Article

Not peer-reviewed version

BCF as an Example of Reification and Serialization in Building Information Modeling (BIM) Practice

Andrzej Szymon Borkowski * , Magdalena Kładź , Mikołaj Michalak

Posted Date: 18 March 2026

doi: 10.20944/preprints202603.1498.v1

Keywords: Building Information Modeling (BIM); BIM Collaboration Format (BCF); reification; serialization; interoperability; openBIM; issue management; cross-industry coordination

Article

BCF as an Example of Reification and Serialization in Building Information Modeling (BIM) Practice

Andrzej Szymon Borkowski [1,*] Magdalena Kładź [2] and Mikołaj Michalak [3]

[1] Faculty of Geodesy and Cartography, Warsaw University of Technology, Politechniki Square 1, 00-661 Warsaw, Poland
[2] Port Polska, Aleje Jerozolimskie 142B, 02-305 Warsaw, Poland
[3] Faculty of Architecture, University of Technology and Arts and Applied Sciences in Warsaw, 12 Olszewska Street, 00-792 Warsaw, Poland
* Correspondence: andrzej.borkowski@pw.edu.pl

Abstract

Building Information Modeling (BIM) has fundamentally changed the way interdisciplinary coordination works in construction projects; however, the theoretical mechanisms underlying open collaboration standards in this field remain insufficiently explored. This article fills this gap by presenting a systematic analysis of the BIM Collaboration Format (BCF) through the lens of reification and serialization, two fundamental concepts in information systems theory. Although the BCF format is widely used in the industry and implemented in major BIM tools for clash detection and issue tracking, the existing literature treats it primarily as an operational tool, overlooking the deeper information-theoretical principles that govern its architecture. The analysis demonstrates that BCF achieves reification by transforming informal coordination knowledge—such as verbally communicated clashes, scattered email threads, and undocumented design decisions—into first-class objects (Topic, Comment, Viewpoint) equipped with unique identifiers, typed attributes, ownership, temporal metadata, and formalized inter-object relationships. Further analysis was conducted on BCF's serialization mechanisms, including XML encoding for file exchange, JSON for RESTful API communication, and ZIP archiving as a distribution container, each of which was selected to balance human readability, schema validation, compression, and cross-platform portability. The complementarity of these two mechanisms was examined: reification determines what to preserve and in what structure, while serialization determines how to encode and in what format, which together enable interoperable, auditable, and automatable coordination workflows in heterogeneous software environments. The analysis was illustrated with a real-world BCF example from a major infrastructure project in Poland, demonstrating practical alignment between theoretical constructs and their implementation. The research results provide both a conceptual foundation for researchers working on open BIM standards and practical guidance for practitioners seeking to optimize issue management, the implementation of a Common Data Environment (CDE), and the specification of Exchange Information Requirements (EIR). Implications for extending the principles of reification and serialization to new fields, including digital twins and IoT-based facility management.

Keywords: Building Information Modeling (BIM); BIM Collaboration Format (BCF); reification; serialization; interoperability; openBIM; issue management; cross-industry coordination

1. Introduction

1.1. Research Background

The Building Information Modeling (BIM) methodology is transforming the construction sector by introducing a work paradigm based on a shared digital model instead of traditional two-dimensional documentation [1]. The central element of BIM is a digital model containing not only

 doi:10.20944/preprints202603.1498.v1

geometry but also semantics, i.e., information about the function, materials, relationships, and properties of elements [2]. This rich information layer distinguishes the BIM model from a simple three-dimensional representation (e.g., 3D CAD) and serves as the foundation for advanced analyses, simulations, variant studies, and the automation of construction processes. However, the BIM model alone is not sufficient for effective interdisciplinary collaboration on complex construction projects, especially so-called mega-projects (large-scale and interdisciplinary). Additional mechanisms are needed for communicating issues, comments, and design decisions among project participants, who often work in different proprietary tools and use different data formats.

The BIM Collaboration Format (BCF) was developed in response to this need and quickly became one of the key standards of the openBIM initiative [3]. It is an open standard developed by buildingSMART that enables the exchange of information about issues detected in a model without the need to transfer the entire model (geometry and semantics), making it an extremely effective communication tool for large design teams. BCF is an excellent example of the practical application of two fundamental IT concepts: reification and serialization, which originate from database theory and software engineering and have direct implications for information management practices in the construction industry.

1.2. BCF in the Context of the openBIM Concept

The first version (BCF 1.0) provided basic functionality for exchanging viewpoints and comments regarding issues detected in BIM models. Subsequent versions (2.0, 2.1, and the current 3.0) gradually expanded the specification to include workflow management capabilities, the assignment of priorities or deadlines, and integration with REST API interfaces for cloud-based Common Data Environment (CDE) platforms [3]. A key feature distinguishing BCF from other openBIM standards is the deliberate separation of issue information from the model's geometric data. While another popular format, Industry Foundation Classes (IFC), serves as a comprehensive data model for exchanging building geometry and semantics, BCF focuses exclusively on coordination issues, referencing model elements via their GlobalId identifiers rather than duplicating geometric information. This architectural decision results in lightweight files, typically a few kilobytes in size, enabling rapid exchange via traditional and modern digital communication channels (email, CDE platforms, or API-based integrations), even in environments with limited bandwidth.

Previous research on BCF has focused mainly on practical implementation aspects. However, the theoretical foundations underlying the BCF data model have received relatively little attention in the scientific literature. The concepts of reification and serialization though fundamental to understanding how BCF transforms informal coordination knowledge into structured, exchangeable data, have not been systematically analyzed in the context of open BIM collaboration standards. This theoretical gap limits the ability of researchers and practitioners to fully understand the design rationale behind BCF and to apply similar principles when developing new standards or extending existing ones for emerging use cases, such as digital twins and facility management.

1.3. Research Gap and Objective

A review of the literature indicates a significant asymmetry between the practical application of the BCF format and its theoretical understanding. On the one hand, BCF is widely used in the construction sector and implemented in most leading BIM tools for information modeling or clash detection, including Solibri Model Checker, Autodesk Navisworks, Trimble Connect, BIMcollab, and as plugins for proprietary applications such as Autodesk Revit and Graphisoft ArchiCAD. On the other hand, the scientific literature treats BCF primarily as an operational tool, neglecting a deeper analysis of the computational mechanisms that enable its functioning.

None of the existing perspectives undertake a systematic analysis of the format's theoretical foundations, particularly the reification mechanisms that transform informal coordination problems into structured objects, and the serialization mechanisms that enable their persistent storage and exchange between heterogeneous systems. The lack of such an analysis has practical consequences

 doi:10.20944/preprints202603.1498.v1

that extend beyond the academic sphere. BIM practitioners who do not understand the principles of reification may underestimate the importance of fully populating BCF object attributes, treating fields such as priority, completion date, or responsible person as optional extras rather than key elements of the reified problem. Similarly, a lack of understanding of serialization can lead to inefficient data exchange practices, such as sending full IFC models instead of lightweight BCF files or relying on screenshots and text descriptions instead of structured viewpoints. In a broader context, understanding these mechanisms is essential for the informed design of potential future openBIM standards and for the effective integration of BCF with other digital technologies, where the reification and serialization of operational events take on new significance.

Thus, the aim of this article is to fill this gap through a systematic analysis of the BCF format through the lens of the concepts of reification and serialization. The article sets three specific research objectives. First, to explain how BCF reifies informal coordination problems into Topic, Comment, and Viewpoint objects with precisely defined attributes and relationships. Second, to analyze the serialization mechanisms used in BCF, including the XML format for files and the JSON format for APIs, along with a discussion of the ZIP archive structure as a container for serialized data. Third, the identification of the practical implications of these mechanisms for BIM project management, with particular emphasis on interoperability, auditability, and the potential for automating coordination processes. The article illustrates these issues using the example of a real-world communication system employing BCF files, native tools, and a CDE platform at Port Polska (formerly Centralny Port Komunikacyjny), demonstrating the relationship between theoretical concepts and their practical implementation.

2. Literature Review

2.1. Reification as a Mechanism for Structuring Knowledge

The term "reification" comes from Latin, where res means "thing" and facere means "to make," which can be translated as "turning into a thing" or "making something a thing." In the context of information systems, reification involves transforming something abstract into a concrete entity that can be manipulated in a formal and repeatable manner [4]. This means representing relationships, events, or processes as first-class objects with their own attributes and identity, which allows them to be stored, searched, modified, and analyzed on par with other entities in the system [5]. This concept is of fundamental importance for the design of information systems, as it allows for the precise modeling of aspects of reality that would otherwise remain hidden or dispersed.

In the classical data model, the relationship between two entities is merely a connection, a line linking two entities without their own identity or attributes. After reification, it becomes an independent entity with full rights to participate in the data model [6]. This makes it possible to track history, report changes, or perform analysis, which would be impossible with a simple relationship. In the context of BIM, reification is of particular importance due to the complexity of relationships found in building models. The IFC standard makes extensive use of reified relationships, and classes such as IfcRelSpaceBoundary—representing the boundary between a space and a bounding element—IfcRelConnectsElements—describing the connection between structural elements—or IfcRelAssociatesMaterial—assigning a material to an element—represent relationships as full-fledged objects with their own properties and the ability to be extended [7]. Research indicates that a proper understanding and use of models is one of the key factors for the success of BIM implementations [8], which brings practice closer to the concept of the Digital Twin.

2.2. Serialization as a Mechanism for Preservation and Exchange

Serialization is the process of converting data structures or objects existing in a computer’s main memory into a format that allows them to be stored on a permanent medium, transmitted over a network, and later reconstructed in the same or a different system [9]. The reverse process, deserialization, reconstructs the original structures from the saved form, restoring the objects’ full

 doi:10.20944/preprints202603.1498.v1

functionality and relationships. Serialization is essential for the permanent storage of data beyond the application's runtime and for data exchange between applications built in different ways. Without serialization mechanisms, modern information systems could not function, as data would exist only in volatile operating memory and be lost with every application shutdown.

In the BIM environment, serialization takes various forms tailored to specific use cases and technical requirements. The IFC-SPF format, based on the STEP standard, is compact and widely supported but difficult for humans to read. The IFC-XML format offers better readability at the cost of larger file sizes. The newer IFC-JSON format is gaining popularity in web applications due to native support in JavaScript environments [10]. In addition to open formats, there are also application-specific formats, such as .rvt for Autodesk Revit or .pln for Graphisoft Archicad, which serialize not only the model but also application-specific settings, views, and configurations. Each of these formats serializes the same conceptual model into a different textual or binary representation, and the choice of the appropriate format depends on the context of use, interoperability requirements, and technical limitations of the target systems.

2.3. The Relationship Between Reification and Serialization in the Context of BIM

Reification and serialization are mechanisms for representing knowledge that together form the foundation for sustainable and exchangeable information management in construction projects. Reification creates objects worth preserving by giving structure and ident ity to phenomena that previously existed only as informal knowledge (e.g., a modeling error) or ephemeral messages (a succinctly described geometric collision). Serialization enables the preservation of these objects (or phenomena) in a form that can be stored, transmitted, and reproduced in various technological and organizational contexts. This can be summarized as follows: reification answers the question of what to record and what structure to give to the recorded information, while serialization answers the question of how to record and in what format to represent the reified objects. These two questions are interdependent, as decisions regarding reification affect serialization options, and the limitations of serialization formats may necessitate specific approaches to reification.

Without reification, much valuable information remains ephemeral and exists only in the minds of project participants, in informal conversations, or in emails that are difficult to search, aggregate, and analyze. Knowledge about why a specific design decision was made, who reported an installation conflict, and what alternatives were considered before choosing a particular solution remains hidden unless it is reified into objects with attributes and relationships. Without serialization, even reified information remains ephemeral and is lost when the application is closed, the system crashes, or the application session ends. Research shows that effective knowledge management in BIM projects requires a conscious approach from various parties, especially if a so-called "BIM mandate"—that is, the mandatory use of BIM methodology in public procurement—is to be adopted in a given country [11].

3. Reification in BCF

3.1. The Coordination Problem as a First-Class Object

Before the introduction of BCF, a typical problem in a BIM project was handled through a conversation (e.g., a coordination meeting), an email sent to a group of recipients (an attempt to resolve a clash), a meeting note jotted down in a participant's personal notebook, or a verbal instruction given in the design office. Information about a collision between a pipe and a beam might have been phrased as: "There's a collision somewhere on the second floor, near the elevator; it needs to be fixed by next week." Such information is unstructured because it lacks unambiguous attributes that allow for systematic processing. It is also incomplete due to the lack of a precise location that would enable the problem to be quickly found in the model. It is unassigned due to unclear responsibility for the solution, which leads to blurred accountability and potential conflicts. Finally, it is difficult to track due to the lack of status and history, which prevents verification of progress and auditing of the problem-solving process. In mega-projects with hundreds of participants and thousands of potential clashes, such an informal mode of communication quickly leads to information chaos, overlooked problems, and costly delays.

BCF reifies the problem into a Topic object (name) with precisely defined attributes compliant with the buildingSMART specification [12]. This object has a unique Guid identifier in UUID format, which ensures global uniqueness and enables unambiguous reference to the problem in any system. It has a TopicType category specifying whether the issue is a geometric clash, a procedural problem, a design query, or an informational note, allowing for filtering and prioritization by type. It includes a TopicStatus indicating the current resolution status in the cycle from opening through work in progress to resolution and closure. It stores a title that serves as a concise description of the problem, as well as a detailed description containing the full context, detection history, and proposed solutions. It specifies the priority—ranging from critical through high and normal to low—the creation date along with the reporter's ID, the person responsible for the resolution, and the completion deadline. After reification, the pro blem becomes a first-class entity that can be managed systematically by assigning it to the appropriate people, filtering by various criteria, reporting its status at coordination meetings, escalating it if deadlines are missed, and formally closing it with documentation of the actions taken.

3.2. Comments and Viewpoints as Supporting Objects

BCF also reifies comments, which in a traditional work model would exist as scattered email correspondence that is difficult to link to specific issues and impossible to search systematically. In BCF, each comment becomes a Comment object with attributes such as a unique GUID, the date and time of addition accurate to the second, an author identifier enabling the assignment of responsibility, the comment content containing the substantive statement, and an optional reference to a viewpoint illustrating the context of the comment. Comments are linked to a Topic by being embedded in an XML structure, creating an easy-to-follow discussion thread with a complete chronological history. Thanks to this reification, it is possible to reconstruct the entire decision-making process, understand the arguments of individual participants, and draw conclusions for the future as part of the learning process (adaptation through assimilation and accommodation).

A particularly innovative example of reification is Viewpoint, a reified way of looking at a model that transforms subjective visual experience into an objective, repeatable record. Instead of a verbal description such as "look from the elevator, from above, and you'll see that pipe entering the beam," BCF records precise mathematical parameters that enable the view to be accurately recreated in any compatible tool. Viewpoint contains the camera position defined by X, Y, Z coordinates in the model's global coordinate system, the viewing direction as a unit vector, a vector defining the top of the frame, and the viewing angle for a perspective view or the scale for an orthogonal view. It also includes a list of selected elements identified by their IfcGuid, visibility information for individual elements allowing irrelevant objects to be hidden, coloring that highlights problematic components by assigning warning colors, and optional section planes that allow hidden elements to be revealed. The reification of a viewpoint enables its exact reproduction in any BCF-compatible tool, allowing the user in the office to see exactly what the author of the submission saw in a completely different location, regardless of the software used [13].

It is worth noting that the reification of the point of view in the BCF format serves as a bridge between the mathematical precision of the model and human visual perception. By recording parameters such as camera position coordinates and direction vectors, the designer's subjective perspective is reduced to a deterministic dataset. In practice, this eliminates so-called information noise because the recipient does not have to guess the sender's intentions, as the system automatically positions the virtual camera in the exact same position where the problem was defined. This approach drastically reduces the time required for so-called context switching, allowing the engineer to immediately transition from analyzing the report to substantive work on resolving the collision.

 doi:10.20944/preprints202603.1498.v1

3.3. Relationship with the IFC Model and Maintaining the Format's Lightness

BCF does not duplicate geometric or semantic information from the IFC model, but refers to it via GlobalId identifiers, which is a deliberate design decision aimed at maintaining the format's lightness and avoiding data synchronization issues. The Component element in BCF contains the element's IfcGuid, which is a globally unique 22-character base64 identifier, the source system name indicating the authoring tool, and an optional internal identifier within that tool. Thanks to this approach, a BCF file containing dozens of issues typically takes up a few dozen kilobytes, while the corresponding IFC model can weigh hundreds of megabytes or even gigabytes. This separation enables the rapid exchange of issue information via email, CDE systems, or instant messengers without the need to transfer heavy geometric models.

This is also an example of reference reification, where the reference to an element itself becomes an object with additional metadata that enriches the context [14]. In addition to the GlobalId identifier itself, the reference may contain information about the system in which the element was created, which is important in multi-tool environments where the same project is developed in various applications from a given vendor. It may also contain an identifier specific to the authoring tool, allowing the element to be opened directly in the source application. This approach implements the principle of the so-called "single source of truth," where geometry and semantics remain in the IFC model, and the BCF contains only information about issues and references to elements, avoiding data duplication and the associated version synchronization problems.

4. Serialization in BCF

4.1. XML Format as a Container for Reified Objects

BCF uses XML as the primary serialization format for the reified Topic, Comment, and Viewpoint objects. The choice of XML as the serialization format was driven by a number of technical and organizational factors relevant to the buildingSMART ecosystem [12]. XML offers human readability, enabling the inspection and debugging of files without specialized tools, using only a text editor. It has broad tool support across all popular programming languages and development environments, which lowers the implementation barrier. It enables formal validation via XSD schemas, allowing for automatic verification of the structural correctness of files before processing. Finally, it ensures compatibility with other buildingSMART ecosystem standards, such as IFC-XML and mvdXML, which facilitates integration and the development of consistent solutions. Alternative formats, such as JSON or binary formats, were considered, but ultimately XML was deemed to best meet the requirements for interoperability and long-term archiving.

A BCF file with the .bcf or .bcfzip extension is a ZIP archive containing XML files and optional raster images representing screenshots of views (Figure 1). Using a ZIP archive as a container provides compression that reduces transmission size, where a single file represents a complete set of problems, and ensures portability thanks to universal support for the ZIP format across all operating systems. The structure of a typical BCF 2.1 file includes a bcf.version file in the root directory that specifies the format version and allows applications to adapt their parsing to the appropriate specification. The optional project.bcfp file contains project metadata such as the name and identifier. The optional extensions.xsd file defines allowed values for enumerated fields, enabling adaptation to specific organizational requirements. For each topic, there is a directory named after its GUID, containing the markup.bcf file with the main topic and comments, one or more viewpoint files with camera viewpoints, and corresponding screenshots in PNG or JPEG format.

 doi:10.20944/preprints202603.1498.v1

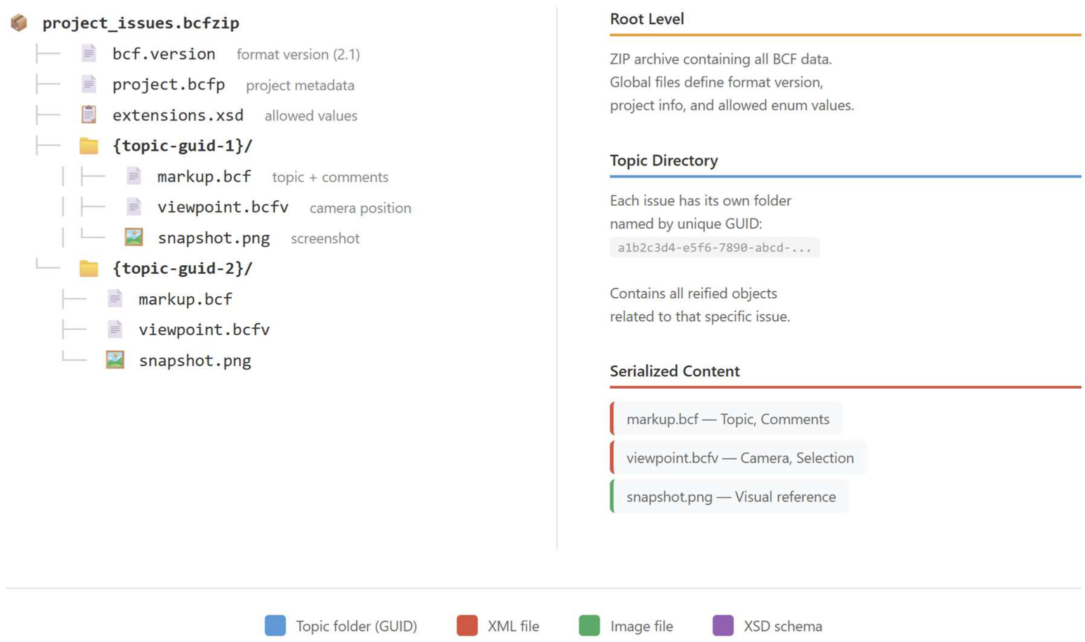


Figure 1. Diagram of the BCF file structure showing the hierarchy of directories and files along with the relationships between them. Source: own elaboration.

4.2. Example of a BCF (Issue Resolution) in Port Polska

The Port Polska investment program includes the construction of a new domestic airport, a high-speed rail network, roads around the airport, a logistics center, Airport City, and activities related to regional development, and is being implemented by the company Centralny Port Komunikacyjny. The state-owned company was established pursuant to Article 11 of the Act of May 10, 2018, on the Central Transport Hub (Journal of Laws 2018, No. 1089) to prepare and implement the Central Transport Hub's Multi-Year Program. The program's goal is to create a modern transportation hub of international significance, connecting various modes of transport: air, rail, and road. The new airport will be designed to handle 34 million passengers annually in its initial years of operation and will be structured in such a way that, in the long term, it can be flexibly expanded in accordance with market needs and growth projections. This is a unique mega-project, the only one of its kind in Poland, integrating many industries, located across three municipalities, and impacting as many as 18 surrounding municipalities. The estimated cost of the investment through 2032 is 131.7 billion zlotys. In implementing such a complex project, it is essential to maintain continuity and efficiency in communication.

The program's objectives are achieved through the execution of specific tasks within individual projects. One of these is the Master Civil Engineer project, which constitutes one of four key contracts. The Contractor's strategic tasks include developing multidisciplinary design plans and technical specifications for the hub's infrastructure, including runways and taxiways, aprons, utilities and service lines, as well as civil engineering structures. The entire program is being implemented in accordance with BIM methodology. The Company has procedures in place specifying information exchange requirements, work on the CDE platform, document review and versioning processes, as well as document naming and numbering conventions. Individual EIR (Exchange Information Requirements) and OPZ (Scope of Work) documents contain requirements regarding the method of communication between the Client and the Contractor.

One of the key tasks in the investment implementation process according to the BIM methodology is model verification. This involves checking the geometric data as well as the semantics of the model, which is intended to reflect the actual structure. Two-dimensional documentation is created based on these models. The process of verifying the representation of requirements and technical specifications can take various forms, but the focus here is on the next step in this process: reporting comments. In the construction sector, it is common to encounter an approach where all errors are reported informally—that is, verbally during a conversation, a coordination meeting, or via messages on communication platforms or email. In such information exchanges, key recipients are typically overlooked, communication continuity is not maintained—and thus its timeliness is compromised—and the messages themselves contain selective data. This leads to delays in construction, increased unplanned costs, and poor quality of the work produced. An approach consistent with BIM methodology clearly specifies, among other things in ISO 19650, how and between which groups effective information exchange should take place. Comments regarding the model should refer directly to it. This eliminates the need to take screenshots, search for information about a faulty element, and locate, for example, a clash each time. The BCF standard is precisely such a solution. Nevertheless, providers of native software or CDE platforms are also introducing their own solutions to increase competitiveness and retain customers within their own product ecosystems. One example is Autodesk, which provides the Autodesk AEC Collection, which includes Navisworks for coordination, clash detection, as well as 4D and 5D simulations. One of the tools available in this environment is the creation of "viewpoints," which are a list of comments related to the model. Such a note consists of the comment content, includes the date and time, author, status, its ID, as well as the viewpoint itself—that is, the camera view specified by the creator—and any markings, e.g., in the form of a revision cloud (Figure 2). The default export of viewpoints from the program is to the XML format. In addition to the displayed information, it stores the comment's source (file name, path, schema), position, rotation, camera vector, view clipping, and element selection. A file exported in this way differs from BCF primarily in terms of compatibility, as it is not suitable for open import into other software but is used for internal information exchange. It also lacks the ability to assign a person responsible for a given issue and to create a history of discussions and changes.

The Navisworks interface allows users to display a list of comments and browse through them, export individual comments or groups of comments to XML format, as well as views in the form of PNG files. Comments are numbered.

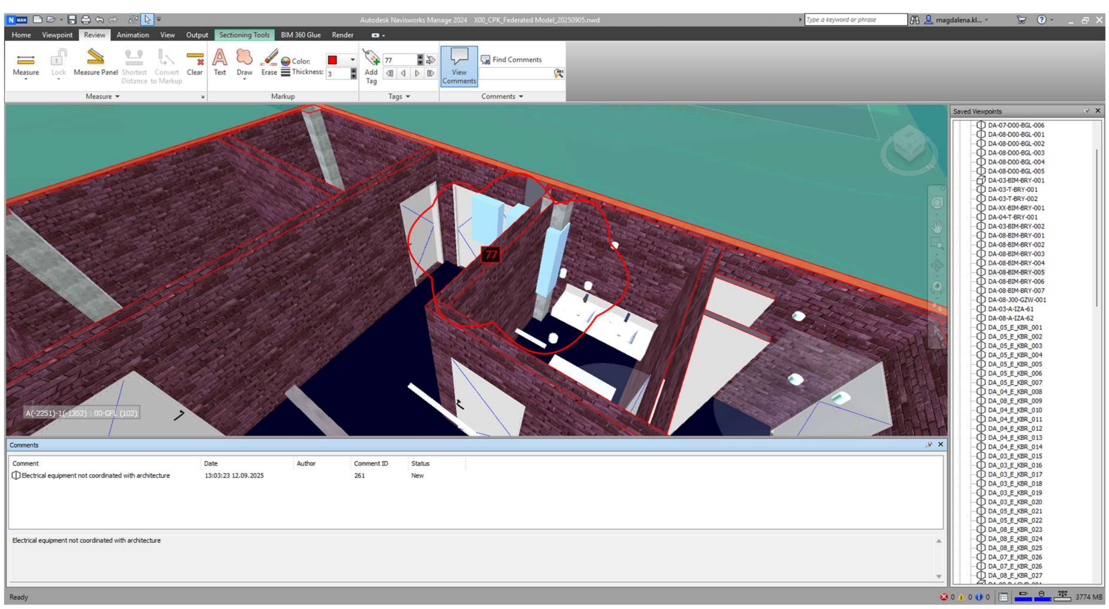


Figure 2. Example of a BCF in the Master Civil Engineer project implemented as part of the Port Polska program in the Autodesk Navisworks interface. Source: own elaboration.

Error communication can also take place within the CDE platform, which serves as a shared working environment for stakeholders in the investment process. The design phase of the Port Polska program is being carried out using the CDE from Bentley Systems. The web version of the ProjectWise platform includes an Issue Resolution (IR) module responsible for exchanging comments and feedback on documentation (Figure 3). This tool allows users to set up information flows between defined user groups, create form templates with specific fields, attach comments to a specific location in a document (creating a reference in the form of a link to the document and markings such as underlining or highlighting with a rectangle), organize comments according to project phases, exchange comments, and add attachments. Each IR form includes a revision history showing the date, time, author, and type of modification made. The IR module provides statistics on, among other things, the statuses of created comments, the time taken to resolve comments, and the number of reports assigned to specific individuals. By using this solution, project participants have constant access to the event history and the resolution status of an issue; they can request further details, and no participant is left out of the comment exchange sequence.

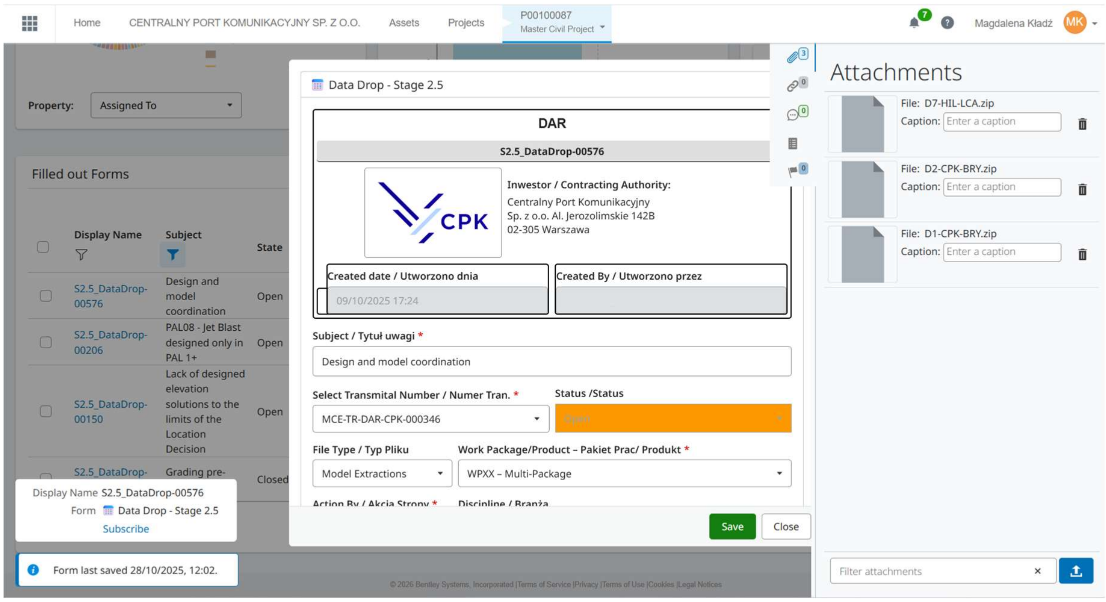
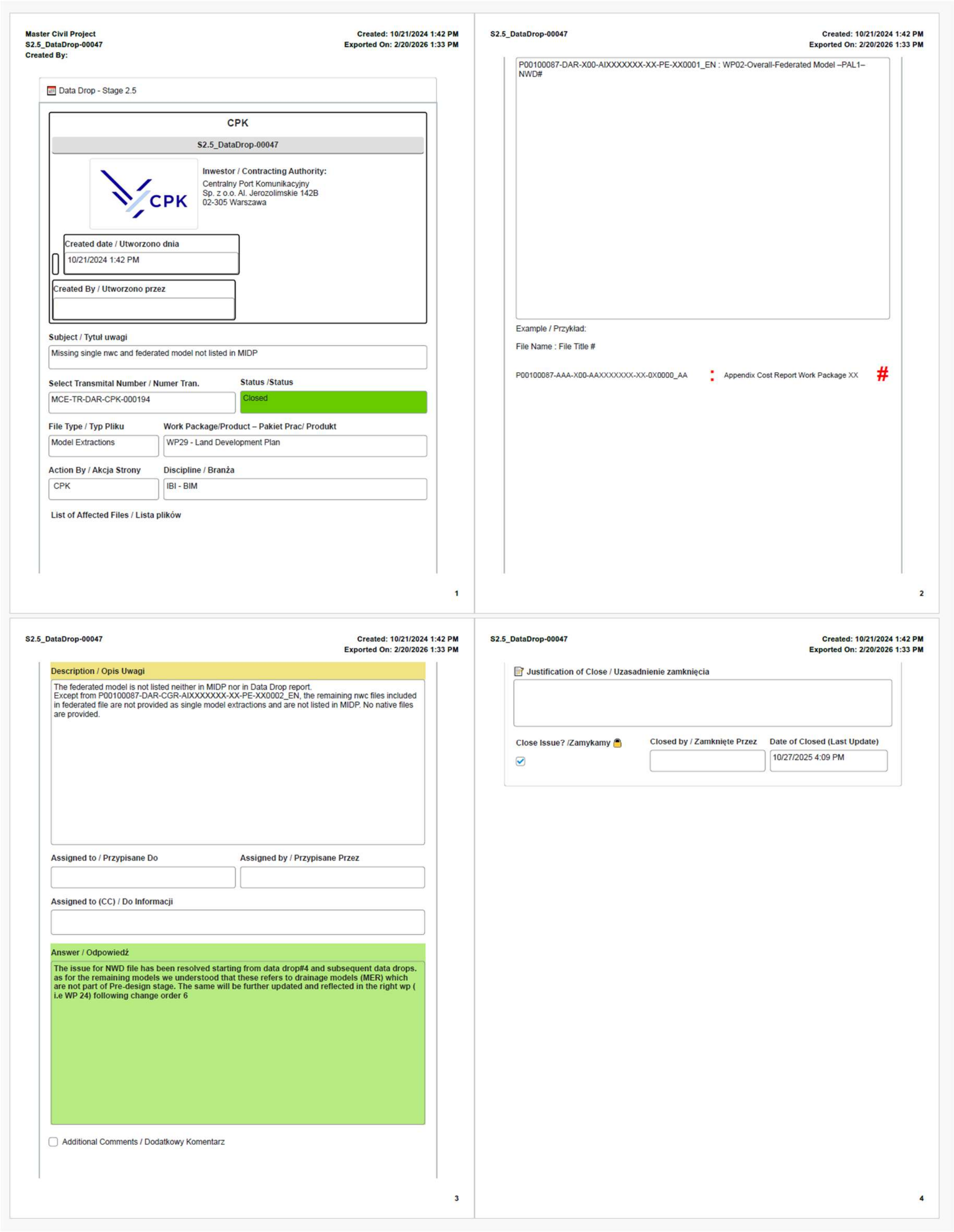


Figure 3. The Issue Resolution module interface in ProjectWise Web with an open issue and its attachments. The form includes the Port of Poland logo as of the date the issue was created, which has since changed. Source: own elaboration.

IRs can also be easily filtered, sorted, and exported as PDF, CSV, ZIP, and even BCF files. The color-coded division of sections clearly indicates which fields should be filled out by the Contractor and which by the Client (Figure 4). A resolved thread is closed, but if there is a need to revisit a specific case, the form can be reopened. By comparison, finding information from two years ago within an email inbox might be impossible. Personnel changes are a natural occurrence. IR retains the personal information of discussion participants; however, if an employee who resolved a given issue via email leaves without adding the appropriate people to the recipients, the rest of the team loses the source of information regarding the verification process. Additionally, that same employee could have moved the thread to the trash, whereas on the CDE Platform, no information is lost.

The Issue Resolution module is also designed for annotating models using the internal iModel tool. This tool allows users to view IFC and native models, review their content, verify clashes, and add comments to specific elements. Bentley ProjectWise includes functionality for importing BCF comments into iModel; however, this capability is limited and has seen little development due to the vendor's own native tools, specifically Issue Resolution.

 doi:10.20944/preprints202603.1498.v1



Master Civil Project
S2.5_DataDrop-00047
Created By:

Created: 10/21/2024 1:42 PM
Exported On: 2/20/2026 1:33 PM

Data Drop - Stage 2.5

CPK

S2.5_DataDrop-00047

Inwestor / Contracting Authority:
Centralny Port Komunikacyjny
Sp. z o.o. Al. Jerozolimskie 142B
02-305 Warszawa

Created date / Utworzono dnia
10/21/2024 1:42 PM

Created By / Utworzono przez

Subject / Tytuł uwagi
Missing single nwc and federated model not listed in MIDP

Select Transmital Number / Numer Tran.
MCE-TR-DAR-CPK-000194

Status /Status
Closed

File Type / Typ Pliku
Model Extractions

Work Package/Product – Pakiet Prac/ Produkt
WP29 - Land Development Plan

Action By / Akcja Strony
CPK

Discipline / Branża
IBI - BIM

List of Affected Files / Lista plików

1

S2.5_DataDrop-00047

Created: 10/21/2024 1:42 PM
Exported On: 2/20/2026 1:33 PM

P00100087-DAR-X00-AIXXXXXXX-XX-PE-XX0001_EN : WP02-Overall-Federated Model –PAL1–NWD#

Example / Przykład:

File Name : File Title #

P00100087-AAA-X00-AAXXXXXXX-XX-0X0000_AA : Appendix Cost Report Work Package XX #

2

S2.5_DataDrop-00047

Created: 10/21/2024 1:42 PM
Exported On: 2/20/2026 1:33 PM

Description / Opis Uwagi
The federated model is not listed neither in MIDP nor in Data Drop report.
Except from P00100087-DAR-CGR-AIXXXXXXX-XX-PE-XX0002_EN, the remaining nwc files included in federated file are not provided as single model extractions and are not listed in MIDP. No native files are provided.

Assigned to / Przypisane Do

Assigned by / Przypisane Przez

Assigned to (CC) / Do Informacji

Answer / Odpowiedź
The issue for NWD file has been resolved starting from data drop#4 and subsequent data drops. as for the remaining models we understood that these refers to drainage models (MER) which are not part of Pre-design stage. The same will be further updated and reflected in the right wp (i.e WP 24) following change order 6

Additional Comments / Dodatkowy Komentarz

3

S2.5_DataDrop-00047

Created: 10/21/2024 1:42 PM
Exported On: 2/20/2026 1:33 PM

Justification of Close / Uzasadnienie zamknięcia

Close Issue? /Zamykamy

Closed by / Zamknięte Przez

Date of Closed (Last Update)
10/27/2025 4:09 PM

4

Figure 4. Exporting an IR comment to PDF. The form contains the Port of Poland logo current as of the date the comment was created, which has since changed. Source: own elaboration.

Another option for exchanging comments is the open BCF standard, which is a package of information about an issue, supported by and across many applications. The format provides complete information about a note created in the BIM process. It is the best solution for use in interdisciplinary coordination. Using the free BIMcollab BCF Manager plugin, created viewpoints are exported to the BCF format. The author can enter additional information, such as the note's priority, type, resolution deadline, and additional comments (Figure 5). The created comments are saved in the project within the BIMcollab environment, which allows for managing comments, verifying their resolution, conducting discussions, and reviewing statistics.

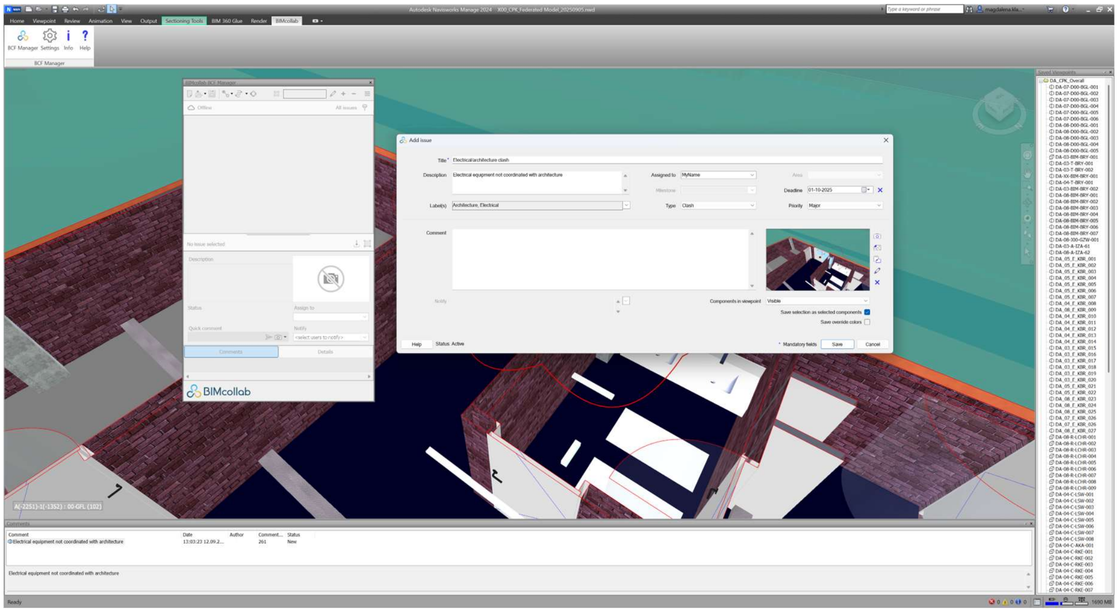


Figure 5. Autodesk Navisworks interface with the BIMcollab BCF Manager plugin – creating a comment. Source: own elaboration.

The example of projects implemented by Port Polska shows that in mega-projects, the BCF format ceases to be merely a technological curiosity and becomes a critical management resource. The use of BCF Manager-type plugins allows for the creation of an efficient feedback loop between coordination software and the proprietary tool, without the need to leave the designer's native work environment. In the context of massive-scale projects, where the number of comments runs into the thousands, automating the process of creating a reified issue by directly retrieving element GUIDs from the model eliminates human errors resulting from manual data entry, which is crucial for maintaining data integrity across the entire CDE ecosystem.

5. Discussion and Conclusions

5.1. The Lifecycle of a Reified Problem in Project Practice

The combination of reification and serialization enables a complete, easily traceable lifecycle of a coordination issue from detection to final closure, while maintaining a complete history. The process begins with the detection of a problem, which can occur automatically via a collision detection tool such as Solibri Model Checker or Navisworks, or manually by a project participant during a model review. At the moment of detection, the problem exists only as an observation in a person's mind or as the result of an algorithm, without a permanent digital representation. Reification then takes place, during which the problem is transformed into a Topic object with a full set of attributes, supplemented by a comment explaining the context and circumstances of detection, and a viewpoint enabling precise localization within the model. From this point on, the problem has an identity and structure and can be the subject of operations within the management system.

The reified object is then serialized into a bcfzip file or sent via API to a central server or CDE platform. Serialization transforms the object existing in the application's memory into a persistent record that can be stored and transmitted. The file or API message is forwarded to the responsible person or team via distribution mechanisms appropriate for the given project environment, whether via email, the CDE system, or automatic notifications from a collaboration platform. The recipient opens the file in their authoring tool equipped with BCF functionality or in a dedicated BCF manager,

 doi:10.20944/preprints202603.1498.v1

where deserialization occurs—that is, the reconstruction of objects from their saved form (Figure 6). The designer views the problem within the context of their model, can navigate to the specified viewpoint, and proceed with analysis and making changes.

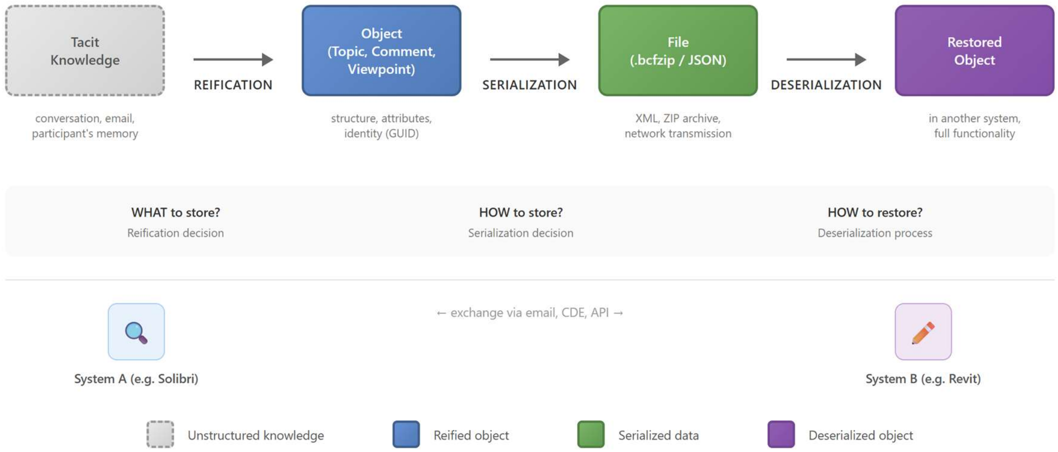


Figure 6. Flow from tacit knowledge, through reification to an object, through serialization to a file, and deserialization to an object in another system. Source: own elaboration.

After changes are made to the model, the BCF object is updated by changing its status, adding a comment describing the actions taken, and optionally adding a new viewpoint showing the state after modification. The updated object is serialized again and returned to circulation, closing a single iteration of the cycle. The cycle may repeat multiple times in the case of complex problems requiring consultation among multiple participants or iterative refinement of the solution. Ultimately, the problem is formally closed by changing the status to Closed, while retaining the full history of all comments, viewpoints, and status changes, which serves as valuable material for analysis and audits.

5.2. Practical Benefits of Reification and Serialization

Thanks to standardized reification, which defines what to record and in what structure, and serialization, which defines how to record and in what format, BCF ensures true interoperability between tools from different vendors, which is one of the pillars of the openBIM concept [16]. A task created in Solibri by the BIM coordinator can be opened in Navisworks by the site manager, commented on in BIMcollab Zoom by the MEP designer, and closed in Autodesk Revit by the architect, with all these operations taking place without loss of information and while preserving the full context. This interoperability is possible precisely because the BCF standard precisely defines both the reification model—which specifies which aspects of the problem are represented as attributes and relationships—and the serialization format—which specifies how these aspects are stored in XML files.

The reification of comments and status changes, along with precise timestamps and author identifiers, creates a complete audit trail that can be reconstructed and verified at any point in the project and building lifecycle [17]. It is possible to answer questions such as who reported the issue and under what circumstances, exactly when it was detected, who was assigned responsibility for resolving it and whether that assignment was changed, what subsequent solutions and arguments were proposed by individual participants in the discussion, who and when the issue was finally closed, and how long it took to resolve the problem from reporting to closure. This information has not only operational value during the project but also strategic value for process improvement and the resolution of potential disputes.

Reified and serialized data in a structured XML or JSON format can be processed automatically by analytical tools and reporting systems, opening up possibilities unattainable through informal

 doi:10.20944/preprints202603.1498.v1

communication [18]. It is possible to automatically generate reports on the status of issues, broken down by industry, priority, or responsible person. It is possible to create dashboards that visualize coordination progress and identify so-called bottlenecks, which aligns with the current trend of storytelling. It is also possible to configure alerts for issues that exceed the set resolution deadlines or are escalated to a higher priority. Another capability is conducting statistical analyses regarding collision types, resolution times, or the distribution of issues across industries, which provides data for improving design processes.

5.3. Transformation of Tacit Knowledge into Explicit Knowledge

Thus, BCF implements in construction practice the theoretical knowledge transformation model described by Nonaka and Takeuchi, transforming tacit knowledge into explicit knowledge that can be stored, searched, and transferred [19]. Design decisions that would normally remain in the minds of participants or in emails are captured in a structured form linked to specific model elements and a visual context. This externalization of knowledge is of fundamental importance for construction projects characterized by long lifecycles, high participant turnover, and complex interdependencies between decisions made at different stages.

When comparing issue management with and without BCF, the differences are noticeable at every level. An imprecise issue location is replaced by camera coordinates that allow for immediate navigation to the exact spot, and GUIDs for elements that enable their programmatic retrieval and analysis. Vague responsibility scattered among meeting participants is replaced by a specific person assigned in the AssignedTo field, with the ability to track assignment changes. A history scattered across dozens of email threads and meeting notes becomes an organized sequence of comments with precise dates, authors, and optional viewpoints. A status that previously required follow-up via phone or messenger becomes immediately visible in the system to all authorized participants. Manual, time-consuming, and error-prone reporting is replaced by automa tic report generation from structured data, with the ability to filter and aggregate as desired.

The BCF format serves as a modern and proven example of the application of reification and serialization in construction engineering, demonstrating how abstract IT concepts translate into concrete benefits in the management of complex projects. Reification transforms the ephemeral coordination problems inherent in informal communication and participants' memories into persistent objects with attributes, identities, and relationships that enable systematic processing. Serialization enables the preservation of these objects in formats suitable for long-term storage, secure exchange between heterogeneous systems, and automatic analysis by IT tools. The combination of both mechanisms creates the foundation for interoperable, auditable, and automatable collaboration in BIM projects, delivering measurable benefits in the form of reduced coordination time, fewer errors, and improved quality of project documentation.

Understanding the concepts of reification and serialization allows BIM specialists to better design information processes, consciously select and configure tools, and more effectively utilize available openBIM standards. This knowledge is particularly valuable when implementing CDE platforms, defining EIR (Exchange Information Requirements) and BEP (BIM Execution Plan) requirements, and training project teams in BIM methodology. The future development of the BCF standard and its integration with digital twin technologies and facility management systems will require further consideration of what information is worth reifying in the context of building operations and how to effectively serialize it for various use cases, including IoT (Internet of Things) monitoring, predictive maintenance, and energy management [20]. It can be expected that the principles of reification and serialization will find application in future buildingSMART standards, expanding the possibilities for structuring and exchanging knowledge about building objects throughout their entire lifecycle.

Thus, the BCF not only streamlines ongoing communication but transforms the individual, scattered experiences of process participants into structured and strategic knowledge capital that can be leveraged in the design of future standards and during the building's operational phase.

 doi:10.20944/preprints202603.1498.v1

Author Contributions: Conceptualization, A.S.B.; methodology, A.S.B. and M.K.; validation, A.S.B.; formal analysis, A.S.B. and M.K.; resources, M.K.; data curation, M.K.; writing—original draft preparation, A.S.B., M.K., and M.M.; writing—review and editing, A.S.B., M.K., and M.M.; visualization, A.S.B., M.K., and M.M.; supervision, A.S.B.; funding acquisition, A.S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Due to restrictions imposed by the information protection policy of the public company Port Polska (formerly Centralny Port Komunikacyjny), the research data used in this article cannot be made publicly available.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sacks, R.; Eastman, C.; Lee, G.; Teicholz, P. *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2018; ISBN 978-1-119-28753-7.
2. ISO 19650-1:2018; *Organization and Digitization of Information about Buildings and Civil Engineering Works, Including Building Information Modeling (BIM)—Information Management Using Building Information Modeling—Part 1: Concepts and Principles*. International Organization for Standardization: Geneva, Switzerland, 2018.
3. buildingSMART International. *BCF Introduction*. Available online: https://technical.buildingsmart.org/standards/bcf/ (accessed on January 26, 2026).
4. Fowler, M. *Patterns of Enterprise Application Architecture*; Addison-Wesley: Boston, MA, USA, 2002; ISBN 978-0-321-12742-6.
5. Davis, R.; Shrobe, H.; Szolovits, P. What Is a Knowledge Representation? *AI Mag.* **1993**, *14*, 17–33. https://doi.org/10.1609/aimag.v14i1.1029
6. Chen, P.P.-S. The Entity-Relationship Model—Toward a Unified View of Data. *ACM Trans. Database Syst.* **1976**, *1*, 9–36. https://doi.org/10.1145/320434.320440
7. buildingSMART International. *IFC 4.3 Documentation*. Available online: https://ifc43-docs.standards.buildingsmart.org/ (accessed on January 26, 2026).
8. Borkowski, A.S. "Digital Twin Conceptual Framework for the Operation and Maintenance Phase in the Building's Lifecycle." *Arch. Civ. Eng.* **2024**, *70*, 139–152. https://doi.org/10.24425/ace.2024.150975
9. Bloch, J. *Effective Java*, 3rd ed.; Addison-Wesley: Boston, MA, USA, 2018; ISBN 978-0-134-68599-1.
10. Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds.) *Building Information Modeling: Technology Foundations and Industry Practice*; Springer: Cham, Switzerland, 2018; ISBN 978-3-319-92861-6. https://doi.org/10.1007/978-3-319-92862-3
11. Borkowski, A.S.; Drozd, W.; Zima, K. The Status of the Implementation of the Building Information Modeling Mandate in Poland: A Literature Review. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 343. https://doi.org/10.3390/ijgi13100343
12. buildingSMART International. *BCF-XML Version 2.1*; Technical Specification. Available online: https://github.com/buildingSMART/BCF-XML (accessed on January 26, 2026).
13. Preidel, C.; Borrmann, A. Towards Code Compliance Checking on the Basis of a Visual Programming Language. *J. Inf. Technol. Constr.* **2016**, *21*, 402–421.
14. Pauwels, P.; Terkaj, W. EXPRESS to OWL for the Construction Industry: Towards a Recommendable and Usable ifcOWL Ontology. *Autom. Constr.* **2016**, *63*, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003
15. buildingSMART International. *BCF-API Version 2.1*; OpenAPI Specification. Available online: https://github.com/buildingSMART/BCF-API (accessed on January 26, 2026).
16. Grilo, A.; Jardim-Gonçalves, R. Value Proposition on Interoperability of BIM and Collaborative Working Environments. *Autom. Constr.* **2010**, *19*, 522–530. https://doi.org/10.1016/j.autcon.2009.11.003
17. Turk, Ž. Ten Questions Concerning Building Information Modelling. *Build. Environ.* **2016**, *107*, 274–284. https://doi.org/10.1016/j.buildenv.2016.08.001

18. Solihin, W.; Eastman, C.; Lee, Y.-C. Toward Robust and Quantifiable Automated IFC Quality Validation. *Adv. Eng. Inform.* **2015**, *29*, 739–756. https://doi.org/10.1016/j.aei.2015.07.006
19. Nonaka, I.; Takeuchi, H. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*; Oxford University Press: New York, NY, USA, 1995; ISBN 978-0-19-509269-1.
20. Sacks, R.; Brilakis, I.; Pikas, E.; Xie, H.S.; Girolami, M. Construction with Digital Twin Information Systems. *Data-Centric Eng.* **2020**, *1*, e14. https://doi.org/10.1017/dce.2020.16