Advances in Engineering Software 87 (2015) 68–85

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A CAD/CAE incorporate software framework using a unified representation architecture

Zhaohui Xia [a], Qifu Wang [a,*], Yingjun Wang [b], Chen Yu [a]

[a] National CAD Support Software Engineering Research Center, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China
[b] Department of Structural Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, USA

ARTICLE INFO

Article history:
Received 17 February 2015
Received in revised form 16 May 2015
Accepted 17 May 2015
Available online 2 June 2015

Keywords:
Incorporate framework
Design and analysis
Unified representation architecture
CAD/CAE integration
Software environment
Boundary element method

ABSTRACT

Currently, there are still some big gaps between the CAD system and CAE system, e.g. the different data structure for model representation, which costs lots of time and effort of engineers in the interaction between these two kinds of systems. In order to bridge these gaps, an incorporate software framework is proposed in this paper. In this framework, the unified representation architecture (URA) is presented that makes CAD and CAE to be an organic entity. The URA contains three components: (1) unified data model (UDD) including unified B-rep, unified feature and unified mesh; (2) unified data management (UDM) consisting of unified interaction, unified data structure, unified Constructive Solid Geometry (CSG) history and unified interface; (3) unified display and post-processor (UDP) for both design and performance analysis. The URA facilitates the incorporation by explicitly representing design and analysis information as design features, which maintains their associations through the history chain. Besides the URA, a unified mesh data (UMD) is proposed to unify the mesh of CAD model display and CAE analysis with the purpose of reducing the redundancy of mesh data. The unified mesh data (UMD) is proposed to unify the mesh of CAD model display and CAE analysis, which greatly reduces the redundancy of mesh generation data. Finally, the high efficiency of the proposed framework is demonstrated by engineering examples.

1. Introduction

In the past decades, the integration of structural design and performance analysis has become more and more important in both CAD and CAE. The goal of this integration is to obtain better performance, wider interoperability, higher reliability, and shorter development cycles. Modern commercial CAD and CAE software are generally treated as separate modules requiring different methods and representations, which makes the geometric models need a large number of tedious repetitive interactions, e.g. small feature removal, model crack treatment, mesh, etc. [1,2]. These tedious interactions usually occupy 40–50% of the entire design time. Commercial CAD environments (Solid Works, Pro/Engineer, etc.) and CAE analysis environments (ANSYS, NX Nastran, etc.) primarily focus on integrating CAD and CAE module in the same software with the finite-element method (FEM). To generate an FEM model, the CAD geometry commonly needs to be modified and simplified to meet the analysis requirements. Hence a more efficient CAD/CAE integrated framework is necessary for the structural design.

CAD/CAE integration aims to reduce human interactions in the design process of geometric modeling and structure performance analysis to improve the efficiency of product design [3]. Nowadays, most of these tools could be broken down into two types. One is interface based on data exchange, which is widely used to integrate heterogeneous platforms. The other is integrated environment of CAD/CAE software implemented through the integration of the third-party finite-element analysis component, which is adopted by the major CAD companies [4–6]. To achieve CAD/CAE integration, model conversion is needed in the first type of methods which adopt standard neutral file (such as IGES and STEP). However, the converted model data often requires a lot of time to fix it manually due to the differences between mathematical representation in distinct systems, which greatly reduces the design efficiency [7]. The second type integrated technology is limited to the integrated environment, where CAD model is still very different from CAE model. In addition, the information might be lost. The data structures are not unified, and it might cost a long time for the design cycle.

* Corresponding author. Tel.: +86 27 87541974; fax: +86 27 87543973.
E-mail address: wangqf@hust.edu.cn (Q. Wang).

http://dx.doi.org/10.1016/j.advengsoft.2015.05.005
0965-9978/© 2015 Elsevier Ltd. All rights reserved.

Unlike most CAD/CAE integrated approaches using domain-based FEM for analysis, we present a CAD/CAE integrated framework adopting the boundary-based boundary element method (BEM). A framework is a set of classes, interfaces and patterns to solve a group of problems, which is a popular method to improve the development efficiency and reduce the development cost. In the proposed framework, designers can complete part design and performance analysis simultaneously instead of using two different software or two interface environment. Moreover, Fast Multipole Method and GPU parallel computing are used to accelerate the computing process of the BEM. This paper attempts:

- To provide an incorporate CAD/CAE software framework with a unified representation architecture which can further reduce the difference of data structure between the CAD geometry and CAE analysis feature.
- To present an algorithm to make the analysis feature information (boundary conditions) be kept and rebuilt automatically for geometry model changing, which can maintain the analysis intents for designers in the CAD environment.

The rest of this paper is composed as follows: Section 2 surveys the related work on the corresponding technologies used in the proposed framework. In Section 3, we describe the data structure of Unified Representation Architecture (URA). The architecture of the integrated framework is illustrated in Section 4. Examples are presented in Section 5, and finally a brief summary and discussion are given in Section 6.

2. Related works

Recently, CAD/CAE integration has become an active research topic, and lots of results have been obtained including approaches of the CAD/CAE integration framework [3,7–16]. However, most of them still take the CAD and CAE as two independent modules. In this section, we focus on literature in the component technologies related to the proposed integrated framework.

Some of the previous work has realized the automatic management of the association of model information. In mechanical practices, the structural analysis usually uses FEA in the high fidelity level (HFL), which is performed by professional CAE systems associated with specific CAD system. However, CAD and CAE systems usually use different data formats to represent the design geometry [7,14]. Yip et al. [17] focused on a knowledge-intensive CAD (KIC) which includes integration of design life cycle and engineering knowledge with CAD, including CAE results. Even so, they did not show how these two aspects interact automatically. Anumba [18] did some of the early work to explore the advantages of integrated CAD systems within a structural engineering context. But the scope was limited to CAD only. Schreier [19] discussed a development of CAD and CAE software tools toward each other and the trends of the software vendors to close the gap between them. The major objective of the author is to describe the ease of associativity between modern CAE and CAE software tools.

Xu and Wang [20] proposed a method integrating CAD/CAE/CAM into the product modeling process together with a feature based modeling technology and features are used to maintain integration between CAD and FEA models. In application studies, Yan and Jiang [21] proposed an integrated method of CAD/CAE/CAM for the development of a duel mass flywheel. One of the major issues is that the system uses a large number of software tools, which makes the interoperability and exchange between models complicated. In order to integrate information between CAD and CAE, a middleware development approach is also favored widely. Dr. Van der Velden [22] developed a GUI based system and proposed parametric CAE output using a platform to utilize multiple CAD and CAE software tools. Foucault et al. [23] addressed the mesh quality enhancement in the conversion of CAD model to finite-element model for analysis. Xu and Chen [24] developed a fully automated product design system with CAD/CAE integration and multi-object optimization. But the system is very difficult to be modified for more detailed and complex engineering problems. Most of the former work were to develop an initial product model and lacked recursive nature of an actual design process. Albers et al. [25] proposed a strategy for the development of an engine crankshaft with the integration of CAD, CAE and genetic algorithm. However, the author did not propose any means to complete the design loop. Cao et al. [12] developed a middleware to transform CAD models into acceptable CAE mesh model, i.e. HEDP (High End Digital Prototyping). It can manage model simplification and denaturing of CAD models to make it adequate to FEA meshing and get quick results, but the integration is one-way traffic and lacks the recursive loop support. Penoyer et al. [26] used KBE along with CAD, CAE and CAM for complete product development. The approach was GUI based with KBE to manage the majority of the product life cycle process. Nevertheless, the author does make use of embedded knowledge rather it is suggested to use direct user interface thus giving lower automation in the process.

Several in-house integrated frameworks aimed to close the gap between CAD and CAE and reduce the product development cycle have been established. Park and Dang [1] presented a framework that performed the integration between the commercial CAD/CAE software, and meta-modeling techniques which include response surface methodology and radial basis functions were applied to the structural optimization. Wang et al. [27] proposed an open and integrated framework that performed the structural design optimization by associating the improved sequential approximation optimization algorithm with the CAD/CAE integration technique. Even so, they are adopted commercial CAD and CAE software tools.

To support the B-Rep and polyhedral model simultaneously, Hamri et al. [10] introduced an HLT (High Level Topology) based on the mixed shape representation, which creates a robust link between CAD and CAE models. Matin et al. [8] presented a knowledge-based, parametric, modular and feature-based integrated CAD/CAE system for the mold design. Chang and Joo [28] presented an integrated system supporting optimization of a general mechanical system, including ground vehicles, which offers engineers to effectively for a mechanism design with optimal kinematic and dynamic performance in a CAD solid model.

Another way of CAD/CAE integration is that both CAD and CAE use the same model. Gujarathi and Ma [9] put forward a common data model (CDM) to implement CAD/CAE integration, which contains parameters required for the CAD modeling and CAE analysis. Lee [14] presented a single master model containing different types of all the geometric models required for CAD and CAE. However, they have a complex data structure. Hughes et al. [15] proposed the Isogeometric Analysis using NURBS (Non-Uniform Rational Basis Spline) basis function as shape function in the FEA, which allows models to be designed, calculated and adjusted in one step, using a common data set. IGA employs the exact geometry at all levels of discretization, whereas FEA uses piecewise polynomial approximations. A primary difference between these methods is that the IGA uses control points to interpolation points, and the basis functions are the same as that of CAD models [29]. However, IGA calculation adopts the control points, and analysis result rendering uses interpolation points. Two sets of data make it more complicated. Meanwhile, it is hard for IGA to automatically decompose the analysis domain into the tensor-product patches for the complex models, which make it very difficult to be for engineering models [30]. Since the Dirichlet conditions are inhomogeneous in the NURBS space and the control points are not usually on the surfaces, the Dirichlet boundary condition is difficult to be applied directly [11].

3. Unified representation architecture

A significant gap typically remains between CAD and CAE since engineers have different usage views toward product information models. Design–analysis integration requires recording the relations between CAE model and CAD model. Targeting the needs of design and analysis integration, a unified representation architecture (URA) is proposed in this section. The URA aims at efficiently supporting the processes containing data of modeling and analyzing. Therefore, it can be used to represent the required data, e.g. the model geometry data (described as a B-Rep or a feature-based model), the boundary conditions (BCs) shape data, mesh data and material data for the analysis preprocess. It is a centralized repository that contains those design and analysis information as well as their explicit constraints. The characteristics of design and analysis settings are kept in a systematic form and can be managed for their consistency.

In other words, the URA should consist of the geometric shape model (geometry feature, analysis feature, etc.) and discrete mesh model for analysis. In order to bridge the gap between CAD and CAE, the URA should satisfy the following requirements:

- To describe CAD design information and CAE analysis information with a unified data model.
- To support the analysis attributes (boundary conditions, material, etc.) representation providing an explicit description.
- To implement a unified interface for both CAD and CAE.
- To support the analysis attributes being adaptive like CAD design information in the CAD system.
- To be able to describe the same discrete model representation for both CAD display and CAE performance analysis.

Therefore, URA can be implemented from three aspects: data model, data operation and management and model display and post-processor. Among them, data model is the basis of the URA which is used as an aid to communicate between the modules in the proposed framework or other external software environment. Data operation and management is mainly used to describe the operation type and operation model of a data model. Model display and post-processor is presented for the users to conveniently watch the results. To meet the requirements, URA can be divided into three blocks shown in Fig. 1, UDD represents the unified data model, UDM is the unified data management, and UDP indicates the unified display and post-processor. “Entity” is the entity of a part model, which is the basis of URA.

Structure of URA shown in Fig. 2 follows the general design procedure. UDD provides a unified way to model with design and analysis information, which can be modified or deleted through UDM, and finally the model can be displayed to the users with UDP. These blocks will be illustrated in more detail as follows.

3.1. Unified data model

To implement the incorporate software framework, the priority is to make all the data model unified and contain both CAD and CAE data. Currently, Boundary Representation (B-rep) is extensively used in a geometric data model [31,32]. It describes the geometry of an object in terms of its boundaries, namely the vertexes, edges and surfaces which represent entities of zero dimension, one dimension and two dimension, respectively [33]. A solid model can be defined by a set of faces and bounded by orientable surfaces. The topology of the model presenting the object as a set of faces is shown in Fig. 3 while each face is bounded by edges, and each edge is bounded by vertexes. The domain-based FEM is pervasively used in most of the 3D product design analysis, in which CAD models need to be converted into domain mesh models and enriched with some material features and boundary condition’s data, etc. The interaction between CAD models and FEM models is intensive [34]. Therefore, FEM is unsuitable for the proposed framework of the CAD/CAE incorporate software. The BEM, due to its dimension reduction characteristics, decreases the complexity of mesh generation and has begun to be used in engineering problems. Consequently, the BEM is used to analyze structures of 3D elastostatics problems in this paper since it is boundary-based like B-rep [35–39].

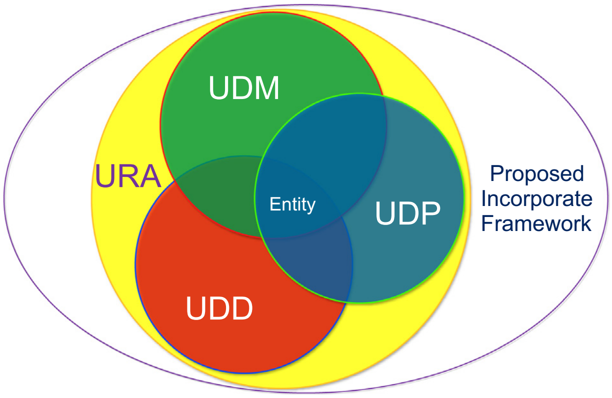


Fig. 1. General aspects of URA.

The BEM can be implemented on the mesh data generated from a solid model B-rep data structure. Therefore, the BEM has a real potential to integrate geometric design and engineering analysis into a completely unified framework. As shown in Fig. 3, $\boldsymbol{F_1}$ is a face of the CAD feature, meanwhile, it is also the face the CAE feature, so the B-rep of CAE feature in the proposed framework is the same as CAD feature, which contains faces, loops, edges and vertexes.

The B-rep describes solids at a very low level of abstraction [40]. A feature constructing a model in terms of functional elements has a particular significance to the design and analysis, which carries geometric and analysis information. In a feature-based system, features are represented by both geometric boundary representation (B-rep) and attribute information as the unified feature shown in Fig. 3. $\boldsymbol{DF}_1$ represents a CAD feature. $\boldsymbol{EF}_1$ is a CAE feature. $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, $\boldsymbol{F}_3$ represents the faces of features respectively, then the feature can be expressed in a unified way:

$$\text{feature} = \bigcup_{n=0,1,\ldots,N} F_n \cup I_{feature}$$
$$I_{feature} \in \begin{cases} \text{geometric dimension, tolerance, roughness, material parameters}\cdots \\ \text{(CAD intent)} \\ \text{mesh, boundary condition, analysis result, rendering}\cdots \quad \text{(CAE intent)} \end{cases} \tag{1}$$

where F_n represents the faces included in the feature, N is the total number of the faces, $I_{feature}$ indicates the feature intent such as geometric dimensions, tolerances, material, boundary conditions, etc. The structure of boundary condition is listed in Structure 1.

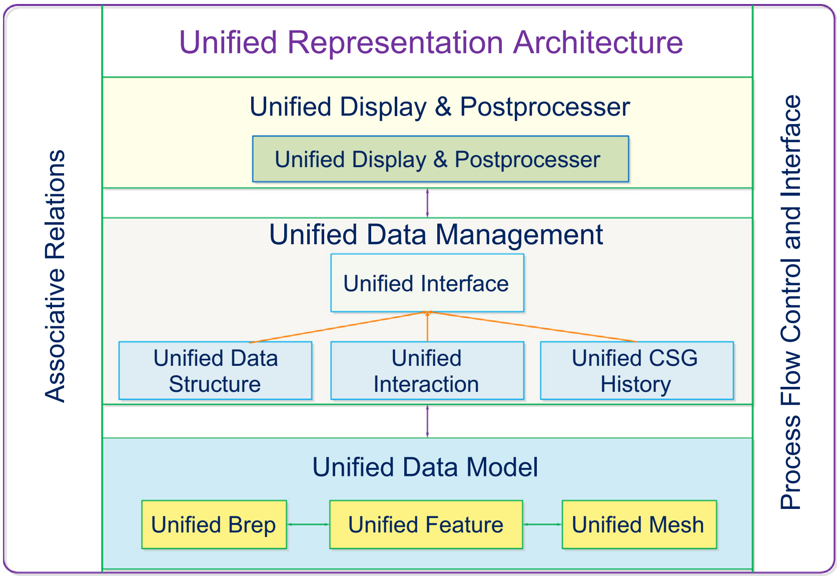


Fig. 2. Structure of unified representation architecture.

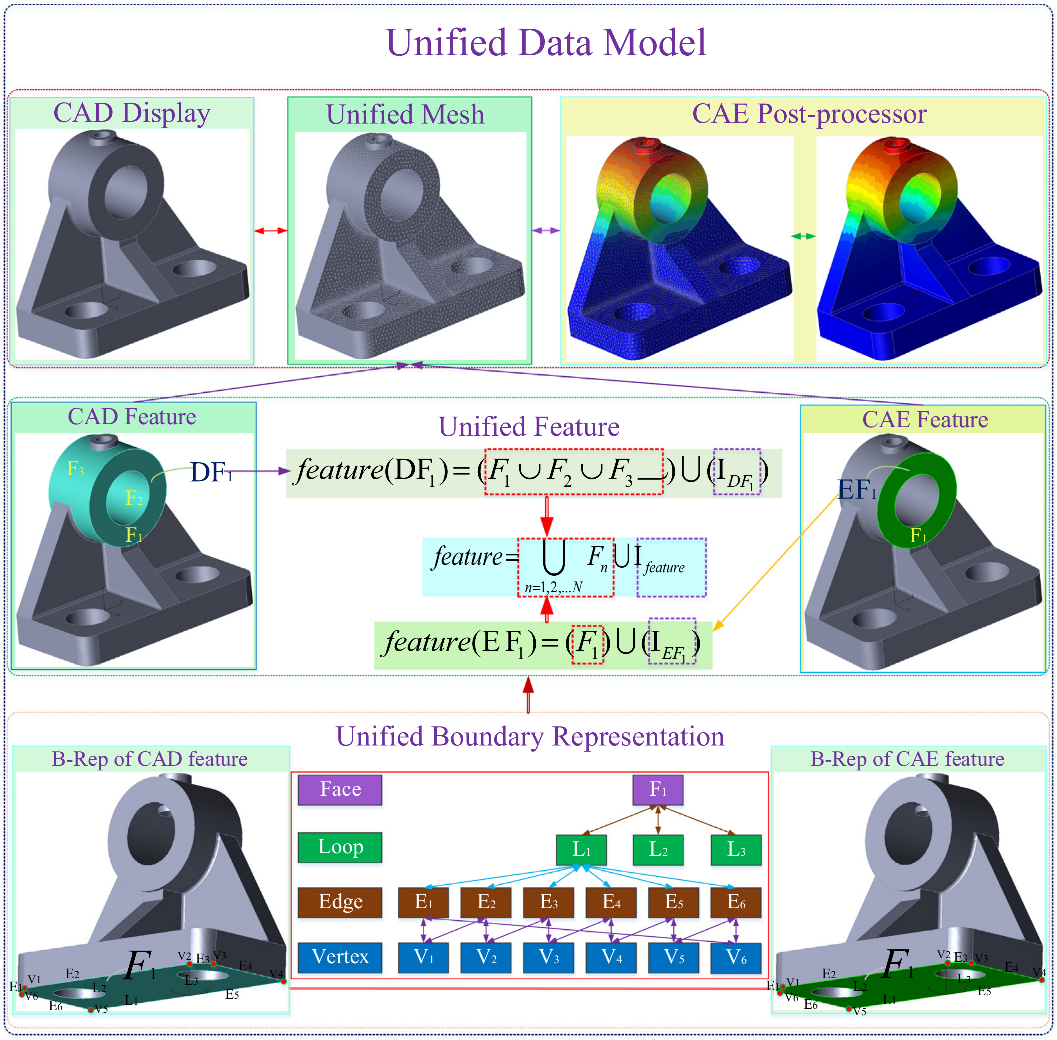


Fig. 3. Structure of unified data model.

Structure 1: Boundary Conditions

```
struct Boundary
  Condition
{
  BCTYPE bcType;            // type of BC (load or constraint)
                            BCTYPE
                            is an enum-type.
  HFaceCode* pFaceCode;     // face code of the bc
  SCOPETYPE nSType;         // scope of the bc point, line or face
  BOOL bNor;                // whether the direction of the BC
                            is normal
  BCDIR bcDirection;        // direction of the BC: X Y Z XY XZ
                            YZ or all
  double fBCValue[3];       // the value of bc in the x, y, z
                            direction
  ........
};
```

In the CAD system, CAD models are created with a parametric feature interactive modeling and save model data using CSG + BREP representation. The data structure of the UDD data structures is shown in Fig. 4, which records the feature history in its attributes and relationship between different features as well. The schematic diagram of UDD is illustrated in the EXPRESS-G notation which is a graphical modeling notation developed within STEP and used for IFC definition [41]. It is used to identify classes, the data attributes of classes and the relationships that exist between classes. EXPRESS-G is directly related to the EXPRESS data definition language. This diagram represents only the entities and attributes referred to in this paper.

Both the CAD feature (geometry) and CAE feature (such as boundary conditions for performance analysis) are derived from ACIS class. ACIS is the 3D modeling kernel of the Spatial Corp, which is an objected-oriented 3D geometric modeling engine (Spatial Corporation, USA) [42]. The ACIS model is highly redundant to enable efficient geometrical and topological operations [43].

The data structure of the feature data model has a merged set of all the user-defined features, which not only contains the geometric information, but also structural performance analysis information. To meet this requirement, the proposed data structure stores the basic information like feature name, feature identity, etc. The origin of a user-defined feature can be classified into two categories: CAD_Feature and CAE_Feature, which are derived from ACIS abstract class ENTITY. CAD features can be categorized into sketch feature and form feature (chamfer, extrude, round, etc.). Form features are described using B-Rep, which allow constructing solids with CSG operations and classified into additive features (+) (extrude), subtractive features (−) (hole, slot, shell, etc.) and intersection features (∩). CAE features not only involve shape geometry information-set entities (FACE, EDGE, CURVE, VERTEX, etc.), but also other analysis attributes, and it can be categorized into boundary conditions feature (BC feature), material feature, mesh feature, rendering feature, etc.

CAD model represented with B-Rep needs to be converted to triangle display format, the post-processor of CAE is based on the discrete model, and the triangular element is most common for the boundary integral. Currently, the conventional CAD/CAE systems adopt FEM for computation. Their CAD display mesh is not compatible with CAE polyhedral mesh, which will lead to mesh

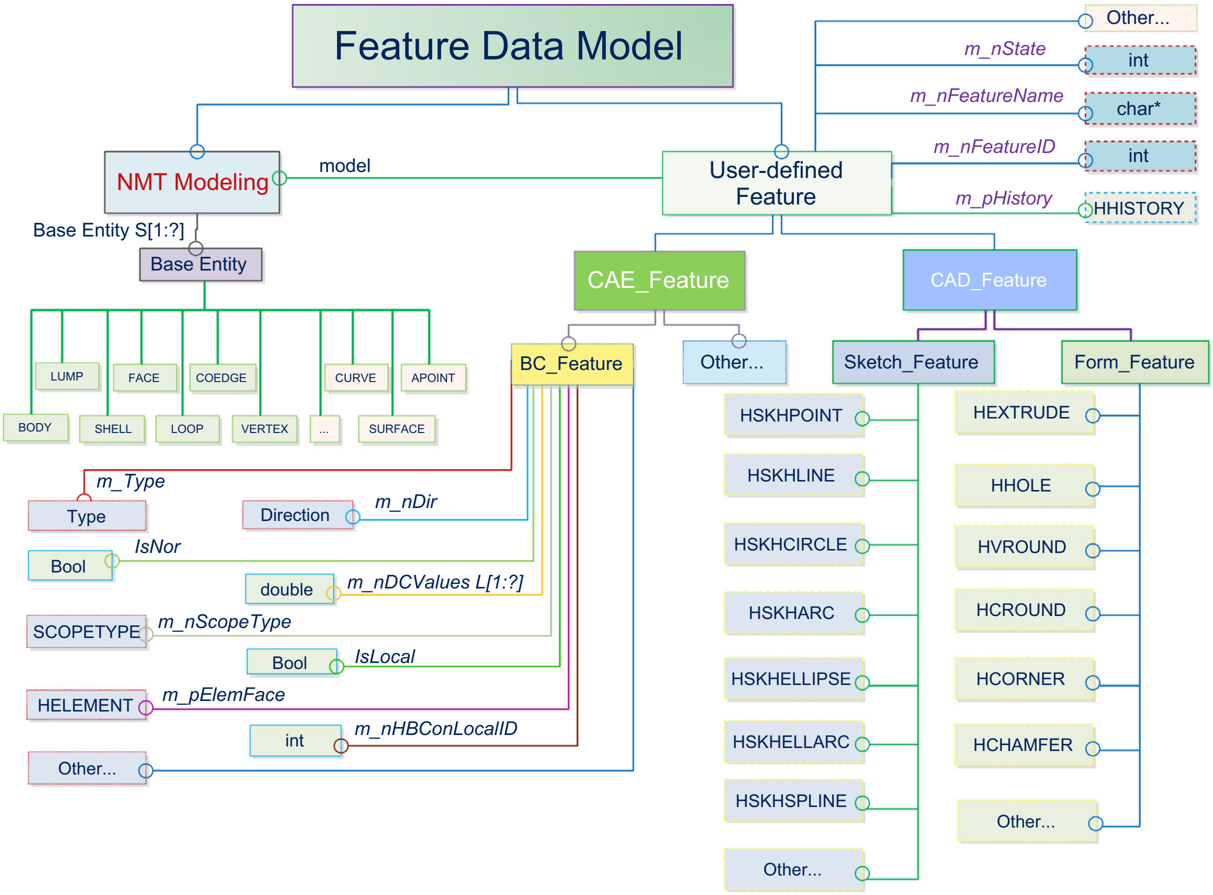
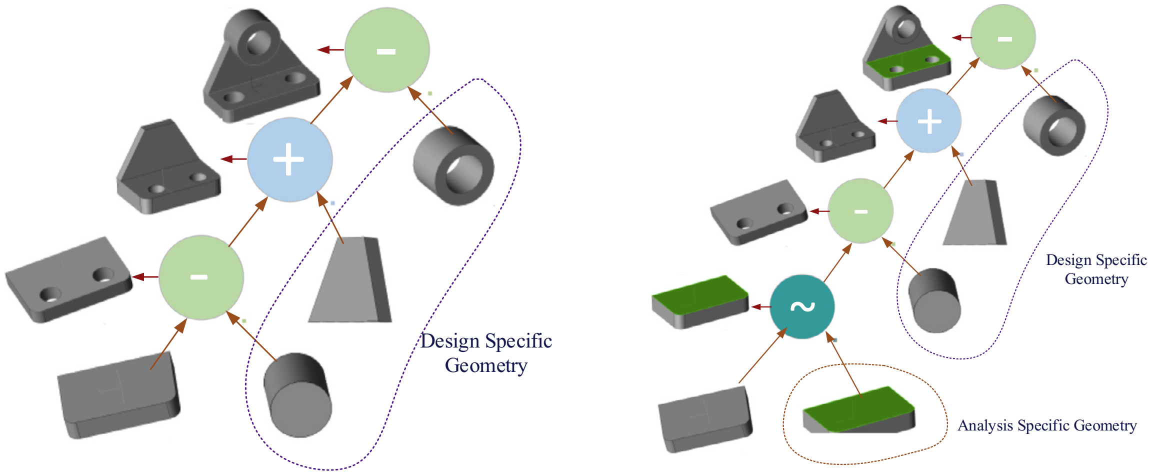


Fig. 4. Data structure of feature data model.



(a) Conventional software framework

(b) Proposed software framework

Fig. 5. CSG history tree of part model design processes.

data redundancy. Here, the BEM is adopted by the proposed framework environment. Meanwhile, the meshing data generated from BEM without more redundant data is not only fit for CAE analysis and post-processor, but also CAD display mesh, and it is called unified mesh data (UMD), whose data structure will be illustrated in detail in Section 3.3.

3.2. Unified data management

In the traditional product design, there are four processes: (1) *Modeling:* generating the product design model. (2) *Preprocessor:* mesh parameters (element type, element size, etc.), mesh data (elements, nodes, coordinate value, etc.), material attributes (Elastic Modulus, Poisson ratio, etc.), constraints for boundary condition (lines, areas, nodes, etc.), loads for boundary condition (force, press, etc.). (3) *Analysis Solving (with FEM):* displacement, Von Mises stress, etc. (4) Postprocessor: result rendering (displacement, stress, etc.), result output, etc. The final product model $\boldsymbol{P}_t$ can be defined by a series of design iterative processes as a CSG tree shown in Fig. 5 (a), the terminal nodes of the tree describe the primitive features. The internal nodes represent the feature operators, and it can be expressed as the following:

$$P_t = (PD, PA) = \left((\sum_{i=1}^{i=m} P_i \delta F_i), PA \right) \quad (\delta \in \{+,-\}, 1 \le i \le m, m \ge 1) \qquad (2)$$

in which $\boldsymbol{\delta}$ is Boolean operator: union (+) and subtraction (−), i represents the design stage, $\boldsymbol{P}_i$ is the ith stage design model, $\boldsymbol{F}_i$ is the ith stage CAD feature, *PD* represents the model design process while *PA* indicates the analysis process.

Unlike the traditional methods, the product design processes in the proposed software framework using URA are listed as follows: (1) *Preprocessor:* to generate a shape model and a discrete model. (2) *Analysis with BEM:* displacement, Von Mises stress, etc. (3) *Postprocessor:* result rendering (displacement, stress, etc.), result output, etc. Fig. 5(b) shows a feature tree that can be used to create an example solid model by applying four features, which can be described in a mathematical form as:

$$P_s = \sum_{i=1}^{i=m} Ps_i \delta s_i Fs_i \quad (\delta \in \{+,-,\sim\}, 1 \le i \le m, m \ge 1) \qquad (3)$$

The operator (∼) is not a Boolean operation between features, and it just attaches the attribute of a feature (such as boundary condition) to the design model without modifying design geometry information.

Therefore, the product model CSG history is unified in the proposed framework. In the UDM, there are unified interaction, unified interface and unified data structure. The unified interaction represents that both CAD and CAE features use the similar operating model. The interface of importing, exporting and the other operations are applied in the unified interface.

3.3. Model display and post-processor

Currently, the display of the product model in the CAD environment is based on discrete surface mesh, which is very different from the mesh generated for FEM computing. UMD includes the triangular discrete boundary representation (B-Rep) of a geometry model. In addition, it contains analysis properties such as material attribute, node displacement, node stress, node temperature and node strain, etc. When only the geometry modeling design was used, the node displacement, stress and strain would be set to 0. It will not only effectively avoid the rapid expansion of system internal data, but also facilitate code implementation and system maintenance. Therefore, the process of CAD display and CAE rendering can be unified with the UMD shown in Fig. 6.

The data structure of unified mesh for model display and post-processor is shown in Fig. 7. It contains the information of element, node (displacement, stress, strain, temperature, etc.), node position, mesh generation parameters (mesh shape: Quadrilateral, Triangular. mesh order: LINEAR, QUADRATIC; Mesh edge length, mesh maximum spanning angle, etc.), etc. And it records the relationship with geometry model in its attributes. If the quadratic element was used for mesh, the former three nodes would be used for CAD model display since the model display just needs the data of vertexes in the data structure.

The next section will illustrate how the URA is exploited to contribute to the processes of iterative design–analysis.

4. Software architecture of the proposed framework

Nowadays, users have to design product model and analysis model respectively for most CAD/CAE integrated system, in which users design a geometry model, and then switch to the CAE interface for analysis. Meanwhile, if users find that the design model is unqualified after completing analysis calculation, they have to return to the design environment to modify the geometry model, which may take plenty of time. To reduce the interaction between the designers and the CAD/CAE system during the integration process, the framework proposed here is based on URA. Consequently,

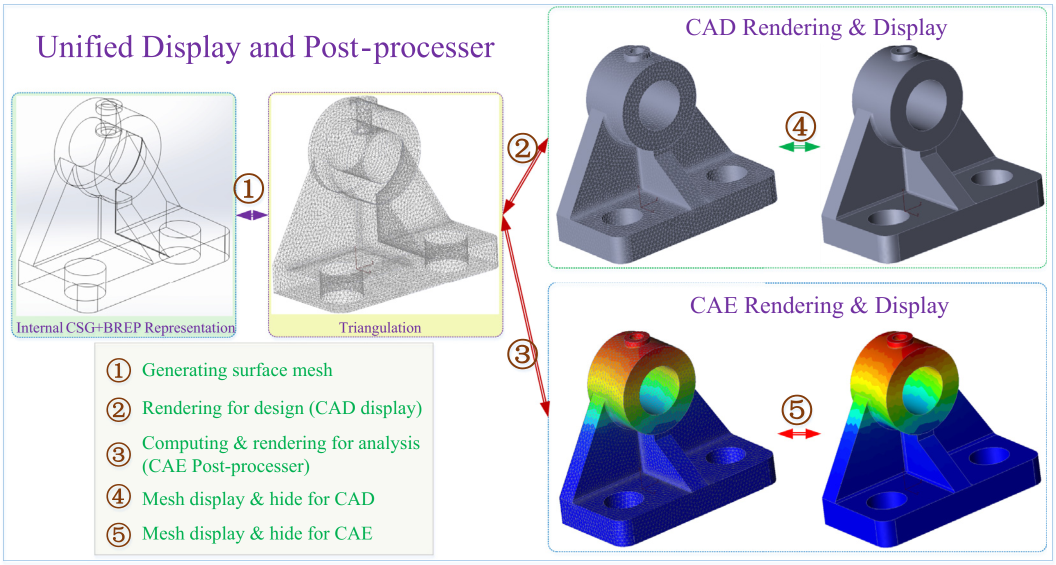

Unified Display and Post-processer
CAD Rendering & Display
CAE Rendering & Display
Internal CSG+BREP Representation
Triangulation
①
②
③
④
⑤
① Generating surface mesh
② Rendering for design (CAD display)
③ Computing & rendering for analysis (CAE Post-processer)
④ Mesh display & hide for CAD
⑤ Mesh display & hide for CAE

Fig. 6. Unified display process in the proposed framework.

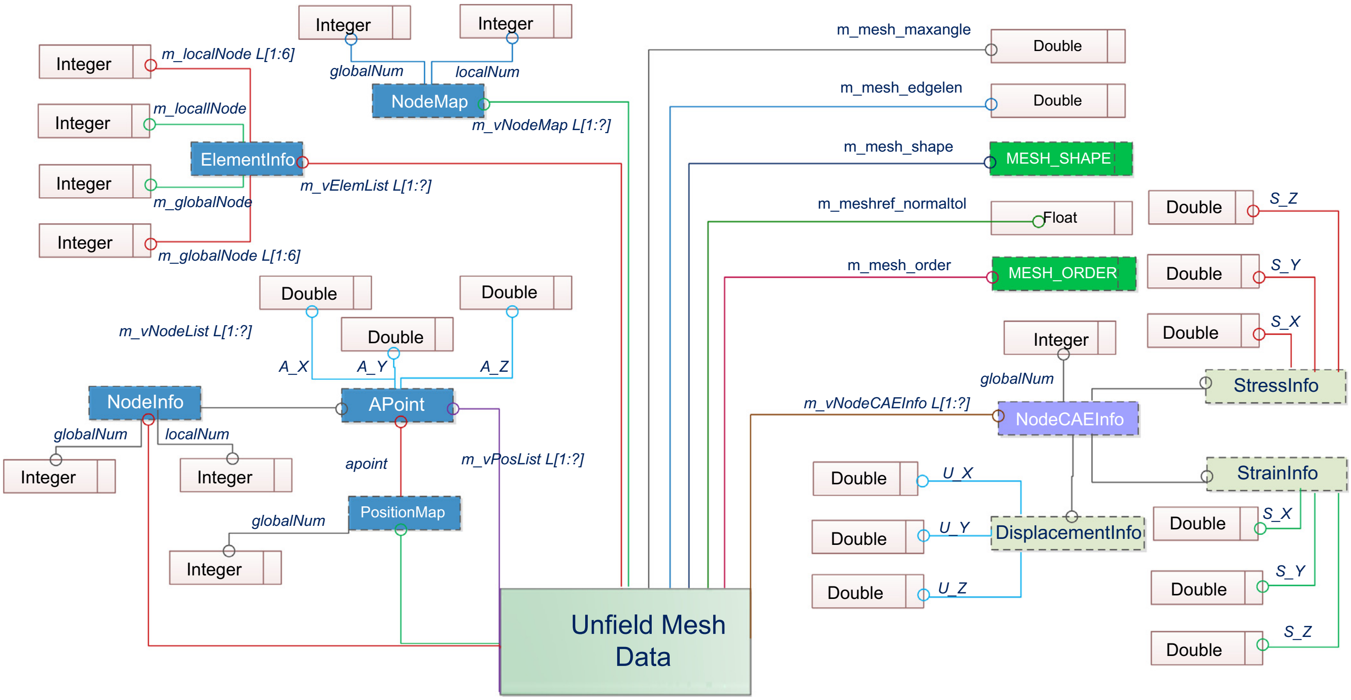


Fig. 7. Data structure of unified mesh.

it speeds up the design process and sets the designers free from monotonous repetitive tasks.

4.1. Framework of the incorporate CAD/CAE system

The architecture of the framework is proposed to implement the structural design and performance analysis composing of incorporate CAD/CAE framework, software and hardware accelerating technique. The framework is implemented based on the platform of Microsoft Visual C++ 2008 with ACIS and HOOPS, which can be effectively integrated by the mode of Model-View-Controller (MVC) [44]. And it has three layers including a basic layer, a function layer and an interface layer as shown in Fig. 8. The basic layer involves a geometric modeling kernel-ACIS and visualization engine-HOOPS, which are developed based on the operating system and hardware. ACIS provides an open architecture framework for wireframe, surface and solid modeling from a common, unified data structure. HOOPS is developed by Tech Soft 3D and technically supported by Spatial [42,45]. ACIS has the great advantage over HOOPS in geometric modeling. In contrast, HOOPS has great advantages over ACIS in visualization operation and human–computer interaction. The function layer is implemented using Microsoft Visual C++ 2008, and it contains two parts: modeling and structural performance analysis. The modeling part is a feature-based module, including CAD features (geometry) and CAE features (such as boundary conditions for performance analysis) which are derived from ACIS class. The model not only contains geometric information, but also structural

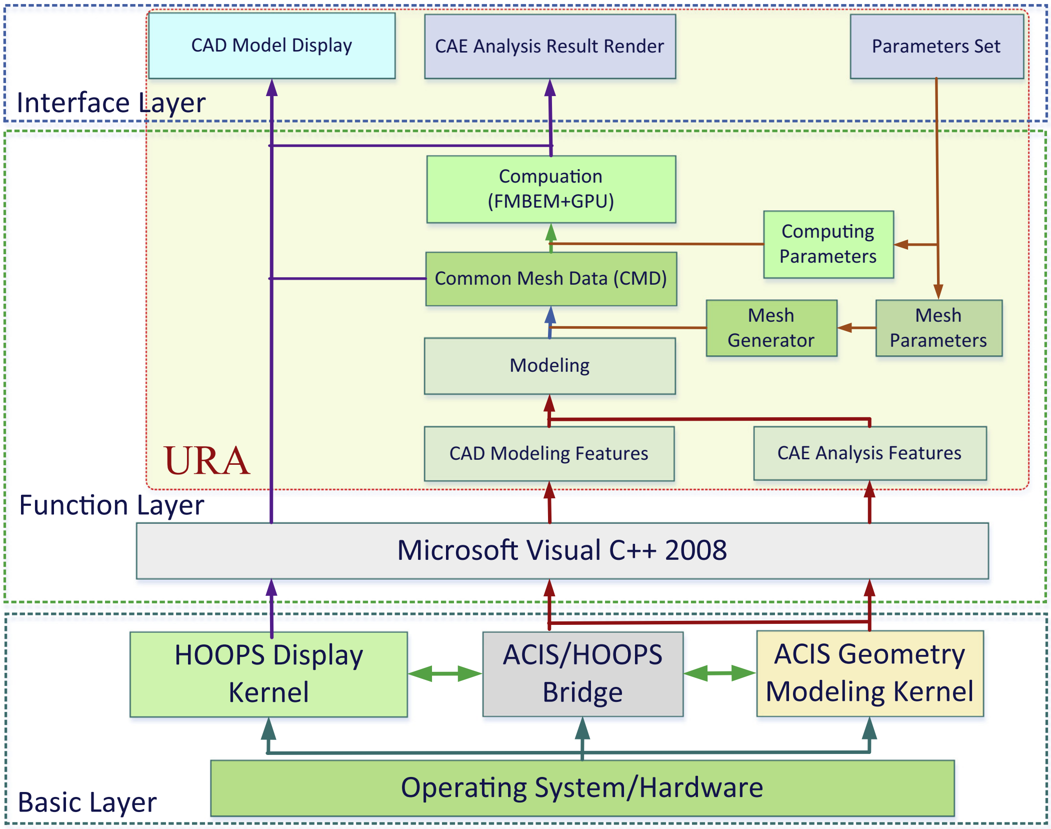


Fig. 8. Architecture of the proposed framework.

performance analysis information. As other commercial CAD software environments, the mesh generator generates surface meshes for model display, which are unsuitable for FEM computing. However, these meshes can play the role of providing Unified Mesh Data (UMD) both for model display and performance analysis computation using BEM in this framework. The integration can be implemented in the same framework without external data exchange. The computation module adopts software accelerating algorithm (FMBEM) and hardware accelerating algorithm (GPU) to enlarge the scale of the solution and shorten the iterative time of solving [46–52].

The application scenario of the proposed integrated CAD/CAE framework system is as follows:

(1) When the user design a part in the process of model creation, all geometric information for the CAD design and analysis information for CAE analysis are created as features respectively, and then merged into the part body model.
(2) If the users modified the geometry design feature or performance analysis feature in the part model, the corresponding feature would be modified adaptively in this proposed framework system.
(3) Users can complete the part design and structure performance analysis computation in the same interface.

4.2. Incorporate mechanism of the proposed framework

URA guarantees the incorporate data in the proposed framework. This section will illustrate the incorporate behavior for both CAD and CAE features. CAD feature is primarily responsible for geometric construction of the product model while CAE feature is used to carry performance analysis information. To implement the incorporation of them, CAE feature information (such as boundary conditions) is attached to the geometry model as shown in Fig. 9, which is not seen in other literature to the best of our knowledge.

The attached analysis information will be broadcasted when the geometric model changed (delete, split, merge, survive, etc.), so it should be maintained in the design–analysis loop process. In order to achieve this purpose, this section will illustrate how to maintain the boundary conditions and be adaptive to the model modification. From Section 3.1, the typical boundary condition (BC) features (one primary type of CAE feature) are shown in Fig. 10. Besides the boundary condition attributes, the BC feature records a set of faces $\boldsymbol{FS} = \{F_1, F_2, \ldots, F_N\}$, $\boldsymbol{N}$ is the total numbers of the faces, each face maps the corresponding EDGE sets $\boldsymbol{ES} = \{E_1^1, E_2^1, \ldots, E_M^1, E_1^2, E_2^2, \ldots, E_M^2, E_1^i, E_2^i, \ldots, E_M^i\}$, i is the index of faces, $\boldsymbol{M}$ represents the number of the edges. One feature might contain several faces. Meanwhile, the face on the CAD geometric feature might also be on the CAE analysis feature. Therefore, an identifying number is used to distinguish whether it contains CAD or CAE feature information. The algorithm of the incorporate feature behavior (such as recording all the feature topological entities and other information, adding them to the history chain) is demonstrated in Algorithm 1.

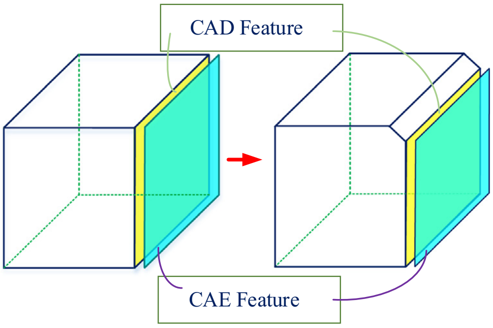


Fig. 9. Attached feature attribute to the face.

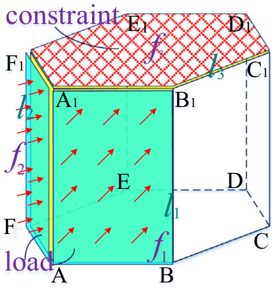


Fig. 10. Boundary condition feature.

Algorithm 1: Incorporate CAD/CAE Feature Behavior

```
// f: one face of the part model ften: a function which can get
   the unique code of the face in the part model
// bc: boundary condition added by user in the interface
// bcList: a vector contains all the face boundary conditions bc
   added by user
// numf: a face generating from CAD feature
switch(feat_type) // feat_type represents the feature type
case sketch: // sketch represents the sketch profile feature //
   ems represents the elements of sketch
  While ems! = null do // one sketch feature might include
   several elements (line, circle, etc)
     Coding ems and identifies its design type as 0.
  End while
  Adding the feature to the design history chain.
break;
case cad_feature: // cad_feature represents CAD feature
  Boolean operation with current model.
  While numf != NULL // one cad feature might generates
   several faces
     Coding numf, attaching the feature identity to the face,
   and identifying its design type as 1.
  End while
  Adding the feature to the design history chain.
break;
case cae_feature: // take boundary condition feature which
   might have several faces as example
  While bc ! = NULL do
  If ften(FACE(bc))! = NULL
     Getting the bc information and attaches to the geometric
   face f(bc), and recording it to the <vector>bcList, then
   identifying the analysis type of bc as 1.
  End if
  End while
  Adding the feature to the design history chain
break;
case analysis_result_feature:
  If resFlag // Determining whether is the analysis result
   feature
     Adding/updating Unified Mesh Data, and adding the
   feature to the analysis history chain
  end if
break;
End switch
Generating surface mesh from the model for CAD display or
   CAE performance analysis and Rendering it.
```

In the modeling process, a Boolean operation will be executed between CAD geometric feature and the model part body. Owing to the Boolean operation, the topological faces attached by a BC feature might be deleted, merged, split or still survived as shown in Fig. 11. Therefore, there should be some specific methods to handle the boundary conditions attached to those topological faces during this process.

While rebuilding the whole part, editing a feature or Boolean operation between feature and part body, those referenced topological entities should be retrieved from the record in the feature which is illustrated in Algorithm 2.

Algorithm 2: Retrieve Information from the Incorporate CAD/CAE Feature Behavior

```
Switch(feat_type)
case sketch:
  Getting all the sketch feature information from the selected
   design history chain node.
break;
case cad_feature:
     Getting the information from the selected design history
   chain node, and then retrieving the feature information
   from the coding.
break;
case cae_feature: // Retrieve the boundary conditions
// bcSize: Number of boundary condition numbers, size of
   <vector>bcList.
  For cnt = 0, 1, . . ., bcSize do
  For fid = 0, 1, . . ., numfaces do // numfaces: the number of
   faces of part model
  If ften(fid) == ften(FACE(bcList (cnt)))
  switch (FACE(bcList (cnt)) changed type) // 1 delete, 2 split,
  3 merge, 4 survive
     case 1: // face is deleted
     Modify the boundary condition attribute, set the applied
   scope to empty.
     break;
     case 2: // face is splitted
     Get all the faces which are the same face before splitting,
   set the applied scope of BC to the new faces.
     break;
     case 3:// face is merged
        Judge the old face merged into the new face whether
   includes BC feature, if Yes, then set the applied scope of BC
   to the merged face; if No, do nothing.
     break;
     case 4: // face is survived
     There is no modification in the related face, so the BC
   feature is survived without changing.
     break;
  end switch
  Update boundary condition feature, get the nodes and
   elements related to the boundary conditions of the faces in
   the part model
  End If    End For    End For
break;
case analysis_result_feature:
  Getting the result information from the Unified Mesh Data
   and re-rendering the result.
break;
End Switch
```

To implement incorporate mechanism, a history chain (HC) is used to mark and store feature data in the design–analysis processes, which contains geometry feature chain (FC_g), reference feature chain (FC_r), sketch feature chain (FC_s) and analysis feature chain (FC_a). Each node of the HC should be a feature, which records the feature information of state, relation, type, entity, etc. The last part

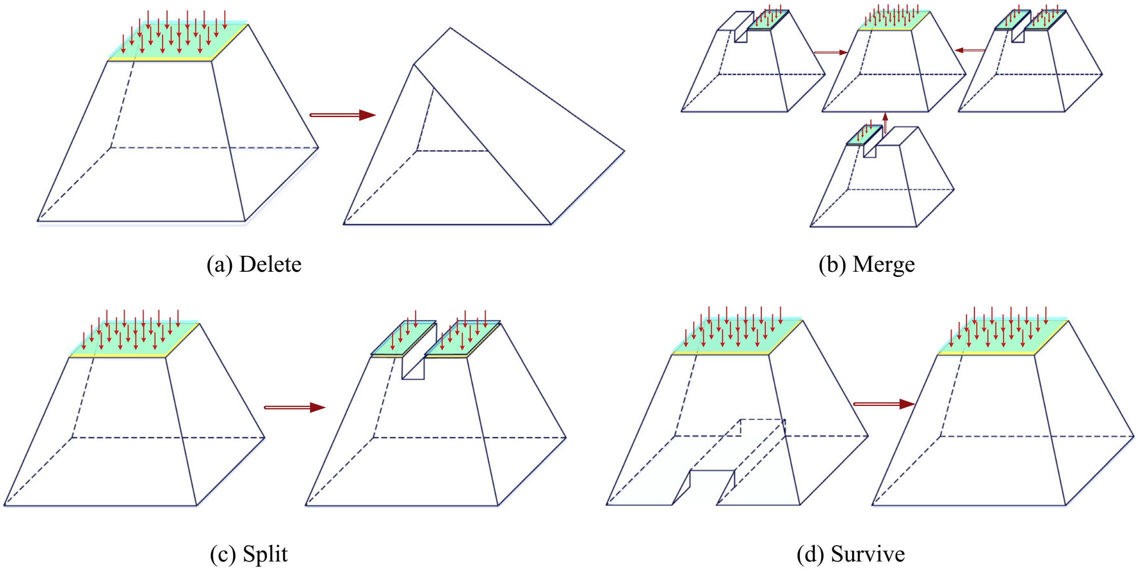


Fig. 11. Handling boundary condition.

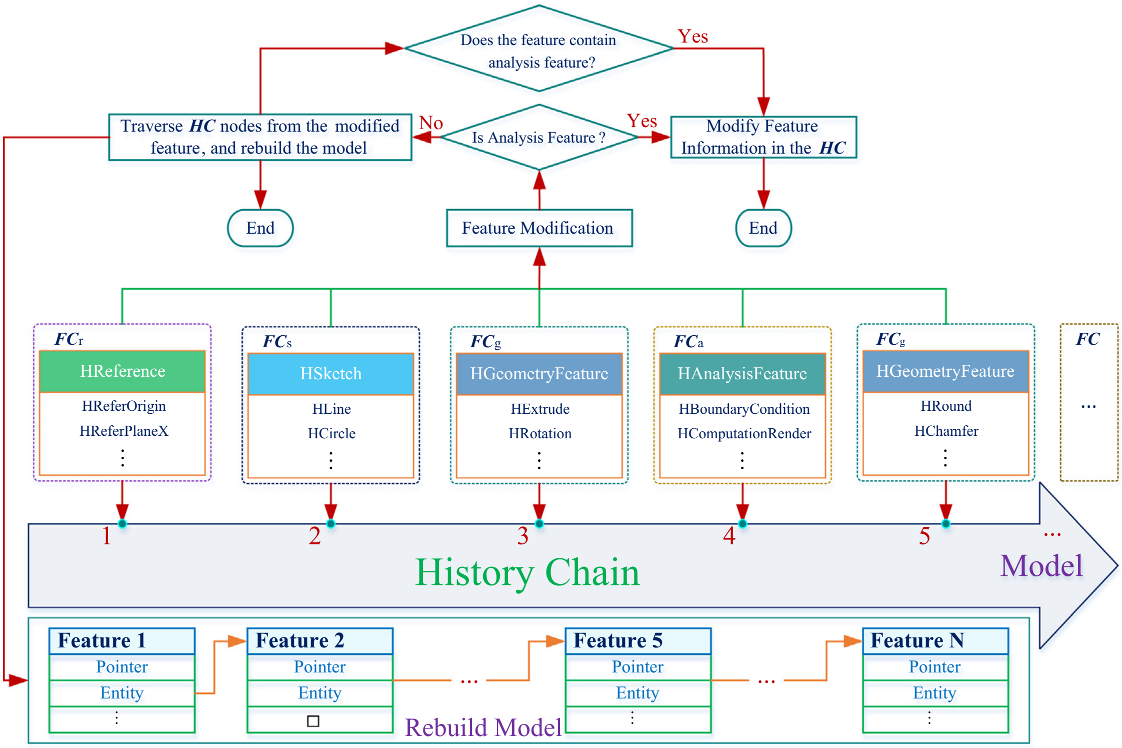


Fig. 12. Rebuilding process of the proposed framework.

model is formed by the set of HC nodes consists, and it can be described as:

$$\mathrm{HC}_i = \{\mathrm{FC}_g, \mathrm{FC}_r, \mathrm{FC}_s, \mathrm{FC}_a\}$$
$$\mathrm{Model} = \bigcup_{i=1}^{N} \mathrm{HC}_i \tag{4}$$

The algorithm of rebuilding mechanism is shown in Fig. 12. All the features will be added to the history chain. Once one feature changed, the algorithm would judge whether it is an analysis feature. The analysis feature does not participate in Boolean operation. Therefore, it just needs to modify feature information without rebuilding the model if it is an analysis feature. The model would be automatically rebuilt if the modified feature was a geometry feature and its topological face contained analysis information, then the analysis feature information would be updated.

4.3. Design–analysis process with URA

Nowadays, users have to design a product model and analysis model respectively for most CAD/CAE integrated system. The iterative design–analysis process of these systems can be expressed generally as:

$$\mathbf{P} = \boldsymbol{FB} + (\boldsymbol{M} \cup \boldsymbol{BC} \cup \boldsymbol{E}) + \boldsymbol{CR} + \boldsymbol{FB} + (\boldsymbol{M} \cup \boldsymbol{BC} \cup \boldsymbol{E}) + \boldsymbol{CR} + \boldsymbol{FB} \ldots \tag{5}$$

In which, **P** represents the last model product, ***FB*** is the feature body, ***M*** contains parameters related to mesh such as element type, element order, edge length, etc. ***BC*** indicates the boundary conditions, ***E*** includes the material parameters (Poisson ratio, elasticity modulus, density, etc.). ***CR*** is the computation and render for computation result. The iterative design–analysis processes are illustrated in Fig. 13 (a).

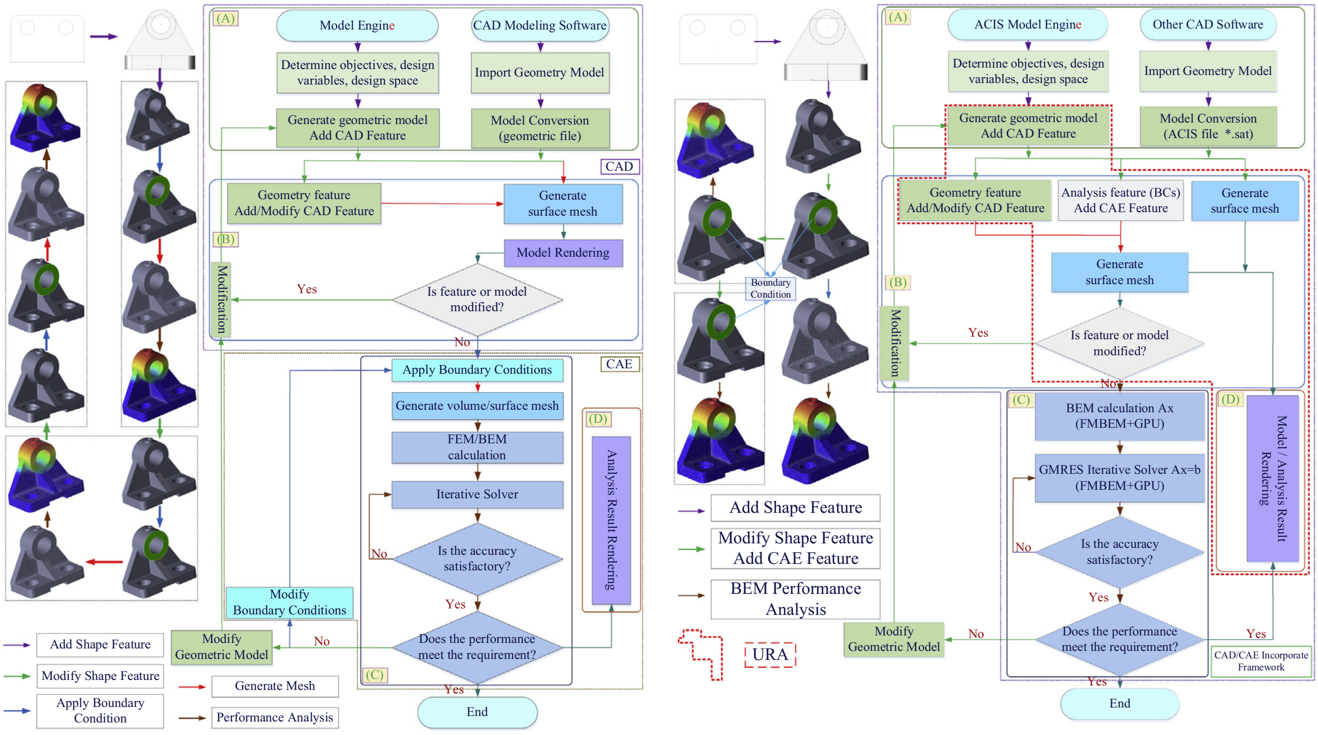
URA acts as a kernel data structure connecting all modules, supported with file and data-management modules, it makes all the modules can be a system in a single interface. The basic purpose of this study is to propose an efficient and time-saving design procedure of design development. An incorporate CAD/CAE system should have the ability to generate all the engineering information, computer models and analysis results automatically. Therefore, the iterative design process using URA can be determined by Eq. (6) which is distinct from Eq. (5):

$$\mathbf{P} = (\boldsymbol{FB} \cup \boldsymbol{M} \cup \boldsymbol{BC} \cup \boldsymbol{E} \cup \boldsymbol{CR}) + \boldsymbol{FB} + \boldsymbol{FB} + \ldots \tag{6}$$

In which, the mesh parameters, boundary conditions, material parameters and computation rendering can be adaptive for the design model modification.

Just like normal design practice process, as shown in Fig. 13 (b), the design–analysis processes with URA are divided into a number of stages following similar logic in the program code, and there is no CAD/CAE integration process because they are an incorporate system as described in Section 3. The architecture of the proposed software framework with URA is composed of four blocks:

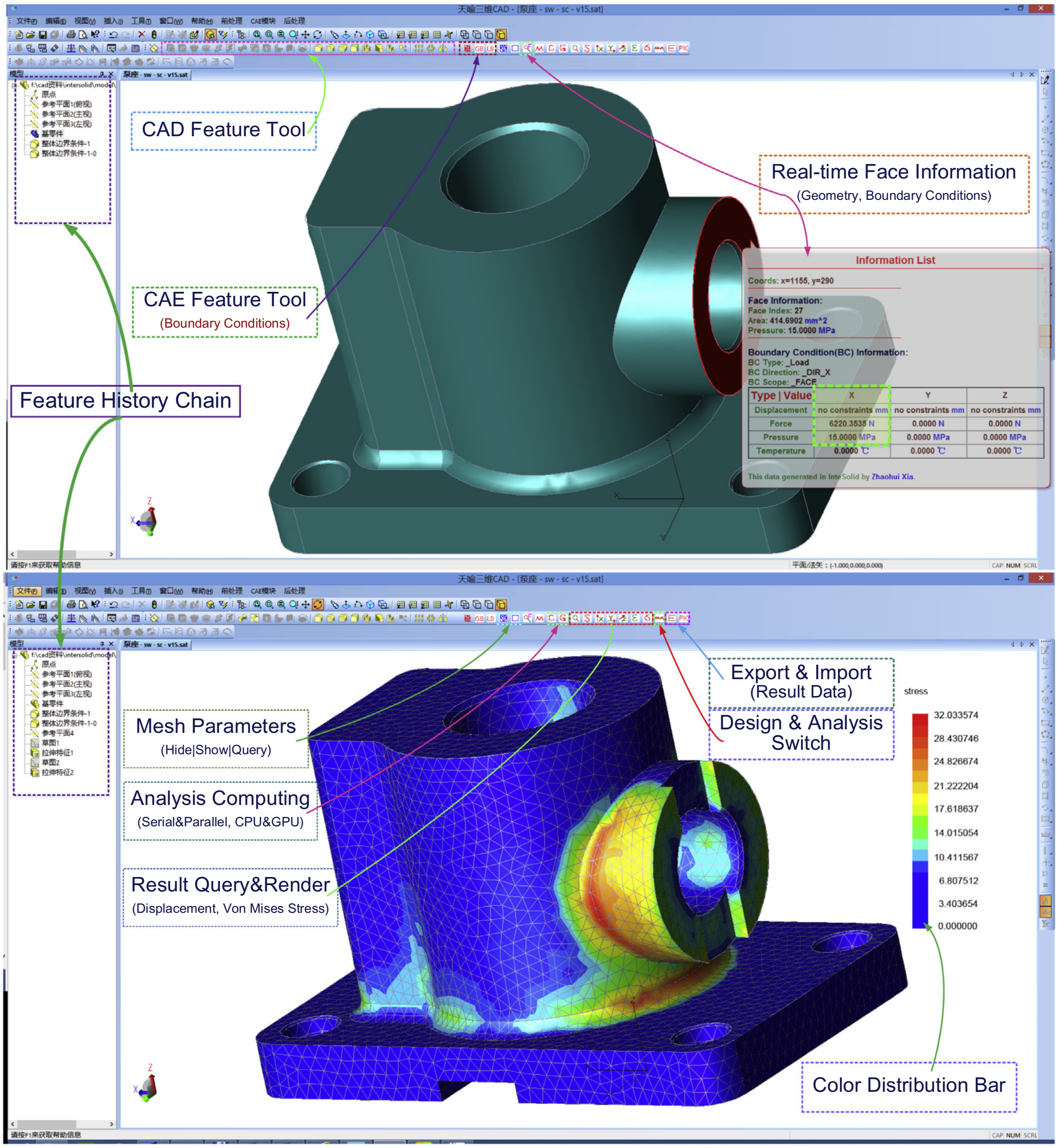
(1) Block (A) represents the level of geometry generation modules. The customer's requirements and technical requirements accepted by the designer become the design intent and they are designed with the ACIS engine embedded the engineering knowledge. This block includes geometry modeler-ACIS, CAD feature based on sketch profile (like extrude feature), and geometry model importing from other CAD software environments. It is used to create the geometric model which can be exported through a standard format such as SAT.
(2) Block (B) represents the processes related to the CAD features based on another feature (such as slot, chamfer and round), CAE features (boundary condition, etc.), and discrete model data. In this section, we describe the processes related to the use of the URA. These can be applied either on a part body generated from the proposed software framework or on a given part converted from other CAD software environments. This block generates common mesh data representing the discrete model once the design model changed, and the model is used both for CAD display and CAE performance analysis computation.
(3) Block (C) represents the processes related to the performance analysis, and BEM is adopted in the proposed software framework as its dimension reduction characteristics. However, the BEM application has so far been limited to relatively small problems since the memory and computational complexity. In order to overcome the long response time defect, a software (fast multipole BEM) and hardware (GPU parallel computing) accelerated algorithm is used to speed up the analysis process which is illustrated in detail in the literature [46–49]. This block is also responsible for checking the accuracy, validating the performance.
(4) Block (D) represents the rendering module. Both the design model (including CAD and CAE feature) and analysis results share the same mesh data (common mesh data) and are rendered in this block. Once the design model is needed to be displayed, the mesh data related to the design


(A)
Model Engine
CAD Modeling Software
Determine objectives, design variables, design space
Import Geometry Model
Generate geometric model Add CAD Feature
Model Conversion (geometric file)
CAD
Geometry feature Add/Modify CAD Feature
Generate surface mesh
(B)
Model Rendering
Modification
Yes
Is feature or model modified?
No
CAE
Apply Boundary Conditions
Generate volume/surface mesh
FEM/BEM calculation
Iterative Solver
(D)
Analysis Result Rendering
No
Is the accuracy satisfactory?
Yes
Modify Boundary Conditions
Modify Geometric Model
No
Does the performance meet the requirement?
(C)
Yes
End
Add Shape Feature
Modify Shape Feature
Apply Boundary Condition
Generate Mesh
Performance Analysis
(A)
ACIS Model Engine
Other CAD Software
Determine objectives, design variables, design space
Import Geometry Model
Generate geometric model Add CAD Feature
Model Conversion (ACIS file *.sat)
Boundary Condition
Geometry feature Add/Modify CAD Feature
Analysis feature (BCs) Add CAE Feature
Generate surface mesh
(B)
Generate surface mesh
Modification
Yes
Is feature or model modified?
No
(C)
BEM calculation Ax (FMBEM+GPU)
GMRES Iterative Solver Ax=b (FMBEM+GPU)
(D)
Model / Analysis Result Rendering
No
Is the accuracy satisfactory?
Yes
Modify Geometric Model
No
Does the performance meet the requirement?
Yes
CAD/CAE Incorporate Framework
Yes
End
Add Shape Feature
Modify Shape Feature Add CAE Feature
BEM Performance Analysis
URA

(a) Traditional CAD/CAE IntegratedFramework

(b) Proposed framework with URA

Fig. 13. Comparison of iterative design–analysis processes.



Information List

Coords: x=1155, y=290

Face Information:
Face Index: 27
Area: 414.6902 mm^2
Pressure: 15.0000 MPa

Boundary Condition(BC) Information:
BC Type: _Load
BC Direction: _DIR_X
BC Scope: _FACE

Type | Value	X	Y	Z
Displacement	no constraints mm	no constraints mm	no constraints mm
Force	6220.3535 N	0.0000 N	0.0000 N
Pressure	15.0000 MPa	0.0000 MPa	0.0000 MPa
Temperature	0.0000 ℃	0.0000 ℃	0.0000 ℃

This data generated in InteSolid by Zhaohui Xia.

Fig. 14. Interface of the proposed framework.

model (like coordinates) without analysis information (like displacement, stress, strain, etc.) would be used for design model display. When the performance analysis results needed to be displayed, the mesh data related to the analysis would be used.

The comparison of design processes between the traditional framework and the proposed framework is shown in Fig. 13. Compared with the former, the latter has several advantages. Firstly, the CAD and CAE are an organic whole while they are two separate parts in the conventional framework. Secondly, the analysis feature has the same data structure with the design feature. Finally, the boundary conditions can be maintained in the history chain as design features, so they need not to be applied again once the model is changed.

5. Case study

This section will demonstrate the efficiency of the proposed framework through two case studies. The simple one is used to validate the design efficiency, while the complex one is to check the performance analysis efficiency. The comparison with other frameworks is described in this section as well. An interface of the prototype is shown in Fig. 14, in which we can know that the operations of geometry design and performance analysis are in the same interface. In the process of structural design, designers only need to pay attention to the design as well as the results of performance analysis, and they have no need to spend time to model transfer, preprocess and choosing various analysis parameters. Therefore, the design efficiency can be improved.

The structural component of the proposed framework is shown in Fig. 15. All the features and operations are attached to the feature history chain. All the cases in this paper are executed on a desktop computer: CPU-Intel Core i7 2600k (3.4 GHz), GPU-NVIDIA GeForce GTX580, OS-Windows 7(32bit), RAM-DDR3 SDRAM (3 GB). The compiler of the proposed integrated framework is Microsoft Visual Studio 2008, and the compiler for GPU codes is NVIDIA CUDA 4.0 (C language), and the results of FEM come from ANSYS 14.5 software.

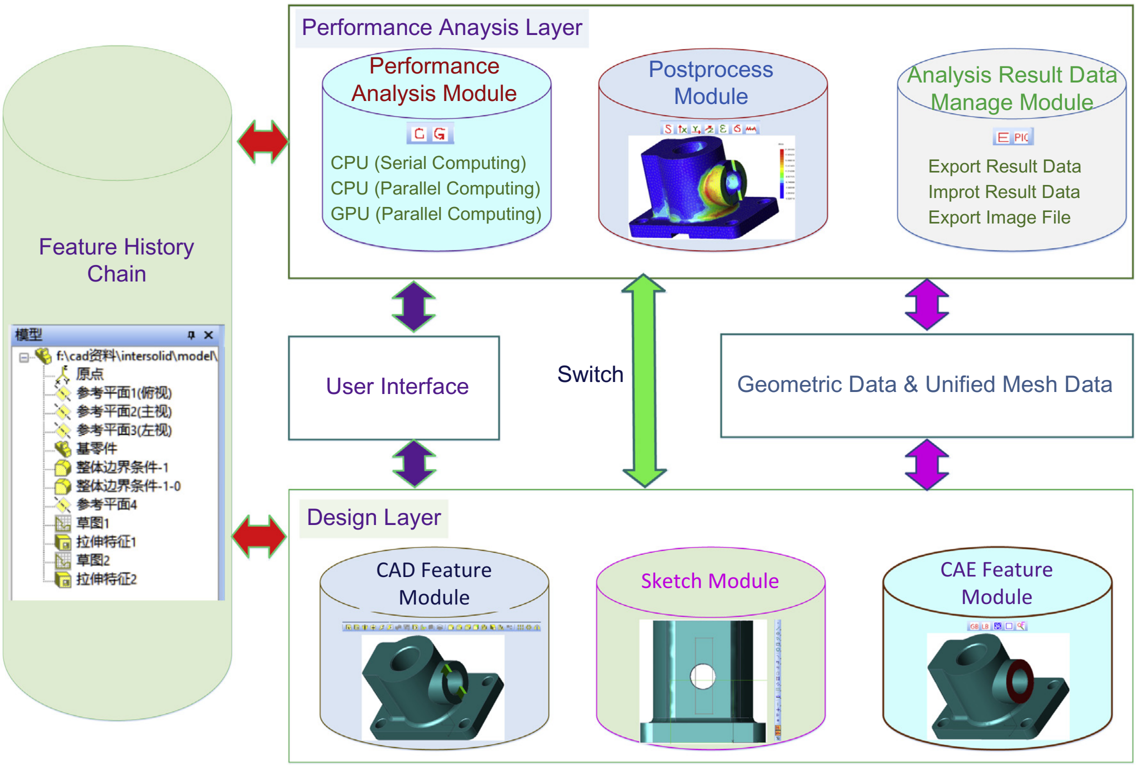
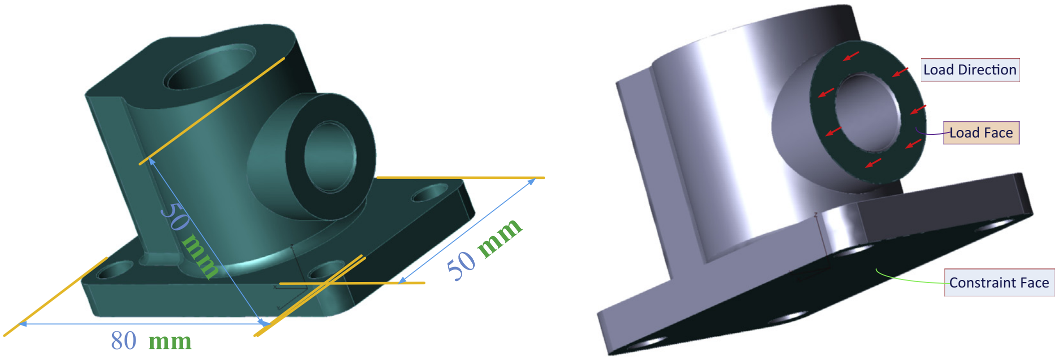


Fig. 15. Structural component of the proposed framework.



(a) Model size

(b) Boundary conditions

Fig. 16. The pump base model.

5.1. Pump base model for design efficiency

This case study is used to check whether this proposed framework can improve the efficiency of the design–analysis–redesign process compared with current commercial framework or not. This example is a pump base model whose size boundary conditions are illustrated in Fig. 16(a). The constraint on the part bottom surface is displacement constraint. The plane load value is 15 MPa, and its load direction and applied face are shown in Fig. 16(b). The Elastic Modulus is set to be 260,000 MPa, Poisson's ratio is 0.3.

The process of design–performance-analysis of the pump base model can be described briefly with the Fig. 17. Performance analysis can be done at any stage at any time with the help of the accelerated algorithm of the proposed framework since the computing time is very short. Once the performance analysis result does not meet the design requirements, the model can be changed in the same interface, which is very convenient for model modification. When the modification completed, the performance analysis can be triggered immediately without applying boundary conditions and meshing again. Apostrophe represents the process can be continued until the performance analysis meets the demand.

Table 1 shows the comparison result of the process for design–analysis–redesign under different software framework environments, among them, the BEM is used for performance analysis in the proposed framework while FEM used in ANSYS. Apostrophe represents the process can be continued.

As shown in Fig. 18, the time curve of the proposed framework depicts a trend of a gentle increase, while the other curve reveals a

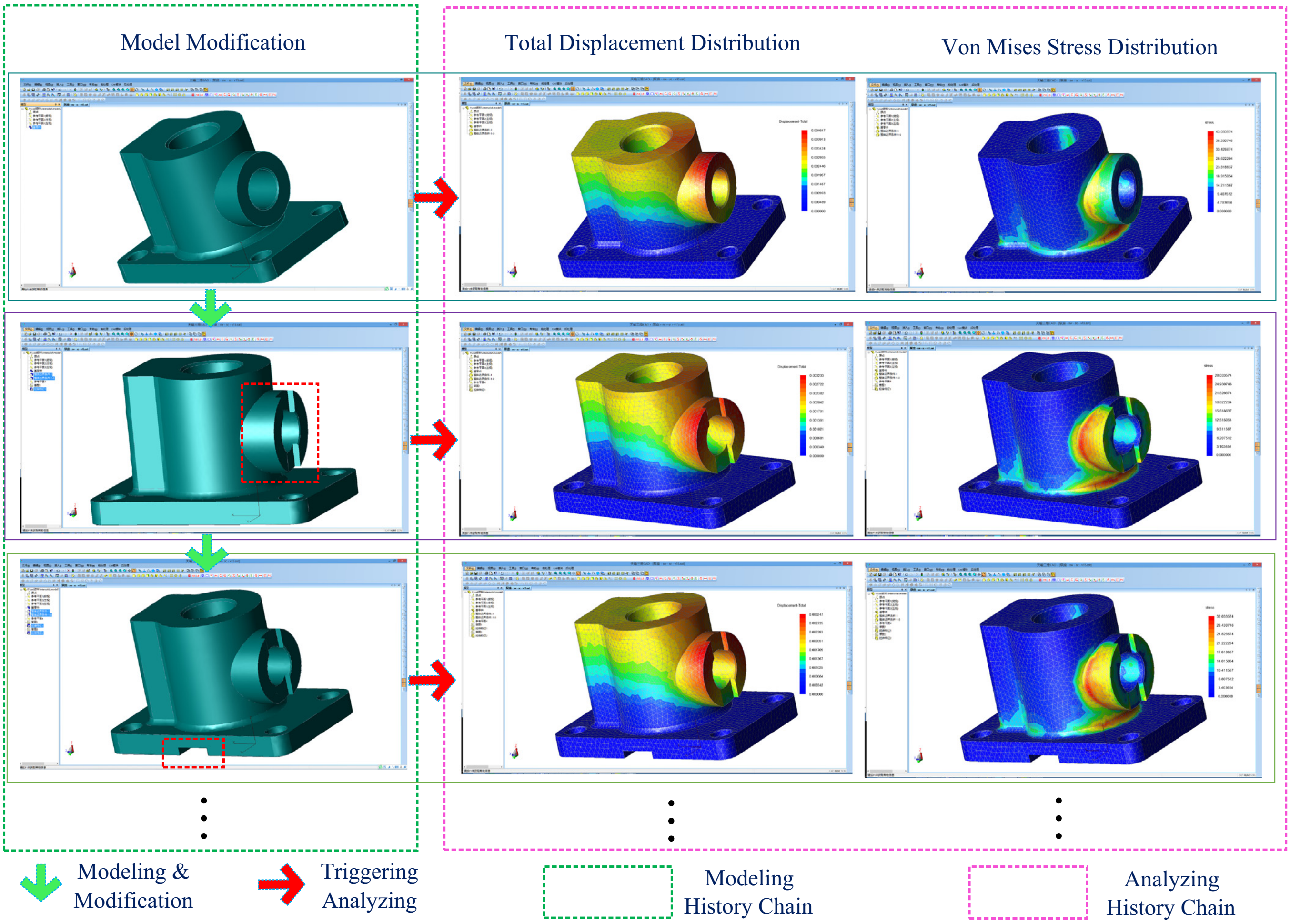


Fig. 17. Process of design–performance analysis–redesign.

Table 1
Time consuming comparison under different framework software environments (time unit: s).

SEs	Proposed framework						CATIA + ANSYS					
Pros.	*M*	*IO*	*P*	*C*	*PP*	*Total*	*M*	*IO*	*P*	*C*	*PP*	*Total*
D	200 + 26	0	10	7	2	245	200 + 25	15	10	11	2	263
DC[1]	28	0	0	7	2	282	25	15	10	12	2	327
DC[2]	30	0	0	8	2	322	28	15	12	13	2	397
DC[3]	30	0	0	7	2	361	28	15	10	12	2	465
DC[4]	30	0	0	7	2	400	27	15	10	13	2	532
DC[5]	32	0	0	8	2	442	29	15	12	12	2	602
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Note: SEs: Software Environments. *Pros.*: Procedures in the design and analysis processes. *SN*-Step Numbers (unit: 1). *M*: Total time for Modeling. *IO*: Time for Exporting model and Importing model to CAE environment. *P:* Time for Preprocessor: model simplification, mesh generation, boundary conditions applying. *C*: Time for performance analysis computation. *PP*: Time for Postprocess. *Total*: Total steps or time for the whole process. *D*: design process without change. DC^n: design the model with *n*th change.

trend of a sharper increase with the growing number of model modification. Meanwhile, it is obvious in the figure that the consuming time of our framework is less than the other, and this difference will be more and more apparent along with the rising times of model changing.

The total displacement and Von Miser stress of the pump base model are given in Figs. 19 and 20. As shown in the figures, it can be observed that their distributions are consistent, but there is a little computing error between the proposed framework and ANSYS, the stress error is bigger than the displacement.

5.2. Machine tool model for performance analysis efficiency

This case study is used to check whether this proposed integrated framework can save time and system resources for designers than current prevailing commercial framework or not. In order to better reflect the distinction between the two environments in the analysis computing process, the geometry and load conditions are more complicated than case study 1. The analysis results of ANSYS are also applied to this problem for comparison. A machine tool is chosen for this case study to show the importance of

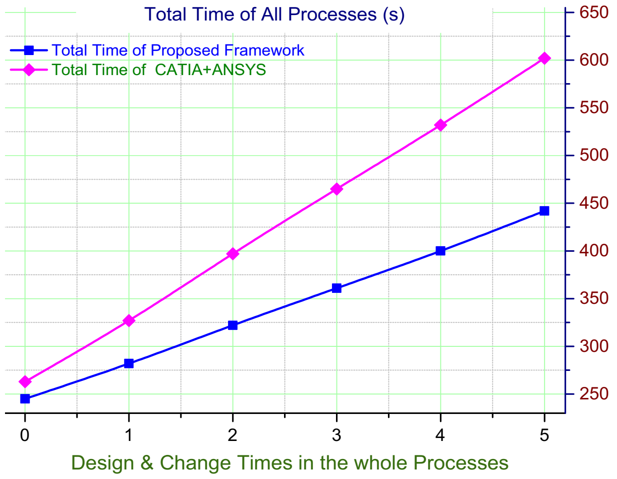


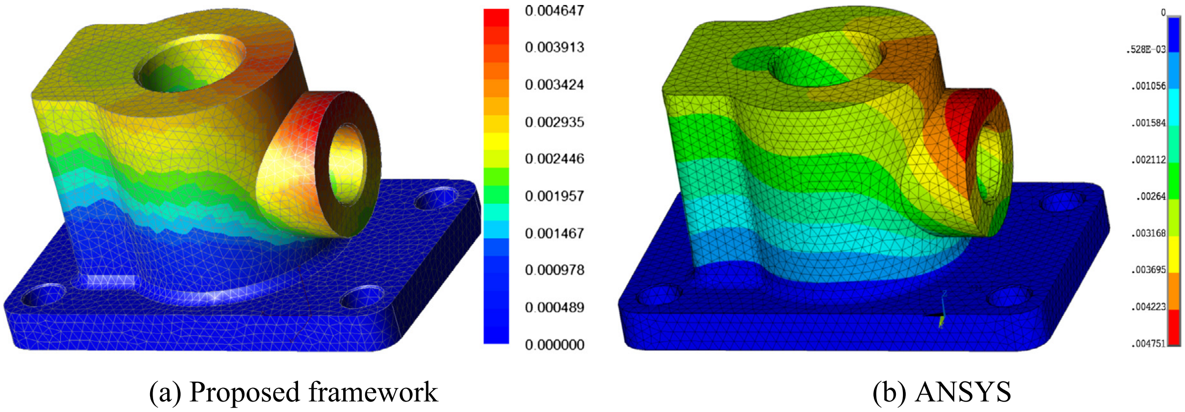
Fig. 18. Time consuming comparison with model modification.

CAD/CAE integration. The size and boundary conditions of this model are shown in Fig. 21(a). The constraint on the part bottom surface is displacement constraint; the load value in the faces is 3 MPa. The other related parameters are similar to the former case study. The total displacement and Von Mises stress distribution of the performance analysis result are shown in Fig. 21.

Table 2 shows the comparison of the machine tool model with the different computing methods to calculate, FMBEM with GPU parallel computing is used in the proposed framework, while FEM is used in ANSYS. There are 6 element size types for the machine tool model. They are respectively 60 mm, 50 mm, 40 mm, 30 mm, 20 mm and 10 mm. In the ANSYS row, two numerical values are separated by the symbol "||", the former represents the total memory allocated for solver, and the latter shows the total memory needed by the computing using in-core way. "***L***" represents linear element while "***Q***" is quadratic element. The in-core way completed in the memory is used for size 60 mm and 50 mm, and the out-of-core way is used for the others. "/" represents that ANSYS cannot complete the computing process since the requiring memory is greater than the available system memory.

As it is shown in Fig. 22(a), the memory curves with linear and quadratic element of the proposed framework show a trend of a gradual increase, while the ANSYS curves reveal a trend of a sharper increase with the decreasing element size. Meanwhile, it is obvious in Fig. 22(b) that the consuming time in our framework is less than ANSYS, and the increased trend is much slower than ANSYS. The difference will be more and more evident along with the growing number of degree of freedoms. The dotted line represents the estimated value because it has no enough available memory to be calculated.

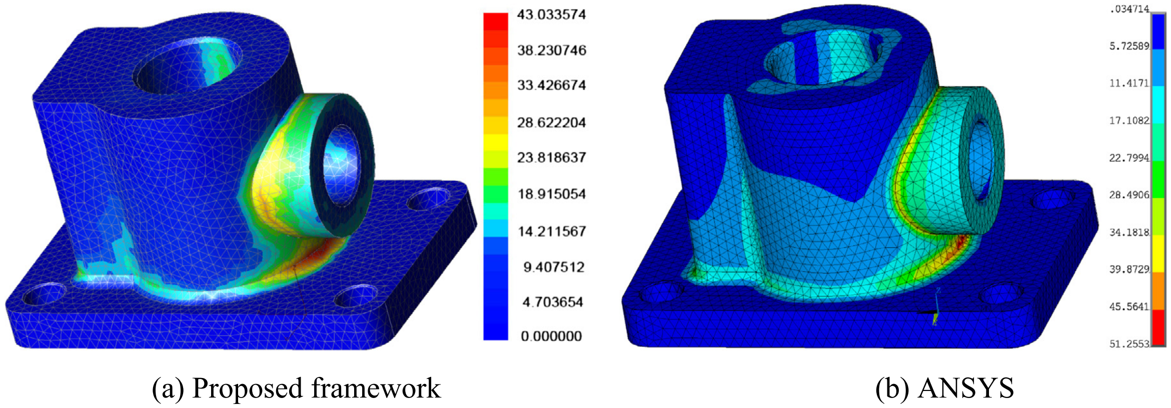
The total displacement and Von Miser stress of the machine tool model are given in Figs. 23 and 24. As shown in the figures, it can be observed that their distributions are consistent, but there is a



(a) Proposed framework

(b) ANSYS

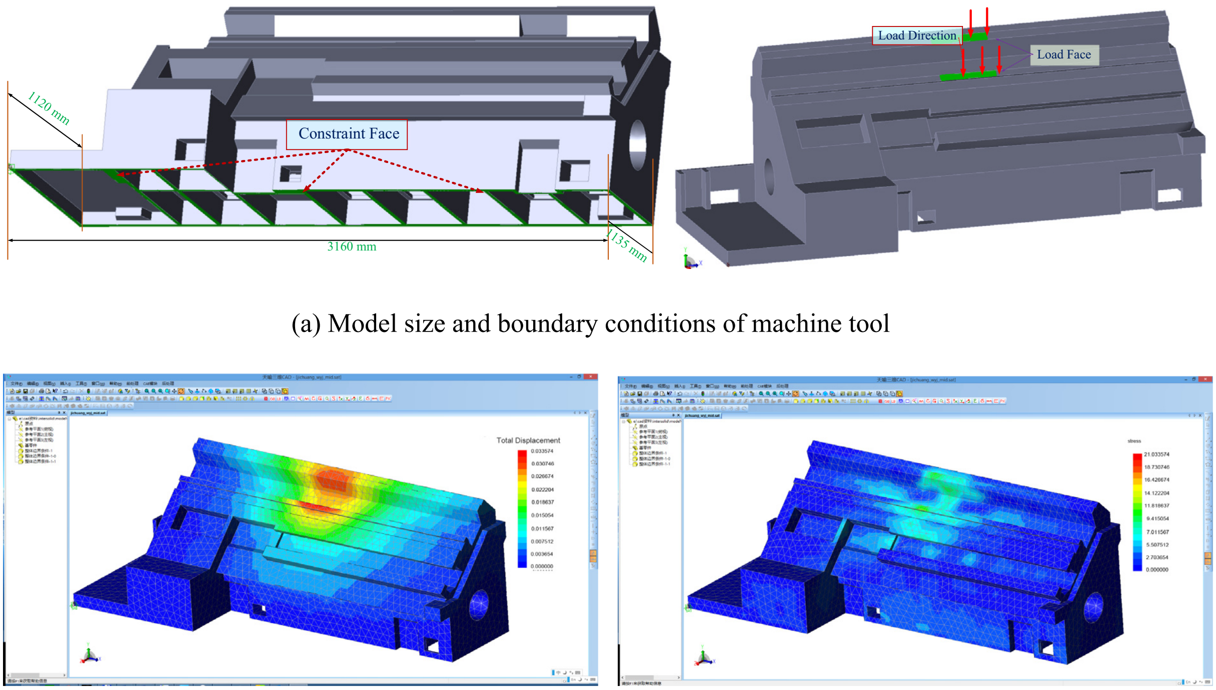
Fig. 19. Total displacement distribution of pump base.



(a) Proposed framework

(b) ANSYS

Fig. 20. Von Mises stress distribution of pump base.



(a) Model size and boundary conditions of machine tool

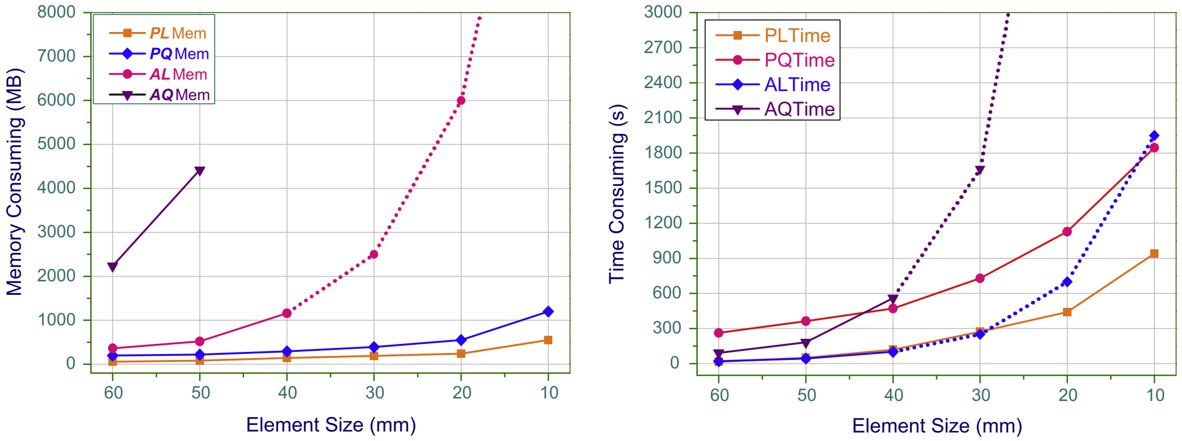
(b) Displacement distribution

(c) Von Mises distribution

Fig. 21. Machine tool model.

Table 2

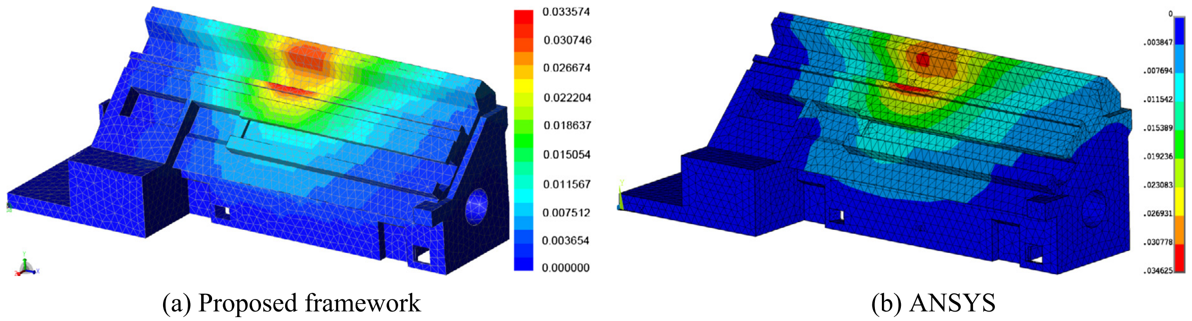
Size (mm)	Proposed framework				ANSYS			
	*PL*Mem	*PQ*Mem	*PL*Time	*PQ*Time	*AL*Mem	*AQ*Mem	*AL*Time	*AQ*Time
60	60.6	199.1	16	262	373.3‖364.3	325.9‖2234.2	20	92
50	85.1	220.1	50	363	682.2‖520.6	530.9‖4419.9	44	182
40	143.3	294.7	220	472	765.9‖1159.7	997.5‖9983.5	101	561
30	190.6	390.4	272	729	/	/	/	/
20	241.1	552.4	440	1128	/	/	/	/
10	549.0	1199.0	1040	2046	/	/	/	/



(a) Memory consuming

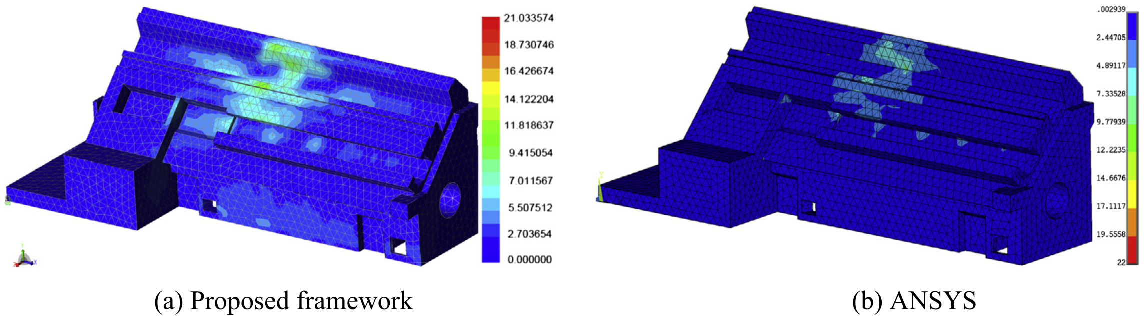
(b) Time consuming

Fig. 22. Comparison of time consuming and memory occupation.



(a) Proposed framework (b) ANSYS

Fig. 23. Total displacement distribution of machine tool.



(a) Proposed framework (b) ANSYS

Fig. 24. Von Miser stress distribution of machine tool.

little computing error between the proposed framework and ANSYS, the stress error is bigger than the displacement.

6. Conclusion and discussion

This study presents an incorporate CAD/CAE software framework based on URA and a geometric modeling engine-ACIS, display kernel-HOOS and Microsoft Visual C++ 2008, which allows an incorporate CAD/CAE operation in the same one software environment interface. The loop of design–analysis–redesign in the process is automatically performed since all the design and analysis information is treated as features using URA. The loop times can be reduced significantly with the software and hardware accelerating algorithms. Meanwhile, the design model display and the performance analysis share the same mesh data (unified mesh data). This approach also widens and develops automated and customized applications of CAD and CAE in design and analysis. Performance analysis processes of modern engineering product design often rely on high-performance workstation computer and complex computer-aided simulations to evaluate candidate designs. This study presents an incorporate design and performance analysis framework for structural design based on boundary element, fast multipole method, GPU parallel computing and URA. The feasibility, fidelity, convenience and computational efficiency of the incorporate CAD/CAE software framework have been tested through two cases of studies from simple to complex. The time-saving and memory-saving benefit applied to structural design process is enormous.

Although the benefits of the proposed framework for structural design are visible through some particular examples, more complex engineering studies will be validated. Meanwhile, there are still several limitations need to be fixed in the future. Firstly, the boundary condition only can be applied to the whole topological face, it cannot be applied to part of a face, a line or point because the algorithm finds the information of BCs from the face in the model, this should be improved in the future research; Secondly, it cannot solve all engineering problems, and it can barely support elastostatics problems currently. Therefore, expanding the type of boundary conditions (such as partial BCs, curve and point BCs) and more problems of structural design (such as dynamic problems, thermal problems, thermal fluid–solid coupling problems) will be involve in the future research, the computing algorithm will be further improved to shorten the computing response time.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (51475180).

References

[1] Park H, Dang X. Structural optimization based on CAD–CAE integration and metamodeling techniques. Comput-Aided Des 2010;42:889–902.
[2] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes T, Lipton S, et al. Isogeometric analysis using T-splines. Comput Methods Appl Mech Eng 2010;199:229–63.
[3] Wang L. Integration of CAD and boundary element analysis through subdivision methods. Comput Ind Eng 2009;57:691–8.
[4] Choi G, Mun D, Han S. Exchange of CAD part models based on the macro-parametric approach. Int J CAD/CAM 2009;2.
[5] Moaveni S. Finite element analysis: theory and application with ANSYS. Pearson Education India; 2003.
[6] Zhu J, Taylor Z, Zienkiewicz O. The finite element method: its basis and fundamentals. Butterworth-Heinemann; 2005.
[7] Deng Y, Lam Y, Tor SB, Britton G. A CAD–CAE integrated injection molding design system. Eng Comput 2002;18:80–92.
[8] Matin I, Hadzistevic M, Hodolic J, Vukelic D, Lukic D. A CAD/CAE-integrated injection mold design system for plastic products. Int J Adv Manuf Technol 2012;63:595–607.
[9] Gujarathi G, Ma Y. Parametric CAD/CAE integration using a common data model. J Manuf Syst 2011;30:118–32.
[10] Hamri O, Léon J, Giannini F, Falcidieno B. Software environment for CAD/CAE integration. Adv Eng Softw 2010;41:1211–22.
[11] Cottrell JA, Hughes TJ, Bazilevs Y. Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons; 2009.
[12] Cao B, Chen J, Huang Z, Zheng Y. CAD/CAE integration framework with layered software architecture. In: Computer-aided design and computer graphics,

2009 CAD/Graphics' 09 11th IEEE international conference on: IEEE; 2009. p. 410–5.

[13] Lin B, Kuo C. Application of an integrated CAD/CAE/CAM system for stamping dies for automobiles. Int J Adv Manuf Technol 2008;35:1000–13.

[14] Lee SH. A CAD–CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques. Comput-Aided Des 2005;37:941–55.

[15] Hughes TJ, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 2005;194:4135–95.

[16] Deng Y, Britton G, Lam Y, Tor S, Ma Y. Feature-based CAD–CAE integration model for injection-moulded product design. Int J Prod Res 2002;40:3737–50.

[17] Yip DC, Law M, Cheng K, Lau K, Barnes S. Knowledge intensive CAD in product design validation. Int J Knowl Intell Eng Syst 2005;9:45–61.

[18] Anumba C. Functional integration in CAD systems. Adv Eng Softw 1996;25:103–9.

[19] Schreier P. Modelling: CAD/CAE: closing the CAD/CAE gap. Sci Comput World 2009:4.

[20] Xu X, Wang Y. Multi-model technology and its application in the integration of CAD/CAM/CAE. J Mater Process Technol 2002;129:563–7.

[21] Yan ZF, Jiang ZF. Research on the integrated design of dual mass flywheel CAD/CAE/CAM. Key Eng Mater 2009;407:169–73.

[22] Van der Velden A. CAD to CAE process automation through iSIGHT-FD. ASME turbo expo 2007: power for land, sea, and air. American Society of Mechanical Engineers; 2007. p. 87–93.

[23] Foucault G, Cuillière J, François V, Léon J, Maranzana R. Adaptation of CAD model topology for finite element analysis. Comput-Aided Des 2008;40:176–96.

[24] Xu B, Chen N. An integrated method of CAD, CAE and multi-objective optimization. In: Computer-aided industrial design & conceptual design, 2009 CAID & CD 2009 IEEE 10th international conference on: IEEE; 2009. p. 1010–4.

[25] Albers A, Leon-Rovira N, Aguayo H, Maier T. Development of an engine crankshaft in a framework of computer-aided innovation. Comput Ind 2009;60:604–12.

[26] Penoyer J, Burnett G, Fawcett D, Liou S. Knowledge based product life cycle systems: principles of integration of KBE and C3P. Comput-Aided Des 2000;32:311–20.

[27] Wang D, Hu F, Ma Z, Wu Z, Zhang W. A CAD/CAE integrated framework for structural design optimization using sequential approximation optimization. Adv Eng Softw 2014;76:56–68.

[28] Chang K, Joo S. Design parameterization and tool integration for CAD-based mechanism optimization. Adv Eng Softw 2006;37:779–96.

[29] Wang YJ, Benson DJ. Multi-patch nonsingular isogeometric boundary element analysis in 3D. Comput Methods Appl Mech Eng 2015.

[30] Shapiro V, Tsukanov I, Grishin A. Geometric issues in computer aided design/computer aided engineering integration. J Comput Inform Sci Eng 2011;11:021005.

[31] Kao C, Kumara SR, Kasturi R. Extraction of 3D object features from CAD boundary representation using the super relation graph method. IEEE Trans Pattern Anal Mach Intell 1995;17:1228–33.

[32] Chen J, Freytag M, Shapiro V. Shape sensitivity of constructively represented geometric models. Comput Aided Geometric Des 2008;25:470–88.

[33] Abouel Nasr ES, Kamrani AK. A new methodology for extracting manufacturing features from CAD system. Comput Ind Eng 2006;51:389–415.

[34] Liu Y. Fast multipole boundary element method: theory and applications in engineering. Cambridge University Press; 2009.

[35] Liu Y. Analysis of shell-like structures by the boundary element method based on 3-D elasticity: formulation and verification. Int J Numer Methods Eng 1998;41:541–58.

[36] Hromadka T, Zillmer D. Boundary element modeling with variable nodal and collocation point locations. Adv Eng Softw 2012;43:96–103.

[37] Kwon H, Hong S, Song J. Development of a noise prediction software NASEFA and its application in medium-to-high frequency ranges. Adv Eng Softw 2015;83:11–8.

[38] Fiala P, Rucz P. NiHu: an open source C++ BEM library. Adv Eng Softw 2014;75:101–12.

[39] Citarella R, Cricrì G, Lepore M, Perrella M. Thermo-mechanical crack propagation in aircraft engine vane by coupled FEM–DBEM approach. Adv Eng Softw 2014;67:57–69.

[40] Hoffmann CM, Juan R. Erep an editable high-level representation for geometric design and analysis. 1992.

[41] Ming X, Mak K, Yan J. A PDES/STEP-based information model for computer-aided process planning. Robot Comput-Integr Manuf 1998;14:347–61.

[42] Xu AP, Zang T, Ji ZP, Qu YX. Heterogeneous object modeling approach based on ACIS and HOOPS. Key Eng Mater 2010;419:793–6.

[43] Thiemann F, Sester M. Segmentation of buildings for 3D-generalisation. ICA workshop on generalisation and multiple representation, Leicester; 2004. p. 20–1.

[44] Krasner GE, Pope ST. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. J Object Orient Programm. 1988;1:26–49.

[45] Xu A, Zang T, Ji Z, Qu Y. HO-CAD: a CAD system for heterogeneous objects modeling based on ACIS and HOOPS. In: Intelligent networks and intelligent systems, 2009 ICINIS'09 second international conference on: IEEE; 2009. p. 90–3.

[46] Xia Z, Wang Q, Liu Y, Wang Y, Wei Y. M2L optimization in FMBEM and its GPU implementation. WIT Trans Modell Simulat 2014;56:307–19.

[47] Xia Z, Wang Q, Huang Y, Yixiong W, Yingjun W. Parallel strategy of FMBEM for 3D elastostatics and its GPU implementation using CUDA. In: ASME 2014 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers; 2014. p. V01AT2A060–V01AT02A.

[48] Wang Y, Wang Q, Wang G, Huang Y, Wang S. An adaptive dual-information FMBEM for 3D elasticity and its GPU implementation. Eng Anal Bound Elem 2013;37:236–49.

[49] Wei Y, Wang Q, Wang Y, Huang Y. Optimizations for elastodynamic simulation analysis with FMM-DRBEM and CUDA. Comput Model Eng Sci (CMES) 2012;86:241–73.

[50] Wang Y, Wang Q, Deng X, Xia Z, Yan J, Xu H. Graphics processing unit (GPU) accelerated fast multipole BEM with level-skip M2L for 3D elasticity problems. Adv Eng Softw 2015;82:105–18.

[51] Wang Y, Wang Q, Wang G, Huang Y, Wei Y. Boundary element parallel computation for 3D elastostatics using CUDA. In: ASME 2011 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers; 2011. p. 347–55.

[52] Wei Y, Wang Q, Wang Y, Huang Y, Zhang L. Acceleration of modal analysis by FMM based on DRBEM. In: ASME 2012 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers; 2012. p. 981–90.