Interface management for automating finite element analysis workflows

Tierney, C. M., Nolan, D. C., Robinson, T. T., & Armstrong, C. G. (2016). Interface management for automating finite element analysis workflows. *Computer-Aided Design and Applications*, *13*(3), 329-339. https://doi.org/10.1080/16864360.2015.1114391

Published in:
Computer-Aided Design and Applications

Document Version:
Peer reviewed version

Interface management for automating finite element analysis workflows

Christopher M. Tierney [0000-0003-3341-6902][1], Declan Nolan[2], Trevor T. Robinson [0000-0002-6595-6308][3] and Cecil G. Armstrong [0000-0001-8695-5016][4]

[1]Queens University Belfast, christopher.tierney@qub.ac.uk
[2] Queens University Belfast, dnolan02@qub.ac.uk
[3] Queens University Belfast, t.robinson@qub.ac.uk
[4] Queens University Belfast, c.armstrong@qub.ac.uk

ABSTRACT

This paper outlines the importance of robust interface management for facilitating finite element analysis workflows. Topological equivalences between analysis model representations are identified and maintained in an editable and accessible manner. The model and its interfaces are automatically represented using an analysis-specific cellular decomposition of the design space. Rework of boundary conditions following changes to the design geometry or the analysis idealization can be minimized by tracking interface dependencies. Utilizing this information with the Simulation Intent specified by an analyst, automated decisions can be made to process the interface information required to rebuild analysis models. Through this work automated boundary condition application is realized within multi-component, multi-resolution and multi-fidelity analysis workflows.

Keywords: Interface management, cellular modelling, virtual topology, equivalencing, idealization, decomposition

1 INTRODUCTION

Solving computational analysis problems often requires analysis specific geometric representations to be generated to facilitate the creation of a fit-for-purpose finite element mesh. Aside from mesh generation, any applicable analysis attributes (e.g. boundary conditions, material properties) are attached to relevant geometric entities of the idealized analysis geometry. Boundary condition loads and/or constraints application is often a manual procedure, carried out through the manual selection of specific geometric faces defining the domain boundary to which the boundary conditions are to be applied. Effective management of boundary conditions is required to maximize the efficiency of CAD-CAE integrations during product development cycles (PDC). Boundary conditions need to be updated to reflect changes in the design, e.g. where new geometric entities are added to the design model or changes in the analysis representation, e.g. from 3D solid geometry to shell mid-surface representation. Therefore, it is logical that a change in dimensionality of the boundary condition definition may be necessary, e.g. replacing distributed face loads with equivalent point or edge loads. Current CAE solutions do not offer suitable functionality to track and manipulate boundary conditions throughout a PDC, leading to the requirement for new methodologies to help automate this aspect of finite element analysis workflows. Such functionality can be facilitated through effective interface management, where interfaces are the geometric interactions used to define boundary conditions.

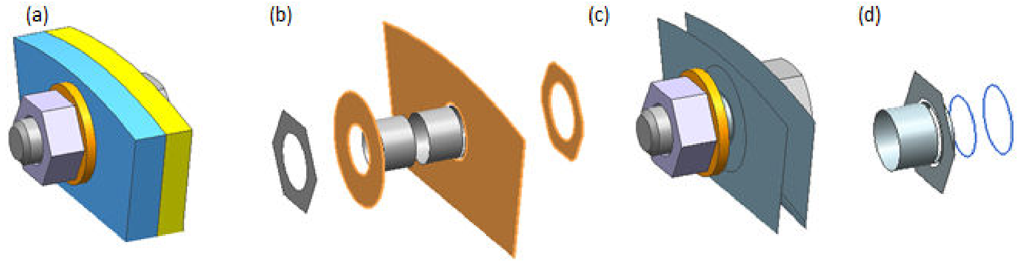

Fig. 1: (a) 3D fastened plate assembly; (b) Interfaces in 3D fastened plate assembly; (c) Mid-surface representation of fastened plate assembly; (d) Interfaces in dimensionally-reduced fastened plate assembly.

Geometric interfaces in multi-component assembly models describe the connectivity between adjacent components. In FEA assemblies any explicit interface definitions are translated to appropriate boundary and contact conditions to capture the physical and mechanical properties to be transferred between components. Fig. 1 (a) and (b) show a simple bolted flange assembly and the interfaces in the assembly respectively. The application of boundary conditions for large assemblies, like whole aero-engine thermo-mechanical models, can be a tedious and time consuming task due to the vast number of physical interactions present. The manual effort required to define boundary conditions becomes repetitive and does not add value during multiple design changes through the product development, where validating and updating the assembly feature interfaces becomes necessary to maintain the integrity of the analysis model. Fig. 1 (c) and (d) illustrate the transformation in interface definition once the dimensionality of the geometric representation of the fastened plates has been modified. The interfaces with the dimensionally reduced plates and the fastener assembly have either reduced in dimensionality, from a face to an edge interface (blue edges in Fig. 1 (d)), or no longer exist because there is no longer physical contact between the geometric component models, the plate-plate, plate-washer and plate-bolt interfaces highlighted from Fig. 1 (b). Solutions are needed to automatically identify these interface transformations and to determine how they shall be treated in order to generate a valid analysis model. For example, where a shell meshed region meets a solid meshed region, multi-point constraint equations (MPCs) can be used to connect the meshes or constraints can be specified to achieve mesh conformity at the interface.

Different analysis objectives can demand a combination of these pre-processing procedures to be used on the same component. The processing of geometric models for computational analysis means that the associativity between the downstream models is unavailable leaving it challenging to manage the flow of analysis information (boundary conditions, results) between different stages of the PDC. Associated links are required between the different geometric and analysis models in order to facilitate the automatic transfer of analysis information between the equivalent representations. These links must propagate between preferred CAD or CAE toolsets within design optimization frameworks. Considering assembly, sub-assembly and single component analyses at various levels of mesh resolution and fidelity, interface management becomes increasingly difficult to maintain in an automated fashion during a PDC.

In this work the aim is to automate analysis model workflows through robust interface management by using three known technologies (Cellular Modelling, Virtual Topology and Equivalencing) to manage the dependencies between equivalent design and analysis models [21]. Coupled with the 'Simulation Intent' concepts [11] many more of the decisions required for treating interfaces and creating appropriate analysis models can be automated. This work provides automatic boundary condition application within multi-component, multi-resolution and multi-fidelity analysis workflows.

2 RELATED WORK

In previous work Shahwan et al. [16] describe how interfaces are essential to derive the functional information required to progress from a Digital Mock-Up (DMU) to an appropriate analysis model. This work focused on using shape properties of geometric interfaces between components in assemblies to classify functional interfaces, which in turn dictate the transformations required to capture the behavior between components. This information is utilized to enrich the DMU to help automate downstream analysis transformations. Boussuge [4] used the geometric and functional interfaces derived by Shahwan [16] to generate user-defined templates for processing repetitive arrangements, such as fastener assemblies, for FE assembly modelling. Boussuge implemented template-based shape transformations on a 250 component assembly and reduced the analysis model preparation time from 25 days to 1.5 hours. It is evident from these statistics that interface management is an essential aspect of automating the transformation process between design and analysis models. The template-based approach of Boussuge is used to automate many of the decisions required to treat interfaces and create analysis models, but his work focused on fastener assemblies and did not address the issue of compatibility between different CAE platforms.

Gostaf [7] described the re-use of assembly topology to define interface regions for the application of boundary conditions in multi-component assemblies. Defining contact regions using assembly constraints requires post-processing, such as imprinting, in situations where only partial contact interfaces exist. Furthermore, there is no guarantee suitable assembly constraints exist, especially in DMU representations where components tend to be positioned at the correct location in 3D space, but the relevant spatial constraints are not explicitly known. Therefore, much work is required to define a logical description of the interfaces between all components in an assembly.

An approach for the mixed-dimensional coupling of analysis models is described in [6]. Dimensional interfaces describe the connection between adjacent regions of different dimensionality and are defined by using face-splitting techniques to create the topological link between adjacent regions, e.g. imprinting the edge of a mid-surface onto the bounding face of an adjacent solid, which dictates the mesh pattern on the adjacent solid domain. Shephard [18] uses the same approach to generate a connection at a mixed-dimensional interface. The imprinting procedure can be avoided using the equivalencing techniques presented in this paper, removing the need for editing the geometry. This is achieved by managing the topological dependencies linking a dimensionally reduced cell to its original representation and transforming the original interface to the required fidelity.

Shephard et al. [18] describe methods to integrate tools between CAD and CAE disciplines. This approach does not link the different design and analysis models, but attempts to automate the different tools required to derive a model for a downstream analysis application. The design model has to be defined upfront and linked to an abstract component model. Different tools deal with the creation of idealized analysis geometry, creating the mesh and utilizing the analysis results. The model management tool stores the topology of the model, but no details are provided on how this topology is maintained. In this work the associativity between equivalent analysis model representations is identified and stored. The data structure used is similar to modeler independent interfaces, such as the CGM [20] or Djinn API [3], which focus on the robust identification and access of entities across platforms. However, in this implementation geometric entities are neglected as only the model topology is required to define the required connectivity between analysis representations. This connectivity information is used to generate the bi-directional links Arabshahi [2] described as necessary for seamless CAD-CAE integration.

3 MANAGING TOPOLOGICAL DEPENDENCIES BETWEEN EQUIVALENT ANALYSIS MODEL REPRESENTATIONS

A precursor to this work was introduced in [21], where links between many different design and analysis models of the same components are identified and stored in an independent data structure enabling the topological information to be interrogated and modified without disturbing the original geometric representation. Three technologies are employed to identify and process the dependencies between equivalent analysis configurations, Fig. 2 (a). **Cellular modelling** is used to store the non-manifold representation of analysis-specific decompositions of the design space. Each cell in the cellular representation has specific simulation significance. Links to the parent / child entity of each cell are maintained. The topology of the non-manifold cellular model can be used to identify interfaces between interacting cells. **Virtual topology** techniques are used to record where partitioning and merging operations have been applied to a model. They operate by modifying the topology of the model, thus avoiding error prone and time consuming geometric modifications. Traditional uses for virtual topology are often limited to geometry clean-up for meshing [17]. However, treating virtual topology as an integral component of an analysis cycle provides a robust framework for model pre-processing where multiple analysis representations may be linked using the virtual topology hierarchy. **Equivalencing** describes the dependencies between equivalent regions of space, which may have a different geometric representation for different analyses, e.g. a mid-surface shell representation of a thin-walled solid. Tracking the topological equivalences provides the opportunity to modify existing boundary conditions when idealizations occur. For example, after the dimensional reduction of a model it would be necessary to dimensionally reduce the associated boundary conditions to maintain solution accuracy [9]. More detail on the identification of links between design and analysis models and the generation of the data structure can be found in [21].

3.1 Simulation Intent

Defining Simulation Intent involves capturing high level modelling and idealization decisions in order to create an efficient and fit–for–purpose analysis. These decisions are recorded as attributes of the decomposed design space [11]. These decisions can include, but are not limited to, the physics to be solved, the required accuracy, mesh type and dimensionality, mesh mating conditions and boundary condition application. The aim is to specify 'Simulation Intent' [11] in a manner that is independent of the underlying geometry. This enables many low-level decisions and operations to be automated and manual rework to be avoided. For example, Fig. 2 (b) where boundary conditions are automatically transferred between equivalent abstract analysis models defined by applying different 'Simulation Intent' to the design model.

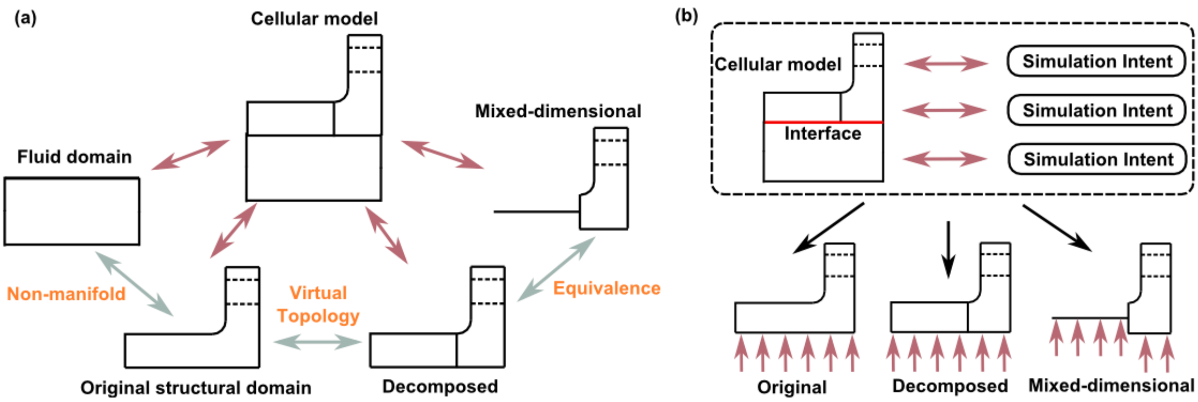

Fig. 2: (a) Technologies used to link analysis representations; (b) Linking with Simulation Intent to create fit-for-purpose analysis models at different levels of fidelity.

4 TRANSFERRING ANALYSIS ATTRIBUTES BETWEEN ANALYSIS MODELS AT VARIOUS LEVELS OF FIDELITY AND RESOLUTION

Interchanging models at various levels of fidelity is required at different stages of a design process, or depending on the physics to be solved. Here it is demonstrated how boundary conditions can be transferred between the three different representations of the same component, shown in Fig. 3.

The analysis decomposition of the structural casing component in Fig. 3 (a) is an automated process carried out by tools developed in [14] and [8], to isolate regions which exhibit certain geometric characteristics which lend themselves to specific meshing styles, namely thin-sheet (green), long-slender (blue) and residual (yellow) regions, Fig. 3 (b). The initial decomposition is used to define a mixed-solid element mesh for a stress analysis, where thin-walled (referenced '_2D' in Fig. 3) and long-slender ('_1D') cells are automatically meshed using hex elements (by sweeping a quad mesh on a source face through the thickness, or along the length, of the region) and the residual region ('_3D') meshed with tet elements, Fig. 7 (b). Hex elements are more efficient than tet elements in thin-walled regions as the restrictions on aspect ratio for an accurate solution means a greater number of tet elements are required to mesh the same domain. However, the robust nature of tet mesh generation algorithms enable tet meshes to be generated for complex geometries. The resulting decomposition also lends itself to mixed-dimensional analysis modelling, where thin-sheet and long-slender cells are easily idealized to mid-surface and mid-line representations respectively, Fig. 3 (c). The analysis attribute of a cell is automatically stored in its name attribute during decomposition, Fig. 3 (b). For example '_2D' signifies a thin-sheet cell that can be idealized to a mid-surface.

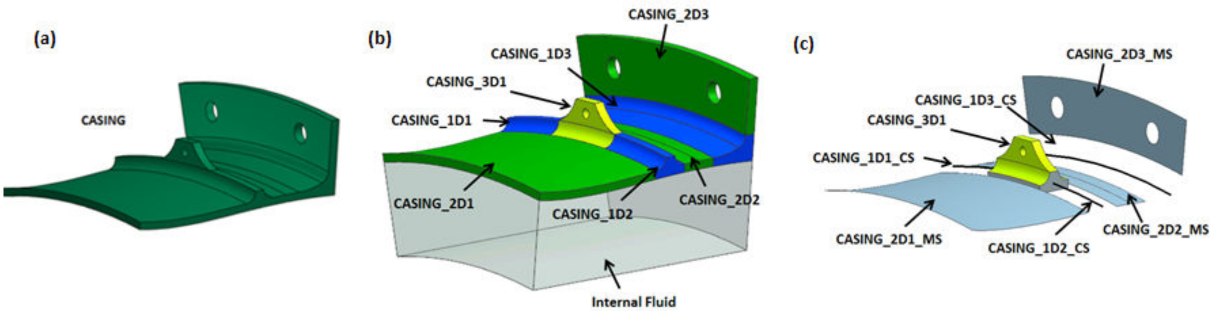

Fig. 3: (a) Aero-casing section; (b) Cellular model of 3D analysis decomposition; (c) Lower fidelity mixed-dimensional model.

The partitioning of the design space to create the cellular model also includes fluid domains, Fig. 3 (b). Explicit representations of fluid domains are essential for multi-disciplinary analyses where results from a CFD analysis can be supplied as the input to structural, thermal or stress analyses. Another benefit of having an explicit definition of fluid domains within the cellular model is exploited in this work to automate the application of boundary conditions at the solid-fluid interface. As an alternative to manually specifying the boundary conditions, it is possible to utilize a Simulation Intent attribute that is independent of the underlying topological entities, which specifies that a pressure load is to be applied on the structural faces at the interface with the internal fluid domain Fig. 3 (b).

$$\text{Pressure Interface} = \text{CASING} \cap \text{INTERNAL FLUID} \tag{1}$$

The non-manifold cellular model provides a geometric framework which can be used to identify the interfaces between interacting cells. An example is its use to highlight the interfaces in Fig. 2 (b). Shahwan [16] extracted interfaces using bounding boxes of components in a DMU to filter non-adjacent components and subsequently check for geometric interactions. Sung [19] used an octree approach to locate assembly interactions for disassembly sequence generation. These interfaces are readily available in the non-manifold cellular model. Interacting volume cells in the non-manifold cellular representation are bounded by the same faces at their interface, albeit with opposing orientations. Therefore, the definition of the interface between two bodies is simply the set intersection, $\cap$, of both sets of bounding faces, Equation 1. In this work the set intersection is carried out using an SQL query on the topology of the non-manifold cellular model, stored within the data structure described in [21].

4.1 Automatic transfer of boundary conditions between analysis representations

Interfaces identified within the cellular model are transferred between the equivalent representations in Fig. 3, by utilizing the model dependencies between the representations. The dependencies between original and decomposed cells are managed using Virtual Topology, where decomposed cells are recorded as subsets of the original cell. In the examples in this section the interface calculations and creation of the non-manifold representation have been performed automatically using a prototype application developed on top of the Parasolid geometry kernel [13]. The original and decomposed analysis model representations were automatically generated within the Siemens NX environment [12].

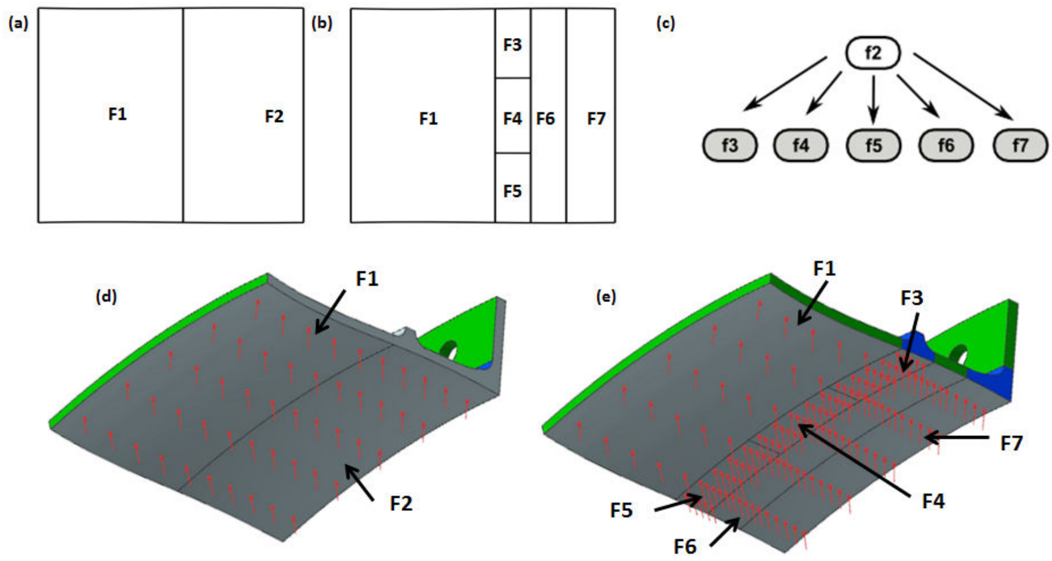

Fig. 4: (a) Original solid-fluid interface region; (b) Equivalent interface post model decomposition; (c) Virtual Topology hierarchical relationship between original and decomposed faces; Pressure load on (d) analysis representation of original model; (e) analysis representation of the decomposed model.

The calculated pressure interface between the 'CASING' and 'Internal Fluid' domain (Equation 1) is represented by the faces in Fig. 4 (a). These faces are considered equivalent to the collection of partitioned faces in Fig. 4 (b) created as a result of a model update. In this work the partitioned faces are stored as virtual subset entities (f3, f4, f5, f6 and f7) which reference their original host entity (f2), as shown in the virtual topology relation diagram in Fig. 4 (c). Fig. 5 shows the data structure entries for the volume cells and interface faces from Fig. 3 and Fig. 4. Storing the virtual topology in a separate relation enables the different topological descriptions of the original and decomposed analysis models to be simultaneously represented within the data structure whilst maintaining the links between them. Generating an analysis-specific topology allows multiple distinct analysis representations to be linked to the original model. The topological relationships between all lower dimensional entities, such as decomposed edges, are also maintained within the virtual topology framework. Using this Virtual Topology information the interface information can be automatically transferred between the original Fig. 4 (d) and decomposed models in Fig. 4 (e). Since Siemens NX is a manifold modelling environment, the manifold name attribute is required to successfully apply the pressure attribute to both the original and decomposed model representations. Also, no consideration of boundary condition orientation is required for manifold models as the surface normal always points away from the volume cell.

Whilst in this example the focus is on the transfer of boundary conditions, it should be noted that any analysis information associated with the equivalent topological entities can be transferred between the representations. For example, consider a common thermo-mechanical analysis workflow for an aero-component. The same defeaturing operations are performed prior to the thermal and structural analyses. An additional decomposition step is performed prior to the structural analysis. Therefore, in order to transfer results between thermal and structural domains it is necessary to manually tag appropriate interface regions. This can be avoided using the approach in this work as explicit relationships are maintained between the models required by the two domains.

Original Topology	
Entity	**Bound Entity**
CASING	F1
CASING	F2

Decomposition →

Virtual Topology	
Host Entity	Virtual Entity
CASING	CASING_2D1
CASING	CASING_1D1
CASING	CASING_1D2
CASING	CASING_3D1
CASING	CASING_2D2
CASING	CASING_1D3
CASING	CASING_2D3
F2	F3
F2	F4
F2	F5
F2	F6
F2	F7

← **Analysis-specific topology**

Analysis Topology	
Entity	**Bound Entity**
CASING_2D1	F1
CASING_1D1	F5
CASING_1D2	F3
CASING_3D1	F4
CASING_2D2	F6
CASING_1D3	F7

Fig. 5: Data structure entries for virtual topology volume cells and interface faces after the model decomposition in Fig. 3.

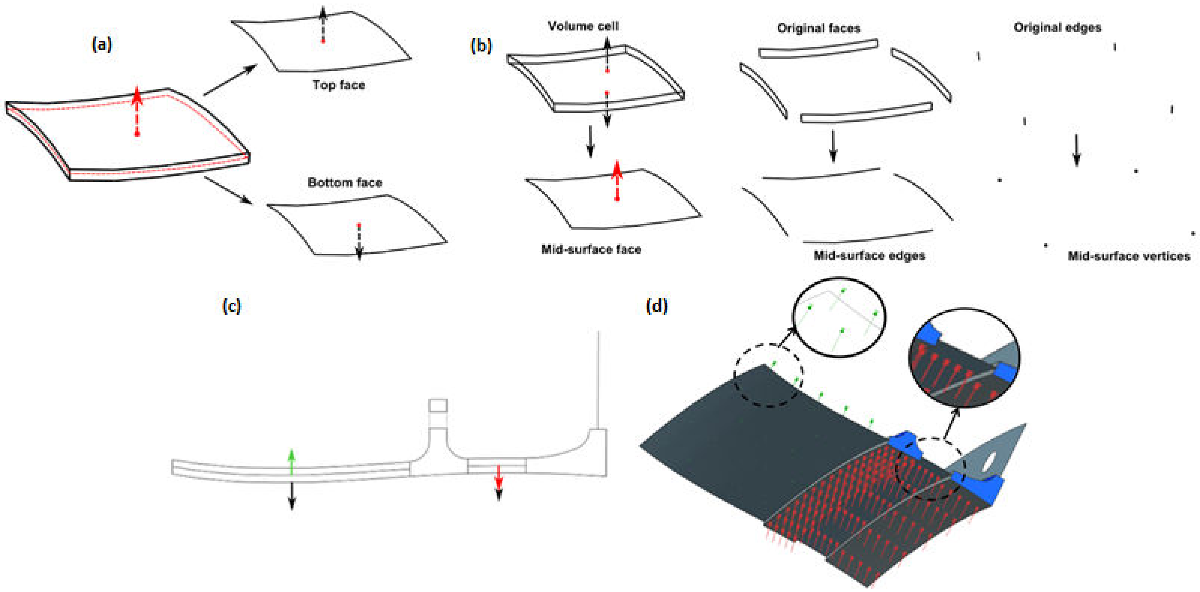

Fig. 6: (a) Thin-sheet region and defining face pairs of mid-surface; (b) Equivalence dependencies for mid-surface representation; (c) Comparison of surface normals between face pairs and their equivalent mid-surface face (d) Pressure load on mixed-dimensional model with correct orientations.

Identification of dependencies between the idealized entities and their equivalent entities in the detailed representation is described in [21] and illustrated in Fig. 6 (a) and (b), where the dependencies of all bounding entities of the mid-surface are defined. Although no explicit interface exists between the fluid domain and the mid-surface representations, equivalence links with 3D interfaces in the cellular model are used to transfer the pressure load to the idealized interfaces in the mixed-dimensional model, Fig. 6 (d). Fig. 6 (c) shows the equivalence between top and bottom faces of a 3D thin-sheet region. The mid-surface face has an orientation attribute defining whether the surface normals point in the same direction. This allows the boundary conditions to be applied in the correct orientation on the mid-surface, inset Fig. 6 (d). This provides additional functionality compared to existing analysis workflows, where switches in model fidelity require boundary conditions to be manually updated. For this example the mixed-dimensional model is automatically generated in Abaqus CAE [1] to emphasize the ability to transfer models and analysis attributes between different packages.

4.2 Connecting independently meshed domains at their interface

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 the interacting volume cells in the cellular representation. The attributes attached are used to determine the type of boundary conditions to be applied after analysis transformations. Different strategies for coupling a mixed-solid mesh are automatically defined for each interface type in Fig. 7 (a), where: 2D-1D interfaces signify hex-hex conformity and may be used to enforce and merge coincident nodes, Fig. 7 (b) inset right; 1D-3D and 2D-3D signify hex-tet interfaces which can be coupled using MPCs in situations where different mesh structures can be used either side of the interface, Fig. 7 (b) inset center, or using pyramid elements to transition from hex to tet in situations where a conforming mesh is required, Fig. 7 (b) inset left.

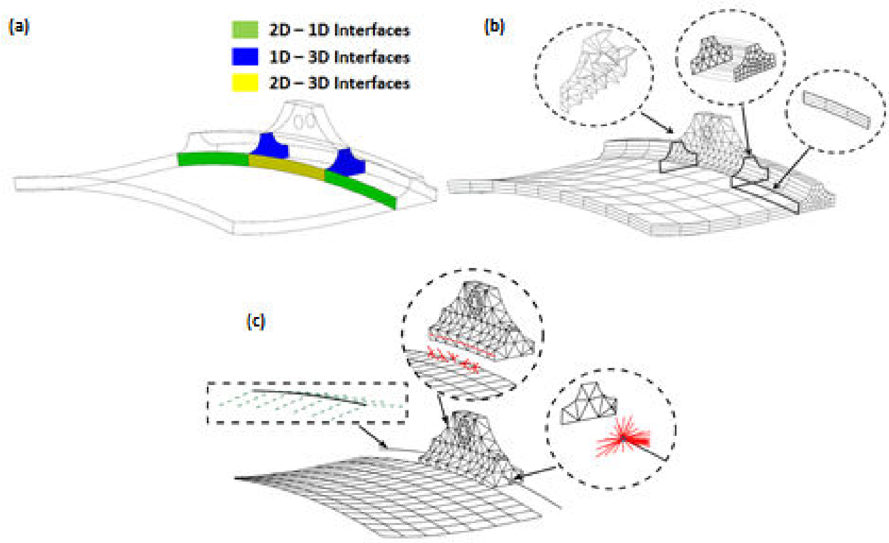

Fig. 7: (a) Highlighted interfaces of a section of the aero-casing component; Interface meshing for (b) mixed-solid mesh; (c) mixed-dimensional mesh.

Another simulation intent definition may require an idealized analysis model for modal analysis, Fig. 7 (c), where thin-walled regions are reduced to a mid-surface and meshed with a shell mesh of quadrilateral elements, long-slender regions are reduced to a line along their centroid and meshed as beams with associated cross-sectional properties. The residual region is meshed with tet elements. The equivalence links and the 3D cellular modelling interfaces are used to define the mixed-dimensional coupling strategies required to account for the rotational degrees of freedom present in shell or beam elements, which are absent from the solid element models, where: 2D-1D interfaces require edge-to-edge MPC connections, Fig. 7 (c) inset left; 1D-3D interfaces require point-to-face MPC connections, Fig. 7 (c) inset right; 2D-3D interfaces require edge-to-face MPC connections, Fig. 7 (c) inset center, where the mid-surface does not need imprinted (red dashed line on 3D mesh) in order to connect the meshed domains. These MPC's can be identified and applied even though there is no physical connection between the idealized geometric entities. Robust management of model interfaces also allows mesh sizing controls to be assigned either side of an interface region in order to achieve mesh conformity, or to simplify any MPC connections. The work described in this paper is utilized as the underlying technology for the Simulation Intent framework [10], to automatically apply the appropriate coupling strategy to adjacent cells in a mixed-dimensional model. This automated process was demonstrated on a component of industrial complexity, Fig. 8, for the CRESCENDO European aerospace project [5].

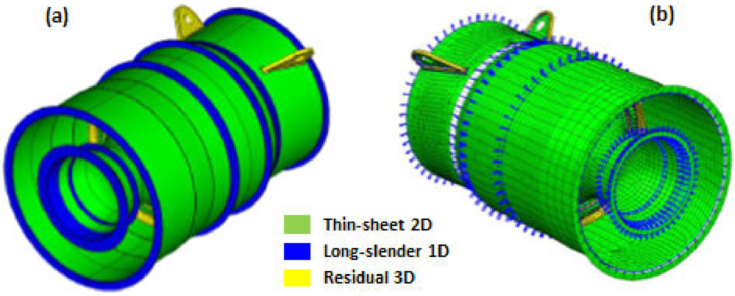

Fig. 8: Crescendo engine intermediate casing (a) Cellular decomposition [8], (b) Mixed-dimensional analysis model [10].

4.3 Assembly interface management

Representing an assembly as a cellular model allows the interfaces between different components in an assembly to facilitate the switch in fidelity between component, sub-assembly and assembly levels. For example, the force exerted on a component at assembly level may be used as the input to perform a detailed stress analysis at the component level.

Interface characteristics may be retrieved from their parent components and their associated analysis attributes to perform the appropriate interface transformation due to design updates or changes in model fidelity. Assembly interfaces are treated as either global or local interface definitions. For example, the contact boundary conditions between structural components are considered as local interface definitions which are highlighted green Fig. 9 (d, e, and f). The contact interface query differs from that required to return the pressure interface. In this instance both interacting faces are needed for the contact boundary condition, whereas for pressure boundary conditions only the interface faces to which the pressure is being applied need returned.

Consider the example of the cyclic-symmetric sector of a bolted flange connection in Fig. 9, where the structural response due to the pressure exerted by the internal fluid flow is to be assessed. The bolted connection contains five components (CASINGA, CASINGB, NUT, BOLT, WASHER), some of which have been partitioned for meshing purposes, i.e. CASINGA_2D and CASINGA_3D. To identify the pressure interface between the structural assembly and the adjacent fluid domain a separate query would be required to determine the interfaces for each component, including the partitioned components. Multiple queries can be avoided by grouping structural components together into a virtual superset called 'ENGINE', where:

$$\text{ENGINE} = \text{CASINGA} \cup \text{CASINGB} \cup \text{NUT} \cup \text{BOLT} \cup \text{WASHER} \tag{2}$$

In this instance only the uncommon boundaries (the faces bounding only one of the cells representing the components being merged) between entities being merged need to be returned. This uncommon boundary does not need to be merged as the virtual entity is required solely for interface calculations. Using this approach it takes a single query to identify the pressure interface between the structural and internal fluid cells. This provides a simple method for identifying interfaces where there are numerous identical components, such as fastener assemblies, or where the analyst is unsure which components interact with one another. In addition, where interfaces are identified for identical components, the same processing technique can be applied to each interface. Current analysis workflows encounter issues when attempting to propagate boundary conditions after design changes due to the tag (application specific identifier) of an entity changing, or new tags being introduced due to the change. These issues can be avoided by specifying the boundary condition, in this case the pressure load, as the calculated interface between a set of components, Fig. 9 (d). Therefore, it is simple to re-evaluate the interface after any design changes, Fig. 9 (e). Removing the need to manually reapply boundary conditions after design changes, along with the ability to group components into a single virtual entity, reduces the overall cost of carrying out the interface calculations, especially for multiple design iterations. In addition, where the Simulation Intent of an analyst specifies a change in assembly model fidelity, the boundary condition transformations are automatically processed within this framework, Fig. 9 (a, b and c), by tracking the equivalent interface transformations.

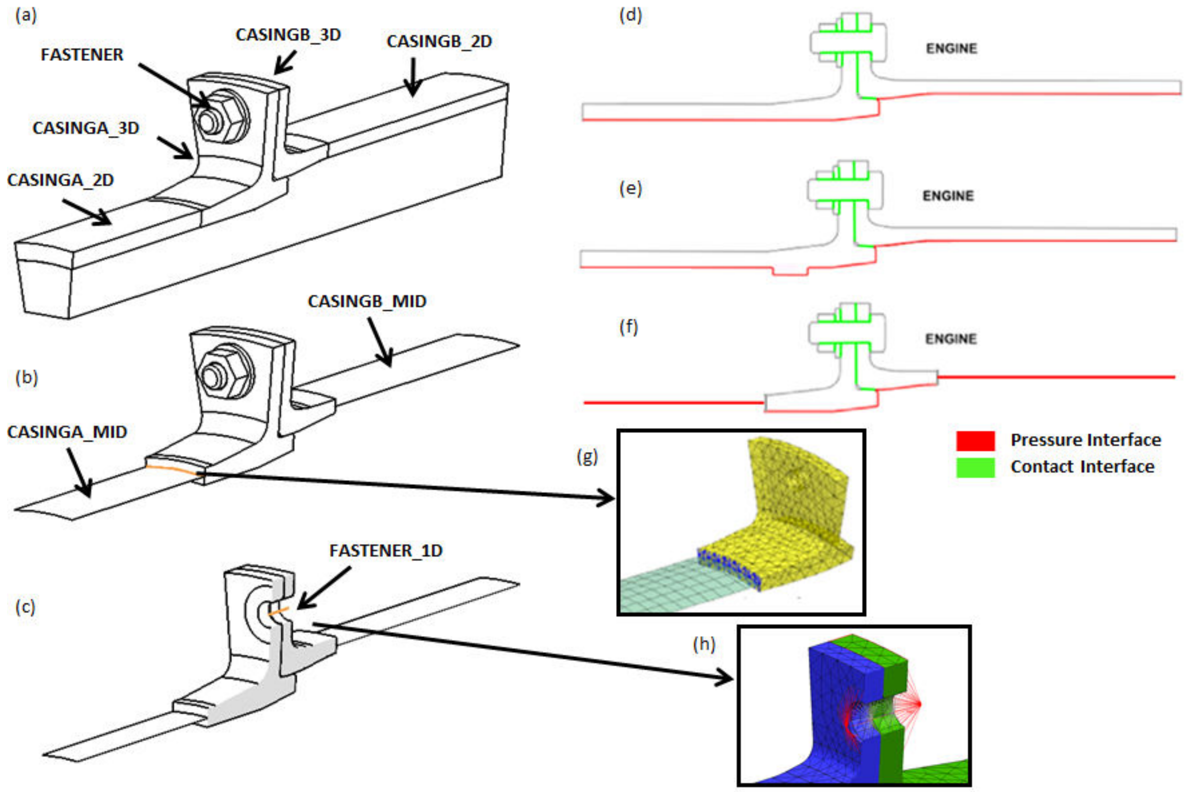

Fig. 9: (a) Decomposed cyclic symmetric sector of bolted flange; (b) 2D idealization of thin-sheet ('_2D') regions; (c) 1D idealization of fastener assembly; Interfaces in: (d) 3D representation; (e) 3D representation after design changes; (f) 2D idealization; (g) 2D-3D coupling; (h) 1D-3D coupling.

5 DISCUSSION AND FUTURE WORK

Automatic boundary condition application within multi-component, multi-resolution and multi-fidelity analysis workflows has been demonstrated in this paper. Calculated interfaces in the non-manifold cellular model are used with Virtual Topology and Equivalencing information to link boundary conditions between different representations and disciplines without loss of integrity. This enables analysis information to be automatically transferred between different stages of a design process without the burden of significant manual operations. Virtual Topology and Equivalencing information maintained within the data structure has been used in the example of Fig. 9 to transfer calculated pressure loads between original, decomposed and mixed-dimensional models. In other work the same procedures have been used to enable equivalent mesh representations, to be automatically transferred between different CAE packages without any compromise in integrity [22]. Automating these manual operations has a significant impact on FE workflows, especially in optimization scenarios where manual input needs to be avoided. For example, the manual application of the boundary conditions between structural components and the internal fluid domain for the industrial use case in Fig. 8 takes approximately 40 minutes compared to less than 3 minutes using the automated solution in this paper. As the automated time includes the generation of the non-manifold model and extracting the topology into the data structure the actual query time is insignificant and the time-saving grows as more interface regions need to be defined. Computational efficiency and mesh validation studies for the decomposed and mixed-dimensional analyses for the industrial use case model can be found in [8, 10].

Analysis attributes attached to cells in the cellular model were used to define interface properties required to couple adjacent meshes. Additional attributes may be exploited to achieve more realistic assessment of the contact behavior between interacting components in an assembly. Certain attributes may be retrieved from the underlying surface definitions of interacting faces from the Attribute relation in the data structure in order to determine contact types i.e. planar, cylindrical etc. It also may be necessary to attach manufacturing attributes such as friction coefficients or stiffness properties to define the properties of an interface, i.e. rigid-to-flexible contact definitions are required for non-linear metal forming analyses where forming tools are considered rigid due to their superior stiffness over the deformable material. In other instances adjacent components may both be treated as rigid bodies and the coincident nodes at their interface may be merged. Regardless of the application, the analysis attributes can be applied at a high level and utilized to automate the downstream analysis model generation. The relational nature of the data structure enables different relations to be easily generated to store the attributes required for certain applications or analysis types.

It has been described how virtual superset entities can be created to enabling a single interface query to be used for the application of boundary conditions for multi-component assembly models. This reduces the number of queries required, especially where multiple design modifications occur and models need to be re-evaluated to generate analysis models for optimization runs. Benefits of this approach are visible for assemblies consisting of numerous identical components. For example, it is common for aerospace components to contain multiple fastener assemblies. To simplify boundary condition application, fastener assemblies may be classified into virtual superset groups allowing interfaces for each fastener group to be treated as one, reducing the number of queries required to identify individual interfaces or the manual work required to treat each individual interface. This highlights another significant advantage of this approach as it is often unclear which components actually interact with each other. This has been demonstrated in this paper where the structural components were grouped in a virtual superset in order to calculate the fluid interfaces. It is feasible that such an approach might be used to re-evaluate interactions at different time points in a time-dependent analysis. For example, impact analyses where contact interfaces may change throughout the duration of the analysis. Inexpensive techniques like this would enable more efficient local treatment of boundary conditions instead of applying costly global definitions.

The work presented in this paper has been incorporated into existing analysis workflows, where direct geometric modifications are carried out during the decomposition phase of the process, Fig. 3 and Fig. 8. Initial implementation has been carried out in this manner due to restriction of current pre-processing (decomposition, mesh generation) tools working in the presence of a custom-generated virtual topology. It is the view of the authors that direct modification of the geometry should be avoided at all costs and analysis transformations should be implemented using virtual topology operations on the original geometry. Current work by the authors is focused on the creation of a suitable 'analysis topology' within the generic data structure using virtual operations, such as virtual decomposition Fig. 10, where a decomposition tool dictates where virtual partitioning needs to happen. Virtual decomposition relies on maintaining a robust description of the model topology and all orientation attributes to allow topological traversals to be utilized to carry out the decomposition. Creation of the analysis topology through the use of virtual topology enables a non-manifold representation to be created without the constraint of needing a non-manifold modeling solution. However, more work is required to determine how to fully exploit the analysis topology within current toolsets. This would require for example an open, transferrable definition of Virtual Topology and mesh generators which can operate in the presence of Virtual Topology. While there are issues to be addressed, a virtual analysis environment provides benefits such as associativity with the base design model and the ability to maintain multiple analysis topology definitions for the same representation.

6 CONCLUSIONS

This research has shown that once high level analysis attributes have been assigned to individual cells in a non-manifold cellular model it is possible to use interface information to automatically generate desired analysis models and seamlessly transfer analysis attributes between equivalent model representations. Cellular Modelling, Virtual Topology and Equivalencing information is used to facilitate the capture of the Simulation Intent of an analyst in order to avoid unnecessary or repeated manual effort in setting up the boundary conditions applied to the interfaces between different regions in analysis models. Boundary conditions are automatically transformed when switching analysis model fidelity regardless of detail or dimensionality. This allows analysis model set-up to be automated regardless of the model fidelity required. To this end the automatic application and transformation of boundary conditions within multi-component, multi-resolution and multi-fidelity analysis workflows has been demonstrated.

Christopher M. Tierney, http://orcid.org/[0000-0003-3341-6902]
Trevor T. Robinson, http://orcid.org/[0000-0002-6595-6308]
Cecil G. Armstrong, http://orcid.org/[0000-0001-8695-5016]

7 REFERENCES

[1] ABAQUS CAE, http://www.3ds.com/products-services/simulia/products/abaqus/abaquscae/, Dassault Systemes.
[2] Arabshahi, S.; Barton, D. C.; Shaw, N. K.: Steps towards cad-fea integration, Engineering with Computers, 1993, 9 (1), 17-26. http://dx.doi.org/10.1007/BF01198250
[3] Armstrong, C.; Bowler, A.; Cameron, S.; Corney, J.; Jared, G.; Martin, R.; Middleditch, A.; Sabin, M.; Salmon, A.: Djinn: A Geometric Specification for Solid Modelling Specification and Report, 2000. http://ralph.cs.cf.ac.uk/papers/Geometry/Djinnbook.pdf
[4] Boussuge, F.; Shahwan, A.; Leon, J.-C.; Hahmann, S.; Foucault, G.; Fine, L.: Template-based geometric transformations of a functionally enriched DMU into FE assembly models, Computer-Aided Design and Applications, 2013, 11(4), 436-449. http://dx.doi.org/10.1080/16864360.2014.881187
[5] CRESCENDO, Collaborative and Robust Engineering using Simulation Capability Enabling Next Design Optimisation, http://www.crescendo-fp7.eu
[6] Cuilliere, J.-C.; Bournival, S.; Francois, V.: A mesh-geometry based solution to mixed-dimensional coupling, Computer-aided design, 2010, 42(6), 509-522. http://dx.doi.org/10.1016/j.cad.2010.01.007
[7] Gostaf, K.; Pironneau, O.; Roux, F.: Finite Element Analysis of Multi-Component Assemblies: CAD-based Domain Decomposition, Domain Decomposition Methods in Science and Engineering, 2014, 927-935. http://dx.doi.org/10.1007/978-3-319-05789-7_90
[8] Makem, J. E.; Armstrong, C. G.; Robinson, T. T.: Automatic decomposition and efficient semi-structured meshing of complex solids, Engineering with Computers, 2014, 30(3), 345-361. http://dx.doi.org/10.1007/s00366-012-0302-x
[9] Monaghan, D.; McCune, W.; Bridgett, S.; Armstrong, C.; Robinson, D.; McKeag, R.: Dimensional Reduction of Analysis Models, in 5th International Meshing Roundtable, Sandia National Laboratories, 1996.
[10] 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, 2014, 30 (4), 689-701. http://dx.doi.org/10.1007/s00366-013-0317-y
[11] Nolan, D.C.; Tierney, C.M.; Armstrong, C.G.; Robinson, T.T.: Defining simulation intent, Computer-aided design, 2015, 59, 50-63. http://dx.doi.org/10.1016/j.cad.2014.08.030
[12] NX, http://www.plm.automation.siemens.com/en_gb/products/nx/, Siemens.
[13] Parasolid, http://www.Plm.Automation.Siemens.com/en_gb/products/open/parasolid, Siemens.
[14] Robinson, T. T.; Armstrong, C. G.; Fairey, R.: Automated mixed dimensional modelling from 2d and 3d cad models, Finite Elements in Analysis and Design, 47(2), 2011, 151-165. http://dx.doi.org/10.1016/j.finel.2010.08.010
[15] Sellgren, U.: Doctoral Thesis - Simulation-driven design, The Royal Institute of Technology, Stockholm, 1999.
[16] Shahwan, A.; Leon, J.-C.; Foucalt, G.; Trlin, M.; Palombi, O.: Qualitative behavioral reasoning from components' interfaces to components' functions for DMU adaption to FE analyses, Computer-Aided Design, 45(2), 2013, 383-394. http://dx.doi.org/10.1016/j.cad.2012.10.021

[17] Sheffer, A.; Blacker, T.; Clements, J.; Bercovier, M.: Virtual Topology Operators for Meshing, in 6th International Meshing Roundtable, Sandia National Laboratories, 1997. http://dx.doi.org/10.1142/S0218195900000188
[18] 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. http://dx.doi.org/10.1016/j.finel.2003.11.004
[19] Sung, R.; Corney, J.; Clark, D.: Automatic Assembly Feature Recognition and Disassembly Sequence Generation, Journal of Computing and Information Science in Engineering, 2001, 1(4), 291-299. http://dx.doi.org/10.1115/1.1429931
[20] Tautges, T.J.: CGM: A Geometry Interface for Mesh Generation, Analysis and Other Applications, Engineering with Computers, 2014, 17(3), 299-314. http://dx.doi.org/10.1007/PL00013387
[21] 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, 2014, 11(2), 193-205. http://dx.doi.org/10.1080/16864360.2014.846091
[22] Tierney, C.M.; Nolan, D.C.; Robinson, T.T.; Armstrong, C.G.: Using mesh-geometry relationships to transfer analysis models between CAE tools, Engineering with Computers, 2014. http://dx.doi.org/10.1007/s00366-014-0377-7