# A Method to Unify Custom Properties in IFC to Linked Building Data Conversion

Jyrki Oraskari[1,*,†], Lukas Kirner[1,†], Marit Zöcklein[2,†] and Sigrid Brell-Cokcan[1,†]

[1] *Chair for Individualized Production, RWTH Aachen University, 52074 Aachen, Germany*
[2] *Center Construction Robotics, 52074 Aachen, Germany*

**Abstract**
The issue of inconsistent property namings and classifications during the export of building element properties from a design program is a common occurrence that significantly hampers the interoperability of created data models. We study this with a use scenario that involves the transportation of a steel beam, and the challenges of inconsistent property namings are highlighted. A generic method for unifying properties is proposed when IFC formatted building models are converted into semantic web and how to share the user-defined and software-specific property mappings. We also discuss the benefits of mapping the namings into buildingSMART Data Dictionary definitions and the limitations of this approach.

**Keywords**
IFC, Linked Building Data, property, bSDD

## 1. Introduction

Linked Data is an attractive approach to address many data-related challenges that prevent increased robotic automation in the construction industry. The construction industry is complex in structure, with many different sources of information creating a heterogeneous data landscape, making interoperability a prerequisite for successful collaboration [1]. Construction robotics adds new requirements to this situation, as knowledge and context are needed to cope with a construction site's unstructured and highly dynamic environment. Our strategy to address these requirements involves the integration of Linked Data methodologies, forging connections between emerging approaches and established concepts such as Linked Building Data (LBD) and the Core Ontology for Robotics and Automation (CORA) [2]. The challenge of disparate property namings and classifications when exporting building element properties from design software is a widespread issue that obstructs the interoperability of generated data models. Therefore, the research questions that guide our work are as follows.

- How to handle and align user-defined and software-specific property sets?

*LDAC 2024: 12th Linked Data in Architecture and Construction Workshop, June 13–14, 2024, Bochum, Germany*
*Corresponding author.
† These authors contributed equally.
oraskari@ip.rwth-aachen.de (J. Oraskari); kirner@ip.rwth-aachen.de (L. Kirner); zoecklein@construction-robotics.de (M. Zöcklein); brell-cokcan@ip.rwth-aachen.de (S. Brell-Cokcan)
https://www.ip.rwth-aachen.de/ (J. Oraskari); https://www.ip.rwth-aachen.de/ (L. Kirner); https://construction-robotics.de/ (M. Zöcklein); https://www.ip.rwth-aachen.de/ (S. Brell-Cokcan)
0000-0002-4723-3878 (J. Oraskari); 0000-0001-9753-2422 (L. Kirner); 0009-0007-3098-5663 (M. Zöcklein); 0000-0003-1463-7515 (S. Brell-Cokcan)

- How can we overcome inconsistent property namings between domain-specific software to enable generalised queries?

In the context of linked data, alignment refers to the establishment of correspondences between entities. We establish correspondences between RDF properties so that IFC-to-LBD conversion can provide equivalent but standard properties in the produced graphs. This work analyses the general problem with a specific use scenario that stems from an intralogistics use case developed by domain experts in the EConoM research project[1]. The aim is to investigate how construction robots can automatically transport components on-site. As is generally the case for automated machines and robots, it is crucial to know the robot's capabilities concerning physical limitations to ensure the safety of all involved and to protect the machines and their environment from damage. When it comes to material handling, such as lifting or carrying, a typical constraint is the maximum weight a machine or a robot can handle. While IFC, IfcQuantityWeight, serves as a standard way to express the weight of building elements; nonstandard properties are needed to express the extra weight of pallets, stretch wraps, tarpaulins, strappings, or crates. In this use case, a mobile robot platform is assigned to transport a building element from a delivery area to its designated storage, as illustrated in Figure 1. To check whether the autonomous robot can perform the transportation task safely, it must be checked if the cargo weight exceeds the machine limits. There are beams from two providers that have designed the beams in different programs. The robot needs to get the weight of the building element from the data models to know which beams to select. Assuming interoperable BIM-based construction planning, the

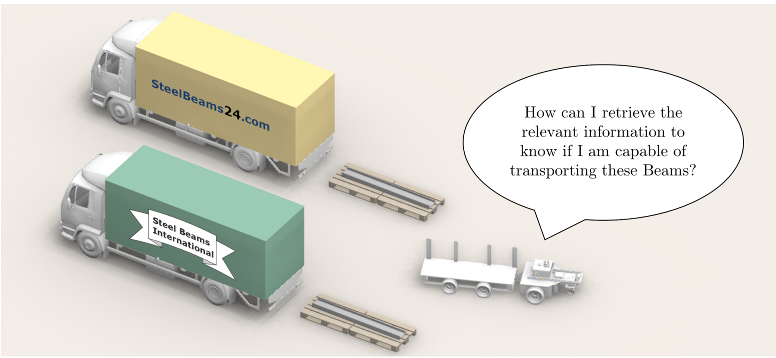


**Figure 1:** The intralogistics use case

weight of the building elements should be provided in the IFC. Here, the information is located in property or quantity sets, which can, in theory, be converted to Ressource Description Format (RDF) using the existing practices and ontologies of the LBD community. However, in practice, information is missing or inconsistently described in the resulting LBD graph, depending on the authoring software.

Section 2 provides an overview of related work in the field. In Section 3, we dive into a practical intralogistics use scenario, wherein a steel beam is modelled using two distinct software systems, presenting the issue of inconsistent property naming. Section 4 presents a

[1]https://www.econom.one/

novel solution to address this challenge, accompanied by a presentation of results. Finally, the article concludes with a discussion and conclusion of the findings and their implications. In this paper, we use the prefixes of the namespaces provided by the http://prefix.cc/ site and, in general, the namespaces are as in Listing 1

```
PREFIX beo: <https://pi.pauwel.be/voc/buildingelement#>
PREFIX inst: <https://www.econom.one/>
PREFIX props: <http://lbd.arch.rwth-aachen.de/props#>
PREFIX opm: <https://w3id.org/opm#>
PREFIX schema: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

**Listing 1:** Namespaces in the listings.

## 2. Related work

buildingSMART[2] is a global organisation that develops standards to improve collaboration and digital BIM workflows. The buildingSMART Data Dictionary (bSDD), provided by buildingSMART International, is an online repository for BIM-related classes and properties. It offers a user-friendly search interface and a management portal that allow authors to publish content, and it also allows integrated BIM software through APIs. Earlier work to integrate this with Linked Data is presented in [3]. Then, Alexiev et al. [4] proposed structural changes in bSDD focussing on GraphQL in [4]. IFC (Industry Foundation Classes) (defined by ISO/PAS 16739 [5]) is an open standard format used in the architecture, engineering, and construction (AEC) industry to exchange building information. On the one hand, there are statically defined properties (e.g. IfcDoorPanelOperationEnum) that are predefined within the IFC specification. On the other hand, dynamically extendable property sets provide a kind of meta-model to be further declared by agreement. These are IfcQuantitySets for quantitative measurements and IfcPropertySets for descriptive properties associated with an object. Linked Building Data (LBD) entails publishing building-related data as Linked Data [6, 7]. Tim Berners-Lee envisioned the fundamental principles of Linked Data in 2009 [8]. As quintessence, they promote the use of HTTP URIs as unambiguous names for things. Furthermore, meaningful information should be provided when looking up a URI. This information should be expressed according to standards such as RDF and SPARQL. In addition, additional links to other related URIs should be included for further exploration. Correspondingly, when converting an IFC data model to RDF using the existing ontologies of the W3C Linked Building Data Community Group, the challenge is to map the data so that each element's attribute and associated properties are expressed consistently. One of the first tools for the conversion of Industry Foundation Classes (IFC) to Resource Description Framework (RDF) is IFCtoRDF[9]. This tool enables the conversion of IFC STEP models into ifcOWL[10] statements. Subsequently, in 2018, IFCtoLBD [11] was developed within the Linked Building Data community. It aims to extract core Building Topology Ontol-

[2]https://www.buildingsmart.org/

ogy (BOT) classes and their relationships and incorporate product data with property values expressed using the Object Process Methodology (OPM) ontology. The article explains how RDF properties are created based on the IFC property names. NIRAS IFC2BOT [3], authored by Mads Rasmussen, is a lightweight command-line tool implemented in Python 3.8. Leveraging the IfcOpenShell Python library, it extracts core BOT elements from an IFC model. Additionally, IFC-LBD [4] from the same author is another tool in this domain that utilises the IFC.js library. BIMSO [12] is an ontology for expressing and organising knowledge about building information. It defines the material, geometric, performance, functional, and relationship properties of building elements. Although there is no explicit one-to-one mapping between IFC and BIM Shared Ontology (BIMSO) elements or alignment to the current LBD ontologies, it does not provide a direct solution to unifying LBD properties. The Ontology for Property Management (OPM) [13] facilitates the monitoring of the evolution of property over time. However, it lacks explicit definitions regarding how the properties of an element are connected with the building elements. The current solution has been to create a property using a mapping rule, which has led to the use of property URLs that are not accessible online. This contradicts the Linked Data principles defined to enable interoperability through the findability and accessibility of information. However, it also does not contain a solution for unifying properties with the same semantics. Furthermore, Bonduel et al. [14] have studied how to attach geometry to LBD-based building elements.

## 3. The Intralogistics Use Case

Following the scenario-based design [15], the described intralogistics use case is incorporated to develop our approach by modelling an IPE 240 steel beam as an element to be transported. In construction practice, it can be observed that subcontractors utilise a diverse range of specialised software tools. Unfortunately, many software solutions use different properties and names for the same information. Even if the same building element is modelled, this can occur. To simulate this, the steel beam is modelled in two different software, Tekla Structures [5] and Autodesk Revit 2024 [6]. Both are widely used in steel construction.

### 3.1. Initial state

In this example, Tekla Structures annotates the value using the quantity set QTO_BeamBaseQuantities and the property *GrossWeight*. In Revit, the weight information property is in a custom property set called *ElementWeight*. Revit does not directly export weights, so custom parameter sets are common. When the models were converted to RDF, the IFCtoLBD converter [11, 16] used the LBD props ontology family[7] namespace but created the RDF properties algorithmically. In the initial test, IFCtoLBD did not export the quantity models, so the values of Tekla structures were missing. IFCtoLBD was modified to handle IfcElementQuantity elements in two

[3]https://github.com/NIRAS-MHRA/IFC2BOT
[4]https://github.com/LBD-Hackers/IFC-LBD
[5]https://www.tekla.com/products/tekla-structures
[6]https://www.autodesk.com/products/revit/
[7]http://lbd.arch.rwth-aachen.de/props#

phases. First, it collects the references to IfcElementQuantity in the to-be-converted ifcOWL data model (Abox). It gets the name for the quantity set using the RDF property path relative to the IfcElementQuantity instance $ifc{:}name\_IfcRoot/express{:}hasString$. The definitions of single quantities are obtained using the relative path $ifc{:}quantities\_IfcElementQuantity$. Then, due to the rich naming in ifcOWL, the name definition is retrieved with a query relative to a single quantity definition using the regular expression and path $*name\_*/express{:}hasString$ and the actual value with $*Value\_*/$ with separate handlings for different data types. Finally, in the second phase, the BIM elements are linked to the associated quantity sets by URI using a relative RDF path $!ifc{:}relatedObjects\_IfcRelDefinesByProperties/ifc{:}relatingPropertyDefinition\_IfcRelDefinesByProperties$ from the BIM element. The sample data has been published in the jyrkioraskari/props_demo repository[8] on GitHub. The repository offers the IFC models and associated Linke Building Data models. testmodel_revit.ttl presents a sample Autodesk Revit model, and testmodel_tekla.ttl has an equivalent Telka Structures model.

### 3.2. Quantity Sets Added

After modifying IFCtoLBD to export element quantities, the weight values for both software can be found in the resulting graph. However, the naming conventions used in different software were not identical, resulting in different RDF object properties (see Figure 2 - red encircled). Practically, this means that a query for the weight written for the Tekla export (see Listing 2) will not return the same information when applied to the export from Revit, or more precisely, it will not return anything. Revit would require its own "dialect" or a "user-defined dialect" for the query to function correctly. When using Autodesk Revit, the naming can vary since custom property sets are used. This means that SPARQL queries are also customised. The example custom property *ElementWeight* cannot be queried with Listing 2.

```
select ?Beam ?Weight where {
   ?Beam a beo:Beam.
   ?Beam ?props ?Property.
   ?Property rdfs:label "Qto_BeamBaseQuantities:grossWeight";
    opm:hasPropertyState ?PropertyLv3.
 ?PropertyLv3 schema:value ?Weight.
}
```

**Listing 2:** SPARQL query for the Tekla exported RDF file.

## 4. Results

We exported the data models to LBD using the Open BIM approach and Industry Foundation Classes (IFC). Since IFC is a buildingSMART standard, we searched for a solution to express the weight value in the use case into a bSDD equivalent value. They have

[8]https://github.com/jyrkioraskari/props_demo

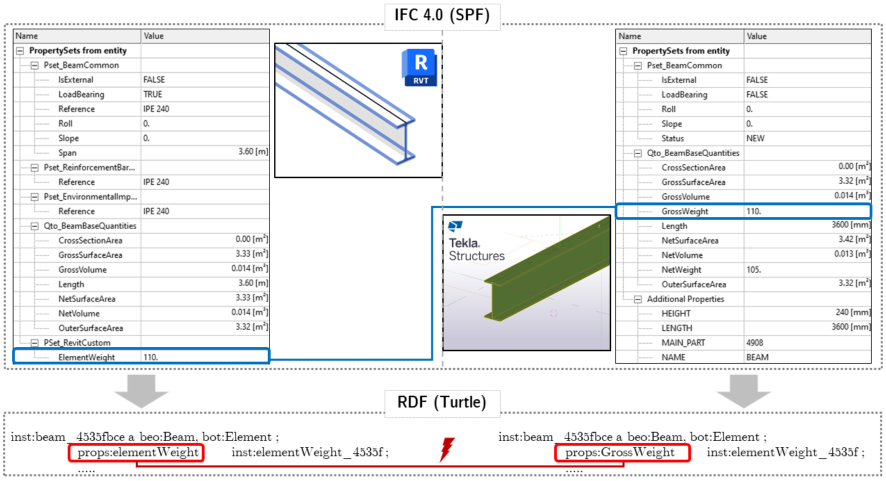


**Figure 2:** IFC export and resulting RDF output of an IPE 240 Steel Beam modelled in Autodesk Revit and Tekla Structures.

published a namespace bsdd:http://bsdd.buildingsmart.org/def# and an unnamed namespace https://identifier.buildingsmart.org/uri/buildingsmart/ifc/4.3/prop/. In the test, both ontologies were offline. In this work, we can only recommend buildingSMART to use an addresses that are accessible online. In our test, we used Postman software to test the connections. The property can be accessed with the HTTP header Accept: text/turtle to get the machine-readable version. It defines :GrossWeight. Our recommendation is to add a statement ":GrossWeigh rdf:type rdf:Property ." and use it as a unified RDF property for the values, or ":GrossWeigh rdf:type owl:Class ." and property instances it as a type. The first one has the benefit that the descriptions for the used properties are from the best dictionary, a single source of truth, emphasising the idea of sharing a standard dictionary in the industry while maximising the gained amount of information in the name of an RDF property, that is, not using RDF properties like :hasPropery, or :hasPropertySet. The latter approach has the benefit of already having an ontology with an online existence. In the following, we present the latter approach. IFCtoLBD was modified to accept a JSON description of URI mappings as input. Internally, each instance of the description of a property set is associated with the given map. Then, in the phase, when the converter writes out the property set RDF assessments, that mapping is used if a match is found. The idea is that there would be published mappings for common software and users could have private ones for their custom mappings. Since the IFC-LBD output already implies that simple property replacement is insufficient, we suggest using SPARQL property paths in the mapping description.

```
select ?Beam ?Weight where {
   ?Beam a beo:Beam.
   ?Beam ?props ?Property.
   ?Property a <https://identifier.buildingsmart.org/uri/buildingsmart/
      ifc/4.3/prop/GrossWeight>;
    opm:hasPropertyState ?PropertyLv3.
 ?PropertyLv3 schema:value ?Weight.
}
```

**Listing 3:** SPAQRL query with the mapping harmonized through bSDD

Figure 3 shows the proposed unified data model with links to the original data values. The corresponding query listing is shown in Listing 4.

```
select ?Beam ?Weight where {
   ?Beam a beo:Beam.
   ?Beam ?props ?Property.
   ?Property owl:equivalentTo ?oldProperty;
    opm:hasPropertyState ?PropertyLv3.
 ?PropertyLv3 schema:value ?Weight.

   ?oldProperty rdfs:label "PSet_RevitCustom:elementWeight".
}
```

**Listing 4:** SPARQL query mapping to bSSD with backwards compatability.

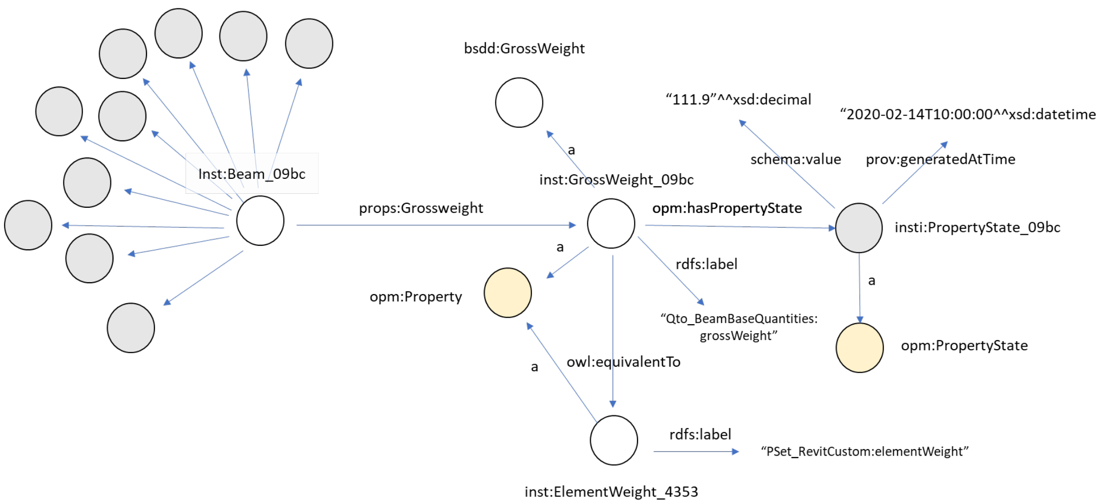


**Figure 3:** The data model with mapping to bSSD with backwards compatibility.

## 5. Discussion

Above, the handling was done using IFCtoLBD, showing that the mapping can be done so that the converter was instructed in a generic JSON description to replace specific property URIs with a given one. The implementation details of the unifying properties are described above and are found in the IFCtoLBD source code. In addition, the Java sample code for usage can be found in Example13.java[9]. To provide better insight, IFC.js-based IFC-LBD was also tested to show that RDF property creation is similar in independent implementations. It has a similar approach for property names but creates properties under https://example.com/resources/ namespace. It created the quantity sets but did not connect them to the building elements, so there were no comparable property URIs for the quantity sets. In this case, the implementation was not changed to include a generic RDF property mapping. Tekla_lbhack.ttl contains the analysed IFC-LBD output in the sample data repository (see Section 3.1). An insight is that a simple mapping rule cannot express a fix for the different ways of presenting the quantity sets in the converters, but the same logic can be implemented in both converters. The solution can be enhanced by providing mappings of company-specific or software-specific properties and commonly used ontology definitions online in JSON format, with frequently used mappings accessible through a public repository to encourage broader adoption. We have used the use scenario presented in the beginning of the article to evaluate the solution. In the scenario, we can use published mapping rules so that element descriptions of different providers can be queried using one query. By implementing the solution, we have shown that it is viable. However, analysing the possibility of standardising the RDF properties that correspond to IFC properties, e.g., using bSDD, revealed limitations in following Linked Data principles.

## 6. Conclusions

Inconsistent property namings and classifications during the export of building element properties from a design programme are common occurrences that significantly hamper the interoperability of created data models. The proposed solution offers a way to share RDF property mapping of user-defined and software-specific properties to commonly agreed definitions. We recommend using the BuildingSMART Data Dictionary (bSDD) definitions. The solution has limitations. For example, we do not unify the IFC models, which affects programmes that do not use RDF. Therefore, the recommendation is also to follow buildingSMART specifications when available. Also, unifying measurement units, conflicting values, and more complex mappings, like aggregations, are left for future studies.

## Acknowledgments

This work is part of the EConoM research project funded by the Federal Ministry for Digital and Transport of Germany within the initiative InnoNT (funding number 19Ol22009F). It was supported within the TARGET-X framework, a project funded by the Smart Networks and

[9] https://github.com/jyrkioraskari/IFCtoLBD/blob/master/IFCtoLBD/src/main/java/examples/Example13.java

Services Joint Undertaking (SNS JU) under Horizon Europe (funding number 101096614). The authors are responsible for the content.

## References

[1] P. Pauwels, Supporting decision-making in the building life-cycle using linked building data, Buildings 4 (2014) 549–579.

[2] J. I. Olszewska, M. Barreto, J. Bermejo-Alonso, J. Carbonera, A. Chibani, S. Fiorini, P. Goncalves, M. Habib, A. Khamis, A. Olivares, et al., Ontology for autonomous robotics, in: 2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN), IEEE, 2017, pp. 189–194.

[3] J. Oraskari, Live web ontology for buildingsmart data dictionary, in: 32. Forum Bauinformatik 2021, volume 32, 2021, pp. 166–173. URL: https://tuprints.ulb.tu-darmstadt.de/21521/. doi:10.26083/tuprints-00019496, forum Bauinformatik, fbi ; Conference date: 09-09-2021 Through 10-09-2021, Online; accessed Oct. 15. 2023.

[4] V. Alexiev, M. Radkov, N. Keberle, Semantic bsdd: Improving the graphql, json and rdf representations of buildingsmart data dictionary (2023).

[5] The International Organization for Standardization, Iso 16739-1: 2018: Industry foundation classes (ifc) for data sharing in the construction and facility management industries–part 1: data schema, 2018.

[6] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. O'Riain, Linking building data in the cloud: Integrating cross-domain building data using linked data, Advanced Engineering Informatics 27 (2013) 206–219.

[7] P. Pauwels, K. McGlinn, S. Törmä, J. Beetz, Linked data, Building information modeling: Technology foundations and industry practice (2018) 181–197.

[8] C. Bizer, T. Heath, T. Berners-Lee, Linked data - the story so far, International Journal on Semantic Web and Information Systems 5 (2009) 1–22.

[9] P. Pauwels, L. J. McGibbney, B. Thurm, J. Oraskari, G. Velludo, pipauwel/ifctordf: v0.4, 2020. URL: https://doi.org/10.5281/zenodo.4008032. doi:10.5281/zenodo.4008032.

[10] P. Pauwels, T. Krijnen, W. Terkaj, J. Beetz, Enhancing the ifcowl ontology with an alternative representation for geometric data, Automation in Construction 80 (2017) 77–94. doi:https://doi.org/10.1016/j.autcon.2017.03.001.

[11] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, R. Klein, The ifc to linked building data converter - current status, Proceedings of the 6th Linked Data in Architecture and Construction Workshop 2159 (2018) 34–43. International Workshop on Linked Data in Architecture and Construction, LDAC ; Conference date: 19-06-2018 Through 21-06-2018.

[12] M. Niknam, S. Karshenas, A shared ontology approach to semantic representation of bim data, Automation in Construction 80 (2017) 22–36.

[13] M. Holten Rasmussen, M. Lefrançois, M. Bonduel, C. Anker Hviid, J. Karlshøj, Opm: An ontology for describing properties that evolve over time, in: CEUR Workshop Proceedings, volume 2159, CEUR Workshop Proceedings, 2018, pp. 24–33.

[14] M. Bonduel, A. Wagner, P. Pauwels, M. Vergauwen, R. Klein, Including widespread geometry formats in semantic graphs using rdf literals, in: 2019 European Conference on

Computing in Construction, European Council on Computing in Construction, 2019, pp. 341–350.

[15] M. B. Rosson, J. Carroll, Scenario-based design, 2002, pp. 1032–1050. doi:10.1201/b11963-56.

[16] J. Oraskari, M. Bonduel, K. McGlinn, P. Pauwels, F. Priyatna, A. Wagner, V. Kukkonen, S. Steyskaland, J. Lehtonen, M. Lefrançois, Ifctolbd: Ifctolbd v 2.43.3, 2023. URL: https://github.com/jyrkioraskari/IFCtoLBD.