Semantic representation of Design for Manufacturing and Assembly offsite housing

Edlira Vakaj [a,*], Franco Cheung [a], Abdel-Rahman H. Tawil [a] and Panagiotis Patlakas [a]

[a] *Faculty of Computing, Engineering and Build Environment, Birmingham City University United Kingdom,*
E-mail: Edlira.Vakaj@bcu.ac.uk
E-mails: Franco.Cheung@bcu.ac.uk, Abdel-Rahman.Tawil@bcu.ac.uk, Panagiotis.Patlakas@bcu.ac.uk

Abstract. Architecture, Engineering and Construction (AEC) is a fragmented industry dealing with heterogeneous data formats coming from different domains and, despite efforts such as Building Information Modelling (BIM), the gap in information exchange is often major. This challenge is particularly evident in the rapidly emerging field of Design for Manufacturing and Assembly (DfMA), which deviates from typical construction methodologies. Semantic web technologies are recognized for overcoming challenges of information exchange in isolated domains in many fields, via the publication of standardized linked data that are highly discoverable and machine processable. While ontologies have been proposed for manufacturing processes in general, this work is the first to apply semantic web technologies in the DfMA domain, supporting its integration to typical AEC workflows. A new domain ontology, Offsite Housing Ontology (OHO), is presented. OHO facilitates the semantic integration of offsite construction knowledge, enabling it to be used in DfMA practice. It semantically defines offsite construction domain terminology and relationships, describing a core vocabulary. This supports a unified model, required for efficient collaborative design management, while improving existing data flows. The efficiency and effectiveness of the OHO approach is demonstrated in a real-world DfMA scenario through the development of a Knowledge Based Engineering tool to automate cost estimation. As OHO is extensible, this approach can be adapted and extended to accommodate a very wide range of offsite housing, delivering important optimization and automation benefit from DfMA solutions.

Keywords: Offsite Construction, Ontology Engineering, Building Information Modeling, DfMA, Linked Building Data, Software Development

1. Introduction

Offsite construction aims to standardise and automate production processes in a factory environment through applying manufacturing design concepts [1]. To construct offsite effectively and efficiently, it requires products to be designed systematically through adopting Design for Manufacturing and Assembly (DfMA), a philosophy and design methodology [2]. DfMA is comprised of Design for Manufacturing (DfM) and Design for Assembly (DfA). DfM targets the selection of materials that minimise wastage, optimise processes and sub-processes, optimise parts and systems and fulfill tolerance requirements [3], whereas DfA focuses on minimising the number of modules for assembly and optimizing the assembly process [4].

Although it is common practice to use Building Information Modelling (BIM) for design and production of information, DfMA has not been widely applied in building design due to the lack of information and structured knowledge [5, 6], big data sources [7], and documented sources of information on process modularisation [8, 9]. There are efforts to develop DfMA frameworks, e.g. the synthesis of construction-oriented DfMA guidelines

*Corresponding author. E-mail: Edlira.Vakaj@bcu.ac.uk.

[10], but there is not yet an approach to systematise DfMA knowledge that enables designers to evaluate modular design. One main obstacle to apply DfMA in designing modules is the lack of tools that evaluate modular design, typically requiring input from several professionals with expertise in off-site production, costing and scheduling. Current practice relies largely on either heuristics or broad estimates, without an analysis of the relations and interactions of processes and sub-processes to the activity level [11]. For DfMA to be applied widely, an appropriate data modelling approach and corresponding data models need to be developed. In this regard, the study first reviewed the current use of BIM and identified that there is a lack of manufacturing information in BIM models. While efficient data exchange relies on the application of semantic web technologies and Linked Data, current ontologies to represent building design, product and production reviewed were found to be insufficient for implementing DfMA. Therefore, a new OWL domain ontology (i.e. Offsite Housing Ontology (OHO)) is proposed here to represent the domain knowledge of DfMA. The ontology developed was used to build an online knowledge-based tool for estimating cost for modular house construction to demonstrate the use of OHO. As OHO is extensible, the semantic web technologies and Linked Data (LD) approach used can be used to accommodate a wide range of offsite products, and extended to other functions such as the optimisation of modular production.

2. Limitations and potentials of data models for DfMA

The introduction of Building Information Modelling (BIM) provides an opportunity to structurally embed data in relation to processes and other attributes within a three-dimensional (3D) building model [12, 13]. For instance, it is common to have process-related data, typically used for the scheduling (i.e. 4D BIM), and cost data, typically for estimating of building costs (i.e. 5D BIM), integrated to the BIM model. While the data for scheduling are process-related, they refer mainly to activities onsite. Cost estimating during design development, on the other hand, does not use process-related data at all [14].

Although there are tools attempting to integrate the process and cost data to enable simultaneous 4D and 5D modelling, their application in practice is limited due to data and disciplinary silos [15]. The use of DfMA offers an opportunity to break the silos as the manufacturing and assembly processes are defined in the design development stage. Unlike how geometric information is kept and represented in BIM models, there is a lack of defined manufacturing concepts defined in the models, e.g. manufacturing processes in relation to particular modular design, which increases dramatically the data exchange requirements.

BIM applies an ontological representation for data exchange. The standard platform-neutral schema is the Industry Foundation Classes (IFC) [16]. While IFC supports the use of semantic web technologies, its data structure does not represent production processes enough. For instance, the product classification for DfMA design is based on the product sub-assemblies, which are generally under-represented in the IFC schema. Also, the schema does not represent the processes of manufacturing production. Thus, the use of the IFC schema alone will miss the chance to optimise modular design through DfMA implementation that requires simultaneous evaluation of the production and assembly attributes as well as cost, to support design decisions [2]. For instance, setting a competitive price is found as one main challenge for prefabricated house builders [17].

3. Related works: Ontologies and Data Models

Semantic web technologies allow the development of a formal representation of information, irrespective of the adopted tool. They have been used in the AEC domain for heterogeneous data formats from different sources and domains [18], supporting flexible data exchange and distributed data management, and providing a basis for logical inference using rules and ontologies [19]. Four different types of related ontologies have been reviewed: building design and construction ontologies, product ontologies, production ontologies and cost-estimation ontologies (Table 1). The limitations of these ontologies for the purpose of DfMA implementation have also been identified.

3.1. Building Design and Construction Ontologies

The conceptual schema for IFC is defined using EXPRESS, a data specification language [29]. Using IFC, data from a BIM model can be exchanged between heterogeneous software applications. However, IFC itself is not a web-compliant standard.

Table 1
List of reviewed ontologies.

prefix	Name	Ontology URI	Reference	Category
bpo	Building Product Ontology	https://w3id.org/bpo	[20]	Product
scp	SolConPro Ontology		[21]	Product
gr	Good Relations	http://purl.org/goodrelations/v1	[22]	Product
schema	Schema.org	https://schema.org/	[23]	Product
bot	Building Topology Ontology	https://w3id.org/bot	[24]	Standardized
ifc	ifcOWL	http://www.buildingsmart-tech.org/ifcOWL/	[25]	Standardized
pto	PRODUCT	http://www.productontology.org/id/		Product
props	PROPS	https://w3id.org/props		Product
	MASON		[26]	Manufacturing
Dicon	Digital Construction Ontologies	https://w3id.org/digitalconstruction/0.5/	[27]	Construction
DOCK1.0.	DOCK		[28]	Construction

The use of semantic web technologies [30], including OWL and the Resource Description Framework (RDF) resolves this limitation and can make IFC data widely available and accessible over the web. The IFC schema has been converted into an ontology in the Web Ontology Language (OWL) (i.e. ifcOWL) to make it usable with semantic web technologies. This makes is possible, for instance, to make inferences using Description Logic (DL) rules [25].

IFC is an extensive ontology, the latest version, IFC4, consists of 1293 classes and 1572 object properties. The complexity of this ontology makes reasoning and management very hard and inefficient, and inevitably, increases the need to develop separate modules based on the existing core IFC modules. An implementation of a modular IFC ontology was proposed [31] and has initiated several research initiatives that focus on how to modularize the IFC ontology to improve its extensibility. This has for example initiated to a large extent the motivation to initiate the W3C Linked Building Data (LBD) Community Group.

BIM adoption has been an industrial interest and various countries and communities have their trajectories to map BIM's development [e.g. Digital Built Britain in the UK [32]]. Generally, it is anticipated that BIM data will be communicated efficiently inter-disciplinarily using web-based technologies. Also, the domain ontologies for BIM are supposed to be extendable according to the concept of Linked Building Data (LBD) [33]. In this regard, the Building Topology Ontology (BOT) forms a small part of a broader concept of LBD, in which additional domain ontologies can be proposed to further extend the core of BOT, e.g. with building element classifications and properties. The current version of BOT (i.e. version 0.3.1) consists of 7 classes, 14 object properties, and one datatype property. The ontology is adopted to answer the competency questions (CQs) relating to the concepts of Zone, Site, Building, Storey, Space and Elements [34]. BOT has three main classes; they are `bot:Zone`, `bot:Element`, and `bot:Interface`. `bot:Zone` and `bot:Element` define the concepts and have sub-classes whereas `bot:Interface` provides information about the relationships between zones and elements. As an interface is assigned to individual zones and elements using the object property of `bot:interfaceOf`, individual zones and elements are connected. This allows users to define properties of this interface such as the area and thermal transmission value, etc. BOT is aligned with other ontologies, such as BRICK (A uniform metadata schema for buildings [35]) and SAREF (Smart Applications REFerence) [36]. It can easily be combined with ontologies that describe products, IoT devices or sensor observations.

A construction knowledge management ontology, Domain Ontology for Construction Knowledge was introduced in 2013, DOCK 1.0 [28]. DOCK 1.0 is a traditional construction management specific ontology that provides defined key concepts in construction (such as actor, resource, product and state). Similarly, the BIM4EE project has very recently proposed digital construction ontologies, DiCon, as shared representations of construction domain knowledge that describes construction activities [27]. The aim of DiCon is to integrate heterogeneous construction workflow information to support site management tasks such as assessing site environmental conditions, quality inspection and scheduling of works. Although DiCon has the class of activity and attempts to develop process-based knowledge, the knowledge representation and relationships defined are for the onsite context, which is not suitable for modelling the knowledge for offsite activities.

3.2. Product Ontologies

Common product ontologies such as GoodRelations [22] and Schema.org are supported by search engines such as Google. However, they do not include complex concepts such as how products are assembled. Building domain specific ontologies have the capacity to include more building related information. For instance, the SolConPro ontology describes multi-functional building products with high level of details. However, it does not support modularity (i.e. linkages to other ontologies is not effective), is not aligned to other ontologies, and contains ambiguous vocabularies that lack clear definitions [20].

There were initiatives by the W3C LBD group to translate IFC to product related machine-readable ontologies in RDF format, PRODUCT and PROPS. The PRODUCT ontology consisted of the IFC classes underneath the IfcElement node, while the PROPS ontology included all properties defined in the IFC property sets. Both are simple ontologies and do not align with other common manufacturing vocabularies [20].

The Building Product Ontology (BPO) is a multi-layered product ontology, which was designed to overcome these shortcomings. It allows products to be defined according to the assembled components, entity interconnections, component and product properties. To be able to evaluate modular construction, BPO needs to incorporate further ontologies in order to facilitate various types of evaluations, e.g. on cost and time.

3.3. Production Ontologies

The Manufacturing's Semantics Ontology (MASON) is a standardized manufacturing ontology that establishes a common semantic vocabulary in the manufacturing domain. It is organized in three main concepts: Entities, Operations and Resources. Entities give an abstract view of the product in terms of geometric features, raw material and cost entities. Operations describe all processes linked to manufacturing including machining operations, control or assembly and logistic operations, human operations and launching operations as well. Resources classify all resources linked to manufacturing such as geographic resources, machine tools, human resources etc. [26]. While the ontology is a standard for the manufacturing domain, its application to modular construction products is limited.

Extending from a MASON-based approach, An et al. [37] developed extended ontologies for wood frame assemblies. The authors identified the limitation of using the BIM model alone to explain how wood frame is assembled for production, i.e. it does not include information about how the product is fabricated. Three classes are created, they are product, operation and resource. The ontology developed helps to identify the required manufacturing operations for wood panel frame construction although the ontology is not detailed enough to perform cost and time estimations and to explain the sequence in which a product is assembled.

3.4. Cost Estimation Ontologies

BIM technologies are used in cost estimation making the cost estimation process more efficient, mainly because standard machine-readable formats can be used in combination with standard measurement methods to automate the construction cost estimation. BIM-based cost estimation tools, such as CostX and BIMMeasure, are designed to streamline the quantity measurement processes, with some degree of automation built-in in those tools. However, data models are yet to have process-related data incorporated and thus, those tools have not been developed to be able to populate or extract process related data, an aspect that is crucial for implementing DfMA, i.e. to facilitate simultaneous design evaluation.

Differing from traditional approach to estimate cost that demands individual professionals to either measure or extract quantities, link quantities with unit rates and present cost report, the use of ontology can automate the estimation processes. The use of semantic web technologies has been proposed for almost 20 years. However, the existing ontologies for cost estimation are either building component-based (e.g. elemental costing) [38–40] or resource-based [41, 42]. Either of them has little account for processes. For instance, Abanda et al. proposed a cost estimating ontology based on the New Rules of Measurement 1 (NRM 1) [40]. Its data structure breaks down a building into elements, sub-elements, group elements and components and has NRM 1-

based rules of reasoning, to enhance reasoning. Liu et al. [41] adopted a semantic approach using Autodesk Quantity Take-Off (QTO) and the Revit data model for the preparation of a light-frame building estimate. A construction-oriented product ontology was defined and used to provide an ontology-augmented BIM model in RDF. Neither of the ontologies proposed is suitable for implementing DfMA. Other researchers have looked at more specific problems, such as automating the costing of tiling works [42], labour costs [41] and labour content [43]. These studies are small in scope, and do not allow for generalisation, especially for modular production concepts in DfMA.

3.5. Research Gap

A comparison of the ontologies reviewed is given in Table 2. It shows that while existing ontologies partially capture specific aspects of DfMA, none of them can be applied directly without further ontology development, particularly for most in the aspects of manufacturing processes and concepts. Both BOT and BPO ontologies were purposely implemented as lightweight ontologies and do not include semantics for offsite manufacturing. Also, they are mainly developed to support a product-based or component-based design approach, typically not supporting the consideration of the resources, activities and processes in production directly. DOCK 1.0 and DiCon develop semantics for processes of construction works but are largely based on a site management context.

Manufacturing-oriented methodologies, such as MASON, or generic product ontologies, such as GoodRelations, fail to capture the richness and complexity of buildings. Finally, existing cost estimation ontologies adopted for producing estimates are building component-based or resource-based rather than process-based. The latter is needed to assess the cost implications for offsite systems, as individual systems have corresponding production methods, which will incur cost differently.

A DfMA-specific ontology (i.e. OHO) was developed in this study to develop the semantics that represent manufacturing processes and concepts as well as the associated cost. Key terms and relationships in relation to production and assembly, are explicitly defined. OHO has concepts that are matched to those in BOT and BPO, which can be connected to both ontologies through general associations rather than strict formal connections. It can complement BOT and BPO by adding new offsite construction knowledge (terminology and relationships), defining specific core vocabulary that can be used in DfMA practice.

4. Method

As identified in the review, the existing ontologies have their limitations in representing DfMA sufficiently for evaluation and process improvement. BPO is the closest to serve the function of evaluating alternative design but it does not include the specific product data needed or the process-related classes. In order to design the OHO ontology common techniques recommended by well established ontology development methods were applied. As in-depth knowledge is needed to develop the complex ontologies, this study used a house production case that has adopted DfMA for its automated wall panel design as the basis for the development of an offsite house design ontology, namely Offsite Housing Ontology (OHO). OHO is a comprehensive representation of DfMA house design. It contains the features that are needed to evaluate DfMA processes for housing.

4.1. Ontology engineering method

The methodology used to design OHO follows best practices for defining such domain ontologies, including the use of competency questions, re-using existing ontological resources (e.g. process maps for production), creating links to other ontologies (e.g. ifcOWL), and using competency questions to assess the ontology. The development of OHO followed the NeOn methodology framework, which is a scenario-based methodology aiming to speed up the development of ontologies through reusing existing ontologies, non-ontological resources and ontology design patterns [44]. There are various models comprising various phases and scenarios to implement NeOn. The Six-Phase Waterfall Ontology Network Life Cycle Model was chosen as it incorporates 2 phases, i.e. reuse and re-engineering phases that are very relevant to the context of OHO, in addition to the base 4-phase model comprising initiation, design, implementation and maintenance phases. For each phase, NeOn has a number of scenarios (9 in total) for users to define the tasks for individual phases. Figure 1 shows the phases and the corresponding scenarios chosen according to NeOn framework for OHO development.

Table 2

DfMA requirement and native support in existing ontologies.

	DfMA requirements				
Existing Ontologies	Building-specific	Offsite aspects	Manufacturing Processes and Concepts	Cost estimation	Complexity
ifcOWL	Yes	No	No	No	High
BOT	Yes	No	No	No	Low
BPO	Yes	Yes	No	No	Low
MASON	No	No	Yes	Yes	Medium
GoodRelations	No	No	No	No	Very Low
PRODUCT / PROPS	Partially	No	No	No	Low
Liu et al [43]	Yes	No	No	Yes	Very Low
DiCon	Partially	No	No	No	Medium
DOCK 1.0	Partially	No	No	No	-

A summary of the Ontology Requirements Specification Document (ORSD) was produced (Table 3) in the *Initiation* phase. Then, an analysis of non-ontological resources such as BIM model, manufacturing process reports etc. was done in the *Reuse* phase. In the *Re-engineering* phase, a literature review was conducted on: i) existing ontologies based on IFC; ii) existing ontologies for building design, construction, and related domains; iii) concepts and relations in addition to what was identified in i) and ii). The review identified a lot of shortcomings for DfMA implementation and concluded that the existing ontologies do not represent offsite construction sufficiently.

Then, a set of competency questions were drafted with reference to the guide for developing an ontology from Stanford University [44] in the *Design* phase. The competency questions guided the discussion with the stakeholders and experts of the offsite house production case involved, who are the architect, the production engineer, site engineer, steel manufacturer, client and cost consultant (Hereafter, the term ‘group of experts’ is used to refer to this set of respondents). There were 4 rounds of group discussions and a number of one-to-one interviews. The competence questions were identified and agreed with the group of experts in the first meeting. Then, a meeting was carried out to agree on the data structure for the various OHO modules. One to one interviews were then conducted to clarify and confirm the concepts and relationships defined in the ontology individually with the relevant experts that hold the expertise in which OHO attempts to represent. The OHO ontology were then presented and refined in the last two group meetings. Finally, the OHO ontology was formalized in a machine understandable format using the OWL language in the *Implementation* phase.

4.2. *OHO Competency Questions*

The scope and purpose of the OHO ontology was evaluated using the following competency questions, which is a standard technique in ontology engineering.

- **CQ1**: What is a product production method?
- **CQ2**: What are the stations for a production line method?
- **CQ3**: What are the activities carried out in each station of a production line method?
- **CQ4**: What type of material is required to produce a product?
- **CQ5**: What type of labour profile is required to work in each activity?
- **CQ6**: What is the cost of a DfMA house?
- **CQ7**: How long does it take to produce a Product / DfMA Product?
- **CQ8**: What are the components of a Product?
- **CQ9**: What are the resources needed for the production of a Product?

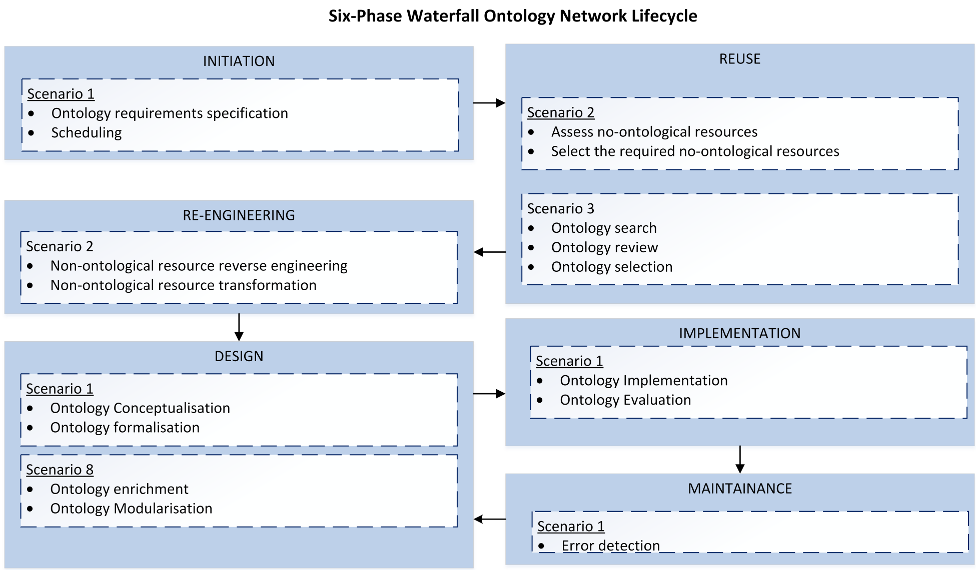


Fig. 1. Phases for development of OHO ontology

Table 3

Ontology Requirement Specification Document

Scope:

The ontology focuses on the products and components of a modular house designed using DfMA (hereafter named as DfMA house),
the process of manufacturing and assembling products, the resources consumed
, and their cost and carbon emissions quantitative impact.

Intended Users:

- Modular Manufacturers
- Architects
- Engineers
- Specifiers
- Quantity Surveyors
- Domain Experts

Intended Use:

- Use Case 1: Cost estimation (demonstrated in this paper)
- Use Case 2: Life cycle carbon emissions (future work)
- Use Case 3: Semantic digital twin of DfMA (future work)
- Use Case 4: Time monitoring using sensor information (future work)
- Use Case 5: Production waste classification (future work)

Ontology Requirements:

Group of Competency Questions defined in the *OHO Competency Questions* section.

5. OHO Overview

OHO is a DfMA domain ontology, which complements existing ontologies. It defines a model of categories within the offsite construction manufacturing Universe of Discourse (UoD), plus sufficient knowledge about those categories to allow reasoning and automatic calculation. OHO is intended as a COmmon REference (CORE) model for offsite construction. The proposed ontological model is language independent, using the broader term 'terminology' for a semantic model linked to the offsite construction manufacturing domain.

At the core of the OHO ontology, a limited number of very high level concepts is needed. This high level schema is a prerequisite for good integration with other data models. This principle is similar to the principle behind BOT as a central core ontology [24]. This OHO core is illustrated in Figure 2 and it responds to the following requirements:

1. Fits closely with building standards especially in applications for design and manufacturing assembly or in the retrieval and classification of OHO concepts.
2. Sufficiently general to be used in different applications for decision support and interoperability.
3. Formally defined in OWL Description Logic (DL)
4. Acts as a general-purpose modelling language for offsite manufacturing.
5. Supports OWL-DL reasoners to allow core OHO concepts to be combined to create new descriptions of classes and instances constructed according to constraints implemented within the ontology.
6. Supports intuitive and practical collaboration between different groups, being easily understood and application independent.

Furthermore, the OHO core has two domains of specialisation, which are presented here as two modules that extend the core. An ontology module is a reusable component of a larger or more complex ontology [45], which is self-contained but has connections and associations with other ontology modules, and can be viewed as an extension of the original ontology [46]. OHO-Pro represents the production module (Production Section), allows the definition of production line data, and imports the OHO core. OHO-Cost, the costing module (OHO Costing Module Section), facilitates the cost estimation of a DfMA house. The OHO-Core, OHO-Pro and OHO-Cost are linked via explicitly defined relations as illustrated in Figure 2. These ontologies are defined in the namespaces and prefixes listed in Listing 1.

Listing 1: OHO prefix.

```
@prefix oho: <https://w3id.org/oho#> .
@prefix oho-pro: <https://w3id.org/oho
    ↪ -pro#> .
@prefix oho-cost: <https://w3id.org/
    ↪ oho-cost#> .
```

OHO-Core concepts are similar to BOT to allow for better alignment in a wider linked building data context, even more in specific scenarios it can even be replaced by BOT, BPO and alike ontologies when there is a need to use only one of the modules i.e. OHO-Pro. For instance, a `oho:House` *is-a* `bot:Building` or when the concepts are used together, they can be linked or distinguished using computer-readable relationships, such as `sameAs` and `differentFrom`.

5.1. OHO Core

The OHO core module describes the core parts of a house constructed with the DfMA approach. Given that manufacturing is a major part of DfMA, this core module also includes production line aspects. The main classes of the OHO core ontology are:

- `oho:Product` categorizes the modular products produced for the scope of the DfMA house (e.g. wall panels are represented as **oho:Product**);
- `oho:Production` defines the production line that produces an oho:product (e.g. panelized systems are represented as `oho:Production`);
- `oho:Component` gives the details of finished modular components installed in a DfMA house (e.g. kitchens or bathroom are represented as `oho:Component`);
- `oho:Interface` is a generic class that defines the type of relationship that connects products or components together

OHO-Core concepts are similar to BOT to allow for better alignment in a wider linked building data context, even more in specific scenarios it can even

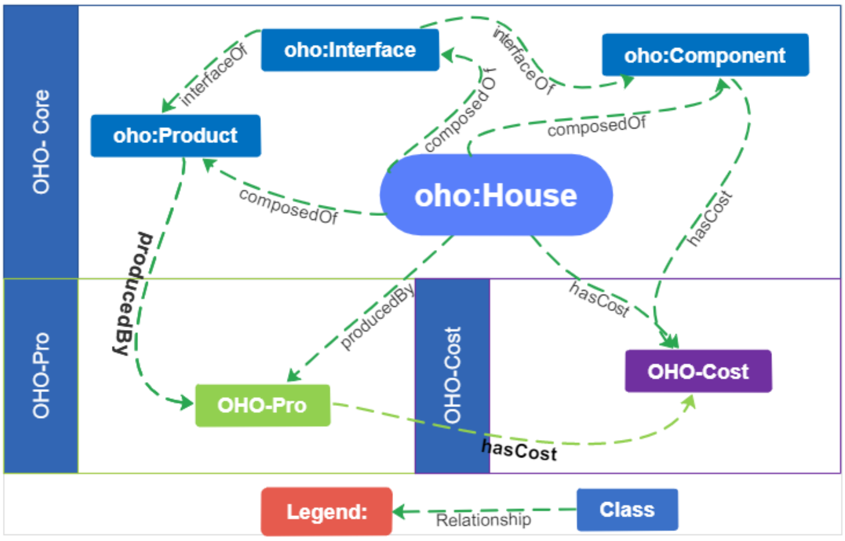


Fig. 2. Illustration of the OHO core classes.

be replaced by BOT, BPO and alike ontologies when there is a need to use only one of the modules i.e. OHO-Pro. For instance, a `oho:House` *is-a* `bot:Building` or when the concepts are used together, they can be linked or distinguished using computer-readable relationships, such as `sameAs` and `differentFrom`.A more detailed description of each of these classes is given below.

5.1.1. OHO House

In DfMA the design of the house and its individual components integrates production line concepts. In order to represent this semantically, the `oho:House` class functions as a central concept, combining all aspects that have to be defined in order for a house to be produced and built with the DfMA approach. The `oho:House` class can also have the object properties `oho:composedOf`, pointing either to a `oho:Product` or `oho:Component`.

5.1.2. Product

The class `oho:Product` defines an object that is manufactured in factory using a DfMA production line. The definition of an `oho:product`, and associated properties such as its cost, take into account the manufacturing concept of the production line. (By contrast, `oho:component` classes are cost via external procurement costs, as in typical modular construction). The `oho:product` class is elaborated further with the subclasses `oho:Pod`, `oho:Panel` and `oho:PodFrame`. Details of the production line and product attributes are required to describe such a product, using the object property `oho:producedBy`, which relates the product to a `oho:Production` object.

5.1.3. Component

The `oho:Component` class represents a part used to compose a house or a product, which is directly procured in a finished state (i.e. outside the DfMA production line), either as discrete components (e.g. windows, doors), or complete modular components (e.g. a kitchen). The subclasses of `oho:Component` are listed in Table 4. The composition of a DfMA house can be defined and inferred to a high level of granularity (e.g. up to the level of individual bolts in a connection), using transitivity between OHO classes. A `oho:composedOf` property has `oho:House` and `oho:Product` classes as a definition domain and `oho:Component` and `oho:Product` classes as a range. For example, if a DfMA house is composed of panels, and the panels are composed of studs, windows, studs etc., then it can be inferred that the DfMA house is composed of the window, the studs etc. as well.

5.1.4. Interface

An interface in the OHO ontology represents a physical connection layer between a product and a component, between components or between products. The `oho:Interface` class allows to define interfaces between products and components as needed. For example the connection points between multiple components of a product (subassemblies)

Table 4

A list of the subclasses of `oho:Component` class.

oho:Window	oho:KitchenCabinet
oho:Tape	oho:RecessedLight
oho:LightingCable	oho:Pipe
oho:Battery	oho:Socket
oho:LightSwithch	oho:SanitaryFitting
oho:SolarPanel	oho:BathroomCabinet
oho:Door	oho:PendantLight
oho:HeatRecocveryUnit	oho:Stair
oho:WhiteGood	oho:Radiator
oho:ElectricityCable	oho:Duck
oho:KitchenCable	

can be represented with an interface. At a higher level, an interface may also be used to define the connections between multiple products in an `oho:House`. Conceptually, this class draws from the `bot:Interface` class of the BOT ontology, but adjusted to a DfMA context.Products and components are linked with interfaces using the `oho:isInterfaceOf` property. The domain of this is an `oho:Interface` class and the range is defined by `oho:Product` and `oho:Component` classes.

5.1.5. *Production*

The class `oho:Production` describes the methodology used in producing an `oho:Product`. As the aim was to develop a modular and extensible ontology, detailed subclasses and object properties were not introduced in the core ontology. Instead, this core ontology was extended with a dedicated ontology aimed at the representation of production methods (Section 5.2). This approach allows:

- the use of other ontologies to define production processes, while still relying on the OHO core,
- the use of the simpler and more general OHO core in case no production details are present,
- management of the OHO Production ontology with appropriate scope and focus.

5.2. *OHO Production*

The OHO Production module (`oho-prod:`) imports the OHO core and defines several dedicated classes and properties. It consists of four parts, namely Method, Station, Activity, and Resource (see Fig. 3). First, the `oho-prod:Method` class allows to categorise different types of production line methods e.g. automated production line; second, the `oho-prod:Station` class allows to define the stations of a production line, it has a starting station (`oho-prod:isStartingStation`) a next station (`oho-prod:hasNextStation`) and a final station (`oho-prod:isFinalStation`); third, the `oho-prod:Resource` class defines the resources needed (related to **CQ9**) for a successful operation handled by a station i.e. `oho-prod:Material` or `oho-prod:Labor`; and the `oho-prod:Activity` class represents distinct activities in the entire production line.

5.2.1. *Production Line*

The `oho-prod:Method` gives a semantic description of a production line. It is composed of stations (`oho-prod:Station`) and assigned to a `oho:Product` via a `oho-prod:hasMethod`. The semantic information is used to answer **CQ1**. Table 5 shows a number of classes available in the OHO Production ontology, with corresponding real-world examples (i.e. implementation instances). This clearly indicates that a lot of diversity is present in stations, production lines, components and activities.

5.2.2. *Method*

The number and type of stations that constitute a production line are imposed by the production method. For example, a panelised system production line has a different production line compared to a pod system production line. A class `oho:Method` is therefore defined to reflect the production line method and each method is connected with the different stations needed to complete the production (object property `oho-prod:hasStation`). However, stations are defined as single instances and can be reused in

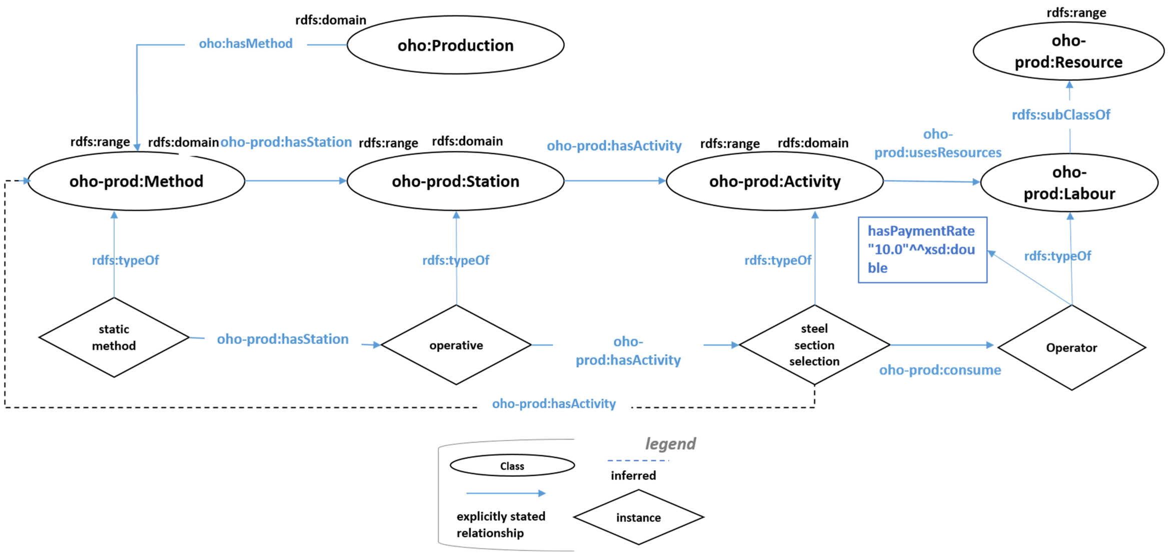


Fig. 3. Illustration of the oho:Production class, its subclasses and examples of instances e.g. static method as an instance of the class oho:Method.

different methods; e.g. a “Loading Station” can be defined once and the same instance reused in different production methods. A decision was made not to define production sub-methods directly in the OHO ontology. These, however, can be defined as subclasses of `oho-prod:Method`.

5.2.3. *Activity*

Capturing the activities that take place in a production line is an important part of a successful representation. The definition of those activities also enables the definition of constraints and checks on these activities. In this regard, OHO clearly divides production processes into activities that produce directly, e.g. `oho:CladdingAsssemblyLine`, and activities that support production (e.g. loading, packaging and transporting). For example, a cladding assembly line is an automated activity that is defined as a production activity; a constraint placed on it is that it can only start after the frame assembly line has been completed; a property of it is that it consumes labour. The constraints in Listing 2 indicate how such restrictions can be included in the data.

Listing 2: Assembly line definition including constraints that can be set on activities and resources these activities consume.

```
oho-pro:CladdingAssemblyLineAutomated
    ↪ :=
 oho-pro:CladdingAssemblyLine AND
oho-pro:beginsAfter ONLY oho-pro:
    ↪ FrameAssemblyLine AND
oho-pro:consumeLabour SOME oho-pro:
    ↪ Labour AND
oho-pro:hasSupportingActivity SOME (
  oho-pro:Loading AND
  oho-pro:consumeLabour SOME oho-pro:
    ↪ Labour )
```

5.2.4. *Station*

A `oho:Station` class defines a work station in the production line which covers **CQ2**. Each work station performs one or more production activities (e.g., cladding assembly line automation, frame assembly line) and needs time and resources (e.g., labour, material, component, overhead, etc.) to produce a product. These stations perform activities (related to **CQ3**), and those activities in turn use resources (define in a `oho-prod:Resources` property) and time (defined in a (`oho-prod:hasProcessingTime` property - **CQ7**). These stations are crucial to the production system in the OHO ontology. The resulting structure is shown in Fig. 4.

A method of production (`oho-prod:Method`) has one or more stations (`oho-prod:hasStation`) and each station has one or more activities (`oho-prod:hasActivity`). The last property is used in a property chain axiom that formalises the following transitive relationship: if a method has a

Table 5
OHO taxonomy of major elementary categories.

Entity Production Station	Example of Activity
Cladding-Assembly-Line	
CP-Boarding-Station	T10-Frame-transfer
	T11-Load-and-Place-CPB
	T12-Mechanical-Fixing
Adhesive-Station	T31-Feed-Adhesive
	T32-Dispense-Adhesive
Briquette-Apply-Station	T35-Place-Briquettes
Briquette-Load-Station	T34-Feeding-Briquettes
Frame-Assembly-Line	
Conveying-Station	T13-Return-Conveyor
Frame-Riveting-Station	T5-Rivet-Joints
	T6-Move-Frame-to-Lift
	T7-Lift-Frame
Frame-Transfer-Station	T9-Transfer-Frame
FrameBeam-Load-Station	T1-Deliver-Pallets
	T2-Select-and-Load-Beam
	T3-Clamp-Beam
Supporting-Activity	Quality-Inspection

station that has an activity, then the method has that activity as well (Listing 3).

Listing 3: Property chain axiom for stations activities and methods represented in Turtle Syntax.

```
PREFIX owl: <http://www.w3.org
    ↪ /2002/07/owl#>
PREFIX oho-pro: <https://w3id.org/oho-
    ↪ pro#>
oho-pro:hasActivity owl:
    ↪ propertyChainAxiom
   (oho-pro:hasActivity oho-pro:
       ↪ hasStation ).
```

Similarly, a property chain axiom for resources is defined: if an activity performing on a station of a production method uses some resources, then the station where the activity takes place, uses these resources as well (Listing 4).

Listing 4: Property chain axiom about resource activity and station presented in Turtle syntax.

```
PREFIX owl: <http://www.w3.org
    ↪ /2002/07/owl#>
PREFIX oho-pro: <https://w3id.org/oho-
    ↪ pro#>
oho-pro:usesResources owl:
    ↪ propertyChainAxiom
   (oho-pro:usesResources oho-pro:
       ↪ hasActivity ).
```

5.2.5. *Resource*

Resources in OHO are classified in these subclasses: `oho-prod:Material`, `oho-prod:Labour`, `oho-prod:Plant` and `oho-prod:Overhead`.

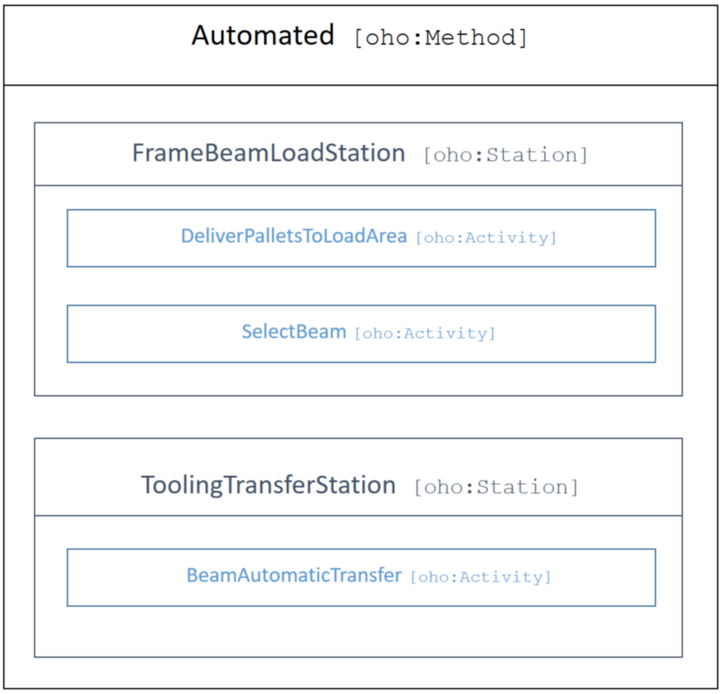


Fig. 4. Illustration of the class dependency of the automated production line.

The oho-prod:Material class collects instances that represent the different types of material needed and used for the production of a product such as a wall panel. This class is described in more detail in the following section. The oho-prod:Labour class captures data about the type of labour that is engaged in the production process and is sub categorized as oho-prod:Skilled, oho-prod:SemiSkilled and oho-prod:Unskilled. This categorisation responds to **CQ5**. Different payment rates can be assigned to different labour types using the oho-prod:hasPaymentRate data property and the activity it is allocated to (oho:workingOnActivity) for a specified amount of time. The class oho-prod:Plant gives details about the plants (e.g., movable tools, static tools) used in the production process and oho-prod:Overhead defines activities of type cleaning, security etc.

The oho-prod:Material class defines the types of material used in different activities of the DfMA house production and assembly (related to **CQ4**). Each material is represented as an instance of this class and is semantically enriched with details for: its cost (oho-prod:hasCost); the source (oho-prod:hasSource) of information such as a reference database of rates; the conversion factor (oho-prod:hasConversionFactor); the quantity of the material (oho-prod:hasQuantity). Cost rates are affected by both the vendor and the time. In order to model this relationship, the oho-prod:hasSource property is used to capture the vendor, and the state property is used to capture the time. The latter is inspired by the OPM ontology [47].

A sample list of the directly procured materials for producing a Pod sub-assembly extracted from a BIM model is listed in Table 6.

The class oho:Material is useful in many way such as to get information about the material used in producing a product or in cost calculation or estimation.

5.3. OHO Costing module

While many ontologies in the LBD community are oriented towards the definition of the geometry and semantics of a building or its elements, one of the advantages of the OHO ontology is that it combines this with the capacity to produce cost estimates. The second OHO ontology module that was developed facilitates the cost estimation of a DfMA

Table 6

Directly procured material for Pod sub-assembly.

Plasterboard	Hot Water Cylinder
Joint tape to internal walls	Electrical Installation
Insulation to external wall	Lamps
Stairs and balustrades	Basin
Timber studs	Battery System
Kitchen Cabinets	Solar PV
Bathroom Cabinets	Telephone and TV Distribution
Sockets	Energy and fault monitoring
Light switches	Bathtub
Pendant lights	Shower screen
Recessed lights	Taps
Lighting Cable	Water Cistern
Data Cable	Shower Thermostatic Mixer
Electricity Cable	Kitchen sink
Hot and cold water pipe	Kitchen mixer
Drainage pipe	Smoke detector
Panel Heaters	Heat detector
MVHR	CO2 detector

house (related to **CQ6**). The main drivers of the cost estimation of a DfMA house (Figure 5) are defined in the following classes:

- `oho-cost:BussinesOperation`,
- `oho-cost:DesignAndEngineering`,
- `oho-cost:OffsiteProduction`,
- `oho-cost:Transportation`,
- `oho-cost:OnsiteProduction`,
- `oho-cost:InUse`.

The cost estimation for a DfMA house is activity-based, which accounts for the chain of activities in the offsite production. In order to capture this process, the `oho:OffsiteProduction` class is defined with a high level of granularity and detail.

6. Evaluation of OHO

According to the NeOn methodology, there are three evaluation criteria: i) *Domain coverage*, ii) *Quality of the modeling* and iii) *Suitability for an application or task.* An additional criterion suggested according to NeOn, i.e. *Adoption and use*, can only be evaluated over time, and thus is excluded from this study. OHO was assessed by the group of experts working in the AEC domain.

6.1. *Domain coverage*

The first version of the ontology was built by extracting knowledge from non-ontological resources and reviewing existing ontologies. Subsequent to the release of the first version, a few focus group meetings with the group of experts were conducted to refine OHO. Additional concepts and relationships were defined, ambiguous concepts were clarified and some terms were amended to facilitate common understanding.

6.2. *Quality of the modeling*

This evaluation criterion assesses the quality of the ontology based on a set of metrics and a list of attributes of the ontology development process. OHO contains 151 classes, with 61 object properties, and 77 data properties. It was tested using Protégé to demonstrate that the schema is consistent. A comparison between OHO and other ontologies is presented later in this article.

The following attributes are used to evaluate the ontology development process [48]:

- **Accuracy**: The ontology development process was assisted from the group of experts in the

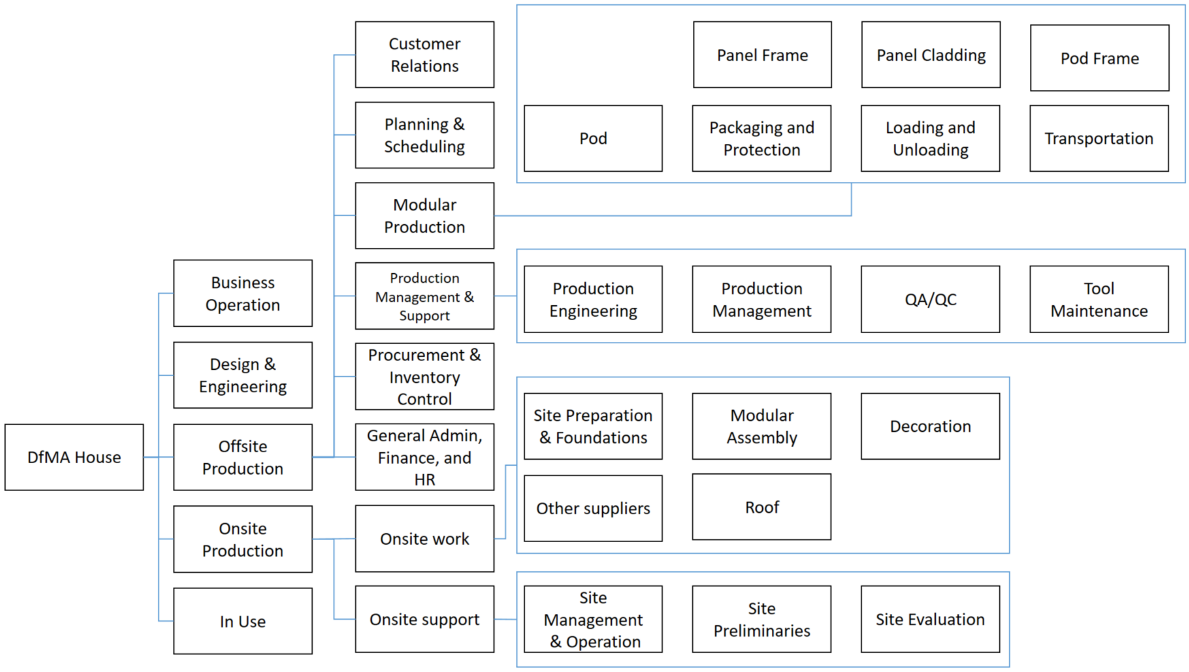


Fig. 5. Main drivers of a DfMA house project implementation.

DfMA domain. The processes were supported by highly accurate non ontological resources such as the BIM model among other sources.

- **Adaptability**: OHO is a modular ontology and each module can be used independently and this provides reusability and extensibility. The OHO-Cost module can be used to estimate traditional construction project as well as offsite construction and OHO-Pro can be reused for other types of production lines.
- **Clarity**: All the defined terms contain non-ambiguous names and are labelled with definitions and supplementary information to ensure common understanding.
- **Completeness**: The OHO ontology can answer all the competency questions defined in the ontology requirement specification.
- **Efficiency**: Querying the ontology using the SPARQL query language protocol was tested in the Protege framework and in GraphDB[1]. The queries run in milliseconds in each of these environments.
- **Conciseness**: The knowledge modelled in the OHO ontology and its modules was captured from domain trusted sources, i.e. knowledge shared from the group of experts in this case.
- **Consistency**: Reasoning based on OHO was performed using the Pellet reasoner[2]. No inconsistencies were found.

6.3. *Suitability for an application*

The OHO ontology was used to implement a KBE tool for cost estimation of housing projects using the DfMA approach. The application is implemented according to the Representational State Transfer (RESTful) architectural pattern, with an application programming interface (API) offering intermediary services to other web frontend endpoints and acts as a gateway to the OHO ontology and database servers. A web-based user friendly interface is provided for potential users such as architects, production engineers, structural engineers, steel suppliers, clients and cost consultants, etc.

6.3.1. *Input of the KBE tool*

DfMA houses are platform-based with standardised design prototypes. To evaluate the cost of a DfMA house, the following input is required:

[1]https://graphdb.ontotext.com/
[2]https://www.w3.org/2001/sw/wiki/Pellet

1. Choice of sub-assembly system
2. Offsite sub-assembly approach
3. Batch size and expected batch delivery period
4. Project site location
5. Project size in terms of house numbers and types of houses
6. Design choice
7. Service installation choices

They are defined according to a set of entries in an evaluation form as shown in Figure 6.

As the data of DfMA houses are kept in the BIM model and various data sheets in the project data environment, the KBE tool has the necessary data for the cost estimate (Figure 7).

6.3.2. *Output of the KBE tool*

The cost estimation module produces cost estimation as well as the breakdown of the costs. In addition to the overall cost estimations, there are 4 levels of breakdown: activity group, activity, sub-activity, and resource. As the focus of the prototype is on Manufacturing and Assembly, all four levels of breakdown are available for the activities in relation to the manufacturing and assembly processes. The estimations and breakdown are shown in an interactive dashboard for the ease of visualisation. A screenshot of the dashboard is shown in Figure 8.

6.3.3. *KBE Architecture*

The KBE tool architecture (Figure 9) is designed with the capability to integrate and process heterogeneous data formats and to accommodate semantic web and web technologies. The data used are based on a platform-based modular house developed in a UK government funded industrial research project by implementing DfMA design methodology. Detailed information was produced for the construction of a prototype of the modular house. The source of data includes a BIM model in an IFC format as well as cost data sheets and other databases. These data were kept in a common data environment accessible by the KBE tool. The BIM model has gone through different stages of optimisation in the project life cycle. Simultaneously, the ontology has learned from the changes made in the design development.

The *Data* layer stores different file formats that contain information used for cost calculation, such as BIM model in IFC format of the house, productivity datasheets,and other sources of information. The `Transformation Layer` parses the different data sources and transforms the relevant information into the RDF format. This data is finally fed into the `DfMA Knowledge Graph` where the data is semantically described with the OHO ontology concepts and relations creating a DfMA Knowledge Graph. A Knowledge Graph is a knowledge bases that store factual information in form of relationships between entities [49] described with formal semantics. The OWL API [50] library and IFCtoLBD [51] converter are used to facilitate the conversion of these semi structured data to a LBD format.The *SPARQL* query protocol or other Query API serves as an intermediate layer between the user `Application` layer and the other layers of the architecture. The user sends queries by entering filtering information and receives a visualized response.

6.3.4. *Validation of the KBE tool*

The KBE tool was presented and tested by the group of experts. The group of experts found the outputs generated were close to their estimates. The estimated costs generated from the KBE tool were also compared against estimates from other data source to validate the output of the tool.

7. Application of the OHO ontology

7.1. *DfMA house wall panels and components of a wall panel*

In order to demonstrate the functionality of the OHO ontology, a case study of a DfMA house composed of 32 wall panels was registered by instantiating the ontology. Different queries were executed (such as “what are the components of a wall panel?”, “what is the production time?”) in order to extract the captured and inferred data. The wall panels were modelled by describing their attributes, and the production line was detailed in terms of the activities required to produce a wall panel (Figures 10, 11).

All activities are related to each other, they are the first activity (`oho:hasStartingActivity`), after or before another activity (`oho:hasNextActivity`) or in parallel to another activity. The knowledge represented in the ontology was used to estimate cost per each activity and the overall cost of producing a sample product (e.g. a wall panel). By estimating the cost per each activity, a designer can get insights about the related activities and the assigned overhead costs. They can revise the design if needed.

The following are example SQWRL (Semantic Query-Enhanced Web Rule Language) queries that

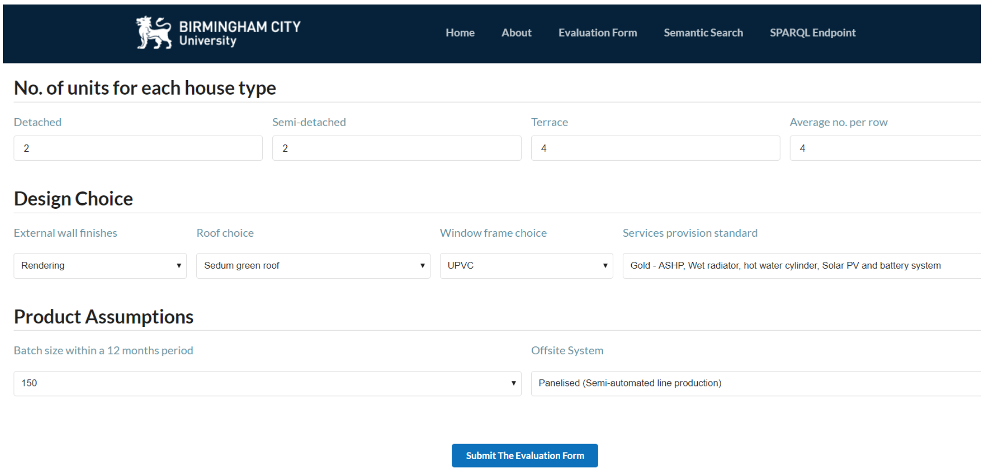


Fig. 6. Interface of the web evaluation form.

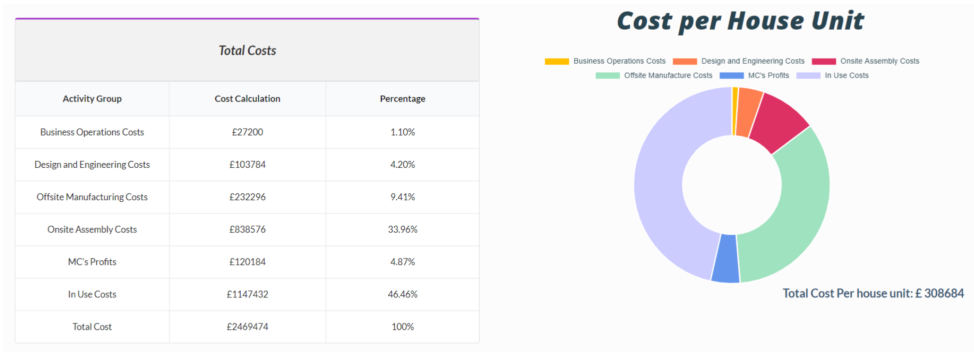


Fig. 7. Dashboard screenshot of cost estimation from KBE tool.

were executed with the Protégé 5 SQWRL plugin in relation to our example. The `sqwrl` prefix is used to denote the SQWRL operator and `swrlb` for identifying SWRL built-ins.

Q1: What is labor cost for each semi-skilled operative working on each activity of the wall panel production?

Listing 5: Semi-skilled labour cost working on a wall panel production activity.

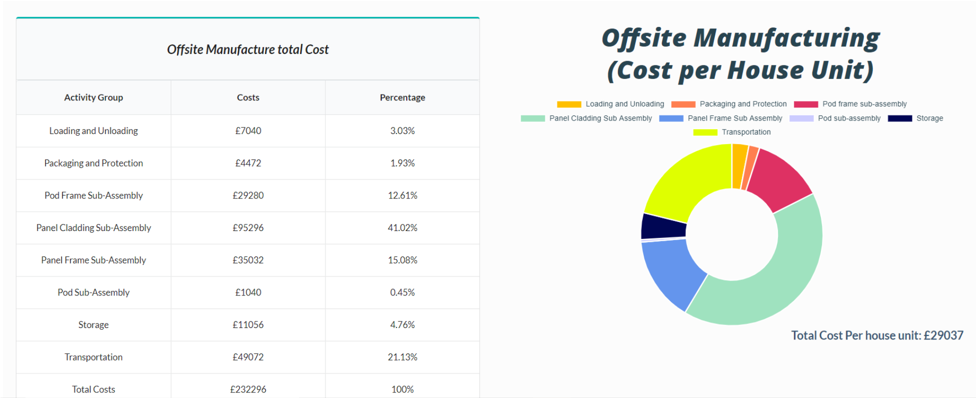
```
PREFIX swrl: <http://www.w3.org
    ↪ /2003/11/swrl#>
PREFIX swrlb: <http://www.w3.org
    ↪ /2003/11/swrlb#>
PREFIX sqwrl: <http://sqwrl.stanford.
    ↪ edu/ontologies/built-ins/3.4/
    ↪ sqwrl.owl#>
Semi-Skilled_Operative(?s) ^
    ↪ Activities(?a) ^
    ↪ workingOnActivity(?s, ?a) ^
    ↪ hasProcessTime(?a, ?p) ^
    ↪ hasLabourHrRate(?s, ?r) ^ swrlb:
    ↪ multiply(?result, ?p, ?r) ->
    ↪ sqwrl:sum(?result) ^ sqwrl:
    ↪ select(?s)
```

Q2: What are the parts (related to **CQ8**) of the product *WallPanel01* wall panel?

Listing 6: List of the components of a wall panel.

```
Product(?p) ^ hasComponentPart(?p, ?
    ↪ Component) -> sqwrl:select(?
    ↪ Component)
```

Q3: What is the starting and upcoming activity for producing *WallPanel02* wall panel?



Offsite Manufacture total Cost

Activity Group	Costs	Percentage
Loading and Unloading	£7040	3.03%
Packaging and Protection	£4472	1.93%
Pod Frame Sub-Assembly	£29280	12.61%
Panel Cladding Sub-Assembly	£95296	41.02%
Panel Frame Sub-Assembly	£35032	15.08%
Pod Sub-Assembly	£1040	0.45%
Storage	£11056	4.76%
Transportation	£49072	21.13%
Total Costs	£232296	100%

Fig. 8. Dashboard screenshot of production cost breakdown

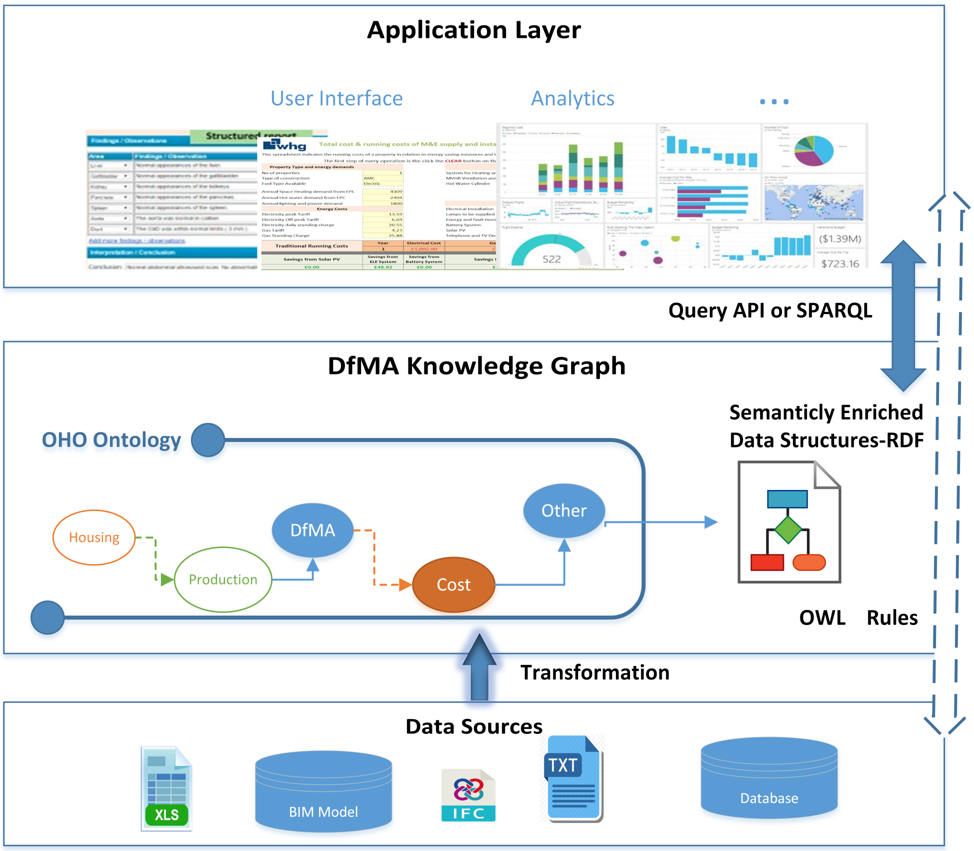


Fig. 9. Web-Application Architecture.

Listing 7: Starting and upcoming activities for producing a wall panel.

```
Product(LSF_3BED_01_LHS) ^
    ↪ hasStartingActivity(WallPanel02,
    ↪ ?StartActivity) ^
    ↪ hasNextActivity(?StartActivity,
    ↪ ?NextActivity)->sqwrl:select(
    ↪ WallPanel02, ?StartActivity, ?
    ↪ NextActivity,)
```

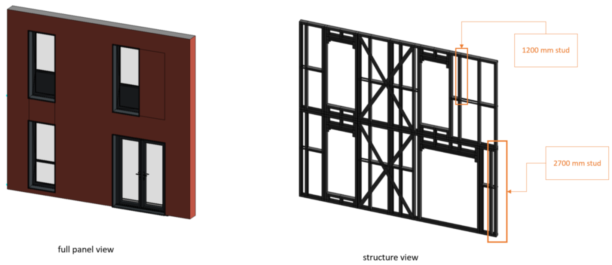


Fig. 10. The image of a wall panel in a DfMA house.

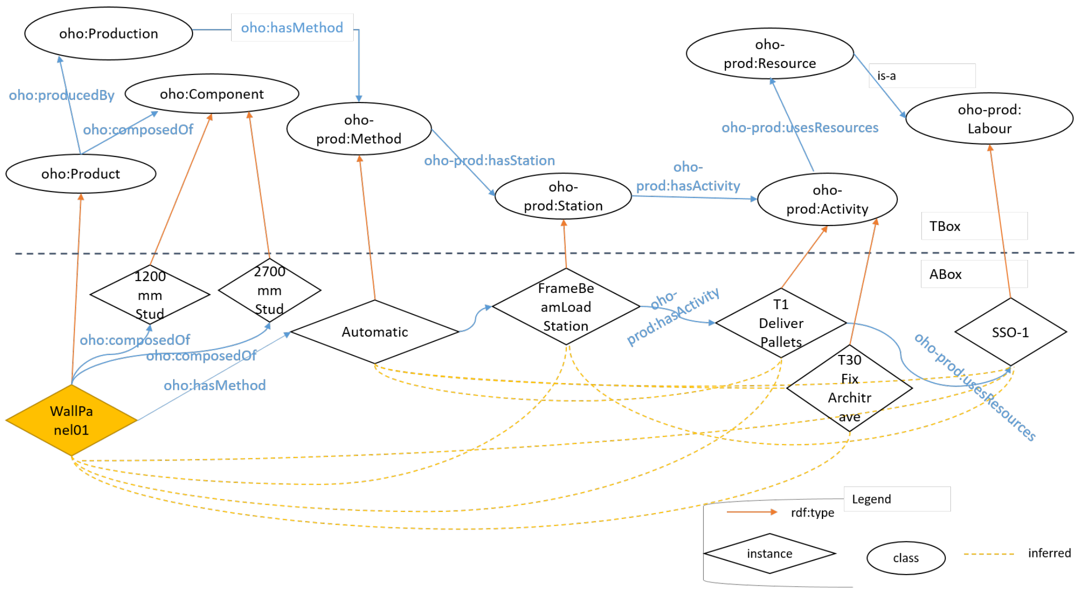


Fig. 11. The representation of a wall panel of a DfMA house in the OHO ontology.

7.2. Querying a Pod product

The level of granularity OHO enables many applications, including knowledge extraction using semantic search. Listings 14 to 10 show a number of relevant SPARQL queries to extract knowledge from a pod product. The queries are executed from Protégé 5 Snap SPARQL Query plugin and the inferred knowledge is leveraged, benefiting from the rules used.

Listing 8: Select the list of directly procured material needed for the pod production and their cost.

```
PREFIX rdf: <http://www.w3.org
    ↪ /1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org
    ↪ /2002/07/owl#>
PREFIX rdfs: <http://www.w3.org
    ↪ /2000/01/rdf-schema#>
```

```
PREFIX xsd: <http://www.w3.org/2001/
    ↪ XMLSchema#>
PREFIX oho: <https://w3id.org/oho#>

SELECT *
WHERE { ?s a oho:
    ↪ DPMforPod_Sub_Assembly; oho:
    ↪ hasActivityCost ?c }
```

Listing 9: Select the method of production for pod frame therefore the stations and activities.

```
PREFIX rdf: <http://www.w3.org
    ↪ /1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org
    ↪ /2002/07/owl#>
PREFIX rdfs: <http://www.w3.org
    ↪ /2000/01/rdf-schema#>
```

```
PREFIX xsd: <http://www.w3.org/2001/
    ↪ XMLSchema#>
PREFIX oho: <https://w3id.org/oho#>

SELECT ?m WHERE { ?p a oho:Product;
    ↪ oho:hasMethod ?m}
```

Listing 10: Select the 'parts' that compose a pod.

```
PREFIX rdf: <http://www.w3.org
    ↪ /1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org
    ↪ /2002/07/owl#>
PREFIX rdfs: <http://www.w3.org
    ↪ /2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/
    ↪ XMLSchema#>
PREFIX oho: <https://w3id.org/oho#>

SELECT ?m WHERE { ?p a oho:Product;
    ↪ oho:composedOf ?m}
```

7.3. *Cost Estimation of DfMA using a Semantic Rule Language*

Rules in the form of the Semantic Web Rules Language (SWRL) are used to provide more powerful deductive reasoning capabilities than OWL alone. For example, the rule in Listing 11 enables the calculation of the labour cost.

Listing 11: Labour cost per activity calculated from a SWRL based on the time spent on an activity and the labour rate.

```
Activities(?a),Labour(?s),
    ↪ hasLabourHrRate(?s, ?r),
    ↪ hasProcessTime(?a, ?p),
    ↪ workingOnActivity(?s, ?a),
    ↪ multiply(?result, ?p, ?r) ->
    ↪ hasActivityCost(?a, ?result)
```

Also, a dedicated set of SWRL rules has been created in addition to the presented OWL ontology modules, in order to evaluate their accuracy for cost estimation. A hypothetical project containing the offsite production of 200 houses of different types (i.e. detached, semi-detached and terrace) that uses an automated production method were registered as input to the ontology using the terms of the OHO core, production and cost ontology modules and the full list of requirements defined as shown Table 7.

As the OHO-Cost ontology module applies activity-based costing methodology, all activities are instantiated with their respective cost values. A starting activity is defined (`oho:isStartingActivity`) and associated with other cost activities using the `oho:hasNext` property. SWRL rules were set to chain the cost value instances together and at the same time sum them up to an accumulated cost (`oho-cost:hasActivityCost`) for each activity stage. Furthermore, a final overall inferred estimated cost is calculated at the end of the final activity (`oho-cost:hasActivityCostFromStart`). The rules that support the cost estimation of an offsite housing project are presented in Listings 12 and 13.

Listing 12: Assigning the cost to the starting activity of a process.

```
hasActivityCost(?a, ?c) ^
    ↪ isStartingActivity(?a, ?b) ->
    ↪ hasActivity-CostFromStart(?a, ?c
    ↪ )
```

The rule on Listing 12 states that if an activity a is the starting activity of a chain of activities and has a given cost c then the value of `oho:hasActivityCostFromStart` will be c also. The main purpose of this rule is to initialize the `oho:hasActivityCostFromStart` data property.

Listing 13: Updating the overall cost estimation after each activity is added.

```
Rule 3: swrlb:add(?nptd, ?ptd, ?npd) ^
    ↪  hasNextActivity(?p, ?np) ^
    ↪ hasActivityCostFromStart(?p, ?
    ↪ ptd) ^ hasActivityCost(?np, ?npd
    ↪ ) -> hasActivityCostFromStart(?
    ↪ np, ?nptd)
```

The `oho-cost:hasActivityCostFromStart` is updated by adding the value of the current cost activity found in `oho-cost:hasActivityCost`. The new value is assigned to the current cost activity `?np` and is a result of the cost activity np and the cost carried on the

Table 7

Project requirement specification.

Requirement	Value
Choice of offsite main frame sub-assembly systems:	Volumetric Type 1 (Pod with infill panel)
Production method	Automated panel
Batch size	200
Period to produce the batch	12
Project Assumptions	
Address (Post Code)	B12 0NW
No of units for each house type	
Detached	2
Semi-detached	2
Terrace	4
Average no. per row	4
Service System choice	
MVHR	Yes
Hot Water Cylinder type	Daikin
Electrical Installation Type	Extreme Low Energy
Lamps to be supplied	Yes
Energy and fault monitoring required	Yes
Battery System	Yes
Solar PV	Yes
Telephone and TV Distribution	Yes

`oho-cost:hasActivityCostFromStart` property from the start of this chain of cost activities (Listing 13).

The data input to the Pellet reasoner and those input to OHO OWL ontology in a Protégé environment give exactly the same result. Explanations for the semantic reasoner (e.g., pellet) inferences is available in the justification of results that is provided as an advanced functionality in Protégé. For the expert that does the evaluation directly using the OHO ontology and the SWRL rules for cost estimations, an ontology-based approach can provide the insight into the reasons behind estimations. The available explanations can serve as a white-box proof of the results and hence, plays an important role in building trust of OHO.

7.3.1. Validating data integrity with SHACL

OWL and SWRL are based on logic, do not support non-monotonic reasoning and use the Open World Assumption (OWA) where the missing data is assumed as not identified yet. In cases where validating data integrity is crucial, there is a need to use the Closed World Assumption (CWA) in order to notify a constraint violation or take action by adjusting data to a standard format. SHACL shapes is one of the newest technologies that has filled this gap in the Semantic Web architecture stack. Listing 14 shows an example of a SHACL shape for validating the properties (`oho:hasHeight`, `oho:hasWidth` etc.) of a `oho:WallPanel` instance before it goes to production. SHACL validation will ensure that geometric inputs conform to the required standards (an OWL to STEP generated file).

Listing 14: Data integrity validation using SHACL model.

```
@prefix rdf: <http://www.w3.org
    ↪ /1999/02/22-rdf-syntax-ns#> .
@prefix sh: <http://www.w3.org/ns/
    ↪ shacl#> .
@prefix xsd: <http://www.w3.org/2001/
    ↪ XMLSchema#> .
@prefix rdfs: <http://www.w3.org
    ↪ /2000/01/rdf-schema#> .
```

```
@prefix owl: <http://www.w3.org
    ↪ /2002/07/owl#> .
@prefix oho: <http://w3id.org/oho#> .

oho:WallPanelShape
   a sh:NodeShape ;
   sh:targetClass oho:WallPanel ;
   sh:property [
      sh:path oho:hasHeight ;
      sh:minCount 1 ;
      sh:datatype xsd:decimal ;
   ];
   sh:property [
      sh:path oho:hasWidth ;
      sh:minCount 1 ;
      sh:datatype xsd:decimal ;
   ];
   sh:property [
      sh:path oho-prod:hasLength ;
      sh:minCount 1 ;
      sh:datatype xsd:decimal ;
   ];
   sh:property [
      sh:path oho:hasFrame ;
      sh:minCount 1 ;
      sh:class oho:Frame ;
   ].
```

8. Conclusion and Discussions

A new domain specific ontology, OHO, was designed and validated to represent and bring together disparate and isolated knowledge and data from various fields. The main contribution of the paper is the very detailed ontologies developed for offsite house construction that supports building design using DfMA. The concepts and relationships defined particularly about the production and costing enable users to retrieve knowledge that can support DfMA design. The use of it is demonstrated from the answers to the competence questions as well as the KBE cost estimation tool.

OHO emerged from the ifcOWL-DfMA ontology [52], which expanded the ifcOWL ontology and derived the core elements of the OHO ontology as a result and was used in various real-life use case production lines to test and demonstrate the benefits of the semantic digital twin in obtaining data of the manufacturing for assessment [6]. Analysing the specifics of the production processes we identified the need for OHO to grow as a separate domain ontology. Naturally, many aspects of a completed DfMA project, such as the building geometry or the material properties, fits ifcOWL concepts and can be represented accordingly. However, as a process with roots in industrial engineering, DfMA engages more with procedural and optimisation aspects, and introduces concepts, such as “assembly” or “sub-assembly”, with different semantics from current BIM and construction technology practice. As such, a separate ontological domain was considered necessary in order to avoid semantic and ontological conflicts, as well as to implement DfMA concepts appropriately. Ideally, OHO is able to facilitate a two-way conversation: enable AEC practitioners to apply DfMA design concepts in a BIM workflow, while simultaneously acting as an introduction to the DfMA concept for BIM-literate AEC practitioners.

Furthermore, such modular extension approach also entirely fits the recommendations put forward by the W3C Linked Building Data Community Group, as well as linked data communities at large. By aiming at a separate domain, such recent research initiatives are followed and implemented, thus moving away from a monolithic ontology approach. Attempting to capture all possible aspects of a building (or any other concept) in a single ontology, mapped to a super-schema, has innate limitations and lacks the flexibility to accommodate different design concepts (i.e. extensibility). DfMA is the future innovative philosophies and practices for construction. It is likely to face similar challenges in BIM implementation.

The OHO ontology can be aligned with the IFC ontology by asserting that every `oho:Building` is an `IfcBuilding` and every `oho:Product` is an `IfcProduct`. An exploitation of OHO alignments with other standardised AEC ontologies such as BOT ontology are made such as an alignment in the instance level i.e. a DfMA product such as Pod (`oho:Product`) is a `bot:Element` and a DfMA house (`oho:House`) is a subclass of a `bot:Building`. The current version of OHO can be added as a module of a standardised ontology, and to make links on instance level (for web of linked data). A formal alignment approach will be developed further after OHO reaches a more mature stage. The instance linking proposed in the paper is recommended as a priority over more stringent ontology alignments due to practical implementations and research initiatives.

There are many directions that the base OHO ontology can be expanded to or used for when building DfMA applications. For example, in this paper, the OHO core semantic model was applied to develop the production and costing modules for cost estimation tool. The use of the OHO ontology has also been experimented in different environments in order to accommodate diverse users' needs with varying levels of knowledge of the underlying semantic web technology including some users with a computer science background and most from the AEC community. Users with knowledge of using an ontology editor (e.g., Protégé) are able to directly interact with the semantic model, while less experienced users required a frontend application that uses an API (e.g. OWL API) to connect with the OHO ontology as discussed in the *OHO Evaluation* Section.

A deep analysis of the best approach to follow in terms of efficiency, complexity and ease of use will be part of future works. Finally, the OHO ontology paves the way to the implementation of Digital Twins that integrates production information of houses with BIM models. Future developments of this work will focus on further iterations of the OHO ontology development life cycle, by expanding and improving the ontology based on more cases of production processes for offsite construction.

9. Acknowledgments

This research was supported by an Innovate UK funded project (Project no. 104798) "Collaborative Knowledge-Based DfMA approach to building cost efficient, low impact and high performance houses". The authors would like to thank the participating industrial partners of the research project including Walsall Housing Group, Hadley Group, QM systems, and Northmill Associates and Energy Systems Catapult.

References

[1] R. Hairstans and R.E. Smith, Offsite HUB (Scotland): establishing a collaborative regional framework for knowledge exchange in the UK, *Architectural Engineering and Design Management* **14**(1–2) (2018), 60–77.

[2] G. Boothroyd, Product design for manufacture and assembly, *Computer-Aided Design* **26**(7) (1994), 505–520.

[3] H.W. Stoll, Design for manufacture: an overview (1986).

[4] R. Bogue, Design for manufacture and assembly: background, capabilities and applications, *Assembly automation* (2012).

[5] M. Arif, N. Blismas and R. Wakefield, Drivers, constraints and the future of offsite manufacture in Australia, *Construction innovation* (2009).

[6] K.O. Ayinla, F. Cheung and A.-R. Tawil, Demystifying the concept of offsite manufacturing method: Towards a robust definition and classification system, *Construction Innovation* **20**(2) (2020), 223–246.

[7] A.-Q. Gbadamosi, A.-M. Mahamadu, L.O. Oyedele, O.O. Akinade, P. Manu, L. Mahdjoubi and C. Aigbavboa, Offsite construction: Developing a BIM-Based optimizer for assembly, *Journal of Cleaner Production* **215** (2019), 1180–1190.

[8] M.B. Murtaza, D.J. Fisher and M.J. Skibniewski, Knowledge-based approach to modular construction decision support, *Journal of Construction Engineering and Management* **119**(1) (1993), 115–130.

[9] C. Pasquire, A. Gibb, N. Blismas et al., What should you really measure if you want to compare pre-fabrication with traditional construction?, in: *13th International Group for Lean Construction Conference: Proceedings*, International Group on Lean Construction, 2005, p. 481.

[10] T. Tan, W. Lu, G. Tan, F. Xue, K. Chen, J. Xu, J. Wang and S. Gao, Construction-Oriented Design for Manufacture and Assembly Guidelines, *Journal of Construction Engineering and Management* **146**(8) (2020). doi:10.1061/(ASCE)CO.1943-7862.0001877.

[11] J.J. Shi, D.-E. Lee and E. Kuruku, Task-based modeling method for construction business process modeling and automation, *Automation in Construction* **17**(5) (2008), 633–640.

[12] J.E. Taylor and P.G. Bernstein, Paradigm trajectories of building information modeling practice in project networks, *Journal of management in engineering* **25**(2) (2009), 69–76.

[13] A.A. HIJAZI and H.A. OMAR, Level of detail specifications, standards and file-format challenges in infrastructure projects for BIM level three, *WIT Transactions on The Built Environment* **169** (2017), 143–154.

[14] B. Dave, A. Buda, A. Nurminen and K. Främling, A framework for integrating BIM and IoT through open standards, *Automation in Construction* **95** (2018), 35–45.

[15] M. Mayouf, M. Gerges and S. Cox, 5D BIM: an investigation into the integration of quantity surveyors within the BIM process, *Journal of Engineering, Design and Technology* (2019).

[16] J. Domingue, D. Fensel and J.A. Hendler, *Handbook of semantic web technologies*, Springer Science & Business Media, 2011.

[17] R. Masood, J.B. Lim and V.A. González, Performance of the supply chains for New Zealand prefabricated house-building, *Sustainable Cities and Society* **64** (2020), 102537.

[18] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane and S. O'Riain, Linking building data in the cloud: Integrating cross-domain building data using linked data, *Advanced Engineering Informatics* **27**(2) (2013), 206–219.

[19] P. Pauwels, S. Zhang and Y.-C. Lee, Semantic web technologies in AEC industry: A literature overview, *Automation in Construction* **73** (2017), 145–165.

[20] A. Wagner and U. Rüppel, BPO: The building product ontology for assembled products, *CEUR Workshop Proceedings* **2389** (2019), 106–119.

[21] A. Wagner, L. Möller, C. Leifgen and U. Rüppel, SolConPro: Describing multi-functional building products using semantic web technologies, in: *EWork and eBusiness in architecture, engineering and construction*, CRC Press, 2018, pp. 447–455.

[22] M. Hepp, GoodRelations: An ontology for describing products and services offers on the web, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **5268 LNAI** (2008), 329–346. ISBN 3540876952. doi:10.1007/978-3-540-87696-0-29.
[23] R.V. Guha, D. Brickley and S. Macbeth, Schema. org: evolution of structured data on the web, *Communications of the ACM* **59**(2) (2016), 44–51.
[24] M. Rasmussen, P. Pauwels, C.A. Hviid and J. Karlshøj, Proposing a central AEC ontology that allows for domain-specific extensions, in: *Proceedings of the Joint Conference on Computing in Construction (JC3)*, 2017, pp. 237–244.
[25] P. Pauwels and W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, *Automation in Construction* **63** (2016), 100–133.
[26] S. Lemaignan, A. Siadat, J.Y. Dantan and A. Semenenko, MASON: A proposal for an ontology of manufacturing domain, *Proceedings - DIS 2006: IEEE Workshop on Distributed Intelligent Systems - Collective Intelligence and Its Applications* **2006**(June 2014) (2006), 195–200. ISBN 076952589X. doi:10.1109/DIS.2006.48.
[27] Y. Zheng, S. Törmä and O. Seppänen, A shared ontology suite for digital construction workflow, *Automation in Construction* **132** (2021), 103930.
[28] T.E. El-Diraby, Domain ontology for construction knowledge, *Journal of Construction Engineering and Management* **139**(7) (2013), 768–784.
[29] L. Patil, D. Dutta, A.D. Bhatt, K. Jurrens, K. Lyons, M. Pratt, R.D. Sriram et al., Representation of heterogeneous objects in ISO 10303 (STEP), in: *ASME International Mechanical Engineering Congress and Exposition, Orlando, Florida*, 2000.
[30] S. Grimm, A. Abecker, J. Völker and R. Studer, Ontologies and the semantic web, in: *Handbook of semantic web technologies*, 2011.
[31] W. Terkaj and P. Pauwels, A Method to generate a Modular ifcOWL Ontology, in: *8th International Workshop on Formal Ontologies meet Industry*, Vol. 2050, 2017.
[32] I. UK, Infrastructure and P. Authority, Creating a Digital Built Britain: what you need to know (2017).
[33] J. Werbrouck, P. Pauwels, J. Beetz and L. van Berlo, Towards a decentralised common data environment using linked building data and the solid ecosystem, in: *Advances in ICT in Design, Construction and Management in Architecture, Engineering, Construction and Operations (AECO) : Proceedings of the 36th CIB W78 2019 Conference*, B. Kumar, F. Rahimian, D. Greenwood and T. Hartmann, eds, 2019, pp. 113–123. ISBN 9781861354877. http://cibw78.northumbria-eee.co.uk/.
[34] M.H. Rasmussen, M. Lefrançois, G.F. Schneider and P. Pauwels, BOT: the Building Topology Ontology of the W3C Linked Building Data Group, *Semantic Web Journal* **0**(0) (2019).
[35] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal et al., Brick: Towards a unified metadata schema for buildings, in: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, 2016, pp. 41–50.
[36] L. Daniele, F. den Hartog and J. Roes, Created in close interaction with the industry: the smart appliances reference (SAREF) ontology, in: *International Workshop Formal Ontologies Meet Industries*, Springer, 2015, pp. 100–112.
[37] S. An, P. Martinez, R. Ahmad and M. Al-Hussein, Ontology-based knowledge modeling for frame assemblies manufacturing, *Proceedings of the 36th International Symposium on Automation and Robotics in Construction, ISARC 2019* (2019), 709–715. doi:10.22260/isarc2019/0095.
[38] S. Staub-French, M. Fischer, J. Kunz and B. Paulson, An ontology for relating features with activities to calculate costs, *Journal of Computing in Civil Engineering* **17**(4) (2003), 243–254.
[39] M. Zhiliang and W. Zhenhua, Framework for automatic construction cost estimation based on BIM and ontology technology, in: *Proceedings of the CIB W*, Vol. 78, 2012, p. 2012.
[40] H. Abanda, J.H.M. Tah, M. Manjia, C. Pettang and F. Abanda, An ontology-driven house-building labour cost estimation in Cameroon, *Journal of Information Technology in Construction* **16**(March 2010) (2011), 617–634.
[41] N. Li, Q. Li, Y.-S. Liu, W. Lu and W. Wang, BIMSeek++: Retrieving BIM components using similarity measurement of attributes, *Computers in Industry* **116** (2020), 103186. doi:10.1016/J.COMPIND.2020.103186.
[42] S.K. Lee, K.R. Kim and J.H. Yu, BIM and ontology-based approach for building cost estimation, *Automation in Construction* **41** (2014), 96–105. doi:10.1016/j.autcon.2013.10.020.
[43] H. Liu, M. Lu and M. Al-Hussein, Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry, *Advanced Engineering Informatics* **30**(2) (2016), 190–207. doi:10.1016/j.aei.2016.03.001.
[44] N.F. Noy, D.L. McGuinness et al., Ontology development 101: A guide to creating your first ontology, Stanford knowledge systems laboratory technical report KSL-01-05 and . . . , 2001.
[45] P. Doran, V. Tamma and L. Iannone, Ontology module extraction for ontology reuse: an ontology engineering perspective, in: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 61–70.
[46] A. Algergawy, S. Babalou, F. Klan and B. König-Ries, Ontology Modularization with OAPT, *Journal on Data Semantics* **9**(2) (2020), 53–83.
[47] M.H. Rasmussen, M. Lefrançois, P. Pauwels, C.A. Hviid and J. Karlshøj, Managing interrelated project information in AEC Knowledge Graphs, *Automation in Construction* **108** (2019), 102956. doi:https://doi.org/10.1016/j.autcon.2019.102956. http://www.sciencedirect.com/science/article/pii/S0926580519300378.
[48] D. Vrandečić, Ontology evaluation, in: *Handbook on ontologies*, Springer, 2009, pp. 293–313.
[49] M. Nickel, K. Murphy, V. Tresp and E. Gabrilovich, A review of relational machine learning for knowledge graphs, *Proceedings of the IEEE* **104**(1) (2015), 11–33.
[50] M. Horridge and S. Bechhofer, The owl api: A java api for owl ontologies, *Semantic web* **2**(1) (2011), 11–21.
[51] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen and R. Klein, The IFC to linked building data converter: current

status, in: *6th Linked Data in Architecture and Construction Workshop*, Vol. 2159, 2018, pp. 34–43.

[52] E.V. Kalemi, F. Cheung, A.-r. Tawil and P. Patlakas, ifcOWL-DfMA a new ontology for the offsite construc- tion domain, *Proceedings of the 8th Linked Data in Architecture and Construction Workshop - LDAC2020 ifcOWL-DfMA* (2020), 105–117.

[53] M. Rasmussen, P. Pauwels, M. Lefrançois, G.F. Schneider, C. Hviid and J. Karlshøj, Recent changes in the building topology ontology, 2017.

[54] F. Barbosa, J. Woetzel, J. Mischke, M.J. Ribeirinho, M. Sridhar, M. Parsons, N. Bertram and S. Brown, Reinventing construction through a productivity revolution, *McKinsey Global Institute* (2017).

[55] I.C. No, Specification for information management for the capital/delivery phase of construction projects using building information modelling (2013).

[56] P. Pauwels, T. Krijnen, W. Terkaj and J. Beetz, Enhancing the ifcOWL ontology with an alternative representation for geometric data, *Automation in Construction* **80** (2017), 77–94.

[57] T. Steiner, R. Troncy and M. Hausenblas, How Google is using linked data today and vision for tomorrow, in: *LDFI 2010, Workshop on Linked Data in the Future Internet at the Future Internet Assembly, Volume 700, December 15-17, 2010, Ghent, Belgium, ISSN 1613-0073*, Ghent, BELGIUM, 2010. http://www.eurecom.fr/publication/3284.

[58] J. Ashraf, R. Cyganiak, S. O'Riain and M. Hadzic, Open ebusiness ontology usage: Investigating community implementation of goodrelations, *CEUR Workshop Proceedings* **813** (2011), 1–11.

[59] A. Borrmann, M. König, C. Koch and J. Beetz, *Building Information Modeling*, Springer International Publishing, 2018. doi:10.1007/978-3-319-92862-3.

[60] *ISO 10303-11 Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual*, International Standardisation Organisation, Geneva, Switzerland, 2004.

[61] C. Boje, A. Guerriero, S. Kubicki and Y. Rezgui, Towards a semantic Construction Digital Twin: Directions for future research, *Automation in Construction* **114** (2020), 103179.

[62] I. Yitmen, The challenge of change for innovation in construction: A North Cyprus perspective, *Building and environment* **42**(3) (2007), 1319–1328.

[63] A. Omigbodun, Value engineering and optimal building projects, *Journal of architectural engineering* **7**(2) (2001), 40–43.

[64] F.J. Emmatty and S. Sarmah, Modular product development through platform-based design and DFMA, *Journal of Engineering Design* **23**(9) (2012), 696–714.

[65] S. Ashley, Cutting costs and time with DFMA, *Mechanical Engineering* **117**(3) (1995), 74–77.

[66] T. Tan, W. Lu, G. Tan, F. Xue, K. Chen, J. Xu, J. Wang and S. Gao, Construction-Oriented Design for Manufacture and Assembly Guidelines, *Journal of Construction Engineering and Management* **146**(8) (2020), 04020085.

[67] S. Kamaruddin, M. Mohammad, R. Mahbub and R. Yunus, Perception towards cost implication of mechanisation and automation approach in IBS projects in Malaysia, in: *29th Annual ARCOM Conference*, 2013.

[68] Y. Jung and S. Woo, Flexible work breakdown structure for integrated cost and schedule control, *Journal of construction engineering and management* **130**(5) (2004), 616–625.

[69] S.-L. Fan, C.-H. Wu, C.-C. Hun et al., Integration of cost and schedule using BIM, *Journal of Applied Science and Engineering* **18**(3) (2015), 223–232.

[70] HM Government, 3-Digital Built Britain Level 3 Building Information Modelling - Strategic Plan, *UK Government* (2015), 1–47. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment{_}data/file/410096/bis-15-155-digital-built-britain-level-3-strategy.pdf.

[71] N.F. Noy, D.L. McGuinness et al., BCA, BIM design for design for manufacturing and assembly essential guide, Building and Construction Authority, 2016.

[72] W. Pan and R. Sidwell, Demystifying the cost barriers to offsite construction in the UK, *Construction Management and Economics* **29**(11) (2011), 1081–1099.

[73] K. Otto and K. Wood, Techniques in Reverse Engineering, Systematic Design and New Product Development, New York: Prentice Hall, 2001.

[74] G. Boothroyd, *Assembly automation and product design*, CRC Press, 2005.

[75] D.A. Steinhardt and K. Manley, Adoption of prefabricated housing–the role of country context, *Sustainable cities and society* **22** (2016), 126–135.

[76] R. Masood, J.B.P. Lim and V.A. González, Performance of the supply chains for New Zealand prefabricated house-building, *Sustainable Cities and Society* **64** (2021), 102537. doi:https://doi.org/10.1016/j.scs.2020.102537. http://www.sciencedirect.com/science/article/pii/S2210670720307538.

[77] A. Hajare and E. Elwakil, Integration of Life Cycle Cost Analysis and Energy Simulation for Building Energy-Efficient Strategies Assessment, *Sustainable Cities and Society* (2020), 102293.

[78] A. Darko, A.P. Chan, Y. Yang and M.O. Tetteh, Building information modeling (BIM)-based modular integrated construction risk management–Critical survey and future needs, *Computers in Industry* **123** (2020), 103327.

[79] T. Greif, N. Stein and C.M. Flath, Peeking into the void: Digital twins for construction site logistics, *Computers in Industry* **121** (2020), 103264.

[80] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, The NeOn methodology for ontology engineering, in: *Ontology engineering in a networked world*, Springer, 2012, pp. 9–34.

[81] K. Ayinla, E. Vakaj, F. Cheung and A.-R.H. Tawil, A Semantic Offsite Construction Digital Twin- Offsite Manufacturing Production Workflow (OPW) Ontology, 2021.