www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

SEMANTIC PRODUCT MODELLING AND CONFIGURATION: CHALLENGES AND OPPORTUNITIES

PUBLISHED: August 2009 at http://www.itcon.org/2009/33

EDITORS: Kazi A S, Aouad G, Baldwin A

Michel Böhms, Dr.,
Netherlands Organisation for Applied Scientific Research (TNO), Delft, The Netherlands;
michel.bohms@tno.nl, http://www.tno.nl/

Peter Bonsma, Ir.,
Netherlands Organisation for Applied Scientific Research (TNO), Delft, The Netherlands
peter.bonsma@tno.nl, http:// www.tno.nl/

Marc Bourdeau, Dr.,
Centre Scientifique et Technique du Bâtiment (CSTB), Sophia-Antipolis, France
marc.bourdeau@cstb.fr, http://www.cstb.fr/

Abdul Samad (Sami) Kazi, Dr.,
Technical Research Centre of Finland (VTT), Espoo, Finland
sami.kazi@vtt.fi, http://www.vtt.fi/

SUMMARY: *The European Semantic Web-based Open engineering Platform, project (SWOP 2008) is concerned with business innovation when specifying products to suit end-user's requirements and objectives. This paper will show how Semantic Web (SW) technology of the Word Wide Web Consortium (W3C) can be used to its fullest to model the products to be developed and configured. It introduces a Product Modelling Ontology (PMO) as the main result of SWOP. It is in essence a fully generic, freely reusable 'upper ontology' specified in the Web Ontology Language (OWL), the most prominent SW technology (OWL 2008). PMO contains all necessary and sufficient modelling constructs to define any end-user product ontology, taking into account all relevant end-user's product classes, properties and relationships (in particular the predefined 'specialization' and 'decomposition' relationships) together with cardinalities, data types, units and default values. Rules in the form of 1) assertions that have to be satisfied and 2) derivations that can be executed add the more complex product knowledge aspects. PMO has already been applied in many end-user situations and other R&D projects.*

KEYWORDS: *Product Modelling, Configuration, Semantic Web, Ontologies.*

REFERENCE: *Böhms M, Bonsma P, Bourdeau M, Kazi A S (2009) Semantic product modelling and configuration: challenges and opportunities, Journal of Information Technology in Construction (ITcon), Vol. 14, Special Issue Next Generation Construction IT: Technology Foresight, Future Studies, Roadmapping, and Scenario Planning, pg. 507-525, http://www.itcon.org/2009/33*

1. THE SWOP PROJECT

1.1 Objectives

The European SWOP, Semantic Web-based Open engineering Platform, project is concerned with business innovation when specifying products to suit end-user requirements. There are two main business drivers behind this innovation: (1) to reduce wasted effort in terms of cost and time in re-designing and re-specifying products when for the most part the work has been done before, and (2) to configure solutions from pre-defined partial solutions ('modules') rather than design from scratch. When there are choices, product configurations are optimized in SWOP by applying Genetic Algorithms (GA) so that the resulting product is not just a valid solution but even a near-optimal solution that can be achieved following design constraints, end-user requirements and optimisation criteria.

SWOP shows how semantic web technology can be used to its fullest to model the products to be developed and configured. It introduces a generic, re-usable ontology for product modelling enabling product decomposition, the specification of units for properties, the handling of default values and the various types of ranges over property values needed to fully describe both the solution and the requirements side with respect to those products. A special topic addressed is the bridge between semantic and non-semantic information in the form of documents, drawings or even visualisations linked to or better, derived from, the semantic information. As an example we will show how IAI IFC data (an open standard for product representations in the construction industry sector) can be fully derived from the intelligent object data involving both semantic and technical mappings (from Semantic Web to ISO STEP technologies).

What unifies this apparent diversity mentioned in the introduction for 'product' is the approach SWOP is taking. First, whatever the sector and particular application are, every aspect is reduced to semantic description – the product itself, the user need, the external influences, the evaluation (i.e. optimisation) criteria, the product's components that make up the product etc. Semantics are the meaning about some-thing, shareable by people and computer systems.

So, depending on context, it may be the model-ling of a 'client requirements' view of a product, a 'front-office' sales department's 'black box' view, a 'back-office' supplier's 'white box' view including all design details, or a process-oriented view on how to fabricate a product. Semantic descriptions (in the form of 'ontologies') are only possible with knowledge of the domain – what the concepts are that give the meaning. This is important not only to the modelling, but also to the user interface in configuration tools. Configuration tools are used to configure solutions. The engine of a configurator may be fairly generic, but the way it is veiled for the user is very context specific. In SWOP, configurators appear in two guises – as tools to formalize all product knowledge and as tools that configure those reference de-signs according to individual requirements leading to an end product for use.

1.2 State of the Art

ISO STEP

The oldest initiative to standardize product descriptions is ISO STEP covering both (1) technologies like STEP Physical File Format (SPFF) for the syntax of the data and the EXPRESS language as syntax for the data structure and (2) the data structures themselves. SPFF can be abstracted via a late-binding Application Programming Interface called Standard Data Access Interface (SDAI).

The problem with STEP is that the technologies involved are overtaken by web-based variants and that the models have been proven to be to complex and difficult to implement.

IAI IFC

Especially in the building industry, the actual modelling work in STEP was slow and not resulting in the right data structures. The initiative was taken by software vendor's to start the International Alliance for Interoperability (IAI) to develop a model in STEP technology called the Industry Foundation Classes (IFC) containing roughly three main parts:

- a limited semantic part;
- a large less end-user oriented geometry part;
- a small escape meta-model part (for proxies and property sets).

However, despite its limitations it is the best specification currently available for the building industry sector for Building Information Modelling (BIM 2008).

W3C Semantic Web (SW)

Although the web gave us already an alternative for SPFF namely eXtensible Markup Language (XML) and an alternative for EXPRESS namely eXtensible Schema Definition language (XSD); the power of the languages is limited to structure only and not really proving mechanisms to add ‘real’ semantics in the form of concepts, properties and rules. This is exactly what the Semantic Web Activity in W3C DOES bring us in the form of Ontology Web Language (OWL) and RDFS-XML as syntax for the content according to OWL-expressed ontologies.

In the next chapter we will tell how we extended this generic approach for the use in (semantic) Product Modelling.

2. SEMANTIC WEB TECHNOLOGY

2.1 Introduction

PMO, short for Product Modelling Ontology, is the main result of SWOP. It is in essence a fully generic, freely reusable 'upper ontology' (generic data structure with knowledge) specified in OWL, the most prominent Semantic Web (SW) technology from the World Wide Web Consortium (W3C). OWL is a more modern, fully web-based and distributed variant of the traditional ISO STEP technologies like EXPRESS and SPFF. Technically, PMO can be seen as a protocol stack layer on top of the series Internet, WWW, XML, RDF, RDFS and OWL specifically targeted at a generic way of 'product modelling'. End-user Products (on any complexity level so including standard catalogue items) will also be modelled by OWL ontologies reusing PMO.

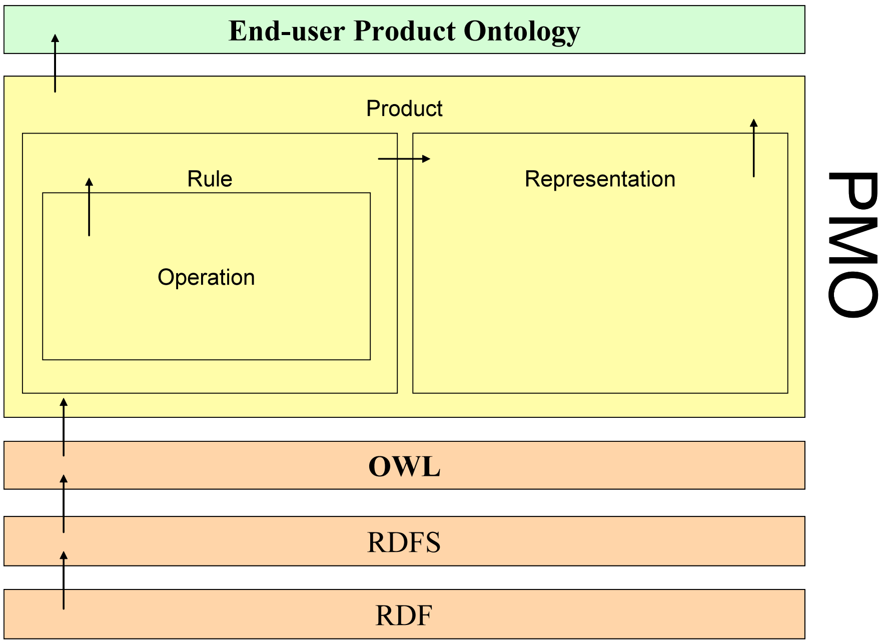

FIG. 1: Layered product modelling.

PMO contains in a necessary and sufficient way all constructs to define any end-user product ontology, modelling all relevant end-user's product classes, properties and interrelationships (in particular specialisation and decomposition) together with cardinalities, data types, units and default values. Rules in the form of assertions that have to be satisfied and derivations that can be executed add the more complex product knowledge. From this semantic end-user ontology in principle any representation/visualisation can be derived (think IAI IFC, COLLADA, OpenDXF or GDL). Currently, IFC2x3 STEP (SPFF) files and their XML variants (according to ifcXML) exports are supported. The primary semantic ontologies from which these formats are derived are however always the specifications that are used to integrate existing third-party software applications. Another, maybe even more interesting, application is where they form the basis for new advanced semantic applications such as smart product configurators often involving optimisation techniques such as Genetic Algorithms (GAs).

In contrast to approaches trying to develop THE ontology for a given domain, PMO envisions a more flexible, evolutionary and moreover distributed approach (in specification, use and maintenance) to product modelling for both software integration and development. PMO can be used in general by Se-mantic Web-tools such as the open source Protégé or the commercial TopBraid Composer (TBC) toolkits. More specialised support is availed by the SWOP modelling tools developed by TNO in SWOP (PMO Editor and PMO Configurator) that are currently being integrated. Some examples of the use of the PMO tools in the Building Construction context have been developed and will be presented along the paper.

2.2 The Resource Description Framework (RDF)

The basic specification underlying the semantic web technologies is the Resource Description Framework (RDF). This is a very well defined basic building block. Many researchers have made it a logically and mathematically sound approach. In its essence RDF is a semantic network graphically equivalent to a so-called 'directed graph'. This graph is fairly simple: there are nodes and directed edges between these nodes. Two nodes and one edge from one node to the other makes a 'triple' and a directed graph is nothing more than a set of these triples (triples become connected via shared nodes).

The input node is called the 'Subject', the output node the 'Object' and the edge the 'Predicate'. This basic construct is used to model almost everything for both information content/data and structure! Changing data or structure (or the links between tem) finally comes down to creating or deleting triples as an atomic action. Said otherwise, the sets of triples can be regarded as the optimally normalized relational model.

<Car> <**type**> <**Class**> .

<Person> <**type**> <**Class**> .

<has> <**type**> <**ObjectProperty**> .

<has> <**domain**> <Person> .

<has> <**range**> <Car> .

<MichelBöhms> <**type**> <Person> .

<CarWithLicenseNumber-70-GG-RR> <**type**> <Car> .

<MichelBöhms> <has> <CarWithLicenseNumber-70-GG-RR> .

FIG. 2: RDF example.

The essential message here is that with RDF we can describe any web content/structure/meta-structure/etc. (when using 'type' in nested ways) in the most simple way. With an RDFS/OWL-hat on we distinguish the structure from the data.

The triples can be represented in several, equivalent, ways. The above example used N-TRIPLE. Other forms are N3 (non XML-based), or RDF/XML or RDF/XML Abbrev(iated). The latter is a more compact form sacrificing however determinism. In RDF/XML Abbrev. there exist more than one equivalent RDF-XML files that differ more then just in the irrelevant order of triples. They are representing the same data but the final form is dependent on the order of parsing the file.

2.3 SWOP "World Assumption"

One of the most basic decisions in any modelling endeavour is the choice of the used 'world assumption': an Open World Assumption (OWA) or a Closed World Assumption (CWA). Without going into too many formal details, we can state the following: In case of an Open World Assumption, every-thing not said (stated, specified, modelled) is assumed to be "unknown", so it can still be true or false. In this case the model (ontology, schema, etc.) never forms a 'closed world': the environment is al-ways taken into account. Nothing is assumed false when unknown because you never now someone else outside the current scope (say a modeller in Timbuktu) might state something is true or false after all.

In case of a Closed World Assumption, every-thing not said is assumed to be “false”. In this case the model (ontology, schema, etc.) does form a 'closed world' on itself where the outside world is kind of ignored. Everything not said in this scope is per definition not true aka false.

Most traditional modelling approaches (like in e.g. ISO STEP's EXPRESS) apply a closed world assumption whereas more modern, especially web-based, approaches start from an open world assumption. In the SWOP project we make use of semantic web technologies which typically assume an open world.

A defined property in OWL is in principle a property of any class. So we have to add so-called ‘domain clauses’ to limit the relevance of properties to certain classes. In SWOP we assume only one class (having this property). So each user-defined property has a domain clause referencing exactly one domain class.

Subclasses are in general non-complete (their un-ion is not spanning the whole superclass) and over-lapping (or not disjunct). In SWOP however we decided for simplicity to only allow complete and disjunct subclasses. For flexibility and simplicity we decided not to model these constraints explicitly. We assume implicitly completeness and disjunctness for all subclasses for each superclass at 'configuration-time' by our software. At design-time this means one can more easily add a certain subclass when desired without changing too many rules for the relevant superclasses. We do not define a default subclass. For the PMO Configurator, the first specialisation sub class encountered (in the OWL file) is selected and displayed.

We (pre)define in PMO a decomposition object property that has the generic 'Product' class as both domain and range class. Hence, any end-user ontology class can be part of any other end-user class. For our product modelling we have to be more precise. That's why we use closures and QCRs (Qualified Cardinality Constraints) to limit the decomposition possibilities. With 'closures' we state what classes of parts are possible/relevant for a certain whole class. For those possible ones the default min and max cardinalities (as for all properties) apply, being 0 for the min cardinality and +INF(inity) for the max cardinality. We use QCRs to further constrain these cardinalities as required. For products that have no further decomposition (or "atoms") we define that the max cardinality of the decomposition property is zero (non-qualified, so for all possible qualifiers). For decomposition however we need some “default amounts” too. Adding a kind of annotation to a (qualified) restriction (min/max cardinalities) was considered too complex. Therefore, the default here is the same as the min cardinality value. Rules will affect these amounts of things (derivations or assertions will be taken into account via cardinality modifications respectively warnings). In the SWOP PMO Configurator GUI by TNO there will be a field for each qualified hasPart_directly object property indicating min and max cardinalities where an end-user can specify an actual amount in between.

2.4 Classes

Classes form the most basic meta concept in OWL. They are used to model the primary concepts with ‘members’ as their ‘extension’. These classes are not “object-oriented classes” with methods but are reflecting sets of members defined in some logical way (like by using predicates or via enumeration). The members of the classes are called ‘individuals’ in OWL.

It is important to note that we have to be very clear on the interpretation of classes and individuals: individuals are occurrences that exist in reality (or could exist if they don’t exist yet): one can (or could) point at them, everything else is a class. This means that a catalogue item is a class of which you can order three individuals. In product modelling we typically prefer a three-level approach (“generic-specific-occurrence”) involving, beyond a generic class and a particular individual, some ‘variant’ in between (partially or fully specified) that can be placed in space and/or time several times. In principle there are several ways of mapping the required three levels to the two levels offered by classes and individuals.

We have chosen for a way where variants are modelled as sub-classes. This is the most natural way since a variant indeed denotes a set of occurrences in the end that comply to the variant structure (just having a different placement in space or time). We can further distinguish predefined “standard” variants and on-the-fly defined 'end-user' variants.

2.5 Properties

Properties in OWL are a bit special for two reasons:

- They are so-called 'first class' concepts which means that they are on the same level as the Class concept. In many other modelling approaches, properties (attributes, slots, etc.) are secondary concepts: first there are entities, classes, etc. and then there are properties which are typically directly associated to such an entity, class, etc. Not so in OWL: classes and properties are considered equally important and modelled independently first and then interrelated where relevant, and
- Properties in OWL do not just denote simple (datatype) properties having a 'value' according to some data type like height, width etc. but they also cover relationships between classes. Said otherwise: if classes and datatypes are the nodes of an ontological network, the properties represent all the edges between them.

Let's first address the more simple Datatype Properties in OWL. For each property a domain and a range is specified. In the example below the do-main is a Window so it means that only a Window can have a windowWidth property. If nothing is specified, in principle, all classes can have this property. Next a range is specified, here being the float datatype reused from the XSD name space. Besides Floats, its also possible to have Integers, Strings, Booleans or more specific ones involving times and dates. In case of strings we can enter enumerations: sets of allowed values. Interrelationships between classes are modelled similarly properties now having individuals both as domain and as range.

With the notions of classes, individuals and properties we introduced so far all main 'archetypes' of OWL modelling. In a sense, all further modelling details are forms of what OWL calls 'condition modelling'. We will first consider a very important type of condition that got its own language element in OWL: subclassing enabling class specialisation to be modelled.

2.6 SubClasses

Remembering that all classes represent classes of individuals it follows quite logically to be able to de-fine subsets of individuals satisfying certain conditions. If A1 is a subclass of A it means that all individuals of A1 are also individuals of A. The subClassOf property is already predefined in the RDFS layer of OWL. The same way another sub-class of Product was defined: the Facade.

At design-time we assume a full open world assumption. However at configuration-time we will assume leaf classes having no further specialisation: if read in memory there is no known further specialisation. We will also always consider (implicitly) all same-level subclasses (sharing the same parent super-class) being disjunct, complete and allow only one super-class for each subclass. Finally we will always assume a choice of subclass to be made when configuring. This way we get not too complex specialisation tree structures and not too much overhead.

2.7 Cardinalities

For each property one can define minimum and maximum cardinalities, in the context of a class (in the SWOP situation: in the context of its one domain class). For each cardinality constraint a new, anonymous, superclass is defined representing 'all things having say a minimum cardinality of 1' (there should be at least one…) for a certain property. By making the class of interest a subClass of this class we actually express the condition that should hold.

3. PRODUCT MODELLING ONTOLOGY (PMO)

3.1 Introduction

As made clear in the previous section, OWL has a lot of power to model 'anything' including products. Still, there are some missing features, some of which are generic and foreseen in the upcoming OWL up-date 2.0 and some which are beyond the scope de-fined for this language. The latter are typically ad-dressed as 'modelling patterns' to be reused as a kind of best practices as described in (W3C BP). Just as with implementation-oriented 'software pat-terns' we can define 'product modelling patterns' that are useful in many situations but that are not directly supported by the language (OWL).

This can be regarded as a kind of layer in between the language (OWL) and the end-user product ontologies. We are talking small, reusable ontology parts like for modelling 'decomposition', 'units', 'default values', etc. Preferably these patterns are reused from a reliable, authoritative source. Fortunately W3C formed a "Semantic Web Best Practices and Deployment Working Group" for identifying, developing and promoting such patterns. Unfortunately they don't provide all patterns needed for SWOP and some in a way that are not directly suitable.

In this section we will define a minimal set of SWOP extensions needed to fulfil the SWOP product modelling requirements identified. In the end, all these constructs are collected in some small reusable OWL ontologies that have to be imported by any end-user ontology:

- product.owl (the top-level PMO ontology);
- representation.owl;
- rule.owl;
- operation.owl.

Collectively we refer to these ontologies as the SWOP Product Modelling Ontology (PMO).

3.2 Qualified Cardinality Restrictions (QCRs)

With normal (unqualified) cardinality restrictions one can say something about the amount of range individuals for a specific property in the context of a specific class. In case of data type properties this is typically fine. In case of object properties there could be alternative range classes relevant. If the range its type is a superclass with say 5 subclasses; we can limit the amount to 10 individuals of type superclass but we're not able to specify in more detail with respect to which type of subclass.

QCRs add exactly this information. Instead of saying a Pizza had max 5 layers we can now express that it should have max 2 cheese layers, exactly one meat layer and one or two sauce layers. This added expressiveness is crucial in modelling product de-composition as will be explained in the next section.

3.3 Product Decomposition

'Decomposition' is one of the most missed OWL built-in mechanisms. One of the standard OWL abstraction mechanisms is "specialisation" using sub-classing. For each class identified we can specify its superclass via a built-in "rdfs:subClassOf" property. This way we can model whole hierarchies of classes that are more or less generic/specific.

Now when specialisation corresponds to the 'logical or'-relationship: a vehicle is a car or a boat or a plane; decomposition corresponds to a complementary 'logical and'-relationship: a car consists of an engine, a chassis and the bodywork. Decomposition becomes a kind of orthogonal hierarchy with respect to the specialisation hierarchy. To some, decomposition is seen as even more important than specialisation since it seems to stand 'closer to reality': we 'think up' superclasses but we 'see' aggregates/composites. At class level we are talking typical decomposition and at individual level we have the actual decomposition (by some referred to as "object trees").

The W3C Best Practices pattern using someValuesFrom has some serious drawbacks:

- Each part always has at least one whole (some == at least one); it has no life of its own but always in the context of a whole, and
- The maximum cardinality cannot be controlled (we cannot state there is exactly one whole or ten, or less than fourteen etc.). The same is true in case of hasPart relationships if these are used (i.e. like 'a house has exactly three bedrooms' or 'maximum five bedrooms').

We can conclude that using "someValuesFrom" is not the optimal OWL mechanism. We expect much more from the yet-to-be-formally-introduced 'Qualified Cardinality Restrictions (QCRs)' which give us the power to control both min and max cardinalities for both directions (partOf and hasPart) in a more precise way indicating the valid target class amounts.

This new mechanism is expected to be present in the upcoming OWL2.0 update. We will now show how this powerful approach, as chosen for SWOP/PMO, works. In SWOP we will only use the hasPart_directly variant. Instead of the 'someValuesFrom' condition we get:

Example

```
<owl:Class rdf:ID="House">
    <rdfs:subClassOf>
        <owl:Restriction>
         <owl:onClass>
            <owl:Class rdf:ID="BedRoom"/>
         </owl:onClass>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1
         </owl:minCardinality>
        <owl:onProperty>
            <owl:ObjectProperty rdf:ID="hasPart_directly"/>
        </owl:onProperty>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Here the ‘someValuesFrom’ is replaced by a min cardinality constraint being 1 and no max cardinality constraint (default being ‘+infinity”). So a House has 1 or more BedRooms. Note that we have to specify clearly the underlying type now being BedRoom via the new (OWL2.0) ‘onClass’ tag.

Why so much fuss about decomposition? Well, we think it is one of the most important abstraction mechanisms for modelling objects around! With a clear best practice in the OWL context and supported by tools, it can be regarded as THE approach for modelling ‘class & object trees’ complementing the built-in specialization mechanism of OWL itself.

3.4 Meta-properties: Units and Default Values

Being able to model that “the weight of this ma-chine is 14” does not say much. We have to indicate the unit of measurement for the value “14”. In a sense, we have to put this value in the right context. There are many ways to add the fact that we mean “14 kg”. Some initiatives put it in the value: “14 kg” instead of just “14”. Other initiatives define a full blown unit ontology which is then related to the property and its value. In SWOP we have chosen a lightweight solution using “annotations”. Annotations are OWL’s way of escaping pure OWL, a means to extend OWL.

Formally annotations are just treated as meta-information not necessarily processed by OWL parsers but all signs are that they will get more importance in future versions of OWL (since meta-modelling is seen as key to more flexible modelling in general).

That’s why we decided in SWOP to use this feature for meta-data on properties, not only units but also for default values. For the current handling/visualization of ontologies (when no individuals are available yet) we use the “defaultValue" information to decide actual values for user-defined datatype properties.

Example

```
<owl:DatatypeProperty rdf:ID="facadeWidth">
    <product:unit rdf:datatype="http://www.w3.org/2001/XMLSchema#string">m</product:unit>
    <product:defaultValue
        rdf:datatype="http://www.w3.org/2001/XMLSchema#anySimpleType>5.5</product:
    defaultValue>
</owl:DatatypeProperty>
```

This way the units are modelled as meta-data at class level without the need to repeat them on individual level.

3.5 Representation

SWOP is primarily concerned with semantic information. Still there is a need to also handle non-semantic, representational information linked to or derived from the pure semantic information. Typically this information deals with shape aspects where implicit semantic information like height, depth and width are represented as parametric cubes, Boundary REPpresentations (BREPs) or even presented in nice (pixel-based) pictures. Unfortunately there is not one standard for expressing this information in a generic way. IAI IFC is a way of doing for the construction industry (especially for modelling 'buildings'). Another problem is the fact that such standards differ greatly in expressive power (even in this type of modelling we can distinguish different levels of smartness ranging from fully parametric explicit shape objects to simple pixels).

For this reason we decided to define a high level representation ontology ("representation.owl"). This ontology is imported by the product modelling ontology. This representation ontology is used as generic linking pin towards domain-specific representation schemes such as IAI IFC for construction. Further details with respect to representation issues in PMO can be found in (PMO 2008).

3.6 Rules

In the previous sections we have seen many forms of local rules or 'conditions' as they are called in OWL which are expressed within the ontology. Clearly in SWOP there is a need for several types of global rules. These rules are typically positioned above/on top of the ontology. We identify two main rules types:

- Assertions as integrity rules, that are checked/validated;
- Derivations as Production/Action Rules, that are done/executed.

Another distinction for rules is in 'determined' or 'undetermined'. A rule is 'undetermined' if it still contains some freedom like in case of logical or-expression or numerical or-range like 'X>25'. Assertions can be 'determined' or 'undetermined'. Derivations have to be determined. Both types involve logical and non-logical (like numerical or string) binary, unary and/or nullary operations. Some more examples to clarify:

Assertions

- wheelDiameter > (2.8 * axisDiameter);
- IF (AMOUNT (PowerSupply) >= 1) THEN (safetyLevel = "1" OR "2");
- IF (AMOUNT (window) >= 3) THEN (AMOUNT (door) < = 1);
- IF machineType = "special" THEN (bladeColour = green).

Derivations

- windowHeight := ((2 * windowWidth) + 0.04);
- IF machineType = "special" THEN (bladeColour := green);
- Again, more details on the rule mechanisms in PMO are described in (PMO 2008).

Typically the drawback of a meta-approach in OWL for rules is the tedious input process. Luckily EU project Manubuild (see chapter 5) developed a PMO Editor where you can specify the rules and their operations in a more user-friendly way and generate the actual OWL code needed.

4. EXAMPLE APPLICATIONS

4.1 In SWOP itself

In this chapter we will show for a, not too complex but typical, example how the SWOP Product Modelling Ontology (PMO) approach is applied. In this example object properties are not yet present. We start with a specific real life situation of a facade individual as depicted in the next figure.

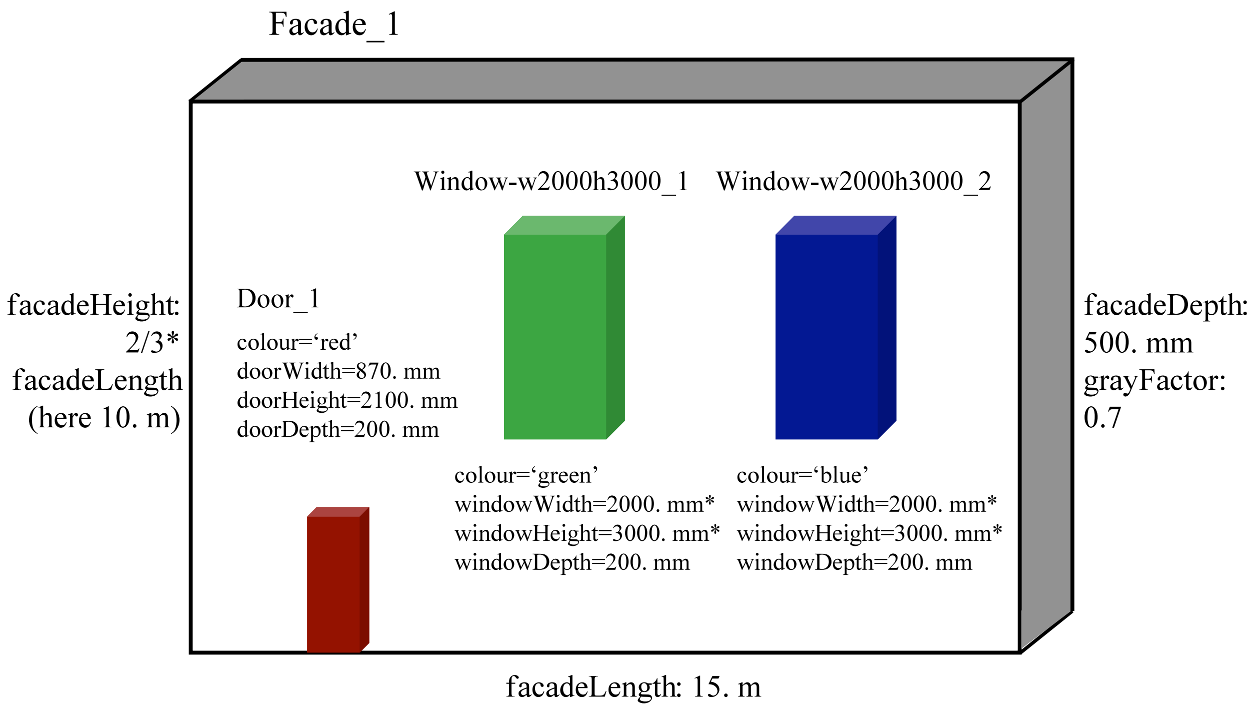

FIG. 3: The intended façade.

Applying PMO to this example we go through the following seven typical steps:

- STEP1: Define the Ontology files/headers;
- STEP2: Define the relevant Classes & Put them in a specialization hierarchy;
- STEP3: Put the Classes in an orthogonal (and typical) decomposition hierarchy;
- STEP4: Define the relevant (data type and object) properties for all the classes if relevant with their data types;
- STEP5: Define the semantic rules;
- STEP6: Define the rules for generating an IFC representation;
- STEP7: Configure the typical example in the PMO Configurator: Show it for the individual we has in mind.

This results in a visualization (figure 4) resembling the previous figure we started with (figure 3); in other words: the ’proof-of-the-pudding’.

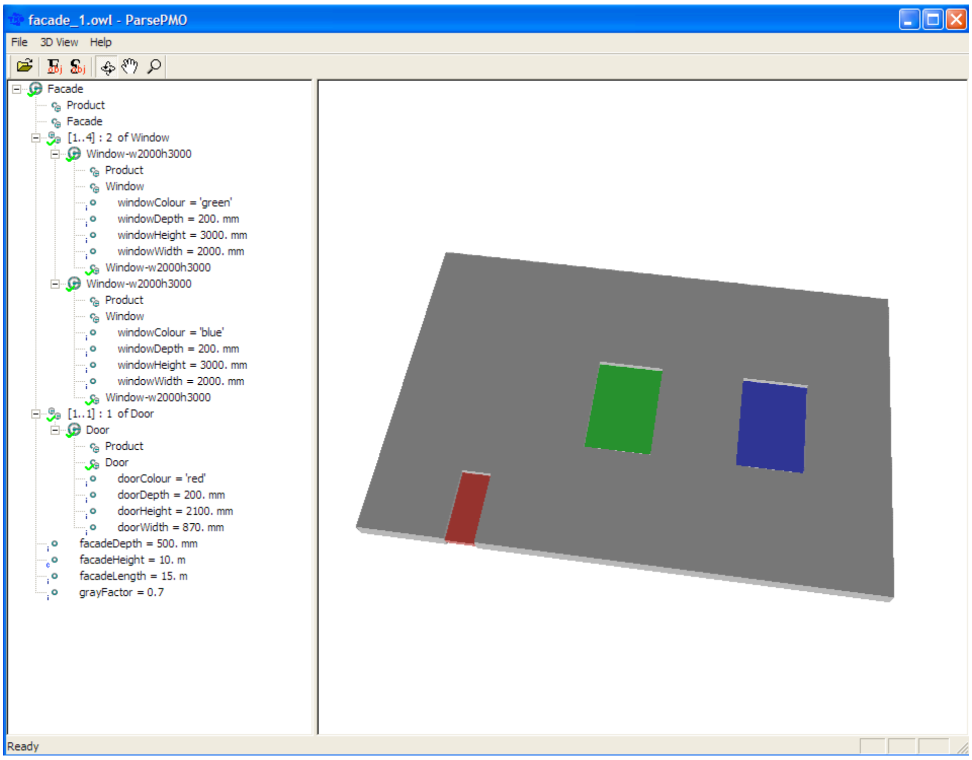

FIG. 4: The product configuration result.

Furthermore PMO is applied to the products offered by the end-user partners within SWOP (Blum/kitchen hinges, Züblin/tunnelling equipment, Saturn Engineering/sealing cap machines and Trimek/measuring devices). In the following figure it is shown that a measuring machine's parts, properties and underlying rules are modelled and visualized via IFC.

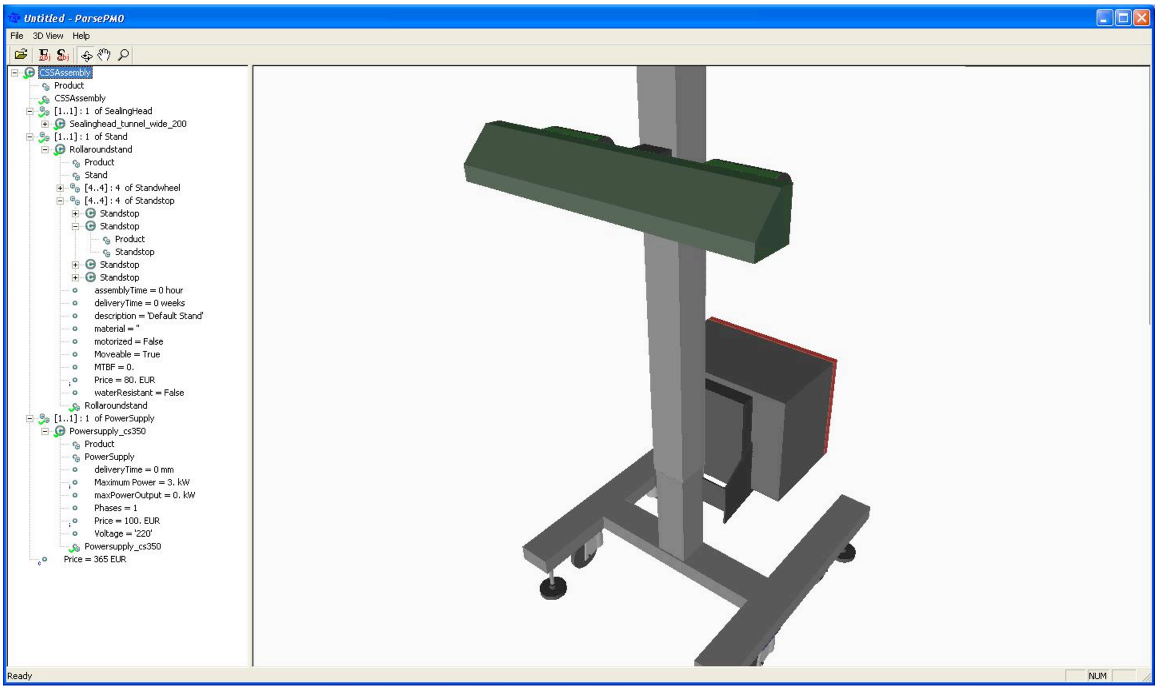

FIG. 5: Example from Saturn Engineering.

4.2 Use in Manubuild

In the European Manubuild IP the semantic modelling approach from SWOP was chosen as base methodology and technology. Software partner Graphisoft has made adapters to and from their proprietary Graphical Description Language (GDL). Application areas include a reference housing project by contractor Taylor Woodrow in the UK. The stair part here is a good example of how PMO can be used to model local regulations with respect to widths, heights, amount of steps etc. and to show on the fly the consequences of changed design parameters.

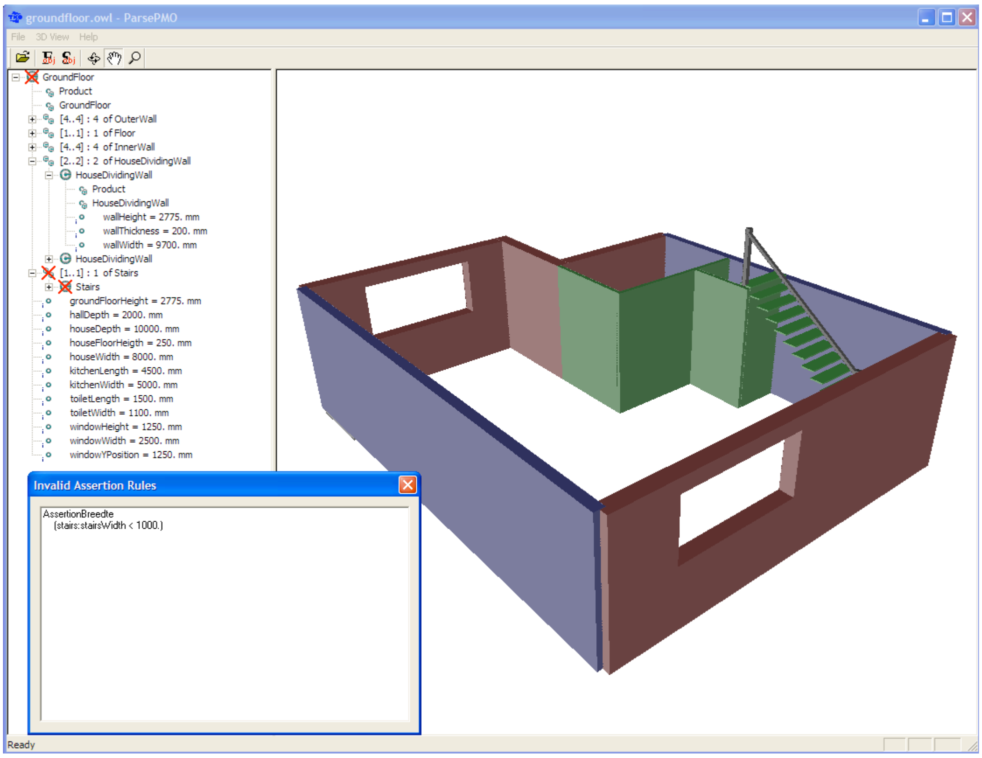

FIG. 6: Taylor Woodrow's 'Wimpey House'.

4.3 Use in InPro

In the running European FP7 InPro IP we defined a practical combination of the IAI Information Delivery Manual (IDM) and Model View Definition (MVD) approaches. Starting with process description we identified Exchange Requirements which were grouped for domains (like 'Architectural De-sign', 'Energy', Cost Estimation' and 'Product Data Management', and 'Time Scheduling') and mapped if possible to existing IFC entities and types.

Things not available in IFC are now modelled as SWOP's PMO-based ontologies defining/controlling the extra proxies and properties exchanged via the meta-model part of IFC (now in a managed way since we provide back links from the meta-data to these ontologies).

4.4 Use in IntUBE

In the European FP7 Intelligent Use of Buildings' Energy information (IntUBE) project PMO will be used as basis for an 'Energy BIM" providing the key component of the integrating platform for a variety of energy-related software application on the scale level of buildings but also their built environment they are positioned in.

One of the partners is SINTEF, also the home of BuildingSmart approaches like International Framework for Dictionaries (IFD) and the related ISO 12006-3:2007 modelling language. Therefore IntUBE is a unique opportunity to bring SWOP PMO- and ISO 12006-3:2007-based 'camps' in line.

4.5 Use in IOS

In the Netherlands there has been set up a group called Initiative group for Open Standards (IOS). Members are leading Dutch software vendors in architectural/installations design and analysis. They want to define and agree data structures for the bilateral links between their software packages in the interest of their clients. We developed especially for them the structured Microsoft Excel import capability that automatically generates PMO data structures.

We also developed for them an export to the much weaker ISO 12006-3:2007 specification that can be alternatively referenced back from an IFC model (the IFC we 100% generate for visualization of our PMO ontologies by means of the PMO rules and operations).

4.6 Use in COINS

In the Dutch Constructive Objects and the INtegration of processes and Systems (COINS) project their COINS BIM (CBIM) model is aligned with PMO. CBIM is positioned as layer around PMO, specializing our generic 'Products' root class into 'Functions', 'Spaces' and 'Physical Objects'. Also product 'requirements' are integrated with the 'solutions' offered by PMO.

5. SWOP ROADMAP

A specific work was undertaken at the end of the SWOP project to elaborate on the key research paths to pursue in the future to develop large-scale solutions for complex product engineering. This resulted in the elaboration of a roadmap that started with the definition of a vision, motivated and sustained by business needs, and continued with the identification and description of the different ways to reach that vision.

The so-called SWOP vision has been expressed as follows: to move from the current state to a widely-spread demand-based product configuration and optimisation that provides maximal value to customer (including time, cost, performance…), by relying on appropriate standard methodologies, models and ICT tools. To reach the vision, three key research topics have been indentified, and each of them has been broken down in a set of short, medium and long term actions in terms of time-to-industry: interoperability (product sharing), product structure (product modelling), and product configuration.

The visual representation of the roadmap is shown in Fig.6 with descriptions of its elements in terms of key concepts, topics (current state, short term, medium term, and long term), potential scenarios, potential business impacts, and main enablers and barriers in following sub-sections. It should be clearly noted that a roadmap is a snapshot at a given moment in time. It is supposed to evolve and be revisited along the time.

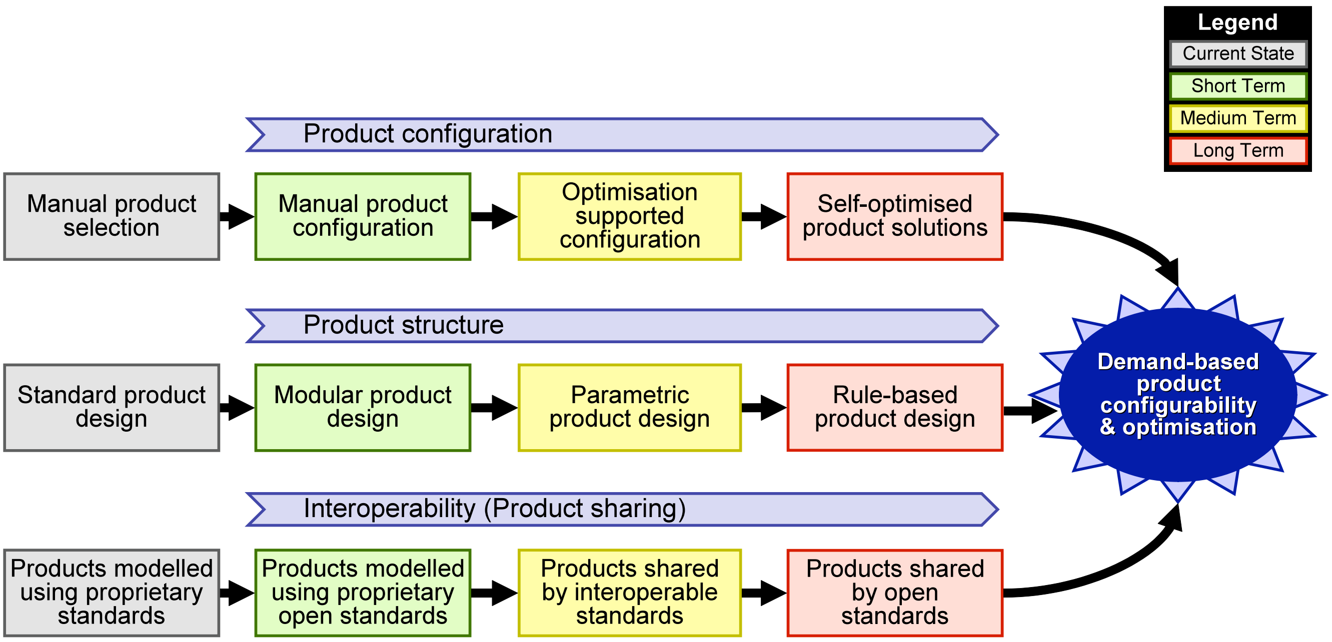

FIG. 7: SWOP Roadmap

5.1 Interoperability (Product Sharing)

Key concepts

It is frequently observed that products are modelled in accordance with proprietary standards. This essentially implies that relevant product data is not accessible by others. While in some organisations there is some openness (proprietary open standards), this requires strict compliance with and access to only limited features/data of a given product or service. To ensure full interoperability and furthermore ease of access there is a need to ensure that products are modelled and shared through recognised open standards. Inter-organisational product assembly and optimisation to deliver demand-based products will only be truly possible through reliance on product definitions (models) using open standards.

Current State (*Products modelled using proprietary standards)*: Most product manufacturers model their products based on proprietary standards. Little (if at all any) data access by others is possible. Most data access by others is through simple text files (e.g. pdf). At best data is only accessible through adhoc applications. These adhoc applications do not work any longer when an organisation or vendor (e.g. CAD/PDM provider) change their proprietary standard.

Short Term (*Products modelled using proprietary open standards)*: As a first step, it is expected that some organisations/vendors will allow some degree of access to product/service data through an API (Application Programming Interface). Even in such instances, access would be limited to only subsets of data that the organisation/vendor allows access to. Access in such cases would primarily be through interfaces supporting data translation from one application to the next.

Medium Term (*Products shared by interoperable standards)*: As organisations understand the need to share data across organisations and software applications, it is expected that product data will be accessible through established international standards. This will support product model sharing across heterogeneous systems. Data access would technically be possible using any application supporting the interoperable standard. The limitation in this case is that some organisations may need (in some cases) to pay a fee to allow them to implement access to data based on a given standard (even if it is international).

Long Term (*Products shared by open standards)*: It is expected that in the long term, products will be modelled using neutral formats that are public and freely accessible. In short, they will allow different organisations to share data, combine data, and access it through any application.

Potential Scenario: A client will have the capability to compare products from different product manufacturers, select the appropriate ones (could be from different providers), assemble and configure them, and then create an optimised product solution.

Potential Business Impact: The main foreseen business impact is access to product information in any format, at any time, from any product manufacturer, and through any software application. This has the potential to create a new market for solution providers that help clients define, select, assemble, configure, and optimise product solutions made of product parts from different manufacturers.

Main Enablers and Barriers: The main enablers can be identified as regulatory authorities that demand product definition/modelling using open standards. In addition to regulatory authorities, demands by large industrial alliances could play a pivotal role in ensuring product manufacturers and software vendors allow proper interoperability (seamless product data sharing). At the same time, some resistance from product manufacturers and software vendors may be expected. Many would like to tie-in clients to their offered solutions and not offer proper means for open sharing / interoperability. Furthermore, transforming product models from proprietary formats (even if they are open) to open standards may be time consuming. It should also be noted that most standards are based on the lowest denominator of sharing, and it may be that certain organisation specific data is lost upon conversion to these standards.

5.2 Product Structure

Key concepts: A product structure is the result of splitting up a product into components that interrelate to each other. The set of components including the interrelations form the structure of the product. Note that in many cases in practice the product is not split into components that have interrelations, but people started with a set of components and added interrelations which resulted in something they defined as a product.

There are different levels of complexity in product models.

The most basic form of a product structure is the one where components can be defined and certain type of relations can be defined between components. Probably the most basic relation that can be defined is specialization, a Car being a specialized version of a Vehicle is a product structure for Car. Another often found relation is decomposition, a Bottle of Water can be decomposed as existing from a Bottle and an amount of Fluid, in this case Water (being specialized component of Fluid).

The situation becomes already more complex when components are allowed to have parameters. The type of interrelation can say something about the existence of parameters. If it is defined that each Vehicle has a length, it will mean that each specialized version of Vehicle, for example Car, also has a length.

An even more complex product structure can be defined when rules are definable. There are two types of rules that could be defined within a product structure:

- Assertion Rules: these are rules that should be valid in any case; for example a Car has to be smaller then 10 meters (Car.length < 10 meters).
- Derivation Rules: these are rules that have to be executed at instantiation of the product, for example the length of the Car is the length of the Chassis + the thickness of the Bumper.

Current State (*Fixed Product structure*): In most cases when a product is described in a product structure this is a given product structure by the application or proprietary standard. In this case it mostly also contains a limited set of variables. Even though the benefits of having a product structure are there everything is fixed to the possibilities of the application and fits only perfectly in very rare cases or very specific software.

Short Term (*Fixed Product structure with freedom in variables*): As a first step, it is expected that some organisations/vendors will allow users to add variables to the different components in the product structure. This will already be a complex step because as described above some type of interrelations are of influence on the variables. Such a development will make the use of more abstract predefined (in the application or standard) product structures possible and with that the amount of possible products that would fit the predefined structures much larger.

Medium Term (*Free Product structure with variables*): The expectation is that more freedom in defining product structures will be asked from end users: a system where users are able to create their own product structures including interrelations and variables in an environment where these structures can be shared among other users all over the world. Some limited assertion rules will be expected to be available at this time also, rules like cardinality restrictions on decomposed products (e.g. a Car always has exactly 4 wheels) and assertion rules on domains of properties (a length of a Car is always between 2 and 10 meters).

Long Term (*Free Product structure with rules*): It is expected that in the long term, end users will be able to freely define product structures including complex assertion and derivation rules. It is expected that the product structure could make reuse of other product structures including rules and external applications and functionality could be integrated perfectly. In case of integrating 3D shape this would automatically result in parametric shape for example.

Potential Scenario: A client is able to reuse a well known more abstract product structure of its newly defined product together with parts of previous work. He is able to extend the used product structures to its own knowledge for his specific product and the inherited rules already give him a large set of the behaviour of the product including feedback on some overseen issues that his new product has. The knowledge from products stored in product structures with rules can really grow and help reusing existing knowledge in new never tested products before any physical version or virtual drawing is created by hand.

Potential Business Impact: The main foreseen business impact is reuse of product knowledge in a by the computer understandable way. This has the potential to create new products with the knowledge of already existing products without needing all persons involved in creating these existing products.

Main Enablers and Barriers: The main enablers are the current speed of computers and connectivity via the internet with high band width. The stored knowledge will ask much more processing power from the PC to understand and calculate through the rules within the product models. Also the high bandwidth will enable fast processing even when model parts are stored all over the world; this makes decentralized storage of product structure practically useful and the users can then gain from the benefits of such a setup.

Key concepts: Product configuration still remains a matter of engineers or experts who have the required (and often implicit) knowledge, e.g. on assembly rules. Configuration tools that are currently available to clients/customers (or even salesmen) generally show restricted functionality, with limited capabilities for customisation of available product alternatives. To enable clients/customers to find the best solutions regarding their problems, with appropriate configuration software tools, all necessary knowledge on products should be explicitly modelled and product/service configuration should be supported by optimisation techniques based on product performance assessment.

Current State (*Manual Product Selection*): Most often products are configured by manually selecting components from existing catalogues that present a list of available alternatives with limited capabilities for customisation per client needs. This process is based on a modular approach (i.e. the product is composed of several parts), but not a parametric one (product parts properties are fixed). E.g. Selection of tyres with alloy wheels for a car.

Short Term (*Manual Product Configuration*): In a first step, it will be possible to select and configure both product components and their parameters to meet user requirements. Choices will be made by matching some component parameters with specific requirements that are considered as constraints for the problem to solve (e.g. selection of walls with 10 cm of glass wool heat insulation for a building; selection of two disk memory of 160 Gbytes for a computer). Configurators are available through the Web where customers can directly configure products with no human assistance, but possibilities of choices are still limited.

Medium Term (*Optimisation Supported Configuration*): Configuration is supported by the optimisation of product and product components parameters based on user needs mainly expressed through technical requirements (e.g. a building consuming less than 50 $kWh/m^2/year$).

Optimisation will need the availability of methods to calculate and assess the global performance of the product (relatively to price, weight, annual energy consumption…). It will consist in minimizing/maximizing some product criteria. In some cases, criteria can have opposite effects (e.g. choosing lighter components can increase the price) in case of multi-criteria optimisation, leading to a set of optimised solutions (not only one). In such a case, it will be up to the end-user to choose his configuration from the possible solutions by privileging one or several criteria.

For very complex products, when the configuration of the whole product can't be achieved in a single process because of the number of parts and combinations, it should be necessary to divide the configuration process into a set of more elementary sub-configurations (of parts). Appropriate methods and tools will then support such sub-configurations of products, and allow distributed configuration amongst several sites and partners who are responsible for specific parts of the product.

Long Term (*Self-optimised Product Solutions*): In this long-term vision, products are configured and customised as a combination of different plug & play product/service modules. The configuration process is supported by optimisation techniques that suggest the optimal combinations providing the greatest global value to the client by taking care of required product performances or services to be delivered (e.g. number of patients to undergo surgery annually in a hospital / number of pupils that must have lunch at a school refectory everyday). As the product/service components will be modular in nature, re-configuration (i.e. disassembly then re-assembly) will be facilitated so as to address the changing product/service needs of the client during the lifecycle of the solution (product/service).

Potential Scenario: In a configuration tool, a client/customer specifies his requirements for a product he wants to design and configure, by using global value criteria based on product performances, delivered services, costs, delivery times, etc., and the tool shows him and values all the possible best solutions. This scenario is based on the assumption that customers define requirements for a product easier than describing product features (Stormer, 2007).

Potential Business Impact: The main foreseen business impacts are reduced product-design times and costs (by re-using pre-defined partial solutions under the form of modules, rather than designing from scratch), and increased customer satisfaction (thanks to more flexible, accurate, quick and optimised configuration and quoting systems). This has also the potential to create a new market for software solution providers around enhanced configurators and their interfacing with corporate systems (CAD, PDM/PLM, ERP, etc.).

Main Enablers and Barriers: Product manufacturers or product/service providers should regard the development of enhanced configuration tools as a significant factor to increase their competitiveness and market shares. On the other hand, there could be some difficulties to access the required knowledge on product modules, especially if those modules come from different manufacturers with conflicting marketing strategies.

6. FURTHER R&D

6.1 PMO Workbench

Currently we have separate design-time and run-time tools: the PMO Editor respectively the PMO Configurator. In the near future these tools will be integrated potentially together with built-in end-user requirements and objective into what we call a PMO Workbench. Genetic Algorithm (GA)-based services could be built in to ask to system to advice some or all input values instead of providing them by hand.

Also the construction-specific IFC-based visualization might be replaced by more generic Java3D variant.

6.2 GIS/GIM-links

Another extension dimension is the inclusion of higher scale levels towards Geographical Information Systems dealing with Geographical Information Models and their underlying technologies like Open Scene Graph (OSG) and COLLAborative Design Activity for establishing an open standard Digital Asset schema for interactive 3D applications (COLLADA), a currently used open standard by Google's SketchUp and Google Earth software products.

6.3 W3C Web Services

Our demonstrators up till now feature one supplier that, when needed, retrieved and modelled product information from his suppliers or their suppliers even further down the supply-chain. In the future when critical mass is building we foresee a situation where all data is modelled and kept up to date at the owner, only reused by others (distributed data).

Reasons like confidentially and sensitive information (prices, building element recipes, etc.) may re-quire connecting not directly to the distributed data but via the functionalities of sub-configurators manipulating that data. In that deployment scenario web services as (sub-) configurators over the web come directly to mind.

6.4 W3C Incubator Group on Product Modelling

Finally, all base PMO future developments ('PMO2') will be openly harmonized in a W3C Incubator Group (XG) for Product Modelling (short name: W3PM). This group has been initiated by TNO, Fraunhofer and the POSC-Caesar Association (PCA), all being member of W3C (W3PM 2008).

7. CONCLUSIONS

Semantic Web technology like OWL/RDFS, RDF-XML and SparQL provide a very well-defined, fully web-based and semantically powerful new way to conceptualise/model the world and access those models. It offers many advantages over alternative approaches including from OMG (UML, MOF), STEP technologies (EXPRES, SPFF, SDAI) and (plain) XML technologies (XSD, XML and .relevant APIs). Thos situation will only improve with OWL 2.0 and agreement on related standards as for 'rules'.

SWOP made this excellent basis even more suitable for "Product Modelling" by adding product decomposition, units, defaults values etc. in a generic reusable upper-ontology called PMO. The resulting (PMO-based) product ontologies are the perfect start for the development of software functionalities like product configurators and optimizers. Furthermore, these ontologies are the right interface specification for any semantic product data transfer or sharing activity.

Finally we indicated room for application and improvement for the short, mid, and long-term giving the approach the needed critical mass to become THE way of semantic product modelling in the future.

8. ABBREVIATIONS USED

3D – 3-Dimensional

API – Application Programming Interface

BIM – Building Information Model(ling)

BP – Best Practices

BREP – Boundary REPresenation

CBIM – COINS BIM

COINS – Constructive Objects and the INtegration of processes and Systems

COLLADA – COLLAborative Design Activity for establishing an open standard Digital Asset schema for interactive 3D applications

CWA – Closed World Assumption

FP – (European R&D) Framework Programme

GA – Genetic Algorithm

GDL – Graphical Description Language

GIM – Geographical Information Model(ling)

GIS – Geographical Information Systems

IAI – International Alliance for Interoperability

IDM – Information Delivery Manual

IFC – Industry Foundation Classes

IFD – International Framework for Dictionaries

INF – INFinity

IntUBE – Intelligent Use of Buildings' Energy in-formation

IOS – Initiative group on Open Standards

IP – Integrated Project

ISO – International Standardization Organization

MVD – Model View Definition

OSG – Open Scene Graph

OWA – Open World Assumption

OWL – Web Ontology Language

PCA – POSC Caesar Association

PMO – Product Modelling Ontology

POSC – Petro-technical Open Standards Consortium

QCR – Qualified Cardinality Restriction

R&D – Research and Development

RDF(S) – Resource Description Framework (Schema)

SDAI – Standard Data Access Interface

SPFF – STEP Physical File Format

STEP – Standard for the Exchange of Product model data

SW – Semantic Web

SWOP – ‘Semantic Web’-based Open engineering Platform

TBC – TopBraid Composer (by TopQuadrant)

W3C – World Wide Web Consortium

W3pm – WWW product modelling incubator group

WS – Web Services

WWW – World Wide Web

XML– eXtensible Markup Language

XSD – eXtensible Schema Definition language

9. REFERENCES

BIM, http://wiki.e-bouw.org/index.php?title=BIM, TNO's Semantic BIM wiki, April 2008.

COLLADA, http://www.collada.org/.

IAI IFC, http://www.iai-international.org/.

IFC (Industry Foundation Classes), International Alliance for Interoperability, http://www.iai-international.org/

InPro, http://www.inpro-project.eu/.

IntUBE, http://www.intube.eu.

ISO 10303 'STEP', Product Data Representation and Exchange, http://www.tc184-sc4.org/ (select 'SC4 Legacy Products' on menu).

Manubuild, http://www.manubuild.org/.

OMG, Object Management Organization, http://www.omg.org/

OWL, Web Ontology Language, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-features/.

PMO, SWOP Semantic Product Modelling Approach, with PMO – The SWOP Product Modelling Ontology, deliverable D23, revision 8, 15 April 2008.

Stormer, H. (2007): Kundenbasierte Produktkonfiguration. Informatik-Spektrum, Vol. 30, 2007, Issue 5, pp. 322-326.

SWOP, http://www.swop-project.eu/.

W3C BP, Best Practices, http://www.w3.org/2001/sw/BestPractices/.

W3C SW, Semantic Web Activity, http://www.w3.org/2001/sw/

W3PM, W3C Incubator Group (XG) on Product Modelling, http://www.w3.org/2005/Incubator/w3pm/, May 2008-April 2009.