

A method for converting IFC geometric data into GeoSPARQL

Joseph O'Donovan¹, Declan O'Sullivan¹, and Kris McGlinn¹

ADAPT, Trinity College Dublin, Dublin 2, Ireland

odonovj2@tcd.ie

www.adaptcentre.ie

Abstract. Geographic Information System (GIS) can provide important contextual information relevant to buildings design, such as related to flooding, orientation, and even construction materials. Currently there is a disconnect between 2D GIS and 3D Building Information Modelling (BIM), as both tend to work from different geometric representations and coordinate systems hindering seamless integration of these representations. This paper presents a method for converting Industry Foundation Classes (IFC) geometries expressed in ifcOWL with additional GeoSPARQL, based upon the IFC defined geolocation, focusing on walls. The goal is to align these geometries with their GIS equivalents, so that an existing IFC model can be overlaid with GIS, or vice versa, allowing for the seamless movement between the geometries in either domain. This also makes possible the tagging of all BIM geometries with geolocation. To support the publication of this data, the paper also presents an export of the resulting GeoSPARQL aligned with the developing standard the Building Topology Ontology (BOT), thus enabling the linking of multiple geometric representations to a less verbose building reference model. A short evaluation of the performance of the algorithm for conversion is also presented as well as a test case to examine alignment of BIM with geospatial geometric representations.

Keywords: Linked Data · Industry Foundation Classes · GeoSPARQL.

1 Introduction

Effective management of smart buildings and cities requires building data that is structured, accessible and reliable [2]. Building Information Modelling (BIM) is the primary method of providing such building data, representing the physical and functional characteristics of a building in a 3D model. It is difficult, however, to describe BIM models in their overall geospatial context. This could result in important external environmental information being overlooked. Graphical Information Systems (GIS) [10] provide models of the environment and enable 2D spatial analyses of these areas. The integration of BIM and GIS can provide benefits in reducing redundant modelling and enabling data flows in both directions, supporting new applications related to the design and operation of smart environments [7].

The combination of BIM and Linked Data (LD) [5] has the potential to meet the requirements for storing, sharing and interlinking BIM with other Linked Data resources on the web described using the Resource Description Framework (RDF), such as geospatial data [12]. It remains the case that use of different geometric representations and coordinate systems in 2D GIS and 3D BIM leads to difficulties in moving between these representations. The disconnect between the two representations make it a challenge to relate a 2D GIS building with its corresponding 3D BIM model. There is as yet no standard process in place to convert a Cartesian coordinate IFC entity to GIS. New methods of carrying out this conversion process are required until an agreement is established between the differing domains on the definition of coordinate systems [13].

This paper presents a process for converting IFC geometries to GeoSPARQL and Well-known text (WKT) so that the two geometries can exist within a single coordinate system. Building walls are extracted from the IFC model and positioned relative to the building geolocation given by the IFC model. This creates a 2D footprint of the building and provides a bases for the alignment of 2D GIS geometries with their 3D BIM equivalent. The paper is structured as follows, section 2 provides state of the art, section 3 provides the design and implementation of the conversion process, along with a performance evaluation and examination of a method to align geometries from different domains using GeoSPARQL, section 4 provides discussion and future work and finally section 5 a conclusion.

2 State of the Art

2.1 Graphical Information Systems

GIS systems are information systems with an added geo-reference [10]. The geospatial information associated with GIS systems allow for spatial analysis to be performed on the system. GIS data has a vast array of use cases including the monitoring of weather systems across a region, predicting population densities in a city and visualising real time traffic jams in a certain location. Analysis of GIS data gives important location based insights which may have previously been overlooked. Geospatial information in a GIS system typically include the coordinates of features of an object, based on the real world location (geolocation) of the object. The relationship between features of the object can also be defined. Such features may include the walls of a building, and how certain walls relate to one another by being associated with a room within the building. GIS features are commonly represented in a 2D coordinate system. A building would be represented by its top-down building footprint. A GIS system representation of a building will also include the building within the context of other geospatial features, such as infrastructure and natural features. This makes the information relevant to both building designers and public planning organisations, planning for navigation, fire services, energy grids, etc.

Recent research in GIS primarily focuses on the development of 3D GIS [10]. Such systems facilitate the modelling of complex internal structures of buildings,

tunnels and bridges in the geospatial domain. City Geography Markup Language (CityGML) is currently the state of the art standard in 3D GIS modelling, providing an XML based data model for the storage and exchange of 3D models [9]. This 3D GIS modelling approach is closer to Building Information Modelling (BIM) than 2D GIS, however work still remains in providing seamless interaction between the two representations.

2.2 Building Information Modelling

The concept of Building Information Modelling (BIM) has been created to support the vast amount of data associated with buildings. This data is generated across the buildings life cycle (BLC) and requires maintenance and management throughout the BLC. BIM describes an integrated data model for storing all information relevant to the BLC, typically relating to the functional and physical characteristics of a building [6]. This primarily includes a 3D model of the architectural design, detailing the positions and dimensions of a buildings walls, rooms, windows, doors, roof, etc. BIM also facilitates the inclusion of non-physical building features such as the building costs, accessibility, safety, security and sustainability [17]. BIM is therefore capable of capturing all aspects of a building that exist throughout the BLC, aiding stakeholders at all stages of the BLC. As the authors state in [10], it is clear that BIM is not just a piece of software, but also a process that contributes to the workflow and project delivery process. However, a BIM model lives in isolation, it is oblivious to the external world that surrounds the building. This is a limitation of BIM. The lack of geospatial analysis functionality available from a BIM model restricts the knowledge one may have of a buildings surrounding environment. The impact a building may have on its environment could be overlooked if the appropriate geospatial analysis is not performed.

2.3 Interlinking BIM & GIS

There is potential to further improve existing use cases by integrating Building Information Modelling (BIM) with Geographic Information Systems (GIS). This integration would enrich BIM datasets, allowing stakeholders to observe new building features that may not have been visible using BIM alone, in particular relating to the buildings wider geospatial context. However, the different geometry representations and coordinate systems used by BIM and GIS make it challenging to interlink the two data types [12][13]. There is no standard process currently in place to transform BIM to GIS [10].

2.4 Industry Foundation Classes and Linked Data

Standardisation is an important part of ensuring data interoperability. Industry Foundation Classes (IFC), developed by buildingSMART, is the current leading standard for BIM in the Architecture Engineering and Construction (AEC)

industry. In IFC, buildings and their components are modelled as objects. Relationships and hierarchies between objects are defined, giving a semantically rich definition of a building. IFC uses the EXPRESS data modelling language to define a building and its component hierarchy. IFC is also the only BIM currently an ISO PAS standard (ISO 2013), and so it remains the de-facto standard for BIM in AEC. The AEC industry relies on IFC for modelling core building processes (architecture, structural analysis, control, etc.), enabling information to be passed between different stakeholders across the BLC. IFC has serializations in STEP, XML and RDF alongside an OWL version of IFC4, called ifcOWL [15], which is built upon Linked Data principles.

Linked Data (LD) is an approach to expose, share, and connect related data, which was not previously linked, on the Web [5]. LD makes use of the Resource Description Format (RDF) to store and link data, allowing linking between domains across the web. This interlinking of domains has significant potential in the AEC industry, where data related to different domains are generated and consumed across the BLC. Each stage of the BLC from design to construction and maintenance require data sourced from a vast set of domains; building geometry and topology data, sensor data, behaviour data, geospatial data, etc. The combination of BIM and LD has the potential to meet the requirements for storing and sharing those data using RDF.

With the development of ifcOWL the potential for linking ifcOWL, and other BIM ontologies, with other domains is an active area of research [19]. IFC, however, has yet to make its expected impact across the AEC industry [8]. One contributing factor is the complexity of the schema. This complexity arises from the extensive component hierarchy native to IFC. This component hierarchy can be challenging for new software developers who are unfamiliar with the IFC schema. A generic two storey building modelled in IFC can result in an IFC file thousands of lines long. Simple geometry extractions require prior knowledge of the IFC schema, which proves as a barrier to its use.

2.5 Building Topology Ontology

The Building Topology Ontology (BOT) [16], currently under development within the auspices of the Linked Building Data on the Web community group [18], may be able to address this issue, providing a bridge between software developers and IFC, and a means to publish data about their buildings easily. BOT provides descriptions for only the most fundamental properties of a building, in terms of its topology, in a Linked Data approach. The primary components of a building topology (walls, storeys, rooms, etc.) are modelled in BOT, avoiding the complexities observed in IFC. The intention here is to support linking to other ontologies when details are required for aspects of the building, such as those related to geolocation, building products, automation and control, HVAC etc.

The ifcOWL and BOT ontologies are used in a Linked Data approach to support this process of linking domains. The challenge still remains that differing domains have different representations of geometry, such that it becomes difficult to reuse a geometry from one domain in another. This paper presents a process

for the extraction of building geometries (building geolocation, building footprint, wall geometries, etc.) from ifcOWL into GeoSPARQL, and representing geometries as Well-Known-Text (WKT) as outlined here [14]. This conversion process makes all building elements essentially geo-tagged, and provides a mechanism to move from 3D geometries to 2D, and back.

3 Design and Implementation of Algorithm for Conversion of IFC Geometries into GeoSPARQL

The algorithm presented here is developed to convert geometries described in ifcOWL into GeoSPARQL. ifcOWL models are generated using the process developed by [15]. The 2D geometry extracted from the 3D IFC model is the top-down view of the model, where the external walls of the building represent the building footprint. The motivation behind extracting the 2D building footprint geometry is to define the BIM building in a 2D GIS context. This enables geospatial analysis to be performed on the BIM building, integrating the two domains. Furthermore, in a Linked Data context, the extracted geometry could be used to align the building to its corresponding building defined in a different Linked Data dataset. The alignment of two building geometries therefore becomes a method of interlinking Linked Data domains.

3.1 Algorithm Design

The algorithm to extract and represent a 2D building footprint geometry from an ifcOWL building can be broken down into three steps:

1. Building geolocation extraction
2. True North extraction
3. Building geometry extraction and representation as WKT

The geolocation of a building is useful in determining the exact location of a building. It allows positioning of a building in the world coordinate system through use of a latitude and longitude coordinate. The geolocation of an ifcOWL building is extracted to determine the real world position of the building. The extracted geolocation also acts as a reference point in the positioning of a building's walls. Extraction of the geolocation is the first step in connecting a BIM building to its corresponding GIS building. The second step is to extract the buildings true north orientation. This is the real world direction that the building is facing. The true north orientation angle is later used to rotate the building walls to the correct building orientation. The final step in creating the building footprint is to extract and represent the walls of the building. Each wall in an ifcOWL building is defined by a position, direction and length, all represented as Cartesian coordinates in metres. These three wall properties, along with the building geolocation and true north orientation, facilitate the placement of a wall in its correct real world position. The walls are positioned relative to the building geolocation and then rotated to the true north orientation angle.

Each external building wall is transformed to its real world position and represented as a Well-known text (WKT)¹ geometry string. WKT is a string based geometry representation language used to represent 2D vector geometries on a map. This is ideal for representing the building geometry in a 2D GIS context. Additionally, GeoSPARQL supports geometry representation as WKT. This enables the use of GeoSPARQL topology relationship functions in SPARQL queries to interact with WKT geometries. The WKT geometry representing the 2D building footprint of an ifcOWL building is inserted back into the ifcOWL file as an RDF triple. This processed ifcOWL file is now ready for GeoSPARQL querying.

3.2 Extracting True North

A building represented in the IFC format is typically modelled to represent a real world building, or at least to at some point be used as such. We are not concerned here with buildings that are intended purely to be used as design models, without a physical copy. This process is intended for buildings that have a real world orientation, or direction. An IFC representation of a building incorporates a buildings orientation with the true north direction of the building. This is the direction of the true north relative to the underlying coordinate system, given by a direction within the xy-plane of the coordinate system. If not given, the true north defaults to the positive direction of the y-axis. The `trueNorth>IfcGeometricRepresentationContext` defines the true north direction of an ifcOWL building as an xy-coordinate, where both x and y are greater than or equal to 0 and less than or equal to 1. The angle between this coordinate and the positive direction of the y-axis (0, 1) is calculated. The arc tangent of the positive distance between the true north x-coordinate and 0, and the negative distance between the true north y-coordinate and 1 gives the angle (in degrees) between the two coordinates. This angle is later used in the program to rotate wall coordinates to the true north orientation.

3.3 Extracting Geolocation

The geolocation of a building is useful in determining the exact location of a building. It allows positioning of a building in a world coordinate system through use of a latitude and longitude coordinate. The geolocation of an ifcOWL building is extracted to determine the world position of the building and assists in the positioning of a buildings entities, as presented here [11]. Entities have a local placement relative to the geolocation of the building (see Fig. 1). The **refLatitude>IfcSite** and **refLongitude>IfcSite** of an ifcOWL building represent the latitude and longitude of the geolocation. Recursive traversal of both attributes extract their respective values, given in the format of degrees, minutes, seconds and millionth of a second. Conversion to decimal degrees is necessary for alignment with Well-known text (WKT) geometries. A WKT point is created with

¹ <http://www.opengeospatial.org/standards/sfa>

the decimal degrees latitude and longitude in the form POINT(longitude latitude). A GeoSPARQL hasGeometry property is created for the **IfcSite**. This property relates to a GeoSPARQL asWKT property, containing the WKT point string that represents the building geolocation. These additions to the building model are subsequently present in the output file.

3.4 Building Geometry Extraction: IfcWallStandardCase

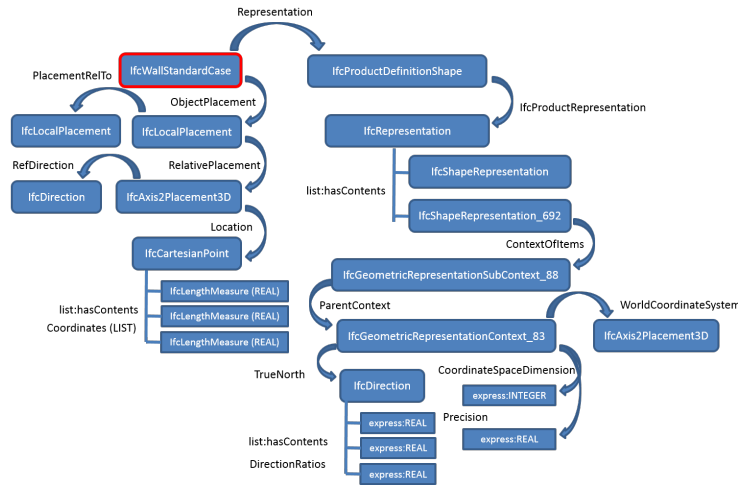


Fig. 1. Overview of IFC and its relationships for describing an entities geometry.

A process for extracting the geometry of an IFC building walls is necessary to create a building footprint. As such, the focus here is on the **IfcWallStandardCase** (Fig. 1), although it should be noted that this process should theoretically work for all building entities with a geometric representation. The building footprint facilitates alignment of the 3D IFC geometry with a 2D GIS geometry. IFC geometry models can contain two wall types: **IfcWallStandardCase** and **IfcWall**. The former is used for occurrences of walls that have a non-changing thickness. The latter is used for occurrences of walls that have a changing thickness, or have a non-rectangular cross section. A hierarchy of entity placements define where a wall is positioned in the buildings global coordinate system. The IFC Site of a building sits at the top of the hierarchy.

The IFC Building is then positioned relative to the IFC Site. Each IFC Building Storey is positioned relative to the IFC Building. All walls on a storey are placed relative to the IFC Building Storey, completing the hierarchy. The following process describes the extraction of walls represented as an **IfcWallStandardCase**. An **IfcWallStandardCase** is defined by three high level attributes: a position (local placement), an orientation/direction (defined by the direction

cosine of the Z and X axes) and a length (defined by a polyline magnitude). The **IfcLocalPlacement** of a wall specifies the position of a wall relative to the IFC Building Storey it is located on. The position is defined as a xyz-coordinate, denoting how far the wall is positioned from the origin. This origin is the origin of the building storey that contains the wall. All test data used in this paper had the origin of the IFC Building Storey set as the default origin (0, 0, 0), which, in turn, was the same origin as the IFC Building and IFC Site origins.

The **IfcLocalPlacement** of a wall also specifies the wall direction. The direction is defined by a xyz-coordinate. The default value for the direction coordinate is (0, 0, 0), indicating a negative rotation of 90 degrees around the z-axis. A value of -1 or 1 for the x-coordinate indicates a negative rotation around the z-axis of 270 or 90 degrees respectively. A value of -1 or 1 for the y-coordinate indicates a negative rotation around the z-axis of 0 or 180 degrees respectively. The length of an **IfcWallStandardCase** is specified by an **IfcProductDefinitionShape**. This shape defines the **IfcPolyline** points that represent the length of the wall. A polyline is constructed by two xy-coordinate points, the xy-origin (0, 0) and a length along the positive y-axis (0, Ylength), where Ylength is the length of the wall. The two polyline coordinates are extracted for each wall of the building.

Positioning of the wall in the global place requires three steps: wall direction rotation, wall global placement and a true north rotation. The two polyline coordinates of each wall are first rotated to the **IfcLocalPlacement** direction (see Fig. 2). These, now rotated coordinates, are translated to the local position of the wall given by the **IfcLocalPlacement** position. As stated above, the hierarchy of xyz local placements for each of the IFC Site, IFC Building and IFC Building Storey are positioned at the global origin. This infers that the positioning of a wall, based on its local placement, translates the wall to its correct global position. The final step in global wall placement is the rotation of a wall to its true north orientation. The true north angle of rotation is used to rotate wall coordinates to their correct global orientation. This three step procedure of global wall positioning is executed for each wall extracted from the ifcOWL file. Wall coordinates, in the global coordinate space, are next converted

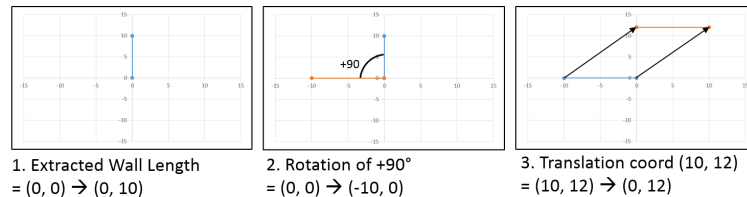


Fig. 2. Example rotation and then translation of a wall with initial polyline coordinates of (0, 0) and (0, 10). The orange line in the image on the left is the resulting rotated line. The orange line in the image on the right is the resulting translated line, translating the previously rotated line via the wall position coordinates. Shown in both images is the x and y axes. The z-axis is represented such that it is coming out of the page

to a latitude and longitude decimal degrees value. The conversion process uses the extracted decimal degrees geolocation as a point of reference. The decimal degrees geolocation is transformed to an xy-coordinate using the WGS 84 Web Mercator (EPSG:3857) projection, giving the geolocation as a coordinate in metres. The xy-coordinates of each wall, given in metres, are added to the metres geolocation coordinate. The resulting xy-coordinates for each wall are converted back to decimal degrees using the WGS 84 (EPSG:4326) projection, giving the walls global latitude and longitude in decimal degrees. A WKT LINESTRING() is created for each wall. The latitude and longitude of all walls are accumulated throughout the program and are concatenated at the end to create a WKT MULTILINESTRING().

The algorithm is implemented in Java using the Jena libraries [3] and building on open libraries for converting IFC to ifcOWL [15]. The Java implementation generate a GeoSPARQL WKT LINESTRING for each wall based upon the above process. These are also combined at a storey level into a 2D footprint of the building as a WKT MULTILINESTRING. The resulting GeoSPARQL geometries can be exported as an extended ifcOWL model or alternatively to a BOT model for the building.

3.5 Export of ifcOWL geometries as ifcOWL GeoSPARQL

The outputs of the converted file AC20-FZK-Haus, taken from an openly available repository of IFC files [1], can be found in the folder ifcOWL here² called AC20-FZK-Haus.geoloc.ttl. As can be seen, **ifcBuilding.434** has an associated GeoSPARQL geometry representing its footprint and so too each **IfcWallStandardCase.18465** has a single line geometry.

The same process then again produces the following BOT file, called bot/AC20-FZK-Haus.bot_geo.ttl. The bot:building has two bot:Storey, each of which has a geometry with a WKT MULTILINESTRING. Similarly, each wall is saved as a bot:element, associated with each bot:space, associated with each bot:storey. The bot:element has a geometry with a WKT LINESTRING.

This process was repeated for seven files, also taken from [1] (see Table. 1 for their names). Once the ifcOWL (or BOT) file is generated and exported, it is possible to test the output visually.

The WKT geometries were then evaluated visually by comparing them to their IFC models as seen in a BIM viewer such as BIM Vision³. This is clearly not the most desirable evaluation method, since no accuracy metric is produced, although visually comparing the two building representations gives an instantaneous and clear indication of how well a WKT geometry represents its BIM IFC model. Using this approach it was found that WKT geometries give an accurate representation of the wall structure of their parent IFC models. Walls are positioned in their correct locations and have dimensions that match the

² <http://theme-e.adaptcentre.ie/bim/>

³ <https://bimvision.eu/en/free-ifc-model-viewer/>

parent IFC model. Intricate models, such as the `Barcelona.Pavilion.ttl` and `Duplex_A_20110907_optimized.ttl` have a corresponding intricate 2D WKT geometry. However, the drawn WKT geometries contain both internal and external walls from all building storeys. A 2D GIS geometry represents a building as a 2D building footprint outline, disregarding internal walls and typically representing the buildings highest level or roof of the building.

The resulting extracted 2D WKT geometries in this solution should therefore represent the external walls of the IFC building model only, providing similar geometries to those in Google Maps. They would then match the 2D GIS standard of building geometry representation. A wall in ifcOWL can be defined as being internal or external using the `internalOrExternalBoundary_IfcRelSpaceBoundary` predicate. However, the ifcOWL test data used varied, meaning that some buildings used this predicate in the definition of a wall whereas others did not. Furthermore, buildings had incorrect wall definitions of this predicate. The ifcOWL building `smallhouse_saref.ttl` had all walls defined as being external, even though some of the walls are clearly internal walls. The usability of the results provided by this solution therefore depend on the quality of the input ifcOWL files. An agreement could be made with the owner of a BIM model that requires them to adequately define the nature of building components, whether they are internal or external, before using the WKT building representation solution provided by this project. This would allow for the discovery and representation of external walls.

3.6 Runtime Performance of the Algorithm

An evaluation of the performance of this solution assists in determining the usability of the solution. The performance can be evaluated by examining the program runtime. A short program runtime is more desirable, allowing users to extract a 2D GIS geometry from their 3D BIM model in a timely manner. The program runtime was recorded for each of the seven ifcOWL files given in Table. 1. A comparison of runtimes can be performed based on the time taken to calculate the Local Placement (LP) of an **IfcWallStandardCase**, in milliseconds, and the time taken to calculate the Relative Placement (RP) of an **IfcWallStandardCase**, in milliseconds. The LP and RP runtimes are listed in Table. 1. The triple count of each ifcOWL file is also given, indicating the complexity of the ifcOWL building representation. It can be seen that as the complexity of the file in terms of triple count increases, so too does the time taken to determine placement.

Table. 1 shows that as the triple count of an ifcOWL file increases the time taken to extract the local and relative placement of a wall also increases. This is partly due to the time it takes Apache Jena to search through an RDF model of an ifcOWL building. The bigger the RDF model, in terms of volume of RDF triples, the longer Jena takes to search and extract triples. The biggest file, `Barcelona.Pavilion.ttl`, takes over 13 seconds to determine the local placement of a wall and over 32 seconds to determine the relative placement of the same wall. This building contains 35 **IfcWallStandardCase** walls, therefore totalling

approximately 26 minutes to extract all 35 walls. This is not particularly desirable for the owner of a 3D BIM model looking represent their model as a 2D GIS geometry. This is something which must be addressed in futre work.

Table 1. Overview of seven ifcOWL files and time to process local and relative placement of IfcWallStandard

File Name	Triples	Triple Diff.	LP (ms)	RP (ms)
smallhouse.ttl	10175	0	44	100
smallhouse_saref.ttl	100246	+90071	776	1849
Duplex_A_20110907_optimized.ttl	205141	+104895	1945	4576
Simple3-storeytestbuilding.ttl	222545	+17404	2400	5642
20170804_Musterhaus_MIT.ttl	239899	+17354	2619	5850
AC20-FZK Haus.ttl	275685	+35786	2991	9900
Barcelona_Pavillion.ttl	1033905	+758220	13600	32400

3.7 Test Case: Alignment of IFC Geometries with Geospatial Geometries

In order to validate the alignment process, initial work has examined the use of the topological relations supported by GeoSPARQL. These allow for the discovery of relationships between geometries, and therefore resources, in a RDF graph such as: equality, disjointness, intersection, touches, within, contains and overlaps [4]. The use of the GeoSPARQL function `geof:sfOverlaps`, a function that matches geometries that partially overlap, has the potential to support building matching with the combination of additional properties, that is, where two buildings do not align correctly. Using the additional footprint area, a SPARQL Filter function, returns results based on the filter condition. Two filter conditions specify that the two returned building WKT geometries overlap (using `geof:sfOverlaps`) and that the calculated area of one WKT polygon is within +/- 25% the area of the overlapping polygon. Both aligned polygons, representing the same physical building, are returned if these two filter conditions are met.

Fig. 3 gives a screenshot of the query used to align an ifcOWL building with a test geometry, based on the two matching conditions. A test geometry was used as currently there is no publicly available IFC model with corresponding geospatial geometry (see future work). This query is performed on the AC20-FZK-Haus.ttl ifcOWL file, after adding WKT geometries, and the BOT version of this file, after BOT processing with added WKT geometries. Also given in Fig. 3 is the resulting aligned geometries, as seen in the YASGUI Geo results page. Colouring of the polygons identify the ifcOWL building from the BOT building. The red geometry represents the ifcOWL building and the green geometry represents the BOT building. Note, the geometries shown are quite small (YASGUI Geo does not permit a high zoom level), making it difficult to differentiate between the red ifcOWL geometry and the green BOT geometry. The



Fig. 3. SPARQL to align overlapping building geometries. Screenshot is taken from YASGUI, the Geo results page draws the ifcOWL (red) and BOT (green) geometries.

green BOT geometry is a rotated version of the ifcOWL geometry and overlays the red ifcOWL geometry. A sliver of red can be seen on the right hand side of the overlapping geometries, representing the underlying ifcOWL building geometry. Managing this alignment process is an important step in providing a means for seamlessly moving from 2D GIS into the BIM, as it is unlikely that the two geometry types will have an exact match.

4 Discussion and Future Work

This paper has demonstrated a method for converting ifcOWL wall geometries into GeoSPARQL and converting the resulting output as ifcOWL with GeoSPARQL and also BOT. It also provided a performance analysis of the algorithm. As can be seen, the process can successfully extract geometries for **IfcWallStandardCase**, convert these to WKT and output as GeoSPARQL. While the performance of the algorithm is strongly influenced by the complexity of the ifcOWL files, it is intended that this process only be run once, so that the extracted geometries can be exported into other less triple intense models, such as BOT. It may also be possible to improve the performance of this process when working directly with IFC step files, and future work will explore this option.

It should be noted that this paper does not specifically look at whether the resulting files match with existing geospatial data. Towards this end further validation will be conducted by examining existing Irish IFC models of known structures against geospatial data maintained by the Ordnance Survey Ireland (OSi), to ensure that they are in fact aligning correctly and to correct any

misalignment's. As OSi maintain authoritative sets of geospatial data in Ireland [12], they can provide a ground truth for geolocation of BIM models.

Future implementations will also support other geometric formats, such as obj, using this approach so that the ifcOWL can be converted into both 2D and 3D geometries and converted into BOT. The desire is to make the conversion method openly available, and therefore provide a method for owners of building data to publish subsets of their building geometry.

5 Conclusion

This paper has demonstrated a method for converting ifcOWL geometries into ifcOWL and BOT aligned with GeoSPARQL. The approach taken sets out to address issues relating to differing geometric coordinate systems between GIS and BIM geometric representations, by supporting the conversion of BIM geometry into 2D GIS. This opens the door to supporting a range of use cases related to geolocation of building entities, and makes available to BIM the functionality of GeoSPARQL. By aligning BIM models with authoritative building data sets such as those provided by the Ordnance Survey Ireland, BIM owners have the potential to connect their BIM data to a source of ground truth, firmly rooted in the physical geometric coordinates of the building structure. We believe this is an important step towards making seamless movement between these two domains a reality, encouraging re-use of available data.

Some challenges still remain to ensure that models are correctly aligned, and authoritative GIS data sets such as those provided by the Ordnance Survey Ireland are a perfect candidate for doing further analysis. Methods to ensure that alignments are correct are therefore still needed, and this paper presents some initial exploration of these for achieving this.

Listing 1.1. Prefixes and namespace URIs used in this paper

```
@prefix ifc:<http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD1/>.
@prefix bot:<https://w3c-lbd-cg.github.io/bot/>.
@prefix geo:<http://www.opengeospatial.org/standards/geosparql>.
@prefix sf:<http://www.opengis.net/ont/sf/>.
```

References

1. IFC repository, <http://smartlab1.elis.ugent.be:8889/IFC-repo/>
2. Big data in the construction industry: A review of present status, opportunities, and future trends. *Advanced Engineering Informatics* **30**(3), 500 – 521 (2016). <https://doi.org/https://doi.org/10.1016/j.aei.2016.07.001>
3. Api, J.O.: Apache Jena pp. 1–23 (1993), <https://jena.apache.org/>
4. Battle, R., Kolas, D.: Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web* **3**(4), 355–370 (Oct 2012), <http://dl.acm.org/citation.cfm?id=2590208.2590211>
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems* **5**(3), 1–22 (jul 2009). <https://doi.org/10.4018/jswis.2009081901>

6. Borrmann, A., Knig, M., Koch, C., Beetz, J.: *Building Information Modeling: Technology Foundations and Industry Practice*. Springer International Publishing (2018), ISBN: 978-3-319-92861-6
7. De Laat, Ruben and van Berlo, Leon: *Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension*, pp. 211–225. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-12670-3_13
8. Ghaffarianhoseini, A., Tookey, J., Ghaffarianhoseini, A., Naismith, N., Azhar, S., Efimova, O., Raahemifar, K.: Building Information Modelling (BIM) uptake: Clear benefits, understanding its implementation, risks and challenges. *Renewable and Sustainable Energy Reviews* **75**, 1046–1053 (2017). <https://doi.org/10.1016/j.rser.2016.11.083>
9. Gröger, G., Plümer, L.: CityGML Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* pp. 12–33. <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2012.04.004>
10. Liu, X., Wang, X., Wright, G., Cheng, J.C.P., Li, X., Liu, R.: A state-of-the-art review on the integration of building information modeling (bim) and geographic information system (gis). *ISPRS Int. J. Geo-Information* **6**, 53 (2017)
11. McGlinn, K., Debruyne, C., McNerney, L., O’Sullivan, D.: Integrating Building Information Models with Authoritative Irish Geospatial Information. In: *ISWC 2017, the 16th International Semantic Web Conference*. pp. 66–74 (2017), ISBN: 978-3-319-68204-4
12. McGlinn, K., Debruyne, C., McNerney, L., O’Sullivan, D.: Integrating Ireland’s Geospatial Information to Provide Authoritative Building Information Models. In: *Proceedings of the 13th International Conference on Semantic Systems - Semantics2017*. vol. 13, pp. 57–64. ACM Press (2017). <https://doi.org/10.1145/3132218.3132223>
13. McGlinn, K., Wagner, A., Pauwels, P., Bonsma, P., Kelly, P., O’Sullivan, D.: Interlinking geospatial and building geometry with existing and developing standards on the web. *Automation in Construction* pp. 235–250. <https://doi.org/https://doi.org/10.1016/j.autcon.2018.12.026>
14. Pauwels, P., Krijnen, T., Terkaj, W., Beetz, J.: Enhancing the ifcOWL ontology with an alternative representation for geometric data. *Automation in Construction* pp. 77–94. <https://doi.org/10.1016/j.autcon.2017.03.001>
15. Pauwels, P., Terkaj, W.: EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction* pp. 100–133. <https://doi.org/https://doi.org/10.1016/j.autcon.2015.12.003>
16. Rasmussen, M.H., Pauwels, P., Hviid, C.A., Karlshj, J.: Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. In: *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*. pp. 237–244. Heriot-Watt University (2017). <https://doi.org/10.24928/JC3-2017/0153>
17. Taylor, J., Bernstein, P.: Paradigm Trajectories of Building Information Modeling Practice in Project Networks. *Journal of Management in Engineering - J MANAGE ENG* **25** (2009). [https://doi.org/10.1061/\(ASCE\)0742-597X\(2009\)25:2\(69\)](https://doi.org/10.1061/(ASCE)0742-597X(2009)25:2(69))
18. W3C: Linked Building Data Group (2019), <https://www.w3.org/community/lbd/>
19. Zhang, C., Beetz, J.: Querying linked building data using SPARQL with functional extensions. In: Christodoulou, S.E., Scherer, R. (eds.) *EWork and eBusiness in Architecture: ECPPM 2016 : Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2016)*, Limassol, Cyprus, 7-9 September 2016. pp. 27–34. CRC Press (2016), ISBN: 978-1-315-38689-8