



Assessment of model-based data exchange between architectural design and structural analysis

Goran Sibenik^{*}, Iva Kovacic

Institute for Interdisciplinary Building Process Management, TU Wien, Karlsplatz 13, 1040, Vienna, Austria

ARTICLE INFO

Keywords:

Data exchange
Industry foundation classes
IFC
Architectural model
Structural analysis model
Building information modeling
BIM

ABSTRACT

The emerging availability of numerous model-based design and analysis tools promises high interoperability in terms of data exchange, thus shorter planning cycles and higher design quality. However, seamless transfer of digital models across domains without significant data inconsistencies is still a major challenge in the practice. This paper aims to identify the procedural problems in the data transfer between architectural design and structural analysis and propose improvement.

This study reviewed the current data exchange practices, and concluded that data exchange based on the industry foundation classes (IFC) is the most spread in the research and software tools supporting architecture, engineering and construction industry. The study also analyzed the processes which affect the IFC-based data exchange. The interoperability of software tools through IFC files was tested in a comparative study. The testing results are reflected on the processes influencing data exchange, identifying the procedural problems and recommending the improvements.

Upon conducted analysis and testing, the authors propose the following strategies for the systematic improvement of the data exchange: (a) introducing interpretation rules in the data exchange standards; (b) focusing on multiple domain-specific building data schemas instead of on the integrated schema; and (c) developing a new certification process based on the domain-specific building data schema. The improvement proposal supported by the analysis provides a roadmap for the development of model-based exchange between architectural design and structural analysis.

1. Introduction

Digital data exchange between architectural design and structural analysis, when both domains use BIM authoring software tools, is still burdened with numerous difficulties, mostly caused by semantical problems and particularly challenging problems of geometric interpretation [1,2]. Seamless data transfer is essential for BIM-based collaboration; therefore the improvement of interoperability is a prerequisite to increase BIM usefulness and usability.

Within the scope of this research, BIM is regarded as a joint digital knowledge base supporting the activities of all stakeholders in AEC. It is based on various data sets, including geometrical and/or non-geometrical information; allowing data generation, exchange and processing within the life cycle of built structures. The full realization of BIM includes domain-specific modeling during the design phase as well as model-based information exchange which, when used together, demonstrate the full potential of BIM.

AEC industry stakeholders benefit in various ways from BIM workflows. The McGraw Hill Survey [3] identifies the reduction of errors and omissions as the most common benefit of BIM workflow, resulting in reductions of costs, duration and the amount of rework [4]. Identified that the main benefit of using BIM models in the design phase is the earlier collaboration of multiple domains, leading to a reduction of errors and omissions, and shortening the design cycle, resulting in a more cost-effective design.

There is a bulk of research involving technical improvements [2] or testing [5] of data exchange between architectural design and structural analysis. Research dealing with the data-exchange analysis on inter-organizational level addresses general processes rather than solutions for a particular domain-specific data exchange [6]. This paper adds to the existing body of knowledge through an assessment of domain-specific data exchange between architectural design and structural analysis models on inter-organizational level, in which next to the end-user processes, also normative, regulative and software

^{*} Corresponding author.

E-mail addresses: goran.sibenik@tuwien.ac.at (G. Sibenik), iva.kovacic@tuwien.ac.at (I. Kovacic).

development processes are included.

The aims of this research are to review the state-of-the-art of data exchange between architectural design and structural analysis, to identify the main problems via a comparative study, and to develop an improvement strategies proposal for this domain-specific data exchange.

The main question this paper aims to answer is: **Why does model-based data exchange between architectural design and structural analysis still not function without significant data and information losses?**

This paper aims to address the problems and issues of still flawed model-based data exchange between architectural design and structural analysis by:

- a) Exploring the model-based data exchange practices between architectural design and structural analysis;
- b) Exploring the procedures influencing the open IFC-based data exchange;
- c) Identifying the origins of data losses and misinterpretations in the IFC-based data exchange between architectural design and structural analysis;
- d) Identifying the necessary actions leading to a systematic improvement of the IFC-based exchange between architectural design and structural analysis.

In order to answer the research question and propose improvement strategies for the data exchange between architectural design and structural analysis domains, we will:

- 1) Assess the state of the research of data exchange between architectural design and structural analysis;
- 2) Analyze the state-of-the-art of data exchange in AEC practice;
- 3) Review and identify processes influencing the IFC-based data exchange using literature and workflow analysis;
- 4) Conduct a comparative study to identify the inconsistencies in the IFC-based data exchange between architectural design and structural domain models;
- 5) Link the detected inconsistencies to the procedure and propose strategies for systematic improvements of inter-domain data exchange.

This paper is structured as follows: Section 2 presents the problem statement and discusses current issues; Section 3 reviews the state of research regarding the data exchange between architectural design and structural analysis domains; current data exchange practices, including the IFC-based open data exchange, are presented in Section 4; the comparative study and results are presented in Section 5; Section 6 discusses the systematic procedural improvement strategies for the IFC-based data exchange and proposes an improvement roadmap; finally, Section 7 concludes the findings, limitations of the research and addresses the need for future research.

2. Problem statement

Building projects are unique by nature; the stakeholders and software tools vary from project to project, creating numerous planning workflows. A building model changes during the design phase and differs between the domains. The design phases and domains depend on the project-specific workflow. In this research, we analyze one of the most common exchanges – architectural design to structural analysis – that are part of the core design team and which is especially important in the developed design phase [7]. Although this research focuses on the design phase before a building permit is granted, similar data exchange practices between architectural and structural analysis models can be found during the whole building life cycle. During the construction, renovation or demolition phase, a structural model coupled with information such as construction schedule and activities, also called 4D

structural information model, is needed for the time-dependent structural analysis [8], which is of particulate importance for structural safety management. The generation of such models from architectural models should be enabled at any user-specified time point, for which again a precondition is seamless exchange from architectural design to structural analysis.

The architectural design and structural analysis domains require two types of building models: physical model for architectural design and discretized model for further numerical simulation. Data exchange has traditionally been conducted by redefining information provided by the architects in a structural analysis domain-specific software tool. The more advanced model-based data exchange practices use digitally transferred building models. However, due to lacking interoperability or coordinated modelling conventions, a model-based data transfer leads to significant remodeling efforts, prone to errors and misinterpretations [9, 10].

Successful data transfers are based on specific modeling instructions, however often requiring various software-dependent workarounds [1]. Those workflows are commonly found within the “in-the-house” practices, or if a third party (e.g. project coordinator or BIM manager) defines the modeling and exchange workflows. Positive experiences also depend on the software tool combinations, as will be shown in Section 5, on interoperability testing. The focus of achieving a successful interoperability needs to be extended from solely technological issues towards the business processes, culture and values and management of contractual issues [11].

Numerous attempts to achieve satisfactory data exchange between architectural design and structural analysis can be found in the research or industrial practice related literature. However, these have not significantly improved the performance of data exchange in practice, and interoperability in the industry continues to improve very slowly [12]. To date, proposals have not been able to cope with the heterogeneity of the AEC industry.

This research aims to tackle the problems that occur within the data exchange process, while taking into account the reported findings from the current literature and research. The identification of errors in data transfer is a topic that is often explored in the literature. A major innovative contribution of this paper is the linking of these errors to the procedures on the inter-organizational level, from the perspectives of end users, software developers and normative and regulative instances.

The processes influencing the open IFC-based data exchange between the architectural design and structural analysis domains on the inter-organizational level are identified and analyzed based on the simulated end-user experience on the project level, which allows for more practice and end-user focused improvements.

The IFC-based data exchange between architectural design and structural analysis is examined on an inter-organizational level within the AEC ecosystem [13,14] in order to grasp the procedural problems. For instance, software development or software certification can be identified as the inter-organizational procedures strongly influencing the data exchange workflows.

Further, current research mostly proposes single and specific frameworks or new regulative or policy making practices (e.g. on the national level), without considering the implications on the existing practices in the industry, omitting to tackle the end-user experience which is dependent on various business processes and cultures and values. The novel contribution of this research paper is the proposal of improvement strategies for the existing processes on an inter-organizational level for the IFC-based data exchange, thereby distancing the improvement strategies from differing practices and cultures on the project level.

3. State of the research

This Section reviews the current research and theoretical academic models for data exchange between the architectural design and

structural analysis domains.

3.1. Data exchange between the architectural design and structural analysis domains

Research within the academic community on the topic of BIM-based data exchange between architectural design and structural analysis mainly involves the use of the open IFC schema. First attempts to improve the IFC-based data exchange involved improvements of the IFC schema. The focus in the work of Weise et al. [15] was to introduce the requirements of the structural engineers into the IFC schema, based on the structural analysis and steel construction requirements. Weise et al. describe a workflow in which the structural engineers require information about the building elements from the architects, but the relationships between the proposed structural elements and the architectural elements were not part of their research. Later, Wan et al. [16] assessed the suitability of the IFC schema for the requirements of SAP2000 and six other structural analysis tools, and suggested several improvements to the IFC schema. In their work, the data exchange workflow was server-based and involved an additional step to transfer IFC building data model to S2K format because the SAP2000 software was not able to read IFC files. Serrero et al. [17] introduce an extension to the IFC schema based on the FEM and CIS2 standards, focusing on earthquake simulation. The current version of the IFC schema is able to support most requirements in their structural analysis representation; however, the relationships between the corresponding domain-specific representations are still not defined within the IFC schema.

Several authors continued to work on improving data exchange from architectural BIM tools to structural analysis tools by proposing third-party tools that redefine exported IFC building models. Deng et al. [18] suggested the conversion of IFC files to an XML-based General Format of Unified Finite Element Model that can be imported to SAP2000, ETABS and, eventually, other structural analysis tools. The workflow suggested by Deng and Chang involved the filtering and adjustment of building element connectivity in order to exchange a useful structural analysis model. Qin et al. [19] further developed Deng and Chang's framework by using XML-based unified FEM as a communication tool between an IFC file holding the architectural model and the structural analysis software tools (ETABS, SAP2000, PKPM and STAAD). Liu et al. [20] created a software-specific integration tool which was able to use the IFC file to create a PMCAD file that can be imported into the PKPM structural analysis tool. Differences between the architectural and structural analysis tools were managed in the proposed third-party software-specific integration tool. Wang et al. [21] showed that the models exported from structural design tools in the form of IFC structural models need to be transformed before they can be used in structural analysis software tools such as MIDAS. Zhang et al. [22] developed an approach using the IFC-based unified information model, which is created from the architectural IFC-based model; thereby proposing the algorithms for four structural analysis tools (SAP2000, ETABS, ANSYS and MIDAS) which can map the prepared data from the unified information model. They propose using WebGL for the representation of geometry and a bidirectional web-based platform for the data exchange. Hu et al. [23] further developed this approach by introducing other case studies on exchanging architectural design and structural analysis models. Oh et al. [9] created an integrated information management system – a platform which can be connected to Revit and can manage multiple domain-specific models to emphasize the need of integrated planning. Ramaji and Memari [24] focused on the interpreted exchange between the two domains, and proposed a framework with the direct and interpreted data exchange units from an IFC file which, in the follow up, is imported to SAP2000 and SCIA. The same authors [2] defined a framework where the interpretations are performed within the IFC coordination view version 2.0 (CV V2.0) and two versions of IFC building data models: an architectural design input and a structural analysis output, with the interpretations similar to their preceding work.

All of the research described above addressed the data exchange problems between architectural and structural domains; identified the considerable differences between the models, and provided various solutions to the problems identified. The problems were addressed on a practical level, with numerous software or project-specific solutions being proposed.

Although most of the contributions listed above propose new frameworks, none of the proposed solutions have been widely accepted in practice, because of the heterogeneity of the AEC industry (particularly in the European market) and varying standards. The most promising proposals involve IFC-based data exchange that is intended to be software independent. However, IFC-based data exchange still depends on the software tool combination (see Section 5), lacks the required confidence of the end users, and keeps the aforementioned frameworks software-specific and dependent on the setup of each individual project.

[25] assessed IFC interoperability over a period of five years. They proposed four interoperability concerns (business, process, service and data), but they investigated interoperability of data only. Their results reveal a slow development of IFC-based interoperability. [5] tested data exchange between architectural design and structural analysis via a comparative study. However, they tested a model originating from Revit only and compared three types of exchange (direct link using native files; direct link using application programming interfaces (APIs); and indirect link using IFC), stating that the indirect link is the most common exchange. They identified the value change and information losses as technical problems of IFC-based data exchange. [26] also tested the interoperability of software tools with the IFC model originating from Revit, identified some technical problems and proposed solutions. Their research focused on IFC performance from the structural analysis perspective, with single building elements imported into various structural tools, but did not include a complete case study reflecting end-user experience.

Various problems, such as lack of motivation for software developers, lack of funding for standards development or insufficient testing procedures, have already been identified and improvements proposed to IFC-based data exchange by the academic community (e.g. [6,12,27]). The improvements proposed are at the level of the AEC industry, and are not focused to a professional domain, domain-specific data-exchange or software tool. The effects of these proposals to the particular exchange between architectural design and structural analysis was not considered by the above-mentioned authors.

3.2. Data exchange testing

This subsection provides a review of the data exchange testing methods, as used to test various software tools and their data exchange properties within the comparative study (presented in Section 5).

Two types of software testing are relevant for data exchange [28]:

- **Conformance testing:** This is performed when a product is compared to a reference system, such as a standard. Software tools are more likely to successfully work together (interoperate) if both conform to the same standard. Conformance, however, does not guarantee interoperability. In the conformance testing, software import and export possibilities were tested against the reference system separately.
- **Interoperability testing:** In software interoperability testing, the behavior of a tool is tested when it is interoperating with another product. Interoperability testing has certain advantages and disadvantages over conformance testing. It indicates problems that are interesting to the user and it can address issues that are not defined in a standard. The number of test cases is substantially bigger because the data exchange is separately tested for each combination of software tools.

In the literature, software tool tests regarding the IFC-based data

exchange were based on both approaches: conformance testing and interoperability testing. In Lee et al. [29], conformance testing was used to validate a single IFC file on IFC schema compliance, its conformity to the IFC subschema (model view definition (MVD)) and design programming requirements. The main part of the IFC certification process for CV V2.0 (IFC2x3) is a conformance test. Software tools have a better chance of working together if they conform to the same standards, but conformance tests do not reflect the end-user experience and software interoperability in practice. A common approach to testing interoperability is by comparing two IFC files [10,30,31]. Multiple methods used for comparing two IFC files are unified in the study that defines metrics for quantifying the similarities and differences between the two IFC files [31]. The workflow described in [31] includes creating an IFC file using architectural BIM software, importing it to another software tool and then generating the second IFC file without making any changes. If software tools are tested using this methodology, the first tool needs to support the export by using the IFC schema, and a second one needs to do this for both the import and export. This requirement reduces the number of available software tools for testing. “Round-trip testing” is a special form of interoperability testing which tests the interoperability of a program with itself. It compares the IFC files imported into a design tool and exported back to IFC without making any changes. Support for round-trip scenarios is excluded from the support of IFC coordination view [32]. However, round-trip testing is specified as a recommendation for self-testing in [33].

4. State-of-the-art in data exchange practices

The following Section describes different data exchange practices; including the IFC-based open exchange, which is the most relevant data exchange practice for this assessment. Further on, the IFC-based data exchange between architectural design and structural analysis is explored in more detail. In addition, the processes influencing the IFC-based data exchange within the AEC ecosystem will be identified and analyzed on the inter-organizational level, from various perspectives – end user, normative and regulative instances, and software developer – via a literature review and workflow analysis.

4.1. Types and formats of data exchange

Data exchange between architectural and structural analysis models is particularly important within the design phase, because providing the architectural design together with the structural concept and calculations is a precondition for a building permit. It is also a process that benefits the most from the BIM workflow [34]. Several approaches to data exchange between the two domains exist. Ramaji and Memari [2] identified three possible ways to create a structural analysis model from BIM authoring architectural tools: (1) the structural analysis model is created in a native software tool; (2) the structural analysis model can be created from a native model by a structural analysis tool; or (3) an additional software tool (or a plug-in) is used to generate the structural analysis model. This classification is based on the primary role of the software tool that generates the structural analysis model: architectural tool, structural analysis tool, or a third-party data exchange tool (Fig. 1).

Looking at the data exchange process from the other direction, Aldegeily and Zhang [5] tested three ways to exchange data: (1) a direct link using a native file; (2) a direct link using API; and (3) an indirect link through a third-party channel. In this classification, the emphasis is on the parties defining the data exchange: a single software provider, two interoperating software providers, or an additional provider that is not part of the interoperating software tools.

Similar to Aldegeily and Zhang [5] and related to the most common BIM-based exchange in practice (buildingSMART open standards), two data exchange approaches are recognized: open and closed BIM. [35] defined the terms as follows: closed BIM refers to a workflow where software tools from only one software producer (or several cooperating producers) are used and the data exchange takes place via software-to-software interfaces; the open BIM approach refers to a workflow where an open, non-proprietary format is used for data exchange purposes. This data exchange classification is based on whether the accessible and modifiable intermediate building data model is used for the communication.

The software industry provides tools for the AEC industry that can support both open data exchange and closed software-to-software exchange. As described in the literature review (Section 3), the academic community focuses on improving open exchange, especially the IFC standard. The focus of the academic community in the data exchange classifications as suggested by Ramaji and Memari [2], Aldegeily and Zhang [5] and Bormann et al. [35] is on an additional software tool, an indirect link through a third-party channel, or open BIM respectively.

There are numerous reasons why third-party non-proprietary formats are considered the most promising solution for the digital data exchange in the AEC industry, such as:

- European statistics on AEC [36] show that offices mostly perform the tasks of a single domain, and therefore usually choose a software tool for their own needs without considering future collaborations. It is unlikely that all possible software tool combinations aiming to exchange information of interest will achieve a seamless data exchange with direct exchange interfaces;
- BIM workflows need to facilitate work with an integrated model. A single software tool that supports all the involved domains in design phase is lacking [4].

Other ways of achieving the exchange are therefore regarded as special cases and do not represent the current or future general practice. The only widely supported open building data exchange standard between architectural and structural analysis domains is IFC. At the moment, IFC is the most frequently implemented standard for BIM-based data exchange, with over 150 software tools claiming to support the schema [37]. Leading market software applications in the fields of architectural design and structural analysis claim to support either import or export or both of IFC data models.

Next to the software tools supporting export and/or import of IFC building data models while primarily dealing with their native data formats, there are tools intended for the management of IFC models. Although most of these support model viewing, mapping to additional file formats or checking the syntax of IFC files, some allow editing IFC

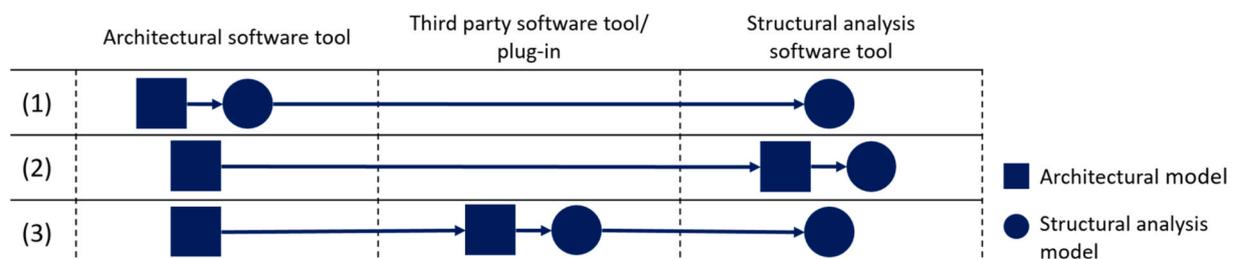


Fig. 1. Ways to generate a structural analysis model based on the authoring tool.

files and preparing them for further collaboration (e.g. open source BIMserver platform, BIMData or xBIM). However, focus of these tools is data management, rather than the conversions between architectural design and structural analysis models.

Data exchange workflows using IFC building data model are still sequential. The first model in the design phase is generally defined by an architect, in the architectural BIM authoring software tool. If the BIM authoring tool supports the IFC-based export, the architect is able to export the model to an IFC subschema, MVD. The export provides the user with the building data model based on the IFC schema, which can then be imported into software tools which support IFC-based import (e.g. structural analysis tools).

4.2. Process analysis in open data exchange

Ecosystems are dynamic and purposive networks where participants collaborate to create value that would not be possible for a single participant alone [38]. In this paper, we conceptualize the AEC industry as an ecosystem, as it is described in [13,14]. The concept of the AEC industry as ecosystem makes it possible to analyze an IFC-based data exchange system on two levels:

- Project level involving data exchange processes of import and export of building models from the end-user perspective;
- Inter-organizational level involving higher-level processes of various AEC stakeholders and their perspectives.

This two-level approach (project and inter-organizational interactions) shifts the focus from the already well examined project level and its processes, namely export and import of data, thus striving to achieve a more systematic insight by analyzing the processes taking place on the inter-organizational level. By investigating one use case (project level) through six similar workflows involving various software tools, the normative and regulative as well as software development perspective is included. Thereby the proposed improvement strategies address inter-organizational issues such as software certification or interoperability issues reflected in various software tool. There is a knowledge gap on the influence of higher-level inter-organizational processes, such as software certification or development, on the efficiency and efficacy of project level processes and data exchange performance.

On the inter-organizational level, the processes influencing IFC-based data exchange between architectural design and structural analysis are assessed from different stakeholder-perspectives in the AEC ecosystem using:

- Literature review [6,27];
- Recommendations of organization buildingSmart [32,37,39];
- Results generated through previous research-led teaching [40] and ongoing collaborative research with the industry [41].
- And finally testing of software tools (comparative study in Section 5).

The analysis showed that in addition to the direct influence of end users, data exchange is affected by the normative and regulative instances, as well as the software industry:

- **End-users: generation and exchange of building models** through various workflows as practiced by the end users (architects or structural engineers);
- **Normative and regulative instances: data exchange standardization and software certification** linking AEC practices with the software industry;
- **Software industry: software development** resulting in tools that support AEC practitioners.

The particular processes and their impact on the project level and

data exchange results are described in the subsections below.

4.2.1. End-users: generation and exchange of building models

Generating and exchanging building data in software tools primarily depends on the authoring software tool interface, which will be more closely described from the software industry perspective (see Subsection 4.2.3). BIM authoring software tools provide architects and structural engineers with object classes representing building elements for specific domains. The software tools can provide multiple ways to generate the same building element. In addition to the software tools, this process depends on the end users' skills and competencies. The end-user concepts of the same building element can differ, which can result in multiple representations of the same real-world product.

Architects and structural engineers use the software tools for their design and analysis tasks intuitively or as instructed. The behavior of the end users can be influenced through the instructions, for instance, on the ways to generate and exchange building data provided by the software industry, but also through the norms and practices that define the workflow.

4.2.2. Normative and regulative instances: data exchange standardization and software certification

Within the AEC ecosystem, the AEC industry recognizes the need for regulative and normative instances to deal with specific topics, and establishes these instances with the aim of improving the planning workflows. Compared with other industries, the AEC industry is slow in adopting new processes and technologies [42]. The lack of new processes and technologies slows down innovation, increases risk and reduces productivity. Construction projects are often over budget and not finished on schedule. The greatest potential to change this comes from digitalization of the industry. However, in workflow digitalization, there are still many topics which have not been addressed, especially regarding the digitalization of data exchange. buildingSMART defines the standards for the IFC-based exchange, integrated building data model, and domain-specific data models (Fig. 2), as well as the regulative process for software certification, as summarized below.

• Normative: inter-domain relations

ISO 29481-1 [43] and ISO 29481-2 [44] describe information delivery manual (IDM) standards; where the former (part 1) defines the methodology and format and the latter (part 2) defines the interaction framework. Both are concerned with interoperability between software applications used during the design phase. Part 1 defines a basis for reliable information exchange and information sharing for users, as well as how user requirements are to be documented. Part 2 specifies the way in which events or "coordination acts" during the project have to be described. IDMs are intended to standardize the mapping of the exchange requirements for the exchange process.

• Normative: integrated data model

ISO 12006-3 [45] describes a taxonomy model; that is, how the

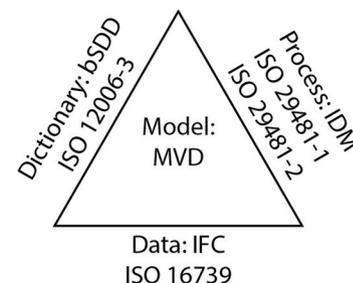


Fig. 2. Basic standards (based on [32]).

concepts are defined, grouped and their relationship, where basic entities of a model are objects, collections and relationships. This taxonomy model is called buildingSMART Data Dictionary (bSDD) (previously called International Framework for Dictionaries (IFD)). The dictionary specifies a language-independent information model that can be used for the development of building data schemas for storing and providing information related to construction works. ISO 16739 [46] specifies the IFC schema defined using the EXPRESS data specification language.

- **Normative: domain-specific data model**

Interdisciplinary data exchange needs to encompass and unify the very diverse sets of information and models required for each domain, so standards are needed which specify the domain-specific data scopes, including the data exchange requirement lists. Therefore, there are currently two MVD standards valid for IFC4 data schema; namely, “IFC4 design transfer view” and “IFC4 reference view”. The MVD reference view is suitable for workflows where a model is used only as a reference, while the design transfer view is designed for workflows where the next task is performed with the exported model and the modification of content is expected.

However, the most widely implemented MVD is CV V2.0, created for IFC 2x3, which is primarily intended for information exchange between architecture, structural and mechanical engineering, as explained in [37,39]: “The shared building information model is supposed to be re-editable by the receiving application. It includes the definition for spatial structure, building, and building service elements with shape representations, including both, parametric shapes for a limited range of standard elements, and the ability to also include non-parametric shape for all other elements.”

- **Regulative: software certification**

The software certification process for MVDs created within the IFC 2x3 schema is only applicable to MVD CV V2.0. Since 26 June 2017 it has been available for IFC4 reference view. The comparative study in this paper (Section 5) tests the software tools certified with the most commonly used MVD, CV V2.0, which is currently the only certified MVD intended for information to be edited in a receiving tool. IFC4 design transfer view, which will be used for the same purpose with IFC4 has not yet been certified (at the time of writing). Several software tools claim to support design transfer view without certification, but the structural analysis tools still do not support the import.

The certification workflow is described in [33]; and for CV V2.0 it is organized by the Model Support Group and the Implementation Support Group within buildingSMART. The workflow is divided into export and import certification. For export certification, exchange requirement concepts are provided depending on the domain for which the participating software is used. There are three different types: architectural, structural and building services. The export validation is carried out through a self-check performed by a software developer and an additional check by a certification tester. Consequently export is checked automatically and manually. Import certification involves importing a series of import calibration files that correspond to the exchange requirements. In addition, complex test files are provided. Next, manual self-tests are carried out and the results are validated by the certification tester. Finally, software developers design an adequate user interface for export and import, as well as appropriate documentation.

4.2.3. Software industry: software development process

The way the building models are defined largely depends on the technical possibilities, usefulness and usability of the interfaces provided by the software industry. Although there are a large number of computational tools on the market, they are often developed without input from people with professional domain knowledge, thus failing to

meet the end-user needs. Digital planning workflows are not standardized, so the software industry aims to support flexible user workflows, but there is a lack of modeling rules for specific tools, and the software tool architecture is not necessarily reflected in the interface.

In specific cases, such as data exchange, the digital tools being used must conform to regulations in order to provide certain exchange functionalities. For example, for data exchange with IFC, software tools can be certified for export and/or import of IFC building data models with a specific MVD (i.e. CV V2.0), which is intended for data exchange between architecture and structural analysis tools and is accepted by all leading software producers.

Fig. 3 illustrates the part of the AEC ecosystem examined by the current research, namely the data exchange ecosystem. It shows the processes taking place during the data exchange between architectural design and structural analysis models using the IFC schema. The data exchange processes taking place on the project level are shown in the inner dashed rectangle. The influencing processes on higher inter-organizational level such as software certification, standardization, software development and generating and exchanging building data and their relation to the project level is represented within the outer dashed rectangle.

5. Comparative study: interoperability testing

After conducting a thorough literature review and analyzing the procedures on the inter-organizational level that affect data exchange on a project level, a comparative study was conducted in order to determine the origin of the data-exchange-related problems. The comparative study presented below tested the software interoperability of various architectural design and structural analysis BIM authoring tools by transferring representative building elements.

The study began by setting the boundary conditions: interoperability testing was chosen as the software testing method; the software tools included Archicad, Revit and Allplan for the architectural design, and SCIA and RFEM for structural analysis; for the IFC-based data exchange the data generation and exchange practices recommended by software producers were used; a use-case building with structural system consisting of load-bearing walls and columns was modelled and transferred.

Next, an evaluation framework was developed, with a focus on geometry transfer, based on the adapted interpretation units from [24]. Four geometrical interpretations are evaluated: punctual, linear, planar and connectivity interpretations.

The comparative study was then conducted and presented according to the evaluation framework.

Finally, several exchange requirements from the comparative study were isolated and investigated from the end-user, normative and regulative, and software developer perspectives through the corresponding processes of generating and exchanging building data, standardization, and certification and software development. The analysis of the requirements within the procedures makes it possible to identify the points in the ecosystem that are responsible for problems in data exchange between architectural design and structural analysis.

5.1. Research design and methodology

- **Software testing methodology**

In order to test the end-user experience, we chose interoperability testing as the software testing methodology, because conformance testing does not give insight into the actual software behavior between software tool pairs. The aim was to test only one-way exchange: architectural design to structural analysis model. The comparison of two IFC files was not considered, because that requires an additional export from the structural analysis tool, and structural analysis tools do not always support import and export with the IFC schema [32]. Therefore, interoperability was tested with one IFC export and one import.

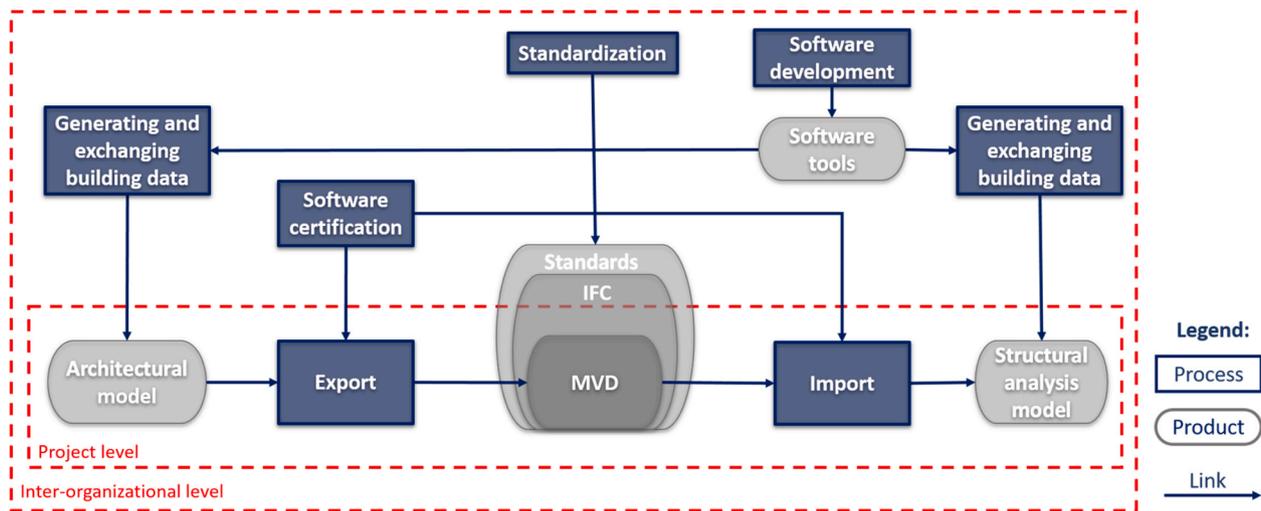


Fig. 3. Data exchange ecosystem: Processes within project and inter-organizational levels.

• Software tools

For the architectural design (creation) of BIM model variants and their export to three IFC building data models, we used three software tools: Allplan [47], Archicad [51] and Revit [48]. The two structural analysis software tools that were used to import the IFC building data models were SCIA [55] and RFEM [54]. Every application that was used in this research was certified for CV V2.0.

• Modeling and exchange workflow

The modeling workflow followed the modeling instructions, if provided, or the most intuitive approach for each software interface. Although all three BIM tools offered the possibility to edit IFC export-related properties of a single building element, the default assigned properties were not changed. Only the default interface settings for CV V2.0 export were used. In this way, three IFC building data models were exported. Following the generation of three model variants, the IFC files were imported into the two different structural analysis software tools. For the import, the default options were used and the complete model was imported. Thus, data exchange was accomplished with minimal end-user intervention.

• Use case design

The iconic building Villa Savoye (plans courtesy of Fondation Le Corbusier) was chosen as a use case model for the comparative study because it incorporates all relevant building elements (foundation, column, slab, wall, roof) commonly used in concrete frame construction, or load-bearing masonry and concrete wall constructions.

5.2. Evaluation framework

The imported models were evaluated by visually comparing the results with the required models, focusing on the geometry. Building elements were analyzed separately to see whether they contain all required geometrical data in the structural analysis software tools; and element coordinates were compared. Based on the reviewed literature, which defines the problems of data exchange between architecture and structural engineering, three groups of geometry-related “interpretation units” were identified: linear elements transformation, planar elements transformation and connectivity adjustment [24]. Single foundations, which are part of the generated use case model variants and are not addressed in the analyzed literature, were added as punctual elements transformation. Punctual elements are represented as points in the

structural analysis software tools and their geometrical shape parameters become relevant for the analysis, but not for the building model. The results were summarized and categorized based on the interpretation type and the building element type, thus demonstrating a systematic interoperability overview.

5.3. Results of the comparative study

The comparative study looked at the inconsistencies that occurred in the data transfer from the architectural to the structural models, focusing on geometry issues. The results showed that the correct import and interpretation of building elements depends on the combination of software tools used, but none of the combinations gave satisfying results. Export from Allplan consists of 33.286 entities, from Archicad of 39.691 entities and from Revit of 32.387 entities. Building elements are represented similarly, with IFC classes *IfcWallStandardCase*, *IfcOpeningElement*, *IfcColumn*, *IfcFooting*, *IfcBuildingElementPart*, *IfcSlab*, *IfcRoof*, *IfcStair*, *IfcDoor* or *IfcWindow*, *IfcStairFlight*, but the number and occurrence of these entities differs between the exports. *IfcShapeRepresentations* with “Body” representation identifier that define 3d geometry of building elements also largely differ. In Allplan export 88% representation types are swept solid and the rest are boundary representation (brep), in Archicad export is 99% swept solid and the rest is surface model, whereas Revit has 66% swept solid, 31% mapped representation, 2% clipping and 1% brep. Inconsistencies found in the comparative study can be traced back to individual IFC entities, but as none of the elements did render completely satisfying, only some will be demonstrated in this paper. Table 1 shows the screenshots of the exported IFC models, Table 2 and Table 3 some of the inconsistencies found in the importing tools. Table 4 is a summary of interoperability performance during the data exchange from the architectural to the structural analysis model. A proper axis position was not regarded as an excluding factor.

Within this study, only a limited number of elements was used, so the identified problems are limited to these particular building elements. Multi-layered slabs were modelled as multiple one-layered slabs in Allplan, since Allplan does not support multi-layered slabs. The problems identified within this research were limited to the geometrical interpretation of the building elements involved in this comparative study. However, there are a number of further building elements relevant for the data exchange (these are often workflow specific). Nevertheless, the scope of the geometry interpretation problem is evident. All the geometry interpretation groups (i.e. punctual, linear and planar element transformation) and connectivity were part of the use case. The interpretation of punctual elements and the interpretation of the

Table 1
Screenshots of models exported to IFC CV V2.0

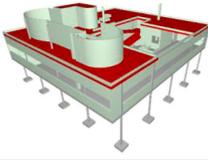
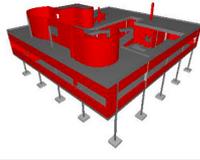
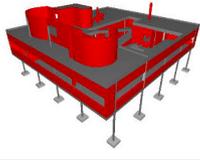
Model screenshot			
Native software tool	Allplan	Archicad	Revit

Table 2
Screenshots of inconsistencies identified in importing software tools (1).

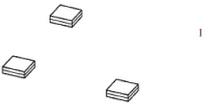
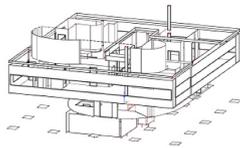
Model screenshot			
Interpretation	Punctual interpretation	Linear interpretation	Planar interpretation (a)
Software tool combination	Revit export/RFEM import	Revit export/RFEM import	Allplan export/RFEM import
Inconsistency	Foundations interpreted as planar and linear elements instead as joints	Columns are missing	Wall interpreted as slab, offset for half of the height

Table 3
Screenshots of inconsistencies identified in importing software tools (2).

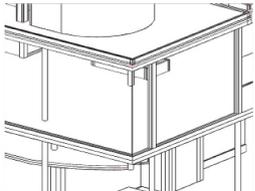
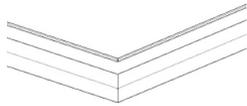
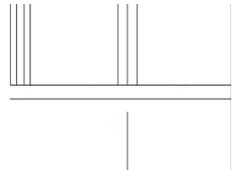
Model screenshot			
Interpretation	Planar interpretation (b)	Planar interpretation (c)	Connectivity
Software tool combination	Archicad export/SCIA import	Allplan export/RFEM import	Archicad export/SCIA import
Inconsistency	Missing multi-layered walls with air layer	Axis of planar element placed on the top plane of a slab, instead of the middle	Elements lose their connectivity after being interpreted

Table 4
Results of the comparative study.

Interpretation Type	Building element properties			Software used for the exchange					
	Type	Number of layers	Flat (-) Tilted (/)	Archicad		Revit		Allplan	
				SCIA	RFEM	SCIA	RFEM	SCIA	RFEM
Punctual transformation	Single foundation			x	x	x	x	x	x
Linear transformation	Strip foundation			x	x	x	x	x	x
	Column (square)			✓	x	✓	x	✓	x
	Column (round)			✓	x	✓	x	✓	x
Planar transformation	Floor slab	1	/	x	x	✓	✓	x	x
		1	-					✓	✓
		2+	-	✓	✓	✓	✓		
	Slab opening	2+		✓	✓	✓	✓	✓	x
	Roof	2+	/	x	x	✓	x	x	x
	Roof opening	2+	-	✓	✓	✓	x	x	x
		2+	-	x	✓	✓	x	x	x
	Wall	1		✓	✓	✓	x	✓	x
		2+		x	x	✓	x	✓	x
	Wall opening	1		✓	✓	✓	x	x	x
Wall opening	2+		x	x	✓	x	x	x	
Connectivity				x	x	x	x	x	x

connectivity of interpreted building elements did not take place in any of the examined use cases. Linear and planar building elements were, in some cases, interpreted but the performance differed depending on the combination of software tools. The variants generated unexpected and unpredictable results. This emphasized that the use of a BIM workflow with IFC is not yet fully qualified to fulfill the inter-domain data exchange tasks.

Product models must be based on accurate information to serve the stakeholders' needs at every specific project stage. As the comparative study showed, at present this is not the case. For instance, the load-bearing property of a layer and information about the adjacent elements was generally lost during the exchange, so even the non-load-bearing elements were imported. Even if the load-bearing property of a specific element was available in the exported building model it was not considered by the importing software tool, thus representing significant problem for structural engineers, as the loadbearing properties get lost throughout the data transfer process. On the other hand, some information – such as axis position or foundations – was interpreted in a way that rendered them incorrectly in the structural analysis software tools. Certified software tools did not provide the means of performing the data exchange satisfactorily, because they failed to achieve interoperability.

5.4. Comparative study: effects of inter-organizational processes

This subsection links the exchange requirements identified as problems in the comparative study to the procedures of data exchange, as proposed in Section 4. Some exchange requirements from the results of the interoperability testing are located within the software tools and their interfaces, the buildingSMART concept lists and certification checklists [32] in order to allocate the problems in the procedures and develop proposals for procedural improvements of inter-domain data exchange.

The effects of inter-organizational processes on data-exchange performance were tested as follows:

- Several exchange requirements were selected and their presence in the software tools for architectural design and structural analysis models was investigated, documenting the effects of software development process;
- The exchange requirements were located in the MVD, where they are described as concepts, and in the IFC schema, in order to analyze the exchange requirements in the standardization process;
- The certification process was analyzed using the certification lists provided by buildingSMART and the software industry;
- The generation and exchange of building data was assessed by examining the availability of instructions and clarity of interfaces to perform the required modeling and exchange tasks.

Exchange requirements in MVDs are defined by buildingSMART through a selection of attributes and relations called root concepts. The root concepts are represented through entities that can be spatial, building, building services or structural elements. buildingSMART uses the concepts lists for the certification process which are available online for each of the root concepts [37,39]. These concepts are categorized as mandatory, optional, not relevant, or excluded. Furthermore, it is indicated whether each is an export or import requirement. The export requirements can be a requirement of architecture, building services or structural software tools. The concepts are the base standard for each MVD implementation in software tools. Additionally, checklists are available [32], which explain how the certified software tools supported the MVD concepts during the certification.

Table 5 presents the correlations between the MVD concept list and the exchange requirements from the end-user perspective. Concepts are described, as in the buildingSMART CV V2.0 concept list, by using the code, name and the relevance for the export and import, as well as the respective exchange requirement based on the end-user conception.

Table 5
Comparison of the MVD concept list and the exchange requirements.

Concept code	Concept according to buildingSMART concept lists	Exchange requirement	Export relevance	Import relevance
030-6-1	Geometry sweptsolid	Column geometry	Mandatory	Mandatory
030-6-9	Geometry mapped	Column geometry	Mandatory	Mandatory
030-6-1	Geometry sweptsolid	Wall geometry	Mandatory	Mandatory
210-1	Property set IFC common	Load-bearing property wall	Mandatory	Mandatory
110-2	Connectivity path	Connectivity wall	Mandatory	Mandatory

Table 6
Behavior of the concepts as described in certification checklists.

Concept code	Export certification			Import certification	
	Archicad	Revit	Allplan	SCIA	RFEM
030-6-1	Supported		Restricted ^a	Supported	Supported
030-6-9		Supported		Supported	Supported
030-6-1	Supported	Restricted ^b	Supported	Supported	Supported
210-1	Supported	Supported	Supported	Not tested	Supported
110-2	Restricted ^c	Supported	Restricted ^d	Not tested	Not supported

^a Certain geometries in the certification process cannot be created within a software tool; however, elements from the use case are supported.

^b Revit does not support layers thinner than 1.6 mm; such layers were not part of the use case.

^c One type of connection is not supported, which was not a part of the use case.

^d Allplan calculates the connectivity path only from straight walls.

Table 6 shows the same CV V2.0 concepts as given in Table 5, but with their behavior in the software tools during the software certification process as described in the certification checklists [32], for each of the tested tools. While columns are exported as swept solid geometry from Archicad and Allplan (concept code 030-6-1), they are exported as mapped representation (030-6-9) from Revit.

Finally, Table 7 shows if and how the concept was supported by the importing tools. The “properly imported” and “not properly imported” columns show the number of software tool combinations in the comparative study where the import results were or were not properly rendered in the importing software tool.

6. Data-exchange improvement roadmap

Based on the previously conducted problem identification on inter-organizational level and its effects on project-based data exchange, in the following Section the improvement strategies for the data exchange between architectural design and structural analysis on inter-organizational level, including relevant stakeholders and processes

Table 7
Behavior of the concepts in the software tools.

Concept code	Properly imported	Erroneously imported	Remarks when the exchange requirement is erroneously imported
030-6-1	2	2	Geometry not properly interpreted
030-6-9	1	1	Geometry not properly interpreted
030-6-1	2	4	Geometry not properly interpreted
210-1	0	6	Element not recognized by the importing tool
110-2	0	6	Property not recognized by the importing tool

will be proposed.

6.1. End-user: generating and exchanging building data

In the comparative study the end-user actions were reduced to minimum deviations from the default options available in the software tools. All exchange requirements were supported in the software tools, and most of them could be created in multiple ways. Most of the software tools did not provide clear instructions on how to properly model the elements: this option was left to the architect. Multiple name and property definitions are left to the architect, without any instructions on how to define them (e.g. material).

The interfaces for the exchange differed significantly between the software tools. To change the export information, the architect has to be familiar with the IFC schema. On the other hand, the importing software tools provide little possibility to influence the import. This nears the “black box” scenario which should be avoided in the AEC industry [49]. [49] state that it is necessary to allow the end users without expert programming skills to make changes in a collaborative framework. This is provided to varying degrees in the investigated software tools; however, none of the export or import interfaces were completely open to the end user (allowed insight into the background processes).

In this research, the interoperability testing was conducted from the point of view of the end user following default settings. This means that the existing instructions, provided mostly by software developers, although lacking the digital building data regulations on the AEC industry level, were used for the generation and exchange of building data. In this way, the research does not focus on various workarounds which exist as common practice in the market, because they are software-tool-specific and exclude numerous stakeholders in the design phase. Thus the strategic improvements for this process are (a) more detailed modeling instructions for the end user and (b) clear descriptions of the open exchange interfaces allowing for end-user interventions. The effect of the improvements would still greatly depend on end-user behavior and it might not result with the same improvements between different software tools.

6.2. Normative and regulative instances: data exchange standardization and software certification

• Normative: inter-domain relationships

The data exchange workflow in the AEC industry lacks important inter-domain relationships between the domain-specific representations that are required for the digitalization of data exchange. The exchange requirements are exported as architectural elements to the IFC building data model, but the relationships between the architectural elements and their respective structural representations were not defined. This issue is partly addressed by buildingSMART through the IDMs, yet it was not considered for the most widely implemented CV V2.0 [50]. The mapping of the business process with the inter-domain relations remains in the hands of the software developers and software industry, and not the AEC industry or the end users.

The insufficiently defined workflows within the AEC industry and lack of collaboration with the software industry result in misinterpretations and models that cannot be used for the structural analysis as such. The outcome of digital data exchange depends on the interoperating domains and the complexity of the interpretation processes, the interpretation between the architectural design and structural analysis domains are significant. Slow standardization is recognized as a general AEC industry problem. The standardization of inter-domain relationships is greatly influencing data exchange and could systematically improve the inter-domain exchange. Defining the relationships for the purpose of digitalization requires a significant effort, and needs to be implemented through software development processes before it can be applied by end users.

• Normative: integrated data model

The IFC2x3 schema supports all the analyzed building elements from the comparative study. The exchange requirements defined in the architectural building data model as well as the exchange requirements needed in the structural analysis model are both represented in the IFC2x3 schema. However, the integrated data model of the IFC2x3 schema was not used for the data exchange between architectural design and structural analysis because MVD CV V2.0 encompasses both disciplines. Only the exported architectural building data model was defined within MVD CV V2.0, so the architectural model was directly imported to the structural analysis tools.

The workflow used in the comparative study involved a single MVD, so the structural analysis exchange requirements were never part of the integrated building data model, and the structural model was created from the architectural export by the structural analysis tools. In the comparative study, the role of the integrated data model is completely taken over by the MVD CV V2.0 for the exchange between architecture and structural engineering. Here, the proposed improvement strategy is to have the integrated data model resulting from the relevant MVDs.

Equally, the IFC schema does not reflect the end-user concepts, which makes IFC exports difficult for most end users to understand. Therefore any intervention with the IFC building data model requires professional assistance (e.g. from a data analyst or schema specialist). Thus the integrated model would need to be either simplified or supported with user-friendly interfaces that would allow for more user interventions.

• Normative: domain-specific data model

MVD CV V2.0 is a subschema defined by the list of concepts for IFC2x3. This MVD defines diverse requirements for a single MVD schema which software tools need to fulfill: three export and unique import requirement lists. However, the ability of satisfying these requirements still varies among the tools, and none of them was able to fully satisfy the requirements. When a specific exchange requirement is not addressed in the concept list, software developers treat it based on their own interpretation or judgment. In the comparative study the concept list was only partly answering the exchange requirements.

The list of concepts for export software tools is based on the unique concept list of the architectural export. CV V2.0 is a subschema of IFC2x3, and although it is rich some information can be represented in multiple ways (i.e. it contains redundancies). Occasionally the necessary exchange requirements are getting lost when being mapped from the architectural BIM software tools.

The list of concepts for import software tools is based on the unique concept list of import. Capabilities for importing specified exchange requirements vary among different software tools, and especially between the domains. Information which is not standardized in the MVD, but is still part of the export, is unexpected and unused in the importing software tool.

The concept of multiple MVDs has not properly come to life within the software tools. Most of the tools only support the CV V2.0 subschema, which was not suitable for any of the analyzed software tools in its full extent. The improvement strategy proposed here is to develop domain-specific schemas with an emphasis on the relationships between the building elements in various representations.

• Regulative: software certification

Another process affecting the quality of data exchange is software certification. The concept list specifies requirements for export and import testing. Certification checklists document the performance of the software tools during the certification process. Although concepts can be mandatory, optional, not relevant, or excluded, these concept descriptions are not necessarily respected for all tested elements, meaning

that mandatory concepts are not always supported in the checklists of a certified software tool.

Domain-specific requirements are only specified for export testing. The standards according to which the software tools are certified do not follow the domain or the requirements of the specific software tool. Consequently, it cannot be expected that all the concepts are supported in each software tool. This is possibly the reason why software is certified without supporting all concepts that are described as mandatory.

The proposed improvement strategy is a new certification process that reflects the domain or software-tool specific exchange requirements. The performance of interoperability testing with the open schema is an important usability indicator for end users, which should be clearly described with the certificate.

6.3. Software industry: software development

The software development process has a direct impact on overcoming the majority of data exchange inconsistencies. Although most of the inconsistencies can be first traced to suboptimal software development, the software tools for the AEC industry conform to the normative and regulative processes, if available. In order to achieve a satisfactory data exchange the software tools need to conform to standards, as is the case with the open data exchange. Software development is primarily focused on the performance of a specific tool, whereby interoperability with the open standards depends on the normative and regulative elements responsible for it. However, because software development attempts to answer the needs of the heterogeneous AEC industry and because of missing standards, the industry is provided with rich as well as redundant ways of generating and exchanging data.

Therefore in this study software development is not regarded as a process that can systematically improve data exchange, because it addresses the improvement of a single software tool, whereas for collaboration purposes within the heterogeneous industry systematic improvement is required.

7. Summary

Table 8 lists the data exchange requirements that showed inconsistencies in the comparative study, the respective procedures and the improvement strategies. The problems are listed according to the effect on the end-user experience: namely, the first problem solved can directly affect the end-user experience and the last problem solved has to be implemented through other processes before it can reach the end user. For example, the inter-domain interpretation standards must be introduced in the integrated schema, domain-specific schema and the software certification first, before they reach the end user. Incorrect or missing exchange requirements in the comparative study can be traced back to all four normative and regulative processes. Exchange

Table 8

Identified problems and proposed improvement strategies that influence the end-user experience during the data exchange.

	Problem	Process	Improvement strategy
1	Exchange requirements tested during certification, but missing in the resulting model	Regulative: Software Certification	Improved or new software tool certification that is software or domain specific
2	Exchange requirements exist in the MVD, but not tested in the software tool	Normative: Domain-specific data model	Domain-specific data models supporting the necessary exchange requirements
3	Exchange requirements not part of the integrated data model	Normative: Integrated data model	Integrated data model encompassing all domain-specific data models
4	Exchange requirements transferred but incorrectly interpreted	Normative: Inter-domain relations	Building data interpretation standards

requirements that do not deliver expected results received different treatments within the processes: some exchange requirements are tested during certification processes but are still missing after the data exchange; some exist within the MVD CV V2.0 but are not tested during the certification; some exist in the IFC schema but are not part of MVD CV V2.0; and some are transferred but rendered incorrectly by the importing structural analysis tool.

Therefore, we propose the following strategy for the systematic improvement of data exchange between architectural and structural analysis:

1. Missing interpretation rules need to be developed and introduced in the data exchange workflow;
2. The interpretations need to take place between domain-specific models, supported by domain-specific building data models;
3. Integrate building data models as a result of multiple interrelated domain-specific building data models;
4. Introduce a domain-specific or software tool specific certification process.

8. Conclusion

This paper assessed the flawed data exchange between architectural design and structural analysis via a literature review and a comparative study, and linked the results to the inter-organizational procedures influencing the data exchange processes on the project level. The particular focus was on the analysis of IFC-based data exchange. The paper proposes and prioritizes improvement strategies for data exchange between architectural design and structural analysis.

The research question “Why does model-based data exchange between architectural design and structural analysis still not function without significant data and information losses?” addressed the identification of the origins for still flawed data transfer between architectural and structural analysis models. Thereby the data exchange was examined not only from the end-user perspective, but also from the normative and regulative, and software industry perspectives. The respective processes of generation and exchange of data, software certification and standardization and software development influencing the data exchange were analyzed and supported by the comparative interoperability testing study.

The generation and exchange of building data and the software development processes directly impact the end-user experience. However, improvements to these processes are dependent on the software industry, which works on a tool-by-tool basis, and cannot systematically improve the data exchange. A systematical improvement is especially necessary because of the heterogeneity of AEC industry. On the plus side, regarding normative and regulative processes: software certification and standardization could systematically improve the exchange.

The certification process directly affects software development. Tested certified software does not necessarily support the exchange requirements it is tested for. A single concept list (as CV V2.0) with some variations cannot satisfy the diverse needs of multiple domains and software tools. Hence, software tool certification cannot be regarded to be a valid indicator of interoperability.

The MVD standard, as a domain-specific data model, is the base for the certification process. In addition to the fact that it is not being properly implemented in the software tools, MVD CV V2.0 does not recognize the need for multiple domain-specific MVDs because it supports three domains at the same time. In that way MVD CV V2.0 overtook the role of the integrated building data model in the case of data exchange between architectural design and structural analysis. The domain-specific models need to reflect the domain-specific concepts in order to achieve communication with domain-specific software tools. The integrated model relates the inter-domain concepts so the communication between the domain-specific concepts could be accomplished.

The data interpretations that occurred in the comparative study are

defined by the software developers responsible for import applications, and are not yet standardized. The study showed that missing standards for the interpretation steps that describe the inter-domain relationships result in inconsistent imports of the same IFC-based model in multiple software tools. The lack of inter-domain interpretation standards reflects on the IFC schema, lack of MVDs, on the erroneous certification process, and finally the software development, affecting the end-user experience.

In conclusion we propose the following solutions to improve end-user experience:

- Improved or new software tool certification process;
- Multiple domain-specific building data models which would eventually reflect on the integrated building data model;
- Introduction of the interpretation rules in the current data exchange workflow.

Systematic improvement needs a top-down approach. First, it is necessary to define interpretation rules, then introduce these to the domain-specific models, then finally create a domain-specific certification procedure. Although there have been numerous attempts to improve data exchange between architectural design and structural analysis in the AEC industry, end users are still not able to achieve a seamless exchange. This review identifies the most important procedural problems and potential for improvements which would systematically develop the data exchange. In particular, the missing inter-domain interpretation rules are found to be the most critical procedural problem. This work will serve as a basis for future research and technological development regarding data exchange between architectural design and structural analysis.

The research described above is limited to the testing of the software tool performance for the particular building elements used in the comparative study (mainly the geometrical properties). The proposed strategy is based on existing data exchange using the IFC standard, and other standards/schemas could show different problems. The applied modeling practices affect the results of the comparative study to a certain degree; the required structural analysis model might also differ based on the workflow and practices.

Future research will examine existing standards in architecture and structural analysis domains in order to generate tools to develop new and improve existing domain-specific knowledge and inter-domain relationships that would optimally meet the needs of the AEC industry. The objective is to expand the knowledge of the BIM-based concepts for structural analysis and relate them to the architectural concepts. In the next step we will aim to define the interpretations between the domain-specific models.

CRediT authorship contribution statement

Goran Sibenik: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Visualization. **Iva Kovacic:** Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was conducted as part of the Doctoral College "Computational Design" funded by TU Wien (Vienna, Austria).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.jobe.2020.101589>.

References

- [1] D. Holzer, *The BIM Manager's Handbook : Guidance for Professionals in Architecture, Engineering, and Construction*, Wiley, Chichester, England, 2016, ISBN 1-118-98240-1.
- [2] I.J. Ramaji, A.M. Memari, Interpretation of structural analytical models from the coordination views in building information models, *Autom. Construct.* 90 (2018) 117–133, <https://doi.org/10.1016/j.autcon.2018.02.025>.
- [3] McGraw Hill Construction, *The Business Value of BIM for Construction in Major Global Markets: How Contractors Around the World Are Driving Innovation with Building Information Modeling*, 2014. https://www.icn-solutions.nl/pdf/bim_.pdf. (Accessed 30 September 2019).
- [4] C. Eastman, P. Teicholz, R. Sacks, G. Lee, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, third ed., John Wiley and Sons Inc., Hoboken, New Jersey, United States of America, 2018.
- [5] M. Aldegeily, J. Zhang, From architectural design to structural analysis: a data-driven approach to study building information modeling (BIM) interoperability, in: T. Sulbaran (Ed.), *Proc., 54th ASC Annual International Conference, ASC Associated Schools of Construction*, Fort Collins, Colorado, 2018, pp. 537–545. <http://ascpro0.org/cd//CPRT152002018.pdf>. (Accessed 30 September 2019).
- [6] M. Laakso, A. Kiviniemi, The IFC standard - a review of history, development, and standardization, *J. Inf. Technol. Construct.* 17 (2012) 134–161. <http://www.itcon.org/2012/9>. (Accessed 30 September 2019).
- [7] G.B. Ozturk, D. Arditi, I. Yitmen, M. Yalcinkaya, The factors affecting collaborative building design, *Procedia Eng.* 161 (2016) 797–803, <https://doi.org/10.1016/j.proeng.2016.08.712>.
- [8] J.P. Zhang, Z.Z. Hu, BIM- and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 1. Principles and methodologies, *Autom. Construct.* 20 (2011) 155–166, <https://doi.org/10.1016/j.autcon.2010.09.013>.
- [9] M. Oh, J. Lee, S. Wan Hong, Y. Jeong, *Integrated system for BIM-based collaborative design*, *Autom. Construct.* 58 (2015) 196–206, [10.1016/j.autcon.2015.05.003](https://doi.org/10.1016/j.autcon.2015.05.003).
- [10] T. Pazlar, Z. Turk, Interoperability in practice: geometric data exchange using the IFC standard, *J. Inf. Technol. Construct.* 13 (2008) 362–380. <http://www.itcon.org/2008/24>.
- [11] A. Grilo, R. Jardim-Goncalves, Value proposition on interoperability of BIM and collaborative working environments, *Autom. Construct.* 19 (5) (2010) 522–530. <https://doi.org/10.1016/j.autcon.2009.11.003>.
- [12] T. Cerovsek, "A review and outlook for a 'Building Information Model' (BIM): a multi-standpoint framework for technological development, *Adv. Eng. Inf.* 25 (2011) 224–244, <https://doi.org/10.1016/j.aei.2010.06.003>.
- [13] L. Pulka, M. Ristimäki, K. Rajakallio, S. Junnila, Applicability and benefits of the ecosystem concept in the construction industry, *Construct. Manag. Econ.* 34 (2) (2016) 129–144, <https://doi.org/10.1080/01446193.2016.1179773>.
- [14] G. Aksenova, A. Kiviniemi, T. Kocaturk, A. Lejeune, From Finnish AEC knowledge ecosystem to business ecosystem: lessons learned from the national deployment of BIM, *Construct. Manag. Econ.* 37 (6) (2018) 317–335, <https://doi.org/10.1080/01446193.2018.1481985>.
- [15] M. Weise, P. Katranuschkov, T. Liebich, R.J. Scherer, "Structural analysis extension of the IFC modelling framework" (Special issue), *J. Inf. Technol. Construct.* 8 (2003) 181–200. <https://www.itcon.org/paper/2003/14>.
- [16] C. Wan, P.-H. Chen, R.L.K. Tiong, Assessment of IFCs for structural analysis domain, *J. Inf. Technol. Construct.* 9 (2004) 75–95. <https://www.itcon.org/paper/2004/5>.
- [17] M.H. Serror, J. Inoue, Y. Adachi, Y. Fujino, Shared computer-aided structural design model for construction industry (infrastructure), *Comput. Aided Des.* 40 (2008) 778–788, <https://doi.org/10.1016/j.cad.2007.07.003>.
- [18] X.Y. Deng, T.-Y.P. Chang, Creating structural model from IFC-based architectural model, in: *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montréal, 2006. Canada, 14–16 June, <http://itc.net/2006-tf577.pdf>. (Accessed 30 September 2019).
- [19] L. Qin, X.-Y. Deng, X.-L. Liu, Industry foundation classes based integration of architectural design and structural analysis, *J. Shanghai Jiaot. Univ.* 16 (1) (2011) 83–90, <https://doi.org/10.1007/s12204-011-1099-2>.
- [20] Z.-Q. Liu, Y.-G. Li, H.-I. Zhang, IFC-based integration tool for supporting information exchange from architectural model to structural model, *J. Cent. S. Univ. Technol.* 17 (6) (2010) 1344–1350.
- [21] X. Wang, Z.P. Cui, Q.L. Zhang, H.Z. Yang, Creating structural analysis model from IFC-based structural model, *Adv. Mater. Res.* 712–715 (2013) 901–904. www.scientific.net/AMR.712-715.901.
- [22] X.-Y. Zhang, Z.-Z. Hu, H.-W. Wang, M. Kassem, "An industry foundation classes (IFC) web-based approach and platform for Bi-directional conversion of structural analysis models, in: *2014 International Conference on Computing in Civil and Building Engineering*, ASCE, Reston, VA, 2014, pp. 390–397, <https://doi.org/10.1061/.1061/>.
- [23] Z.Z. Hu, X.Y. Zhang, H.W. Wang, M. Kassem, "Improving interoperability between architectural and structural design models: an industry foundation classes-based

- approach with web-based tools, *Autom. ConStruct.* 66 (2016) 29–42, <https://doi.org/10.1016/j.autcon.02.001>.
- [24] I.J. Ramaji, A.M. Memari, Interpreted information exchange: systematic approach for BIM to engineering analysis information transformations, *J. Comput. Civ. Eng.* 30 (6) (2016) 230–246. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000591](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000591).
- [25] M. Muller, A. Garbers, F. Esmanioto, N. Huber, E.R. Loures, O. Canciglieri, "Data interoperability assessment through IFC for BIM in structural design – a five-year gap analysis, *J. Civ. Eng. Manag.* 23 (7) (2017) 943–954, 10.3846/1341850.
- [26] R. Ren, J. Zhang, H.N. Dib, BIM interoperability for structural analysis, in: *Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 2018*, pp. 470–479, <https://doi.org/10.1061/>.
- [27] A. Kiviniemi, J. Karlshøj, V. Tarandi, H. Bell, O.J. Karud, Review of the Development and Implementation of IFC Compatible BIM, 2008. <https://orbit.dtu.dk/files/1011145/230871.230883>. (Accessed 30 September 2019).
- [28] J.D. Kindrick, J.A. Sauter, R.S. Matthews, Improving conformance and interoperability testing, *StandardView* 4 (1) (1996) 61–68, <https://doi.org/10.1145/230871.230883>.
- [29] Y.-C. Lee, C.M. Eastman, J.-K. Lee, Validations for ensuring the interoperability of data exchange of a building information model, *Autom. ConStruct.* 58 (2015) 176–195, <https://doi.org/10.1016/j.autcon.2015.07.010>.
- [30] H. Ma, K.M.E. Ha, C.K.J. Chung, R. Amor, Testing semantic interoperability, in: *Joint International Conference on Computing and Decision Making in Civil and Building Engineering, JICCDMCBE*, 2006, pp. 1216–1225, <https://cs.cmu.edu/~trebor/papers/2006/1216-1225.pdf>. (Accessed 30 September 2019).
- [31] G. Lee, J. Won, S. Ham, Y. Shin, Metrics for quantifying the similarities and differences between IFC files, *J. Comput. Civ. Eng.* 25 (2) (2011) 172–181, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000077](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000077).
- [32] buildingSMART, buildingSMART International home of openBIM - home Page. <http://www.buildingsmart.org>, 2019. (Accessed 1 October 2019).
- [33] T. Liebich, R. Steinmann, Ch Groome, K. Linhard, K.H. Häfele, A. Geiger, K. Hausknecht, IFC2x3 Coordination View Version 2.0 Certification – Workflow Description, 2010. https://standards.buildingsmart.org/documents/BSI_IFC_Workflow_CV-V2.0_Draft1.1.pdf. (Accessed 30 September 2019).
- [34] I. Kovacic, M. Filzmoser, Key success factors of collaborative planning processes, *Eng. Proj. Organ. J.* 4 (4) (2014) 154–164, 10.1080/10.1080/.
- [35] A. Borrmann, M. König, C. Koch, J. Beetz, *Building Information Modeling*, Springer Fachmedien Wiesbaden, Wiesbaden, Germany, 2015.
- [36] Mirza, Nacey Research Ltd, *The Architectural Profession in Europe 2018*, 2019. https://ace-cae.eu/fileadmin/New_Upload/.Publications/___EN_FN.pdf. (Accessed 1 October 2019).
- [37] buildingSMART, buildingSMART International home of openBIM - technical Resources. <http://www.buildingsmart-tech.org>, 2018. (Accessed 27 September 2018).
- [38] L.D.W. Thomas, E. Autio, The fifth facet: the ecosystem as an organizational field, in: *Proc., DRUID Society Conference*, 2014. <https://www.druid.ac.uk/acc/vqx0vto0qkn5vg3uiqafb24.pdf>. (Accessed 1 October 2019).
- [39] buildingSMART, buildingSMART International home of openBIM - technical Resources. <https://technical.buildingsmart.org>, 2019. (Accessed 27 September 2018).
- [40] I. Kovacic, D. Vasilescu, M. Filzmoser, R. Suppin, L. Oberwinter, BIM in teaching - lessons learned from exploratory study." *Organization, Technol. Manag. Construct.* 3 (7) (2015) 1358–1366.
- [41] G. Sibenik, I. Kovacic, Proposal for a classification system supporting interdisciplinary model exchange between architecture and structural analysis, in: *Proc., CIB World Building Congress 2019, The Hong Kong Polytechnic University, Hong Kong SAR, China, 2019*.
- [42] R. Agarwal, S. Chandrasekaran, M. Sridhar, Imagining construction's digital future, *Blog post*, June, <https://mckinsey.com/industries/capital-projects-and-infrastructure/insights/future>, 2016. (Accessed 30 September 2019).
- [43] ISO, International Organization for Standardization, ISO 29481-1:2016: Building Information Models - Information Delivery Manual - Part 1: Methodology and Format, ISO, Geneva, Switzerland, 2016.
- [44] ISO, International Organization for Standardization, ISO 29481-2:2012: Building Information Models - Information Delivery Manual - Part 2: Interaction Framework, ISO, Geneva, Switzerland, 2012.
- [45] ISO, International Organization for Standardization, ISO 12006-3:2007: Building Construction - Organization of Information about Construction Works - Part 3: Framework for Object-Oriented Information, ISO, Geneva, Switzerland, 2007.
- [46] ISO, International Organization for Standardization, ISO 16739-1:2013: Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries, ISO, Geneva, Switzerland, 2013.
- [47] Allplan, Computer Software, Allplan GmbH, Munich, Germany, 2016 (2015).
- [48] Revit, Computer Software, Autodesk Inc, San Rafael, United States of America, 2016.
- [49] D. Holzer, Y. Tengono, S. Downing, Developing a framework for linking design intelligence from multiple professions in the AEC industry, in: A. Dong, A.V. Moere, J.S. Gero (Eds.), *Computer-Aided Architectural Design Futures (CAADFutures) 2007: Proceedings of the 12th International CAADFutures Conference*, Springer, Dordrecht, Netherlands, 2007, pp. 303–316.
- [50] buildingSMART, Streamline IDM/MVD/Certification/Validation. http://iug.eu/resources/abu-dhabi-iug-meeting/IDMC_016_2.pdf/view, 2011. (Accessed 23 April 2019).
- [51] ArchiCAD 19, Computer Software, Graphisoft, Budapest, Hungary, 2015.
- [52] RFEM 5.05, Computer Software, Dlubal Software GmbH, Tiefenbach, Germany, 2015.
- [53] SCIA Engineer 15.3, Computer Software, SCIA nv, Herk-de-Stad, Belgium, 2015.