CREATING VENDOR INDEPENDENT PARAMETRIC KNOWLEDGE USING CAD SYSTEMS

Peter Bonsma[1]*, Ton Damen[2], Roberto Di Giulio[3], Klaus Luig[4]

1: RDF Ltd, Bankya/Sofia (Bulgaria)
e-mail: peter.bonsma@rdf.bg, web: http://www.rdf.bg

2: DEMO Consultants BV, Delft (Netherlands)
e-mail: ton@demobv.nl, web: http://www.demobv.nl

3: Department of Architecture, University of Ferrara, Florence (Italy)
e-mail: dgr@unife.it, web: http://architettura.unife.it

4 3L architects + industrial designers, Menden (Germany)
e-mail: klaus.luig@3-l.de, web: http://www.3-l.de

Keywords: CAD, CAD API, CMO with Extensions, CSO, ICT, IFC, Parametric Design

Abstract *Within the EU project Proficient the open standard CMO with Extensions is developed. This Semantic Web based standard enables storing distributed parametric geometry and is a further development of the standard CMO developed in several other EU projects.*

An important practical issue within Proficient is to generate the content and knowledge according to this new standard. For clear reasons –in order to ensure the open and unlimited application of the end-users- no new CAD package will be developed as part of the project supporting this new standard. The result is that mainly existing CAD software will have to be used to create content.

As a result a conversion from CAD to CMO with Extensions needs to be created. Within the Building & Construction industry all major CAD packages support IFC export. IFC is an open standard with the goal to store the building data for all disciplines in the process, however with only very limited parametric representation power and missing coherent and holistic data. This means that the required parametric knowledge needs to be added afterwards in a dedicated application.

An alternative solution is to create a dedicated exporter for CMO within a certain CAD package. Given a powerful enough API and strong enough parametric capabilities this enables modelling the parametric knowledge directly in the CAD package in order to encourage the end users to operate the new system on a well-known software platform.

This paper describes the issues we found as well as benefits and drawbacks of conversion from IFC to CMO with Extensions compared to creating a dedicated CAD exporter given that also parametric knowledge needs to be integrated.

1. PROFICIENT E-MARKETPLACE

The research leading to these results has received funding from the European Community's Seventh Framework Programme under Grant Agreement No. 312219 (Project PROFICIENT). The aim of the Proficient project, funded under the FP7 programme 'Energy efficient Buildings' (EeB) is to facilitate and promote Collective Self-Organised (CSO) housing for energy-efficient neighbourhoods.

An e-Marketplace is developed within the project Proficient. The e-Marketplace enables end-users to make individual choices based on a parametric 3D model. It also allows users to add their own (structured) knowledge and integrate items from distributed catalogues. Both the suppliers and the end-users can find each other to complete the configured houses of the end-users via the e-Marketplace.

To make this possible a new Semantic Web standard has been created as back bone of this e-Marketplace. This standard is 'CMO with Extensions' [3], based on the Semantic Web standard CMO that was co-developed in other EU projects. The parametric 3D model is developed by an architect; it is assumed that this work cannot be done by an end-user. To make this e-Marketplace accessible for a large community we will need to support the current state-of-the-art tools architects are using.

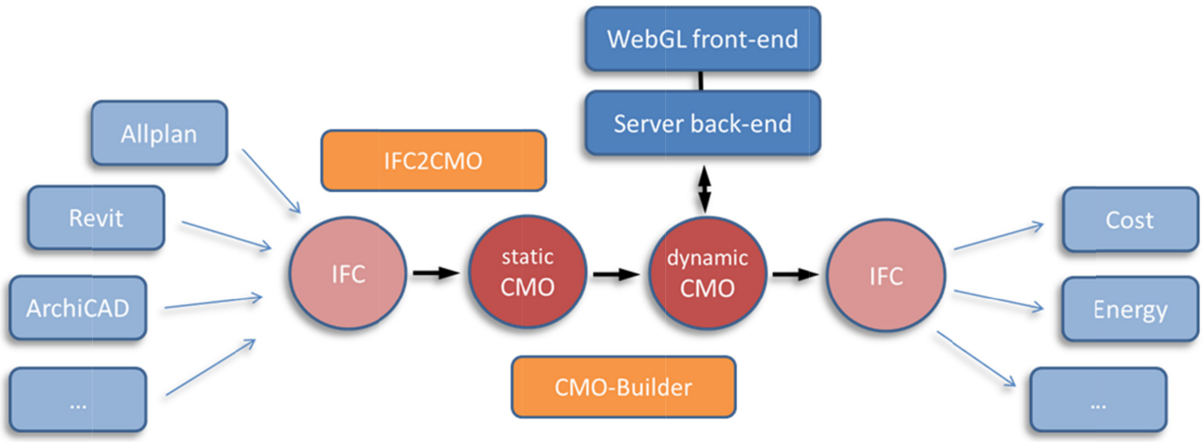

Figure 1, Architecture based on IFC

Above picture shows the initial process foreseen. As IFC has very limited parametric support it is expected we are mainly storing static information after the conversion IFC2CMO. A dedicated application will be needed to add the parametric knowledge. The e-Marketplace is used by the end-user to create a configuration within the boundaries of the parametric freedom. This configuration will be converted to a 'static' IFC file that can be used for quantity take-off and energy calculations giving 'real-time' feedback to the end-user on its configuration.

The main issue of this architecture is that a specialist is needed to add the parametric knowledge via the CMO-Builder tool, see also Figure 1. Instead of CMO-Builder commercial applications like Top Braid Composer could be used.

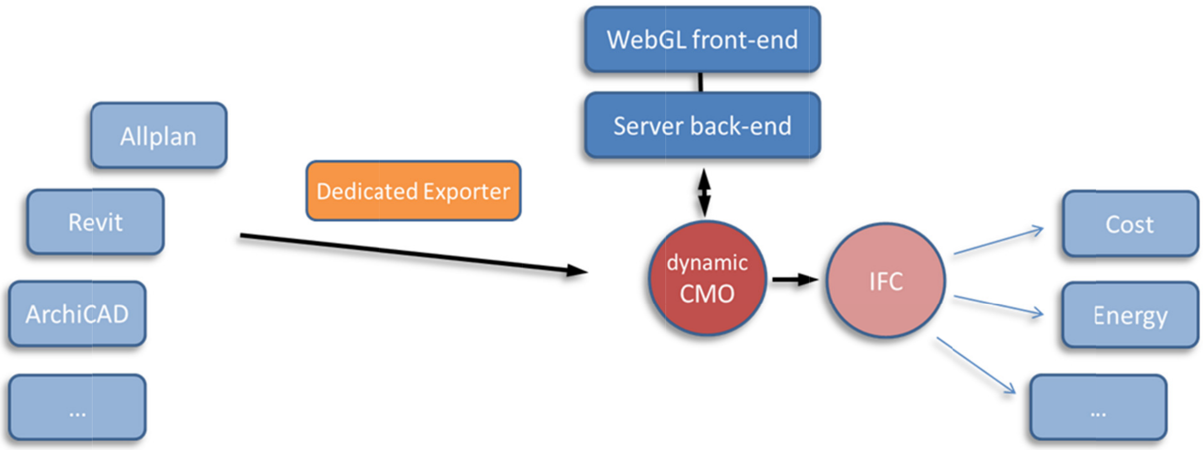

Figure 2, Architecture based on Dedicated Exporter

Above architecture shows an alternative architecture where the parametric knowledge is added by the architect in his CAD system directly.

This paper discusses the investigation and findings. The choices made within Proficient and reasoning behind are described in the conclusion.

2. ABSTRACTION

Within Proficient we started testing and prototyping both architectures, i.e. creating a dedicated exporter and creating an IFC2CMO converter as well. The following parts can be identified as being important for the eventual choice of architecture:

- CMO with Extensions
- Parametric knowledge within IFC
- Parametric knowledge within CAD

Eventually we will also include the context wherein the architecture has to be applied. Together with the findings on parametric knowledge this will results in a choice in architecture.

3. CMO WITH EXTENSIONS

The open standard 'CMO with Extensions' is an extension of the open standard CMO and based on the Semantic Web [3].

3.1. Semantic Web

The Semantic Web is a collaborative movement led by the World Wide Web Consortium (W3C) [5]. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web, dominated by unstructured and semi-structured documents into a "web of data". The Semantic Web stack builds on the W3C's Resource Description Framework. OWL is layer on top of RDF/RDFS and used as basis for CMO.

According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries." The term was coined by Tim Berners-Lee for a web of data that can be processed by machines. The e-Marketplace uses this functionality to connect and integrate data from the parametric 3D model and the distributed components (including 3D geometry) from suppliers.

3.2. CMO

CMO [6] is a small set of restrictions on top of OWL like a minimal layer on the stack. CMO with Extensions adds parametric and geometry support to the archetypes from CMO. CMO contains 7 archetypes, i.e. Activity, Concept, Energy, Object Space, SpaceTime and Time. Any class within a CMO ontology is expected to have at least one of these archetypes as one of its (indirect) parents.

Similar to the CMO archetypes, also one typical relation is named as CMO level, i.e. decomposition. Decomposition can take place on both instance and class level and there is a distinction between direct and indirect decomposition where the later allows nested decomposition relations between the two classes (and/)or instances.

Concerning properties a standard way of defining units is added (reuse of the ‘NASA model’ QUDT) and typical properties can be defined. A typical property in the context of a class is a property that is relevant for the class, however does not need to have cardinality restrictions.

3.3. CMO with Extensions

Most advanced CAD packages have parametric knowledge defined like a procedural script. What we found is that having a procedural language represented by an ontology was interfering with the idea behind multiple inheritance and the Semantic Web idea in general. As procedural scripts are the base for almost any parametric geometry we had a conflict, i.e. within the ontology it is unwanted to store (part of) a procedural language and for parametric geometry we require a procedural language.

In line with the work of solvers and inference engines the idea for creating a ‘converter’ from typical equations towards a procedural language was developed. This ‘converter’ is actually a ‘math kernel’ that similar to solutions in the Operation Research of mathematics is able to rewrite a set of equations into a new set of equations where the correct order based on interdependency is defined. The result is a lot of freedom on the ontology side of developing equations and applying multiple inheritance on classes (concepts) that are restricted to these equations. The ‘math kernel’ can then be applied to generate the procedural script based on a collection of equations. A first beta version of this math kernel is developed within Proficient and is working as part of the e-Marketplace.

The power of this solution was even more convincing after looking at the work of Yang Ji doing a survey of the parametric power of state-of-the-art CAD packages, like Siemens NX, Pro-Engineer and SolidWorks. While for all of them solutions for procedural scripts for parametric geometry are possible, the number of ‘bi-directional’ relations on top of geometry is limited to a small set of operations like ‘same angle’, ‘same length’, etc. Defining parametric geometry like done within CMO with Extensions enables a more generic solution for such 'bi-directional' parametric rules and integrates it with the standard procedural scripting possibilities. Therefore CMO with Extensions [1] not only enables parametric geometry within Semantic Web environment, but also creates a more generic solution for handling parametric knowledge. Not in the least it also is an Open Standard for storing parametric geometry, we will see later-on that although current open standards support some way of parametric knowledge, it often is a much more limited and predefined way of parametric knowledge.

Technically this parametric geometry/knowledge part of CMO with Extensions exists of three ontologies:

- Expression Ontology
- Relation Ontology
- Geometry Ontology

Expressions from the expression ontology can be defined independent from the other two. The Relation ontology imports the expression ontology but can be used independent of the geometry ontology. All of them import the CMO ontology itself.

4. PARAMETRIC KNOWLEDGE WITHIN IFC

During the development of the IFC2CMO converter an interesting capability of IFC was found. Although IFC has only very limited explicit parametric capabilities it is rich on relations and advanced geometry representations. This enables a lot of possibilities for implicit parametric knowledge.

4.1. Explicit Parametric Knowledge

Explicit parametric geometry is limited to a set of profiles. An example of such explicit parametric geometry is the A-Symmetric I-profile.

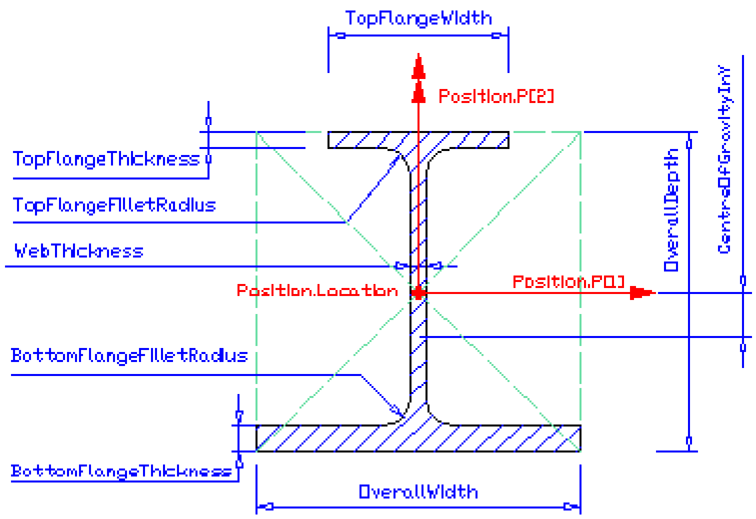

Figure 3, A-Symmetric I-Profile

The parameters drawn in the Figure 3 are given and geometry is defined based on these parameters. IFC contains 10-15 of such parametric profiles depending on the IFC version.

4.2. Implicit Parametric Knowledge

IFC contains a lot of implicit parametric knowledge. A better word for this kind of parametric knowledge could be re-definable parametric knowledge. It is important to understand that a lot of this parametric knowledge is depending on the application generating the IFC file. Different CAD packages used to draw exactly the same example building in the same manner could result in different IFC files with more or less re-definable parametric knowledge. Also the experience and style of working with the CAD package is of influence on the quality, correctness and completeness of the IFC file.

Two examples of implicit parametric knowledge are explained below. In principle IFC is rich enough to hold a large set of parametric relations, however recognizing such parametric information is not trivial from technical point of view. For Proficient it means that IFC2CMO can directly create dynamic CMO, however this means the converter needs to recognize the relevant parametric information.

4.3. Example 1

Windows and doors are defined as objects within an opening element. This opening element can contain its own geometry independent from how the door/window looks like. The geometry from the opening is subtracted from the object it is embedded in, this could be for example a wall or slab. If defined correctly the opening and door/window are places relatively towards each other and towards the object that is containing them. Although this looks like an overcomplicated relation, it actually makes replacing windows and doors within a wall or slab relatively easy. This is therefore a typical example of implicit parametric knowledge.

4.4. Example 2

Walls are often defined as polygons in a xy-plane and an extrusion in z direction. The top of the wall often is not exactly similar to a xy-plane, therefore operations on this basic extrusion definition is required. This often is done by defining clipping on a plane or polygonal bounded clipping. These clipping planes are implicitly depending on the roof, a clever application is technically able to adjust the clipping planes when the roof is adjusted. The same can be said for moving connected walls, for example when the rear wall is moved this should impact the geometry of the side walls.

5. PARAMETRIC KNOWLEDGE WITHIN CAD

The parametric capabilities of CAD systems for the Building & Construction industry have increased a lot. Although there is still a serious gap with applications from SolidWorks (in combination with TactonWorks), Pro Engineer and Siemens NX that have more advanced capabilities in parametric modelling.

For prototyping of a dedicated exporter a single version of a CAD package needed to be selected. As the available examples were all modelled with Autodesk Revit, the most recent version of this software package was chosen to build a prototype on. Although we also investigated ArchiCAD software with GDL (Geometric Description Language) the findings are exclusively depending on Revit.

In general the findings are that conversion from geometrical concepts, i.e. primitives like Box, Cylinder, but also Extruded Polygons, Swept Area Solids and Boolean Operations is possible although some semantics is inaccessible via the API. Building and retrieving geometrical material information like texture maps and normal maps is complicated. Not only defining this information parametrically in Revit is complicated, also converting this towards CMO with Extensions is complex if not impossible given that this information needs to be parametric also.

Within Revit parametric families can be created. This functionality is often used for components and libraries of such families can be created by the architect also. The expectation was that these families would enable modelling parametric knowledge by the architect for the e-Marketplace. In the example cases we looked at the parametric knowledge was needed at a higher aggregation level. For example adding a bay window, a dormer or even an extra floor is the typical knowledge that should be added.

Another option was to look at the Revit (family) templates. In our example cases the architects were not able to use Revit families and/or templates to define the parametric knowledge on this higher aggregation level. It was not excluded that there is a possibility to store such parametric knowledge within Revit. We found an existing solution where the parametric knowledge was modelled by using a project specific interpretation of layers where a dedicated Revit exporter was able to convert Revit content into parametric knowledge outside Revit.

6. CONTEXT

From technical point-of-view it is most interesting to define the parametric knowledge as close as possible to the specialist, preferably by the specialist itself. The specialist is in this context the architect with deep knowledge of the tools, i.e. CAD software, he is using.

What we found is that CAD software is supporting modelling of complex parametric knowledge, however this is mainly on the level of components. Although we did not exclude this is also technically possible for higher aggregation levels, typically needed for Proficient, it cannot be expected every architect is capable of defining this within his tool.

As Proficient is an EU project and the results expected to be used in a generic context and support for open standard would favour over vendor dependency. This means that unless the architecture of Figure 2 has strong benefits over the architecture of Figure 1, the later on is preferred.

7. CONCLUSION

During prototyping it was found that although IFC contains very limited parametric knowledge within the (geometric) concepts in the definition, the rich structure allows clever applications to recognize a significant larger set of parametric knowledge.

On the other side we found that modern CAD packages contain good parametric capabilities, however focused on components and complicated to use for many users of the CAD package. Typical parametric capabilities that are requested by the architects cannot be added to the CAD system or addition is too complicated.

The mismatch between parametric capabilities of CAD packages and what is understood by the architect and the implicit parametric capabilities of IFC makes the choice for using IFC as intermediate preferable within Proficient. Another reason is that by choosing the open standard IFC a larger set of CAD applications can be supported.

An important note is that with this architecture (see also Figure 1) the way of modelling is very important. Within Proficient we found that models from experienced 3D modellers are more correct and retrieving parametric knowledge from these models is more successful. Also each CAD package and version has its own behaviour in the quality of the exported IFC file. As the goal of Proficient and this process is to minimize the changes for architects we do not advise certain CAD packages to the architects but depending on the used package and experience of the architect more or less work is needed from the specialist adding parametric knowledge.

The chosen path via IFC seems to be the best solution for Proficient, however within Proficient there are specific requirements for parametric knowledge. Also only CAD packages were checked that are currently used by connected architects. Similar situations but with different required parametric knowledge and/or different used CAD software could lead to a different choice in architecture.

REFERENCES

[1] P. Bonsma, I. Bonsma, T. Zayakova, A. van Delft, R. Sebastian, M. Böhms, “Open standard CMO for parametric modelling based on semantic web” ECPPM 2014, pp. 923–928, 2014.

[2] Pasqualis, M. B, & Quentin, C. & Bektas, E.& Brouwer, J. & Savanović, P. & Bonsma, P. Guidelines for Participatory and concurrent design (D1.2 deliverable of the Proficient project) 2014.

[3] Bonsma, P. & Bonsma, I. & Zayakova, T. & Bohms, H.M. Semantic web platform and interfaces (public D6.2 deliverable of the Proficient project), 2014.

[4] Y. Ji & A. Borrmann & J. Beetz, M. Obergrießer: Exchange of Parametric Bridge Models using a Neutral Data Format. Journal of Computing in Civil Engineering (in print), 2012

[5] Website describing the basic concepts behind semantic web http://en.wikipedia.org/wiki/Semantic_Web, last accessed 30 October 2014.

[6] PowerPoint describing all concepts and related issues concerning CMO and CMO with Extensions, http://www.modelservers.org/public/documents/cmo.pptx, last accessed 30 October 2014.