model into thin-sheet and long-slender cells which can be efficiently meshed [11,13]. The decomposed model of a section of an aero engine casing is shown in Fig. 3 (a). It consists of multiple cells, each of which can be utilized for different analysis types and therefore has specific analysis significance. The original model is referred to as the parent of the decomposed cells. Name attributes assigned to decomposed cells during the decomposition process define the parent entity and simulation significance of a cell. For the example in Fig. 3 (a), one manifold body has the tag 'CASING_2D2'. Its parent entity is 'CASING', which is the name of the original cell. Its shape type is '2D' which signifies that it is a thin-sheet region which the decomposition tool has identified as suitable for idealization as a mid-surface shell model for analysis purposes.

The main focus of the approach is to use the three core technologies in Section 2 to automatically link the various geometric representations required for design and analysis in different packages. In order to successfully achieve this goal a flexible integrated data structure has been proposed. All detailed and abstract versions of the same model can reside within this data structure, allowing links to be identified and used to transfer analysis attributes between different models. Automated prototype tools have been developed to demonstrate how the proposed data structure manages the information required to link models at various levels of abstraction. The data structure and tools interact with multiple CAD and CAE tools (Parasolid, Siemens NX, Abaqus and CADFix) maintaining consistency between different model representations. By way of example, the decomposed model described above is used to demonstrate how this approach can be used at any stage of a design process to link the decomposed analysis model and its attributes to the original design model and other analysis models. It also lends itself to demonstrating multi-dimensional aspects, as the same model is also suitable for automated dimensional reduction and meshing [12].

3.1. Non-manifold Cellular Model

The process begins with a design model that has been partitioned into an assembly of bodies with specific analysis significance. It is then necessary to convert this manifold assembly to a non-manifold cellular representation. This model can then be used to identify interfaces between different interacting cells.

During the creation of the cellular model the entire design space is partitioned in order to also create fluid domains where no structural components exist. The bounds of a model's design space are used to split the infinite void region of the model, creating manifold bodies representing the fluid domains. Once all manifold bodies have been created Parasolid Boolean union operations are used to convert the manifold assembly to its non-manifold cellular representation. One manifold body is arbitrarily selected as the target body with the remaining manifold bodies provided as the tool bodies. The first step of the process imprints the target and tool bodies onto one another. After the imprinting a single non-manifold body is created by combining the face sets from all target and tools bodies. Faces coincident to two manifold bodies in the manifold assembly are now represented by a single non-manifold face shared by two cells in the non-manifold model. The bounding entities (edges and vertices) of the coincident faces are also merged so they are shared by the adjacent cells. Adjacencies in the non-manifold representation are identified by finding two or more cells which share the same bounding entity. The resulting non-manifold representation consists of a single non-manifold body comprised of multiple adjacent cells representing structural and fluid regions.

Due to the differences between the manifold and non-manifold representations, the position and identity of cells is transferred between the representations to ensure valid models are created. Cells in the non-manifold model are related to their equivalent manifold body. Applying the name attribute of a manifold body to its solid region in the manifold representation enables the cell to be identified in the non-manifold representation. Assigning instance transformations to bodies in the manifold assembly enables cells in the non-manifold model to retain their correct assembled position.

The non-manifold Boolean Union automatically imprints faces in proximity to one another. This allows boundary conditions to be applied to imprinted faces. These imprinted faces are complete descriptions of interfaces between interacting cells. One application might be applying contact conditions between the nut, washer and flange cells shown in Fig. 3 (b). As a result of the imprinting a hexagonal face exists on the washer which can be used for such an application. Ordinarily, these imprints would be created manually by the analyst, which is a time consuming process.

Assembly models of multiple parts can be treated as a non-manifold model. In this case the sub-regions of all the parts in the model will be represented by individual non-manifold cells, with the relationship between the cells and the original part models stored. There are some implementation issues with this step which have been discovered and are detailed in the discussion.

The non-manifold cellular model is used to integrate the various analysis representations with the original design model. This is achieved by extracting the non-manifold topology from the CAD / CAE environment and storing it in a relational database. Topological adjacencies in the non-manifold model can then be identified and used to transfer analysis attributes between simulation models at different levels of detail.

3.2. The Data Structure

In this work a relational database is used to store the topological connectivity information of the non-manifold cellular decomposition, Fig. 3 (a). A relational database stores data in relations, with each relation consisting of certain number of attributes [6]. Its advantages are the ease with which bespoke queries can be used to retrieve associations between different representations of the model, and that it is CAD / CAE package independent. The need for a simple data structure that can be used to integrate CAD models for simulation needs was recognized by Hoffman and Joan-Arinyo [9]. While neutral CAD formats are available for transferring CAD models, a similar robust neutral format is unavailable for CAE models. The database discussed here is not being proposed as an alternative to existing complex data structures used to represent geometric models [14], but for managing equivalent representations required between design and analysis domains. The database presented here is capable of maintaining a complete set of topological adjacencies.

3.2.1. Extracting the non-manifold cellular topology

The boundary representation (B-rep) of a CAD model describes the connectivity between all topological entities. Every topological entity in a model is related to a corresponding geometric entity. Therefore, storing the topology of a model effectively stores the link to geometric entities as well. Due to underlying architectures topological entities differ between packages. For example, Parasolid uses topological entities called fins to represent the use of an edge by a face, while the ACIS modeling kernel (the underlying kernel for Abaqus) uses topological entities called co-edges for a similar purpose. Fins and co-edges allow an edge to bound more than two faces, enabling non-manifold conditions to be successfully represented. Such entities do not exist in a manifold modeling environment. This approach stores only the main topological entities which are required to represent a model and are common to the different packages i.e. vertices, edges, faces and 3D cells (may be referred to as bodies/regions or lumps in different environments).

Label	Dimension	Identifier
1	3	(x, y, z)
2	3	(x, y, z)
3	2	(x, y, z)
4	1	(x, y, z)

Entity	Bound Entity	RelOr
1	3	+1
2	3	−1

Entity	Entity Name
1	CASING_2D2
2	CASING_1D3
3	CASING_2D2_3
3	CASING_1D3_3
4	CASING_2D2_4

Tab. 2. Sample Relations: (a) Entity relation, (b) Topology relation, (c) Manifold relation.

These main topological entities along with their relative orientations to their bounded entities are sufficient to allow the adjacencies to be found in manifold or non-manifold representations. This work uses these adjacencies to locate interfaces between interacting non-manifold cells in order to link design and analysis models. If the need arose, these adjacencies could also be used to identify application specific entities like fins and co-edges. This ensures this approach is independent of the packages used.

Once the non-manifold topology has been created its topology is extracted and stored in the database. Standard data structure queries are used to automatically extract the connectivity information for all topological entities in the non-manifold model. Only the method of extracting the topological information from each package differs.

Each entity in the geometric model is stored in the Entity relation, Tab. 2 (a). The Entity relation has three attributes: ***Label:*** the unique label of the entity within the database, which is assigned as the primary key. Any attribute in other relations using the Label is designated a foreign key. This ensures every new entity created in the database must exist in the Entity relation first. The Label attribute contains the unique tag for an entity in the database. This is used instead of internal CAD ids which will differ between systems and sometimes even between different sessions in the same system. Having unique tags for entities enables new entities to be created independently of the geometric model. ***Dim:*** the manifold dimension of each entity. Vertices = 0; edges = 1; faces = 2 and structural / fluid cells = 3. ***Identifier:*** contains a Cartesian point lying within the boundary of a topological entity. The Identifier is the geometric position of a vertex; mid-point of an edge; point lying on the surface of a face; point inside a cell. Due to the nature of the non-manifold model two entities of the same dimension will never have the same Identifier. The Identifier attribute stores the information required to associate entities in the database to their equivalent geometric entities in the CAD environment. This allows geometry to be manipulated using information in the database.

The topology of a model (which may be different for different representations) is stored in the Topology relation, Tab. 2 (b). The Topology relation has three attributes which can completely define the topological connectivity of a manifold or non-manifold model: ***Entity:*** the Label of an entity in the model which is bounded by another topological entity. ***Bound Entity:*** specifies the label of an entity that forms the boundary of the entity in the Entity attribute. ***RelOr:*** stores the relative orientation of the bound entity to the entity in the Entity attribute. The relative orientation attribute of a face bounding a region is positive if the underlying surface normal is pointing out of the region. Tab. 2 (b) shows that the common face (3) between two body cells is positive relative to one body (1) and negative relative to the other (2). An edge has a positive orientation relative to a face if the direction of its underlying curve is anticlockwise around the face when viewing from the direction of its outward pointing face normal. A vertex has a positive orientation relative to its edge if the tangent of the underlying curve at the vertex points out of the edge.

Tab. 2 (c) shows the Manifold relation, which is used to link the non-manifold topology stored in the database to its equivalent manifold representation. The manifold representation can then be used by CAD / CAE packages which do not support non-manifold topology. This relation stores the name attribute of each topological entity and provides a convenient method for linking the topology in the CAD environment to its equivalent database entity. The name attribute consists of the parent body joined with the database label of each topological entity. An SQL query automatically creates the name attribute for each entity using the topology relation and database label to ensure unique names are assigned. For the example shown in Fig. 4 (a), the edge bounding one body cell is assigned a single name attribute, CASING_2D2_4. However, non-manifold entities have multiple names assigned based upon their connectivity, reflecting the fact they are equivalent to multiple entities in the manifold model. The common face, Fig. 4 (a), bounding two body cells is assigned two different name attributes, Tab. 2 (c), referring to each bounded region.

Designating name attributes in the manner described stores the link between manifold and non-manifold representations, allowing interfaces to be calculated for a non-manifold representation and used within a manifold CAD / CAE environment. For analysis this facilitates the transfer of boundary conditions across domains or the mapping of results from one cell to another. The Manifold relation enables manifold modelers to interact with the automated process described here.

3.2.2. *Managing Virtual Topology*

The concept of Virtual Topology was discussed in Section 2.2. Decoupling the topology from the CAD model allows it to be manipulated for analysis purposes without affecting the geometry and topology of the original CAD model. Virtual Topology is used to define new cellular representations using subsets and supersets of the existing decomposition. Virtual Topology can be created by different CAE packages. The results of the Virtual Topology created by the package can be written to the database allowing other CAE packages to access, and make use of this information. The Virtual Topology relation, Tab. 3 (a) is used to store the relationship between original host entities and any new virtual entities. This relation has two attributes: ***Entity:*** database Label of any new virtual entity created due to a merge or split operation. ***Host Entity:*** database Label of the host entity on which the virtual entity relies on.

Entity	Host Entity
CASING	CASING_2D2
CASING	CASING_1D3
F5	F1
E7	E5
E7	E6

New Entity	Original Entity
2D1_DR	2D1
2D1_15_DR	2D1_15
2D1_16_DR	2D1_16
2D1_30_DR	2D1_30

Tab. 3. Sample Relations: (a) Virtual Topology relation, (b) Equivalence relation.

Virtual subsets can be created to partition an existing model so that appropriate analysis attributes can be applied. For example, the partial imprinted faces in Fig. 3 (b) are stored as subsets of the original face, linking the design and analysis representations. Virtual subsets are created by inserting new bounding entities into the description of a model, therefore dividing an entity into a number of smaller entities. Virtual supersets are created by ignoring the boundaries between adjacent entities to be merged with the same manifold dimension. The process for creating virtual supersets is described using the example shown in Fig. 4. Using the body names as the input to pre-built SQL queries, all compatible lower bounding topologies are automatically identified and merged, creating the correct topology of the superset entity. Therefore when created the new superset CASING = CASING_2D2 ∪ CASING_1D3, the first step in the process identifies the common boundary between the two cells to be merged, red face Fig. 4 (a). This face will not be part of the definition of the superset cell but its bounding entities indicate other entities which may also need to be ignored. Fig. 4 (b) shows some of the bounding entities of the common boundary, and the entities bounded by them may need to be merged when defining the superset. Finding the bounded edges of vertices V1 and V2 identifies possible edges to be merged, E3, E4, E5 and E6 Fig. 4 (c). During the topology extraction phase all underlying curve and surface types of each edge and face are stored in a separate relation named Attribute. When entities are being merged as a consequence of an entity it bounds being used to create a superset, consideration is given to entity compatibility. Edges are only merged if their underlying curves are compatible and continuity conditions are met. In this example, edges E3 and E4 in Fig. 4 (c) are not merged as their underlying curves are incompatible (circle and line). However, edges E5 and E6 can be merged as both underlying curves are lines and have parallel tangents at their common vertex. The original entities being merged E5 and E6 are stored in the Host Entity attribute of the Virtual Topology relation with new superset edge E7 being stored in the Entity attribute.

The same approach is used to find the faces to be merged. Finding the bounded faces of common bounding edges E1 and E2 identifies the faces to be merged, Fig. 4 (b). Faces F1 and F2 can be merged as they both have underlying planar surfaces. A new superset face F5 is created in the Entity attribute of the Virtual Topology relation with faces F1 and F2 stored in the Host Entity attribute. However, faces F3 and F4 are not compatible as their underlying surface types (planar and torus) are different. Where it is desirable to merge specific incompatible surface and curve types the appropriate virtual entity can be created in the database using a specific superset operation on the entities.

Once all virtual entities have been successfully identified and created, their bounding topologies are automatically entered into the topology relation using an SQL query on the Entity, Virtual Topology and Topology relations. For example, superset face F5, Fig. 4 (d), is bound by superset edge E7 instead of its host entities. This creates the topology of the new superset body, Fig. 4 (e), with all merged topologies forming the boundary. In this manner all body cells with the same parent entity in the decomposed model are automatically merged together, creating the original un-decomposed design topology. A virtual superset of original cells is created in the data structure, with the decomposed cells stored as subsets to remove any unnecessary partitions.

Multiple decompositions of a model can be stored in the same database. Attributes assigned to individual cells allow them to be identified as belonging to different virtual supersets for different analysis applications. For example, a virtual superset could be created between the hole feature cell in Fig. 1 (b) and its adjacent structural cell for an analysis of global

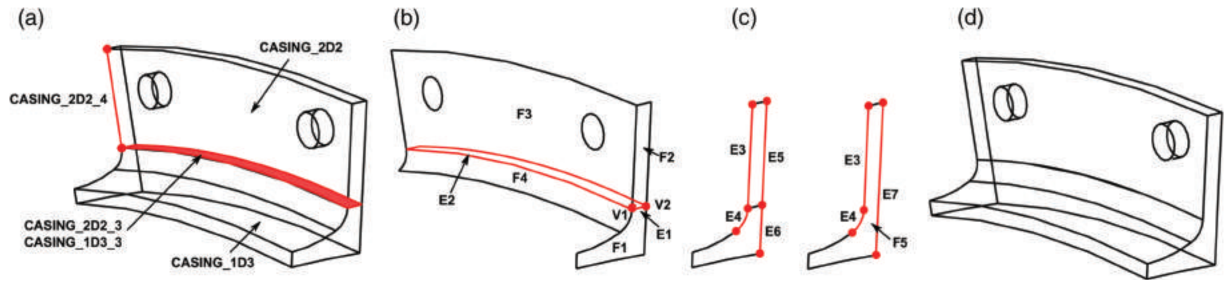

Fig. 4: Superset of two bodies: (a) Subset cells with common face highlighted, (b) Bounding entities of common face, (c) Possible edges to be merged, (d) Topology of merged face, (e) Superset body.

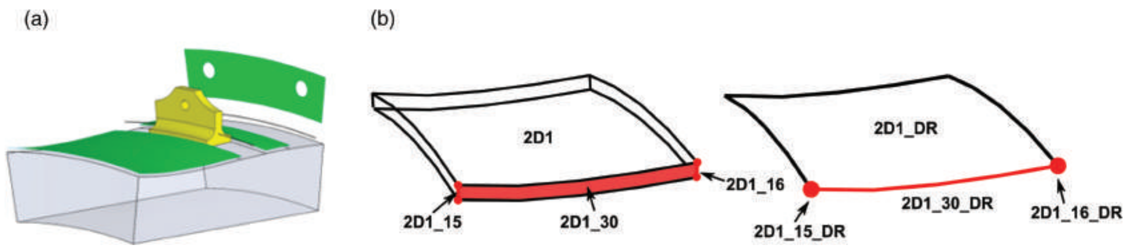

Fig. 5: Equivalencing cells: (a) Idealization of engine casing, (b) Thin-sheet region and mid-surface.

structural performance. Another superset could combine the hole cell with the surrounding fluid domain to analyze flow through the hole. Both representations can reside within the data structure.

Other types of virtual supersets can be created to aid with the application of boundary conditions. For example, by grouping the assembly of all structural cells into a virtual superset, the boundary faces of the assembly can be identified as the faces which only bound one structural cell. These boundary faces form the definition of the new virtual superset. Finding the common boundary between this superset and an adjacent fluid cell indicates where a pressure load is to be applied on the structural model from the fluid.

3.2.3. Linking Equivalent Representations

Throughout a design cycle it is common to use many different analysis models for different applications which all represent the same component. Dimensionally reduced analysis models are used during early stages of the design process where multiple iterations of a concept need to be analyzed quickly. One example is when thin-sheet and long-slender regions are represented by mid-surface and beam idealizations respectively, Fig. 5 (a). Equivalent relationships are stored in the Equivalence relation, Tab. 3 (b), consisting of two attributes: ***New Entity:*** the label of the idealized representation. For example the mid-surface representation of the 3D thin-sheet region shown in Fig. 5 (b). ***Original Entity:*** the label of the original cell that has been dimensionally reduced.

After the creation of an appropriate dimensionally reduced cell, equivalent relationships can be automatically identified and created in the database allowing information to flow between the detailed and idealized models. For the example model the type of dimensional reduction dictates the equivalent relationships required to link the idealized entities to their detailed representation. Due to having one dimension collapsed a mid-surface idealization has a difference in manifold dimension of one compared to its equivalent 3D cell, Fig. 5 (b). Using this logic it carries that mid-surfaces edges and vertices are equivalent to faces and edges respectively in the body cell.

Equivalent relationships are identified by interrogating the idealized entity and linking it to its detailed counterpart using the information stored in the database. The first step is to interrogate the mid-surface representation Fig. 5 (b), to store its topology in the database. Determining the equivalent relationships of idealized entities with the lowest dimension (bounding vertices of the mid-surface) allows the topology in the database to be used to identify any remaining equivalent relationships. The equivalent edge to an idealized vertex is identified because its mid-point is coincident to the vertex. Idealized vertices 2D2_15_DR and 2D2_16_DR are equivalent to edges 2D2_15 and 2D2_16 respectively in Fig. 5 (b). These equivalent relationships are stored in the Equivalence relation with dimensionally reduced entities in the New Entity attribute and the equivalent entity in the Original Entity attribute, Tab. 3 (b).

Once all vertices bounding the mid-surface have been identified their common bounded edges can be equivalenced using the topology information in the database. The mid-surface edge bounded by two idealized vertices is equivalent to the face bounded by the edges equivalent to the idealized vertices. Fig. 5 (b) shows idealized vertices 2D2_15_DR and 2D2_16_DR are equivalent to edges 2D2_15 and 2D2_16. The face bounded by these equivalent edges is 2D2_30. This face is equivalent to the edge 2D2_30_DR bounded by the two idealized vertices. Once all mid-surface edges have been linked to their equivalent faces the mid-surface can be related to its equivalent body cell. In the case of idealization it is common for one idealized entity to represent multiple entities in the detailed model. In such cases orientation information is used when establishing the equivalence to provide specific links to the higher dimensional entities. For example, in Fig. 5 (b) the top and bottom faces of thin-sheet region 2D1 can be equivalenced to the different sides of the mid-surface face by comparing face normals at specific locations on each face. These relationships allow boundary conditions or results to be transferred between representations. A similar approach is used to identify the equivalent relationships of any 1D or 0D idealizations. Mappings can be one to many between idealized and non-idealized entities i.e. the residual thin-sheet region faces linked to the mid-surface face. In other idealizations an edge in a 1D idealization can be linked to multiple longitudinal edges and faces in its detailed representation. These links can be determined using simple SQL queries to find any residual entities in the detailed representation that are not equivalent to an idealized entity.

In the prototype software SQL queries are used to relate entities to one another based on geometric position or topological relationships. Using the information stored in the database along with pre-built queries ensures the functionality is package independent. This enables idealization tools to be used within different environments and the resulting idealized geometry to be related back to its detailed representation. Once these relationships have been defined analysis attributes can be transferred between models at various levels of detail and dimensionality, with idealized interfaces derived from 3D non-manifold interfaces.

3.3. Transferring Analysis Attributes

In industry it is commonplace for structural and fluid analyses to be performed by different engineers using different analysis packages. It is essential to be able to transfer analysis attributes between different packages and analysis models. Adjacencies in the non-manifold cellular model represent interfaces between different body cells. These interfaces are used to automatically link multiple simulation models with varying levels of detail and dimensionality.

The identification of interfaces through topological adjacencies in the non-manifold cellular model provides a robust mechanism for the application of boundary conditions. Consider a pressure applied between all structural cells and FLUID1 in Fig. 3 (a). The non-manifold cellular model can return the required interface between these cells and apply the pressure automatically, even after changes to the design model occur and the model topology changes. This enables boundary conditions, loads etc. to be applied automatically after design updates, eliminating manual re-work of boundary conditions etc. Calculated interfaces in the non-manifold cellular model are used with Virtual Topology and equivalencing information to link boundary conditions and results between different representations and disciplines without loss of integrity. Using the calculated fluid-structural interfaces fluid loads can be automatically applied to structural body cells, or dimensionally reduced cells, depending on the complexity of the structural analysis to be carried out.

Sellgren [15] describes an interface as the interaction between two mating faces, where the interface characteristics are derived from the mated features. In this work the simulation significance of each interface can be derived from the analysis attributes attached to its parent entity. The attributes attached to interacting cells determine the type of boundary conditions to be applied. Interfaces between cells in the decomposed cellular model in Fig. 3 (a) are used along with analysis attributes of the interacting cells to automatically couple incompatible meshes. For example, where a shell meshed region meets a solid meshed region, MPCs can be created automatically.

It is essential to provide a way to track topological entities after geometric updates to successfully transfer analysis attributes between packages. Tab. 1 shows that not all design and analysis tools allow name attributes to be attached to topological entities. For example, Parasolid allows attributes to be attached to topological entities but it has been identified in this work that it has difficulty tracking these after split operations. This is a problem when progressing from a non-manifold to a manifold model. A procedure has been developed which can relate topological entities to their equivalent database entities, regardless of the package. Identifying lowest dimension topological entities first enables SQL queries to be used to identify higher dimension entities based on the topological connectivity stored in the database. This is similar to the method for identifying equivalent relationships. All vertices in the 3D representation are linked to vertices in the database by identifying coincident points within a specified tolerance. This assumes that at the most basic level all CAD / CAE packages should be able to return the position of a vertex so that it can be compared to the Identifiers of entities with a zero dimension in the database. Once vertices have been identified common bounded queries are used to form the relationships between the remaining edges, faces and their equivalent database entities. Geometric information stored in the Attribute relation of the database is used to identify edges with no bounding vertices. Analysis attributes are related to database and body entities. Therefore, once all topological entities in the CAD / CAE environment have been related back to their database entities, any analysis attributes can be transferred to the geometric model.

The process described here allows multiple different packages to be utilized as part of a single design process, giving access to the specialist technology in each, with all models robustly linked. For example, one tool might be used to decompose design geometry for meshing. Another might be used to apply efficient meshing strategies to decomposed cells based upon their geometric properties, Fig. 6 (b). Based upon analysis requirements other packages might be selected for their solving capability, but using current tools these packages will only have access to the mesh but not its associativity with the geometry Fig. 6 (c).

Storing the Virtual Topology and equivalence relationships and all associations with mesh entities in the database allows the mesh to be fully associated with the original geometry, Fig. 6 (d). In this example Abaqus has been used to automatically apply Virtual Topology to create a fit for purpose mesh. Any virtual entities are explicitly stored in the database along with their host entities. This allows the mesh to be transferred and related back to the original geometry without having to recreate the Virtual Topology. Once the mesh has been created there is no requirement to recreate the Virtual Topology in other packages as the mesh can be related back to the original host entities. Each element in a mesh is considered a subset of the component being meshed. This creates a neutral CAE mesh representation, allowing meshes to be transferred between different packages, removing the need to create the mesh from scratch in each package and allowing results on meshes to be transferred between different packages.

Multiple design and analysis models can be linked at a topological level regardless of the level of fidelity. Relating analysis attributes (mesh entities, loads, boundary conditions etc.) to topological entities in the analysis models creates the bi-directional associativity essential for seamless CAD and CAE interoperability. These associations can be used to identify subsets of the design model for remeshing or abstraction at a local instead of global level, or for transferring results across domains for multi-disciplinary analysis. During optimization many design changes and topology updates may occur in an automated fashion by maintaining strong links between design and analysis models.

4. DISCUSSION

This paper has introduced a method for using a relational database to link design models to the various analysis models used throughout the design evolution. Storing topology at its lowest cellular level enables fit for purpose analysis models to be derived using Virtual Topology, equivalencing and calculated interface information, with the ability to interrogate and manipulate both the individual cells and complete domain. The fact that many commercial packages do not contain non-manifold capabilities, Tab. 1, does not restrict them from interacting with the automated tools presented here.

Cellular Modeling requires the design space to be divided into cells with different simulation significance. Cells may represent structural or fluid domains. Using existing tools the creation of fluid domains is a manual operation involving complicated Boolean union and subtraction operations. This counteracts the automation of the processes described in this work. In Section 3.1 it was described how the design space was partitioned to automatically create definitions of surrounding fluid domains. A fluid analysis might require further simplification of these fluid domains into cells with analysis significance appropriate to the analysis being performed.

This technology could remove the need for direct geometric editing when generating analysis models. For example, storing decomposed cells as subsets of an original body, with a definition of the splitting surface would negate the need to actually split the geometry. Mesh generation tools could use the information in the database to create the desired mesh based on Virtual Topology decisions made in another package and apply appropriate loading and boundary conditions. Updates in the CAD environment would be required purely for visualization purposes and for design updates.

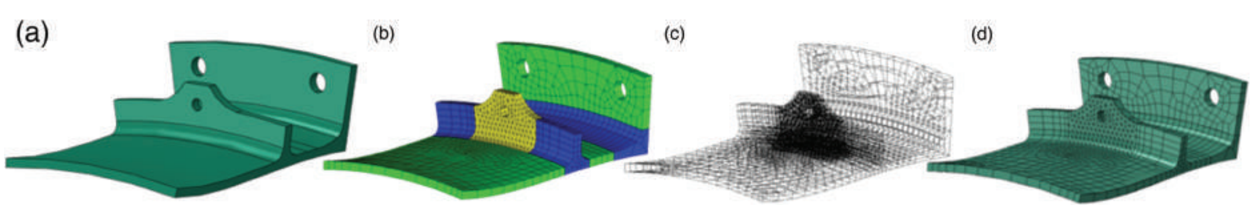

Fig. 6: Mesh geometry associativity: (a) Original design geometry, (b) Efficient mesh on decomposed model, (c) Orphan mesh, and (d) Efficient mesh fully associated with original geometry.

Equivalencing information stored in the database keeps track of dimensionally reduced idealizations, allowing analysis models of different dimensionality to be automatically generated. Cells may be used for different applications in different analyses. For example it may be that cells are considered bodies for one stage of the design, but may be represented as 0D masses and inertias in another analysis. This work establishes and stores the link between these different representations. Using interface information in the cellular model and the information in the database the desired bi-directional associativity is created. Analysis attributes such as mesh, boundary conditions, loading etc. can be transferred from one domain to another. As all analysis attributes are associated back to the design geometry these relationships could even be used to influence simplification procedures or update design geometry based on analysis results. The relationship between mesh entities and topological entities could be used for optimization routines where links between design and analysis models are robustly defined.

The approach described in Section 3 illustrates how this work can be used at any stage of the design process. It has been demonstrated how a geometry decomposition tool can be integrated within the process. However, other tools (e.g. defeaturing tools) could be integrated just as easily by assigning relevant attributes to cells in the cellular model. This would allow multiple virtual supersets to be created for different design applications i.e. different decompositions may be required for different load cases where different features may be simplified depending on where low/high stress regions occur. Other idealized analysis models like 2D axisymmetric, 2D plane stress/strain or 3D symmetrical models could be linked to their detailed representation using the tools developed here. For example, the procedure for linking a 2D axisymmetric model to its equivalent representation is similar to that for a mid-surface, where 2D axisymmetric edges are equivalent to 3D faces. All of these simulation models can reside within the one data structure, which maintains robust links between all representations. Consider a multi-partner project in which many different tools are utilized across different domains. If one partner is tasked with simplifying the geometry for analysis, another partner can use the tools developed here to link the abstracted geometry to the equivalent original design geometry, allowing analysis attributes and results to be transferred.

One important feature of this approach is to maintain the relationship between the topology in the database and its CAD model representation. Name attributes provide the most convenient method for linking topology between CAD models and the database. However, Tab. 1 shows that not all packages allow name attributes to be applied to lower-dimensional topological entities. In this work lower-dimensional topological entities can be identified using their Identifier and generic SQL queries to find common bounded entities. Thus, persistent naming of low level entities effectively allows the persistent naming of high level entities. The process of tracking entities is simplified even further when packages allow name attributes on a given body cell. Transferring a model between different design and simulation tools can compromise the integrity of a model. Possible ambiguities may arise due to topological inconsistencies or loss of attributes attached to geometric or topological entities. Tab. 1 shows the Parasolid tolerance for geometry is 10^{-8} mm, compared to that of Abaqus which is 10^{-6} mm. This means that points considered different in Parasolid may be coincident in Abaqus. Geometric inaccuracies between systems are accounted for by applying tolerances within the data structure, which allows all topological entities to be persistently named between packages with different data structures. Whilst the current solution can track entities between different packages this work has to be extended to deal with situations where new topological entities are introduced during CAD model transfer. These entities may be introduced due to different underling architectures, with some packages introducing new vertices or edges to split seamless edges and faces respectively. Virtual Topology and equivalencing will be able to represent such changes. For example, where one package splits a seamless edge into two edges, these new edges can be stored as subsets of the original seamless edge. In other circumstances individual parts may contain dirty geometry and CAD healing tools may introduce new faces to fill voids in a model. An alternative way to identify entities would be to use the Identifier of each entity to find its closest topological entity. This would require geometric searching functionality, which is only available in select packages. The approach described here is independent of any underlying package.

One issue not discussed in this paper is how significant design modifications (changing topology / adding new features) are reflected in downstream simulation models. The same methodology used to identify topological entities between different packages could be used to identify changes in a model. The identification of new entities along with any redundant entities would enable the database to be updated using standard SQL queries. Design changes can be linked to specific cells allowing simplification tools to work at a local instead of global level, reducing the time required to regenerate an analysis model after design changes.

These tools can be used with assemblies consisting of multiple components. Boundary conditions such as contact conditions between interacting components can be automatically identified and used to automatically rebuild simulation models for assemblies. However, the current tools are limited to dealing with assemblies consisting of clean interfaces between adjacent volumes. These interfaces can be automatically identified in the non-manifold cellular representation, allowing analysis models of various levels of fidelity to be linked to one another. In reality, assemblies can contain slight misalignments, causing gaps or overlaps [5]. It is believed tolerant modeling can be used to identify these interfaces so they can be stored in the database without any geometric modifications. These inaccuracies can occur due to CAD model translation or even components being modeled to tolerance instead of nominal dimensions. Issues such as these can become major problems when dealing with assemblies where different components may come from different sources.

5. CONCLUSIONS

In this paper procedures have been described which make use of Cellular Modeling, Virtual Topology and Equivalencing to integrate the different representations generated by and required for CAD and CAE. It has been demonstrated how these technologies can be combined to:

- Link detailed design models to simplified analysis models.
- Link multiple analysis models required for different tasks within one data structure.
- Link design and analysis models between multiple CAD and CAE packages.
- Transfer analysis attributes between analysis models at different levels of fidelity and between different analysis packages.

Future work will demonstrate how the same procedures described in this paper can be used to strengthen the bi-directional links required to:

- Reflect design modifications in downstream simulations models.
- Robustly link models between different packages when new topological entities are introduced due to different underlying data structures or geometry clean-up procedures.
- Deal with the practical issues involved with inconsistent interfaces in assembly models.

REFERENCES

[1] Arabshahi, S.; Barton, D. C.; Shaw, N. K.: Steps towards CAD-FEA integration, Engineering with Computers, 9 (1), 1993, 17–26. DOI: 10.1007/BF01198250

[2] Armstrong, C. G.: Modelling requirements for finite-element analysis, Computer-Aided Design, 26(7), 1994, 573–578. DOI: 10.1016/0010-4485(94)90088-4

[3] Butlin, G.; Stops, C.: CAD Data Repair, 5th International Meshing Roundtable, Pittsburgh, 1996.

[4] Cavalcanti, P.R.; Carvalho; P. C.; Martha, L. F.: Non-manifold modeling: An approach based on spatial subdivision, Computer-Aided Design, 29(3), 1997, 209–220. DOI: 10.1016/S0010-4485(96)00066-8

[5] Clark, B. W.; Hanks, B. W.; Ernst, C. D.: Conformal Assembly Meshing with Tolerant Imprinting, Proc. 17th International Roundtable, 2008, 267–280.

[6] Codd, E. F.: A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, 26 (1), 1970, 64–69. DOI: 10.1145/357980.358007

[7] Crocker, G. A.; Reinke, W. F.: An editable non-manifold boundary representation, IEEE Computer Graphics and Applications, 11 (2), 1991, 39–51. DOI: 10.1109/38.75589

[8] Gujarathi, G. P.; Ma, Y. S.: Parametric CAD/CAE integration using a common data model, Journal of Manufacturing Systems, 2011, 118–132. DOI: 10.1016/j.jmsy.2011.01.002

[9] Hoffman, C. M.; Joan-Arinyo, R.: CAD and the Product Master Model, Computer-Aided Design, 30 (11), 1998, 905–918. DOI: 10.1016/S0010-4485(98)00047-5

[10] Lee, S. H.: A CAD-CAE integration approach using feature based multi-resolution and multi-abstraction modeling techniques, Computer-Aided Design, 2005, 941–955. DOI: 10.1016/j.cad.2004.09.021

[11] Makem, J. E.; Armstrong, C. G.; Robinson, T. T.: Automatic decomposition and efficient semi-structured meshing of complex solids, Engineering with Computers, DOI: 10.1007/s00366-012-0302-x, 2012.

[12] Nolan, D. C.; Tierney, C. M.; Armstrong, C. G.; Robinson, T. T.; Makem, J. E.: Automatic dimensional reduction and meshing of stiffened thin-wall structures, Engineering with Computers, DOI: 10.1007/s00366-013-0317-y, 2013.

[13] Robinson, T. T.; Armstrong, C. G.; Fairey, R.: Automated mixed dimensional modeling from 2d and 3d cad models, Finite Elements in Analysis and Design, 47 (2), 2011, 151–165. DOI: 10.1016/j.finel.2010.08.010

[14] Sang, H. L.; Lee, K.: Partial Entity Structure, ASME Journal of Computing & Information Science in Engineering, 1(4), 2001, 356–365. DOI: 10.1145/376957.376976

[15] Sellgren, U.: Doctoral Thesis - Simulation-driven design, The Royal Institute of Technology, Stockholm, 1999.

[16] Sheffer, A.; Blacker, T.; Clements, J.; Bercovier, M.: Virtual Topology Operators for Meshing, in 6th International Meshing Roundtable, Sandia National Laboratories, 1997. DOI: 10.1142/S0218195900000188

[17] Shephard, M. S.; Beall, M. W.; O'Bara, R. M.; Webster, B. E.: Toward simulation-based design, Finite Elements in Analysis and Design, 2004, 1575–1598. DOI: 10.1016/j.finel.2003.11.004

[18] Sypkens Smit, M.; Bronsvoort, W. F.: Integration of Design and Analysis Models, Computer-Aided Design and Applications, 2009, 795–808. DOI: 10.3722/cadaps.2009.795-808

[19] Thakur, A.; Banerjee, A. G.; Gupta, S. K.: A survey of CAD model simplification techniques for physics-based simulation applications, Computer-Aided Design, 41, 2009, 65–80. DOI: 10.1016/j.cad.2008.11.009

[20] Weiler, K.: The Radial Edge Structure, in Geometric Modeling for CAD applications, North-Holland, 1988, 3–36.