Incremental IFC updates with 'why' and 'how': A case of IFC extension for LLM agent-enhanced simulation-based optimization for building energy efficiency

Lingming Kong, Qianyun Zhou, Fan Xue

Department of Real Estate and Construction, University of Hong Kong, Hong Kong SAR, China

u3009509@connect.hku.hk   qianyun8@connect.hku.hk   xuef@hku.hk

Abstract -

Simulation-based optimization (SBO) for energy-efficient building design exemplifies a typical design-change process that remains disconnected from open BIM-enabled collaboration. The gap stems from IFC's limited capacity to represent the information generated by SBO. To overcome the limitations, this study proposes a novel IFC schema extension for LLM Agent-enhanced SBO, named *IfcSboExtension*, that enriches incremental IFC updates with contextual metadata capturing not only what changed, but also why the change occurred and how it was derived. The proposed extension model enables exchangeable and traceable SBO iterations by embedding decision rationale and methodological provenance directly into incremental IFC updates. The feasibility and value of the extension are demonstrated through a case study on building energy efficiency optimization, in which an LLM agent enhances the interpretability of design changes.

Keywords –

Building information modelling (BIM), Industry Foundation Classes (IFC), Incremental IFC updates, Simulation-based optimization (SBO), Large language model (LLM)

1 Introduction

Building information modelling (BIM) is recognized as a digital integration of information throughout a building lifecycle, and increasingly mandated in the AEC (Architecture, Engineering, and Construction) industry [1]. BIM-based collaboration has progressively grown to level 2, namely "open BIM", which signifies a fully integrated, collaborative environment for project stakeholders through federated discipline models and open exchange standards [2]. Industry foundation classes (IFC), and ISO 16739-1:2018 standard, are *per se* open BIM that provides a vendor-neutral data format for seamless data exchange between various BIM applications [3].

However, the scalability and adoption of IFC are constrained by the monolithic, file-based exchange paradigm. In practice, BIM stakeholders receive the modified model as a new monolithic file from collaborators and check manually to identify the potential impact on their own disciplines, which is intensive in labor and prone to errors.

IFC 5, the upcoming modular version, plans to implement incremental IFC updates to address these issues [4]. From a theoretical perspective, the transition toward incremental IFC updates aligns with the process-centric knowledge philosophy [5], which emphasizes that construction knowledge is dynamic, iterative, and context-sensitive regarding knowledge dissemination among project partners. Consequently, the modular architecture of IFC 5 is expected to better support both operational efficiency and effective knowledge management in construction projects.

Simulation-based optimization (SBO) is a typical design-change process in which design parameters are iteratively adjusted, leading to modifications to model design. Recent studies have leveraged large language models (LLM) to promote the interpretability of traditional black-box SBO workflows [6]. However, the information generated by the LLM-enhanced SBO remains neither exchangeable nor traceable within the context of sustainable, collaborative design based on open BIM. To address the gap between LLM-enhanced SBO and open BIM, this paper proposes an IFC extension for LLM-enhanced SBO, which explicitly answers 'why' and 'how' in addition to 'what' to enrich such incremental IFC updates.

The remainder of this paper is organized as follows: Section 2 reviews the state-of-the-art research on incremental IFC updates and simulation-based optimization. Section 3 introduces the proposed IFC schema extension for incremental update. Section 4 demonstrates the implementation of the extension through a case study employing large language model (LLM) agents for building energy efficiency optimization. Section 5 provides the conclusion.

The primary contributions of this study are twofold. First, the proposed IFC schema extension emphasizes that the 'why' and 'how' dimensions are essential to the 'what' of incremental IFC updates. Second, the feasibility of the proposed extension was validated through a real-world project.

2 Literature Review

2.1 IFC and schemas for changes

IFC was originally conceived as a monolithic, object-oriented schema based on the EXPRESS data modelling language [7]. As an object-oriented data schema, the IFC schema contains several classes, namely "entities", and serves as "templates" to describe building information. The semantic meaning and the implementation of entities are generalized through inheritance mechanisms. Each entity declares a set of local attributes, in addition to the global attributes inherited from its supertype. There are two types of IFC entities, namely rooted and non-rooted entities, which contain or do not contain a globally unique identifier (GUID). In practice, the instantiated IFC entities are transformed into ASCII text rows in the text-based IFC-SPF (STEP Physical File) formatted file [8]. An IFC-SPF file is exchanged in a snapshot-based exchange paradigm, which means each IFC-SPF file represents a complete state of a model at a given time, regardless of actual changes. This exchange paradigm leads to inefficiencies in data transmission, version control, and change impact analysis for decision making in the collaborative, multi-disciplinary environment [9].

To overcome the limitations of holistic IFC file exchange, recent research and standardization efforts have shifted toward change-based IFC exchange approaches. In the literature, three categories of difference (or delta) computation methods have been developed to capture changes between two IFC models, which are (1) GUID-based; (2) flattening-based; and (3) graph-based approaches.

The GUID-based approach recognizes two IFC instances as identical if they share the same GUID, which is widely adopted by commercial BIM platforms such as Autodesk Revit, Navisworks and Graphisoft ArchiCAD. However, a significant limitation of this approach is that GUIDs may not remain consistent across different BIM authoring tools, rendering GUID-based comparison unreliable. The flattening-based approach eliminates randomly referenced identifier numbers (i.e., the "#"-IDs) in IFC files. The "flattening" process is to replace all the "#"-IDs with specific instant IDs, such as content-based hash ID [10][11]. Therefore, flattening-based approaches enable deterministic textual comparison.

Both the GUID-based and flattening-based methods are fundamentally text-oriented. In contrast, the graph-based approach leverages graph-based algorithms to compute semantic differences between two graph-represented IFC models. There are two main steps involved in the graph-based approach. First, the graph representation of IFC models is developed, such as RDF (resource description framework) graph and directed property graphs. Then, graph-based algorithms, such as graph transformation [12], are implemented to capture the changed part of two IFC models.

2.2 Knowledge engineering applications of changes in construction

The dynamic theory of knowledge creation developed by Nonaka and Takeuchi [13], namely the SECI model, posits that knowledge is created, transferred, and applied through the creative tension between tacit and explicit know-how. The SECI model reveals a significant source of knowledge, that knowledge is generated from continuously combined, refined, and reconfigured explicit assets. This definition aligns with the iterative development processes of construction projects [1]. Accordingly, changes in a construction project, such as design iterations, site conditions, and stakeholder feedback, are recognized as a knowledge source which has great influence on successful project delivery [14].

Knowledge engineering (KE) is the process of designing, building, and maintaining knowledge-based systems by identifying relevant domain knowledge, representing it in formal structures (such as ontologies, or knowledge graphs), and implementing reasoning mechanisms that can use the knowledge to make inferences, decisions, or recommendations [15]. A core KE approach is the development of domain ontologies that explicitly define change types, actors, causes, and impacts. Early work by Beetz et al. [16] transforms the EXPRESS schema of IFC into an OWL ontology, namely IfcOWL, and paves the way to change detection as ontology alignment. The following work of Oraskari and Törmä [17] introduces a signature algorithm for RDF diff computation. Furthermore, a domain ontology for pumped storage hydropower system was developed by Zhang et al. [18] to support design changes management.

IFC also can be serialized into various data structures, such as ifcJSON and labelled property graph, to support algorithmic reasoning for change detection. For example, A semantic differential transaction (SDT) approach and the following traceable SDT approach were developed based on the JSON-format IFC to capture the semantic changes through structure matching [10][11].

2.3 Simulation-based optimization as a typical change

Simulation-based optimization (SBO) is widely applied to building design by iteratively tuning design parameters towards sustainable objectives [19]. The generic SBO workflow comprises iterative simulations that are evaluated through objective functions defined in optimization algorithms. SBO constitutes a typical design-change process in which the design parameters vary across iterations, producing corresponding modifications in design. For example, energy performance optimization is a key task in SBO for early stage building design, aiming to reduce the environmental impact of future buildings. SBO involves tuning design parameters such as window-to-wall ratio and building materials. In each iteration, these parameters are adjusted to gradually improve energy performance.

Traditional SBO is challenged by high computational complexity and low interpretability. A recent study by Zhou and Xue [19] introduces an Automatic Information Gain-guided Convergence (AIGGC) method to refine critical design parameters through interpretable information gain analysis. Subsequent research integrated the large language model (LLM) to further enhance the interpretability and efficiency of complex SBO [6].

However, the information produced by LLM-Agent enhanced SBO is neither exchangeable nor traceable within the context of sustainable, collaborative design based on open BIM. Existing literature characterizes BIM-supported SBO as a one-way workflow that portions of the required parameter data are extracted from BIM [20], yet the SBO iteration results are not embedded back into BIM for collaborative use. The limitation arises for two reasons. First, contemporary SBO practices are predominantly implemented on visual programming platforms such as Grasshopper, which do not natively support IFC. Secondly, the IFC schema lacks the capacity to represent the information generated by LLM-Agent enhanced SBO, such as design parameters, constraints, and convergence analysis. The proposed IfcSboExtension uniquely encode the decision rationale, optimization provenance, and LLM interaction context to support change-based IFC exchange for traceable and interpretable collaborative design.

3 IFC extension for LLM agent-enhanced SBO

To extend IFC's capability of representing information of LLM Agent-enhanced SBO, an IFC extension model named '*IfcSboExtension*' was proposed. The extended schema illustrates 'what' data should be involved, 'why' the IFC schema should be extended, and 'how' to capture the incremental IFC changes within the LLM Agent-enhanced SBO. A general IFC extension workflow is illustrated in Figure 1, which constitutes three steps: (1) Information requirements in LLM Agent-enhanced SBO, (2) *IfcSboExtension* model development, and (3) verification and implementation for incremental IFC updates.

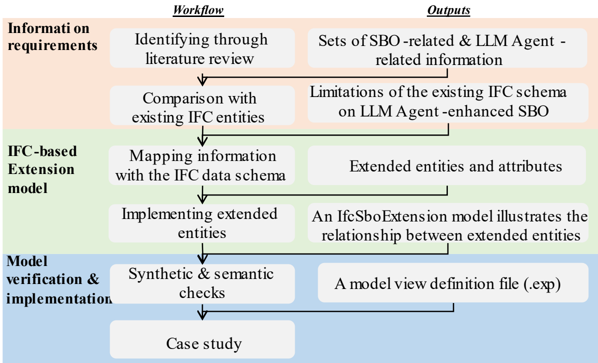

Figure 1. IFC extension workflow

The first step answers 'what' and 'why' the IFC schema should be extended to LLM Agent-enhanced SBO. Table 1lists the crucial data of two categories: (1) SBO-related data such as design parameters and optimization constraints, as well as (2) LLM Agent-related data such as prompts and structured responses.

A meticulous comparison with the existing entities within IFC4 ADD3 revealed that both SBO-related and LLM Agent-related information required extension to enable seamless data exchange. For example, *IfcProcess* is defined in existing IFC schema to represent one individual activity that has sequence relationships with other processes. Three subtypes of *IfcProcess*, namely *IfcTask*, *IfcEvent*, and *IfcProcedure* do not contain the semantics such as parameters, constraints, and objective functions. One compromise approach supported by IFC schema is identifying specific property sets, such as *Pset_SBO*. However, this approach converts SBO semantics to unstructured strings and leads to semantic ambiguity and semantic loss.

3.1 The *IfcSboExtension* model

According to the required data listed in Table 1, three rooted entities and two non-rooted entities were extended align with the IFC4 ADD3 schema. As Figure 2 shows, the extended entity *IfcSboWorkflow* and *IfcAiAgentInteraction* are two subtypes of the abstract *IfcControl*, and *IfcOptimizationIteration* is a subtype of the abstract *IfcProcess*. The reason lies in the semantic definition of *IfcControl* and *IfcProcess*. In IFC4 ADD3, *IfcControl* represents all concepts that control or constrain the utilization of products, processes, or resource, whereas *IfcProcess* is defined as one individual activity or event. According to the definition, *IfcControl* and its subtypes are suitable for representing a workflow, whereas *IfcProcess* and its subtype focus on one specific process. In LLM Agent-enhanced SBO, SBO is a workflow that contains a set of 'simulation-optimization' iterations, and the interaction between stakeholders and LLM Agent is also recognized as a set of communications rather than single process. One optimization iteration aligns with the definition of *IfcProcess* which transforms input in output. *IfcObjectiveFunction* and *IfcSboOptimizationDirectionEnum* are two non-rooted entities extended at the resource layer. These two non-rooted entities cannot exist independently but can only exist if referenced by the rooted entity *IfcSboWorkflow*. Due to page limits, only the extended *IfcSboWorkflow* entity and relevant attributes is present in Figure 3, which illustrates the EXPRESS representation of *IfcSboWorkflow* to ensure both machine and human readability.

Table 1. Crucial data in LLM Agent-enhanced SBO

	Crucial Data	Data type (in IFC)	Example
SBO-related	SBO parameters	SET [1: *n*] of REAL	(0.45,0.45,1.1,2,5.0733)
	SBO constraints	STRING	$\left\|FAR_{ori} - FAR_{opt}\right\| \leq 0.05$ (FAR for floor area ratio)
	SBO objectives	STRING	EUI = 112.684 $kWh/m^2 \cdot yr$
	SBO results	STRING or BINARY	JSON or CSV
LLM Agent-related	LLM prompts	STRING	*"Please identify which design parameters most influence the f(S)?"*
	LLM responses	STRING or BINARY	Response text or Log file

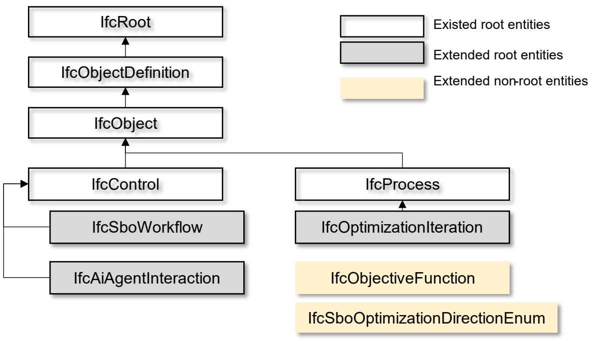

Figure 2. Extract of the IFC LLM Agent-enhanced SBO extension with new entities

3.2 Model verification

For verification, the official IFC documentation toolkit *IfcDoc* and a test software platform APSTEX were employed. The verification procedure involves two steps: syntactic verification and semantic verification. As summarized in Table 2, syntactic verification ensures compliance with the formal grammar of the EXPRESS language. The extended IFC entities were documented using *IfcDoc* to verify their compliance with EXPRESS language grammar. The exported use-case model view definition, designated as *IfcSboExtension*, incorporates 19 existing IFC entities and 5 newly defined IFC entities.

In the second stage, semantic verification was performed by compiling the extended *IfcSboExtension* model into Java classes using the Apstex IFC Framework. Additionally, accessor methods of explicit, derived, and inverse attributes of the extended IFC entities are developed. Critically, domain rule enforcement methods were developed, and corresponding unit tests were designed to verify the *IfcSboExtension* model based on the created Java classes.

```
ENTITY IfcSboWorkflow
  SUBTYPE OF (IfcControl);
    OptimizationType : IfcLabel;
    AlgorithmName   : OPTIONAL IfcLabel;
    StartTime       : OPTIONAL IfcDateTime;
    EndTime         : OPTIONAL IfcDateTime;
    Status          : IfcLabel;
INVERSE
    HasVariables    : SET [1:?] OF IfcPropertyBoundedValue FOR BelongsToWorkflow;
    HasObjectives   : SET [1:?] OF IfcObjectiveFunction FOR BelongsToWorkflow;
    HasIterations   : SET [0:?] OF IfcOptimizationIteration FOR BelongsToWorkflow;
    GeneratedFromLog : OPTIONAL IfcAiAgentInteraction FOR TriggersWorkflow;
END_ENTITY;
```

Figure 3. The extended IfcSboWorkflow entity and relevant attributes

Table 2. Model verification procedure, criteria, and tools

Verification procedure	Criteria	Tools
Syntactic checks	**Formal compliance with the grammar defined in ISO 10303-11(EXPRESS):**	*IfcDoc*

	• Notation conventions **EXPRESS-compliant declaration of** • Entities ("entity data types") and types ("defined data types"), • Enumerations ("enumeration data types"), • Subtype and supertype constraints, • Explicit, derived, and inverse attributes of entities	
Semantic checks	**Compilation of the LLM Agent-enhanced SBO schema into Java classes for semantic checking:** • Java classes for entity, type, enumeration considering the inheritance hierarchy • accessor methods for explicit, inverse, and derived attributes • methods to check domain rules and uniqueness rules	Apstex IFC Framework

```
public void whereRuleCheck(WhereRuleCheckResultList list) {
    super.whereRuleCheck(list);
    List<IfcRelAssignsToSboWorkflow> rels =
this.getInverse("HasObjectives");
    boolean applicableObjectivesCheck = true;
    if (rels != null) {
        for (IfcRelAssignsToSboWorkflow rel : rels) {
            List<Object> relatedObjects = rel.getRelatedObjects();
            if (relatedObjects != null) {
                for (Object obj : relatedObjects) {
                    If
(!BUILTINFUNCTIONS.TYPEOF(obj).contains("IFCSBOMODEL.IFCSBOOB
JECTIVEFUNCTION")) {
                        applicableObjectivesCheck = false; break; } } } } }
    if (!applicableObjectivesCheck) {
        reportError(list, "IfcSboWorkflow: ApplicableObjectives");
    }
}
```
(a)

```
// create a new SBO workflow entity
IfcSboWorkflow ifcSboWorkflow = new IfcSboWorkFlow();
// create a relationship entity
IfcRelAssignsToControl ifcRelAssignsToControl = new
IfcRelAssignsToControl();
ifcRelAssignsToControl.setRelatingContnrol(ifcSboWorkflow)
SET<IfcObjectDefinition> RelatedObjects = new SET<>();
ifcRelAssignsToControl.setRelatedObjects(RelatedObjects);
// add SBO workflow components to related objects
RelatedObjects.add(new IfcObjectiveFunction());
// add module group to SBO workflow must cause an error
RelatedObjects.add(new IfcGroup());
// execute the domain rule check
WhereRuleCheckList list = new WhereRuleCheckList();
ifcSboWorkflow.whereRuleCheck(list)
```
(b)

Figure 4. Examples of domain rule checking method and unit test for "HasObjectives" of the IfcSboWorkflow entity

Figure 4 presents an example of a domain rule checking method and its associated unit test. For efficient compilation, so-called "lambda expressions" were used to implement the domain rule checks. The example 'whereRuleCheck' is defined to check the applicable objectives of the extended entity *IfcSboWorkflow*. Corresponding unit test is shown in Figure 4(b). "HasObjectives" of the *IfcSboWorkflow* entity is shown in Figure 3, which represents a set of *IfcObjectiveFunction* instances are assigned to one *IfcSboWorkflow* instance. In the unit test, the process of adding a module group to *IfcSboWorkflow* results in a 'False', which indicates an incorrect state of the IFC test model.

4 A case study of LLM agent-enhanced SBO

To illustrate 'why' and 'how' in addition to 'what' of IFC incremental updates, an energy consumption optimization scenario is selected as a representative design-change case study. The design optimization of a modular integrated construction (MiC) project for a university student hostel located in Hong Kong is selected as a case. The developed LLM Agent for efficient and interpretable SBO is presented in the published work of Zhou and Xue (2025).

4.1 Case settings

In this case, the optimization objective was to minimize the energy use intensity (EUI), defined as the total annual energy consumption divided by the gross floor area ($kWh/m^2 \cdot yr$). As shown in Figure 5, the project features a symmetrical layout with fixed modules and movable modules. The optimization process focuses on three types of movable modules which vary in length, namely M_A, M_B, and M_C. Thus, module layout, window size and material properties are involved as design parameters shown in Table 3.

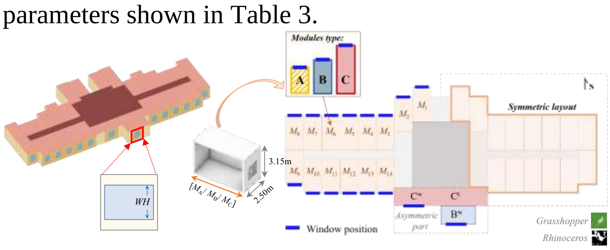

Figure 5. MiC case layout and schematic of module types (adopted from Zhou & Xue (2025))

4.2 Results

4.2.1 Schemas and sample changes

The implementation of the *IfcSboExtension* model using *IfcDoc* is presented in Figure 6 as an EXPRESS-G diagram. Black rectangles represent rooted entities, whereas grey rectangles denote non-rooted entities. Entity's attributes – comprising own and inherited attributes - are separated by an ellipsis and categorized as either explicit or derived attributes (in black font) or inverse attributes (in light grey font). Relationships between the extended entities are connected through inverse attributes. For example, an *IfcOptimization-Iteration* is referenced by an *IfcSboWorkflow* through its inverse attributes '*HasIterations'*.

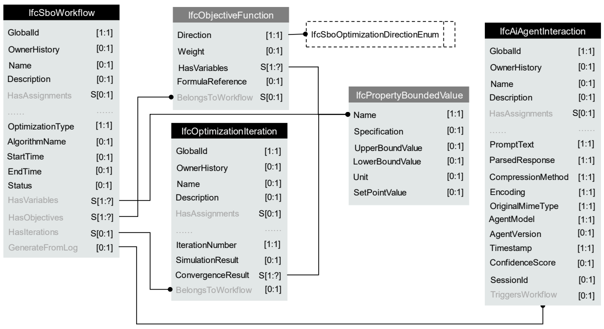

Figure 6. EXPRESS-G representation of *IfcSboExtension* model

The embedded LLM Agent-enhanced SBO semantics is illustrated in Figure 7. *IfcSboWorkflow* is assigned to a module through *IfcRelAssignsToGroup*, and *IfcAiAgentInteraction* is assigned to *IfcSboWorkflow* through *IfcRelAssignsToControl*. To illustrate the change of parameters during SBO iterations, parameters listed in Table 3 are selected. For instance, when an iteration increases the north window area from 4.33 m² to 4.49 m², embedded convergence semantics allow the owner to immediately identify this change as the primary driver for reducing Energy Use Intensity. This transparent exposure of the 'why' (energy optimization) and 'how' (sensitivity analysis) enables the owner to grasp the technical rationale instantly, facilitating an informed approval decision.

4.2.2 Sensitivity analysis

The text-based IFC-SPF file exhibits acute sensitivity to: (1) excessive line lengths, (2) special characters such as unescaped Unicode or JSON brackets, and (3) exponential file growth. Specifically, the extended *IfcAiAgentInteraction* records prompt and structured responses, which may comprise multi-megabyte strings, that can significantly decrease data exchange efficiency.

In this study, two mitigation strategies were evaluated: (1) compressing the structured response using the Gzip algorithm and encoding the compressed result as Base64 strings; and (2) storing LLM Agent prompts and structured responses in external files referenced via *IfcURIReference*. Gzip and Base64 are open standards for data compression and encoding, respectively. Metadata describing the compression and encoding algorithms are embedded in *IfcAiAgentInteraction* to ensure self-descriptiveness. URIs for external files should conform to standard URI schemes (e.g., HTTP and HTTPS), and the corresponding MIME type should be explicitly specified.

The evaluation was performed over 30 iterations of LLM Agent-enhanced SBO, with results presented in Table 4. The result indicates that the 'Gzip + Base64' strategy increased the IFC file size by a maximum of 2.47% and an average of 1.91%. The *IfcURIReference* strategy effectively eliminates the impact of LLM Agent responses on IFC file size.

Table 3. Design parameters for MiC parametric design

Design parameters	Parameter type	Unit	Parameter domain intervals	Precision
MiC Module Type	Categorial	-	$[M_A, M_B, M_C]$	-
Window-to-Wall Ratio (WWR_N, WWR_S)	Continuous	-	$0.10 \leq WWR_N, WWR_S \leq 0.60$	0.01
Window Height (WH_N, WH_S)	Discrete	M	$1.0 \leq WH_N, WH_S \leq 2.5$	0.1
Window U-value (U_w)	Continuous	$W/(m^2 \cdot k)$	$0.50 \leq U_w \leq 5.78$	0.01
Window Solar Heat Gain Coefficient (g)	Continuous	-	$0.100 \leq g \leq 0.775$	0.001
Window Visible Transmittance (τ_v)	Continuous	-	$0.300 \leq \tau_v \leq 0.881$	0.001

4.2.3 User acceptance and comments

The extended *IfcSboExtension* model was implemented on a real MiC project for a university student hostel located in Hong Kong. Stakeholders of the MiC project gave positive comments. For example, the owner of the project claimed that "The LLM Agent platform using extended IFC has transformed our design optimization. The black box SBO is interpretable, traceable and exchangeable through open BIM. This enables non-professional stakeholders to understand the design intentions and make decisions. Technical users emphasized that "Through the adoption of open BIM, we can work smoothly between different software and disciplines with an open standard, which helps us streamline and automate many manual processes in LLM Agent and SBO. This has greatly increased our productivity by eliminating misunderstanding of design changes that happened in the black box SBO process.

Figure 7. Embedded LLM Agent-enhanced SBO semantics in IFC

Table 4. File size comparison of two strategies

Strategy	Original	Gzip & Base64		External Ref.	
	(KB)	(KB)	Δ (%)	(KB)	Δ (%)
Minimal	1939	1946	+0.36	1940	+0.05
Maximum	1939	1987	+2.48	1941	+0.10
Average	1939	1976	+1.91	1941	+0.10

5 Discussion and Conclusion

The snapshot-based exchange paradigm of IFC introduces inefficiencies in data transmission, version control, and change impact analysis for making decisions in the collaborative, multi-disciplinary environment. To address these limitations, a modular version of IFC 5 is under development to support incremental IFC updates.

Simulation-based optimization (SBO) for energy-efficient building design exemplifies a typical design-change process which remains disconnected from open BIM-supported collaboration. The gap stems from IFC's poor capability of representing information generated by SBO. To bridge the gap and illustrate the full provenance of such design changes, including the 'why', 'how', and 'what' in incremental IFC updates, this study proposes a novel IFC schema extension for the LLM Agent-enhanced SBO.

The contributions of this study are twofold:

(1) An *IfcSboExtension* model was proposed and verified to enhance the IFC's capacity to represent the LLM Agent-enhanced SBO process.

(2) The feasibility and practical value of the proposed extension were validated through a MiC case study.

The case study yields two key findings:

(1) All information generated by the LLM Agent-enhanced SBO is embedded within the extended IFC entities, involving three rooted IFC entities: *IfcSboWorkflow*, *IfcAiAgentInteraction*, and *IfcOptimizationIteration*; and two non-rooted IFC entities: *IfcObjectiveFunction* and *IfcSboOptimizationDirectionEnum*.

(2) The changes of design parameters throughout the LLM Agent-enhanced SBO process can be captured as incremental IFC updates.

Nevertheless, several limitations remain that require further investigation:

(1) The extended *IfcAiAgentInteraction* faces a trade-off between semantic integrity and storage efficiency. If LLM Agent prompts and responses are fully encoded as structured data within *IfcAiAgentInteraction*, individual lines in the IFC file may exceed parser limits, resulting in increased file size and reduced exchange efficiency. Conversely, compressing the data (e.g., via Gzip) or referencing external files avoids file bloat, but incurs semantic opacity, requires stakeholders to employ additional tools to interpret the compressed string or external files.

(2) Semantic changes of the extended IFC entities

are not supported by the mainstream commercial BIM platforms (e.g., *Revit*) or open-source IFC parsing tools (e.g., *IfcOpenshell*). This limitation arises because the proposed IFC extension has not yet been submitted to buildingSMART for formal standardization – a necessary step for broader adoption.

Acknowledgement

The work presented in this paper was supported by the Hong Kong Research Grants Council (Nos. C7080-22GF and 17200221).

References

[1] C. Eastman. BIM handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. John Wiley & Sons, 2011.

[2] M. C. Georgiadou. An overview of benefits and challenges of building information modelling (BIM) adoption in UK residential projects. *Construction Innovation*, 19(3):298-320, 2019.

[3] Y. Yu, S. Kim, H. Jeon, and B. Koo. A Systematic Review of the Trends and Advances in IFC Schema Extensions for BIM Interoperability. *Applied Sciences*, 13(23):12560, 2023.

[4] buildingSMART International. IFC 5 Core Consultation. Online: https://www.buildingsmart.org/ifc-5-core-consultation/, Accessed: 01/25/2026.

[5] G. Mentzas, D. Apostolou, R. Young, and A. Abecker. Knowledge networking: a holistic solution for leveraging corporate knowledge. *Journal of Knowledge Management*, 5(1):94-107, 2001.

[6] Q. Zhou and F. Xue. L-SCOPE: An LLM-assisted interactive platform for efficient building design optimization. *Proceedings of the International Conference on Smart and Sustainable Built Environment (SASBE).* 96-105. https://isasbec.abc2.net/index.php/sasbe/article/view/2554/9

[7] International Organization for Standardization. ISO 16739-1:2024 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries Part 1: Data schema. Online: https://www.iso.org/standard/84123.html, Accessed: 01/30/2026.

[8] H. Liu, G. Gao, and M. Gu. A parallel IFC normalization algorithm for incremental storage and version control. *30th International Workshop on Intelligent Computing in Engineering (EG-ICE)*, pages 511-520, 2023.

[9] K. Dietmair, S. Esser, and B. Harder. Incremental model updates in BIM: a critical analysis of the collaboration platform Speckle. *Proceedings of the 35th Forum Bauinformatik 2024*, 2024.

[10] F. Xue and W. Lu. A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration. *Automation in Construction*, 118:103270, 2020.

[11] L. Kong, R. Zhao, C. Anumba, W. Lu, and F. Xue. Open BIM exchange on Blockchain 3.0 virtual disk: A traceable semantic differential transaction approach. *Frontiers of Engineering Management*, 12:510-528, 2025.

[12] S. Esser, S. Vilgertshofer, and A. Borrmann. Graph-based version control for asynchronous BIM collaboration. *Advanced Engineering Informatics*, 53:101664, 2022.

[13] I. Nonaka, and H. Takeuchi. *The Knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press, 1995.

[14] H. Wang, and X. Meng. BIM-supported knowledge management: Potentials and expectations. *Journal of Management in Engineering*, 37(4):04021032, 2021.

[15] E. Feigenbau. Knowledge engineering. *Annals of the New York Academy of Sciences*, 426(1):91-107, 1984.

[16] J. Beetz, J. Leeuwen, and B. de Vries. IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Ai Edam*, 23(1):89-101, 2009.

[17] J. Oraskari, and S. Törmä. RDF-based signature algorithms for computing differences of IFC models. *Automation in Construction*, 57:213-221, 2015.

[18] S. Zhang, S. Zhang, H. Liu, X. Wang, Z. Zhao, C. Wang, and L. Yan. Integration of BIM and ontologies for pumped storage hydropower design change management in EPC projects. *Automation in Construction*, 175:106189, 2025.

[19] Q. Zhou, and F. Xue. Automatic information gain-guided convergence for refining building design parameters: Enhancing effectiveness and interpretability in simulation-based optimization. *Building and Environment*, 275:112788, 2025.

[20] Y. Shen, and Y. Pan. BIM-supported automatic energy performance analysis for green building design using explainable machine learning and multi-objective optimization. *Applied Energy*, 333:120575, 2023.