XML-based Automated Information Requirement Import to a Modelling Environment

Martina Mellenthin Filardo, Paul Debus, Jürgen Melzner and Hans-Joachim Bargstädt
Bauhaus-Universität Weimar, Germany
martina.mellenthin.filardo@uni-weimar.de

Abstract. When working with the Building Information Modeling (BIM) method or with Digital Twins, information requirements (IR) regarding usage and output of the model must be defined for results to be maximally usable. Different approaches exist or are being developed to define IR, from PDF documents to XML-based specifications. This paper proposes a workflow to integrate XML-based IR into a modelling environment (ArchiCAD) prior to modelling itself, thus avoiding data losses and time-consuming changes, by making all information about the requirements available from the beginning and allowing digital support systems to aid during the design process. The workflow was tested with an Add On for ArchiCAD implementation, that imports an exemplary XML file containing project-specific requirements relevant for the design and modelling phase, as well as requirements using the Level of Information Need specification, executes the necessary functions within the ArchiCAD API, adds required properties to the designated elements, and generates the defined output (e.g., drawings and IFC file) with just one click.

1. Introduction

When working with the Building Information Modelling (BIM) method or with Digital Twins, information requirements for the model, as the data to be delivered, must be defined to avoid incomplete models or rework as well as redundant content unsuitable for subsequent phases, in accordance with BIM level 1 (Borrmann, Forster et al., 2021). These requirements touch on topics of various disciplines, comprising asset, project, exchange, and organizational information, such as described in (International Organization for Standardization, 2018). Different approaches exist to define Information Requirements (IR), such as Employer Information Requirements (EIR), Information Delivery Manual (IDM), Information Delivery Specification (IDS) or Level of Information Need (LOIN), with the latter two still under (normative) development. One of the main issues of IR is that they are still mostly handled by exchanging PDF files (Mellenthin Filardo, 2019), which still classifies as paper-based manner (Wang et al., 2012), thus falling in the BIM level 0 scope (Eastman et al., 2018, p. 15).

2. Proposed Workflow

Currently, requirement definitions are used only after the fact (ex post), meaning that they are used for checking deliverables (at data drops or milestones) and not during design. This is evidenced by the abundance in solutions for requirement definition and checking (AEC3 GmbH, 2023; dRofus AS, 2023; Plannerly - The BIM Management Platform, 2019). Ideally, meticulously defined IR would also be available for use within the modelling environment – with import prior to the modelling itself to guarantee the compliance of all subsequent (modelling) operations. This procedural change (prepositioning requirements) would avoid information losses and time-consuming, iterative changes (Mellenthin Filardo and Krischler, 2021), but demands that IR must be defined in a machine-readable format to enable import. To explore and illustrate the potentials of this application of information requirements, an Add On for such requirement integration was developed for the ArchiCAD 26 modelling software. ArchiCAD was chosen given its market adoption in the architectural community in combination with its comprehensively documented Application Programming Interface (API) and developer kit (GRAPHISOFT, 2022).

The developed tool uses an XML (Extensible Markup Language) file for import and assigns the requirements to their respective counterparts within the modelling environment, as depicted in Fig. 1. The proposed ArchiCAD Add On, called iReq, is used for two main operations: One establishes the support system, the import and (back end) assignment of requirements by the Add On. The user must choose the XML file, thereby triggering a series of functions (user dialog as well as assignment of the information defined in the XML, such as project information, location settings, disciplines, information container naming conventions, project phase(s), LOIN, export settings, etc.) to import, store, and assign the requirements within the project. These loaded requirements can then be regarded during the operational design/modelling, so the user makes informed decisions within the modelling application and does not have the additional effort of manually searching for the right (PDF) document and then searching for (and interpreting) specific requirements within the document each time. After modelling, the iReq Add On offers a second functionality, which triggers the export of the required output (described in the XML file and stored within the project), thus generating e.g., Industry Foundation Classes (IFC) files based on the defined Model View Definition (MVD), as well as drawings with the correct structure, and project-specific file naming and header information.

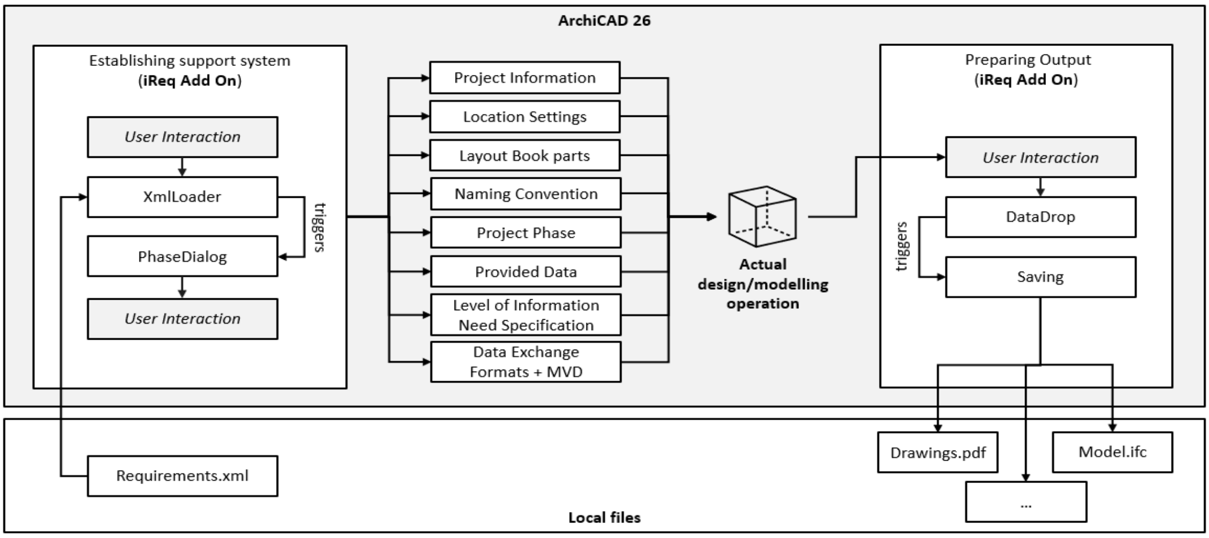

Figure 1: The proposed workflow of the Add On.

The assigned requirements used in the Add On were formulated exemplarily, based on the previously mentioned paper-based EIR documents, sometimes aided by spreadsheets, collected from the industry (Mellenthin Filardo, 2019), elaborated in chapter 3, where an overview of the state of the art regarding information requirements is also given. These existing approaches are currently limited to the use of defined requirements in the export and controlling phases. The proposed workflow of using information requirements already during the design and modelling phase is still untapped and therefore one of the key contributions of this paper. The range of market-ready solutions working with requirements within design is still limited. A solution similar to the workflow proposed in this paper is a series of Plug-Ins by dRofus AS (2023), in which the project in the modelling environment is connected to a project-specific database. Based on the requirements described in chapter 3, the implementation, with input data, functionality and results, is presented in chapter 4. The paper closes with a conclusion and outlook in chapter 5.

The added value of this requirement integration is the support for planners during the concretization of the design as well as the modelling phase, given that the requirements are not only visible within the modelling environment, but also structured and therefore, regarded in the export of the output model. Further, a reduction of exchange iterations during data drops, given the storage of complete requirements within the project and therefore skipping manual configuration at export, can be presumed. Consequently, a decrease of both manual labor and the risk of missing requirements (e.g., by overseeing specific PDF files), which culminates in a reduced management effort, is also expected. Finally, by offering a secured (and potentially more comfortable, if compared to PDF or printed requirement lists or worse, descriptive texts, which are also open to interpretation) handling of IR within the modelling environment, the proposed approach aims at using these to reinforce the adoption of the BIM method within the industry.

This workflow is understood by the authors as a design support system, similarly to the one defined by Liebich (1993). The research is classified as exploratory, and, given the prototypical nature of the developed iReq Add On, it falls under the domain of experiment-driven research (GUSTEDT et al., 2009). This experiment offers a frame for iteratively deriving a machine-readable EIR schema. It is supported by inductive reasoning, in which the premise, the initial situation (EIR content), and the theorem, the hypothesis of information requirements in the modelling environment (the Add On), together are set to form an axiom, namely the input of XML-based EIR content (Denning et al., 1989, p. 10), based on which the authors aim to derive relevant conclusions and potential constraints for a schema addressing machine-readable EIR content. Additionally, this experiment is a step towards verifying the applicability of the fairly new LOIN specification.

3. Information Requirements

3.1 State of the Art

An introduction into different information requirement definition approaches is given in (Tomczak et al., 2022). One of the more established approaches, IDM (International Organization for Standardization, 2016), offers the possibility of generating an XML-based IDM in which the exchange requirements can be defined (in association with processes and transactions), which can in turn be used to create functional parts for the generation of an MVD (Lee et al., 2013), thus being very relevant for software implementation and certification. Another XML-based approach is IDS, which relies strongly on the structure of the main format used in the AEC (Architecture, Engineering and Construction) industry (IFC), and aims at defining alphanumerical IR and the way they should be exchanged (BuildingSMART International, 2020). Meanwhile, the Level of Information Need scope (European Committee for Standardization, 2020) aims for a more agnostic approach of IR definition, enabling its use also for other formats needed in construction, such as LandXML or CityGML (GitHub, 2023b), while also addressing geometrical and documentation requirements (in addition to alphanumerical, supported by the Data Templates approach (International Organization for Standardization, 2020)). Both schemas, IDS and LOIN (the latter through the incorporation of the Data Templates schema), offer the possibility of not only defining required properties, but also setting specific values in their current versions (GitHub, 2023a, 2023b). The XSD (XML Schema Definition) for IDS, LOIN and Data Templates are currently still under development. EIR do not have a machine-readable schema, even though they are the primary source of information requirements in the industry, given that they are obligatory during tendering and therefore binding contractual documents (VDI Verein Deutscher Ingenieure e.V., 2020).

Developments in the area of requirement definition and standardization are predominantly XML-based, as evidenced by the presented approaches. This can be explained by the exchange facilitation between application pairs enabled by XML (Eastman et al., 2018, p. 86), as well as the compatibility of XML with RDF (Resource Description Framework) and other Linked Data approaches (Borrmann, König et al., 2021, 226 f.).

3.2 Requirement Content as Data Input

The content of EIR documents can be multifaceted, given their project-specific nature (VDI Verein Deutscher Ingenieure e.V., 2020). This chapter addresses typical EIR contents, emphasizing their suitability as input data for integration in the modelling environment through the proposed XML-based iReq Add On.

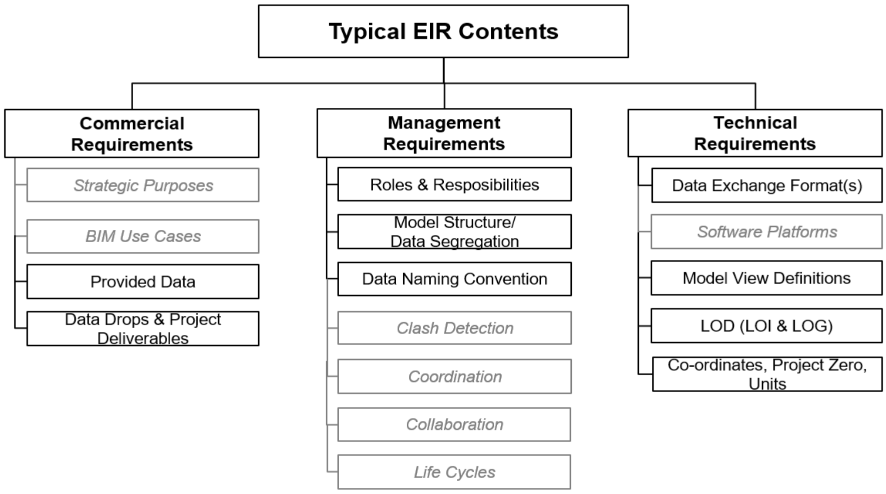

Figure 2: Typical EIR content based on collected industry documents (Mellenthin Filardo, 2019). Grey content is classified as not compatible with transfer to the modelling environment.

A study to collect and evaluate contents of EIR documents has been presented in (Mellenthin Filardo, 2019), as depicted in Fig. 2. Regarding the developed workflow, a major distinction of EIR content must be made: Requirements suitable for integration in the modelling environment and requirements relevant in project management and in the controlling stage, which do not necessarily need to be included in the model, but are relevant, nonetheless. In the latter category fall the grey contents from Fig. 2. Strategic purposes are amongst the unsuited requirements, given their unstructured nature. This experiment partly tested the LOIN specification to define the Level of Information Need, making the traditionally separate assignment of use cases, life cycles and the Level of Development (LOD) definition (spreadsheets) obsolete. Also unsuitable for model integration are requirements regarding clash detection since they only become relevant after the export. Some modelling environments have built-in clash detection tools, so that an integration could be feasible in future. Similar considerations go for coordination requirements. Requirements regarding collaboration can be varied and therefore be partially relevant during modelling and partially outside the modelling environment/phase. To illustrate a possible (and minimal) integration of collaboration requirements, minimal partial model definitions (such as Arch, MEP, Electro) were included in the implementation of this experiment, given their relevance for the export of project deliverables. Provided data may or may not be suitable for integration, depending on the nature of the data (scans of drawings, lists, surveys, CAD files etc.).

Contents suitable for model integration can be defined as those necessarily included in the project deliverables for a specific data drop. Therefore, requirements for project deliveries and their LOIN (or LOD) are the main content of an EIR document. In this experiment, the LOIN specification was used to transfer requirements for an object type (e.g., a Window) and a project-specific property (Life Cycle Costing Codes) and its predefined, project-specific values (4.110-4.170) needed for Life Cycle Costing (LCC) as an example, even though the LOIN specification has many more possibilities (Tomczak et al., 2022).

A data drop usually belongs to a project phase and/or a milestone. Project deliverables are the output expected at the data drop (e.g., IFC files, drawings, schedules or lists). The information container (typically a file) naming convention is applicable to the output, as well as the model structure and/or model segregation, which are project-specific, i.e., a building complex divided in wings, a division by trades, or separate financiers, that might require separate output documents, such as lists, costing and schedules. Roles and responsibilities are relevant for the automated generation of title blocks in drawings and other PDF output. Information container naming conventions usually differ for each client/project and can play a significant role based on the chosen Common Data Environment (CDE). The file naming codes used in the presented Add On implementation were defined as Date (YYMMDD), Project Abbreviation (in this case iReq), Partial Model (such as Arch, MEP, Electro; also mirroring collaboration requirements), Project Phase (e.g., BA01 or BA02). The project phase can be relevant for certain objects with connection points between two phases, such as cables or structural elements. The designation phase can be primarily temporal or spatial, with both possibly having effects on the model structure. File type extensions are feasible and the file name elements (tokens) are separated by a customizable separator. Further codes, that can be included in information container naming conventions, address indexes and numbering of files, and possibly the LOD of the file's content. Linked to the project deliverables are the data exchange formats, such as PDF for drawings or IFC (including its version) for models, the latter linked to an MVD, such as Design Transfer View. Roles and responsibilities are relevant in all project phases and are therefore suitable for integration in the modelling environment. An innocuous yet critical requirement for correct modelling and especially for later model coordination, one of the key benefits of the BIM method according to users (Bundesingenieurkammer, 2022), regards the georeferencing information, which includes defining the project zero. Location data traditionally consists of latitude, longitude, altitude, a coordinate reference system as well as geodetic and vertical datums, which were included in the XML. Additionally, references to external sources were also included, such as EPSG (European Petroleum Survey Group) codes.

The objective of this implementation was to illustrate the requirement integration in the modelling environment, potentially paving the way for machine-readable EIR. To avoid developing an umpteenth specification, this experiment tested the usability and inclusion of the LOIN specification, applied to the classical definition of LOD requirements. Therefore, an XML with custom values was generated for the purpose of this experiment and used as data input, which includes requirements according to the LOIN schema. Upon a successful integration, further development should go into formalizing the EIR schema and the information depth used by the developed Add On as well as delimiting it against existing specifications.

4. Implementation

Based on the discussed input data and within the possibilities of the ArchiCAD 26 API, an Add On for the integration of information requirements was implemented for the ArchiCAD 26 modelling environment. The implementation is divided into a description of its functionality and the achieved results of the Add On, which is called iReq.

4.1 iReq Functionality

The structure and functionalities of the developed iReq Add On prototype are limited to the functions provided by the ArchiCAD API through the C++-based developer SDK, in its 26th version (GRAPHISOFT, 2022). Central functionalities were implemented in multiple components, the main Add On file (AddOnMain.cpp), as well as a custom XML loader (XmlLoader.cpp), an interaction component for data drops (DataDrop.cpp) and multiple user interfaces (e.g., PhaseDialog.cpp), which are specific to the iReq Add On and are responsible for loading the project-specific XML file (Fig. 3 A), generating tailored the Data Drop output files (Fig. 3 B) and the phase dialog box (Fig. 4 A) respectively. These header files connect the main script to the functionalities implemented by the authors.

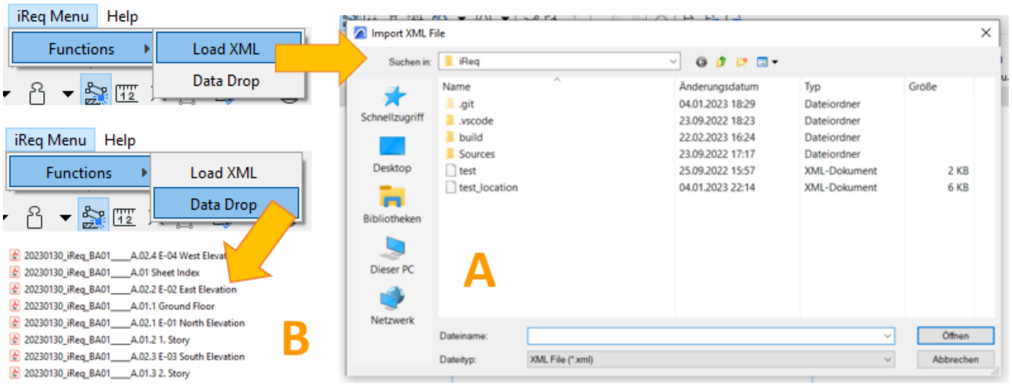

Figure 3: The two main functionalities of the developed Add On: Load requirements, by importing an XML file (A), and Data Drop, by which output is automatically generated (B).

After import, iReq adds the content of the loaded, project-specific XML to the respectively implemented ArchiCAD parts. Specifically, it takes the partial LOIN specification mentioned in chapter 3.2 (LCC Code property and predefined values) and assigns them to construction objects of the targeted type (Window). When an object is modelled, the Event Handler is triggered by the iReq Add On and the required property, previously stored in the project, is assigned as an IFC property. The LOIN specification defines a ConstructionObject element, which has a Uniform Resource Identifier (URI). The property is defined in a separate element (Property), which also has a URI. Property and ConstructionObject are linked by the DataTemplate element, which is defined as a link in the currently available version of the Data Templates schema and maps the URI of the Property element to the ConstructionObject element. This link allows the definition of specific properties and sets of properties to various objects within the specification.

The georeferencing properties are assigned to the project location in ArchiCAD using the API functions to set environmental properties. The information container naming convention and project phase are loaded into the project as well. The latter requires a user interaction to select the project phase, which is later regarded in the file name. The file naming convention is stored in the project and applied when triggering the iReq Data Drop function. Disciplines are loaded into the Layout Book (which functions like a drawing/output manager), where the assignment of drawings, schedules, lists to disciplines is possible (or model parts, or other concepts), and is also regarded in the file naming at saving. The saving component of iReq triggers both the documents defined in the Layout Book as well as the IFC output. iReq reads the format (IFC), its version (4) and the MVD (Design Transfer View) defined in the imported XML to select and configure the correct IFC translator.

4.2 iReq Results

The results of the Add On implementation are the integration of project-specific requirements into the modelling environment, offering an automated approach to project management and technical requirements within the design and modelling phase.

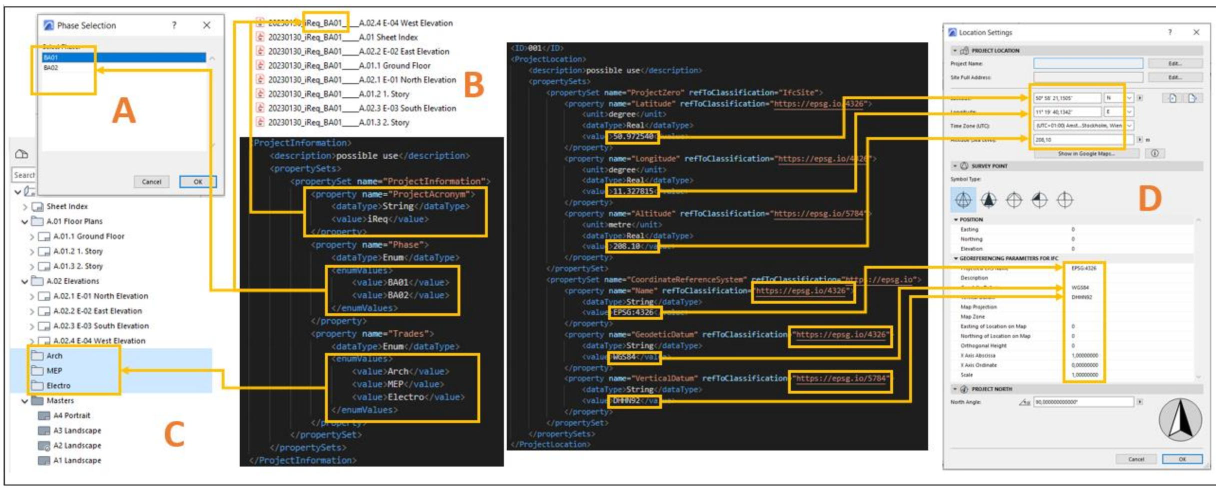

Figure 4: Parts of the added requirements in the modelling environment and where they came from (XML).

Fig. 4 (A) shows the implemented interactive menu for choosing the project phase from the options given in the imported XML. This is relevant e.g., for the data drop and file naming, given that the phase might be included in the file name for easier handling by subsequent users. (B) shows the generated drawings for the data drop already with the naming according to the defined information container naming convention from the imported XML (date, abbreviation, project phase). (C) shows the required trades (also understood as partial models) exemplarily as a support system within the Layout Book. The idea behind this is reminding the user that those drawings must be delivered at data drop, thus avoiding that certain details get overlooked, but being just a first step with many feasible extensions. Finally, (D) show the complete location settings imported from the XML file. The addresses of the EPSG codes for all the georeferencing parameters contained in the IFC schema were included in the XML, but unfortunately ArchiCAD does not yet support its display in the user interface.

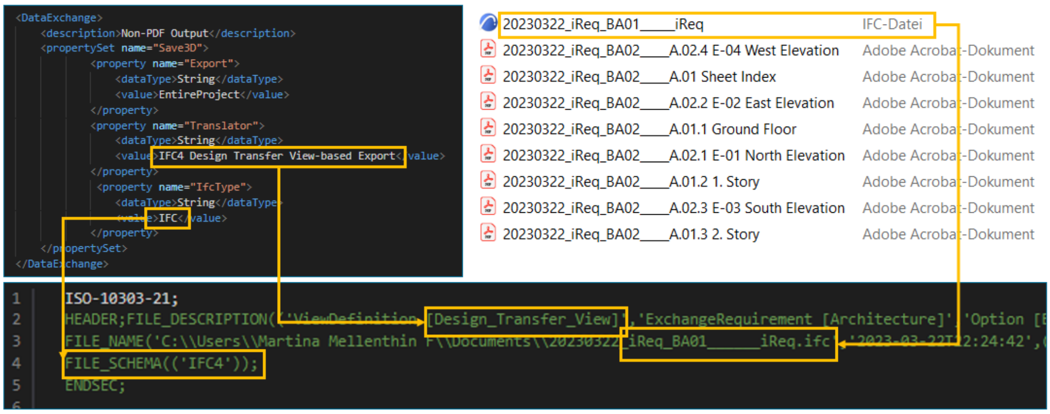

Figure 5: Added export requirements (XML) identified in the generated output (IFC file).

With the location information incorporated into the project, the required location properties are already included in the exported IFC file (IfcSite with global coordinates). With the implementation of IFC 4.3, the EPSG codes, as mentioned in chapter 4.1, will also be included in future IFC output from ArchiCAD. The current iReq implementation exports the Data Drop output to local files, but a connection to the respective CDE can be a possible future extension. The functionality of predefining the IFC and MVD output was also successfully tested, shown in Fig. 5, based on the values described in 4.1.

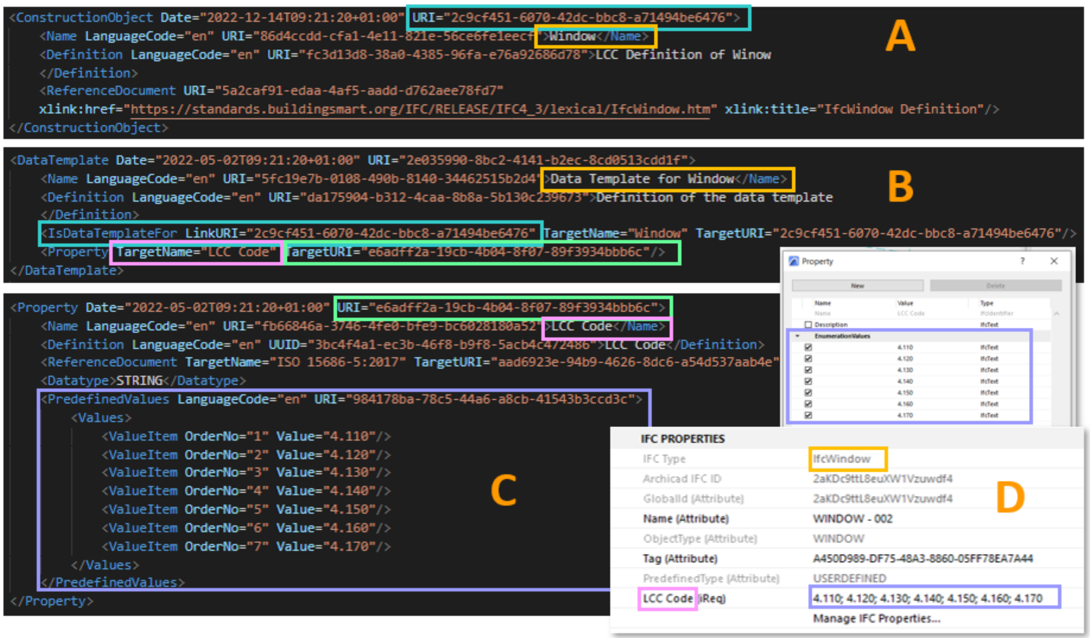

Figure 6: LOIN specification defining a specific property (LCC Code, in pink, with its URI in green), for an object class (Window, in yellow, with its URI in cyan) with specific values (data type and predefined values in lilac).

Fig. 6 shows the relevant parts of the input LOIN XML file (Fig. 6 A, B and C), respectively the ConstructionObject, and Property elements with their name and URI each (yellow/pink, cyan/green correspondingly), as well as the DataTemplate element linking the URI from the Property to the URI from the ConstructionObject. In Fig. 6 C, PredefinedValues are highlighted (in lilac), whose assignment through iReq was successful, as can be verified by Fig. 6 D, where the LCC Code property was assigned to the (IFC) properties of the modelled window, and the imported predefined values can be selected from a menu.

5. Conclusion and Outlook

The proposed workflow was proven successful by the implemented iReq Add On for ArchiCAD. The imported XML-based and project-specific requirements were imported and either directly assigned or stored in the project and triggered, when necessary, thus verifying the viability of an a priori requirement integration. Additionally, the developed iReq Add On accepted the partial LOIN specification, thus offering a first validation of the current schema version as well as a tangible use case for its application.

This design support system, based on parsable requirements, can improve the design quality and reduce the manual effort for information assignment and retrieval significantly. In addition to more robust requirement definition (regardless of whether they are information requirements or actual requirement values), and less iterations during data drops, an expected added benefit of the presented approach is an increase in BIM adoption in the industry. As of now, not all stakeholders are convinced of the benefits associated with modelling and with the BIM method. A secured assistance system can be perceived as a comfort but must be established as a need for our professionals if we are to avoid more shortages of qualified staff. The tools our market offers today can be powerful but require an alarming amount of programming. These skills are expected to be acquired by experienced professionals on-the-go and by younger experts in addition to a full architecture or civil engineering study programme, instead of expecting the tools we use to be more user friendly and more comfortable. To achieve that, the components of automation that directly overlap with those of project and contract management must be defined. EIR are meanwhile an established tool of project management, so they could be suited as a standard for specification of information requirements.

IDS relies on the IFC structure exclusively, which is very relevant for export, but differs from the project structure within the authoring application. Relying on IDS can facilitate implementations, since the IFC structure is widely documented, but it also links the scope and development of requirement definition directly to those of IFC, thereby excluding options of requirement definition for elements outside the IFC scope, that still can be relevant for construction projects, such as LandXML, CityGML, amongst others. Nonetheless, a harmonization with the export format is advisable. Given those reasons combined with the success of this experimental implementation, the LOIN specification is indicated as a possible component of an EIR wrapper schema.

Future work should examine, if and how much such an EIR schema can rely solely on the LOIN specification and which contents must be defined in the additional wrapper schema. Further, handling and maintenance of comprehensive IRs should be addressed. The a priori incorporation of requirements requires machine-readable, e.g., XML-based IR. Part of the future scope must include the generation of i.e., the eirXML, which must not consist of manually writing XML files by the end user. On an operational level, the iReq Add On must be extended to support requirement versioning and possibly a wider range of functions, such as links to provided data or direct connectivity with the CDE.

Statement of data availability

Data and source code can be made available upon reasonable request to the corresponding author.

References

AEC3 GmbH (2023). BIMQ: Intelligent Information Management of BIM Projects [Online]. Available at https://www.bimq.de/en/ (Accessed 18 January 2023).

Borrmann, A., Forster, C., Liebich, T., König, M. and Tulke, J. (2021). Germany’s Governmental BIM Initiative – The BIM4INFRA2020 Project Implementing the BIM Roadmap, in Proceedings of the 18th International Conference on Computing in Civil and Building Engineering, Cham, Springer International Publishing, pp. 452–465.

Borrmann, A., König, M., Koch, C. and Beetz, J. (eds) (2021). Building Information Modeling: Technologische Grundlagen und industrielle Praxis, 2nd edn, Springer Vieweg.

BuildingSMART International (2020). Technical Roadmap buildingSMART.

Bundesingenieurkammer (ed) (2022). Umfrage: Fehlende Nachfrage bremst Digitalisierung von Ingenieurbüros aus [Online]. Available at https://bingk.de/umfrage-fehlende-nachfrage-bremst-digitalisierung-von-ingenieurbueros-aus/ (Accessed 10 March 2023).

Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J. and Young, P. R. (1989). Computing as a discipline, Computer, vol. 22, no. 32, pp. 63–70.

dRofus AS (2023). dRofus A Nemetschek Company [Online]. Available at https://www.drofus.com/product (Accessed 13 March 2023).
Eastman, C., Teicholz, P., Sacks, R. and Lee, G. (2018). BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers, 3rd edn, Hoboken, New Jersey, John Wiley & Sons.
European Committee for Standardization (2020). EN 17412-1: Building Information Modelling - Level of Information Need - Part 1: Concepts and principles.
GitHub (2023a). buildingSMART/IDS: Computer interpretable (XML) standard to define Information Delivery Specifications for BIM (mainly used for IFC) [Online]. Available at https://github.com/buildingSMART/IDS (Accessed 18 January 2023).
GitHub (2023b). LOIN-XML-Schema [Online]. Available at https://github.com/anborr/LOIN-XML-Schema (Accessed 13 March 2023).
GRAPHISOFT (2022). General API Development Kit 26 [Online]. Available at https://archicadapi.graphisoft.com/api-reference (Accessed 5 April 2023).
GUSTEDT, J., JEANNOT, E. and QUINSON, M. (2009). Experimental Methodologies for Large-Scale Systems: a Survey, Parallel Processing Letters, vol. 19, no. 03, pp. 399–418.
International Organization for Standardization (2016). ISO 29481-1:2016: Building information models - Information delivery manual - Part 1: Methodology and format.
International Organization for Standardization (2018). ISO 19650-2:2018: Organization and digitization of information about buildings and civil engineering works, including building information modeling (BIM) - Part 2: Delivery phase of the assets.
International Organization for Standardization (2020). EN ISO 23387:2020: Building information modelling (BIM) - Data templates for construction objects used in the life cycle of built assets - Concepts and principles.
Lee, G., Park, Y. H. and Ham, S. (2013). Extended Process to Product Modeling (xPPM) for integrated and seamless IDM and MVD development, Advanced Engineering Informatics, vol. 27, no. 4, pp. 636–651.
Liebich, T. (1993). Wissensbasierter Architekturentwurf - von den Modellen des Entwurfs zu einer intelligenten Computerunterstützung: Ein Weg zu den Entwurfsgrammatiken und zur multiplen graphischen Repräsentation, Dissertation, Weimar, Bauhaus-Universität Weimar.
Mellenthin Filardo, M. (2019). Praxisrelevante Auftraggeber-Informations-Anforderungen für Infrastrukturprojekte als Indikatoren des Einführungsgrades von BIM in Deutschland, in 31. Forum Bauinformatik: Proceedings, pp. 9–16.
Mellenthin Filardo, M. and Krischler, J. (2021). An A Priori EIR-Compliant Modelling Approach, 32. Forum Bauinformatik 2021: Proceedings. Darmstadt, pp. 183–191.
Plannerly (2019). The BIM Management Platform - Plannerly [Online]. Available at https://plannerly.com/ (Accessed 13 March 2023).
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.
VDI Verein Deutscher Ingenieure e.V. (2020). VDI 2552 Blatt 10:2020-01: Richtlinie VDI 2552 Blatt 10 Entwurf, Berlin: Beuth Verlag GmbH.
Wang, Y., Bai, Y. and Shen, Q. (eds) (2012). Proceedings of 2012 International Conference on Construction & Real Estate Management, China Architecture & Building Press.