Attendees

- Pieter Pauwels [Ghent University]
- Jun Wang [Curtin University]
- Claudio Mirarchi [POLIMI]
- Walter Terkaj [ITIA-CNR]
- Thomas Krijnen [TUe]
- Mads Holten Rasmussen [DTU]
- Aaron Costin [University of Florida]
- Kyriakos Katsigarakis [TUCrete]
- Georg Ferdinand Schneider [Fraunhofer IBP]
- Georgios Lilis [TUCrete]
- Gonçal Costa [LaSalle University]
- Maxime Lefrançois [École des Mines de Saint-Étienne]
- Ana Roxin [uB]

Excused

- Saeed Karshenas [Marquette University]
- Vladimir Vukovic [Teesside University]
- Victor Malvar [neanex]
- Emilio Sanfilippo [LOA]

Date and time

- 07/09/2017
- 17:00 CEST

Agenda

1. Review of open issues in Git repos and Working Group documents
2. Product data and property handling
3. Geometry on the web

Minutes

1. Review of open issues in Git repos and Working Group documents (tentative list below)
A number of issues, requests, questions and remarks are popping up in the diverse subgroup. The proposal is to gather these within the different GIT repositories that are available in https://github.com/w3c-lbd-cg/ (issues, pull requests, commits, branches). A procedure on how to set up these GIT repositories in your own environment, is made available in the homepage of each repo (e.g. https://github.com/w3c-lbd-cg/bot#version-control). Another description for working with GIT is available at https://www.w3.org/2015/spatial/wiki/How_to_work_with_GitHub_for_ssn.

Furthermore, comments can be made directly in the Google Doc Working Group documents that are listed in our home page: https://w3c-lbd-cg.github.io/lbd/.

The upcoming meetings will go through the open issues and topics and decide on them. For this meeting, the following issues were tracked:

1. bot currently contains an alignment to saref4bldg directly in the ontology. Good practice would be to externalize the alignment in a separate ontology document, that imports the two ontologies that are aligned and defines alignment axioms. This is proposed by Georg in https://github.com/w3c-lbd-cg/bot/issues/4
   => pull request: https://github.com/w3c-lbd-cg/bot/pull/5 => accepted
2. Other alignments between bot and external ontologies may be proposed later (similar to 1. Above - e.g. GoodRelations, SAREF4BLDG, ifcOWL). In the future, such alignments are recommended to be published in a separated ontology document. An example may be found in SAREF4BuildingAlignment.ttl. The suggested namespace is always https://w3id.org/<ontologySource>/<ontologyRange>Alignment#.
3. When other alignments between two other external ontologies may be proposed later, is this repository the right place to store them? Yes, the alignment ontology is ideally stored along with the domain ontology (e.g. with the BOT ontology in the case of the saref4bldg alignment.
4. issue https://github.com/w3c-lbd-cg/bot/issues/3 discusses the alignment to saref4bldg specifically, in particular for the bot:hasSpace property, which is declared to be a subProperty of the saref4bldg:hasSpace property, although domains don't seem to match. => the issue remains open! See issue comments and please contribute. The two most straightforward options:
   a. Remove the hasSpace alignment
   b. Relax the domain of the bot:hasSpace property so that the domain mismatch with saref4bldg does not occur.

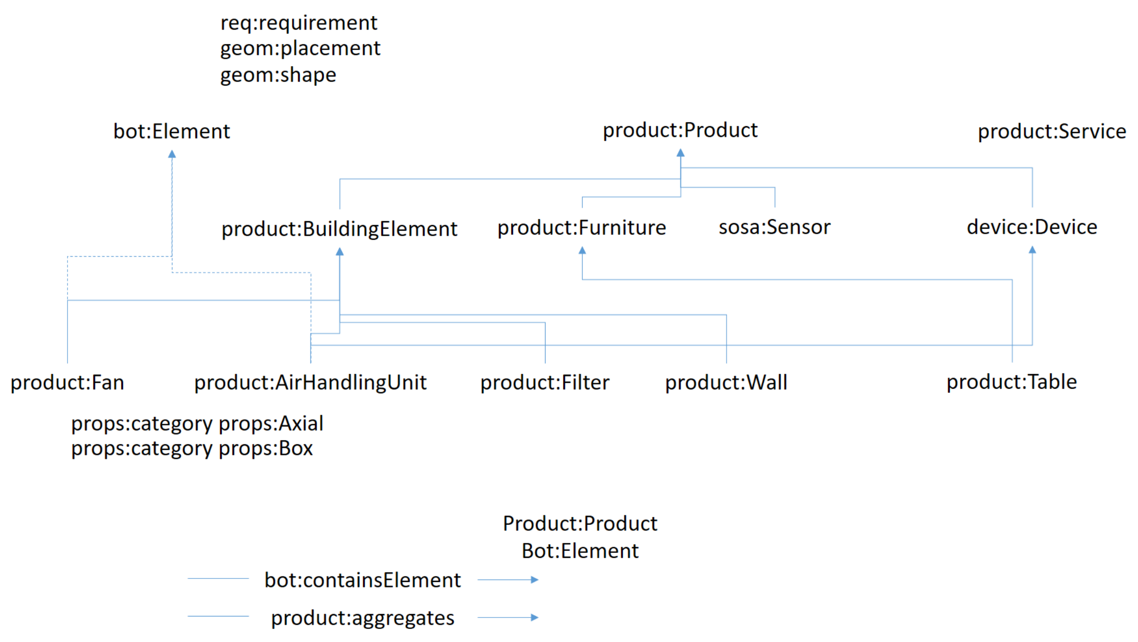
   => to be discussed and decided later
5. Some BOT examples are still present in the LBD repo. Shouldn't we move them from ldb repo to bot repo? => Yes; yet, the main idea is to keep examples listed in the home pages of the repositories.

2. Product data

Work has been done in the Product subgroup. This work focused on the main ontology hierarchy. How to handle the correlations between product, sensor, device and bot ontologies? What is the main scope of the ontology and how to structure it accordingly. This is an issue that popped up in discussing the prod:Product working group document:

https://docs.google.com/document/d/1ArLPgMC4js6i0tGFcYyWHIvlYGNq2Tou-Ed_tPHVo7A/edit

The suggestion is to aim at the following diagram:



This includes the following main definitions:

```
product:Product
	a owl:Class ;
	rdfs:comment "A product is an article or substance that is manufactured or
refined for sale"@en .

bot:Element a owl:Class ;
	rdfs:label    "Building element"@en ;
	rdfs:comment   "Constituent of a construction entity with a characteristic
technical function, form or position [12006-2, 3.4.7]"@en .
```

These two are main classes that are not placed in a hierarchy, but rather kept separate. The bot:Element class is mainly part of the building topology and is not specifically aimed to be very specific (e.g. decomposition, aggregation, relationships, specific attributes, …). The product:Product class on the other hand aims at capturing building products, with a top-level class to capture a generic product, also outside the building industry. This ontology aims primarily at the products and its decompositions and structures and properties, also as they are available on the manufacturer side. The BOT ontology is therefore primarily linked to geometric, requirement, HVAC, flow systems, and placement ontologies (describing the building structure); the PRODUCT ontology is mainly linked towards property ontologies and e-commerce ontologies.

Maxim: we could also consider product:Product and bot:Element to be “roles”, instead of naming them that specifically. This way, an element can switch role as soon as it moves from outside of the building into the building.
Pieter: Yes, this is indeed possible. But for now, we could also just keep it as such and assume that such roles will take form anyhow in using the actual ontologies in practical applications.

Mads: There are multiple options to describe such structures and manage it. Perhaps we should consider to maintaining multiple ontologies, and make sure that data can be mapped between the diverse use cases.
Pieter: Indeed. Also for managing product properties, there are multiple ways to model this. Depending on the use case, one might opt for an ontology that strongly support versioning, making the ontology often more complex, yet making data better manageable over time. In other use cases, one might opt to keep the ontology really simple for express data exchanges. If we have both setups, and we are able to relate them, end users can pick their choice. Yet, for now, let's first stick on the simpler version, to not end up in long discussion strings into an overly complex ontology as a basis.

Georg: about the definitions for the bot:Element and product:Product, we should consider looking at the GoodRelations definition at http://wiki.goodrelations-vocabulary.org/Documentation/Product_or_Service. GoodRelations distinguishes between three kinds of products:

1. *A real product like my laptop, my car with this VIN and mileage, a particular item on an eBay auction -* ***gr:Individual****.*
2. *A product model, i.e. a datasheet, like "Nikon T90", "iPod Nano 16 GB", or similar. -* ***gr:ProductOrServiceModel****.*
3. *Then we have a third case, in which the entities exposed on the Web are neither products nor product models, but instead "black boxes" of products. -* ***gr:SomeItems****.*

These three subtypes are powerful, but it is of course also allowed to use the common abstraction gr:ProductOrService.

Pieter: In that case, we should probably opt to align bot:Element and product:Product to a number of these subClasses. Likely, bot:Element should then align with gr:Individual, while product:Product should align with **gr:ProductOrServiceModel**. => To be checked with Prof. Martin Hepp [call 5th October]

Pieter: Note that we do aim to set up some sort of hierarchy underneath product:Product, which can serve as an indicative taxonomy. Let us not be too strict in hierarchization though.

Maxime: There are still a number of open questions to be raised:
1. Maxime: Can we also relate to the sosa:Actuator? Pieter: Yes, of course, the names in the above schema are just example indications.
2. Maxime: Should we not try to borrow concepts from IFC (Product, ConstuctionProduct, Fan, ElectricFlowStorageDevice, MedicalDevice, ProtectiveDevice, UnitaryEquipment, etc.)? Pieter: Yes, but we should take care not to include the entire IFC schema. The way in which we are currently modelling the diverse ontologies (PRODUCT, BOT, GEOM, PROPS, ...) is fairly modular, even keeping alignments separate so that end users and implementers are free to opt in or out of particular parts of the schema. This is a very strong principle, which is absent in IFC. We should take care to keep that principle. But the idea is indeed to have inspiration *and* alignment with the IFC schema.
3. Maxime: Some names are similar, but can mean very different things. E.g. IFCFilter is identical to product:Filter, but IFCTable (spreadsheet!) is not the same as product:Table. We should take care

of not mis aligning. Pieter: That is correct! We should take care, but we should also assume that people do read the descriptions of classes.

Maxim: What is the total coverage of our PRODUCT ontology? We cannot model it all.

Pieter: Correct. Perhaps we can limit the ontology to the most commonly used concepts, recommending specific companies with specific concept to extend the ontology within their own environments?

Mads: Yes, well, but it would perhaps be a good idea to support such companies, so that there would be a website where people can claim concepts to be added in product ontology, the interface files the request automatically as an issue in the github repository, and so forth.

Maxim: Can we also integrate dogont, M3-lite, and so forth? => yes, absolutely. In addition to finding inspiration there, it would be good to set out separate alignment ontologies.

Mads: a number of properties are now available in the PRODUCT ontology to describe decomposition and simple element topology relations:

```
product:aggregates
        a owl:ObjectProperty ;
        rdfs:domain product:Product ;
        rdfs:range product:Product .
```

And in the BOT ontology:

```
bot:adjacentElement
        a owl:ObjectProperty ;
        rdfs:domain bot:Space ;
        rdfs:range bot:Element .
bot:containsElement
        a owl:ObjectProperty ;
        rdfs:domain bot:Space ;
        rdfs:range bot:Element .
bot:hostsElement
        a owl:ObjectProperty ;
        rdfs:domain bot:Element ;
        rdfs:range bot:Element .
bot:aggregates a owl:ObjectProperty .
```

The hosts object property captures for example a window in a wall.

Maxime: can't it also capture, for example, a Space as an aggregation of other spaces? In that case, we might not need domains and ranges there.

Pieter: Well, but that makes this a very generic property as well, on the other hand. So, not really in favour.

Walter: Similarly, does it make sense to have an aggregation relationship in a topology ontology? Probably not. Isn't it the idea to keep such things in the PRODUCT ontology only? => yes!

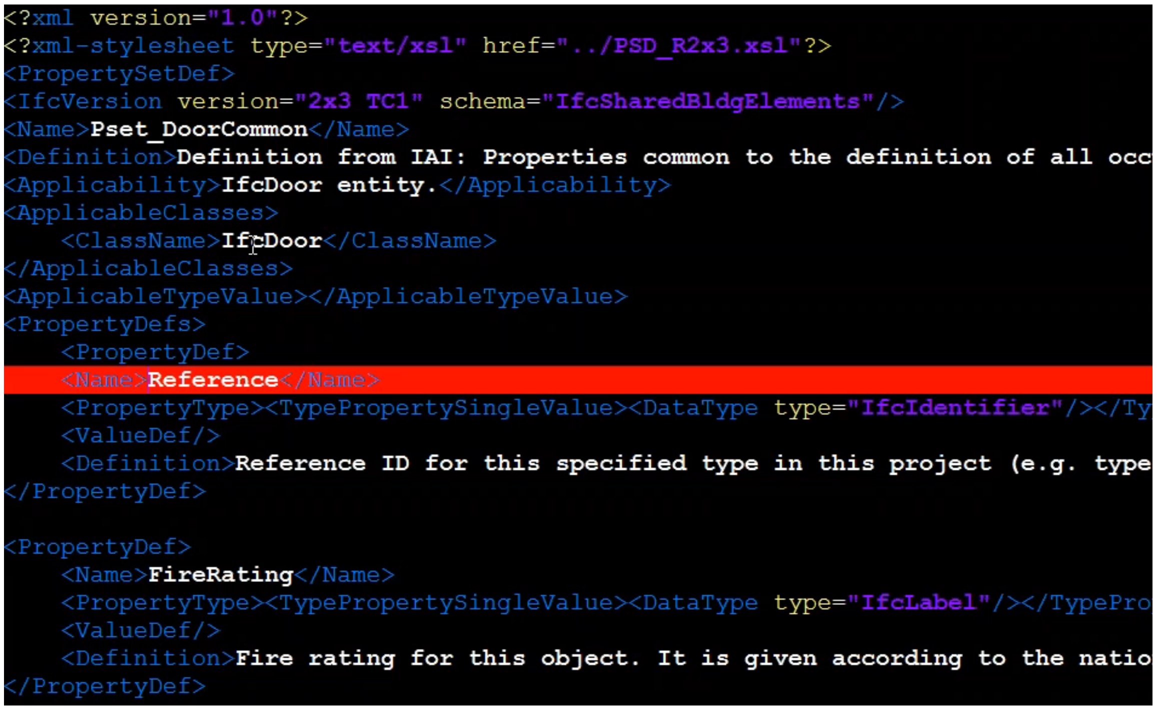
Pieter: also here, we might want to 'relax' the domain restrictions (bot:Space) on the adjacentElement and containsElement properties, and not include the bot:Space domain restrictions.

Mads: Also a bot:Site class has now been included in BOT, including corresponding hasBuilding object property relating to the bot:Building class.

Maxime: What about Groups, can we support Grouping of elements in the BOT ontology?

3. Properties and property handling

A large number of propertysets (PSETs) has been defined within buildingSMART. These are captured in a large number of XML files (400 or so per IFC schema specification). For IFC2X3, these PSET XMLS are captured as indicated in the below image (Reference and FireRating properties made available for 'IfcDoor' as a domain).



```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="../PSD_R2x3.xsl"?>
<PropertySetDef>
<IfcVersion version="2x3 TC1" schema="IfcSharedBldgElements"/>
<Name>Pset_DoorCommon</Name>
<Definition>Definition from IAI: Properties common to the definition of all occ
<Applicability>IfcDoor entity.</Applicability>
<ApplicableClasses>
    <ClassName>IfcDoor</ClassName>
</ApplicableClasses>
<ApplicableTypeValue></ApplicableTypeValue>
<PropertyDefs>
    <PropertyDef>
    <Name>Reference</Name>
    <PropertyType><TypePropertySingleValue><DataType type="IfcIdentifier"/></Ty
    <ValueDef/>
    <Definition>Reference ID for this specified type in this project (e.g. type
</PropertyDef>

<PropertyDef>
    <Name>FireRating</Name>
    <PropertyType><TypePropertySingleValue><DataType type="IfcLabel"/></TypePro
    <ValueDef/>
    <Definition>Fire rating for this object. It is given according to the natio
</PropertyDef>
```

The main idea is to capture this data and make this available as a props ontology. There are some difficulties though, as names occur in multiple files with different domains. To make properties unique, we could make 400 separate ontologies (which would be unusable, really), or look for a unique naming convention. Also, not all data should be transposed, as a considerable number of data relates to typing and relations, which is already captured differently in the available set of W3C LBD ontologies. If we go through the list of PSET properties in order to select a subset, we can just as well structure the domains and property names so that property names are unique, or follow a decent hierarchical inheritance.

4. Geometry on the web
Not handled in the current meeting

Previous minutes

https://docs.google.com/document/d/11ltaT0_0ajG10BXBTKaTY57_ppuXmBk2v0GCFicqArc/edit

Next Calls

Monday 25 September, 5-6PM CEST

Thursday 5 October, 5-6PM CEST - Prof. Martin Hepp (GoodRelations ontology)

Monday 16 October, 5-6PM CEST - Markus Rickert and Alexander Perzylo (Fortiss BREP ontology - https://github.com/OntoBREP/ontobrep)

Monday 30 October, 5-6PM CEST

Monday 6 November - Friday 10 November: TPAC meeting