BPO: The Building Product Ontology for Assembled Products

Anna Wagner[1] and Uwe Rüppel[1]

[1] Technische Universität Darmstadt, Darmstadt, Germany
{wagner,rueppel}@iib.tu-darmstadt.de

Abstract. With the current trend of using Linked Data to describe buildings during their entire lifecycles, the importance of product descriptions in a Semantic Web context is growing while most product ontologies are designed for mass-produced goods of little variance. But especially in the construction industry, products are often innovative and individually manufactured and previous attempts to depict them with ontologies failed to include meaningful alignments to already existing approaches.
Thus, this paper gives an overview of existing product ontologies in general and analyses previous approaches in more detail to identify potential improvements that can be made. Based on this analysis, this paper presents the Building Product Ontology (BPO), including its concepts and alignments. To obtain a modular ontology, the BPO focuses on the non-geometric description without defining templates by determination of taxonomies and includes concepts to model assembly structures, interconnections of components, and complex properties and property values. The BPO enables manufacturers to freely model their products while still benefitting from the Semantic Web in respects of findability and availability of product data. By going through the given examples and demonstrations, inexperienced users are supported to apply the BPO and exploit its benefits.

Keywords: product ontology · linked data · product description

1 Introduction

The progression of digitisation in the architecture, engineering and construction domain (AEC) is premised on various approaches and different technologies. Recently, approaches using Semantic Web technologies and Linked Data have produced interesting results for digital building models, sensor data, collaboration, and other fields of interest. However, for the industry to recognise their potential, all aspects of the building lifecycle must be addressed by these approaches, including product descriptions.

Currently, several approaches to describe products in a Semantic Web context exist, but most of them were developed from other perspectives and domains and thus are not suitable for building products. To create means for describing innovative and multi-functional building products, the SolConPro ontology was introduced in [12]. The ontology tried to bridge the gap between generic and flexible parametric product descriptions, their geometric representations, and domain specific definitions for products and therefore constitutes an eligible candidate to be used for describing building products.

Nonetheless, the SolConPro ontology still exists as an island solution, since it is neither aligned nor linked to other – more prevalent – ontologies and approaches. It also does not follow basic principles of web data like re-using and adapting existing approaches and ontologies instead of defining already known concepts anew.

To increase the benefit users may have by using this ontology, we are presenting a revision of the former SolConPro ontology in this paper. The revision includes alignments to other ontologies, modularisation of the former SolConPro ontology, re-usage of concepts from other ontologies, and refinement of the defined concepts.

Next, we will discuss approaches for product-related ontologies, including a more detailed analysis of the SolConPro ontology, including an analysis on its shortcomings. Based on this discussion and analysis, we will present the new Building Product Ontology (BPO), which is build upon the SolConPro ontology. Following, a proof of concept of the BPO ontology is shown. Finally, we conclude this paper and give an outlook on future work.

2 Related Work

Within this section, we differentiate between three types of related ontologies: *Common product ontologies*, which are widely used, but focus on offering either very generic product descriptions or descriptions for store-based products in e-Marketplaces; *product ontologies from different domains*, i.e. from mechanical engineering, food industries, etc.; *domain ontologies* from the architecture, engineering and construction sector.

Each of these categories must be regarded within the ontology revision. However, the impact those categories should have on it vary. For example, an alignment towards common product ontologies is of high importance for manufacturers, so their products will be easier to find and be interpreted by search engines and thereby also by their potential customers. But at the same time, these ontologies may not be suitable to describe the product's structure or interconnections. On the other hand, product ontologies from different domains may already introduce concepts to describe assembled products or product interconnections, while their benefit from the perspective of search engines might not be as crucial. Meanwhile, domain ontologies are of high importance for the end-users of the product descriptions, as they want to easily include the products into their digital building models and the building lifecycle.

All considered ontologies of this paper are summarised in Tab. 1, together with their prefixes and domains.

2.1 Common Product Ontologies

The most popular common product ontology is the *GoodRelations* ontology. It was originally designed for online services of the non-parametric retail market with mass produced products [5]. The ontology offers concepts to describe products in different abstraction levels (from models to specific individuals that exist in the real world)

Table 1. Summary of reviewed and re-used ontologies.

Prefix	Name	Domain
bpo	Building Product Ontology	https://w3id.org/bpo
scp	SolConPro Ontology	
schema	Schema.org	https://schema.org/
gr	GoodRelations	http://purl.org/goodrelations/v1
lupo	Lightweight Upper Level Ontology	http://www.loa.istc.cnr.it/lupo
fpro	Feature-based Product Ontology	http://www.loa.istc.cnr.it/fpro
bot	Building Topology Ontology	https://w3id.org/bot
opm	Ontology for Property Management	https://w3id.org/opm
seas	Smart Energy Aware Systems Ontology	https://w3id.org/seas/
omg	Ontology for Managing Geometry	https://w3id.org/omg
fog	File Ontology for Geometry formats	https://w3id.org/fog
qudt	Quantities, Units, Dimensions and Types Catalog	http://qudt.org/schema/qudt

and unifies approaches for adding properties to products. Because of its wide support from search engines as Google, Yahoo!, etc., [1] claims that by describing products online using the GoodRelations ontology may cause an increase of traffic by up to 30 %. Nonetheless, since the ontology is intended for retail market and services, it does not provide means to describe complex, assembled, and parametric products, as it would be required for multi-functional building products.

The GoodRelations ontology is widely included into the *schema.org* ontology that aims to serve as a central vocabulary to help provide structured data on the web in a unified way. It was founded by Google, Microsoft, Yahoo and Yandex and is supported by web applications of different domains. The ontology consists of several parts, one of which is dedicated to describing products and services. Though the product part of schema.org is based on the GoodRelations ontology, both approaches differ in detail and abstraction levels at some points as well as their namespaces, with schema.org allowing more flexibility for describing not just products but also parts of products.

Another common ontology with wide range is the *Suggested Upper Merged Ontology* (SUMO)[4], which is merging domain ontologies from various fields. It is – in contrast to the other mentioned ontologies – written in the KIF language, but a translation into OWL is also available. Within SUMO, various concepts from a wide array of domains are collected, including more generic concepts like such for (manufactured) products and generic property or attribute properties. Nonetheless, since this ontology is not built in modular manner and its based on KIF with OWL just being a translation, its application for a flexible and modular product ontology in RDF seems not viable.

2.2 Product Ontologies from Different Domains

The topic of product description ontology has been discussed for many domains, e.g. manufacturing, mechanical engineering, and the food industry. The underlying approaches differ in multiple aspects, but the most distinct observed differentiation is

[4] http://www.adampease.org/OP/

the modelling of product (de)composition in form of geometric assemblies, e.g. with *Open Assembly Models* (OAM, [4]), or by using Bills of Material (BOM), as realised in *PRONTO* [11]. Since the BPO is closely related to geometry descriptions but does not contain them itself, it follows an approach in between OAM and BOM. Following, we discuss product ontologies with similar scopes.

In 2010, [2] presented an ontology-based modelling language for products with three abstraction layers. Since the proposed ontology could not be found online, its influence on the BPO ontology is merely conceptional. The core concepts build an abstract Tbox, which serves as a basis for product models that are created as subclasses of those concepts within the second layer. The third layer contains individuals of the product models, representing real-life product individuals. However, this way of modelling abstraction levels holds disadvantages if specific, real-life products are re-used in product models, as could be the case during restorations of products. Furthermore, concepts to define assembled artefacts were proposed and differentiated between assembly and containment relations to allow more refined definitions of the composition of parts. [2] also introduced concepts to define connections between parts, with the option to model the connection in two levels of details.

More recently, the *Feature-based Product Ontology* (FPRO) which is based on the *Lightweight Upper Level Ontology* (LUPO) were designed [10]. Within the LUPO, generic concepts that could also be used for other purposes than the description of products are defined. This also includes assembly structures and the differentiation between physical and non-physical items, which enforces geometry and material descriptions on every product model. In the FPRO, this differentiation is projected on product features, thus increasing the extent of this enforcement to each element of product descriptions. The FPRO additionally introduces a new assembly subproperty to enable to distinguish between assembly and containment relation, similar to the concepts described by [2]. As its name suggests, the FPRO is also defining feature-based concepts, e.g. Boolean operations for void features. Apart of these rather geometric aspects, it introduces concepts to describe the production process as well.

The discussed ontologies are not aligned towards other ontologies, i.e. schema.org and GoodRelations, and contain additional concepts as process management, geometry- and time-based constraints for the assembly process, and material descriptions.

2.3 AEC Domain Ontologies

Within the AEC domain, efforts have been made to describe buildings and products within a web context and should be considered during the design of the BPO.

Regarding product descriptions, definitions of terms for components and properties or property sets from the widely used Industry Foundation Classes (IFC) were translated to RDF (*PRODUCT*[5], *PROPS*[6]). The PRODUCT ontology additionally provides means to describe simple aggregations between products. However, it restrains these connections to products alone, therefore every part of a product must be a product itself, which does not hold true in all cases.

[5] https://github.com/w3c-lbd-cg/product

[6] https://github.com/w3c-lbd-cg/props

Meanwhile, the PROPS ontology can be created automatically based on the IFC standard. It also defines the property domains and ranges, where concepts of the *Building Topology Ontology* (BOT) are reused. BOT serves as a reference ontology that describes a building's topology [9].

PROPS is not aligned towards common vocabularies and does not allow to add further information or meta data to the properties. In order to overcome the lack of connection to meta data, the *Ontology for Property Management* (OPM) was introduced. As shown in [8], meta data like authoring information or even automatically created versions of properties can be added by creating intermediate nodes based on concepts of the OPM and parts of the *Smart Energy Aware Systems* Ontology (SEAS) [7]. The relevant part of SEAS is the part for Features of Interest. It contains concepts to describe features and their properties, including possible relations between them.

3 Analysis of the SolConPro Ontology

Before we present the new BPO, we will briefly summarise the SolConPro ontology [12] and then discuss its shortcomings. The SolConPro ontology was created to describe multi-functional facade components such as building integrated photovoltaic (BIPV) modules. It addresses multiple fields, as the structural description of complex and innovative building products, their interconnections and properties, as well as means to describe parametric dependencies between properties and geometric features. The SolConPro ontology was defined as being a core layer to a multi-layered product ontology, which would also contain vocabularies for classification and geometric descriptions. To link these layers, a property for the buildingSMART Data Dictionary (bSDD) ID and a mapping of classes with the GEOM ontology [3] were defined.

The SolConPro ontology itself introduces concepts for assembly relations, including the usage of so called entity nodes for typification of objects with individual values. Since a relevant part of a BIPV module's description are its electrical interconnections, the SolConPro ontology also contains classes and properties to describe generic interconnections between placed entities. Apart of the product structure, the ontology can also describe properties of varying complexity. The considered properties span from simple, one-value properties to ones with value ranges, step sizes or intervals; the most complex properties are based on unstructured, two-dimensional lists that relate the values of two different properties.

3.1 Identified Shortcomings

Missing Modularity. The SolConPro ontology, though still small in comparison to e.g. the ifcOWL, addresses multiple, non-related topics, i.e. general product description and parametric dependencies. Thus, re-using one specific part would require to also import the non-related functionalities and thereby cause unnecessary overhead. Instead, non-related topics should be separated and imported only where needed.

Also, the proposed methods to link the SolConPro ontology to other ontologies are not ideal in respect of modularity. By only linking to IDs instead of individuals, information is harder to evaluate. And creating direct mapping structures on Tbox level may be more effective for the considered ontology combination, however, the mapping must be adapted or re-designed for other combinations. This is not just cumbersome but can also impede unified querying.

Undesired Impacts of Inheritance. Inheritance relations may have undesired effects on the data, if not considered carefully. Within the SolConPro ontology, two of those cases appear. First, the `scp:Product` class inherits from the `scp:Assembly` class, which represents components that are assembled by other components. Yet, this relation – in combination of the disjointness of the `scp:Assembly` and `scp:Element` classes – causes a restriction upon products that they have to be assembled by components, which may not hold true.

Second, the SolConPro ontology differentiates between properties with exactly one value (`scp:Attribute`) and those with value ranges or multiple values. However, the latter (`scp:DynamicAttribute`) inherits from the other, which leads to inconsistencies: `scp:Attribute` individuals should have exactly one value, while `scp:DynamicAttribute` individuals may have up to one value, but inherit the restriction from `scp:Attribute`. This inconsistency was not detected by reasoners, since it does not cause logical contradictions.

Missing Expressivity. One can argue whether it is beneficial to apply highly expressive ontologies, since they create a valuable benefit for generating information on the one hand, but cause performance issues and may quickly become so complex that they are barely manageable on the other hand. The SolConPro ontology has an expressivity of $\mathcal{SRIQ}$ and can therefore be considered more expressive. Nonetheless, it is not using this level of expressivity to a full extent: Most sibling classes are not defined to be disjoint, restrictions are not defined as class axioms consistently and it contains one defective chain axiom for entity connections. The latter is supposed to create a chain for two entities that share the same `scp:ComponentConnection`. Yet, one of the relations is misdirected.

Ambiguity. Another shortcoming of the SolConPro Ontology are the chosen vocabulary names and a missing documentation. If an ontology is expected to be re-used by other parties, it must contain a detailed description of its concepts and, ideally, a public step-by-step instruction on how to use them.

Missing Alignment. One of the main arguments for describing products using Semantic Web technologies is the advantage during customer-searches created by the search engines. As discussed in Sec. 2.1, some ontologies like *schema.org* and *GoodRelations* are supported by common search engines. To gain this advantage, product ontologies must be aligned to those ontologies.

The advantages from search engines are not the only reason for aligning ontologies. [6] discuss the best practices for publishing Linked Data. Amongst others, they recommend to re-use and adapt already defined concepts from other ontologies as possible instead of creating new concepts. Also, ontologies from the same domain but with different scopes should be considered for alignment to create an unbroken understanding of concepts within one domain. This could even be widened to related domains.

However, the SolConPro ontology is not aligned to other ontologies. Thus, most benefits from using Semantic Web technologies for product descriptions are voided.

3.2 Roundup of the SolConPro Ontology

The SolConPro Ontology was designed based on the needs to describe highly innovative and multi-functional building products. It introduces concepts that allow users to describe their products without restrictions by templates, while still being able to define products in high detail. Nonetheless, the ontology is unlikely to create a noticeable impact on the industry, since it is not exploiting its full potential. Therefore, a revision of the ontology with the aim to solve the identified shortcomings may increase the effectiveness and thereby also applicability by the industry.

4 BPO: Building Product Ontology – A Revision of the SolConPro Ontology

Based on related approaches for describing building products in a Semantic Web context and the discussed shortcomings of the SolConPro ontology, we designed the Building Product Ontology (BPO) presented in this section.

With the overall concept of a multi-layered product ontology, the BPO is meant to serve to describe a product structures and properties only. Thus, the connections to the other layers – both in general and via parametric dependencies – are not a central topic for the BPO. Besides, the field of application of the SolConPro ontology is transferred to the BPO: highly innovative, multi-functional building products on the example of facade components. Subsequently, requirements for the SolConPro ontology – except for parametric dependencies – hold true for the BPO: it should follow a flexible, non-template-driven approach allowing multi-classification and can easily be extended [12].

As concepts and terms from the SolConPro ontology are the basis for the BPO, Tab. 2 compares the statistics of both ontologies. The BPO does not introduce concepts or properties for new topics, but enriches the existing terms, while at the same time reducing the overall expressivity in favour of performance and handling. It is visible that, though less classes and properties are part of the BPO, it still contains more triples than the SolConPro ontology. This highlights the advancements made in respect of exploiting the used expressivity. Concepts of the SolConPro ontology the BPO is not re-using, are either dropped because the scope of the ontology was reduced or replaced by suitable concepts of other already existing and popular ontologies (e.g. schema.org).

Table 2. Comparison of Statistics.

	BPO	**SolConPro**
Triples	376	313
Classes	14	19
Class Axioms	60	53
Datatype Properties	7	12
Object Properties	18	27
Chain Properties	4	5
Transitive Properties	2	3
DL Expressivity	$\mathcal{SRIN}(\mathcal{D})$	$\mathcal{SRIQ}(\mathcal{D})$

Next, we will discuss examples of the changes between the SolConPro ontology and the BPO. These changes can be grouped into the categories *Modularity*, *Inheritance*, and *Expressivity*. After the adaption on schema-level is shown, we will continue to present its alignment to relevant ontologies presented in Sec. 2.

4.1 Schematic Changes

Apart of more tangible changes, the naming was re-evaluated, unified in its convention, and adapted to be more unambiguous, where needed. The aimed non-ambiguity of the ontology is also the reason for renaming it, since the name *Building Product Ontology* is defining the ontology's scope more clearly than the name *SolConPro Ontology*. Additionally, the BPO – with its online documentation – are published online.

Modularity. The scope of the BPO is reduced to describing the product structure itself. This includes defining assembled components with optional entity nodes, entity interconnections, component and product properties, and complex attributes. Subsequently, all concepts except those for parametric dependencies are re-used.

Moreover, connecting geometry descriptions is no longer realised by mapping classes. Instead, we recommend to use the *Ontology for Managing Geometry* (OMG) resp. *File Ontology for Geometry formats* (FOG) to create flexible and uniform links. Since the authors are unaware of a vocabulary for building products and properties in a Semantic Web context, the bSDD is still used within the BPO and connected via datatype properties that have the classification superclass as domain.

Inheritance. The discussed problems with undesired impacts of inheritance have been resolved by adapting the class hierarchy (see Fig. 1): For instance, `bpo:Product` is now a direct subclass of `bpo:Component`, while it is the only class that is not disjoint with all sibling classes. Thus, users can define any Assembly or Element to also be a product. Furthermore, the `bpo:Attribute` and `bpo:RangedAttribute` are modelled as sibling classes, removing the inconsistency regarding value cardinalities.

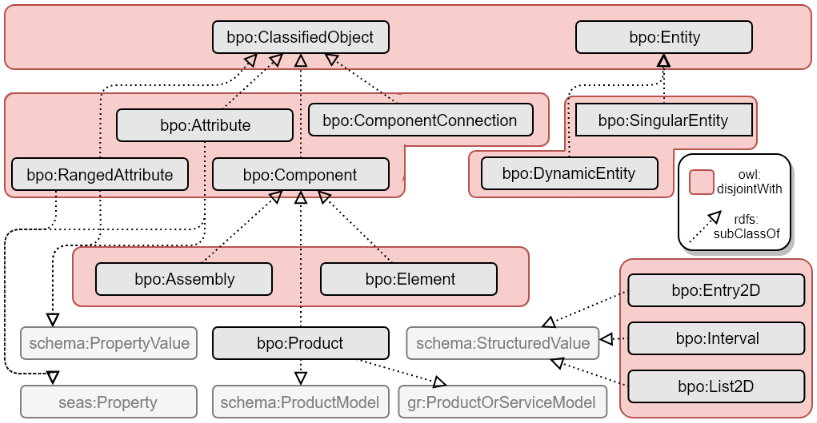

Fig. 1. Overview of the BPO Class Hierarchy

Expressivity. As is shown in Fig. 1, the BPO defines disjointness (red boxes) between all sibling classes with the exception of the `bpo:Product` class. Besides disjointness, additional class and chain axioms are part of the BPO:nine class axioms and four chain axioms. In comparison, the SolConPro ontology includes seven class axioms and four chain axioms for the concepts that were taken on to the BPO. All axioms from the SolConPro ontology were re-used, wherefore one class axiom was supplemented and two others were aligned towards other ontologies while the faulty chain axiom was rectified.

The rectified chain axiom belongs to the `bpo:isConnectedTo` property (see Fig. 2) and is defined as (1). To fix the error, the former `scp:hasIncomingConnection` property is inverted and renamed as `bpo:connectsInputOf`. Apart of the rectified chain axiom, the symmetric `bpo:isConnectedWith` property between two `bpo:Entity` individuals is introduced to for non-directed connections.

$$bpo\!:\!hasOutgoingConnection \circ bpo\!:\!connectsInputOf \tag{1}$$

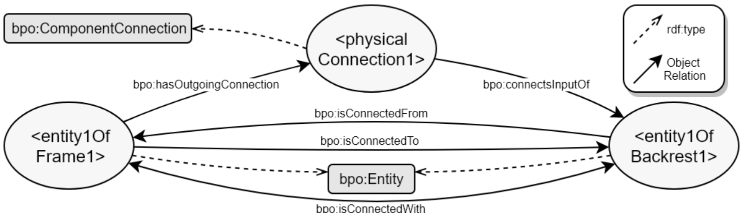

Fig. 2. Modelling Component Connections

The supplemented class axiom belongs to the renamed `bpo:RangedAttribute` class (based on `scp:DynamicAttribute`) and was adapted to restrict every instance of it to have at most one specific, minimum and maximum value as well as one unit and permitted step size. Previously, the `scp:DynamicAttribute` was restricted to have exactly one specific, minimum and maximum value and at most one unit and step size.

Furthermore, two new class axioms are introduced, both concerning components. The first defines `bpo:Assembly` to consists of at least one component and at one of the connected components has to be of the type `bpo:Element`. Meanwhile, the second axiom determines `bpo:Element` not to use the `bpo:consistsOf` property.

These axioms were added to emphasise and specify the assembly descriptions of BPO (see Fig. 3): Assemblies can consist of at least one other component, while elements cannot – or will not – be further disaggregated. Since the BPO is disconnected from geometry and material definitions, it is possible for manufacturers to describe components as elements, even though they are not made from one material or part. This can be used to keep more detailed product information that might contain corporate secrets like used materials or techniques from being spread.

However, by allowing the definition of `bpo:DynamicEntity` nodes without requiring a geometry description as well, information as the quantity of placed entities

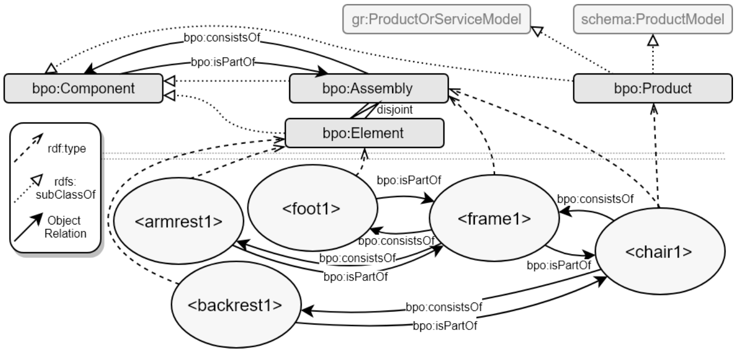

Fig. 3. Modelling Assembled Components and Products

get lost. Therefore, we introduced the new datatype property `bpo:hasQuantity` with the domain of `bpo:DynamicEntity` and range of an `xsd:integer`.

4.2 Alignment to Other Ontologies

The adaption of the ontology does not only concern its structure, but also its alignment towards other ontologies. Figures 1 and 3 already give insight about re-used classes. The complete alignment will be presented in this section and is separated into three categories: *Assembly structure*, *property* and *product*.

Assembly Structure. The description of assembled components is discussed by several ontologies, the most relevant for the BPO are schema.org, GoodRelations, LUPO, and FPRO. Schema and GoodRelations mainly focus on properties between products to define spare parts or consumables. Both relation types are not suitable for the BPO and would also impose the restriction that every component would be a product.

The more product – and assembly – focused LUPO and FPRO, on the other hand, introduce a differentiation between part-whole and assembly relations of components (`lupo:hasPropertPart` and its subproperty `fpro:hasComponent`). This allows to give a more detailed, part-based description even for components that are not technically assembled (e.g. screws). However, at the same time, these ontologies restrict components and products to either be physical or not and thereby enforce geometry and material descriptions on every part of a product. Another restriction on assembled structures is that they must consist of at least two products of single materials. Again, this implies every component to be a product. Thus, from the perspective of assembly structures, no alignment is proposed.

Property. Multiple alignment options for property descriptions exist: From generic ontologies as schema.org or GoodRelations, other product ontologies and already implemented approaches of the building domain, i.e. OPM and SEAS. A challenge in finding optimal alignments is the requirement for the BPO to still enable flexible modelling.

Within GoodRelations, property values are not linked directly to objects, but defined with an intermediate node between products and its property values. Also, a separation between quantitative and qualitative properties is defined both on class and property level. Quantitative ones may use properties to define minimal, maximal, and singular values, as well as units. On the other hand, qualitative values consist of descriptions and relations (e.g. lesser, equal, etc.) between each other. The vocabulary also predefines common product properties as width, height, etc. Nonetheless, the separation of qualitative and quantitative properties complicates unified querying if the product architecture is not known beforehand, and since all properties are defined to belong to products, this would imply every component with a property would be a product.

Since schema.org re-uses most concepts of the GoodRelations vocabulary, the concepts presented above are part of schema.org as well. Furthermore, schema.org introduces concepts to add `schema:PropertyValue` nodes to places, products, and other properties for the description of neutral (either qualitative of quantitative) properties via the `schema:additionalProperty` property. It can additionally be enriched by a property identifier, which could be used to add the bSDD ID of the property. Both schema.org and GoodRelations are using unit codes to define the used units instead of linking to ontology-based vocabularies of units (e.g. the QUDT).

In the building domain, the OPM was proposed to describe properties on different levels of detail. The approach for property modelling that was chosen in the BPO compares to the Level 2 of [8]. Thus, the BPO attributes could be seen as subclasses of the `opm:Property`. As the OPM is based on concepts of the SEAS ontology, the BPO could also be aligned towards the `seas:Property` class as well as the `seas:hasProperty` property to relate `seas:FeatureOfInterest` and property individuals. Because of the generic nature of features of interest, a classification of components as those would not restrict the freedom of modelling. However, SEAS does not provide datatype properties for value definition.

The overall alignment of the BPO attributes can be seen in Fig. 4. From the aspect of attributes, both `bpo:Attribute` and `bpo:RangedAttribute` are subclasses of `schema:PropertyValue` and `seas:Property`. The `bpo:hasAttribute` property is a subproperty of `schema:additionalProperty` and `seas:hasProperty`. Instead of using units that are defined within the BPO, units should be linked from the QUDT using `qudt:Unit` individuals and `qudt:unit` properties. Values can be described using `schema:minValue`, `schema:maxValue`, and `schema:value`. Only `bpo:permittedStepSize` is not substituted by properties from existing ontologies.

For complex attribute values, `schema:value` is re-used for interval values and all classes (`bpo:List2d`, `bpo:Entry2D`, and `bpo:Interval` are defined as subclasses of `schema:StructuredValue`. Thus, we refrain from re-using `scp:AttributeValue` and define the `bpo:hasList2D`, `bpo:hasEntry`, and `bpo:hasInterval` properties as subproperties of `schema:value`, as well. Furthermore, the SolConPro ontology enabled the description of unstructured, two-dimensional lists via Well-Known-Text (WKT), wherefore a new property was introduced. In BPO, we suggest to instead use the terminology as it was proposed in GeoSPARQL (`geo:asWKT`).

Product. Product alignment should focus on generic ontologies as GoodRelations and schema.org. Suitable concepts for alignment are the `gr:ProductServiceOrModel` and its schema.org counterpart `schema:ProductModel`. Based on the requirement of free modelling, it is important to note that these classes should be aligned only to the `bpo:Product` class and not components in general. Thereby, it is not feasi-

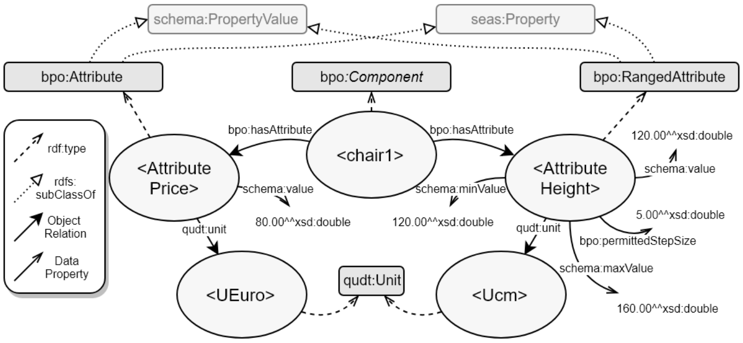

Fig. 4. Modelling Assembled Components and Products

ble to create broader alignments to GoodRelations. Instead, we propose to model properties of the product itself also in alignment to GoodRelations. Since property modelling for products should not differ to that for components, this cannot be realised fully-automatic by creating subclasses and subproperties.

Alignment to Domain Ontologies. Finally, the BPO should be aligned to domain ontologies as BOT and taxonomies like the bSDD. Considering theoretical applications of the BPO outside of the building domain, we do not recommend to create this alignment on Tbox level. Alternatively, concepts of GoodRelations or schema.org could be used to realise connections between a `bot:Element` and a `bpo:Product`. Promising properties are `gr:hasMakeAndModel` and its schema.org counterpart `schema:model` that define the type of an `gr:Individual` or `schema:IndividualProduct`, each describing a specific, real-life instance of one product type.

Moreover, the alignment to the bSDD is aligned to the `schema:additionalType` property by establishing the `bpo:hasBSDDGUID` property as its subproperty.

5 Proof of Concept

To demonstrate the functionalities of the BPO, a *SPARQL-visualizer* demo was created and published online[7]. The visualiser consists of several tabs that are grouped into seven categories: Modelling and classifying products, direct assembly structures, indirect assembly structures using entities, modelling connections between components, adding properties to components, complex property values (Intervals), and complex property values (unstructured, two-dimensional lists).

The demo guides inexperienced users in applying the BPO to describe their own products, beginning with the simple definition of products. It also demonstrates how assembly structures can be described on two levels: directly or via `bpo:Entity` nodes and how a reasoner can create the direct connection from the indirect one (see Fig. 5, blue: inferred triple).

[7] https://madsholten.github.io/sparql-visualizer/?file=https:%2F%2Fwww.dropbox.com%2Fs%2F33ah5crs4a0a25c%2Fbpo-demo.json

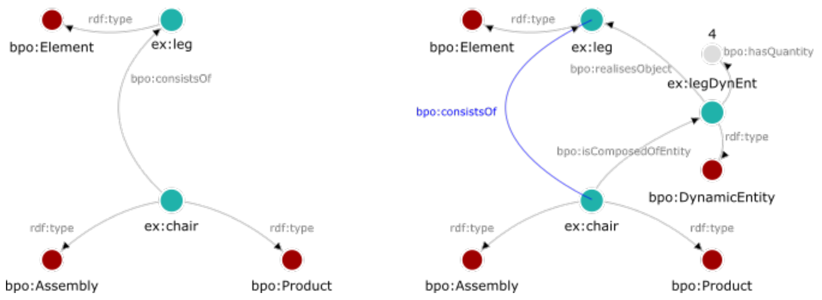

Fig. 5. Example of modelling assembly structures directly or using intermediate nodes.

Furthermore, the demo demonstrates how connections between entities can be modelled and how properties can be modelled. The latter also includes examples on how complex property values as intervals and two-dimensional unstructured lists can be handled using the BPO.

6 Conclusion and Outlook

In this paper, we introduced the Building Product Ontology (BPO) which was created after a thorough analysis of the SolConPro ontology and common (product) ontologies. The BPO re-uses suitable and proven concepts from the SolConPro ontology and enhances them with new properties and axioms. It also is closely aligned towards existing (product) ontologies as schema.org, GoodRelations, and SEAS, while an alignment for domain ontologies is proposed on Abox level. With the presented SPARQL-visualizer demo, the BPO's concept is proven and at the same time, users and other researchers are supported in applying the BPO for their own needs.

However, the application of the BPO for describing real products must be extended further. Since the BPO is based on the SolConPro ontology, building products that were modelled using the SolConPro ontology can also be described by the BPO, but these products are limited to multi-functional facade components. Thus, in future work, the utilisation of the BPO for building products from a more general perspective is of high importance. Also, the connection of BPO product descriptions to the products' geometry descriptions via the OMG and FOG ontologies needs to be evaluated.

As the scope of BPO is more precise than the one of the original SolConPro ontology, some concepts of the latter are not re-used in BPO. With the goal of creating modular ontologies that can be adapted and re-used in as many scenarios as possible, the left-over concepts - especially those dealing with parametric dependencies - should be revisited and used as a basis for a modular and more universal ontology with a scope focused on parametric dependencies or equation only.

7 Acknowledgements

This work is part of the research project EnOB: SCOPE, founded by the German Federal Ministry for Economic Affairs and Energy (BMWi). The concepts of the ontology were developed in collaboration with Fraunhofer Institute for Solar Energy Systems ISE and Ed. Züblin AG and implemented by the authors of this paper.

References

1. Ashraf, J., Cyganiak, R., O'Riain, S., Hadzic, M.: Open ebusiness ontology usage: Investigating community implementation of goodrelations. In: LDOW. vol. 813 (2011). https://doi.org/10.1016/j.ijhydene.2012.08.140
2. Bock, C., Zha, X., Suh, H.w., Lee, J.H.: Ontological product modeling for collaborative design. Advanced Engineering Informatics **24**(4), 510–524 (nov 2010). https://doi.org/10.1016/j.aei.2010.06.011, `https://linkinghub.elsevier.com/retrieve/pii/S1474034610000558`
3. Bonsma, P., Bonsma, I., Zayakova, T., van Delft, A., Sebastian, R., Böhms, M.: Open-standard-CMO-for-parametric-modelling-based-on-semantic-web.pdf. In: eWork Ebus. Archit. Eng. Constr. pp. 923–928 (2014)
4. Fenves, S.J., Foufou, S., Bock, C., Sriram, R.D.: CPM2: A Core Model for Product Data. Journal of Computing and Information Science in Engineering **8**(1), 014501 (2008). https://doi.org/10.1115/1.2830842, `http://computingengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1401049`
5. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) Lecture Notes in Computer Science. vol. 5268, pp. 329–346. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87696-0$_2$9, `http://link.springer.com/10.1007/978-3-540-87696-0{_}29`
6. Hyland, B., Atemezing, G., Villazón-Terrazas, B.: Best Practices for Publishing Linked Data (2014), `https://www.w3.org/TR/ld-bp/`
7. Lefrançois, M.: Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns. In: Proc. Work. Semant. Interoperability Stand. IoT, SIS-IoT. p. 11p. Amsterdam, Netherlands (2017)
8. Rasmussen, M.H., Lefrançois, M., Bonduel, M., Hviid, C.A., Karlshø, J.: OPM: An ontology for describing properties that evolve over time. CEUR Workshop Proc. **2159**, 23–33 (2018)
9. Rasmussen, M.H., Pauwels, P., Hviid, C.A., Karlshøj, J.: Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. In: Lean Comput. Constr. Congr. - Vol. 1 Proc. Jt. Conf. Comput. Constr. pp. 237–244. No. July, Heriot-Watt University, Edinburgh (jul 2017). https://doi.org/10.24928/JC3-2017/0153, `http://itc.scix.net/cgi-bin/works/Show?{_}id=lc3-2017-153`
10. Sanfilippo, E.M.: Feature-based product modelling: an ontological approach. International Journal of Computer Integrated Manufacturing **31**(11), 1097–1110 (nov 2018). https://doi.org/10.1080/0951192X.2018.1497814, `https://www.tandfonline.com/doi/full/10.1080/0951192X.2018.1497814`
11. Vegetti, M., Leone, H., Henning, G.: PRONTO: An ontology for comprehensive and consistent representation of product information. Engineering Applications of Artificial Intelligence **24**(8), 1305–1327 (dec 2011). https://doi.org/10.1016/j.engappai.2011.02.014, `https://linkinghub.elsevier.com/retrieve/pii/S0952197611000388`
12. Wagner, A., Möller, L.K., Leifgen, C., Rüppel, U.: SolConPro: Describing multi-functional building products using semantic web technologies. In: Karlshøj, J., Scherer, R.J. (eds.) EWork and eBusiness in architecture, engineering and construction: proceedings of the 12th European Conference on Product and Process Modelling (ECPPM 2018). pp. 447–455. 12, CRC Press (2018)