INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

SUPPORTING MULTIPLE ENGINEERING VIEWPOINTS IN COMPUTER-AIDED DESIGN USING ONTOLOGY-BASED ANNOTATIONS.

Chun Lei LI (1), Chris MCMAHON (2), Linda NEWNES (1)
1: University of Bath, United Kingdom; 2: University of Bristol, United Kingdom

ABSTRACT

This paper describes an approach to supporting, in computer-aided design (CAD), the multiple evaluations that occur when engineers bring their expertise to bear, especially in the later phases of the design process. The aim of the support is to reduce the work in integrating external tools with CAD systems, and to increase the coordination between the different tools. The paper presents a general-purpose ontology-driven annotation approach to record viewpoint-dependent information such as manufacturing process and costing data. The annotation data are contained within a consistent ontology framework which supports the integration of multiple specialist viewpoints by associating annotation content with anchors in a boundary representation model. The ontology also allows checking of data structures and other reasoning. The paper gives an overview of the relevant background in CAD technologies, annotation and ontology, and then describes the embedding of an annotation tool based on the Web Ontology Language OWL in a commercial CAD system. The usefulness of the tool is evaluated through a case study of the incorporation of cost estimation tools.

Keywords: computer-aided design, design informatics, information management, annotation, design ontologies

Contact:
Prof. Chris McMahon
University of Bristol
Department of Mechanical Engineering
Bristol
BS8 1TR
United Kingdom
mecmcm@bris.ac.uk

1 INTRODUCTION

Engineering design is often a cyclic process in which solution concepts are proposed and developed, and their fitness for purpose is evaluated from multiple engineering viewpoints (MEV) such as performance, structural integrity, manufacturability, cost and so on. There is extensive computer support for the process, with computer-aided design (CAD) used for the modelling of the artefact's structure and physical characteristics, and a variety of tools, for example for simulation and analysis, used to support evaluation activities (McMahon and Browne, 1998). The trend in engineering computing has been for a closer integration of these tools into computer-aided design and manufacturing (CADCAM) and into product lifecycle management (PLM) systems used to support the product life cycle from requirements formulation to end of life (Saaksvuori and Immonen, 2008).

Since design requires specialist professional inputs based on special experience, the MEV evaluations need to be made iteratively using tools in a collaborative way. A problem is the rather ad hoc nature of the integration of MEV tools with CAD systems. Some tools are embedded – for example a CADCAM system may offer an embedded finite element analysis (FEA) feature – but some tools are independent, such that data has to be extracted from a CAD model and fed into an external tool for example to estimate part cost or life cycle impact. Whether embedded or not the MEV are often poorly integrated within CAD systems, with for example the material knowledge used for structural analysis being unavailable for making judgments about manufacturability. This paper seeks to contribute to improved methods of integrating such MEV evaluations and tools into CADCAM applications.

This paper describes an approach to supporting the multiple evaluations of a design that occur when specialist engineers bring their expertise to bear especially in the later embodiment and detail phases of the design process. Building on Davies and McMahon (2006), a general-purpose ontology-driven annotation approach to record viewpoint-dependent information associated with design models, such as manufacturing process and costing data, etc. is described. Annotations are used as general-purpose data carriers in the manner that they are used in the Semantic Web. The annotation data structure establishes persistent association between design information/knowledge and the design object. The data are contained within a consistent ontology framework which supports MEV integration in CAD. For example, a cost engineer might extract mass property information from the model and make shape complexity judgements for the purposes of cost estimation. MEV activities are supported by a consistent ontological framework in the mechanical engineering domain, which also allows checking of data structure and other reasoning. Ontology technologies establish a coherent and extendable approach, in which annotation plays the role of information media.

2 BACKGROUND AND RELATED WORK

During the process of engineering design, CAD systems play a critical role in providing a formal process of creating models of the product structure and form, especially through various three-dimensional (3D) modelling schemes of which the boundary-representation method (B-rep) currently dominates. The properties of the artefact are in turn evaluated through further models. Some of these are closely integrated with CAD models (Li et al. 2005), but there are many more or less specialist models used in different contexts, such as for costing analysis, assembly simulation or other specialised evaluation, that are not closely coupled with CAD tools. It is expensive and constraining to embed such analytical tools in CAD packages and this has only been done for a limited range of tools. Model transfer between tools often involves extensive and expensive manual interaction which is a non-trivial process limited by the participating engineers' understanding of the different viewpoints (Bond and Ricci, 1992). It is difficult to incorporate new tools into CAD systems or to specialise tools to particular application requirements. For these reasons there has recently been considerable interest in tools centred on multiple point-of-view modelling (e.g. Demoly et al., 2010)

2.1 Annotation

In a digital context, annotation is extra information referenced to a particular location of a digital information object. Annotation generally consists of two components: the annotation anchor (the location) and the annotation content (the extra information). We argue that annotation can be used widely in design, that design intent, purpose or constraints and so on can be recorded by annotating design models, for example when an engineering specialist associates a boundary condition with an analysis model or a machining parameter with a manufacturing model. However, while annotations are capable of representing data/information, they may be weak in representing knowledge. To overcome this, annotation has been combined with ontology-based technologies.

An ontology comprises a set of knowledge terms (i.e. vocabulary), the semantic interconnections, and rules of inference and logic for some particular topic (Hendler 2001). Ontology is formally modelled in a specification language in order to be computable. In recent years, many ontology specification languages and modelling tools have been developed (Corcho and Gómez-Pérez, 2000). Among these languages, the web ontology language (OWL) (Smith et al., 2004) has been adopted in the present work due to the fact that it is the major ontology specification language of the Semantic Web and based on the de facto web standard – the extensible markup language (XML). Protégé has been chosen as the modelling tool due to its usability and popularity (Protégé team, 2013).

The OWL language uses classes to specify concepts and these can be instantiated as individuals. Classes or/and individuals can be associated through two types of properties (relations): object property and data property, where the first describes relations between objects and the latter connect an object with data (e.g. integer, string). In addition, OWL also has supporting languages to enhance the expressiveness of logical and mathematical rules thus easing development and maintenance.

Ontology has been used extensively in engineering in recent years to provide a formal modelling framework for engineering concepts. The Knowledge Intensive Engineering Framework (KIEF) (Yoshioka et al. 2004) stressed a reasoning capability for multiple engineering viewpoints based on ontological features of function, behaviour and state. The work by Cera et al. (2002) describes a semantic annotation system for electromechanical assembly. However, these approaches are based on high levels of geometrical abstraction, e.g. assembly or part level or using predefined modelling features, rather than at the level of B-rep entities, and therefore they have limitations in their ability to associate knowledge with fine geometric detail, e.g. with faces, edges or vertices of part models.

3 USING ANNOTATION TO DESCRIBE ENGINEERING VIEWPOINTS

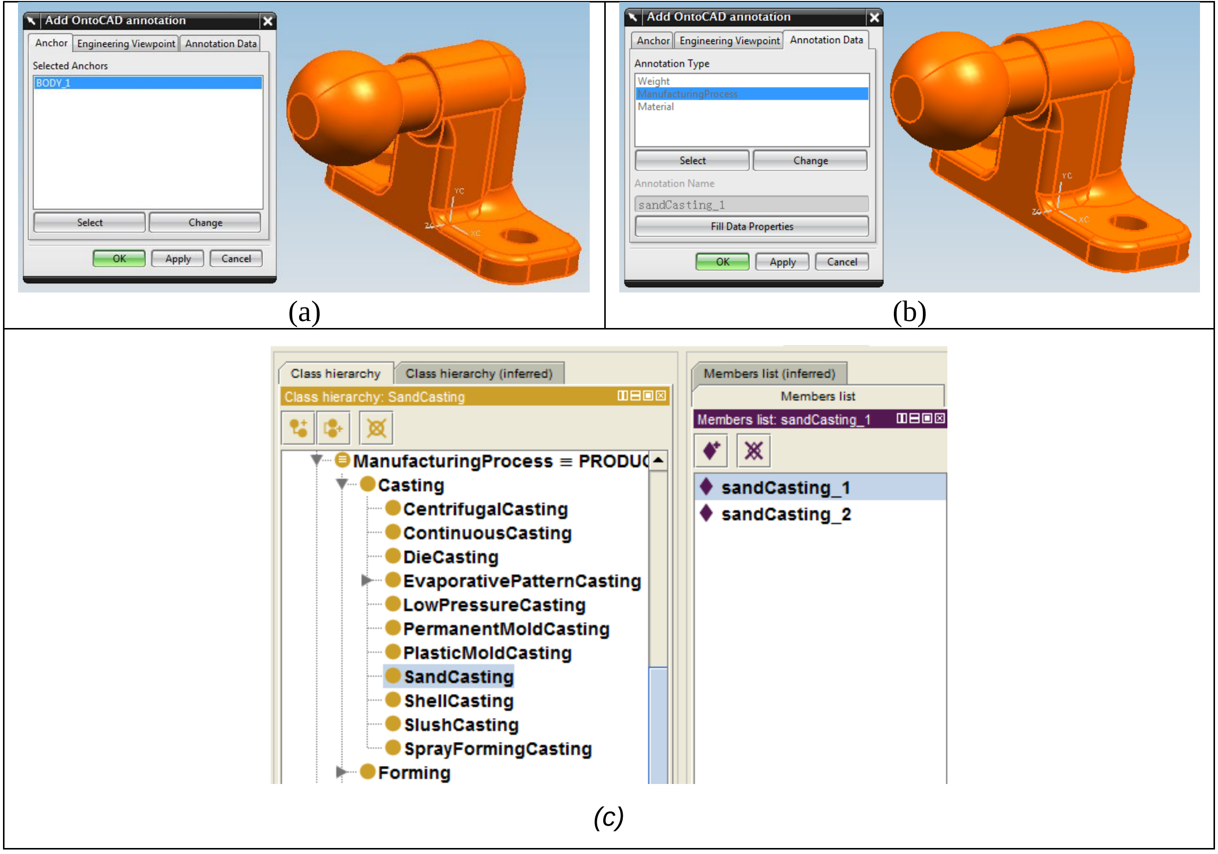
The tools used to evaluate artefact properties during the design process very often use a combination of data relating to the artefact and external data relating for example to the properties of materials, the parameters of the manufacturing facilities (labour cost, machine tool capabilities etc.). Data relating to the artefact can largely be extracted from or associated with geometric models of the artefact. Table 1 shows some of the data used in an engineering viewpoint model – cost estimation - and the elements of a boundary-representation CAD model with which the data can be associated or that can be used to derive the data. This table also shows the computational data types that would be needed to represent the data, for example in the form of an attribute:value pair (e.g. material:T304-stainless-steel).

Table 1 Engineering Viewpoint Data Embedded in CAD Models

Element	Data	Data type	Element	Data	Data type
Body	Material	String	Faces	Surface finish	Real number
	Manufacturing process	String		Manufacturing process	String
	Overall dimensions	Real number		Machining allowance	Real number
	Volume	Real number		Dimension	Real number
	Weight	Real number		Datum	String
	Numbers to be made	Integer		Tolerances	Real number

The objective is to identify whether data of the type shown in Table 1 can be associated with Boundary-representation CAD models in a consistent fashion and then used to automatically apply engineering reasoning, prepare data for and receive data from external applications such as the SEER-MFG™ cost estimation tool (Galorath, 2008) without specialist programming – i.e. the new viewpoint model can be incorporated simply by modifying the ontology. The basis of the approach is to allow annotations to be generated as instances of ontology classes anchored to elements of a B-rep CAD model. An experimental implementation of the approach, called OntoCAD, has been produced based on a CAD system (Siemens' NX) and the Protégé ontology editor. Figure 1 illustrates how users may use this prototype application to annotate a design artefact (a and b), in which the annotations are managed by the ontology-based model (c) working behind the scene. Persistent graphical visualisation of the annotations is the subject of future work.

The data of annotations in the experimental implementation is specified in OWL and their basic data structure is proposed as shown in Figure 2. There are three main types of annotation in terms of structure: data annotation, object annotation and annotation chain. Analogous to object and data properties of OWL ontology, object annotation uses an OWL individual as anchor and fills its content with another individual, while data annotation content is filled with data values. The third type of annotation chain uses chained properties and cannot link any further if a data property is inserted. For instance, *part1* → *hasMaterial* → *ABS* → *hasRawMaterialCost* → *ABS_Cost* → *hasValue* → *xx.xx*, where *ABS_Cost* may have other sub-links (e.g. unit £/kg), but the value of xx.xx cannot be linked any further. The annotations can be direct annotations that anchor geometry elements (e.g. point, edge, face, and hole represented as OWL individuals) or indirect annotations that anchor an arbitrary object, such as any part of the chain example.



Selected Anchors: BODY_1

Annotation Type: Weight, ManufacturingProcess, Material

Annotation Name: sandCasting_1

Class hierarchy: SandCasting

- ManufacturingProcess ≡ PRODUC
  - Casting
    - CentrifugalCasting
    - ContinuousCasting
    - DieCasting
    - EvaporativePatternCasting
    - LowPressureCasting
    - PermanentMoldCasting
    - PlasticMoldCasting
    - SandCasting
    - ShellCasting
    - SlushCasting
    - SprayFormingCasting
  - Forming

Members list: sandCasting_1

- sandCasting_1
- sandCasting_2

(a)

(b)

(c)

Figure 1. The OntoCAD Annotation Interface

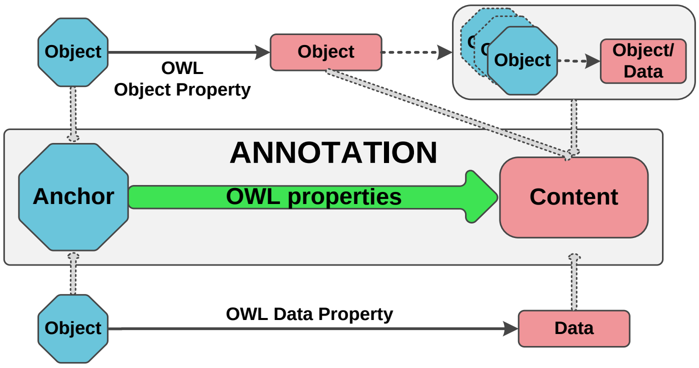


Figure 2 Annotation Structure

The advantage of this approach is that annotation data is specified in a unified language, which eases maintenance and data exchange. Another major advantage is the stand-off feature – data are recorded independently from target CAD models, therefore this structure enables the knowledge base to be portable to any CAD system using a compatible geometric representation. Since geometric models are specified as part of instantiated OWL ontologies and used as anchors, the same mechanism can be readily applied to anchor other types of product definition documents, such as text documents and multimedia documents. Therefore, this anchoring mechanism is ideal to but not limited to CAD models, which improves the evolvability of complex systems. To open up this possibility, an annotation language needs to be based on a formal language architecture, which is described in the following sections.

Based on the basic annotation data structure an experimental language architecture (Figure 3) has been developed. The bottom layers are designed on top of XML and in turn on URI (uniform resource identifier) and character set. The next higher level is the OWL, which represents engineering knowledge, and is potentially exchangeable with other knowledge systems. At the very top, knowledge is represented as annotation anchors and contents, which can be exchanged externally through a user interface and applications under the administration of an agent. Since the annotation data are specified in the OWL, the data can be queried and reasoned under the control of the agent. Furthermore, STEP standards are adopted so that the approach can provide unified terminologies that are common to external and internal systems (e.g. geometries, data types, and etc…).

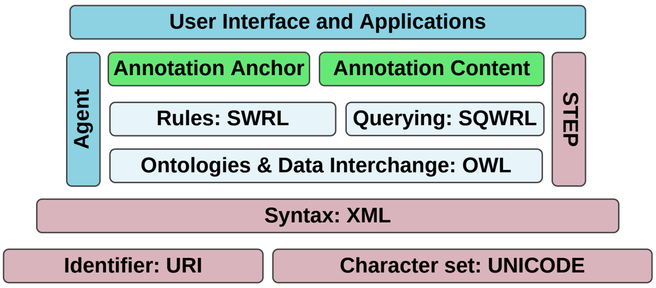


Figure 3 OntoCAD Architecture Stack

Based on this architecture a more concrete general-purpose ontological annotation framework is illustrated in Figure 4. The OntoCAD system is composed of three main modules: the Graphical User Interface (OGUI), the Knowledge Base (OKB) and the MEV Agent (OMA).

The OGUI is a user interface that is embedded into the CAD system, so that end users can annotate CAD parts interactively. In principle, anchors can cover any level of granularity and coverage as a feature inherited from STEP, i.e. the anchor can range from fine grain (e.g. a point or a surface) to coarse grain (e.g. parts, assemblies); it can also possibly be multi-point anchoring that references multiple geometry elements, or multi-directional anchoring that enables annotation data to be traceable reversely.

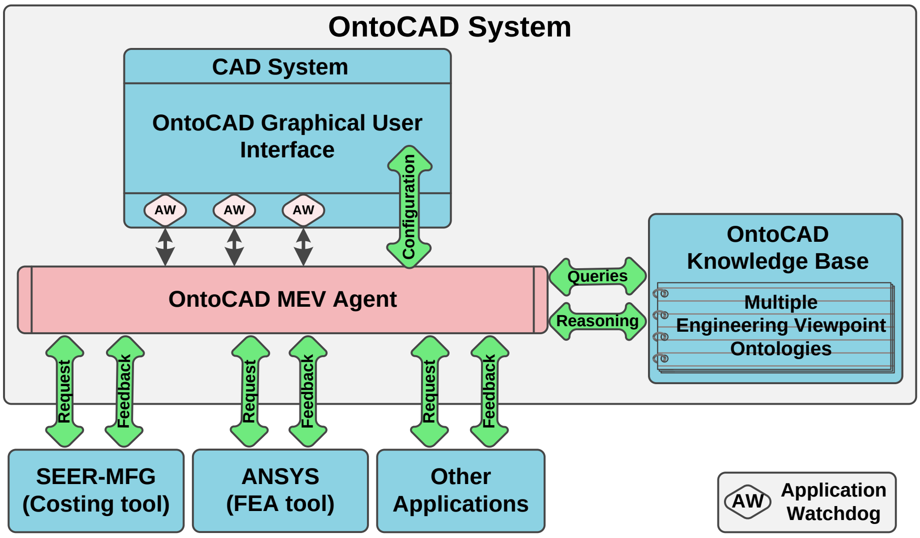


Figure 4 Overview of OntoCAD System

Through the OGUI interface, annotations are collected from users and then integrated with existing knowledge under the control of the OKB, which is responsible for representing both annotation anchor and content. To avoid the loss of annotations (e.g. labels, other attributes) during exchanges between CAD systems, a stand-off annotation strategy (Li et al., 2009) and STEP-compliant B-rep models are adopted as the semantic anchor representation in supporting CAD systems. Initially, three levels of granularity are covered in the experimental approach: G1 (body), G2 (face or faces) and G3 (edge or edges) as listed in Table 2.

Table 2 Three Levels of OntoCAD Annotation Anchoring Granularity

Annotation Anchor Granularity	**Annotation Anchor (OWL Class)**	**Annotation Anchor Identifier (OWL Individual)**
G1	manifold_solid_brep	BODY_1
G2	advanced_face	FACE_23
G3	edge_curve	EDGE_103

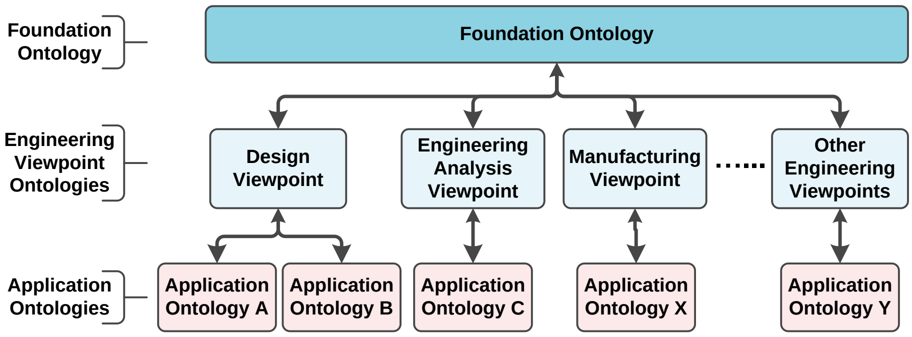


Figure 5 Architecture of OntoCAD Knowledge Base (OKB)

Annotation data is classified into overlapping MEV controlled by the three-layered OKB, shown in Figure 5. The first layer is the Foundation Ontology (FO), where common knowledge is defined including data types, measurement units and geometric representation. Figure 6 shows the non-geometric elements of the foundation ontology. Two notable FO classes are “measure_with_unit” and “data”. A measurement is generally composed of a measure value (or values) and a corresponding unit (or units), compliant with STEP standards. Data has many different types, including partially listed: binary, integer, string, real and so on. “measure_value” may associate with a single data type – real number. On the other hand, a unit may have a sub type – a named unit that has many other sub types, e.g. si_unit, length_unit, and so on. Each entity may associate with subtypes and contains axioms to define its conditions. For example “si_unit” has two attributes: “si_unit_name” and “si_prefix”.

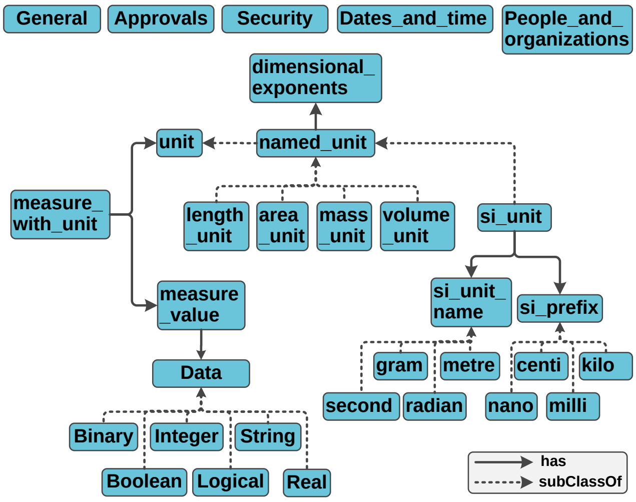


Figure 6 Partial View of the FO for Non-Geometric Classes

The second layer, Engineering Viewpoint Ontologies (EVOs), is the aggregation of EVs and each EVO can be treated as a collection of partials in other EVs. In other words, there can be some overlap among EVs. For instance, material itself can be treated as a primary EV, which can be referenced by not only the Design EV, but also the Analysis EV and the Manufacturing EV. This ultimately constructs interlaced taxonomies. An EVO of cost is shown as an example in Figure 7. It has two attributes in general – the value and unit. It has three subclasses: labour, material and tooling costs.

Engineering viewpoints cannot stand alone. Each is affected by a number of other classes or ontologies. For example, manufacturing processes have an effect on labour cost, as well as tooling cost, while material, part weight and shape representation affect material cost. For reasons of conciseness, not all ontology classes and interconnections are depicted in this diagram. Furthermore, the direct annotation associativity and granularity constraints (G1, G2 and G3) are defined. G1 indicates that this class can associate with the highest level of geometric representation, namely a body, while G2 comprises face(s) and G3 elements include edge(s). As shown in Table 1, geometric elements (annotation anchors) and all their attributes (annotation contents) are modelled into corresponding ontologies in OWL, and associated in the way as shown in Figure 1.

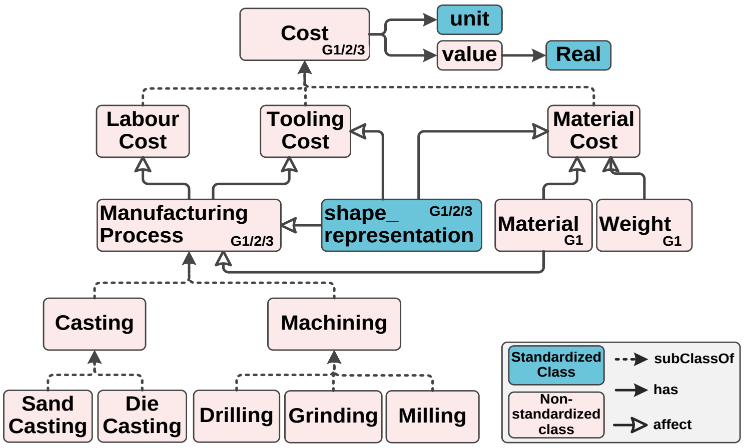


Figure 7 Partial View of Cost EV Ontology

The third layer comprises the Application Ontologies (AOs), which are basically thesauri. It defines the terms and properties used in specific applications. For example at Ontology Level EVO the definition in AO_SEER ‘PRODUCT_DESCRIPTION_-_Material’ has a corresponding class EVO_Material; at the FO level, ‘PRODUCT_DESCRIPTION_-_Finished_Weight’ has a corresponding class Weight.

The third key module – OntoCAD MEV Agent (OMA), is a broker that is responsible for handling requests from users or software applications, for instance to accept annotation data inputs via OGUI and update to OKB or to process a data query from a software application. Many EVO and AO ontologies can be developed by experts and developers as needed by following a regulated process. As more EVOs and AOs are collectively integrated, the OMA evolves a growing capability to coordinate among more EVs and engineering applications. More importantly, the OMA plays the important role of reasoning. Inference over the OKB decides knowledge reusability, and inferred knowledge can be reused again, which makes the system evolvable. Inference is classified into three main categories: factual reasoning, conceptual reasoning and methodological reasoning. These three types of reasoning activities have been developed in the prototype OntoCAD by editing reasoning rules, which can then be driven by existing reasoners, e.g. Pellet or Fact++.

Factual reasoning refers to reasoning operations at the data level. The appliance of factual reasoning includes consistency checking on individuals, individual membership and data query. Consistency checking ensures that ontology metadata is legitimate and individuals are properly instantiated during the development (e.g. debugging ontology data) and annotating processes. Individual membership computes the ownership of individuals, namely checking whether a given individual is an instance of a class, or whether a class is satisfied to have individuals. Individual membership makes the use of rules for engineering experience and design constraints through quantifier restrictions and value partitions. Value partitions restrict the range of possible values to an exhaustive list, for example, a specific number of products to be produced can be categorized as “MassProduction” or “Prototype” according to the conditions of “ProductionScalePartition” in the ontology.

Another intensive use of individual membership is the ‘application watchdog’ (AW), which is a named class with conditions representing rules constructed from axioms. An OMA reasoner checks whether this class is satisfied with any individual (instantiable) in the event of ontology changes. As a consequence, the status of an ontology model can be monitored. For example, an AW can monitor whether a set of parameters becomes available, such as some necessary cost drivers for a cost analysis activity, e.g. production quantity, manufacturing processes, mass and materials. Factual reasoning also contributes to data query in conjunction with other two types of reasoning actions, which will be described as follows. Once this set of parameters becomes available, the reasoner believes it can be instantiated, therefore provides corresponding service.

Conceptual reasoning is to reason over the conceptual level of knowledge based on class subsumption including equivalency checking. Similar to factual reasoning, conceptual reasoning can be used for consistency checking on classes. In the experimental implementation, conceptual reasoning configures the intuitive and adaptive OGUI for users or computer applications to collect data and knowledge. For example, the OGUI should be cost oriented if users claim themselves as cost engineers, and all cost related annotation options should be available. Conceptual reasoning also enforces end users to comply with the constraints of anchors. As noted, all information/knowledge is rooted in geometric models and the legitimate associations are defined in the OKB ontologies (Table 2). The behaviours of engineers or computer agents are restricted accordingly. For example, when a G2 anchor (face) is selected, weight is not available for annotating, but becomes available if a G1 anchor (solid body) is selected. Therefore, the bridges across FO (containing geometric models) and the other two lower levels of ontologies – EVOs and AOs can be established.

Methodological reasoning refers to dynamically deducing a result over both data and conceptual levels of knowledge, and maybe across ontologies. One appliance of methodological reasoning is semantic data query, which differs from standard data query in the means of involving reasoning on class conditions and ontology interrelations. Semantic query here refers to accurate and explicit data or class retrieval according to its context from OKB, such as using the synonyms of the ontological vocabulary. For example, the AO term "Powdered_Metals" (a subclass of "PRODUCT_DESCRIPTION_-_Process") defined for the costing tool SEER-MFG™ is equivalent to the EVO term "PowderMetallurgyMolding" (a subclass of "ManufacturingProcess"). When querying an instance of "Powdered_Metals", instances of its synonyms will be also retrieved. This mechanism builds the bridges among EVO and AO.

Methodological reasoning can also be used for applying engineering constraints through cooperation between EVOs, such as engineering rules for manufacturing and cost engineering. For example, methodological reasoning can inform users of the options to manufacture a particular product within a particular manufacturing factory. For instance, if a particular plant can only do sand casting in in the ranges from 200 grams to 100 kilograms, it will be identified (factual reasoning) for the plant whether a particular product can be produced. Or in the case of the weight of a design artefact is known, the reasoner can judge if sand casting is a candidate manufacturing process for this plant.

Moreover, the expressiveness and semantics can be extended by rule languages. With the SWRL rules, mathematical relations can be understood. As a result, conceptual reasoning can test for equivalency of classes, and in turn can assist with semantic data query. For examples, equivalent classes with necessary and sufficient conditions can be defined as one kilogram is equivalent to 1,000 grams or 2.204 pounds, or the mass of a solid body equals its material density times its volume. Therefore, unit conversion can be automated within the knowledge base and further support semantic data query. The OMA uses such sophisticated rules for complex reasoning to improve the level of process automation. As a fact, this is the feature of procedural annotation.

4 CASE STUDY AND IMPLEMENTATION

To show that product information including design and engineering analysis data can be recorded, retrieved and further manipulated, a prototype software application to demonstrate the capabilities of the OntoCAD approach was developed in the JAVA language, operating as an add-on application to a CAD system. Through this prototype application, engineering evaluation cases may be implemented to evaluate the feasibility and usability of the approach. The cost estimation case is described here, and illustrated in particular that OntoCAD supports the incorporation of MEV not currently supported by CAD systems, where it focuses on the demonstration of modelling methodology. The case is focused on the embodiment and detail phases of the design process, where CAD systems are widely employed by specialist engineers.

To integrate the SEER-MFG™ costing tool with the CAD system, the costing tool needs parameter inputs including dimensions/volume, materials, geometric features and so on to calculate cost results. In order to feed data to the costing tool the cost engineering viewpoint needs to be integrated into mechanical engineering ontologies and then populated with data.

First of all, the OKB was modelled starting with upper ontologies (namely the FO) from scratch. Secondly, a cost EVO was modelled based on the required costing tool input data. Some other EVO prototypes such as manufacturing, material were also modelled in their very initial state for demonstration purposes only (to explore cross-viewpoint reasoning). In the third step, an AO corresponding to the costing tool was modelled based on terms used in the application itself. The ontologies were modelled in OWL language using an ontology editor Protégé (version 4.1 Beta).

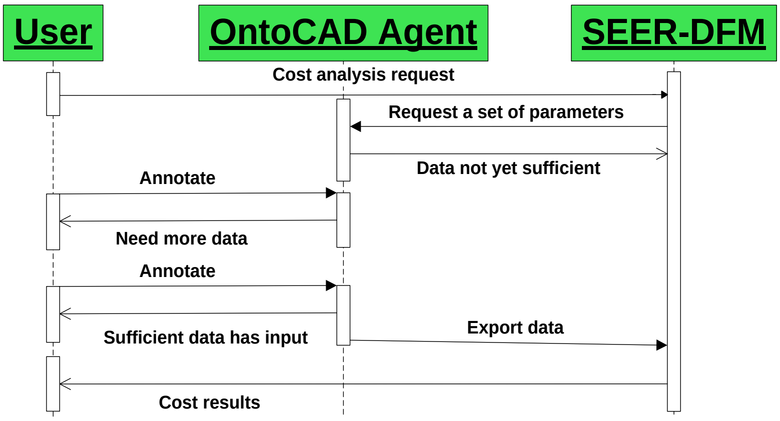


Figure 8 Sequence Diagram for Integration of CAD System and Costing Tool

As illustrated in Figure 8, once the user initiates a cost analysis request, the costing tool sends a request to the OntoCAD system, in which the agent makes a judgement whether the current data is sufficient for the costing tool to operate, as the example for AW made earlier. If not, the agent asks the user to provide more information (by annotating the CAD model as required). This is because, the OntoCAD programly extract data from a CAD model through a CAD APIs, and pre-populate into ontoliges. The agent then assigns an AW, which implements three types of reasoning actions. It observes the status of OKB for when data becomes sufficient (factual reasoning). Once sufficient data become available, the OntoCAD agent will retrieve and export the data (conceptual and methodological reasoning) to the costing tool to compute and then return a real-time result to the user.

The costing tool has an interface that takes commands to read a spreadsheet for importing parameters so that it remotely builds a project. Having built an EVO for cost estimation and an AO for this costing tool, a set of required data can be explicitly queried and retrieved from the knowledge base and a spreadsheet in the appropriate format generated, in order to operate the costing tool. The OntoCAD system can automatically extract and populate CAD model related data into the OKB, and some other data requires interactively entered by users. In addition, the state of satisfaction for the dataset queries can be monitored based on ontological rules, without modification to the cost tool itself, or to the OntoCAD system. The only changes made are to the OMA module, which acts as a central configuration manager. This shows the ability to readily integrate a legacy application with a CAD system without laborious programming and the corresponding skill requirements.

5 CONCLUSIONS

In the present work, developments in PLM and CADCAM systems have been briefly reviewed, and the technologies and applications of annotations and ontologies as computational enablers have also been explored in diverse engineering fields. From this preliminary study, current research challenges have been identified, including knowledge representation in the incorporation of multiple engineering viewpoints (MEV) assessments in CADCAM in order to assist collaboration in the engineering design process, data interoperability and CAD oriented tool integration, especially at the later design stages: embodiment and detailed design.

To overcome these challenges in current CAD approaches, the OntoCAD system of semantic annotation is proposed to assist mechanical piece part design activities, especially in the multiple engineering assessments and actions (e.g. costing, manufacturing process planning, FEA) undertaken in automotive, machine design etc., especially at the fine detail of design artefact. Some key features include:

- A general-purpose object-oriented framework that combines CAD system, annotation and ontology so that each module complements each other. This approach provides a closed loop of knowledge and information management including knowledge acquisition, representation, management and manipulation, which can incrementally evolve.
- A new semantic annotation data structure that uses STEP-compliant ontological boundary representation to describe persistent annotation anchors in order to strengthen the association between CAD model and engineering knowledge. The anchoring mechanism is capable to deal with all levels of granularity, e.g. to explicitly handle face and edge entities of CAD models, which distinguish this fundamental architecture from the others. The stand-off annotation strategy in which annotation data are stored separately from the annotated models allows annotation data to be portable or exchangeable.
- A three-layered ontology that allows the knowledge base to be modularised so that it can be configured to a lightweight system, or to integrate more engineering services or tools into an integral environment. This allows the system to be incrementally extendable based on current generation CAD systems. Furthermore, this knowledge base supports automatic reasoning based on MEV to aid the engineering design process, benefits data interoperability and improves collaboration among MEVs.

Prototype software has been developed as an add-on application to a CAD system. With this prototype, it is demonstrated that the OntoCAD approach has positive contributions in terms of application integration, semantic data exchange and reasoning over the knowledge base in two example applications. For the ongoing prototype and the ontology metadata please refer to the digital resources, available from http://dl.dropbox.com/u/45907729/OntoCAD.rar.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of EPSRC Grant EP/E00184X/1 for the work reported in this paper.

REFERENCES

Cera, C.D., Regli, W.C., Braude, I., Shapirstein, Y. and Foster, C.V. (2002), A collaborative 3d environment for authoring design semantics, *IEEE Computer Graphics and Applications*, Vol. 22 pp.43-55.

Corcho, Ó. and Gómez-Pérez, A. (2000), A roadmap to ontology specification languages, in: *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, Springer-Verlag.

Davies, D. and McMahon, C.A. (2006), Multiple viewpoint design modelling through semantic markup, *Proceedings of IDETC/CIE 2006*, ASME, Philadelphia

Demoly, F, Monticolo, D, Eynard, B, Rivest, L and Gomes, S, (2010), 'Multiple viewpoint modelling framework enabling integrated product–process design', *International Journal on Interactive Design and Manufacturing*, Volume 4, Issue 4, pp. 269-280

Li, W.D., Lu, W.F., Fuh, J.Y.H, and Wong, J.S., (2005), Collaborative computer-aided design--research and development status, *Computer-Aided Design*, Vol. 37, pp.931-940.

Li, C.L., McMahon, C.A. and Newnes, L.B. (2009), Annotation in product lifecycle management: a review of approaches, in: *29th Computers & Information in Engineering Conf*, ASME, San Diego

McMahon, C.A. and Browne, J., (1998), *CADCAM: Principles, Practice, and Manufacturing Management*, 2nd ed., Pearson Education, Harlow, UK.

Protégé Team, The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/

Saaksvuori, A and Immonen, A., (2008), *Product Lifecycle Management*, 3rd ed., Springer, London..

Bond, A.H. and Ricci, R.J. (1992), Cooperation in Aircraft Design, *Research in Engineering Design*, Vol.4, pp.115-130.

Smith, M.K., Welty, C. and McGuinness, D.L. (2004), *OWL Web Ontology Language Guide*, World Wide Web Consortium.

Yoshioka, M., Umeda, Y., Takeda, H., Shimomura, Y., Nomaguchi, Y., Tomiyama, T. (2004), Physical concept ontology for the knowledge intensive engineering framework, *Advanced Engineering Informatics*, Vol.18, pp.95-113.