Using Mesh-Geometry Relationships to Transfer Analysis Models between CAE Tools

Christopher Tierney, Declan Nolan, Trevor Robinson, and Cecil Armstrong

School of Mechanical and Aerospace Engineering, Queen's University Belfast, BT9 5AH
`ctierney06@qub.ac.uk`

Abstract. Integrating analysis and design models is a complex task due to differences between the models and the architectures of the toolsets used to create them. This complexity is increased with the use of many different tools for specific tasks during an analysis process. In this work various design and analysis models are linked throughout the design lifecycle, allowing them to be moved between packages in a way not currently available. Three technologies named Cellular Modeling, Virtual Topology and Equivalencing are combined to demonstrate how different finite element meshes generated on abstract analysis geometries can be linked to their original geometry. Establishing the equivalence relationships between models enables analysts to utilize multiple packages for specialist tasks without worrying about compatibility issues or rework.

Keywords: Mesh-geometry ownership, CAD / CAE integration.

1 Introduction

The introduction of Computer Aided Design (CAD) and Computer Aided Engineering (CAE) tools has had a major impact on the Product Development Process. Designs can be developed and tested in the virtual environment, reducing the need for expensive prototypes. Computational analysis methods like finite element analysis (FEA) have progressed from validation and failure verification tools to design and concept verification tools, resulting in them being employed earlier in design cycles where analysis results drive the design process [12]. Consequently, the capabilities of modern analysis tools are rapidly increasing, along with the complexity of the analyses being undertaken. This has led to more detailed analyses being performed at earlier stages of design processes. Aerospace companies undertake multi-level, multi-disciplinary analyses of components throughout the design process. The multi-disciplinary analyses allow for a more accurate assessment of the overall behavior of a system, but leads to a significant burden when preparing the different analysis models.

J. Sarrate & M. Staten (eds.), *Proceedings of the 22nd International Meshing Roundtable*,
DOI: 10.1007/978-3-319-02335-9_21,

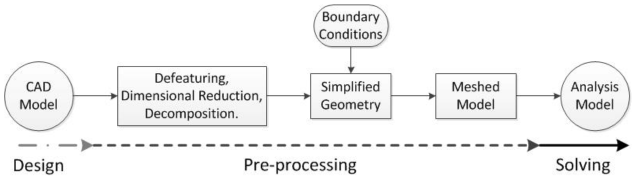


Fig. 1 Common analysis process using current methods

In many analysis cycles there are numerous pre-processing steps required to enable geometric domains to be more adequately meshed, Fig. 1. Pre-processing steps may include model simplifications where manufacturing or aesthetic design details which are assumed to have no simulation significance are removed [15]. The pre-processing steps tend to vary depending on analysis requirements and the stage of the design process, e.g. details that are significant for a stress analysis may be irrelevant for a modal analysis. In other circumstances different geometric decompositions may be required for analyzing different load cases for a product so that high stress areas can be accurately represented. With multiple decompositions of the same CAD model required for different analysis tasks, it is essential that bi-directional links exist between equivalent models [1]. These links are necessary to ensure that results can be exchanged between different analyses, especially for coupled multi-disciplinary analyses where the output from one simulation acts as the input to another e.g. aero-elastic analysis.

In reality industrial companies utilize numerous specialist tools for analysis activities. The tools employed depend heavily on the task being performed, the type of physics involved and the stage of the design process [4]. Early in a design process it may be appropriate to use an automatically generated unstructured tetrahedral mesh to gain an initial insight into the performance of the product. However, later in the design process it may be desirable to use multiple different analysis packages with specific capabilities. This can result in multiple analysis models existing for a given component with no robust link between them and no link back to the original design geometry. This significantly adds to the complexity of the CAD/CAE integration problem. The analysis models to be solved consist of a finite element mesh along with any applicable analysis attributes such as boundary conditions and material properties. The application of boundary conditions for large assemblies like whole engine thermo-mechanical models can be a tedious and time consuming task, due to the vast number of physical interactions. Without a robust method to transfer analysis information between packages these tasks may have to be repeated many times. It is shown in this work that the ability to link meshes from various analysis geometries back to the original CAD model (and by extension each other) reduces rework in terms of setting up the analysis model.

Mesh-geometry ownership is the relationship between individual mesh entities and their parent B-Rep entity. During mesh generation modern packages automatically impose mesh-geometry ownership, allowing boundary conditions applied to B-Rep entities to be transferred to the corresponding nodes and elements of the mesh [2]. This is more convenient than applying boundary conditions to individual mesh entities, which can be cumbersome for even the simplest of models. For the purposes of assembling the finite element matrices, boundary conditions on the CAD model are internally converted to equivalent nodal 'loads' or imposed 'displacements'. Therefore, CAD model parameter updates or mesh modifications can be made without having to reapply loads. Using current tools it is not always the case that mesh-geometry ownership can be successfully transferred from its native package to a different downstream package, especially when the models are of a different level of abstraction or fidelity. This makes repetitive manual operations essential in order to successfully recreate the mesh-geometry ownership.

In this work, recording the equivalent relationships between design and analysis models enables finite element meshes created on abstract analysis geometries to be fully associated with the original design model, regardless of the packages they were created in. This enables analysis attributes like meshes and boundary conditions to be transferred between packages and models at various levels of fidelity, without loss of integrity. This gives an analyst the freedom to select the desired tools for specific aspects of the analysis process without having to worry about compatibility issues or any substantial rework involved in transferring and rebuilding analysis models.

2 Three Technologies for Linking Design and Analysis Models

This work describes how three technologies named Cellular Modeling, Virtual Topology and Equivalencing are used to manage and manipulate the topology of geometric design and analysis models, enabling them to be linked, independent of any underlying CAD or CAE package [16]. These technologies are not new by themselves, but their combination for CAD/CAE integration in the manner described is novel. In this section a brief description of the three technologies is provided.

2.1 Cellular Modeling

Cellular representations are defined as non-manifold representations of both positive (solid) and negative (void) regions [3]. In manifold representations any point on the boundary of a solid region has a neighborhood homeomorphic to a 2-dimensional disk [17]. Geometric representations that are not manifold are referred to as non-manifold. Cellular representations have been used in assembly mesh generation, where non-manifold topological entities between interacting components provide suitable interfaces for conformal meshing [18]. In other work Sypkens Smit and Bronsvoort [14] used cellular modeling representations for remeshing feature models. Cells in the cellular model are related to their parent features so they can be tracked after model updates, allowing boundaries to be compared so that local remeshing can be efficiently achieved.

In this work a region in a non-manifold representation is referred to as a cell. Cells can be of any manifold dimension, i.e. volumes, faces, edges and vertices are all just regions of space bounded by other cells. Cellular models subdivide the entire design space into regions with specific analysis significance. Every cell in the cellular decomposition contains information specifying its origin in relation to the design model. An example of a non-manifold condition is a face adjacent to two distinct cells. The face and its bounding entities are shared between the two cells. It will be shown later in Fig. 11 that the interfaces between interacting cells in the non-manifold cellular representation can be used to link design and analysis models at various levels of fidelity.

2.2 Virtual Topology

Virtual Topology was introduced by Sheffer et al. [11] as a technique for preparing CAD models for analysis purposes. It allows simplifications to be made on a model without worrying about the complications of direct geometric editing, which may introduce even more changes from the original model. Virtual Topology operations use real topological entities called hosts in order to create virtual entities. Virtual superset entities represent the combination of multiple adjacent entities, while virtual subset entities represent a partial section of an original entity. Parasite entities are used to split a higher dimensional topological entity into several subset entities.

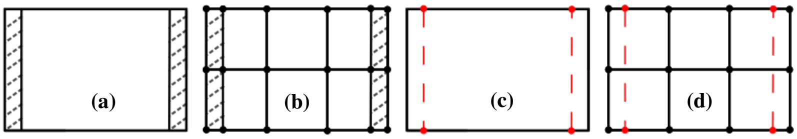


Fig. 2 Virtual Topology for mesh generation: (a) Model with sliver faces, (b) Mesh on original model, (c) Superset face, (d) Mesh on superset face

One application of Virtual Topology is for mesh generation, as shown in Fig. 2. Small features like sliver faces, hatched in Fig. 2 (a), may be created during CAD modeling or model translation. These sliver faces provide problems to mesh generation algorithms as nodes are usually distributed along all bounding entities in a model. When faces are small in comparison to the target element size, poorly shaped elements are created, hatched in Fig. 2 (b), which can have an adverse effect on the accuracy and efficiency of an analysis. Merging these sliver faces with adjacent larger faces ignores their common bounding entities, dashed in Fig. 2 (c), creating a virtual superset face, Fig. 2 (d) to avoid poor mesh quality.

Many analysis or mesh generation packages have Virtual Topology capabilities. Some packages make these Virtual Topology decisions internally without reporting the details to the analyst, making it difficult for the analyst to determine where it has been used. Also, once Virtual Topology has been used to simplify a model for meshing, certain packages do not allow the simplified geometry to be exported. These issues make it difficult to reuse the Virtual Topology generated. In this work Virtual Topology decisions are identified and stored centrally so that they can be accessed and used by other packages. Additionally, by using Virtual Topology along with the appropriate non-manifold interfaces, multiple analysis decompositions can be linked to each other. Once analysis geometries have been linked to the original geometry it is possible to relate any resulting meshes to the original geometry, and by extension to each other.

2.3 Equivalencing

Different analysis models and therefore different meshes are required for different applications during the design process. The analysis models can differ due to the stage of the analysis or even the type of analysis being performed. Early in design processes simpler analysis models with fewer details and degrees of freedom may be utilized as approximate results are often acceptable, provided they are returned quickly. Analysis complexity normally increases as the design evolves. Different meshes can also be required to solve different physics problems and to get more detailed results.

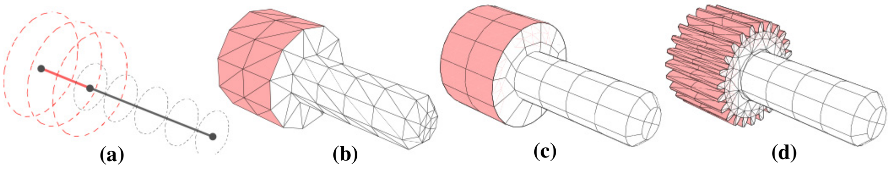


Fig. 3 A simple pinion shaft with different meshes applied for specific tasks during a design process: (a) Beam elements with different cross-sectional properties, (b) Linear tet mesh, (c) Structured hex mesh, (d) Hybrid mesh

In this work it is considered that as a mesh is a representation of a domain, all meshes of that domain can be considered equivalent to it and each other. The representative example in Fig. 3 shows various analysis models that are considered equivalent as they represent the same component. For example, early in the design process a pinion shaft may be represented with two 1-dimensional beam elements with different cross-sectional properties, Fig. 3 (a). A coarse tet mesh may be required for a modal analysis, Fig. 3 (b) and a structured hexahedral mesh may be required at a later stage for an impact analysis, Fig. 3 (c). During detailed design stages a hybrid mesh may be necessary for a stress analysis, Fig. 3 (d). While these different representations are adequate for their specific purpose, equivalent relationships can be determined between equivalent cells in the model.

For example, the pinion body is represented as a beam element in Fig. 3 (a), a tet meshed or hex meshed cylinder cell in Fig. 3 (b) and (c) and a tet mesh of a detailed geometry including gear teeth in Fig. 3 (d). The end vertices of the beam elements are considered equivalent to planar faces of the pinion head in other representations. Similarly, relationships between the simple gear and detailed gear cells can be determined using a non-manifold combination of the cells, where non-manifold faces represent interactions between cells. By establishing the equivalences between individual analysis geometries and the original model it is possible to transfer analysis attributes and results between all models generated for that domain.

3 Robust Mesh Transfer Process

It is normal for boundary conditions to be applied to topological entities in the simplified geometric model, Fig. 1. If analysis and design models are disconnected then boundary conditions need to be reapplied for each subsequent analysis model. Using the approach described here boundary conditions can be defined on the original design geometry without having to worry about any downstream idealizations that may occur, Fig. 4. Once fit-for-purpose meshes have been generated, the simplified geometric model is no longer required to achieve the analysis solution. Storing equivalent links between the simplified and original models allows mesh-geometry ownership to be transferred between the models. Therefore, boundary conditions assigned to the original model can be automatically transferred to the idealized mesh before it is solved. This results in integrated design and analysis capabilities that can have major benefits in large collaborative projects with many distributed partners. Different departments and sub-contractors are assigned specific tasks within the analysis process. Each partner may prefer to use their toolsets of choice without having to worry about how this may affect downstream collaboration, or without sharing any information about how the models were generated to protect intellectual property. Example tasks which could be integrated include model simplification, meshing, assignment of boundary conditions etc. Integrating the models produced and required by the different analysis tools ensures tighter integration of the entire design and analysis process.

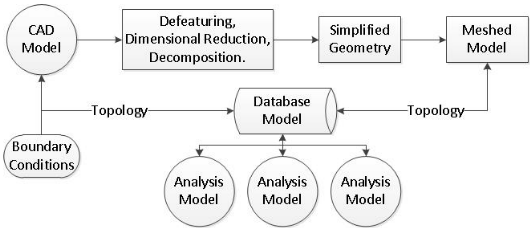


Fig. 4 Analysis process using the master database to link various representations

In order to transfer various finite element attributes between different models residing in different packages it is essential that the various models are robustly linked [5, 13]. The ability to relate different meshes back to the same design model provides tighter integration between the disciplines. In order to link these models a simple data structure has been developed which is independent of any underlying CAD or CAE package. The data structure has been created in the form of a relational database whose entity relation diagram is detailed in Fig. 5. Its purpose is not to replace existing data structures used to represent models in CAD and CAE systems, rather it is used to store the non-manifold topology of the cellular model for the product being designed, and also the different approximations and analysis models that represent it. All equivalent decompositions of the product are stored regardless of their dimension or the tools used to create them. The database can be seen as a master model for linking these equivalent representations. The advantage of using a non-manifold model is that non-manifold interfaces can be used, along with Virtual Topology and equivalence information to integrate all models. A complete description of the data structure and its use for analysis applications is available in [16].

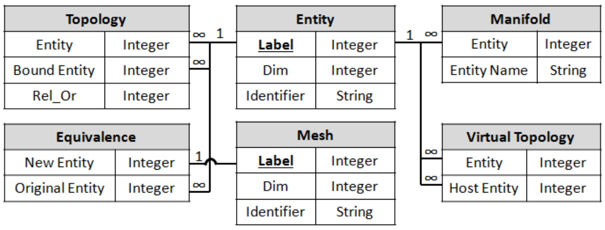
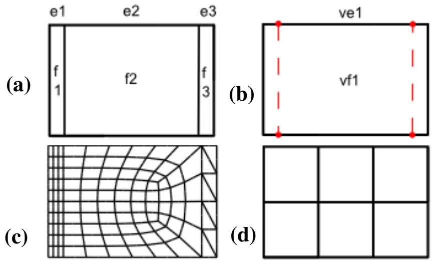


Fig. 5 Database entity-relation diagram

Once the analysis geometry has been created its non-manifold topology can be stored in the database, Fig. 5. The topology is stored independent of the packages used to create the model. The main topological entities (vertices, edges, faces and volumes) are stored in the Entity relation, allowing the topology to be manipulated without affecting the underlying CAD model so that various virtual decompositions can be created. Fig. 5 shows that bi-directional links exist between the master database and the different models. In order to achieve this, each entity is identified by a point in space lying within the boundary of a topological entity. This Identifier enables robust identification of entities between various packages. This is in comparison to naming attributes that may go missing, change or cannot be applied to a topological entity. The topological connectivity of a model is stored in the Topology relation of the database, whose rows define a cell, one of its bounding entities and the relative orientation of the two (e.g. whether a surface normal points into or out of a body). It is stored in a generic format that makes it accessible by any CAD or CAE package.



Entity	Host Entity
ve1	e1
ve1	e2
ve1	e3
vf1	f1
vf1	f2
vf1	f3

(e)

Fig. 6 Linking different decompositions: (a) Original geometry, (b) Simplified geometry, (c) Fine mesh on original geometry, (d) Coarse mesh on simplified geometry, (e) Virtual Topology relationships

The database can store the topology of various decompositions of the same component, along with any mesh-geometry ownership relationships. By identifying the links between the original, Fig. 6 (a), and simplified geometries, Fig. 6 (b), it is possible to link their respective meshes. The Virtual Topology relation of Fig. 5 stores the link between the models. Superset edge ‘ve1’ is created by merging edges ‘e1’, ‘e2’ and ‘e3’, which are stored as its host entities, Fig. 6 (e). Virtual superset face ‘vf1’ is stored in the same manner for faces ‘f1’, ‘f2’ and ‘f3’. Linking the original and virtual models enables their meshes to be linked by manipulating their mesh-geometry relationships. Therefore, results may be transferred between the different models. Consider the original geometry in Fig. 6 (a), where a mesh for a thermal analysis has been created to calculate the temperatures on each face, Fig. 6 (c). A fine mesh can be used for a thermal analysis as nodes have only one degree of freedom and the analysis is relatively inexpensive. Once temperatures have been calculated Virtual Topology relationships are used to link them to the simplified model in Fig. 6 (b). To achieve this, the collection of element faces of the thermal analysis mesh are related back to their parent topological faces (‘f1’, ‘f2’ and ‘f3’). These topological faces are linked to the virtual face in the simplified model and by extension to the coarse mesh, Fig. 6 (d), applied to the virtual face. Once the links have been determined temperature values can be mapped and interpolated between source and target meshes in order to execute a structural analysis. The interpolation between meshes is not explored in this paper. The important point is that the meshes can be linked by using Virtual Topology to link the two equivalent geometries.

In this work the goal is to establish and store relationships between the topological entities in equivalent design and analysis representations. Virtual Topology is used to store the relationships between different representations when a one-to-one correspondence does not exist between related entities. These cells have to be defined as supersets or subsets of different cells. Equivalent relationships are stored when the same region of the design space is represented at different levels of fidelity. Once these relationships have been defined, analysis attributes can be transferred between models at various levels of fidelity.

3.1 Identifying Virtual Topology Relationships

Automated Virtual Topology tools available in commercial CAE packages can be used for geometry clean-up. They operate by identifying small features that may hinder the mesh generation process and merge them with adjacent larger features in the model, without altering the actual CAD geometry. Entities are merged together by ignoring their common bounding entities, i.e. common edges are ignored to merge adjacent faces. To enable Virtual Topology operations to be reused in other downstream applications it is necessary to establish relationships between virtual entities and their host entities so they can be stored in the database. Since many CAE tools do not report the details of Virtual Topology operations, either the bounding entities that have been ignored or the entities that have been merged together need to be identified. The two types of virtual entities to be identified in clean-up operations are superset edges and faces. For clarification, the model resulting from the Virtual Topology will be referred to as the virtual model and the cellular model will be referred to as the host model.

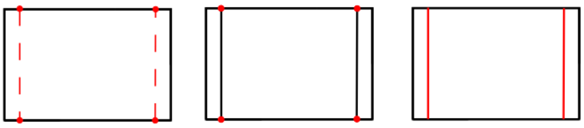


Fig. 7 Virtual Topology relationships: (a) Virtual model, (b) Ignored vertices in host model, (c) Ignored edges in host model

After Virtual Topology has been applied the topology of the virtual model, Fig. 7 (a), can be compared to the topology of the cellular model already stored in the database in order to find the ignored edges and vertices. The first step compares the positions of vertices in the host model to those in the virtual model. Ignored vertices are host vertices that do not exist in the virtual model, Fig. 7 (b). The topological connectivity of the host model in the database is then used to find the connected set of bounded edges of the ignored vertices. These bounded edges are the host edges that have been merged together to create a virtual superset edge. The topology of the superset edges is automatically created in the database by finding the unique bounding entities of the host entities, i.e. the unshared bounding entities.

A comparison between the edges in the virtual and host models is used to return any ignored edges, Fig. 7 (c). Once all vertices in the virtual model have been identified, the topological connectivity in the database is used to identify edges in the virtual model. Any edge in the host model not linked to an edge in the virtual model is identified as an edge that has been ignored due to the Virtual Topology operation. The bounded faces of the ignored edges in the host model represent the host faces that have been merged. Therefore, the new superset face is automatically linked to its host faces and the relationship stored in the Virtual Topology relation. Storing these relationships in the database creates the links between the virtual and host models and enables them to be reused by downstream applications. Therefore, once the mesh has been generated it can be linked to a different model in a different package without having to recreate any virtual entities.

3.2 Mesh-Geometry Ownership

In this work a mesh is considered to be an equivalent representation of the geometry it represents. Mesh generation processes in commercial packages position nodes and elements onto topological entities in the geometric model. The relationship between a mesh entity and its parent topological entity is stored in the Equivalence relation of the database. Mesh entities are stored in the 'New Entity' attribute and their parents in the 'Original Entity' attribute. Mesh entities can have only one distinct topological entity as their parent, while topological entities may have multiple mesh entities linked to them. This is shown in Fig. 5 where mesh entities can only appear once in the 'New Entity' attribute, imposing a one-to-one relationship between the mesh entity and its parent entity. However, topological entities can appear many times in the 'Original Entity' attribute as they can have multiple child entities. These relationships are used to successfully transfer a mesh between models at various levels of fidelity.

Mesh entities include nodes, element edges, element faces and elements. Different relationships may exist between mesh entities and their parent topological entities. Nodes may have B-Rep vertices, edges, faces or volumes as their distinct parent entity. Element edges and faces can have B-Rep edges and faces as their respective parents if they lie on topological boundaries. Elements have B-Rep parent entities equivalent to their dimensionality. For example, solid elements will have B-Rep volumes as their parent while shell elements will have B-Rep faces as their parent entity. These relationships are considered equivalences for the purposes of this work and are stored in the Equivalence relation of the database.

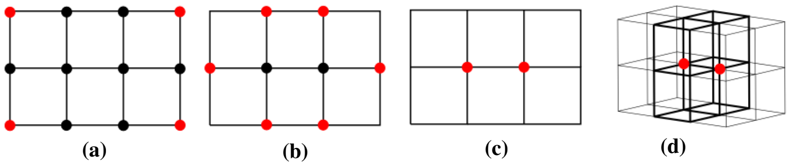


Fig. 8 Nodal B-Rep parentage relationships; (a) Vertex parents, (b) Edge parents, (c) Face parents, (d) Volume parents

Storing relationships between all mesh entities and their parent B-Rep entities would introduce redundant relationships that would complicate the transfer process. For example, element edges and faces are not equivalent to topological edges and faces if they lie inside the boundary of a region. Some packages assign pressure loads either directly as nodal values while other package may assign them to element faces or edges. Therefore, parentage information may or may not be required for certain mesh entities depending on the package. However, it is certain for all packages that loads and boundary conditions are ultimately represented on nodes and elements of the finite element mesh before an analysis is executed [8]. Due to these issues it has been decided to store parentage information for only node and element mesh entities. If certain packages require the use of element edges and faces for boundary condition application they can be derived from the element connectivity and equivalence relationships.

Each B-Rep topological entity in the model is queried to find its mesh child nodes. To ensure each node has only one distinct parent only nodes that do not lie on the bounding entities of a B-Rep entity acquire that entity as their parent. To achieve the B-Rep entities with the smallest manifold dimension are addressed first. For example, all vertices in the model are interrogated to find their child node, Fig. 8 (a). Once the associativity for these vertices has been identified their nodes cannot have another parent entity assigned, despite the fact that they may lie on other B-Rep entities. This is shown in Fig. 8 (b) where the edges are interrogated to find their child nodes. Only nodes that lie within the bounding vertices of an edge are assigned as children of the edge. Therefore, it follows that B-Rep faces and volumes are assigned as the parent of any nodes lying within their bounding entities. Each element in the mesh is related to its parent cell, whose dimensionalities must match one another, i.e. solid, shell and 1D elements have B-Rep volumes, faces and edges as their respective parents.

3.3 Relating Mesh Entities to the Original Model

After the non-manifold topology of the simplified model along with any Virtual Topology and mesh-geometry ownership relationships have been stored in the master database it is possible to use this information to transfer the mesh entities with full associativity to their parent B-Rep entities in the original model. Mesh entities with a virtual entity as their parent topological entity are automatically related back to the host entity. Virtual entities include virtual subset, superset and parasite entities. Nodes lying on subset edges, faces or volumes, Fig. 9 (a) are assigned the original edges, faces and volumes, Fig. 9 (b) as their parent entity.

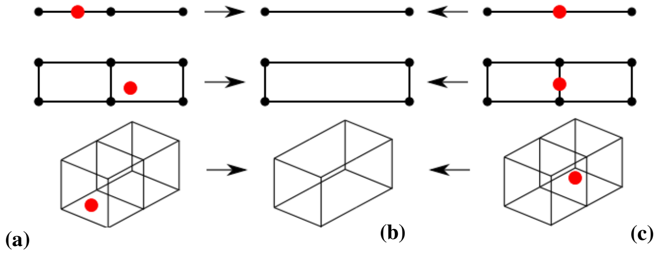


Fig. 9 (a) Node with virtual subsets as their parent, (b) The parent entity of nodes with parent subsets or parasite entities, (c) Node with parent parasite entity

Mesh nodes with parasite entities as their parent entity are assigned the host entity of the parasite entity as their parent. The host entity will normally have a manifold dimension of one more than the parasite entity. For example, the highlighted nodes in Fig. 9 (c) have parasite entities as their parent entity. Their equivalent original parent entities are shown in Fig. 9 (b). The node with a parasite vertex as its parent entity is assigned the host entity of the parasite vertex as its parent, which is the original edge. Similarly, the nodes with a parent parasite edge or face have host faces or volumes as their respective parents in the original model. The same process is used to transfer the ownership of elements between models. Elements have a parent topological entity of the same manifold dimension as the element dimensionality and will therefore only have virtual subsets or supersets as their parent entity, never parasite entities.

3.4 Contribution towards an Integrated Design Process

This section demonstrates the automatic implementation of the process for two commercial packages. The packages utilized are Abaqus for mesh generation and CADFix for creating the input file for analysis, including boundary conditions applied to entities in the original design model.

Defeaturing, dimensional reduction and decomposition tools [6, 7, and 10] are commonly used to create idealized analysis models which are less computationally expensive. There are occasions where a model may be decomposed or partitioned into idealized sub-regions in order to meet the specific meshing requirements of an analyst. A requirement for many analyses is the creation of a mesh comprised of only hexahedral elements. Robust automated hex mesh generation is still a largely unsolved problem. For example, the simple model in Fig. 10 (a) cannot be automatically hex meshed by many commercial CAE packages, although it can be automatically tet meshed, Fig. 10 (b). Some CAE packages are able to automatically subdivide a model such as this into hex meshable sub-regions, Fig. 10 (c), which can then be automatically hex meshed in that or a different package, Fig. 10 (d). The problem is that the mesh is created on a model with a different topology than the original model. This means that any analysis attributes such as loading or boundary conditions defined on the topology of the original model need to be recreated for the exported mesh. This is currently a manual process and can be time consuming to achieve for complex models. The procedures presented here allow multiple different packages to be used to create suitable analysis geometry and meshes whilst maintaining the links between them and therefore back to analysis attributes applied to any model.

In this example a decomposed model has been created and is imported into Abaqus. The choice of tool used to arrive at the subdivided analysis geometry, or destination package, does not restrict the process described in this paper. For this example the model was decomposed into hex meshable sub-regions manually. After importing the analysis model into Abaqus a non-manifold representation is created using the Abaqus Boolean Union tool. The non-manifold model is

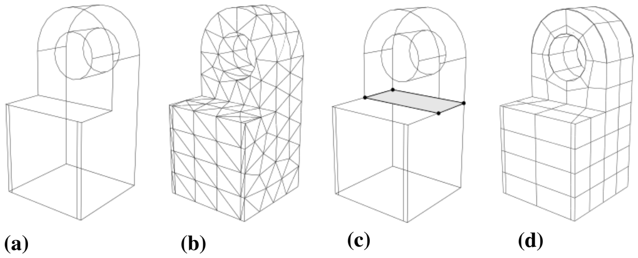


Fig. 10 Domain decomposition for a simple component: (a) Original geometry, (b) Original geometry tet meshed, (c) Subdivided analysis geometry with parasite splitting entities highlighted, (d) Idealized analysis geometry hex meshed

interrogated and its topology is extracted and stored in the database. Abaqus has an automatic Virtual Topology tool which identifies faces and edges to be merged based on a set of geometric parameters. The default parameters of this tool are sufficient to remove the unwanted sliver faces in the test model. Abaqus can return the entities that have been ignored, but as this functionality is not available in all packages the generic procedure described in Section 3.1 is used instead. Any Virtual Topology relationships are identified and stored in the database.

Abaqus was selected for this demonstration as it can automatically select suitable meshing strategies to apply to certain cells, i.e. a non-manifold representation with multiple interacting cells can be automatically meshed. After creating the mesh, all nodes and elements are stored in the Equivalence relation of the database along with their parent B-Rep entities, Section 3.2. This is achieved by using Abaqus queries to find the nodes belonging to each vertex, edge, face and volume in the non-manifold model. During this process an orphan mesh file is automatically created. The structure of the mesh file is manipulated to suit the target package. For example, element codes are changed to suit the target system, i.e. a 20 node hex element in Abaqus has an element code of C3D20R, which corresponds to the HE20 element in CADFix. Node and elements in the orphan mesh file are linked to their parent entities in the database.

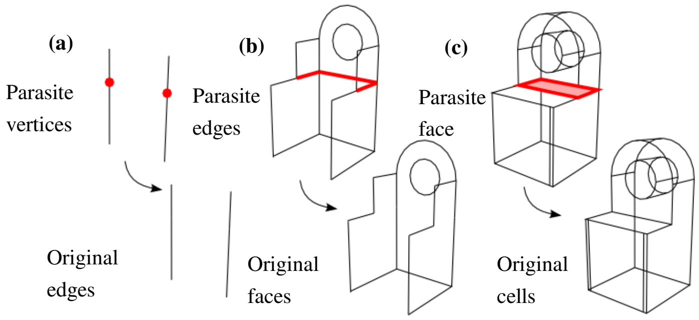


Fig. 11 Linking the original and sub-divided models: (a) Parasite vertices (highlighted) used to split original edges, (b) Parasite edges (highlighted) used to split original faces, (c) Parasite face (highlighted) used to split an original volume

With the objective of relating the mesh generated on the simplified geometric model to the original model, it is necessary to establish the correct equivalences between the models. A similar approach to that described in Section 3.1 for linking virtual and host models can be used to link the sub-divided analysis geometry, Fig. 10 (c), to the original geometry, Fig. 10 (a). Entities that exist in the non-manifold topology of the sub-divided model in the master database but not in the topology of the original model are identified as parasite entities. The manifold dimension of parasite entities is one less than the entity they partition, i.e. vertices are used to split edges, edges are used to split faces and faces are used to split volumes, Fig. 11. The identification of the non-manifold vertices, edges and faces in the sub-divided cellular model in the database enables the partitioned entities to be found and stored as subsets of their original entity, i.e. the volumes in the sub-divided analysis model are stored as subsets of their original volume. This is another application for Virtual Topology. Tags are automatically assigned to all topological entities within the database. This requires volume cells to be named upfront in order to determine subset entities. Alternatively, entity Identifiers may be used, i.e. multiple Identifiers of subset volumes will lie within the boundary of a single host entity.

Once the topology of the decomposed model in the database has been linked to the original model in CADFix, the mesh-geometry ownership can be transferred between the two models. Mesh entities having either parasite or subset entities as their B-Rep parent are linked to the equivalent host entity, Section 3.3. For example, nodes lying on subset edges, faces or volumes, Fig. 12 (a) are assigned the original edges, faces and volumes (Fig. 9) as their parent entity. Elements with parasite parent entities, Fig. 12 (b), in the simplified representation will have the original un-partitioned host cell as their parent in the original representation, Fig. 12 (c). This completes the transfer of mesh-geometry ownership between models at different levels of fidelity. Analysis attributes assigned to topological entities in the original model can be automatically applied to the correct (equivalent) mesh entities, allowing analysis input files to be generated.

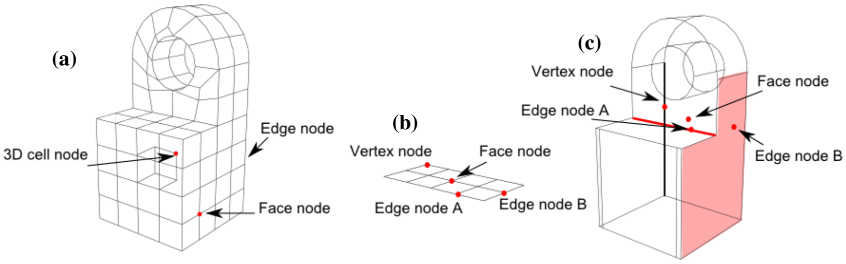


Fig. 12 Transferring mesh-geometry ownership: (a) Nodes with subset parent entities, (b) Nodes with parasite parents, (c) Equivalent original parent entities

The automated tool described in this section enables different packages to be selected based upon their strengths for specific analysis processes. Here, Abaqus has been selected for its meshing capabilities and CADFix has been selected as it can provide the input file for different analysis packages. The mesh generated in Abaqus on simplified analysis geometry has been automatically transferred to the original model in CADFix.

4 Discussion

This paper introduces a method to transfer finite element meshes at different levels of fidelity between different packages by identifying and storing the links between equivalent geometries. The underlying theme is to track the equivalences between the original design model, analysis geometries at various levels of detail and abstraction, and the finite element meshes generated on them. The memory overhead for the database is insignificant for the small model presented here. For industrial design applications the database memory overhead will depend on component complexity, the number of equivalent representations and the meshing attributes used. It is anticipated this overhead will be small relative to the corresponding CAD and CAE data files and structures.

Ideally the database described in this article would interface directly with all design, pre-processing and solving tools making it easier to store the required topological links simultaneously with the ongoing design activities. However, in this work the interfacing has been prototyped using the scripting interfaces to certain packages used in the design process. It has been shown that the links between different design and analysis geometries can be identified without having to interact with all packages in the design process. For example, the database can identify the links between the design and analysis geometry without having to interact with the simplification tools used to create the analysis geometry. This highlights the robustness of this approach, enabling the desired tools to be used for certain applications without having to worry about the downstream effect.

The automated tools show how design and analysis models from different systems can be linked to each other. Due to the different underlying data structures of CAD and CAE packages new topological entities may be introduced to partition seamless edges and faces. These changes can be identified by comparing model topologies and creating suitable virtual subsets and supersets. Although the models can have different topologies due to these changes, they are considered equivalent and creating suitable virtual entities enables them to be treated as such.

Another method to find the Virtual Topology relationships described in Section 3.1 is to use the Identifier of a topological entity. If a package contains geometric searching functionality the Identifier information in the database can be used to identify entities that have been merged together. The Identifier contains the coordinates of a position within the boundary of an entity and can be used to find the closest entity located at those coordinates. A simple query on the database returns the Identifier of all edges and faces in the host model. The virtual model is interrogated to find the closest edges and faces for each Identifier. Entities in the virtual model containing multiple host Identifiers are superset entities.

In order to demonstrate the effectiveness of the proposed approach Section 4 describes how analysis geometry can be automatically linked to its equivalent original representation, allowing for robust transfer of the mesh. Once appropriate models have been created the approach is fully automated and can be used to establish the equivalence between any number of CAD or analysis models from a variety of sources. While the analysis geometry used in this work is the output from a decomposition tool, other simplified models can also be integrated using the proposed approach. For example, a de-featured model could be linked to its original model using the same topological comparison approach. In terms of linking dimensionally reduced entities more details are provided in [16] and [7]. Once relationships have been established between dimensionally reduced cells and their equivalent original representation, mesh parentage relationships can be transferred between the models as described throughout this paper. An application to models of industrial complexity is described in [6].

Robust transfer of meshes between different models and packages enables boundary conditions and other analysis attributes applied to the original model to be automatically transferred to a mesh generated on abstract analysis geometry. During automatic Virtual Topology operations there is the possibility that topological entities with boundary conditions applied may be merged with adjacent entities. It has been assumed in this work that entities with boundary conditions applied will not be involved in downstream clean-up operations. In practice it may be necessary to store the original topology in the database along with pointers to entities with boundary conditions applied to ensure these entities would be preserved. Additionally, boundary conditions can be attached to virtual entities, where for example virtual subsets may be created to apply contact between faces.

The procedures described in this paper could be used for non-conforming interface regions, i.e. hex and tet meshes meeting at a common face results in a non-conforming mesh at the interface. One solution to the non-conformity problem is the node insertion method [9]. Hex elements at a non-conforming interface can be partitioned by inserting a node at the centroids, generating two tet elements and five pyramid elements, creating conformity at the interface. The non-manifold model enables interface elements to be easily identified so that they can be partitioned. Once partitioned, the resulting tet and pyramid elements could be considered as subsets of the original hex element. In cases where it is desirable to preserve the hex elements at the interface, conformity may be achieved by merging adjacent tet elements together. The new pyramid elements could be stored as virtual supersets of the original tet elements. Therefore, Virtual Topology could be used to store different decompositions of equivalent meshes.

5 Conclusions

Novel techniques have been presented in this paper to facilitate the transfer of finite element meshes between equivalent analysis geometries residing in different

CAE packages without any loss of integrity. This is achieved by tracking the equivalences between all cells in the equivalent design and analysis models. Virtual Topology merging and partitioning are used to create a cellular decomposition of the design space, which provides a framework upon which to specify all the necessary analysis attributes.

In order to transfer the mesh between packages, the simplified analysis model is linked to its equivalent design model. Equivalent relationships are stored in a robust manner so they can be reused downstream in different packages, enabling mesh-geometry ownership to be transferred between the models at different levels of abstraction and fidelity. Therefore, boundary conditions applied to an original model can be automatically transferred to the finite element mesh generated on abstract analysis geometry. This enables toolsets of choice to be selected by an analyst without having to manually link models, resulting in an integrated analysis process.

References

1. Arabashi, S.D., Barton, C., Shaw, N.K.: Steps towards cadfea integration. Engineering with Computers 9(1), 17–26 (1993)
2. Beall, M.W., Shephard, M.S.: Accessing CAD geometry for mesh generation. In: 12th International Meshing Roundtable, Santa Fe (2003)
3. Cavalcanti, P.R., Carvalho, P.C., Martha, L.F.: Non-manifold modeling: An approach based on spatial subdivision. Computer-Aided Design 29(3), 209–220 (1997)
4. Harlin, G.: Engineering value of simulation process and data-management applied to aero engine design. In: Nafems World Congress, Salzburg (2013)
5. Lee, S.H.: A CAD-CAE integration approach using feature based multi-resolution and multi-abstraction modeling techniques. Computer-Aided Design, 941–955 (2005)
6. Makem, J.E., Armstrong, C.G., Robinson, T.T.: Automatic decomposition and efficient semi-structured meshing of complex solids. Engineering with Computers (2012), doi:10.1007/s00366-012-0302-x
7. 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/s00366-013-0317-2013
8. Owen, S.J., Shepherd, J.F.: Embedding features in a Cartesian grid. In: 18th International Roundtable, Salt Lake City (2009)
9. Owen, S.J., Cannan, S.A., Saigal, S.: Pyramid elements for maintaining tetrahedra to hexahedra conformity. ASME, 123–129 (1997)
10. 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), 151–165 (2011)
11. Sheffer, A., Blacker, T., Bercovier, M.: Virtual Topology Operators for Meshing. International Journal of Computational Geometry and Applications 10(3), 309–331 (2000)
12. Shephard, M.S., Beall, M.W., O'Bara, R.M., Webster, B.E.: Toward simulation-based design. Finite Elements in Analysis and Design, 1575–1598 (2004)

13. Sypkens Smit, M., Bronsvoort, W.F.: Integration of Design and Analysis Models. Computer-Aided Design and Applications, 795–808 (2009)
14. Sypkens Smit, M., Bronsvoort, W.F.: Efficient tetrahedral remeshing of feature models for finite element analysis. Engineering with Computers 25, 327–344 (2009)
15. Thakur, A., Banerjee, A.G., Gupta, S.K.: A survey of CAD model simplification techniques for physics-based simulation applications. Computer-Aided Design 41, 65–80 (2009)
16. Tierney, C.M., Nolan, D.C., Robinson, T.T., Armstrong, C.G.: Managing equivalent representations of design and analysis models. Computer-Aided Design and Applications (2013)
17. Weiler, K.: The Radial Edge Structure. In: Geometric Modeling for CAD applications, pp. 3–36. North-Holland (1988)
18. White, D., Saigal, S.: Improved imprint and merge for conformal meshing. In: 11th International Meshing Roundtable, New York (2002)