Improving interoperability between architectural and structural design and among structural analysis models: an industry foundation classes-based approach and web-based tools

Zhen-Zhong Hu[a,*], Xiao-Yang Zhang[a], Heng-Wei Wang[a], Mohamad Kassem[b]

[a]*Department of Civil Engineering, Tsinghua University, Beijing, China*
[b]*Technology Future Institute, Teesside University, Middleborough, UK*

Abstract

Medium and large construction projects typically involve multiple structural consultants who use a wide range of structural analysis applications. These applications and technologies have inadequate interoperability and there is still a dearth of investigations addressing interoperability issues in the structural engineering domain. This paper proposes a novel approach which combines an Industry Foundation Classes (IFC)-based Unified Information Model with a number of algorithms to enhance the interoperability:(a) between architectural and structural models, and (b) among multiple structural analysis models (bidirectional conversion or round tripping). The proposed approach aims to achieve the conversion by overcoming the inconsistencies in data structures, representation logics and syntax used in different software applications.

The approach was implemented in both Client Server (C/S) and Browser Server (B/S) environments to enable central and remote collaboration among geographically dispersed users. The platforms were tested in four large real-life projects. The testing involved four key scenarios:(a) the bidirectional conversion among four structural analysis tools; (b) the comparison of the conversion via the proposed approach with the conversion via direct links among the involved tools; (c) the direct export from an IFC-based architectural tool through the Application Program Interface (API), and (d) the conversion and visualization of structural analysis results. All these scenarios were successfully performed and tested in four significant case studies. In particular, the conversion among the four structural analysis applications (ETABS, SAP2000, ANSYS and MIDAS) was successfully tested for all possible conversion routes among the four applications in two of the case studies (i.e., Project A and Project B). The first four steps of natural mode shapes and their natural vibration periods were calculated and compared with the converted models. They were all achieved within a standard deviation of 0.1s and 0.2s in Project A and Project B, respectively, indicating an accurate conversion.

Keywords: BIM, Data Model, IFC, Model Conversion, Structural Analysis, WebGL

*Corresponding author. Tel.: +8601062789225
Email addresses: huzhenzhong@tsinghua.edu.cn (Zhen-Zhong Hu), zhangxiaoyang13@mails.tsinghua.edu.cn (Xiao-Yang Zhang), whw13@mails.tsinghua.edu.cn (Heng-Wei Wang), m.kassem@tees.ac.uk (Mohamad Kassem)

1. Introduction

The multitude of disciplines, technologies and teams and the multi-phased and temporary nature of project in the construction industry make them very challenging environments for information management and collaboration. Inadequate integration and interoperability are still inflicting an economic burden and are often considered key factors inhibiting the diffusion of innovation systems in the Design, Construction and Operation (DCO) industry. Within the structural engineering domain, building projects typically involve several consultants and engineers performing structural analysis utilizing different technologies and software applications. Structural analysis processes require them to share very diverse set of information and data models. In the absence of integration solutions between various structural analysis technologies, this task is very challenging, time and resource consuming due to the amount manpower required for re-modeling work and resolving inconsistency and incompatibility issues. Therefore, it is of paramount importance to develop approaches and tools that can provide an efficient conversion of data models between such technologies with adequate quality and fidelity levels.

Building Information Modeling/Model (BIM) technologies and workflows are increasingly adopted in the DCO industry. A BIM is a digital, parametric, intelligent and object-based representation of the physical and functional characteristics of a building creating a shared database and knowledge resource for project and building information [1]. With the emergence of BIM, open and neutral data schemas were developed to enhance interoperability [2]. Interoperability is considered a key factor in streamlining information flows between different disciplines and influencing the value proposition of BIM in industry [3]. Interoperability challenges are often associated with the export and import capabilities of data models among different technologies which is one of the barriers to BIM advancement [4].

In this paper, we aim to address the conversion challenge between architectural models and structural models and among different structural analysis models. First, we review available studies in this area and we discuss the challenges of converting models between several structural analysis applications. Second, we present a BIM-based approach and its components which include: the IFC-based Unified Information Model; the conversion algorithm between BIM architectural models and structural models, and the conversion algorithms among various structural analysis models. Third, we illustrate the implementation of the proposed approach which included: Client Server (C/S) and Browser Server (B/S) technologies to transfer and display the model, and algorithms for the optimization of transmission. Finally, we explain the workflow and demonstrate the results from deploying the platforms in four complex and very large real life construction projects which were used as case studies to verify the conversion process.

2. Interoperability and Integration Enhancement: Related studies

The multidisciplinary nature of BIM is now widely acknowledged within by the DCO industry [5]. Yet, interoperability issues still persist among various BIM technologies [5]. McGraw Hill [3] states that 8 in 10 users of BIM technologies in the United States consider the lack of interoperability a limiting factor in achieving the full potential of BIM.

Over the last decade research and development activities aimed at enhancing interoperability attracted a significant interest from both industry and academia [6–8]. The common overarching

aim among these efforts is to improve the usability of BIM for various stakeholders in the DCO industry. Although IFC is a rich and neutral schema, exchanges of project data using the IFC can be affected by inaccuracies due to inconsistencies in different implementers [2].

In industry, major efforts to enhance interoperability are being undertaken by buildingSMART. They proposed the Model View Definition (MVD) as a key concept to address the interoperability challenge. An MVD is a domain-specific subset of the IFC data which can be used to exchange information for specific purposes between project partners. One of the first and most popular views is the Coordination View [9]. This view is extensively implemented in most of the commercially available BIM technologies. It allows the sharing of BIMs between the major disciplines of architecture, structural engineering, and building services (mechanical) [10]. However, the support of round-trip scenarios is excluded from the support of the coordination view [10]. In structural domain, the IFC2x3 Structural Analysis View covers the exchange requirements to transfer the structural analysis model to one or many structural analysis applications. However, it only defines the information that is exchanged between the structural design applications and structural analysis applications. The exchange between architectural design and structural design is not addressed by this view and the ways to transfer and re-use such information at project level still requires investigation efforts [11]. Another significant development by buildingSMART is the IFC4 Design Transfer View (IFC4 DTV V1.0) which was released on 10.07.2015. The objectives of the IFC4 DTV V1.0 are to enable collaboration on design elements impacting multiple disciplines and provide the capability of handing over design models to others. The support of round-trip scenarios is excluded from the scope of the Design Transfer View [12]. Given its recent release, there are not commercial software tools that are compliant with this view yet.

Jim et al. [13] presented lessons learned from the translation of BIM between various tools. The semantic incompatibility in representing product data in different authoring tools was identified as the most significant challenge. The semantic interoperability can be addressed in two ways: methodologically by defining consistent modelling styles, or technically by defining ontologies and building bridges that enable transformation. Nawari [14] addressed this challenge for wood structure. The approach used consisted of standardizing BIM using the Information Delivery Manual (IDM) and MVDs to provide a reference to data required by the wood structure design process. Sanguinetti et al. [15] presented an MVD-based system architecture approach to facilitate support for an open-ended set of analysis and assessment tools to enable feedback during architectural design. Jeong et al. [16] proposed a new approach to translate between BIM and Building Energy Modeling (BEM) using MVD.

Chi et al. [17] identified the impact and future development trends for current structural design practices. Processes for systematic modeling and interfaces for data exchanges were identified as key trend to enhance the structural design [17].

Table 1 summarizes and compares the key related studies on model conversion. Chen et al. [18] developed an IFC-based web server to generate structural models from the corresponding IFC-based architectural models. Redmond et al. [19] proposed an integrated platform that exploits the

capability of ifcXML[1] or Simplified Markup Language[2] (SML) in enhancing the BIM usability experience for various disciplines and facilitating their early input to the design phase. Deng et al. [22, 23] developed an algorithm to automatically generate structural models from the IFC-based architectural model, which was based on a transformation mechanism between an IFC-based BIM and an XML-based Finite Element Model (FEM). Liu et al. [24, 25] proposed an integration tool for exchanging information between an IFC architectural model and a PKPM[3] structural model, and a conversion platform to convert between two structural engineering tools (i.e. ABAQUS and SATWE). A similar but more universal approach was proposed by Hu and Zhang [26]. Their approach aimed to achieve a BIM-based dynamic and integrated environment for conversion among structural information models. To deliver this environment, they proposed a universal data source that shares relevant information with the corresponding linked structural analysis applications. Wang et al. [27, 28] proposed, using the ObjectARX–an Application Programming Interface (API) that is supported by AutoCAD, a software application within the AutoCAD environment to generate the information of IFC structural models and transform it into the corresponding structural model.

Table 1. A non-exhaustive list of studies into model conversion and interoperability enhancement

	BIM software to structural model		Supported structural analysis software		Conversion mode		Platform architecture			Performance		
	Common data exchange	API	Single	Multiple	One-way trip	Two-way trip	Standalone	C/S	B/S	Visual display	Portability	Scalability
Chen et al. [18]	•				•				•	•		
Redmond et al. [19]	•	•			•				•			
Deng et al. [22, 23]	•			•	•			•		•	•	•
Liu et al. [24]	•		•		•		•			•		•
Liu et al. [25]				•		•	•					•
Hu and Zhang [26]	•		•		•			•		•	•	•
Wang et al. [27, 28]	•	•	•		•			•		•		•
Proposed platforms	•	•		•		• *		•	•	•	•	•

Note: (*) The conversion is bidirectional among the structural analysis tools and one way from the architectural BIM to the structural analysis models.

[1]IfcXML files are domain specific type of XML files which are generated from BIM's IFCs with data instances identified through unique identifiers which are used to connect one data instance to other [20].

[2]Simplified Markup Language (SML) or simplified XML is a schema used for the extraction of partial data for exchanging information through an internet-based service [21].

[3]PKPM is a widely used structural engineering software in China, developed by China Academy of Building Research Technology which is one of the China's top DCO software firms.

As indicated in Table 1, the majority of the reviewed research efforts are implemented as either standalone or C/S applications, provide one-way trip conversion only, and do not have the capability to convert between both an architectural BIM and a structural BIM and among multiple structural analysis models.

The development and implementation of solutions for the integration, management and sharing of building information can supported by BIM servers such as IFC Model Server, EDM Model Server and BIM Server [29]. Emerging Web standards, such as HTML5 and WebGL, also provide routes for developing solutions for displaying 3D shapes in browsers which can be supplemented with embedded metadata to form rich Web applications [30]. For example, 3D visualization in CityGML was enabled using WebGL [31] and 3D visualisation in browsers using HTML5 is explored in bioWeb3D [32]. Efforts that are more pertinent to the building sector are those attempting to develop WebGL applications that support the IFC format. Key initiatives in this area include the BIMSurfer, IfcWebViewer and XBIM. In other industries such the oil and gas sector, where data sets are very large, WebGL-based approaches and technologies for information integration are also being explored [33] in combination with Three.js, a fast object-oriented and high level JavaScript library [34].

In this research, we address the conversion challenge between an architectural BIM and a structural BIM and among several structural BIMs among geographically dispersed DCO users. This effort will contribute to streamline information flows within a BIM-based project and eliminate inefficiencies related to the need for recreating or editing models which were already created at prior stages. We first present a thorough explanation of the differences in semantic, syntax and information representation between various structural analysis models. Second, we provide an IFC-based Unified Information Model, which forms the foundation for model conversion. Third, we propose an approach and a number of algorithms that will be used in conversion to overcome the diverse data representation and syntax used within the different structural analysis tools. Fourth, we illustrate the developed platforms (i.e. C/S and B/S) for model display and sharing. Finally, we present and discuss the results from testing the proposed approach and tools in four complex real life case studies and we outline recommendations for future work.

The key characteristics of the proposed approach and platforms compared to related studies in Table 1 are:

- The IFC-based Unified Information Model exploits the emerging approach for exchanging information between DCO users, which is based on centrally shared data model. The shared data model organizes data elements and standardizes how the data elements are represented and how they relate to one another. Applications around the centrally shared data model can access and retrieve required data. Together the IFC-based model and the embedded algorithms around it, to overcome semantic, syntax and information representation between the corresponding applications, help to achieve a bidirectional conversion within a BIM based workflow.

- The proposed approach is not limited by specific set of interfaces embedded within specific applications. By sharing a universal data model centrally, the solution can accommodate further applications around the central model thus, providing a scalable solution.

- The conversion is achieved over both C/S and B/S enabling the solution to meet the need of DCO users who can share and collaborate with each other centrally or remotely. Furthermore, the platforms can support the visualization of not only 3D models and their components properties but also the result of structural analysis performed in different applications.

3. Differences in the semantic and syntax among structural analysis software tools

In complex construction projects, several structural analysis models and technologies are utilized to verify the structural safety and compliance of the proposed design. The current capability of model transformation is limited to a one-way direct conversion between some of these technologies (see Fig. 1). As a result, processes and information need to be repeated and recreated entailing further time, cost and manpower and resulting in an inefficient workflow. With a one-way trip transformation, the checking of the consistency and accuracy of the transferred information is also challenging. BIM concepts, standards and technologies bring about new perspectives and ways to address these challenges. As the literature review evidences, one of the key BIM-based approaches to enhance interoperability between different technologies is through shared data models that organizes data elements and standardizes their representation. This approach has not been fully achieved to resolve the conversion between an architectural BIM and a structural BIM and among several structural analysis BIMs. In this paper, we introduce the concept of using an IFC-based Unified Information Model which act as an integrated information layer for the functions of model conversion (two way trip) among multiple structural analysis models and technologies (see Fig. 1). This approach based on the Unified Information Model and augmented by algorithms that overcome the differences in the representation syntax and grammar of various structural analysis tools provides the following advantages: (a) avoid the need to make changes to the data structure of commercial structural analysis tools, (b) enable an open bidirectional conversion between several commercial structural analysis tools through the Unified Information Model and has the scalability to accommodate new tools in future. The proposed approach and tools lay the foundation for developing a new generation of model conversion technologies among multiple platforms especially in the structural engineering discipline.

Fig. 1 shows some of the main design and structural analysis technologies affected by the conversion challenge. A typical structural analysis model mainly consists of geometry, materials, sections, loads, constraints among others. Fig. 2 illustrates the differences of data structure, semantic and syntax affecting coordinates references, geometry, materials, etc. among these technologies. Overcoming these semantic and syntax discrepancies between the different models and technologies is a key challenge for the conversion process. In the proposed approach and tools, this challenge was addressed for the following technologies/file formats: ETABS / .e2k; SAP2000 / .s2k; MIDAS / .mgt; ANSYS / .mac and all IFC supported technologies.

The challenge derived from the inconsistent representation of data models (e.g. geometry, materials, sections, loads, constraints, etc.) is also exacerbated by the multiple relationships or instances that each entity can have. For example, entities under ‘geometry’ include ‘joint’, ‘frame’ and ‘area’ information. Each of these entities could have its own material and section information and may or may not have a load bearing role. Therefore, the proposed approach does not focus

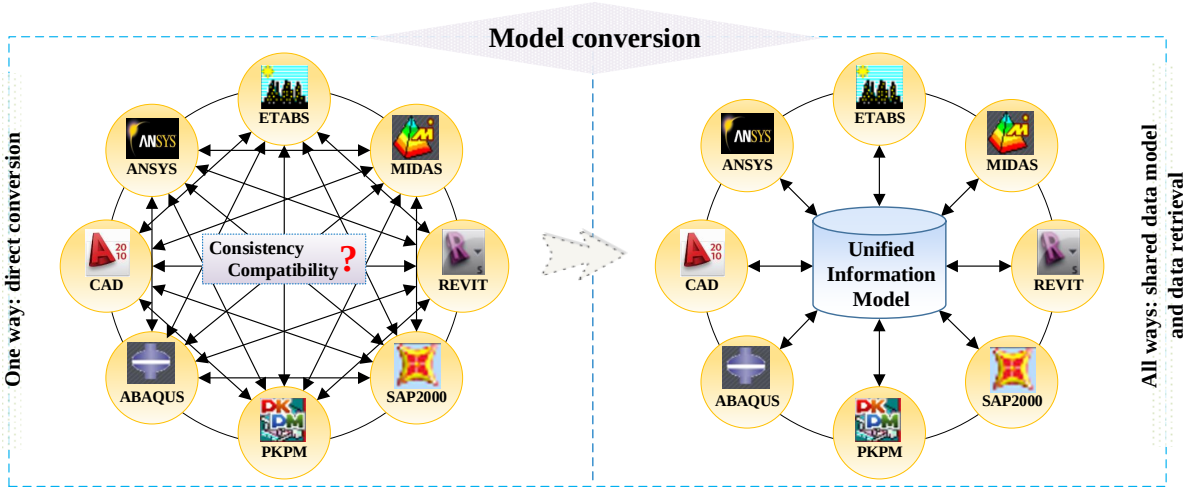


Fig. 1. Different types of model conversion among various software applications

only on solving the distinct representation of such entities among various technologies, but also on capturing and considering the relationships between them.

4. Approach for BIM-based model conversion

4.1. IFC-based Unified Information Model

The IFC-based Unified Information Model is a data model that is utilized in our proposed approach to bridge the gap between different structural analysis technologies and reduce the inefficiencies associated with the need to recreate information. The model organizes data elements and standardizes how the data elements are represented and how they relate to one another (see Fig. 1). The model concurrently achieve these objectives: (a) overcomes semantic, syntax and information representation between various structural analysis models; (b) provides a centralized sharing layer and a universal information exchange, and (c) enables information exchange between geographically dispersed professionals when it is implemented over a network technology. It also complements the ongoing effort by buildingSMART and in particular, the Structural Analysis View and the IFC4 Design Transfer View (IFC4 DTV V1.0).

The upper part of Fig. 3 shows the data included within the Unified Information Model and their structure, relationships and management. The detected architectural information includes all architectural components of a building. Each architectural component is assigned to a building storey and linked to Solid3D entities that visualize the structural performance of a building part. The mesh representing each architectural component is divided into a number of triangles to improve the display of results. Structural components are referenced to the global 3D Cartesian coordinate system. The structural information includes all structural elements and their attributes such as axis and profile of the elements. The key structural information associated with structural component include: structural loads, boundary conditions and the different load cases and their combinations. The *IfcRelAssociates* entity encapsulates the properties of the material and section.

		Information expression			
File type		ETABS (*.e2k) SAP2000 (*.s2k) MIDAS (*.mgt) ANSYS (*.mac) IFC (*.ifc)			
Geometry 	ETABS	POINT "pt name" {x} {y}	LINE "line name" [COLUMN / BEAM / BRACE] "pt1" "pt2" [1/0] [1/0]		
		AREA "area name" [FLOOR/PANEL] {number} "pt1" "pt2" "pt3" "pt4"…… [1/0] [1/0] [1/0] [1/0] …			
	SAP2000	Joint={pt name} CoordSys=GLOBAL CoordType=[Cartesian/Cylindrical] XorR={value} Y={value} Z={value}			
		Frame={line name} JointI={pt1} JointJ={pt2} IsCurved=[Yes/No] Length={value}			
		Area={area name} NumJoints={number} Joint1={pt1} Joint2={pt2}… Perimeter={value} AreaArea={value}			
	MIDAS	{pt name}, {X}, {Y}, {Z}	{line name}, "TYPE", {mat number}, {sec number }, {pt1}, {pt2}, {angle}		
		{area name}, "TYPE", {mat number }, {sec number }, {pt1}, {pt2}, {pt3}…, { [1/2], 1-thick，2-thin}			
	ANSYS	K, {point number}, {X}, {Y}, {Z}	LSTR, P1, P2	A, P1, P2, P3, P4…	V, P1, P2, P3, P4, …
	IFC	IfcProduct—ObjectPlacement: spatial location information + Representation: geometric shape information			
Material 	ETABS	MATERIAL "mat name" M {mass} W {weight} TYPE ["ISOTROPIC" / "ORTHOTROPZC"] E {e} U {u} A {a}			
	SAP2000	Material={mat name} Type=[Concrete/Steel/…] SymType=[Isotropic/Orthotropic /…] TempDepend=[Yes/No]			
	MIDAS	{mat number}, {Type[Concrete/Steel/…]}, <Data>			
	ANSYS	MP, [ex/alpx/prxy/gxy/dens/……], [material number], C0, C1, C2, C3, C4…{value}			
	IFC	IfcMaterialProperties			
Section 	ETABS	FRAMESECTION "sec name" MATERIAL "mat name" SHAPE "type" {parameters}			
		SHELLPROP "sec name" MATERIAL "mat name" PROPTYPE ["WALL"/"SLAB"…] TYPE ["SHELL"/"PLATE"] {thickness}			
	SAP2000	SectionName={sec name} Material={mat name} Shape=[Rectangular/Circle…] {parameters}			
		Section={sec name} Material={mat name} MatAngle={value} AreaType=[Shell/Plane/Asolid] Thickness={value}			
	MIDAS	{sec number}, {TYPE}, {shape name}, <OFFSET>, {SHAPE}, <DATA>			
	ANSYS	SECTYPE, SECID, Type[beam//joint/shell/…], Subtype[SECDATA/SECOFFSET], Name			
	IFC	IfcProfileProperties— IfcProfileProperties + IfcRibPlateProfileProperties			
Load / Restraint / Other information ……					

ABC —PropertyName
{Value} —PropertyValue
[A/B/⋯] —OptionValue

Fig. 2. Information representations in various structural analysis tools

They can be shared with both the architectural information model and the structural information model.

The bottom part of Fig. 3 shows the data management layer. Due to the size of the models that can be involved in the conversion process, this layer must enable effective storage space and efficient operation. This objective can be achieved by selecting adequate representation and storage of 3D models and data. There are many schemes of representing solids including parameterized primitive instancing, Constructive Solid Geometry (CSG[4]), Boundary representation (B-rep[5]), Surface mesh modeling and so on [37]. These representations have different ways of organizing the same geometric and topological data in the form of a data structure. The modeling space of any particular representation is finite, and any single representation scheme may not completely suffice to represent all types of solids [38]. The ability to convert representations in a scheme into corresponding representations in other schemes is therefore of great practical importance for well-engineered modeling systems [37]. In this research, we mainly use a combi-

[4]CSG is a solid modeling technique that utilizes Boolean operators to combine objects and create a complex surface or object [35].

[5]B-rep is a method for representing shapes using the limits where each solid is bounded by its surface elements such as rectangles or triangles [36].

nation of solid model and triangle based B-rep model. Solid model consists in totally recording the modeling process and geometric properties including the combination of solid atom and CSG. Solid atom consists of profile and axis, which are based on a series of points and lines, storing the part of important information for structural analysis. B-rep of solids is familiar to most computer scientists because of their use in computer graphics. In this research a solid is represented by segmenting its boundary into a finite number of 'faces' and representing each face by triangles.

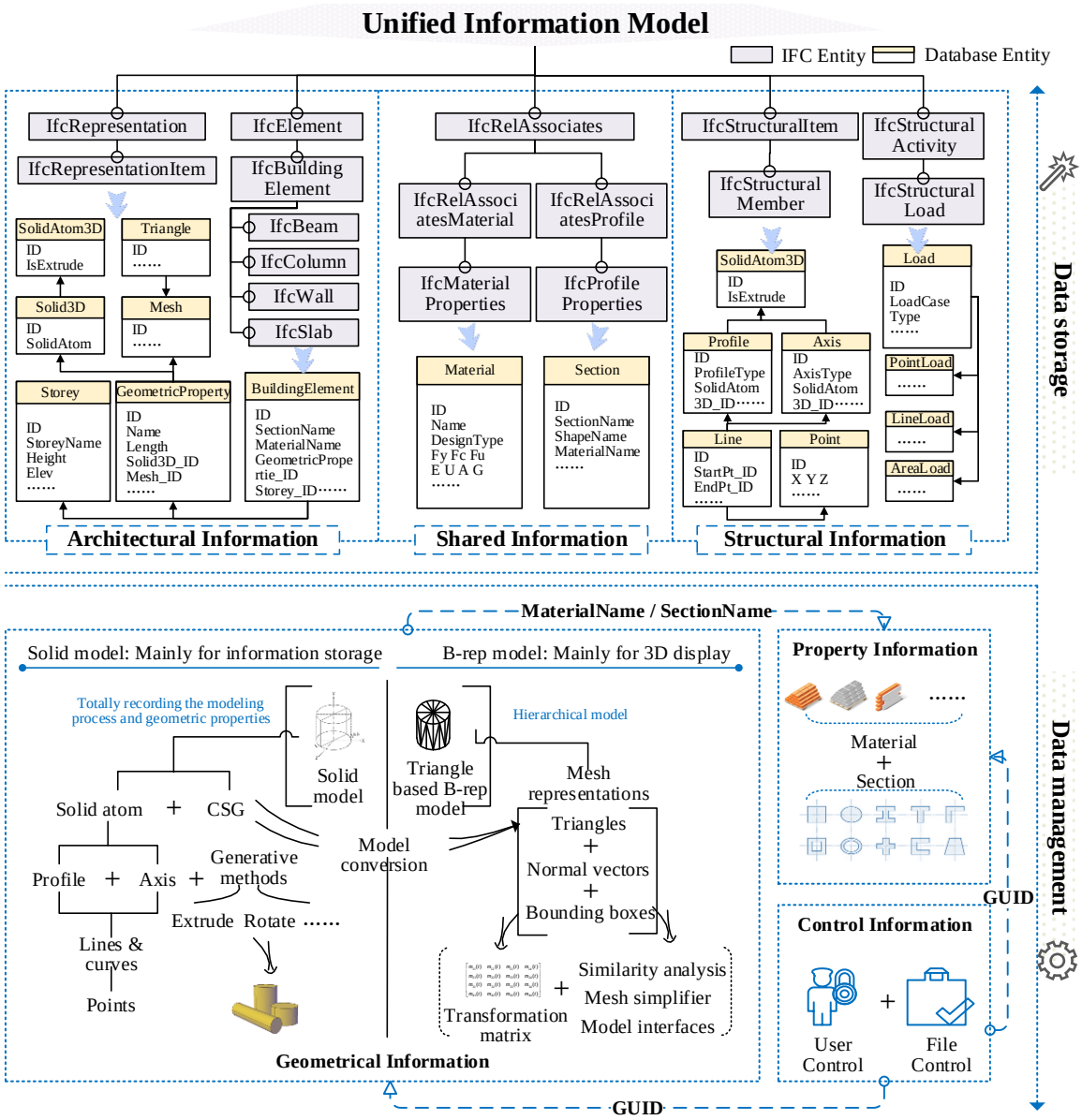


Fig. 3. Architecture of storage and management for unified information model

The detailed parameters and properties regarding the material and section properties are not associated with each component directly. Each component stores its own MaterialName and SectionName only. Their detailed parameters will be obtained based on two properties (i.e. MaterialName and SectionName) from the corresponding tables. Globally Unique Identifiers (GUIDs) are used to uniquely link each component to its properties and project file and give users the ability to control and track their information within their own models.

The structure and functions of the proposed IFC-based Unified Information Model conform with the definition of model views as a subset of the exchange schema [2]. The Unified Information Model has a structure, content and the necessary information to help in enabling conversion between architectural and structural design and among structural analysis models. It is also open and compact to create a bidirectional conversion among various commercial software tools. These features will be demonstrated and tested in more depth in the following sections.

4.2. Conversion algorithm between architectural model and structural model

The research mainly targets two types of model conversion. One is for converting between architectural models and structural models and another for converting among structural analysis applications. An *architectural model* mainly describes and represents building geometry and appearance representation while a *structural model* includes all structural elements involved in vertical and lateral load transferring. Using the structural models, structural engineers perform structural analysis by adding different load cases, geometric boundary conditions, among other information. These three model types (*architectural model*, *structural model*, *structural analysis model*) are illustrated in Fig. 4.

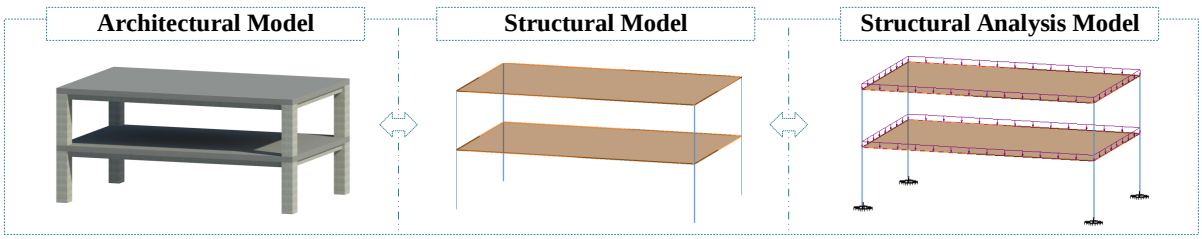


Fig. 4. Different representations of three kinds of information model

Information from the architectural model, that is pertinent to the structural model, is detected, classified, processed and transmitted to the structural model during the process of conversion between these two model types. For example, for the frame structure, first the axis and section information are retrieved from the architectural model and then the node tolerance on the non-coinciding segments are calculated in the structural joints. The computation of the tolerance for bound of near nodes (*Tol*) (see Fig. 5) is important to determine the strategy (i.e., combining or not combining) for dealing with closer nodes (e.g. cross between a beam and a column) during the conversion process. If the distance between two nodes is less than the *Tol*, they can be combined reasonably. The analysis and the detailed workflows in different situations are discussed in our previous publication. [39]. For shear wall structures, the algorithm first retrieves information

about the wall thickness and material. Subsequently, the algorithm classifies the wall types based on the values of these attributes. For a thickness than 160 mm, the wall is classified as an infill wall and is ignored during the conversion from the architectural to the structural model. If the thickness is more than 160 mm, the algorithm captures the material of the wall prior to classifying it type. If the wall is made of reinforced concrete, the algorithm will classify it as a shear wall and will push it through the conversion process.

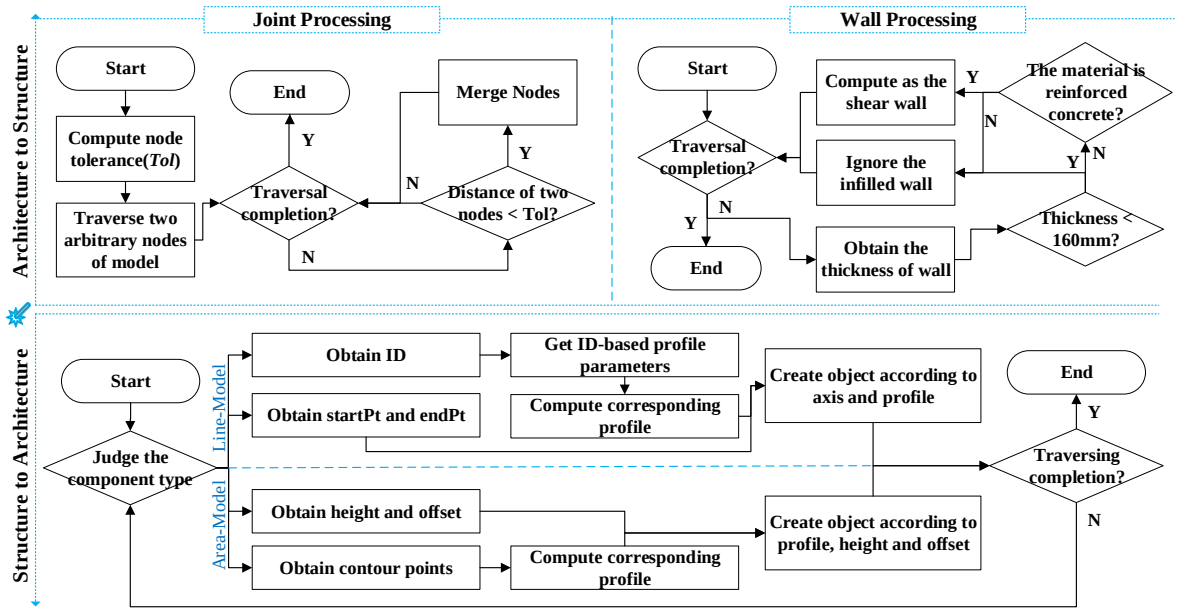


Fig. 5. Logic and details of data extraction within the conversion algorithm between structural and architectural models

The algorithm enables also the conversion from structural models to architectural models. The algorithm first classifies the types of elements (lines and areas) and then identifies their relevant geometric information from the structural model before converting them into the corresponding architectural elements. The algorithm differentiates between a 'line' model and an 'area' model. For a 'line' model representing a structural component (e.g., beam, column and brace) the algorithm detects the axis and section information. For an 'area' model representing a structural component such as slabs and walls, the algorithm identifies their outline, thickness and offset and convert them into the corresponding architectural components. Fig. 5 shows the logic and details of data conversion within the conversion algorithm between an architectural model and a structural model. The current algorithm enables the conversion of not only common objects such as walls, beams, columns, floors, with or without openings, but also complex ones like multi-bay cylindrical shells with joints and area elements. However, it cannot handle geometries such as variable arches with solid elements now.

Fig. 6 shows the whole conversion workflow from the upload of the source file, through the retrieval and transformation of data from a section, to the final 3D display. First, the source

file (mainly in text format, such as .e2k file for ETABS, .mac file for ANSYS etc.) is uploaded and the conversion of section parameters to feature points is conducted. Second, the axis and profile information of the building element from the database are generated. Third, based on the section angle and its position, the new feature points are generated and positioned in the right location using a number of algorithms. Finally, the feature points are used to generate triangles and triangle-based data are displayed into an OpenGL/WebGL-based platform. As a result, the conversion can be achieved from the text-based data to 3D representation of components over the Web where users can view and edit their models.

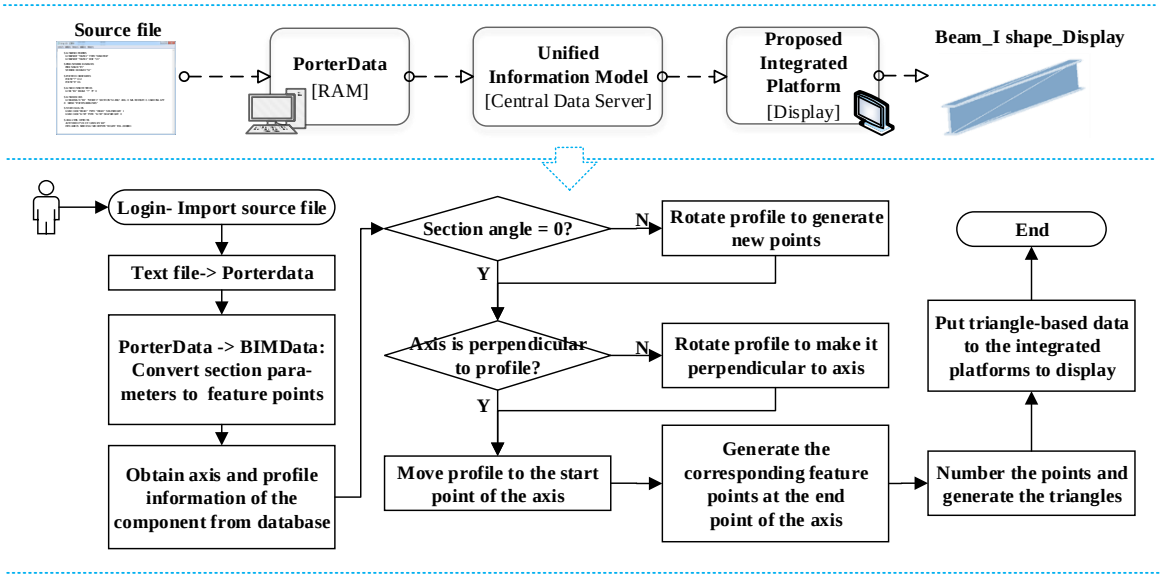


Fig. 6. Application workflow from model data file to model 3D display

4.3. Conversion algorithm among various structural analysis models

In a typical medium large construction project, several structural consultants are involved using different software tools. Such tools have inconsistent and incompatible semantic and geometric representation of data. This limits the collaboration, hinders the workflows, and adversely affects the productivity. However, main structural analysis applications support text-based input and output files and adopt similar modeling techniques. The proposed Unified Information Model converts among these mainstream structural analysis application using a text-based conversion.

To achieve conversion using this approach, there are key challenges to overcome.

The first challenge is to overcome the varying representations of the properties of structural elements among the different applications. For example, point information are represented differently among the considered structural analysis application (see Fig. 7). The proposed solution overcomes this challenge through interfaces for data mapping and validation that are positioned between the Unified Information Model and the different structural analysis applications. A second challenge is that different structural analysis applications adopt different spatial references or

origins for the same building element, resulting in an inconsistent coordinate system among these applications. For example, ETABS refers the position of each element on a storey to the local storey's based coordinates, while the other systems use the absolute global Cartesian coordinates. Corresponding algorithms among the structural analysis applications are implemented alongside the Unified Information Model to identify and resolve these inconsistencies (see Fig. 7). Moreover, the representations of other key information regarding geometry, materials and sections (as shown previously in Fig. 2) vary also between the different structural analyses applications. The differences in data structure and information representation are captured by the proposed algorithms which establish a mapping between the Unified Information Model and the selected commercial structural analysis applications to enable the conversion.

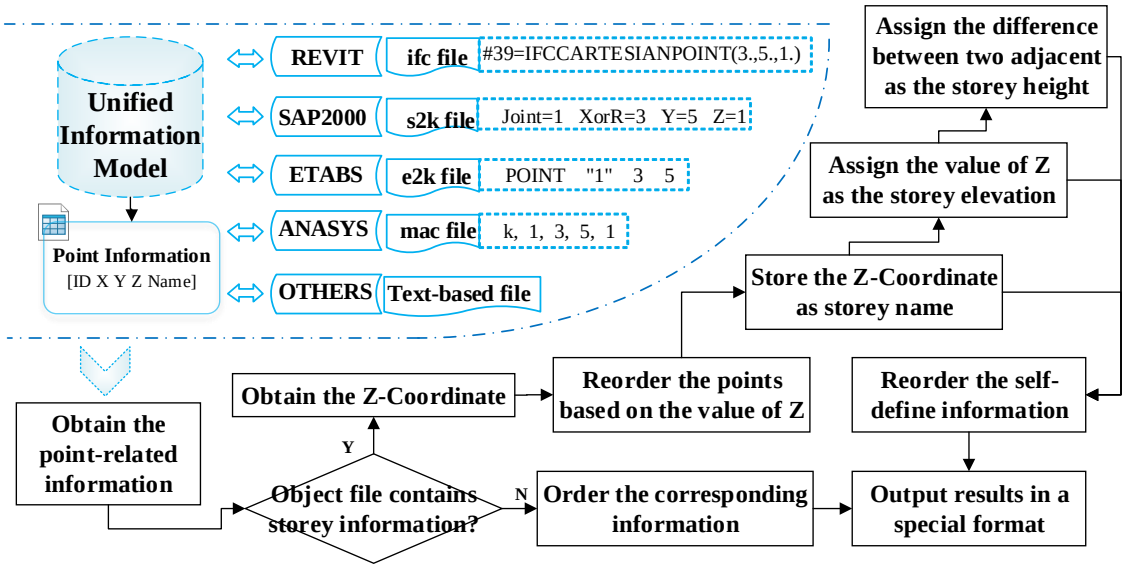


Fig. 7. Different representations of point information and referencing among structural analysis applications and the algorithms to overcome them

5. The proposed integration system

5.1. Architecture of model conversion and display

The architecture of the prototype for model conversion and display is described in Fig. 8. The IFC-based Unified Information Model is implemented as a central data server. The bidirectional data interfaces are laid between the Unified Information Model and the different data structures or file formats of different structural analysis applications.

First, the data of a source file is converted into the PorterData, which consists of a range of classes stored in the RAM. Then, the data is transformed from the PorterData to the Unified Information Model and stored in the database using special relationship and conversion methods between them. Export is an inverse process in which the conversion transforms a source file into an

object file enabling adequate portability and interoperability. The necessary data can be retrieved from the Unified Information Model, displayed using GraphicModel for OpenGL-based platform (C/S platform) and JsonData on a WebGL-based platform (B/S platform), and shared among the users of different structural analysis applications.

The B/S architecture and C/S architecture that were utilized to develop the two platforms for model display are called '*Web-BIM*' and '*Unitive-BIM*', respectively (see Fig. 8). The '*Web-BIM*' platform was based on WebGL for model display, which is a royalty-free Web standard for a lightweight 3D graphics API [40]. One of the most popular library/API for Web-based 3D graphics is Three.js [41]. It is an open-source JavaScript library which enables high-level programming of browser-based 3D scenes [42] and can be utilized to create a canvas renderer for model display. The '*Unitive-BIM*' platform was based on OpenGL (Open Graphic Library), which is a cross-language multi-platform API for rendering 2D and 3D vector graphics. It is typically used to interact with a graphic processing unit (GPU) to achieve hardware-accelerated rendering [43].

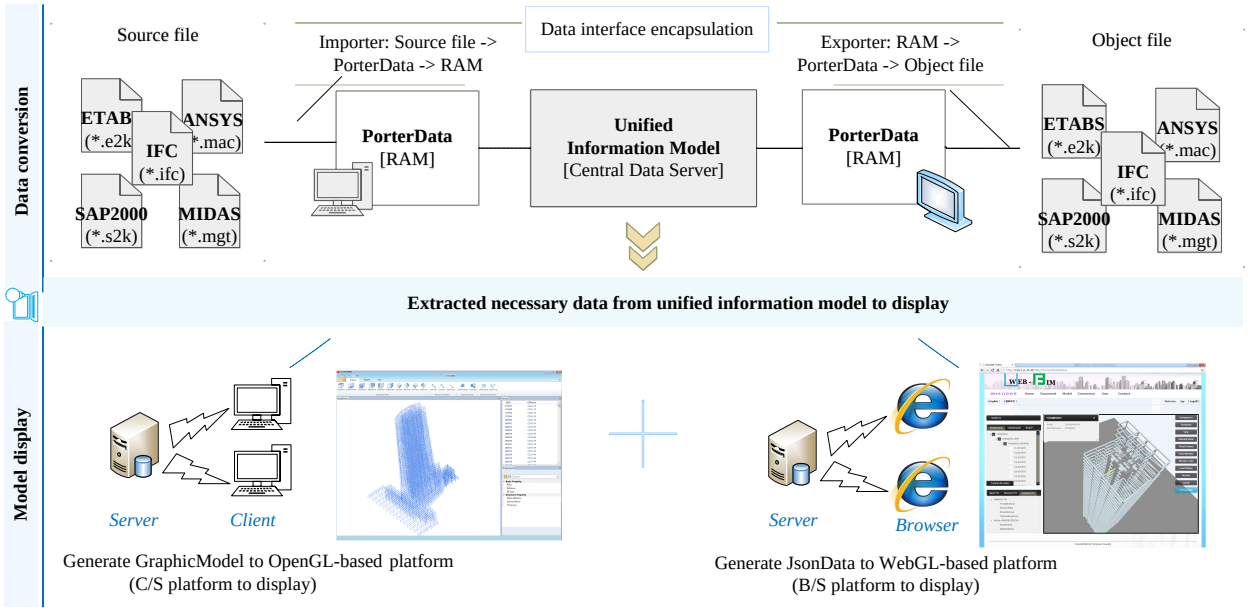


Fig. 8. Architecture of data conversion and model display

This approach for model conversion has these advantages: (a) the conversion of models and their display share the same database but are underpinned by distinct processes providing efficient operation and scalability; (b) the Unified Information Model, stored in the corresponding database, enable users to share and manage their own models, and (c) the conversion interfaces are designed in two software system architectures (i.e. C/S and B/S) to combine the advantages of both approaches in terms of offsite accessibility for geographically dispersed users by the B/S and the high interactivity of the C/S. For example, the conversion interfaces are available for both the C/S and B/S platforms and users can access through them the necessary data from the Unified Information Model and generate the corresponding information for their model display. Also the conversion algorithm is executed only once and the results are shared over the two platforms hence, providing efficiency in, and accessibility to, the system operation.

5.2. Optimization of the transmission of model information

Real world projects can be considerably large and complex. Their communication and display, especially their 3D content display over B/S platforms, can be challenging to achieve technically with suitable performance (e.g. time, rendering, etc.). To address this challenge, this research adopted two ways for model transmission through the B/S: the JavaScript Object Notation (JSON) interface and the method of compression-based web transmission.

JSON is a lightweight, text-based, language-independent data interchange format [44]. It is suitable for the front-end development of the proposed system and is used as the format of data interface to transform all necessary information from server to client. JSON is built on two structures: (a) a collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array, and (b) an ordered list of values, which is mostly realized as an array, vector, list, or sequence [45]. Fig. 9 shows the structure of the JSON data. The interface is a special object, which contains all necessary information to be transformed from server to client. It aims to achieve an efficient and accessible conversion for users. This interface mainly involves 25 objects and 9 enum types (see Fig. 9). The core of the JSON data is the spatial structure and the properties of all its objects are defined including their geometric, material, structural and type properties. All this information can be conveyed to the Web for users to check and analyze. The structure and content of this JSON data interface provide an adequate solution for web-based BIM information conversion and lays the foundation for web-based model display.

Reducing the size of the data to decrease the storage space and transmission time is essential in information and communication systems [46]. Nowadays, there are many data compression algorithms including Huffman coding, arithmetic coding, LZ series algorithm and so on [47]. GZIP (GNU zip) is a compression utility designed to be a replacement for compress. Its main advantages over compress are much better compression and freedom from patented algorithms [48]. All these algorithms support data compression for unknown data streams. This research has a clearly defined data interface which provides the possibility for deploying also a fixed dictionary-based targeted compression. Therefore, a web-oriented fixed compression dictionary technique was also adopted (see Fig. 9). Based on the definition of the JSON data interface, the object names can be coded as fixed dictionary as they are repeatedly used in the data interface and would otherwise, occupy large storage memory. The fixed dictionary is generated after obtaining the name list into the interface file and eliminating duplicates and short terms. Once the fixed dictionary is established, it can be embedded into the script file of the Web and the original content in the fixed dictionary is changed from multiple transmissions into a single transmission, enabling a more efficient transmission. The transmission of data from the server to the web-client starts with the server executing the code process for JSON information based on the fixed dictionary. Then, the GZIP compression, transmission and decompression are executed regularly. Finally, the compressed information will be decompiled by the fixed dictionary defined in the web-client. The method facilitates data processing and provides the benefit of reducing storage memory and decreasing the traffic in data transmission.

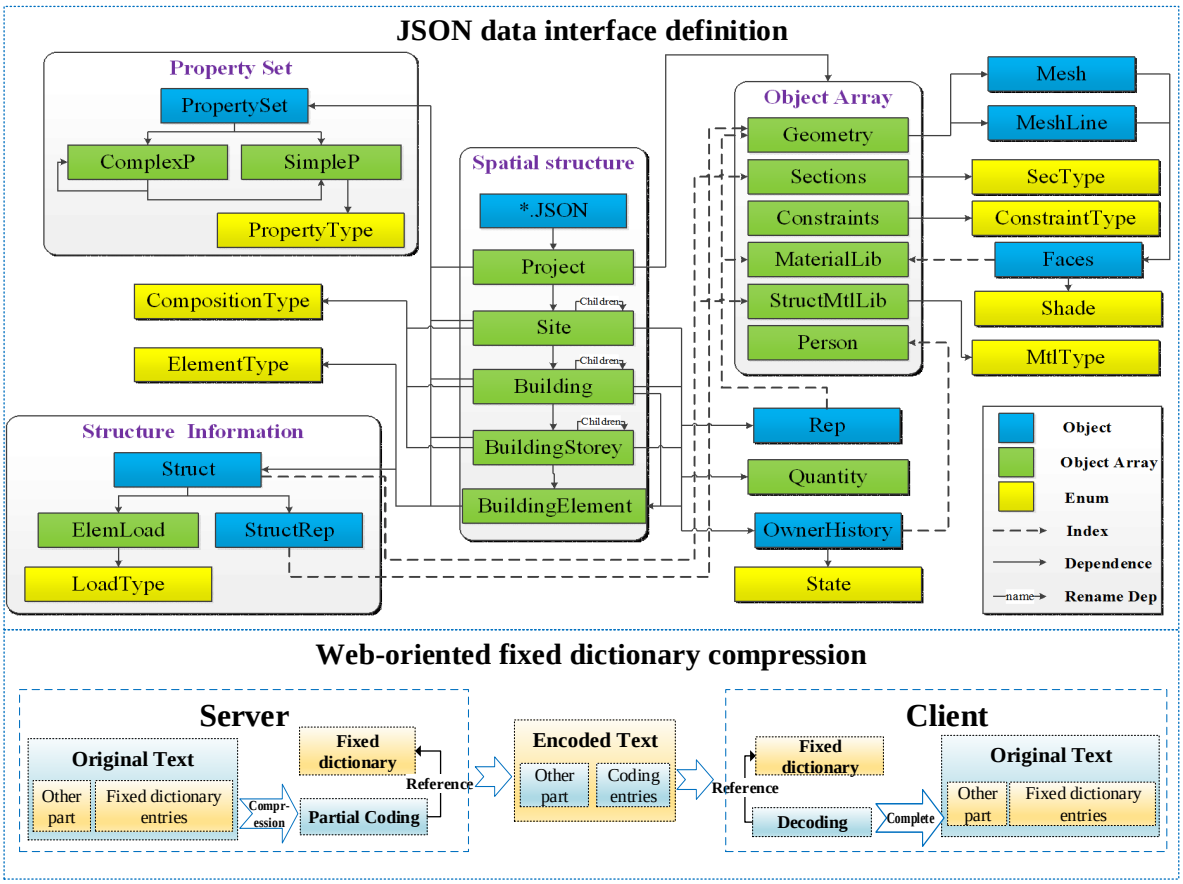


Fig. 9. Data interface and compression techniques for the optimization of information transmission

5.3. Prototype architecture and user workflow

Integration and conversion technologies in the DCO are often C/S architecture based. In this research, the new web standard HTML5, which supports WebGL technology [49], was utilized to achieve model transformation and 3D display of BIM over a browser. Several modern technologies were selected to enable the design of the prototype and user-interface, the presentation and display of models, the real-time interaction, and the optimization of conversion and display of results (see top lane in Fig. 10). Fig. 10 illustrates the technology framework and the user workflow. First, model files are submitted by users to the server in one of the structural analysis file formats. The necessary information will be stored in the database through the conversion interfaces. When the model to display is selected, the corresponding information are retrieved from the database through special GUID and transformed into JSON data using the corresponding interface. This information is then transferred from the server to the client in string format and converted to JavaScript objects using the method JSON.parse() method. These objects are then loaded onto the user's Web page in two parts. Geometric information is loaded into the page canvas, while other information such as spatial structure and property set are loaded asynchronously into the

control tree hence, providing efficiency and speed in the display process. Once all the necessary information is loaded, a series of calculations are activated to acquire the bounding box of the whole building and adjust the corresponding parameters. At this stage, key functionalities such as model rendering, model checking, model editing, 3D architectural or structure representation, can be executed. Finally, the corresponding files can be exported to other commercial structural analysis systems as required.

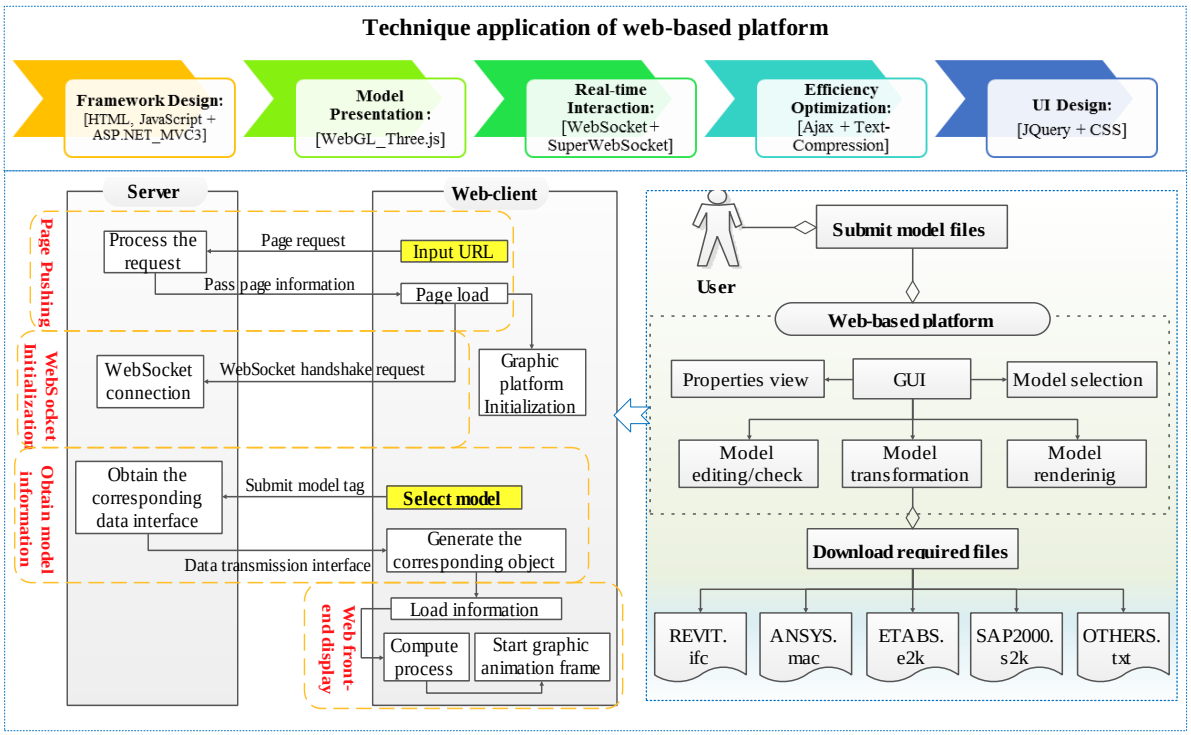
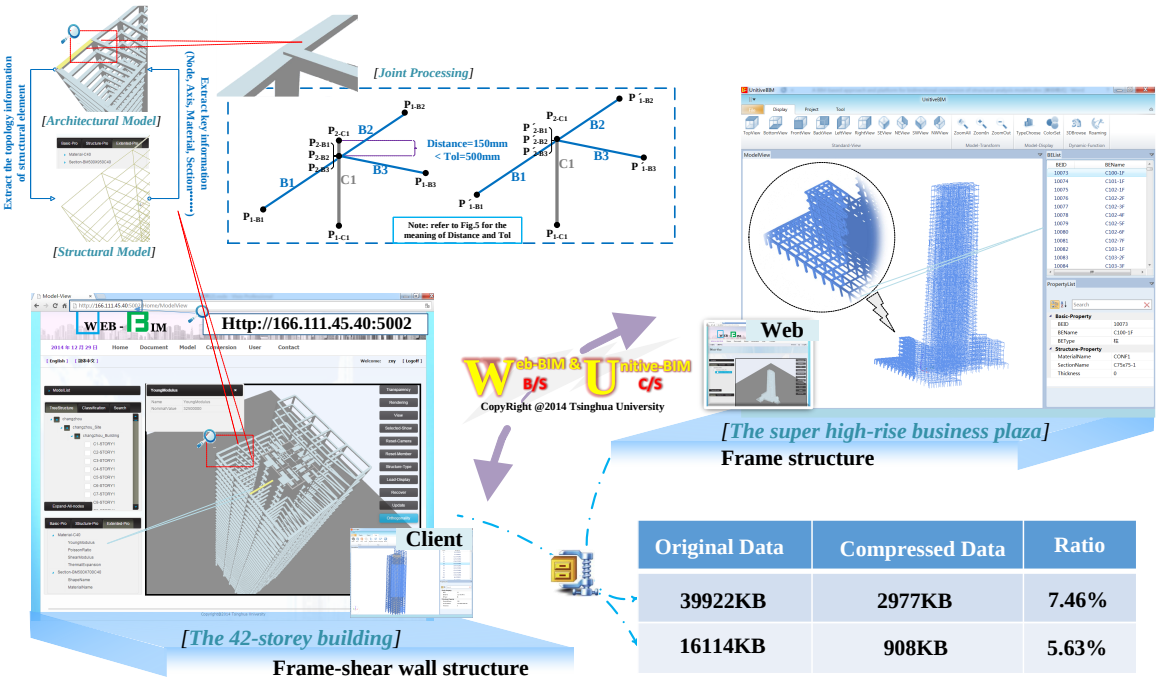


Fig. 10. Framework and application flow of web-based platform

6. Case studies

Four different very large case studies were used to test the different platforms. The '*Web-BIM*' (B/S) and '*Unitive-BIM*' (C/S) were deployed in two real life projects. The project displayed on the right side of Fig. 11 is the Business Plaza (Project A) which is a high-rise building with a height of 240.6 meters and is made of concrete frame structure. The project displayed on the left side of Fig. 11 is a 42-storey building (Project B) with a frame-shear wall structure. The structural models for both projects were firstly developed in ETABS structural application (in 'e2k' format with file sizes of 8.4MB and 3.5MB for Project A and Project B, respectively) and they included thousands of building elements and several types of materials and sections. Fig. 11 shows the interfaces of the two platforms and the conversion outputs from the two projects over the B/S and the C/S platforms. In both projects, the two platforms (B/S or C/S) were capable to correctly

import the whole structural models and convert them to structural analysis models in other file formats (i.e. .s2k, .mac and .mgt) (see Fig. 12). Their models are centrally stored and managed in the server regardless of the utilized platform (i.e. '*Web-BIM*' or '*Unitive-BIM*'). As a result, the models can be managed using one account but they can be displayed in different platforms at the same time. The key challenge for displaying 3D content in Web browsers using the '*Web-BIM*' platform was successfully overcome. The outputs, illustrated at the left side of Fig. 11, show a very adequate quality of rendering for this type of engineering application despite the significant size of the tested project files. The optimization algorithms performed exceptionally well by achieving high compression ratio. In the Project A and the Project B the achieved compression ratios were 7.46% and 5.63%, respectively. These compressed ratios are high as common Web3D data format have rather small compression ratio [50]. As a result, the two platforms were capable of delivering an efficient transmission of data.



Original Data	Compressed Data	Ratio
39922KB	2977KB	7.46%
16114KB	908KB	5.63%

Fig. 11. Testing of the two prototypes in large real life projects

All the four possible conversion routes (i.e. conversion from each structural analysis application towards the other three applications and towards the source application itself) were tested resulting in 16 conversions for each of the two case studies. Fig. 12 shows the results for the conversion from ETABS application towards other applications. To validate the accuracy of the conversion, the source file (in 'e2k' format) and the converted files (.s2k, .mac, .mgt) imported into corresponding applications, were utilized to calculate and compare the first four steps of natural mode shapes and their natural vibration periods of the conversion models. This process mainly considers two factors (i.e. the mass and the stiffness) which can demonstrate the conversion accu-

racy for material, section, constraint information, etc. from the side. The results for the reinforced concrete frame structure (Project A) and the frame-shear wall structure (Project B) are summarized in Fig. 12 and they were all obtained within a standard deviation of 0.1s and 0.2s, respectively for Project A and Project B. This testing, combined with the manual checking of converted models and their rendering quality, demonstrate that the IFC-based Unified Information Model and the two platforms enable a reliable conversion between different structural analysis applications. These results also prove the capability of the proposed approach in enabling a round tripping among the considered structural analysis applications. Indeed, all relevant information for every conversion route is stored into the Unified Information Model and converted accordingly by its algorithms between each pair of structural analysis application.

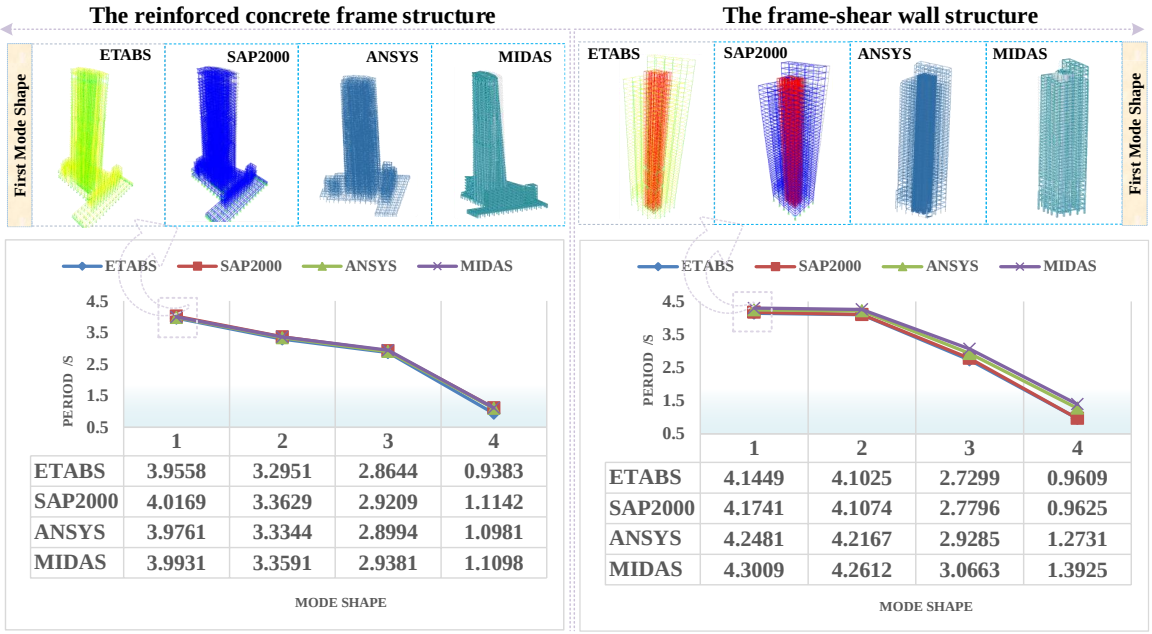


Fig. 12. The periods of four mode shapes of the conversion analysis models

An important scenario is to compare the proposed approach and platforms, based on a centralized data model, with the direct link conversion approach between two structural analysis applications. This scenario was tested in a mega project called 'Ping An International Finance Centre', also known as 'Ping An IFC' (see upper left part of Fig. 13). The 'Ping An IFC' is a 115-storey skyscraper under construction in Shenzhen, Guangdong province, China. It is expected to be completed in 2016 and will become the 4th tallest building in the world. When the source file was imported to ETABS (File size of 11MB) and the conversion to SAP2000 was tested using the direct link, a message displaying several warnings appeared indicating a failed conversion. Direct conversions from ETABS to MIDAS and ANSYS were also neither supported. However, the same conversion routes were successfully achieved through the developed platforms (see upper left part of Fig. 13).

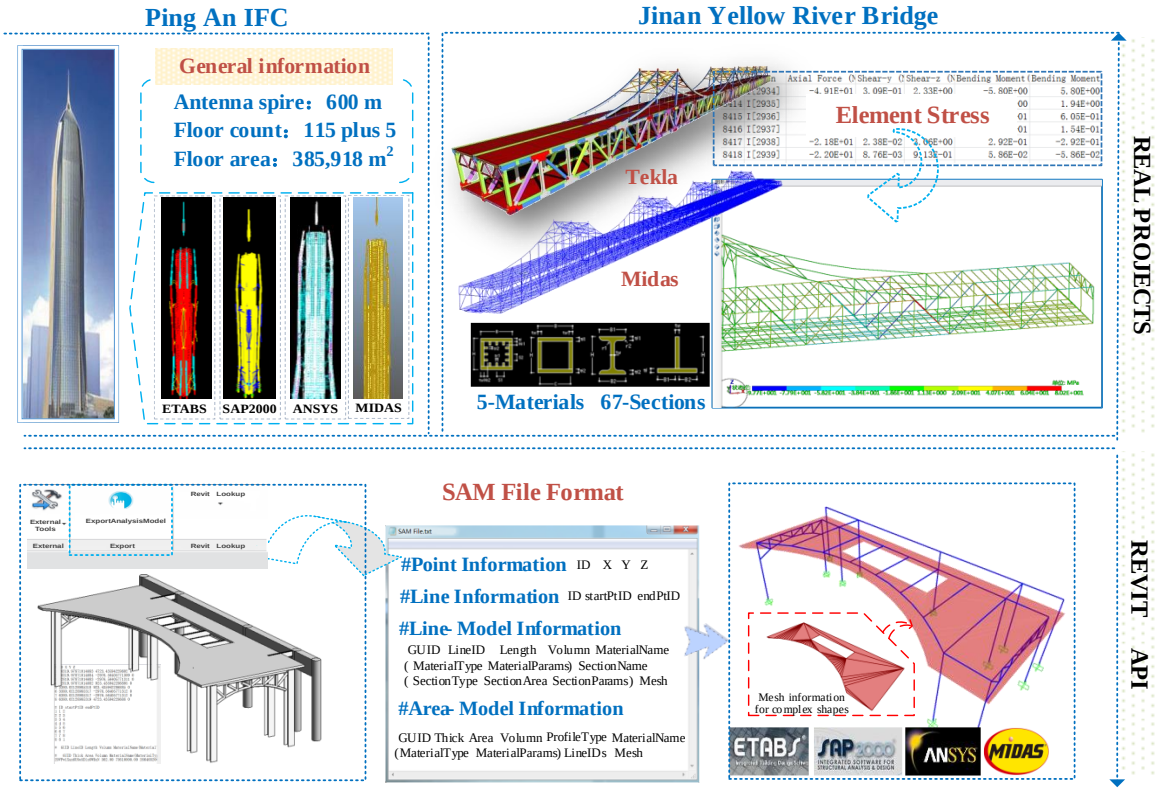


Fig. 13. Two real projects and exploration for Revit API

The conversion from common BIM platforms (e.g., Revit) to structural analysis software tools (e.g., Etabs or SAP 2000) is another important area of investigation. Some structural analysis software vendors provide embedded plugins to perform the conversion between established BIM architectural models and structural models. However, this conversion is often perceived using proprietary file formats. This research in addition to the conversion approach via the IFC-based unified approach, explored this direct conversion. An Application Program Interface (API) for Revit was adopted to convert complex objects. The API retrieves the relevant information from an architectural model to a Structural Analysis Model (SAM) file format. This conversion process employs existing and enhanced algorithms that enable the mesh transformation of complex objects with openings. The SAM is then imported into other structural analysis applications (i.e., ETABS, SAP2000, ANSYS and MIDAS). The testing of direct conversion from IFC-supported tools to structural analysis applications via this API was successfully achieved (see bottom part of Fig. 13). The advantages of this approach over the embedded conversion plugins available commercially are: (a) Improved reach and collaboration: the conversion can be implemented in both C/S and B/S environments to enhance the local or remote collaboration between geographically dispersed users; (b) Improved versatility: the proposed conversion interface can be implemented in a single

plugin tool to serve several structural analysis technologies hence, avoiding the need for ad-hoc conversion solutions by each software vendor, and (c) Improved user flexibility: the content that requires converted can be selected flexibly according to users specific requirements. However, compared to the embedded tools in commercial solutions the proposed conversion approach falls short in the conversion of complex shape. Improvement in the conversion of complex shapes shall be further explored.

Another significant testing scenario is the conversion and visualisation of the results of the structural analysis among different applications. This scenario was tested in real life project called 'Jinan Yellow River Bridge' in Jinan, Shandong Province, China (see upper right part of Fig. 13). The size of the model in Tekla in IFC format was 900MB. This project is co-developed recently with 'Steel Structure Engineering Co.Ltd.of China Tiesiju Civil Engineering Group' and included complex sections and different types of material. The stress results for the beam elements were successfully transferred and the corresponding stress nephogram was displayed in the developed platform (see upper right part in Fig. 13).

A further challenge to the interoperability in the DCO industry is the versioning compatibility. This challenge should be considered for conversion approaches and platforms. The solution proposed in this paper adopts an interface that utilizes text-based files (e.g., *.e2k, *.s2k, etc.), which are not usually much affected by versioning and software updates. This provides the proposed solution compatibility with multiple software vendors over the years. The potential maintenance of the proposed solution will only require some minor adjustment to the interfaces to guarantee the interoperability among the mainstream structural analysis software tools (SAP2000 v14, v15 are all supported now). As new features are frequently incorporated into the BIM authoring tools, coupled with potential new version of IFC, the proposed solution can be updated by capturing the changes and ensuring compatibility with different versions. The current IFC interface complies with the latest certified IFC version (IFC4) implemented by commercial BIM Authoring tools.

Currently, the two platforms (C/S and B/S) are freely opened, with versions in both Chinese and English, for users to perform conversion of structural analysis models. Users can use it to convert in all directions (round trip conversions) all entities included in the Unified Information Model (e.g. beams, columns, slabs, areas, sections, materials, etc.). Other available open source applications such as Autodesk A360, IfcWebServer, Solibri Model Viewer are mainly for viewing purpose and they do not enable model conversion and editing. Our proposed platforms support not only model viewing but also model conversion and editing over the Web (the B/S). Users can convert between structural analysis models from different proprietary technologies as and when required. In addition to enabling the conversion, the proposed platform enables the revision and editing of the properties (structural basic and extended property set) of the elements of the corresponding structural representation. However, the current solution does not support the conversion of finite element topology and detailed results in finite element meshes as these are not represented in the IFC-based Unified Information Model. It also considers static forces only. Dynamic forces and pre-stressed loads are still not addressed. These limitations will be also gradually addressed in future work.

7. Conclusions

Inadequate interoperability still persists among the project functions at different phases (e.g. design, construction and operation) and within the same phase (i.e. design phase). This study addressed the challenges of data conversion between an architectural BIM and a structural BIM and among several structural analysis BIMs.

To achieve this conversion challenge, this study proposed an IFC-based Unified Information Model and several algorithms. The IFC-based Unified Information Model formed an integrated central information layer for model conversion. It standardized the entities, their attributes and relationships required for such a conversion. The algorithms helped to overcome the inconsistent representations of data and information among different structural analysis applications. The proposed solution was prototyped in two platforms namely, the '*Unitive-BIM*' platform (C/S) and the '*Web-BIM*' (B/S). The prototyping process exploited an array of modern technologies to address key challenges such as the transmission and display of 3D models over the Web and the optimization of model transmission. The former challenge was addressed by generating triangles and triangle-based data that are transferred and displayed into an OpenGL/WebGL-based platform. The optimization of model transmission employed the GZIP compression algorithm and a fixed dictionary that was held in both the server and web-client. On the other hand, some real life projects demonstrated satisfactory performance in terms of conversion quality, accuracy and latency.

Four real life case studies were performed to test several scenarios including the bidirectional conversion among structural analysis tools; the comparison of the proposed conversion approach and the conversion through direct link between the structural analysis tools; the direct export from IFC-based tool through API, and the visualization of structural analysis results. Despite the significant size and complexity of the real life projects, all tested scenarios were successfully executed. The bidirectional conversion among four structural analysis applications (ETABS, SAP2000, ANSYS and MIDAS) was validated by executing all possible 16 conversion routes in the first two case studies (referred to in the paper as Project A and Project B). The calculation and comparison of the first four steps of natural mode shapes and their natural vibration periods of the conversion models were within a standard deviation of 0.1s and 0.2s in the two case studies. This validation process considered two factors (i.e. the mass and the stiffness) which proved the conversion accuracy for material, section, constraint information, etc. from the side. In the 'Ping An IFC' case study the proposed solution successfully achieved the conversion among the involved platforms, while the direct link approach failed to complete the conversion. Compared to the direct link conversion approaches, currently implemented in commercial tools, the proposed platforms enabled improved reach and collaboration (i.e., local and remote access), versatility (i.e., use with multiple commercial tools) and flexibility (i.e., adaptable to specific user requirement). The 'Jinan Yellow River Bridge' case study was used to test the capability of the proposed solution of converting and displaying structural analysis results. The capabilities were demonstrated by the successful conversion and display of the stress nephogram of beam elements.

These results demonstrated that the proposed approach and the two platforms are promising developments for addressing the challenge of conversion between architectural models and structural analysis models and among different structural analysis applications. Offering this conversion

over both a C/S and a B/S platforms enables the proposed solution to meet the need of a large number of DCO users who can share and collaborate centrally or remotely. This effort complements key buildingSMART efforts such as the IFC2x3 Structural Analysis View and the IFC4 DTV.

The key limitations to address in future work include: (a) the conversion of complex objects and FEM mesh which is challenging due to the wide-ranging representations of information for openings, members offset and FEM meshes among software applications, and (b) the classification and combination of different loads and restraint conditions and the conversion of analysis results including strains and reinforcements.

Acknowledgements

We are grateful for the support provided by the National Natural Science Foundation of China (No.51478249), the National High-tech Research and Development Program of China (No.2013AA041307) and the Tsinghua University-Glodon Joint Research Center for Building Information Model (RCBIM).

References

[1] NIBS, National Building Information Modeling Standard - version 1.0 - part 1: overview, principles and Methodologies, http://www.wbdg.org/pdfs/NBIMSv1_p1.pdf, 2007.

[2] M. Venugopal, C. M. Eastman, R. Sacks, J. Teizer, Semantics of model views for information exchanges using the industry foundation class schema, Advanced Engineering Informatics 26 (2012) 411–428.

[3] N. Young, S. Jones, H. M. Bernstein, J. Gudgel, The Business Value of BIM-Getting Building Information Modeling to the Bottom Line, Technical Report, McGraw-Hill Construction, New York, NY, U.S., 2009.

[4] C. Eastman, Y. Jeong, R. Sacks, I. Kaner, Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards, Journal of Computing in Civil Engineering 24 (2009) 25–34.

[5] P. Bynum, R. R. Issa, S. Olbina, Building Information Modeling in Support of Sustainable Design and Construction, Journal of construction engineering and management 139 (2012) 24–34.

[6] J. Plume, J. Mitchell, Collaborative design using a shared IFC building model–Learning from experience, Automation in Construction 16 (2007) 28–36.

[7] M. H. Serror, J. Inoue, Y. Adachi, Y. Fujino, Shared computer-aided structural design model for construction industry (infrastructure), Computer-Aided Design 40 (2008) 778–788.

[8] R. Jardim-Goncalves, A. Grilo, Building information modeling and interoperability, Automation in Construction 19 (2010) 387.

[9] C. Eastman, I. Panushev, R. Sacks, M. Venugopal, V. Aram, R. See, E. Yagmur, A guide for development and preparation of a national bim exchange standard, buildingSMART Report (2011).

[10] buildingSMART, Coordination View Version 2.0, http://www.buildingsmart-tech.org/specifications/ifc-view-definition/coordination-view-v2.0, (accessed August, 2015).

[11] buildingSMART, Structural Design to Structural Analysis, http://www.buildingsmart-tech.org/downloads/view-definitions/structural-analysis-view/StructuralAnalysisView_Overview.pdf, (accessed August, 2015).

[12] buildingSMART, IFC4 DTV Objectives, http://www.buildingsmart-tech.org/specifications/ifc-view-definition/ifc4-design-transfer-view/ifc4-dtv-objectives, (accessed August, 2015).

[13] J. Steel, R. Drogemuller, B. Toth, Model interoperability in building information modelling, Software & Systems Modeling 11 (2012) 99–109.

[14] N. Nawari, BIM Standardization and Wood Structures, in: Computing in Civil Engineering, ASCE, pp. 293–300.

[15] P. Sanguinetti, S. Abdelmohsen, J. Lee, J. Lee, H. Sheward, C. Eastman, General system architecture for BIM: An integrated approach for design and analysis, Advanced Engineering Informatics 26 (2012) 317–333.
[16] W. Jeong, J. B. Kim, M. J. Clayton, J. S. Haberl, W. Yan, Translating Building Information Modeling to Building Energy Modeling Using Model View Definition, The Scientific World Journal 2014 (2014).
[17] H. Chi, X. Wang, Y. Jiao, BIM-Enabled Structural Design: Impacts and Future Developments in Structural Modelling, Analysis and Optimisation Processes, Archives of Computational Methods in Engineering 22 (2015) 135–151.
[18] P. Chen, L. Cui, C. Wan, Q. Yang, S. K. Ting, R. L. Tiong, Implementation of IFC-based web server for collaborative building design between architects and structural engineers, Automation in construction 14 (2005) 115–128.
[19] A. Redmond, A. Hore, M. Alshawi, R. West, Exploring how information exchanges can be enhanced through Cloud BIM, Automation in Construction 24 (2012) 175–183.
[20] N. Duttachoudhury, ifcXMLNetwork: A Visual Interface for Exploring and Understanding ifcXML Data, `http://www.cs.ubc.ca/~tmm/courses/547-14/projects/nayantara/report.pdf` (accessed August, 2015).
[21] A. Redmond, Designing a Framework for Exchanging Partial Sets of BIM Information on a Cloud-Based Service, Ph.D. thesis, Dublin Institute of Technology, 2013.
[22] X. Deng, T. Y. P. Chang, X. Liu, Automatic generation of structural model from IFC-based architectural model, Tumu Gongcheng Xuebao(China Civil Engineering Journal) 40 (2007) 6–12.
[23] L. Qin, X. Deng, X. Liu, Industry foundation classes based integration of architectural design and structural analysis, Journal of Shanghai Jiaotong University (Science) 16 (2011) 83–90.
[24] Z. Liu, Y. Li, H. Zhang, IFC-based integration tool for supporting information exchange from architectural model to structural model, Journal of Central South University of Technology 17 (2010) 1344–1350.
[25] L. Zhaoqiu, L. Yungui, L. Huipeng, Q. Hu, Information Integration and Collaborative Computation for Complex Structural Model, Industrial Construction 10 (2012) 175–179.
[26] Z. Hu, J. Zhang, BIM- and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 2. Development and site trials, Automation in Construction 20 (2011) 167–180.
[27] X. Wang, Z. P. Cui, Q. L. Zhang, H. Z. Yang, Creating Structural Analysis Model from IFC-Based Structural Model, Advanced Materials Research 712 (2013) 901–904.
[28] X. Wang, J. Hu, H. Yang, Q. Zhang, Transformation Technique of Structural Models Based on Industry Foundation Classes Standard, Tongji Daxue Xuebao/Journal of Tongji University 42 (2014) 836–843.
[29] C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors, John Wiley & Sons, 2011.
[30] H. Wang, Z. Hu, J. Lin, J. Zhang, Web-oriented BIM 3D Viewing and Information Management, Journal of Information Technology in Civil Engineering and Architecture 3 (2013) 1–7.
[31] G. Gesquière, A. Manin, 3D Visualization of Urban Data Based on CityGML with WebGL, International Journal of 3-D Information Modeling 1 (2012) 1–15.
[32] J. Pettit, J. C. Marioni, bioWeb3D: an online webGL 3D data visualisation tool, BMC bioinformatics 14 (2013) 185.
[33] E. Rasys, M. Hodds, N. Dawood, M. Kassem, A Web3D Enabled Information Integration Framework for Facility Management, Australasian Journal of Construction Economics and Building Conference Series 2 (2014) 1–12.
[34] O. Jr, Sixto, Is 3D Finally Ready for the Web?, Computer 43 (2010) 14–16.
[35] J. D. Foley, Computer Graphics: Principles and Practice, Addison-Wesley Professional, 1996.
[36] I. Stroud, Boundary Representation Modelling Techniques, Springer Science & Business Media, 2006.
[37] Requicha, A. G., Representations for Rigid Solids: Theory, Methods, and Systems, ACM Computing Surveys 12 (1980) 437–464.
[38] A. A. Requicha, Representations of rigid solid objects, Springer, 1980.
[39] Z. Hu, J. Zhang, Z. Deng, Construction Process Simulation and Safety Analysis Based on Building Information Model and 4D Technology, Tsinghua Science & Technology 13 (2008) 266–272.
[40] C. Marrin, WebGL Specification, `https://www.khronos.org/registry/webgl/specs/latest/1.0`, (accessed August, 2015).

[41] R. Cabello, Three.js, http://threejs.org, (accessed August, 2015).
[42] A. Evans, M. Romeo, A. Bahrehmand, J. Agenjo, J. Blat, 3D graphics on the web: A survey, Computers & Graphics 41 (2014) 43–61.
[43] M. Segal, K. Akeley, The OpenGL Graphics System: A Specification Version 4.5 (Core Profile), https://www.opengl.org/registry/doc/glspec45.core.pdf, (accessed August, 2015).
[44] D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), https://tools.ietf.org/html/rfc4627, (accessed August, 2015).
[45] D. Crockford, Introducing json, http://json.org, (accessed August, 2015).
[46] E. J. Leavline, D. A. A. G. Singh, Hardware implementation of LZMA data compression algorithm, International Journal of Applied Information Systems 5 (2013) 51–56.
[47] S. Fang, Y. Li, H. Gang, The Summary of Text Compression Technology, Industrial Engineering Journal 5 (2002) 15–18.
[48] J. Gailly, M. Adler, The gzip home page, https://www.gzip.org, (accessed August, 2015).
[49] N. B. T. Ferreira, A WebGL application based on BIM IFC, Ph.D. thesis, Universidade Fernando Pessoa, 2012.
[50] R. Srinivasan, Speed Web delivery with HTTP compression, https://www.ibm.com/developerworks/web/library/wa-httpcomp/wa-httpcomp-pdf.pdf, (accessed August, 2015).

Figure1

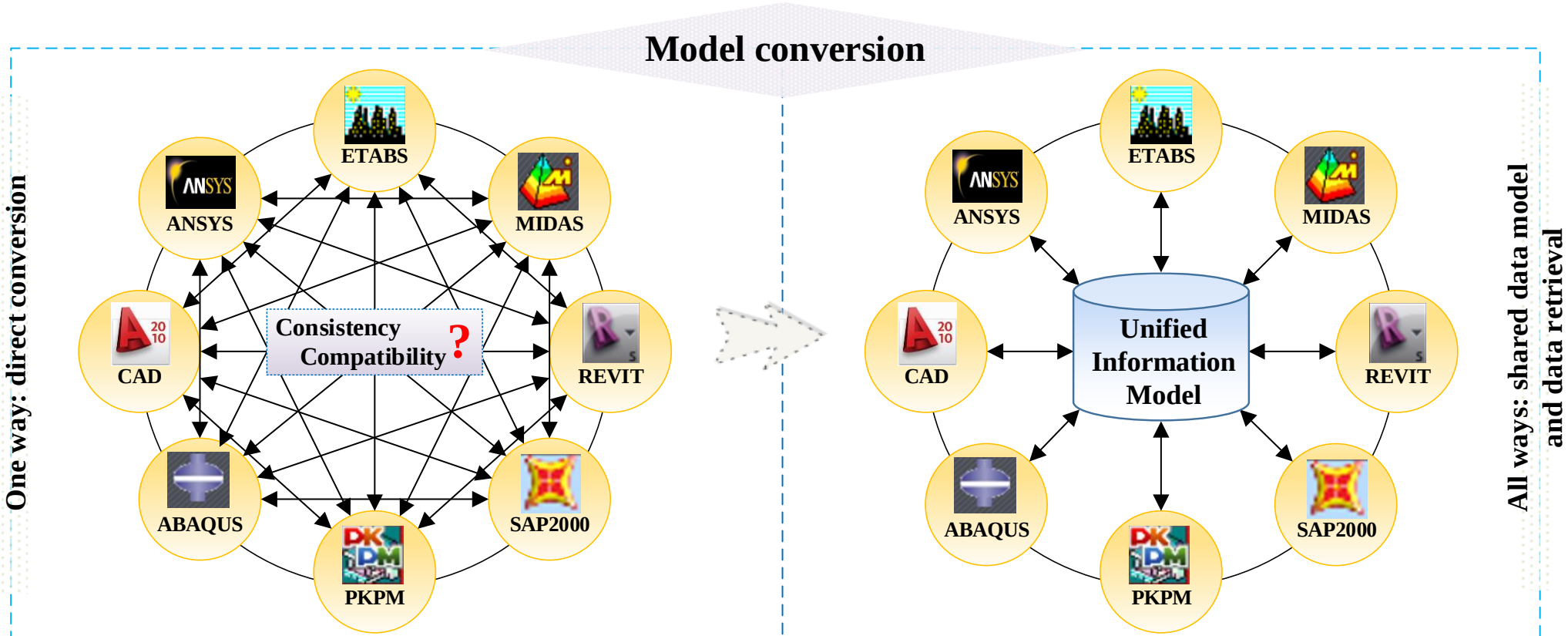
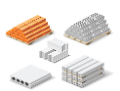

Model conversion
One way: direct conversion
All ways: shared data model and data retrieval
ETABS
ANSYS
MIDAS
CAD
REVIT
ABAQUS
SAP2000
PKPM
Consistency Compatibility ?
Unified Information Model

Figure2

	Information expression			
File type	ETABS (*.e2k) SAP2000 (*.s2k) MIDAS (*.mgt) ANSYS (*.mac) IFC (*.ifc)			
Geometry 	ETABS	POINT “pt name” {x} {y}	LINE “line name” [COLUMN / BEAM / BRACE] “pt1” “pt2” [1/0] [1/0]	
		AREA “area name” [FLOOR/PANEL] {number} “pt1” “pt2” “pt3” “pt4”…… [1/0] [1/0] [1/0] [1/0] …		
	SAP2000	Joint={pt name} CoordSys=GLOBAL CoordType=[Cartesian/Cylindrical] XorR={value} Y={value} Z={value}		
		Frame={line name} JointI={pt1} JointJ={pt2} IsCurved=[Yes/No] Length={value}		
		Area={area name} NumJoints={number} Joint1={pt1} Joint2={pt2}… Perimeter={value} AreaArea={value}		
	MIDAS	{pt name}, {X}, {Y}, {Z}	{line name}, “TYPE”, {mat number}, {sec number }, {pt1}, {pt2}, {angle}	
		{area name}, “TYPE”, {mat number }, {sec number }, {pt1}, {pt2}, {pt3}…, { [1/2], 1-thick，2-thin}		
	ANSYS	K, {point number}, {X}, {Y}, {Z}	LSTR, P1, P2	A, P1, P2, P3, P4… V, P1, P2, P3, P4, …
	IFC	IfcProduct—ObjectPlacement: spatial location information + Representation: geometric shape information		
Material 	ETABS	MATERIAL “mat name” M {mass} W {weight} TYPE [“ISOTROPIC” / “ORTHOTROPZC”] E {e} U {u} A {a}		
	SAP2000	Material={mat name} Type=[Concrete/Steel/…] SymType=[Isotropic/Orthotropic /…] TempDepend=[Yes/No]		
	MIDAS	{mat number}, {Type[Concrete/Steel/…]}, <Data>		
	ANSYS	MP, [ex/alpx/prxy/gxy/dens/……], [material number], C0, C1, C2, C3, C4…{value}		
	IFC	IfcMaterialProperties		
Section 	ETABS	FRAMESECTION “sec name” MATERIAL “mat name” SHAPE “type” {parameters}		
		SHELLPROP “sec name” MATERIAL “mat name” PROPTYPE [“WALL”/“SLAB”…] TYPE [“SHELL”/“PLATE”] {thickness}		
	SAP2000	SectionName={sec name} Material={mat name} Shape=[Rectangular/Circle…] {parameters}		
		Section={sec name} Material={mat name} MatAngle={value} AreaType=[Shell/Plane/Asolid] Thickness={value}		
	MIDAS	{sec number}, {TYPE}, {shape name}, <OFFSET>, {SHAPE}, <DATA>		
	ANSYS	SECTYPE, SECID, Type[beam//joint/shell/…], Subtype[SECDATA/SECOFFSET], Name		
	IFC	IfcProfileProperties— IfcProfileProperties + IfcRibPlateProfileProperties		
Load / Restraint / Other information ……				

ABC —PropertyName
{Value} —PropertyValue
[A/B/…] —OptionValue

Figure3

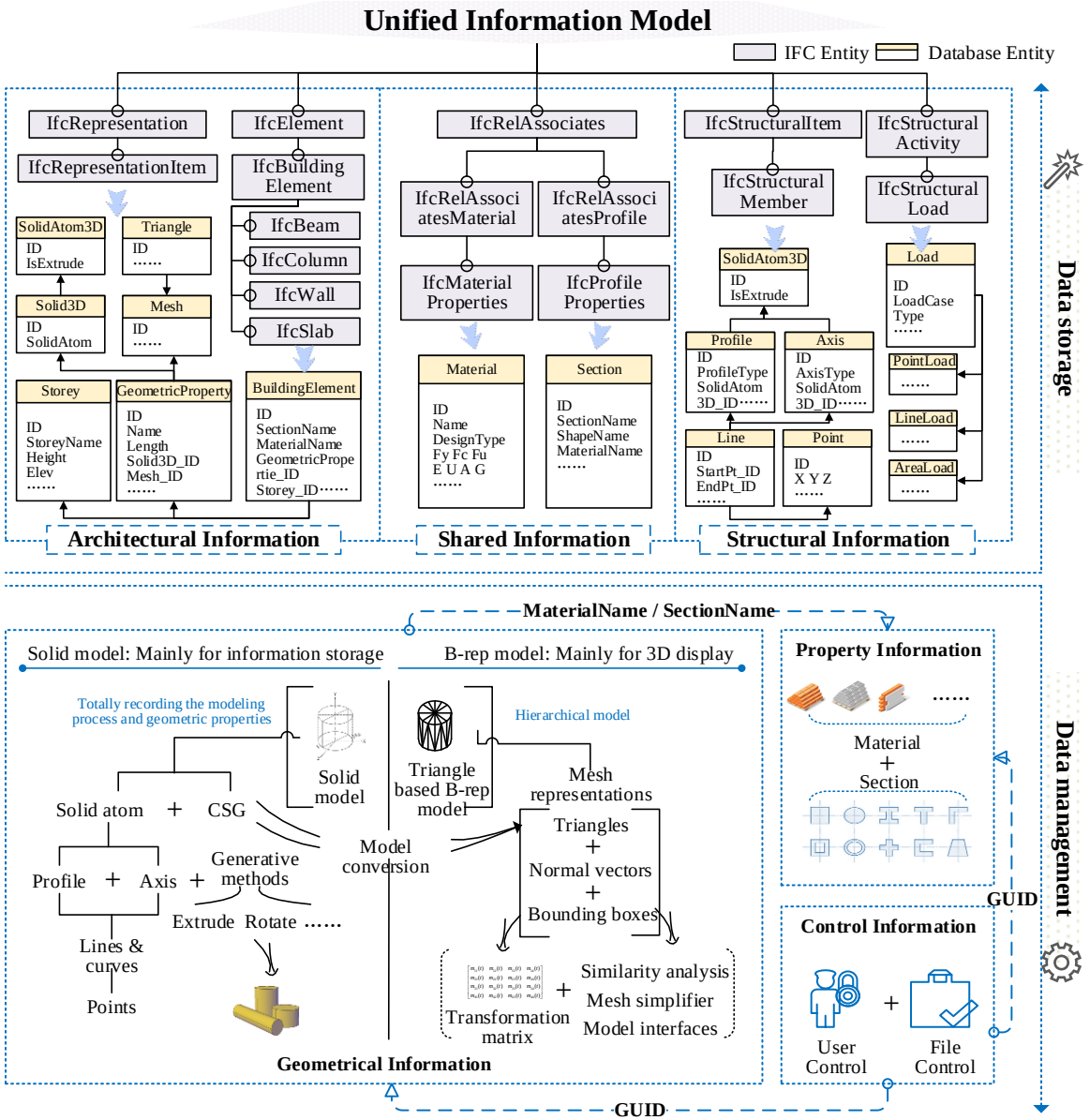

Unified Information Model
IFC Entity
Database Entity
IfcRepresentation
IfcRepresentationItem
SolidAtom3D
ID
IsExtrude
Triangle
ID
……
Solid3D
ID
SolidAtom
Mesh
ID
……
IfcElement
IfcBuilding Element
IfcBeam
IfcColumn
IfcWall
IfcSlab
Storey
ID
StoreyName
Height
Elev
……
GeometricProperty
ID
Name
Length
Solid3D_ID
Mesh_ID
……
BuildingElement
ID
SectionName
MaterialName
GeometricPropertie_ID
Storey_ID……
Architectural Information
IfcRelAssociates
IfcRelAssociatesMaterial
IfcRelAssociatesProfile
IfcMaterial Properties
IfcProfile Properties
Material
ID
Name
DesignType
Fy Fc Fu
E U A G
……
Section
ID
SectionName
ShapeName
MaterialName
……
Shared Information
IfcStructuralItem
IfcStructural Activity
IfcStructural Member
IfcStructural Load
SolidAtom3D
ID
IsExtrude
Load
ID
LoadCase
Type
……
Profile
ID
ProfileType
SolidAtom
3D_ID……
Axis
ID
AxisType
SolidAtom
3D_ID……
PointLoad
……
LineLoad
……
AreaLoad
……
Line
ID
StartPt_ID
EndPt_ID
……
Point
ID
X Y Z
……
Structural Information
Data storage
MaterialName / SectionName
Solid model: Mainly for information storage
B-rep model: Mainly for 3D display
Totally recording the modeling process and geometric properties
Solid model
Solid atom + CSG
Profile + Axis + Generative methods
Extrude Rotate ……
Lines & curves
Points
Model conversion
Triangle based B-rep model
Hierarchical model
Mesh representations
Triangles + Normal vectors + Bounding boxes
Transformation matrix
Similarity analysis
Mesh simplifier
Model interfaces
Geometrical Information
Property Information
Material + Section
GUID
Control Information
User Control
File Control
GUID
Data management

Figure4

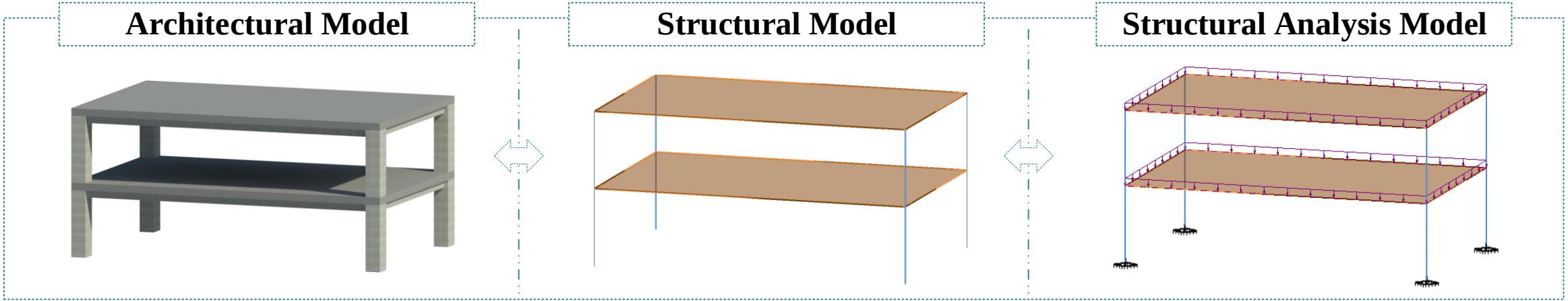

Architectural Model
Structural Model
Structural Analysis Model

Figure5

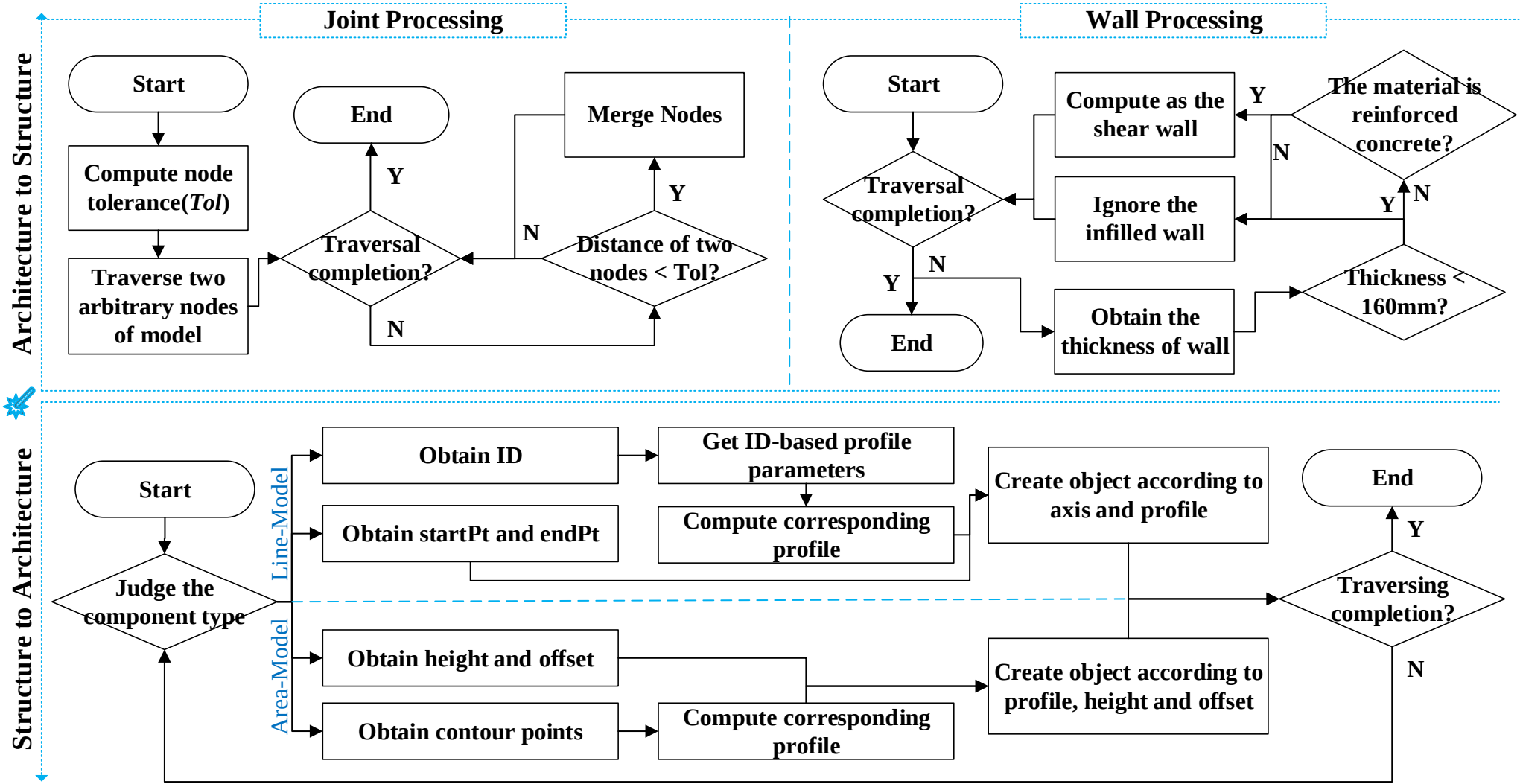

Architecture to Structure
Joint Processing
Start
Compute node tolerance(*Tol*)
Traverse two arbitrary nodes of model
Traversal completion?
Y
N
End
Distance of two nodes < Tol?
Y
N
Merge Nodes
Wall Processing
Start
Traversal completion?
Y
N
End
Compute as the shear wall
Ignore the infilled wall
Y
N
The material is reinforced concrete?
Y
N
Obtain the thickness of wall
Thickness < 160mm?
Structure to Architecture
Start
Judge the component type
Line-Model
Obtain ID
Get ID-based profile parameters
Obtain startPt and endPt
Compute corresponding profile
Area-Model
Obtain height and offset
Obtain contour points
Compute corresponding profile
Create object according to axis and profile
Create object according to profile, height and offset
Traversing completion?
Y
N
End

Figure6

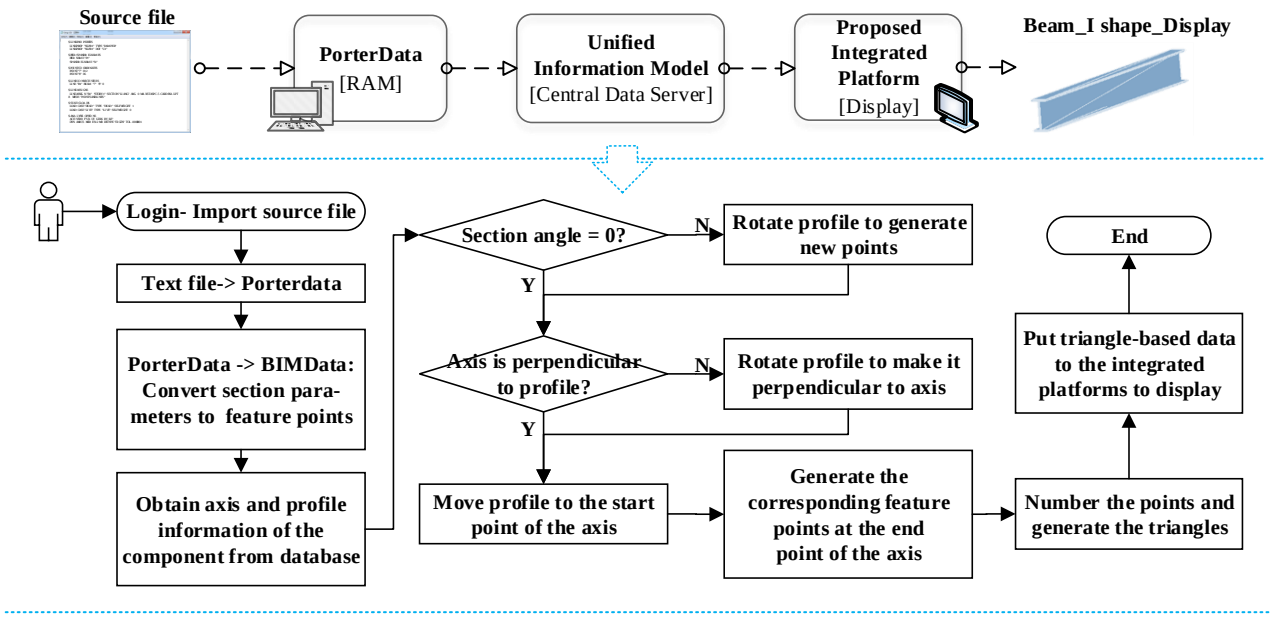
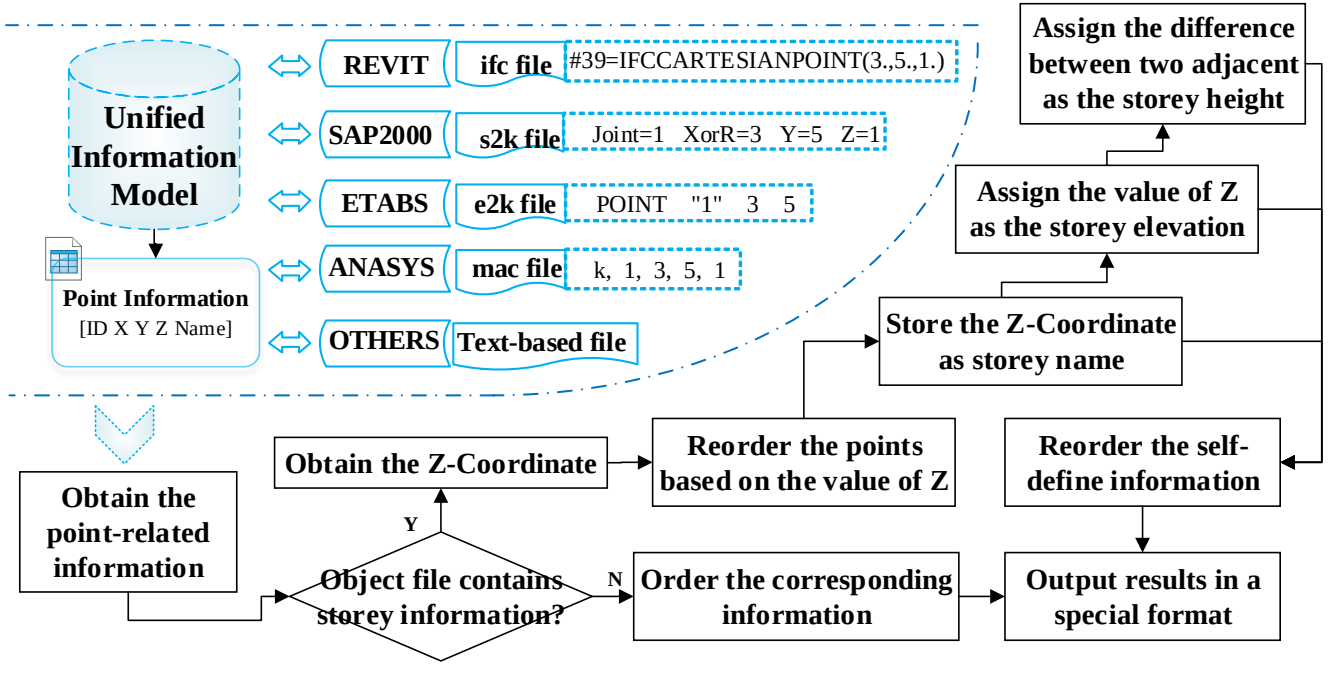

Source file
PorterData
[RAM]
Unified
Information Model
[Central Data Server]
Proposed
Integrated
Platform
[Display]
Beam_I shape_Display
Login- Import source file
Text file-> Porterdata
PorterData -> BIMData:
Convert section para-
meters to feature points
Obtain axis and profile
information of the
component from database
Section angle = 0?
N
Rotate profile to generate
new points
Y
Axis is perpendicular
to profile?
N
Rotate profile to make it
perpendicular to axis
Y
Move profile to the start
point of the axis
Generate the
corresponding feature
points at the end
point of the axis
Number the points and
generate the triangles
Put triangle-based data
to the integrated
platforms to display
End

Figure7


Unified Information Model
REVIT
ifc file
#39=IFCCARTESIANPOINT(3.,5.,1.)
SAP2000
s2k file
Joint=1 XorR=3 Y=5 Z=1
ETABS
e2k file
POINT "1" 3 5
ANASYS
mac file
k, 1, 3, 5, 1
OTHERS
Text-based file
Point Information
[ID X Y Z Name]
Obtain the point-related information
Obtain the Z-Coordinate
Object file contains storey information?
Y
N
Reorder the points based on the value of Z
Store the Z-Coordinate as storey name
Assign the value of Z as the storey elevation
Assign the difference between two adjacent as the storey height
Order the corresponding information
Reorder the self-define information
Output results in a special format

Data conversion

Source file

Importer: Source file -> PorterData -> RAM

Data interface encapsulation

Exporter: RAM -> PorterData -> Object file

Object file

ETAB (*.e2k)

ANSYS (*.mac)

IFC (*.ifc)

SAP2000 (*.s2k)

MIDAS (*.mgt)

PorterData [RAM]

Unified Information Model [Central Data Server]

PorterData [RAM]

ETABS (*.e2k)

ANSYS (*.mac)

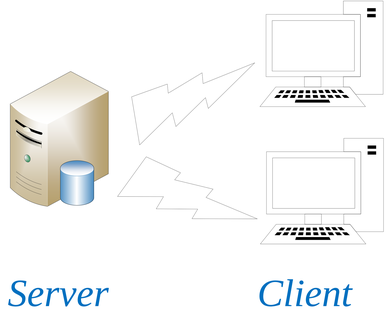
IFC (*.ifc)

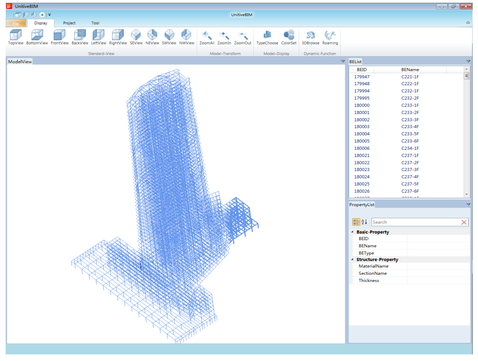
SAP2000 (*.s2k)

MIDAS (*.mgt)

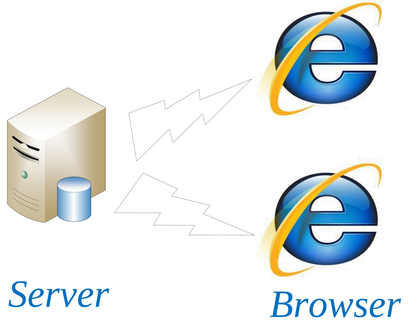
Extracted necessary data from unified information model to display

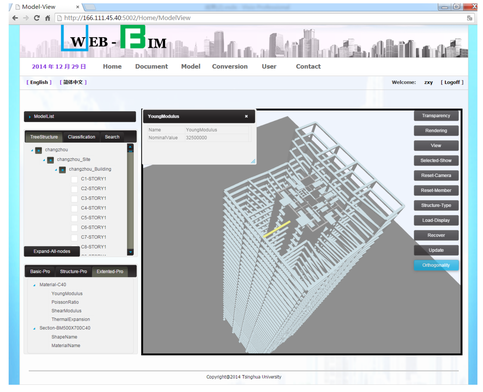
Model display





Generate GraphicModel to OpenGL-based platform
(C/S platform to display)





Generate JsonData to WebGL-based platform
(B/S platform to display)

Figure9

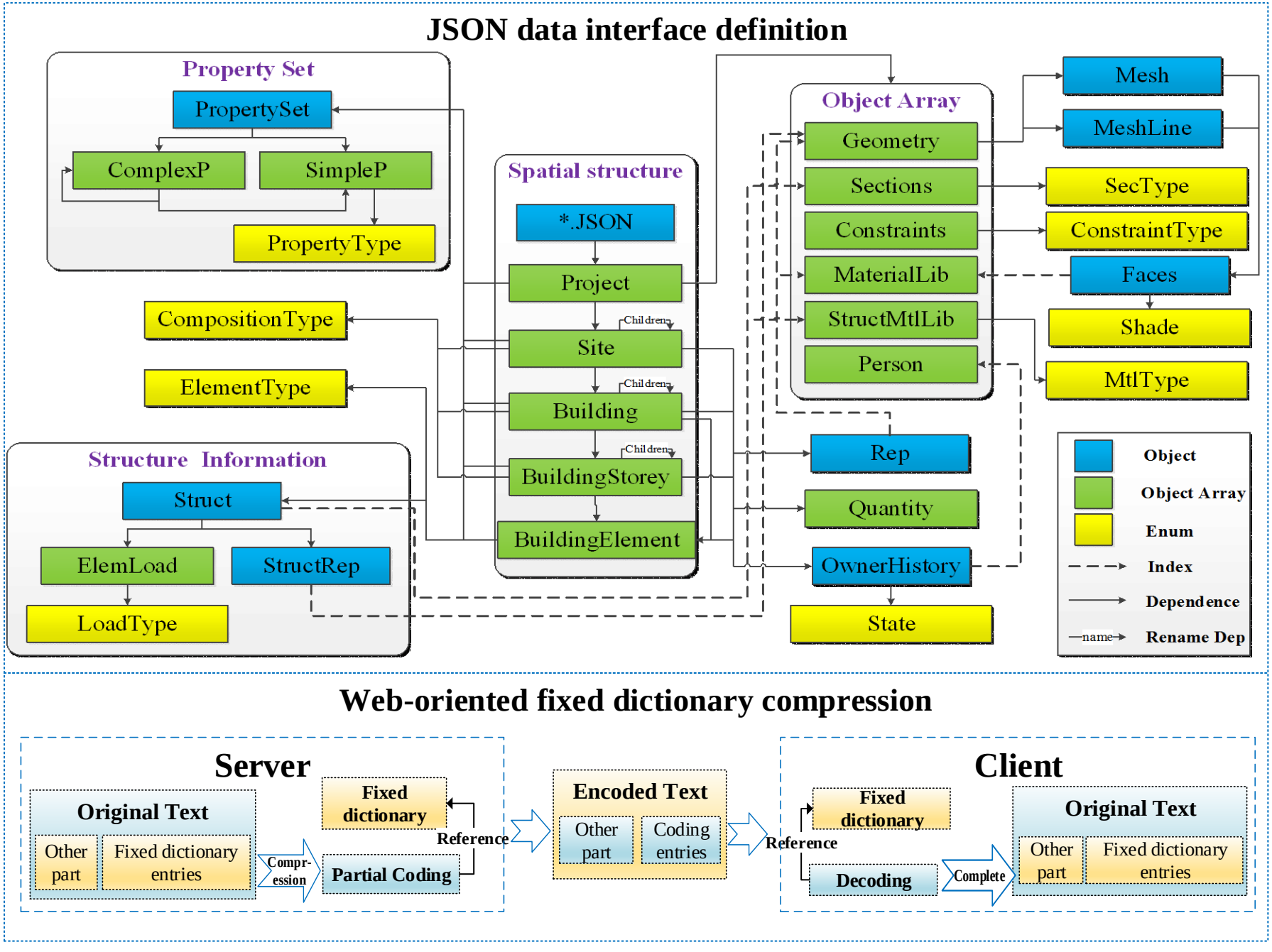
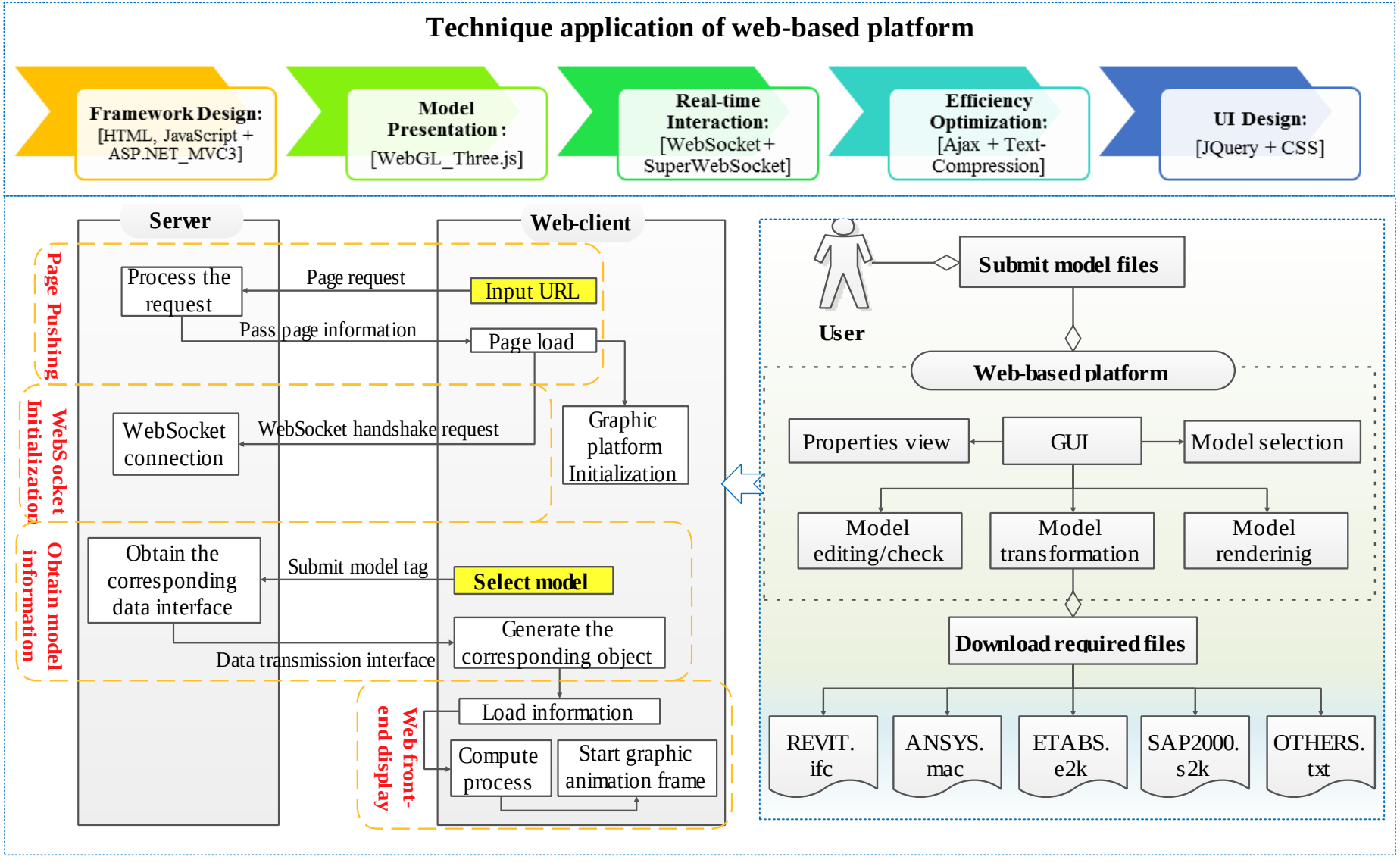
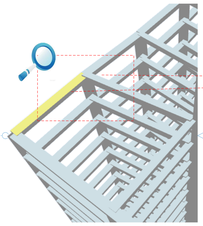

JSON data interface definition
Property Set
PropertySet
ComplexP
SimpleP
PropertyType
CompositionType
ElementType
Structure Information
Struct
ElemLoad
StructRep
LoadType
Spatial structure
*.JSON
Project
Children
Site
Children
Building
Children
BuildingStorey
BuildingElement
Object Array
Geometry
Sections
Constraints
MaterialLib
StructMtlLib
Person
Mesh
MeshLine
SecType
ConstraintType
Faces
Shade
MtlType
Rep
Quantity
OwnerHistory
State
Object
Object Array
Enum
Index
Dependence
name
Rename Dep
Web-oriented fixed dictionary compression
Server
Original Text
Other part
Fixed dictionary entries
Compr-ession
Fixed dictionary
Partial Coding
Reference
Encoded Text
Other part
Coding entries
Reference
Fixed dictionary
Decoding
Complete
Client
Original Text
Other part
Fixed dictionary entries

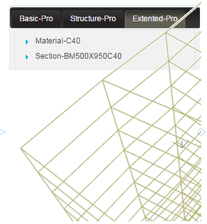
Figure10


Technique application of web-based platform
Framework Design: [HTML, JavaScript + ASP.NET_MVC3]
Model Presentation: [WebGL_Three.js]
Real-time Interaction: [WebSocket+ SuperWebSocket]
Efficiency Optimization: [Ajax + Text-Compression]
UI Design: [JQuery + CSS]
Server
Web-client
Page Pushing
Process the request
Page request
Input URL
Pass page information
Page load
WebSocket Initialization
WebSocket connection
WebSocket handshake request
Graphic platform Initialization
Obtain model information
Obtain the corresponding data interface
Submit model tag
Select model
Generate the corresponding object
Data transmission interface
Web front-end display
Load information
Compute process
Start graphic animation frame
User
Submit model files
Web-based platform
Properties view
GUI
Model selection
Model editing/check
Model transformation
Model renderinig
Download required files
REVIT. ifc
ANSYS. mac
ETABS. e2k
SAP2000. s2k
OTHERS. txt

Extract the topology information of structural element

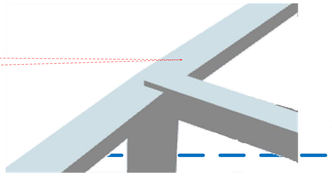


[Architectural Model]

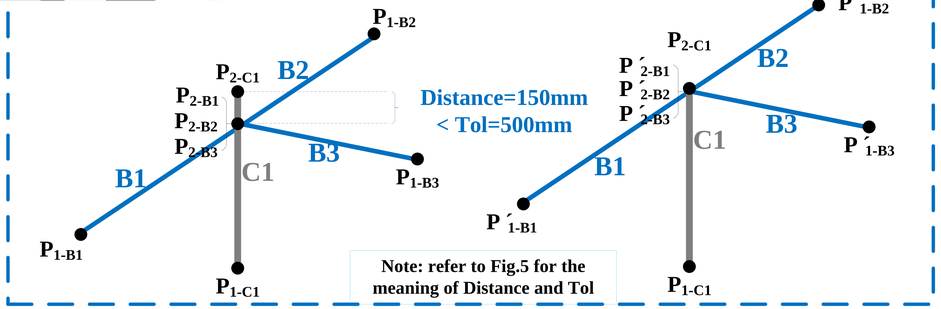


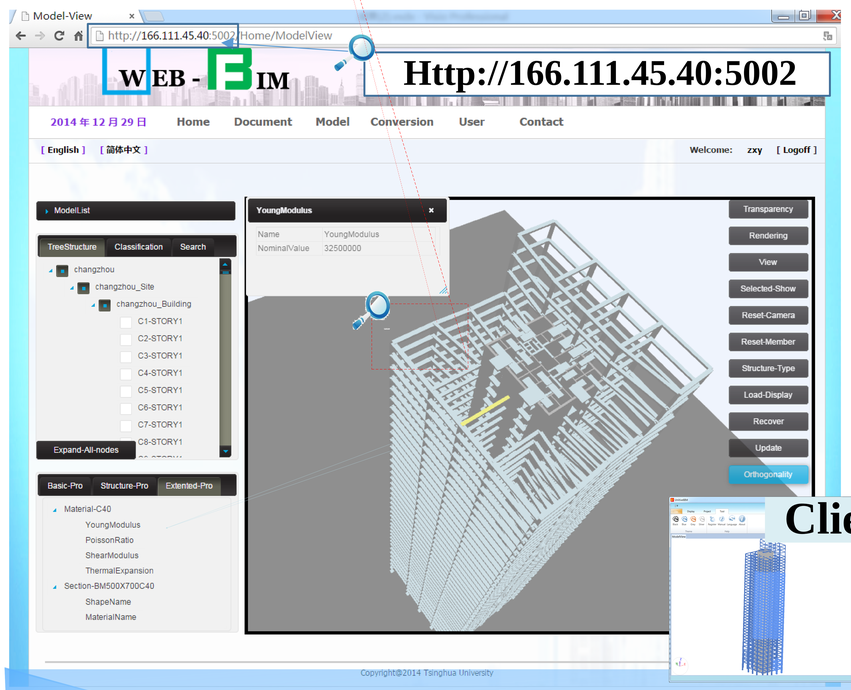
[Structural Model]

Extract key information (Node, Axis, Material, Section……)



[Joint Processing]





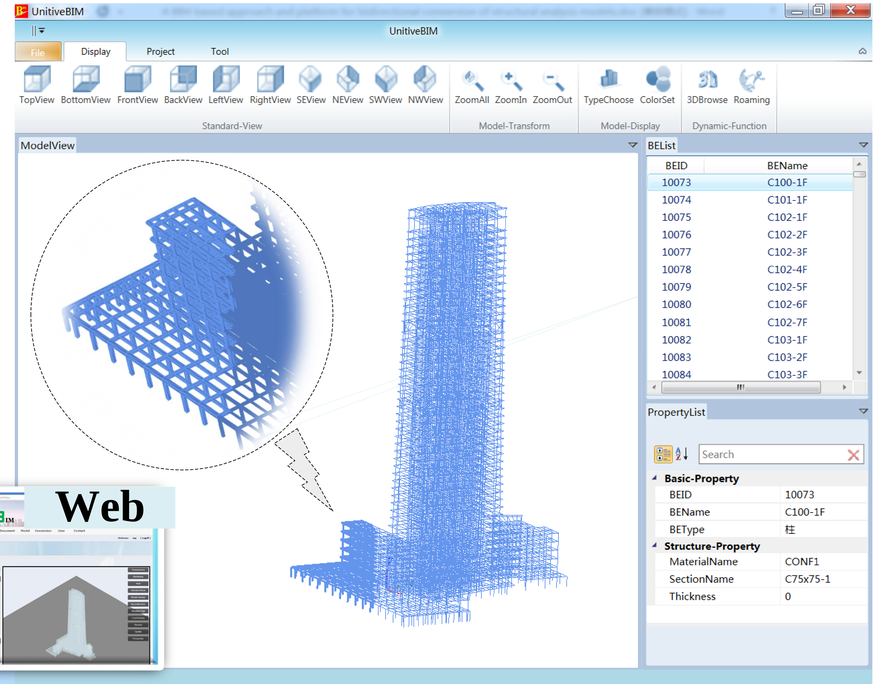
[The 42-storey building]

Frame-shear wall structure

Web-BIM & B/S

Unitive-BIM C/S

CopyRight @2014 Tsinghua University



Web

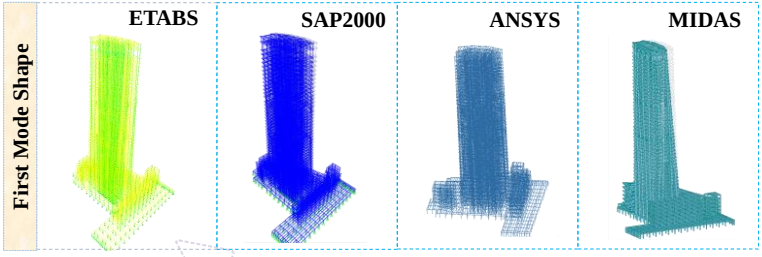
[The super high-rise business plaza]

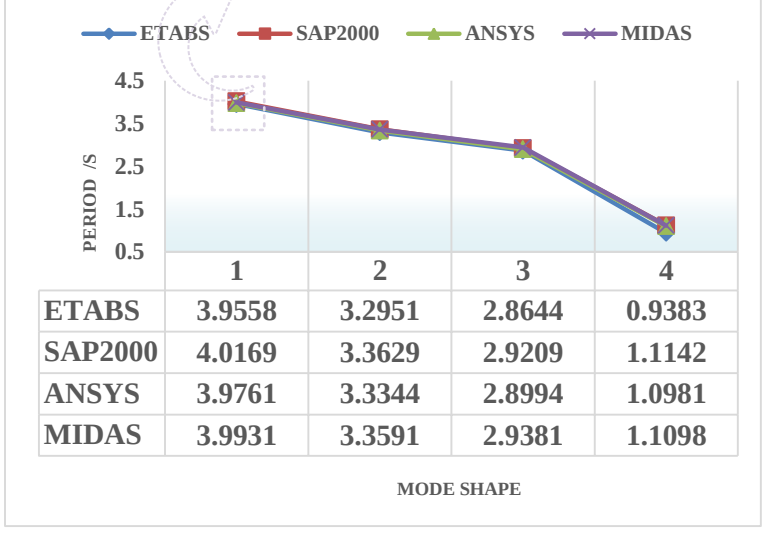
Frame structure

Original Data	Compressed Data	Ratio
39922KB	2977KB	7.46%
16114KB	908KB	5.63%

Figure12

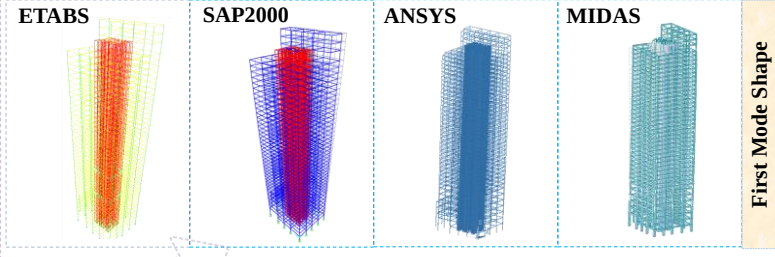
The reinforced concrete frame structure

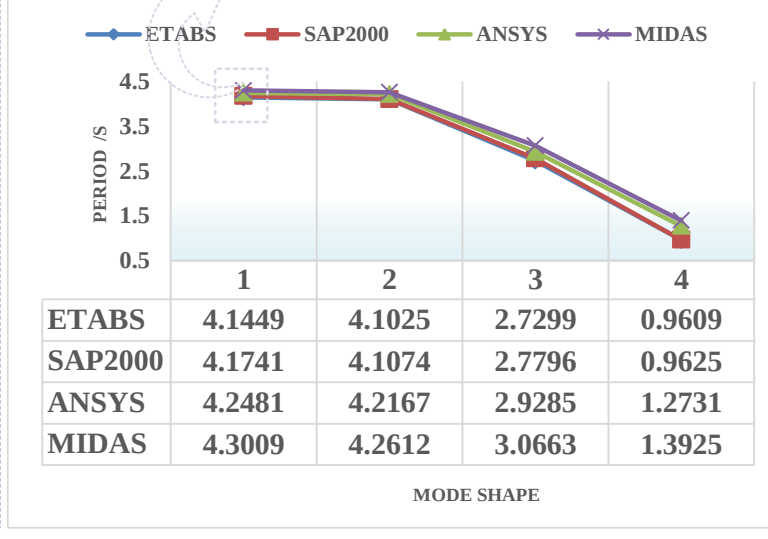




	1	2	3	4
ETABS	3.9558	3.2951	2.8644	0.9383
SAP2000	4.0169	3.3629	2.9209	1.1142
ANSYS	3.9761	3.3344	2.8994	1.0981
MIDAS	3.9931	3.3591	2.9381	1.1098

The frame-shear wall structure





	1	2	3	4
ETABS	4.1449	4.1025	2.7299	0.9609
SAP2000	4.1741	4.1074	2.7796	0.9625
ANSYS	4.2481	4.2167	2.9285	1.2731
MIDAS	4.3009	4.2612	3.0663	1.3925

Figure13

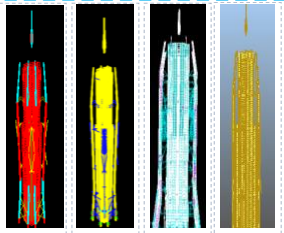
Ping An IFC

General information

Antenna spire：600 m

Floor count：115 plus 5

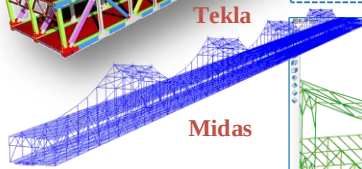
Floor area：385,918 m^2



ETABS SAP2000 ANSYS MIDAS

Jinan Yellow River Bridge

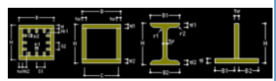
Tekla



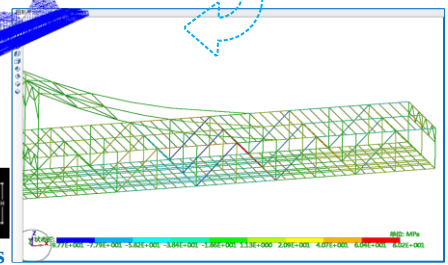
Midas

	Axial Force (N	Shear-y (N	Shear-z (N	Bending Moment (N	Bending Moment
[2934]	-4.91E+01	3.09E-01	2.33E+00	-5.80E+00	5.80E+00
4114 I[2935]				00	1.94E+00
8415 I[2936]				01	6.05E-01
8416 I[2937]				01	1.54E-01
8417 I[2938]	-2.18E+01	2.38E-02	2.06E+00	2.92E-01	-2.92E-01
8418 I[2939]	-2.20E+01	8.76E-03	9.13E-01	5.86E-02	-5.86E-02

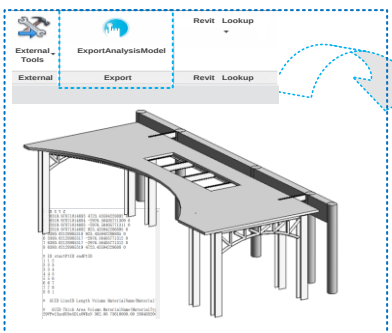
Element Stress



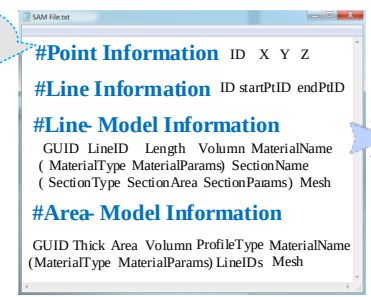
5-Materials 67-Sections

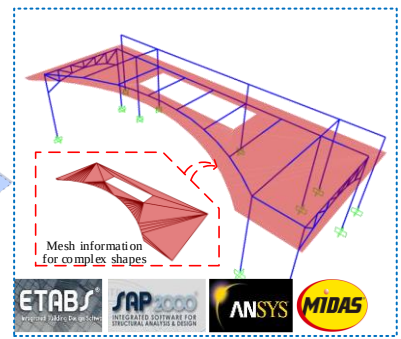


REAL PROJECTS



SAM File Format





REVIT API

LaTeX Source Files

Click here to download LaTeX Source Files: LaTeX source files-Final.zip