

Verificação da conformidade dos dados de projectos de construção com *Linked Data*

<https://doi.org/10.21814/uminho.ed.164.4>

**Marin Ljuban^{1,2}, Mathias Bonduel²,
José Carlos Lino¹**

¹ *ISISE – Departamento de Engenharia Civil da Universidade do Minho*

² *Neanex Technologies, Antwerp, Belgium*

marin.ljuban@gmail.com

mathias.bonduel@neanex.com

jclino@civil.uminho.pt

Resumo

O sector da construção tem assistido nos últimos anos a uma tendência crescente para a digitalização, resultando num aumento substancial de dados. No entanto, estes dados tendem a ser segregados em sistemas e formatos de ficheiros específicos, o que dificulta a criação de uma representação abrangente do ativo construído. Uma solução potencial para colmatar a lacuna entre estes diversos modelos de dados é a aplicação dos princípios de *Linked Data*. A utilização de *Linked Data* leva à criação de uma representação estruturada e explícita do conhecimento, permitindo a verificação da conformidade dos dados provenientes de várias fontes. Apesar do seu potencial, a escalabilidade da aplicação de *Linked Data* no domínio da construção tem sido impedida pela ausência de diretrizes de modelação claras. Estas diretrizes foram oficialmente introduzidas em 2022 com a norma EN 17632-1:2022: *Building Information Modelling (BIM) – Semantic Modelling and Linking (SML)*. Esta norma define um modelo de informação de alto nível e um conjunto de padrões genéricos de modelação da informação. Dado que o esquema de dados *Industry Foundation Classes (IFC)* é um esquema largamente utilizado ao longo do ciclo de vida de um ativo construído e normalizado a nível internacional (ISO 16739-1:2018), este estudo examinará e demonstrará o processo de alinhamento do IFC com as orientações descritas na norma EN 17632. Além disso, estabelecerá uma ligação entre o IFC e o modelo de informação de alto nível, conforme especificado pela norma EN 17632. Por último, o estudo analisa o potencial de verificação da conformidade quando se trata de conjuntos de dados IFC convertidos.

1. Introdução

O *Building Information Modelling* (BIM) é uma metodologia que digitaliza o sector da construção, ajudando-o a passar de um sector ineficiente e fragmentado para um mais eficiente. Isto deverá permitir poupanças na fase de projeto, construção e operação de um ativo construído. No entanto, para colher plenamente os benefícios do BIM, é importante considerar quais os casos de uso com interesse comercial para os investidores, que os incentivariam a prescrever a utilização do BIM nos seus projetos.

A tónica deve ser colocada em soluções tecnológicas que utilizem normas abertas e sejam fáceis de utilizar, bem como suficientemente flexíveis para permitir a aplicação em muitos casos de uso diferentes a nível internacional, nacional e municipal. Uma abordagem cada vez mais popular é a utilização de *Linked Data*. A utilização de *Linked Data* está a ganhar força, mas ainda sofre de uma falta de normalização na sua aplicação. Um desses esforços de normalização ganhou reconhecimento oficial através da introdução da norma EN 17632-1:2022: *Building Information Modelling (BIM) – Semantic Modelling and Linking (SML)*. Esta norma define um modelo de informação de alto nível, bem como um conjunto de padrões genéricos de modelação da informação que visam ajudar as organizações a aplicar os princípios do *Linked Data* no ambiente construído.

Devido à sua relativa novidade, tanto quanto é do conhecimento dos autores, até à data não foi feita qualquer tentativa de investigar as possibilidades de ligar esta norma recém-publicada à norma aberta mais amplamente utilizada na construção, a *Industry Foundation Classes* (IFC), reconhecida internacionalmente como ISO 16739-1:2018.

Por conseguinte, este documento começará por apresentar as partes do IFC e os princípios de *Linked Data*. Posteriormente, será apresentado um enquadramento teórico que permitirá a ligação entre o IFC e o modelo de informação de alto nível EN 17632. Na etapa seguinte, o enquadramento teórico será aplicado na prática através da criação de uma *Object Type Library* (OTL) e da estruturação de dados de projetos de teste, provenientes de um ficheiro IFC exemplo, em conformidade com esta OTL recém-criada. Por fim, o artigo será concluído com uma síntese dos resultados.

2. Estado da Arte

2.1. *Industry Foundation Classes*

A complexidade do *Building Information Modelling* torna improvável a utilização de uma única ferramenta ao longo do ciclo de vida do ativo. Pelo contrário, é mais provável que o modelo BIM seja desenvolvido e utilizado por várias ferramentas com objetivos diferentes. Para garantir fluxos de informação simplificados, a indústria sentiu a necessidade de um esquema de dados aberto e neutro em termos de fornecedores. O alinhamento das necessidades de todas as partes interessadas é um

processo muito difícil. O esforço mais notável para essa normalização foi feito com a introdução das *Industry Foundation Classes* (IFC), um modelo concetual de informação (esquema) atualmente utilizado para o intercâmbio de informações. O IFC tem estado em desenvolvimento contínuo desde 1994, sendo as versões mais notáveis o IFC 2x3 e o IFC 4, que ganharam reconhecimento internacional através da normalização ISO. Atualmente, o esquema IFC 4.3 aguarda a aprovação da normalização ISO, enquanto que o IFC 4.4.0 irá alargar o IFC 4.3 com funcionalidades adicionais, principalmente para túneis [1].

Tecnicamente, o esquema IFC é descrito na linguagem EXPRESS, o que significa que adota os seus princípios orientados para os objetos, como a abstração e a hereditariedade. Os conjuntos de dados IFC, aplicando o esquema IFC, podem ser serializados em vários formatos diferentes, sendo o mais utilizado o IFC-STEP [2]. No entanto, para este caso de uso, as serializações de *Linked Data* do IFC são as mais importantes, permitindo a utilização do IFC em aplicações de *Linked Data*. O EXPRESS para modelos de informação concetual e a serialização STEP para conjuntos de dados limitam a expressividade e a capacidade de ligar dados BIM a outros dados em formatos de dados heterogéneos. O IfcOWL é uma tentativa de ultrapassar estas limitações através da tradução direta do esquema IFC para formato *Linked Data*. As versões oficiais IFC 2x3 e IFC 4 da IfcOWL foram disponibilizadas utilizando a ferramenta EXPRESS-toOWL [3]. Esta investigação utiliza a versão oficial IFC 4 ADD2 TC1 IfcOWL [4] que contém 1798 classes.

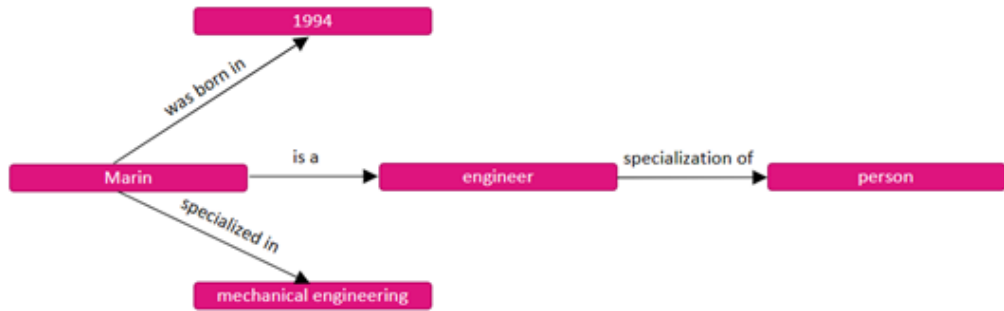
2.2. *Linked Data*

A *World Wide Web* pode ser considerada como uma camada adicional no topo da Internet que tem por objetivo organizar as formas como os dados são estruturados e partilhados para melhorar a interoperabilidade [5]. A principal organização que desenvolve estas normas Web é o *World Wide Web Consortium* (W3C), sendo que as suas principais normas preconizam:

Modelação de dados – Um grande desafio da interoperabilidade é garantir que os dados são trocados através de um modo comum de representação da informação [6]. Para o conseguir, no mundo do *Linked Data* existem várias formas de modelação de dados. A recomendada pelo W3C é o Resource Description Framework (RDF), que especifica que todas as coisas que podem ser descritas são recursos e que uma relação entre dois desses recursos é designada por *triple*. Trata-se de um simples gráfico de modelo de dados. Cada *triple* é constituído por três partes: um sujeito, um predicado e um objeto (Figura1).

Figura 1

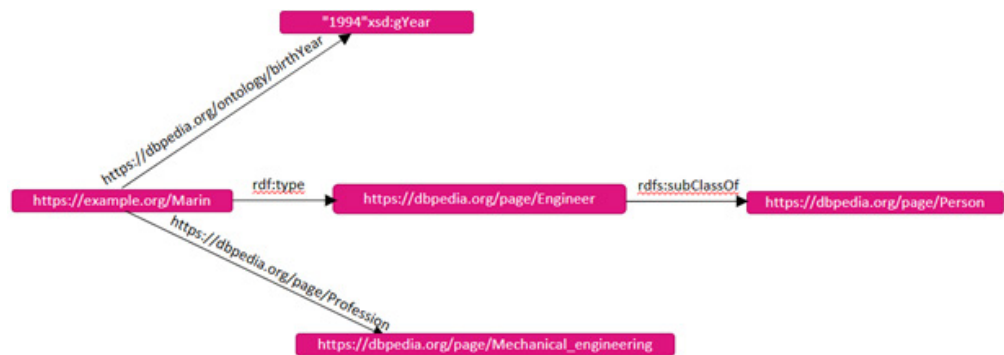
Representação do conhecimento através de tripletos.



Semântica – Depois de se chegar a uma forma comum de representação da informação, recomenda-se que se assegure que os dados sejam também compreensíveis por máquinas. Por conseguinte, é necessário estabelecer um quadro terminológico partilhado para garantir que os conceitos sejam compreendidos por todos os intervenientes de forma inequívoca. Isto é normalmente feito através da criação de modelos conceptuais de informação, frequentemente designados por "ontologias". Uma ontologia é definida como "uma especificação formal e explícita de uma conceptualização partilhada", o que significa que é um modelo legível por máquina, normalizado e explicitamente definido de algum fenómeno [7]. Com base em linguagens ontológicas genéricas como SKOS, RDFS, OWL e SHACL, qualquer ocorrência no mundo pode ser adequadamente representada através da criação de uma ou mais ontologias. O mesmo conhecimento da Figura 1 é representado na Figura 2, mas utilizando conceitos acordados para termos comuns, incluindo conceitos (engenheiro, pessoa), propriedades (ano de nascimento, tipo de recurso, especialização) e até tipos de dados (ano). Os URL habitualmente utilizados podem ser encurtados substituindo-os por prefixos (por exemplo, <https://dbpedia.org/ontology/> pode ser encurtado para dbp).

Figura 2

Representação do conhecimento utilizando ontologias comumente aplicadas.



Consulta - Depois de uma certa quantidade de dados ser convertida numa representação de *Linked Data* aplicando conceitos definidos em ontologias, esta seria inútil sem a possibilidade de aceder a essa informação quando necessário. Para garantir isso, foi criada uma linguagem de consulta chamada "SPARQL Protocol and RDF Query Language" (SPARQL). Uma consulta típica especifica quais as variáveis a obter e o que fazer com elas, utilizando palavras-chave como SELECT, CONSTRUCT ou INSERT, embora existam partes adicionais que podem refinar a consulta. Uma consulta

SPARQL simples que recupera o ano de nascimento do recurso das figuras 1 e 2 é apresentada na Figura 3.

```
prefix dbp: <https://dbpedia.org/ontology/>
prefix ex: <https://example.org/>

select ?birthYear where {
    ex:Marin dbp:birthYear ?birthYear .
}
```

Figura 3
Consulta SPARQL
para obter o ano de
nascimento.

Validação - As linguagens ontológicas comumente utilizadas, como OWL ou RDFS, partem do pressuposto de mundo aberto (Open World Assumption – OWA) e do pressuposto de ausência de nome único (No Unique Name Assumption – NUNA), o que dificulta a verificação dos conjuntos de dados quanto à existência de determinados elementos ou propriedades. Para facilitar a verificação da conformidade, é utilizada a SHACL (Shapes Constraint Language) do W3C. A SHACL assume o pressuposto de mundo fechado (Closed World Assumption – CWA) e o pressuposto de nome único (Unique Name Assumption – UNA), permitindo uma criação mais simples de dados que prescrevem qual deve ser a estrutura de um conjunto de dados ou quais as propriedades necessárias para que os conjuntos de dados cumpram determinados requisitos.

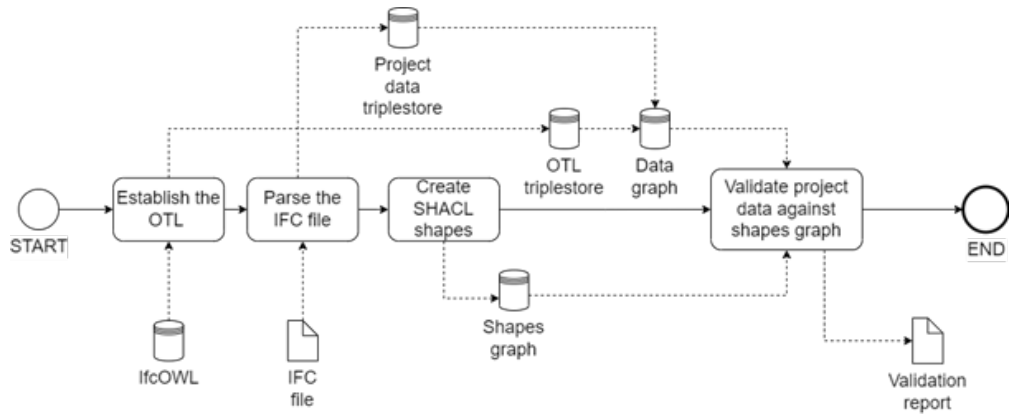
Para aplicar o *Linked Data* no contexto do ambiente construído, foi publicada a norma europeia EN 17632-1:2022 (*Semantic Modelling and Linking – SML*). Esta norma aborda "a interoperabilidade semântica e sintática para a informação que descreve os ativos que passam pelo seu ciclo de vida no ambiente construído". A especificação tem quatro partes normativas, quatro partes informativas e quatro exemplos (duas pontes, uma rede rodoviária e um hospital), para além da parte textual. A combinação destas duas normas (IFC e EN 17632) resultará numa *Object Type Library* (OTL) que aplica a SML como ontologia de nível superior. A biblioteca de tipos de objetos, tal como definida na plataforma *Linked Data Nederland*, "é uma biblioteca com nomes de tipos de objetos normalizados (por exemplo, estrada, viaduto) e propriedades ou especificações" [8]. É frequentemente considerada um tipo específico de ontologia, contendo a maior parte dos tipos e propriedades de objetos específicos de um domínio.

3. Enquadramento teórico e prova de conceito

A primeira parte do fluxo de trabalho é a criação de um IFC-OTL em conformidade com a norma SML. Esta OTL será criada ligando o modelo de informação de alto nível (ontologia) da norma EN 17632 com os conceitos adequados do IFC (serializados em IfcOWL). A OTL será então instanciada ao nível do projeto, analisando ficheiros IFC específicos (conjuntos de dados). Estes dados serão então testados quanto à sua conformidade, validando-os em relação às restrições armazenadas num gráfico de formas separado e expressas através da SHACL. A Figura 4 apresenta uma visão geral de alto nível do fluxo de trabalho.

Figura 4

Visão geral de alto nível do fluxo de trabalho proposto.



Uma vez que o esquema IFC é grande e complexo, esta investigação fará as seguintes simplificações:

- Apenas um subconjunto do esquema será considerado - O foco atual está nos elementos físicos (subclasses de `IfcBuildingElement`, `IfcDistributionElement`, `IfcFurnishingElement` e `IfcElementComponent`), aberturas (subclasses de `IfcOpeningElement`) e elementos espaciais (subclasses de `IfcSpatialElement`). Estes serão considerados e expressos em conformidade com a norma EN 17632, enquanto os restantes não serão tidos em conta.
- Modularização – A parte considerada do IFC será dividida, o que significa que o processo apresentado na Figura 4 será analisado para cada uma das partes que se seguem:
 - Taxonomia – Em primeiro lugar, a taxonomia dos tipos de objetos (por exemplo, parede, porta, etc.) será estabelecida através da consulta ifcOWL ligando as classes e enumerações encontradas aos conceitos definidos na NEN2660.
 - Relações – Para estabelecer as relações entre diferentes recursos ao nível do conjunto de dados em conformidade com a NEN2660, os ficheiros IFC serão analisados e certas relações IFC de interesse (decomposição e contenção espacial) serão substituídas pelos conceitos NEN2660 adequados.
 - Atributos – A ontologia ifcOWL será consultada para obter atributos das classes consultadas na etapa 1. Estes atributos serão anexados às classes recém-criadas, mas através de um padrão alinhado com a norma EN 17632.

3.1. Taxonomia

Para garantir a flexibilidade da OTL, os conceitos IFC serão convertidos em objetos OTL semelhantes (ou seja, `ifc:IfcWindow` será convertido em `otl:Window`), mantendo a ligação aos URIs ifcOWL através do predicado `rdfs:seeAlso`. Serão criados três tipos de classes OTL, mostrando a sua ligação aos conceitos da norma EN 17632 (NEN2660) na Figura 5.

1. Classes IFC genéricas, tais como `otl:BuildingElement`, `otl:SpatialElement`. Estas terão uma ligação direta com os conceitos NEN2660
2. Classes IFC específicas, como `Window` e `Space`, serão criadas na OTL e ligadas como subclasses a conceitos OTL mais genéricos criados no passo 1
3. As especializações das classes IFC, definidas no IFC como enumerações do atributo IFC `PredefinedType`, serão criadas como classes individuais na OTL e ligadas como subclasses às classes criadas na etapa 2 (estando assim indiretamente ligadas aos conceitos mais genéricos da etapa 1).

**Figura 5**

Ligação da EN 17632 (NEN2660) e um excerto dos conceitos OTL (IFC) a um nível taxonómico.

A aplicação da prova de conceito é efetuada carregando o IfcOWL versão 4 ADD2 TC1 na ferramenta Ontotext GraphDB e consultando-a com SPARQL. Esta consulta devolve valores que podem ser visualizados diretamente na ferramenta GraphDB, mas também devolvidos como um objeto JSON e utilizados diretamente no ambiente de programação em Javascript. Foi utilizada a última abordagem: os objetos JSON foram devolvidos e analisados. Dependendo se a consulta retorna uma classe IFC com uma enumeração `PredefinedType` ou sem ela, conceitos adequados seriam criados na OTL, como mostra a Figura 6.

```

TERSE TRIPLE LANGUAGE (TTL)

otl:BuildingElement a owl:Class ;
  rdfs:subClassOf nen2660:DiscreteObject ;
  rdfs:seeAlso ifc:IfcBuildingElement ;
  skos:prefLabel "Building Element"@en .

otl:Covering a owl:Class ;
  rdfs:subClassOf otl:BuildingElement ;
  rdfs:seeAlso ifc:IfcCovering ;
  skos:prefLabel "Covering"@en .

otl:Covering-INSULATION a owl:Class ;
  rdfs:subClassOf otl:Covering ;
  rdfs:seeAlso ifc:INSULATION ;
  skos:prefLabel "Covering INSULATION"@en .

otl:SpatialElement a owl:Class ;
  rdfs:subClassOf nen2660:SpatialRegion ;
  rdfs:seeAlso ifc:IfcSpatialElement ;
  skos:prefLabel "Spatial Element"@en .

otl:Space a owl:Class ;
  rdfs:subClassOf otl:SpatialElement ;
  rdfs:seeAlso ifc:IfcSpace ;
  skos:prefLabel "Space"@en .

otl:Space-INTERNAL a owl:Class ;
  rdfs:subClassOf otl:Space ;
  rdfs:seeAlso ifc:INTERNAL ;
  skos:prefLabel "Space INTERNAL"@en .

```

Figura 6

Trecho de código mostrando uma parte da OTL.

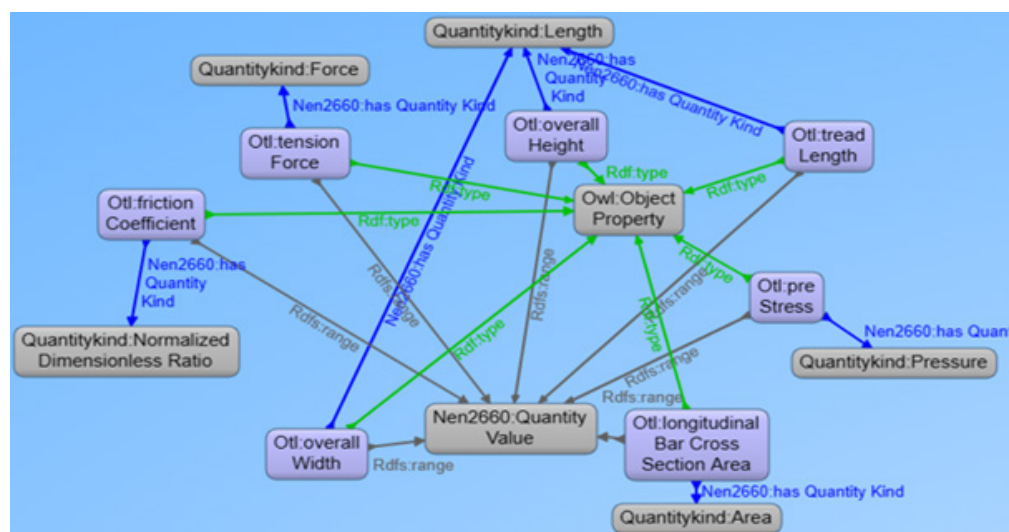
3.2. Atributos

Existem muitas formas de modelar atributos. A modelação mais simples de um atributo é através de owl:DatatypeProperty, também chamado padrão de modelação de nível 1 [9]. Este padrão de modelação é adequado para a modelação de atributos qualitativos (ou seja, cor, id, nome). Para atributos mais complexos, como os atributos quantitativos que consistem num valor e em metadados adicionais, como uma unidade de medida, é necessário um padrão de modelação mais complexo com nós intermédios que separam o valor e a unidade de um atributo complexo, ligando-os ao nó do atributo através dos predicados rdf:hasValue e nen2660:hasUnit, respetivamente. Além disso, a norma EN 17632 prescreve a utilização da ontologia QUDT para os tipos e unidades de quantidade. Por conseguinte, os conceitos do IFC foram substituídos por conceitos adequados da ontologia de tipos de quantidades QUDT, como se mostra na tabela 1. A aplicação do padrão de modelação de nível 2 resulta na criação de objetos de atributos OTL, como se mostra na figura 7.

Tabela 1 – Mapeamento dos tipos de medidas IFC com conceitos quantitativos adequados

IFC measure type	Replacing concept
IfcPositiveLengthMeasure	quantitykind:Length
IfcAreaMeasure	quantitykind: Area
IfcForceMeasure	quantitykind: Force
IfcNormalizedRationMeasure	quantitykind: NormalizedDimensionlessRatio
IfcPressureMeasure	quantitykind: Pressure
IfcIdentifier	xsd:string (level 1 property modelling)
IfcLabel	xsd:string (level 1 property modelling)
IfcGloballyUniqueID	xsd:string (level 1 property modelling)

Figura 7
Visualização dos atributos modelados de acordo com o padrão de modelação de propriedades de nível 2.



3.3. Análise dos ficheiros IFC-STEP

Depois de estabelecida a taxonomia OTL com atributos, o passo seguinte foi desenvolver um script que pudesse pegar num ficheiro de dataset IFC-STEP, processá-lo e produzir um grafo RDF (nos formatos JSON-LD e TTL) que estivesse de acordo com a OTL. A implementação é feita em JavaScript com a utilização do pacote web-ifc desenvolvido como parte do projeto IFC.js. Os modelos IFC de teste são 2 modelos de arquitetura e 1 modelo de sistemas.

A API web-ifc permite obter todos os elementos pela sua `IfcClass` ou pela `IfcRelationship`. Por conseguinte, a primeira etapa da análise IFC-STEP consiste em recuperar todas as entidades do conjunto de dados, as suas classes e enumerações (se disponíveis) e utilizar esta informação para criar nós no grafo RDF, em que o GUID IFC (`GlobalId`) serve de identificador único do elemento em causa, ligando-o através da relação `rdf:type` às classes existentes na OTL. A análise dos ficheiros resulta num conjunto de dados em que as instâncias estão ligadas aos tipos OTL, como mostra a Figura 8.

```
dis:2U2bEFdi9D29Y9qmkfORiT a otl:PipeFitting-BEND .
dis:29JEkkq6v4deitAoBmBEWS a otl:AirTerminal-GRILLE .
dis:29JEkkq6v4deitAoBmBEFu a otl:WasteTerminal-ROOFDRAIN .
dis:29JEkkq6v4deitAoBmBDmU a otl:DuctSegment-RIGIDSEGMENT .
```

Figura 8

Excerto de conjunto de dados IFC convertido – ligação de instâncias a tipos OTL.

A análise dos atributos resulta em atributos de nível 1 e nível 2 modelados e anexados aos elementos relevantes, como mostra a Figura 9.

TERSE TRIPLE LANGUAGE (TTL)

```
dis:1qHFr7JC9CT05IS3NzkT0p otl:globalId "1qHFr7JC9CT05IS3NzkT0p" ;
otl:name "Doors_ExtDbI_Flush:BIMA+_Door_ExteriorGlassDoor_1510x2310mm:317917" ;
otl:objectType "Doors_ExtDbI_Flush:BIMA+_Door_ExteriorGlassDoor_1510x2310mm" ;
otl:tag "317917" ;
otl:overallHeight dis:1qHFr7JC9CT05IS3NzkT0p-overallHeight ;
otl:overallWidth dis:1qHFr7JC9CT05IS3NzkT0p-overallWidth ;
otl:operationType otl:DOUBLE_DOOR_SINGLE_SWING-Door ;
a otl:Door-DOOR .

dis:1qHFr7JC9CT05IS3NzkT0p-overallHeight rdf:value 2310 ;
nen2660:hasUnit <http://qudt.org/schema/qudt/Milim> .

dis:1qHFr7JC9CT05IS3NzkT0p-overallWidth rdf:value 1510 ;
nen2660:hasUnit <http://qudt.org/schema/qudt/Milim> .
```

Figura 9

Trecho de TTL do conjunto de dados mostrando atributos de nível 1 e nível 2 para um elemento IFC.

3.4. Controlo da conformidade com o SHACL

Por último, para garantir a validade do quadro proposto, bem como dos dados produzidos pela aplicação de prova de conceito, é necessário verificar o conjunto de dados com as restrições SHACL. A validação SHACL será efetuada a dois níveis:

O primeiro nível consiste em verificar a conformidade do conjunto de dados com as formas SHACL existentes na OTL. Estas formas SHACL expressam a natureza inerente do esquema IFC (como por exemplo, que cada indivíduo, classificado numa subclasse da classe `IfcRoot`, deve ter um GUID) e são, portanto, universalmente aplicadas independentemente do projeto.

O segundo nível é a verificação das restrições específicas do projeto contidas num gráfico de formas separado. Estas formas diferem de projeto para projeto ou mesmo dependendo da disciplina que está a ser verificada

O SHACL pode validar dados para restrições de valor, como mostrado na Figura 10, bem como para restrições relacionais.

Figura 10
Restrições de valor a nível do projeto.

```

TERSE TRIPLE LANGUAGE (TTL)

shapes:doorAccessibilityShape rdf:type sh:NodeShape ;
  sh:property _:r0 ;
  sh:targetClass otl:Door .

shapes:valueShape rdf:type sh:PropertyShape ;
  sh:minInclusive "2400"^^xsd:integer ;
  sh:path rdf:value .

_:r0 sh:message "The door height is not at least 2400" ;
  sh:path otl:overallHeight ;
  sh:property shapes:valueShape .

```

4. Conclusões

O principal objetivo deste trabalho foi investigar as possibilidades de ligar modelos de dados normalizados no ambiente construído. Foi efetuada uma análise do estado da arte do IFC e da *Linked Data*. No que respeita à tecnologia, a revisão da literatura confirma que as soluções implementadas devem basear-se em normas abertas, fáceis de utilizar e flexíveis. Tendo isto em mente, o conteúdo das normas (ISO 16739-1:2018 ou IFC, e EN 17632-1:2022 ou SML) foi analisado na perspetiva da sua interconectividade. A investigação introduziu meios teóricos de ligação destas normas para a taxonomia, as relações e os atributos, juntamente com uma proposta de modelação de restrições SHACL para validação de dados, incluindo o primeiro exercício de verificação da conformidade de casos de uso. É necessário mais trabalho

para explorar melhor o potencial do SHACL no domínio do licenciamento e da verificação automática da conformidade.

Depois de apresentar o enquadramento teórico, este foi aplicado na prática com recurso a ferramentas de software já existentes, à linguagem de programação JavaScript e a várias bibliotecas JavaScript, sendo a mais importante a IFC.js. As aplicações para criação de OTL, análise de modelos e validação foram desenvolvidas e testadas com sucesso.

Agradecimentos

Esta investigação foi realizada com o apoio da União Europeia através da Bolsa Erasmus Mundus, no âmbito do Mestrado Conjunto BIM A+ Erasmus Mundus.

Referências

- [1] “The status of IFC 4.3 and the benefit of further extensions as IFC 4.4 – buildingSMART International.” Accessed: Sep. 08, 2023. [Online]. Available: <https://www.buildingsmart.org/>
- [2] buildingSMART, “IFC Formats,” buildingSMART Technical. Accessed: Jun. 26, 2023. [Online]. Available: <https://technical.buildingsmart.org/standards/ifc/ifc-formats/>
- [3] P. Pauwels and W. Terkaj, “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology,” *Automation in Construction*, vol. 63, pp. 100-133, Mar. 2016, doi: 10.1016/j.autcon.2015.12.003.
- [4] “ifcOWL ontology (IFC4_ADD2_TC1).” Accessed: Nov. 21, 2023. [Online]. Available: https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL/index.html
- [5] Britannica, “World Wide Web | History, Uses & Benefits | Britannica.” Accessed: Jun. 26, 2023. [Online]. Available: <https://www.britannica.com/topic/World-Wide-Web>
- [6] Ž. Turk, “BIM A+4: Advanced BIM data systems and interoperability ISO Step,” 2023.
- [7] R. Studer, V. R. B. B'c, and D. Fensel, “Knowledge Engineering: Principles and methods,” 1998.
- [8] “Object Type Library – Platform Linked Data Nederland.” Accessed: Sep. 25, 2023. [Online]. Available: <https://www.pldn.nl/wiki/OTL>
- [9] M. H. Rasmussen, M. Lefrancois, M. Bonduel, C. A. Hviid, and J. Karlshø, “OPM: An ontology for describing properties that evolve over time”.