Advancements and Gaps in Information Compliance Checking (ICC) automation with Information Delivery Specification (IDS)

Tomo CEROVSEK, tomo.cerovsek@fgg.uni-lj.si
Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia

Mohamed OMAR, mohamed.omar@hilti.com
BIM Competence Center, Hilti Netherlands B.V., Rotterdam, The Netherlands

Abstract
The paper assesses the information delivery specification (IDS) and tries to identify functional and non-functional aspects of IDS implementations. IDS is becoming an essential tool for the consistency, accuracy, completeness, and reliability of project information throughout BIM project delivery.

Through a review of IDS, its implementations in selected solutions, we use a process centred method for the identification of advancements and gaps that may serve as input to continual improvement of project information requirements management. We also propose criteria for the assessment of IDS, we explore checkability of requirements, and expose the potential of historical data from past compliance checks. The results were obtained based on real-world case study with in-depth analysis of practical applicability of automated information compliance checking, in addition, we analysed IFC mapping issues and workflows, and summarised findings in the SWOT matrix.

The research may contribute to a better management of information requirements and to the automation of compliance checking. The results offer some valuable insights for practitioners and developers or researchers working on the quality and reliability of information in BIM models.

Keywords: information requirements, compliance checking, IDS process model, openBIM, IFC

1 Introduction
As BIM implementations on projects mature, so do grow requirements for alphanumeric information in models. Furthermore, rising numbers and complexity of BIM projects, especially very large scale building and infrastructure projects, amplify the need for advanced management of requirements. Collectively information requirements result in higher demand for automated model information compliance checks (ICC). Such ICC may include, but is not limited to model quality assurance, data validation, code compliance checking, field definition, values and consistency checks (Hjelseth, 2009, 2015, 2016; Schwabe et al., 2019). Note that in this paper we use the terms *semantic* and *alphanumeric information* interchangeably.

The analysis shows that building projects produce huge amount of data (Sacks et al., 2018), a larger project may have 10.000s of document pages and over 1.000 drawings, which are now BIM deliverables (i.e. data on drawings comes from semantic information attached to BIM elements).

Semantic information in architecture and engineering enhances usability, management, and integrity of project information allows engineers to understand and utilize geometrical and non-geometrical data more effectively across domain and project stages, standard semantic information also makes diverse applications interoperable. In addition, semantic information enables information retrieval, ensures consistency and accuracy, and supports automation of compliance checking against standards and regulations. Semantic information also facilitates collection, analysis and utilization of historical BIM data across projects that can improve processes. Semantic information plays a pivotal role in collaboration by allowing different project teams to interpret and integrate data seamlessly, ultimately driving more informed decision-making and fostering innovation in engineering projects. More on project information value and management in (Cerovsek, 2024).

A significant challenge in model checking are also non-functional software requirements such as very limited flexibility of available tools. Many of existing tools solely rely on "black-box" approaches, utilizing hard-coded rules that lack adaptability (Preidel and Borrmann, 2016). This inflexibility highlights the need for a "white-box" workflows that bring an openness into the compliance checking along with a sufficient level of customization.

The three essential drivers for the development and advancement of “white box” model of information requirements compliance checking may be summarized by:

- *Transparency*: Users can understand how the checking process is conducted, which enhances confidence in the results, repeatability and facilitates the troubleshooting.
- *Customizability*: The ability to customize the checking process enables users to reuse and tailor needs to a specific project, ensuring relevant and meaningful requirements.
- *Extensibility*: It allows for the integration of additional checks and rules, making it adaptable to evolving requirements across projects portfolio and growing standards.

In this paper, we explore the possibilities offered by information delivery specification (IDS), which is one of a more recent standards developed by buildingSMART that addresses inflexibility of existing tools, and enables desired "white-box" workflow that also supports openBIM.

IDS can significantly advance and enhance the efficiency and effectiveness of communication and unification of information requirements and offers new ways of defining model requirements (Eichler et al 2024). The IDS is a machine-interpretable definition of alphanumeric Exchange Information Requirements (EIR) in the XML format that has predefined structure, which is controlled by the IDS XML schema. With IDS we can specify on a project level which semantic information should be attached to which objects in BIM model at particular phase of a project (van Berlo *et al.*, n.d.).

IDS ensures validation of IFC to clients, modellers, and other stakeholders. We may expect that IDS will become a core reference document that will constitute contractual agreement for BIM project delivery. IDS allows for the development and customization of requirements of specific project. The IDS provides a solution for reliable and predictable workflows in project information exchange. The illustration on Figure 1 is a generic process in IDEF0 that shows how IDS functions in the process of specification of requirements, their exchange and checking. The process model bellow is generic, this means that the process is independent of a specific IDS-enabled tools and standards. A tool and standard specific process model is presented later in the concluding part of the paper.

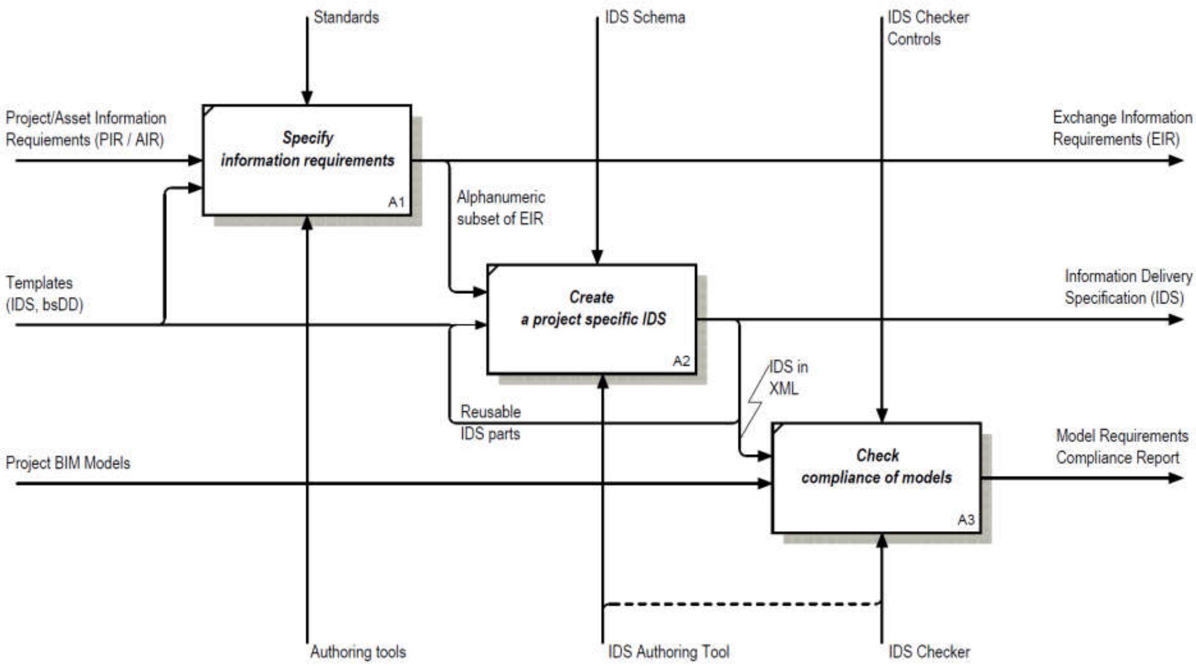


Figure 1. Generic IDEF0 process model illustrating the application of Information Delivery Specification (IDS)

1.1 The process of requirements management using IDS

The compliance checking using IDS may be divided into three main activities:

A1: Specify Information Requirements:
- The Project and Asset Information Requirements (PIR & AIR) provide input to the information exchange requirements (EIR), which also encompass alphanumeric part.
- Formal (e.g. ISO 19650-1, ISO 19650-2) and informal standards (e.g. office, project) also control the requirements for discipline models may be used on particular project.

A2: Create a project specific IDS
- The inputs to the project IDS are alphanumeric subsets of EIR for each discipline, which may be taken from the pre-existing product data templates, bSDD (buildingSMART Data Dictionaries), or from previously developed parts of IDS in the XML format.
- The goal of the activity is to create the IDS compliant rules in a structured document in the XML format that prescribed by the IDS schema developed by buildingSMART.

A3: Check compliance of the models
- The input to the compliance check is the IDS in XML and the project BIM model in the IFC format, on which IDS-based information compliance checks can be performed.
- The BIM Checker tool generates reports highlighting the issues and discrepancies based on the compliance checks of the BIM model. The function of IDS Checker is to identify and communicate any deviations from the specified information requirements in the IDS.

There are two main mechanisms that support the development and use of IDS, i.e. IDS Authoring Tool and IDS Checker. IDS Authoring Tool could be build using any XML editor that can create a valid XML based on IDS schema, and can be checked by IDS-Audit tool (or official IDS authoring tools listed on buildingSMART). IDS Checker must parse IFC and IDS and run checks of IFC based on IDS. After the review, the compliance check reports help to identify and resolve any issues found in the BIM model. It may be also necessary to iterate through the process, making necessary modifications to the IDS rules or the BIM model until all requirements are met.

1.2 Research Focus

In this paper, the research focuses on the implementation of openBIM workflow which focuses on the utilization of the buildingSMART format of IDS for information requirement checking and exchange with the utilization of IFC and BCF for other forms of information exchange (Table 1).

Table 1. Overview of the research focus

Research focus	Description of the purpose
Checkability of Requirements Case Study	Checkability denotes the ability to map parts of IER to IDS XML. Practical insights on checkability were obtained through a larger use case and documented best practices. The focus here was to identify characteristic types of requirements and their translation into the IDS logic.
Functional Non-Functional IDS software requirements	Understanding the distinction between functional and non-functional requirements helps in development, deployment, and maintenance of data management systems. A good balance of respective requirements serves as the foundation for creating robust, efficient, and user-friendly data management solutions.
Advancements and gaps for the automation of ICC	Exemplifying advancements and identification of current gaps in information requirement checking is instrumental for the continual improvement of both, IDS schema development and its implementation in different software package along with diversification of solutions.

In the follow up section we briefly outline the methods that were used for the research focus.

2 Methods

The checkability of requirements can be divided into: **explicit**, **implicit**, and **unsupported**. Methods for testing of checkability using IDS can be borrowed from the codification of standards, e.g. (Fenves 1995; Hjelseth 2015), though, IDS was not designed for encoding of complex rules. For example, RASE (Requirement, Applicability, Selection, and Exception) identifies structured information that can be mapped into sophisticated rule languages (Hjelseth and Nisbet, 2011; Schwabe et al., 2016, 2019; Solihin and Eastman, 2015), and it can be used to assess checkability; viable mapping of EIR into applicability, requirements and restrictions of IDS schema (Figure 2).

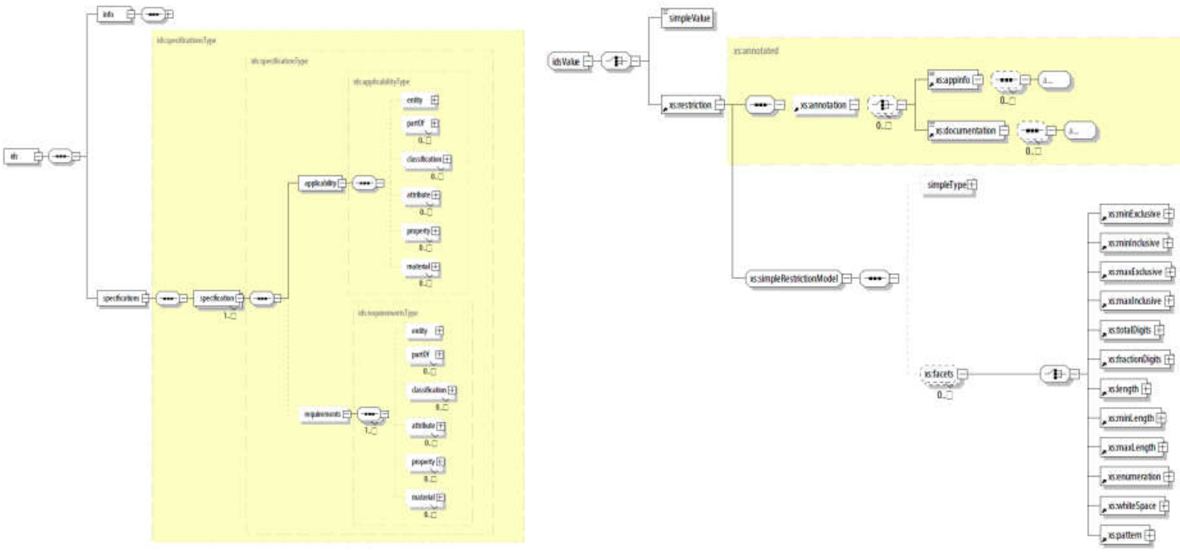


Figure 2. Information Delivery Specification (IDS) XSD Schema

Explicit checks are straightforward and can be easily mapped to applicabilityType (e.g. entity, classification, attribute, property or material) with corresponding requirementsType and facets of restrictions (Figure 2). Implicit checks involve relational, derived values and conditional logic that must be inferred from the data and translated into the IDS logic. Implicit checks may be more challenging to define and verify because they depend on understanding of the relationships and interactions between model entities. The not-supported checks relate to the requirements that are technically not possible to check either due to the scope of the IDS or its temporal limitations.

Note that from the point of view of functional requirements, model processing, reporting, implementation, or execution of regular expressions in the model checker may differ between IDS software implementations. In addition, there may be discrepancies between actual content in the authoring environment and the exported content in the IFC format due to different structural and non-structural mappings – extraction and translation of the native content, therefore it is important to assess the issues related to mapping and eventual use of checks in authoring environment. In addition, IDS implementation, both IDS Builders and IDS Checkers respectively, needs to be tested against non-functional requirements as suggested in provisional Table 2, which is beyond the scope of the paper, but might provide valuable input for future tests.

Table 2. Non-functional requirements for the implementation of Information Delivery Specification (IDS)

Criteria	Description
Usability	User friendliness and responsive interaction, intuitive interpretation of inputs/outputs.
Performance	The speed of loading times, interface responsiveness, efficiency of data integrity checks.
Scalability	No non-proportional performance degradation due to project size or data complexity.
Maintainability	Measure for flexible adaptation to specific project needs and ease of IDS maintenance.
Reliability	Demonstrate consistency/repeatability of checks and data processing stability.

3 Results

In this section we first analyse the three checkability categories (section 3.1), we show some examples of issues due to mapping to IFC (section 3.2) that stem from authoring environment and can affect the results of checks, and last but not least we analyse example of specific IDS implementation (section 3.3). The checkability was tested on a case study that served as testbed for mappings of EIR into a set of explicit and implicit checks using the IDS logic (see Table 3).

Table 3. Examples of EIR mappings (source: a case study project of 15.000 sq. m - analysed by Omar (2023))

Rule	Checkability	Example of translation into IDS logic
The Levels' names are according to RP Standard Levels & Grids Chapter	Explicit	The model MUST contain entities that have: • IFC class IFCBUILDINGSTOREY that MEET the following requirements: • MUST HAVE attribute Name matching the pattern (Streha|ST-Streha|Nadstropje-4.|ST-Nadstropje-4.|Nadstropje-3.-Staro|Nadstropje-3.|ST-Nadstropje-3.|Nadstropje-2.|ST-Nadstropje-2.|Nadstropje-1.|ST-Nadstropje-1.|Pritlicje|ST-Pritlicje|Klet-1.|ST-Klet-1.|Klet-2.|ST-Temelji|Medetaža|Mezanin|Tehnižna etaža|Mansarda)
Material names are according to the standard	Explicit	The model MUST contain entities that have • IFC class IFCCOVERING (or: IFCDOOR; IFCFURNISHINGELEMENT; IFCMEMBER; IFCPROJECTIONELEMENT; IFCPLATE; IFCREINFORCINGELEMENT; IFCROOF; IFCSLAB; IFCSTAIR; IFCSTAIRFLIGHT; IFCWALL; IFCWALLEXTENDEDELEMENT; IFCWINDOW; IFCBEAM; IFCBUILDINGELEMENTPART; IFCCHIMNEY; IFCCOLUMN; IFCPILE; IFCPILECONNECTION; IFCFLOWTERMINAL; IFCTRIMMEDCURVE; IFCFURNISHINGELEMENTTYPE; IFCBUILDINGELEMENTPROXY) that MEET the following requirements • MUST HAVE material matching the pattern RP-.*
Walls, Ceilings, Curtain Walls, Doors, Windows, and Floors have defined (at least) one of the parameters DT_Function_Ext or DT_Function_Int	Explicit	The model MUST contain entities that have • IFC class IFCWALL (or: IFCCOVERING; IFCCURTAINWALL; IFCDOOR; IFCWINDOW; IFCSLAB) that MEET the following requirements • MUST HAVE property DT_Function_Ext (or: DT_Function_Int) of PSet RP_Custom (IFCBOOLEAN) The model MUST contain entities that have • property DT_Function_Ext of PSet RP_Custom (IFCBOOLEAN) = TRUE that MEET the following requirements • MUST NOT HAVE property DT_Function_Int of PSet RP_Custom (IFCBOOLEAN) = TRUE The model MUST contain entities that have • property DT_Functio n_Ext of PSet RP_Custom (IFCBOOLEAN) = FALSE that MEET the following requirements • MUST NOT HAVE property DT_Function_Int of PSet RP_Custom (IFCBOOLEAN) = FALSE
Non-load bearing walls should be modelled separetly	Implicit	The model MUST NOT contain entities that have • IFC class IFCWALL • attribute Name matching the pattern .*[Aa][Rr][Cc].* that MEET the following requirements • MUST HAVE property LoadBearing of PSet Pset_WallCommon (IFCBOOLEAN) = TRUE
Every space is set with a proper boundary height	Implicit	The model MUST contain entities that have • IFC class IFCSPACE that MEET the following requirements • MUST NOT HAVE property Height of PSet Qto_SpaceBaseQuantities (IFCLENGTHMEASURE) > 4.5
Every Room Number is unique	Unsupported	Workaround: pre-processing of IFC and automated generation of IDS
Limited number of instances of a type	Unsupported	Workaround: pre-processing of IFC and automated generation of IDS

3.1 Checkability

3.1.1 Explicitly Checkable Requirements

The IDS excels in checking of naming conventions within the model, adhering to organizational or company/project-specific standards. Regular expressions features (Friedl, 2006) can be easily employed in applicability and requirement blockers to filter and define versatile requirements, offering flexibility to target specific elements and very complex naming rules. However, verification can encompass a wide range of entities, data types and classifications requiring repetitive definitions.

IDS can ensure completeness of information in the model, so that required properties are included in the appropriate property sets. For example: Walls, Ceilings, Curtain Walls, Doors, Windows, and Floors have defined at least one of the following parameters DT_Function_Ext or DT_Function_Int to define whether they are exterior or interior elements. Moreover, to ensure sufficient reliability of provided information, extra specifications can be created to ensure that elements do not have contradictory or illogical semantic information (true-true values for both interior and exterior parameters at the same time, or the opposite of having false-false values). It may be limiting that the entity facet should refer to a instantiated and not abstract entity.

3.1.2 Implicitly Checkable Requirements

Some values can not be verified directly to ensure the specific information requirements are fulfilled. Using IDS, it can be challenging to explicitly check a rule like: "All structural walls and structural columns should be separated by structural levels and not modelled based on architectural or supporting levels". As in the IFC schema, the direct relationship between an IfcWall and its containing storey is not stored as an attribute within the IfcWall itself. Instead, this relationship is established through the IfcRelContainedInSpatialStructureentity. The IfcWall does not have a specific attribute that directly holds the storey information. Instead, you need to follow the relationship defined by IfcRelContainedInSpatialStructure. There are several ways to implicitly validate if the model complies to the above-mentioned rule. The first approach can be implemented by ensuring that all stores in the IFC structural model are at structural level, and by verifying the attribute name of IfcBuildingStorey. Then, using PartOf Facet, similarly, another specification can be implemented to ensure that all structural columns are within the spatial structure container. The second possible logic can be using a predefined Property in the modelling / IFC preparation phase that captures Information about building Storey. In that case, IDS can be used to validate the model explicitly. However, there a is risk of relying on manual information because it can lead to contradictory information across different stages of modelling.

3.1.3 Unsupported Checks of Requirements

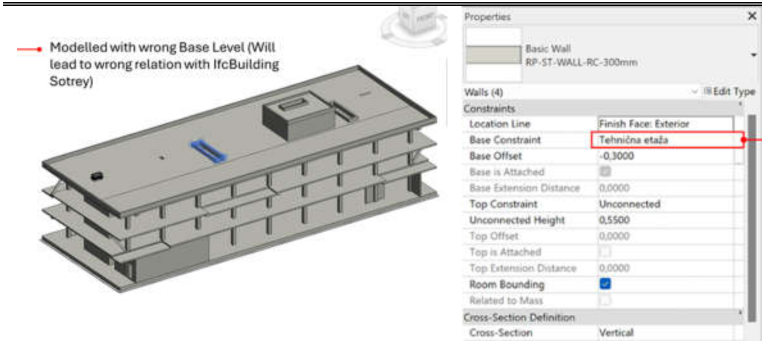
Though IDS is not designed to check the compliance or level of detail of 2D or 3D geometry existing alphanumeric information and QTO-included data can implicitly verify geometry it would be useful to be able to compare some properties. For instance, consider the rule that every room should have a proper boundary height. By utilizing the Height property included in the QTO property set, a specification can be created to ensure that all spaces do not exceed a certain height, such as 4.5 meters, according to project-specific requirements.

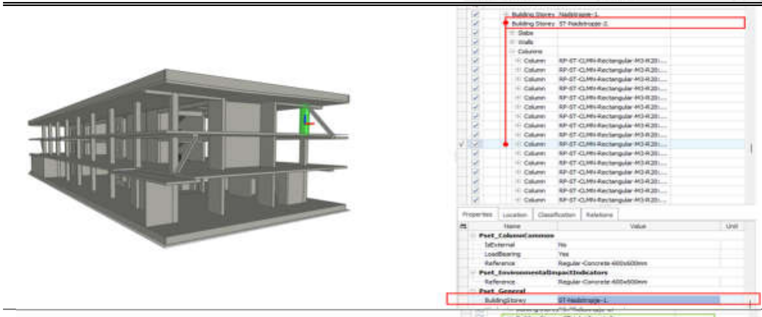
Another example involves checking the file name of the model. In the IFC schema, there is no specific class designed to store the filename of the IFC file itself. Instead, metadata about the file, including the file name, is typically stored in the IFC header section, which is not part of the IFC schema entities. Renaming the IFC file itself will not automatically update or reflect any changes in the IfcProject entity's Name attribute or any other attributes within the IFC data. Therefore, the file name and the project name are managed independently, and only the project name can be verified using IDS.

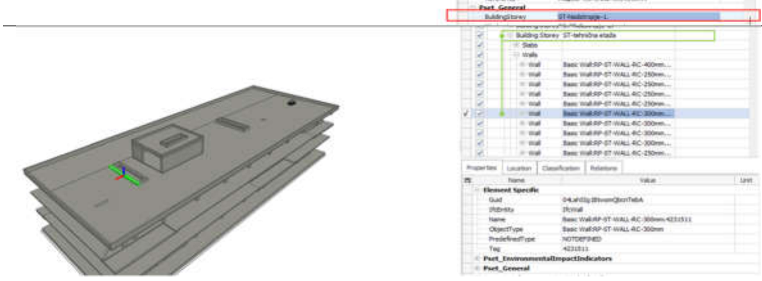
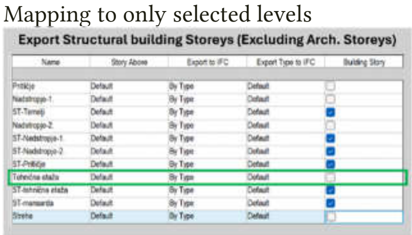
3.2 Mapping to IFC issues

In this subsection we show some examples of mapping issues from native file format to the IFC. Information mapping is very important as built-in or custom properties might not be transferred correctly, leading to incomplete IFC models. On the other hand IFC export can be adjusted to match some basic requirements, e.g. storey containers. For example, if some elements were not following the modelling guidelines, it can be exported IFC file to look correct (see Table 4).

Table 4. IFC mapping issues relevant for information compliance Checking

<table>
<tr><th>Graphical representation</th><th>Mapping issue</th></tr>
<tr><td>


</td><td>
The mapping from native authoring environment to IFC may be executed in several ways. The element category can be changed, property values are derived from geometry (e.g. QTO) and may not match actual IFC geometry,
Example of incorrectly modelled wall hosts wrong base constraint. During the export to IFC there more option of mapping elements to stories:
All levels included
Only selected levels included
</td></tr>
<tr><td>

</td><td>
Example of export of all levels


IFC has wrong levels
</td></tr>
<tr><td>

</td><td>
Mapping to only selected levels


IFC has correct levels
</td></tr>
</table>

3.3 Specific implementation of a process

At the time of writing, there are 17 IDS enabled tools for IDS authoring and checking that are listed on buildingSMART. Figure 3 illustrates implementation of IDS in Plannerly that takes care of mapping requirements into IDS and exports IDS XML for use in Solibri Model Checker.

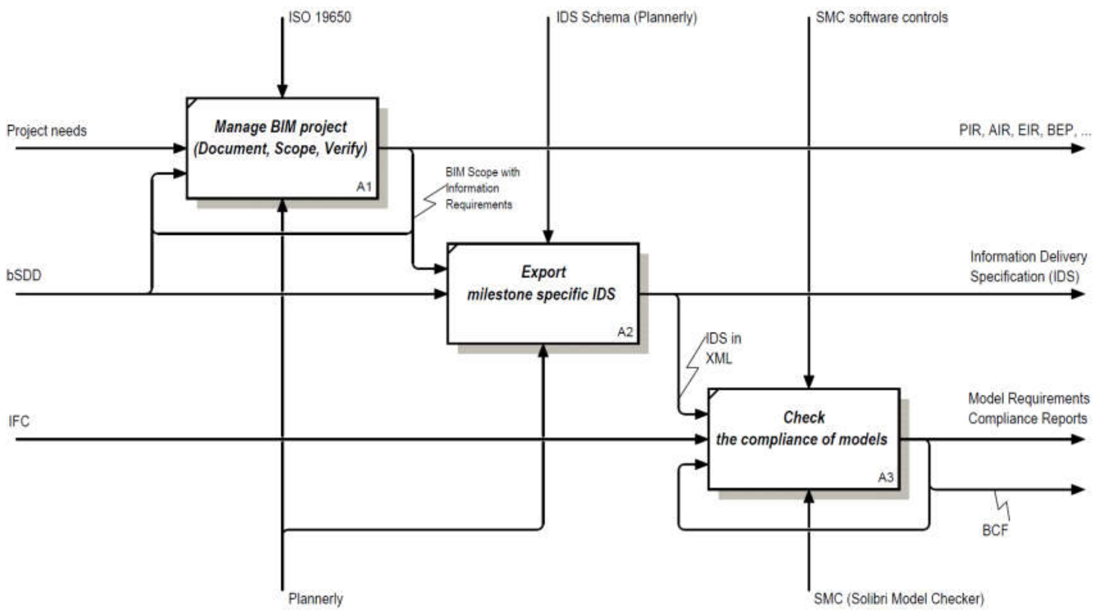


Figure 3. The use of Information Delivery Specification (IDS) implemented in Plannerly and used in Solibri Model Checker

4 Discussion

The discussion is grouped into two main parts. First, we identify limitations of current IDS schema and how this causes barriers to map EIR to IDS. Second, we provide a summary of advancements and gaps in IDS through a brief SWOT analysis and conclude with a future outlook.

4.1 IDS schema limitations

The initial iteration of IDS focuses on the common basic checks and relationships in IFC that are valid for all disciplines. However, IDS cannot be equipped to handle Geometric checks and validations, and domain-specific requirements that involve complex interdependencies between elements (van Berlo *et al.*, 2019; buildingSMART International, 2024; Tomczak *et al.*, 2022) . These checks are crucial to ensure design compliance and construction feasibility but are not possible within the current IDS capabilities. This requires additional tools and manual verification processes, which can be time-consuming and error-prone. If the protocol is put in place this can be put together in a seamless workflow through which semantic information could be generated to identify geometric inconsistencies.

IDS can't use the concept of "rules" to check the model against specific design requirements(van Berlo *et al.*, 2019). It cannot automatically verify compliance with specific design rules, building codes, or standards. This limitation means that IDS cannot be solely relied upon for comprehensive model validation, especially for projects with stringent regulatory or design requirements. Additional rule-based checking software is required.

Another limitation can be happening in structuring the IDS file from lack of knowledge of IDS Schema in accordance with IFC schema. As an example, while utilizing the Entity facet provides a convenient means to navigate directly to model elements, limitations arise when abstract entities are used. For instance, IfcBuildingElement cannot be utilized directly as it remains uninstantiated. Consequently, all included entities such as IfcWall or IfcBeam must be explicitly defined within the applicability section. This ensures precise identification and handling of relevant elements within the model, highlighting the necessity of a thorough understanding of the IDS schema.

IDS may not cover all aspects of data definition and information requirements specified in the bSDD. While IDS focuses on information delivery specifications for specific projects or phases, bSDD provides a comprehensive data dictionary covering a wide range of buildingSMART standards and data definitions. IDS and bsDD may evolve and be updated independently, leading to potential discrepancies or inconsistencies over time.

The compatibility between different versions of IFC and IDS can be challenging over time, mainly when relying on historical information, and expected lDS libraries that created previously on older versions of IDS. Organizations may need to revise their IDS requirements and validation processes to accommodate changes in IFC and IDS versions.

IDS is currently under extensive development by buildingSMART absorbing all inputs and feedback from Academics, Professionals and pioneers in the BIM industry which means continuous development and IDS schema updates. This means the possibility of wider functionalities of IDS but also means changes in schema and the risk of corruption of currently and previously developed specifications. That is why End-users including Clients, designers and contractors need to audit their IDS files in accordance with the latest releases of IDS for better implementation of the information requirement exchange process. In this context, the paper proposes a simple auditing workflow for end-users to detect the current bugs and adjust IDS files.

The auditing process can be done manually with the same concept of creating IDS using an XML editor. However, a manual auditing process can be challenging detect and find the bugs, especially in long IDS files that are thousands of lines. Thus, using any of the approved IDS validators can be potentially an advantage to get where the errors are located with referencing to the ID schema. Another possible option is to use an advanced IDS Editor Tool which contains audit functions, such as usBIM.IDSEditor and able to detect and solve the IDS, with the advantage of upgrading the IDS to the latest version the tool is using. As illustrated on Figure 3, Plannerly maps the information requirements that are attached to the scope into the IDS in the XML format, which can then be used by other IDS Checkers, in particular case SMC (Solibri Model Checker).

4.2 Concluding SWOT analysis of IDS

The table below presents a analysis of the strengths, weaknesses, opportunities, and threats associated with the advancements and existing gaps in the development of IDS. These insights are framed as observations that may provide input for the practical implementation and eventually influence the future evolution of IDS.

Table 5. IDS SWOT Matrix

IDS	Description
S - Strengths	• Enables automated information compliance checking using a standard format • Provides exact definition of information requirements for communication between appointing and appointed parties for each major milestone. • Enhances trust, collaboration leverage and the power of consistent models. • White-box approach using openBIM formats with a sufficient flexibility level. • IDS is both computer interpretable and human-readable file, which makes the requirements easy to communicate and use for the compliance checking. • Enables checking of field name definitions, data types, values & assignments. • Extensive support for checking of IFC attributes (“Name”, “Description”), IFCtype, properties (“Pset”), quantity, classifications, composition, and materials, and requirements constraints facets are powerful. • Supports regular expressions that allow for definition of complex patterns.
W - Weaknesses	• Some checks that require cross references and counting are not supported. • Though geometry checks are out of the IDS scope, it would be highly desirable to better support alphanumeric geometry specific checks. • Current lack of support for authoring environments (for instance, RVT), checks may be working fine on IFC, but inside authoring environment would be quicker – no need to export and import, IFCs can be tweaked outside authoring environment and might be inconsistent with the native project file. • Not supported exchange of issues using BCF is in some openBIM software, the issues that are identified via IDS are much harder to resolve without BCF. • Hard to handle exceptions even if the such a check is within 1st order logic.
O - Opportunities	• Many software vendors will be supporting IDS, in general IDS Checker could use a native format and IFC, which would allow for to assess mapping to IFC. • Introduction of aliases (or substation list) would ease IDS authoring. • Possibility to define constants that are project specific would simplify IDS, for example if there are several requirements that include requirements that change for each project, e.g. project specific names, originators codes, etc. • Implement checks equivalent to the unique and distinct SQL statements. • Make use of historical data from several projects, including BCF exchanges to generate IDS that may be specific to a particular context or project type. • Chance to make templates for IDS use for specific projects or/and BIM uses.
T - Threats	• Poor information exchange quality (in exporting IFC from the repository) may lead to information loss and wrong results and irrelevant IDS checks. • Over specification may lead to the alphanumeric information overflow and redundant definitions that are hard to comprehend and hard to be managed. • Additional information requirements might be defined and cause not necessary semantic information to compensate current deficiencies of IDS.

The authors are working on the analysis of information requirements using ontology, and will propose an adapted Information compliance checking process with applicability of the potential workflows inside and outside authoring tools.

Acknowledgements

We gratefully acknowledge Protim, Rzisnik & Perc, Architects and Engineers for their collaboration in the research, for validation and insights from real-life scenarios. This research was partly supported by the project 101033743—SEEtheSkills—H2020-LC-SC3-2018-2019-2020/H2020-LC-SC3-EE-2020-2.

References

van Berlo, L., Drogemuller, R. and Omrani, S. (2024). “What is Information Delivery Specification (IDS)”, Information Delivery Specification (IDS), available at: https://www.buildingsmart.org/what-is-information-delivery-specification-ids/ (accessed 4 May 2024).

van Berlo, L., Willems, P. and Pauwels, P. (2019), “Creating Information Delivery Specifications using Linked Data”, Proceedings of the 36th CIB W78 2019 Conference, presented at the Advances in ICT in Design, Construction and Management in Architecture, Engineering, Construction and Operations (AECO), pp. 647–660.

buildingSMART International. (2024). “Information Delivery Specifications (IDS)”, Information Delivery Specification (IDS), available at: https://technical.buildingsmart.org/projects/information-delivery-specification-ids/ (accessed 4 May 2024a).

buildingSMART International. (2024). “2024 IDS validation 1.0”, 2024 IDS Validation 1.0, available at: https://technical.buildingsmart.org/projects/2024-ids-validation-1-0/ (accessed 10 May 2024b).

buildingSMART International. (2024). “2024 IDS 1.0 Schema”, 2024 IDS 1.0 Schema, available at: https://standards.buildingsmart.org/IDS/1.0/ids.xsd (accessed 10 May 2024c).

_______________ (2024). BIM Cube: Information Management for Digital Construction. Wiley-Blackwell. ISBN 978-1-119-24290-1. Estimated publ. date 2024.

Dias, C. (2024). “IDS Converter”, Https://Idsconverter.Streamlit.App/, available at: (accessed 4 May 2024).

Eichler, C. C., Schranz, C., Krischmann, T., Urban, H., Hopferwieser, M., & Fischer, S. (2024). BIMcert Handbook: Basic Knowledge openBIM. Edition 2024, buildingSMART, p 224.

Fenves, S.J., Garrett, J.H., Kiliccote, H., Law, K.H. and Reed2, K.A. (1995), “Computer Representations of Design Standards and Building Codes: U.S. Perspective”, The International Journal of Construction Information Technology, Vol. 3 No. 1, pp. 13–34.

Hjelseth, E. (2009), “Foundation for development of computable rules”, Managing IT in Construction/Managing Construction for Tomorrow, pp. 257–266.

Hjelseth, E. (2015), “Foundations for BIM-based model checking systems Transforming regulations into computable rules in BIM-based model checking systems Prinsipper for BIM basert modell sjekking Omforming av bestemmelser i regelverket til digitaliserte regler i BIM-baserte systemer for modellsjekking”, Vol. 2015, p. 54.

Hjelseth, E. (2016), “Classification of BIM-based model checking concepts”, ITcon, Vol. 21 No. CIB W78 2015 Special track on Compliance Checking, pp. 354–369.

Hjelseth, E. and Nisbet, N.N. (2011), “Capturing normative constraints by use of the semantic mark-up RASE methodology”, CIBW78 2011 28th International Conference-Applications of IT in the AEC Industry.

[ISO 19650-1]. (2018), Organization and Digitization of Information about Buildings and Civil Engineering Works, Including Building Information Modelling (BIM) - Information Management Using Building Information Modelling - Part 1: Concepts and Principles, Vol. 1, CEN-CENELEC Management Centre.

[ISO 19650-2]. (2018), Organization and Digitization of Information about Buildings and Civil Engineering Works, Including Building Information Modelling (BIM) - Information Management Using Building Information Modelling - Part 2: Delivery Phase of the Assets.

Omar, M. (2023), Automated BIM-based model checking workflows with exchange information requirements: master thesis no.: 283/II. GR-BIM [online]. Master’s thesis. Ljubljana. DOI: 10.13140/RG.2.2.22772.56960

Preidel, C. and Borrmann, A. (2016), “Towards code compliance checking on the basis of a visual programming language”, Journal of Information Technology in Construction (ITcon), Preidel & Borrmann, Vol. 21 No. CIB W78 2015 Special track on Compliance Checking, pp. 402–421.

Friedl, J. E. (2006). Mastering regular expressions. " O'Reilly Media, Inc.".Schwabe, K., König, M. and Teizer, J. (2016), “BIM Applications of Rule-Based Checking in Construction Site Layout Planning Tasks”, doi: 10.22260/ISARC2016/0026.

Sacks, R., Eastman, C., Lee, G. and Teicholz, P. (2018), BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors. Eastman, 3rd ed., Wiley.

Schwabe, K., Teizer, J. and König, M. (2019), “Applying rule-based model-checking to construction site layout planning tasks”, Automation in Construction, Elsevier, Vol. 97, pp. 205–219, doi: 10.1016/J.AUTCON.2018.10.012.

Solihin, W. and Eastman, C. (2015), “Classification of rules for automated BIM rule checking development”, Automation in Construction, Elsevier, Vol. 53, pp. 69–82, doi: 10.1016/j.autcon.2015.03.003.

Tomczak, A., Berlo, L.V., Krijnen, T., Borrmann, A. and Bolpagni, M. (2022), “A review of methods to specify information requirements in digital construction projects”, IOP Conference Series: Earth and Environmental Science, Vol. 1101 No. 9, p. 092024, doi: 10.1088/1755-1315/1101/9/092024.