Title	A product design ontology for enhancing shape processing in design workflows
Author(s)	Catalano, Chiara E.; Camossi, Elena; Ferrandes, Rosalinda; Cheutet, Vincent; Sevilmis, Neyir
Publication Date	2009-10
Publication information	Journal of Intelligent Manufacturing, 20 (5): 553-567
Publisher	Springer
Link to publisher's version	http://dx.doi.org/10.1007/s10845-008-0151-z
This item's record/more information	http://hdl.handle.net/10197/1444
DOI	http://dx.doi.org/10.1007/s10845-008-0151-z

A Product Design Ontology for Enhancing Shape Processing in Design Workflows

Chiara E. Catalano[1], Elena Camossi[2], Rosalinda Ferrandes[1,3], Vincent Cheutet[4],

Neyir Sevilmis[5]

[1] Istituto di Matematica Applicata e Tecnologie Informatiche (CNR), Department of Genova, Via De Marini, 6, 16149 Genova, Italy,

{chiara.catalano@ge.imati.cnr.it}(*)

[2] School ol Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland

{elena.camossi@ucd.ie}

[3] Laboratoire G-SCOP, 46 Avenue Felix Viallet, Grenoble cedex 1, France

{rosalinda.ferrandes@g-scop.inpg.fr}

[4] Laboratoire LISMMA, SUPMECA, 3 rue Fernand Hainaut - 93407 Saint-Ouen Cedex France

{vincent.cheutet@supmeca.fr}

[5] Fraunhofer Institut für Graphische Datenverarbeitung, Fraunhoferstr. 5, D-64283 Darmstadt, Germany

{neyir.sevilmis@igd.fhg.de}

Abstract. Effective and efficient information management and knowledge sharing has become an essential part of more and more professional tasks in the Product Development Process. Among the varius Knowledge Management technologies, ontologies offer new possibilities for representing, handling and retrieving product related knowledge, and for online collaboration. We conceived a Product Design Ontology (PDO) which especially addresses researchers in industrial product design and engineering analysis who need to share shape data and to develop software tools. In particular, we formalised the task-specific information associated to a shape, and the functionality and usage of shape processing methods in specific tasks of the design workflow. The PDO, thanks to the ontology-driven metadata on shapes and tools, may be also useful in industry on the one hand for training and for retrieving shape-related information, on the other hand for supporting benchmarking of processing tools and gathering the knowledge about shape processing workflows.

---

*(*) Corresponding author; phone:+39 010 6475669, fax: +39 010 6475660

Keywords: Ontology-driven retrieval, Product Design workflow, Digital shapes, Shape processing tools

1 Introduction

Due to worldwide competition and to recent technological improvements, product time-to-market has been reduced in the last years and specialisation of competencies in the Product Development Process (PDP) has been growing. PDP is a very complex process which requires a variety of expertise according to the peculiar activity considered. Thus, the number of different actors involved in PDP increases, as well as the multiplicity of their specific knowledge areas.

In some cases, companies are becoming Virtual Entreprises (Katzy and Schuh, 1998), i.e. (possibly temporary) networks of independent companies which share their means, competences and resources in order to carry out joint projects which can exceed the capacities of each single unit. Indeed, Virtual Entreprises are generally multi-site, multi-lingual, multi-cultural, multi-function and multi-competence. In car industry, for instance, even traditional companies are moving the range of their main knowledge and competency areas from the complete product design and manufacturing to the design and the assembly of the product only. As an example, the number of car suppliers is increasing, as well as their specific competencies: often, suppliers do not only manufacture a part according to the main company's design, but they also participate in the part specification process.

Due to such a change of mentality in the design activity, to perform an efficient job, companies and actors of PDP need to access the right information in a usable format at the right time. Throughout the PDP, a large number of information suitable for each specific application is available. This information needs to be efficiently and effectively represented and retrieved; thus, a strong interaction among the actors is required to share product data.

Among the proposed information technologies, Electronic Data Interchange (EDI) has been introduced to exchange transaction details electronically between customers and suppliers, such as purchase orders or part specifications (Mukhopadhyay et al., 1995). Alternatively, Knowledge Management (KM) consists of a range of practices used by organisations to identify, create, represent, and distribute knowledge for reuse, awareness and learning (Nonaka and Takeushi, 1995).

Beyond the mere support for collaboration and share, the need for easy, quick and secure access to data during the product design process led to the development of Product Data Management (PDM) systems (Lee, 1999). PDM systems are widely adopted in industry: they combine the scope of technical and engineering data, and support managers in the organisation of the increasing amount of information required by product design, manufacturing, and maintainance. Rooted in CAD (Computer Aided Design) and PDM systems, Product Lifecycle Management (PLM) aims at connecting various product stakeholders over the whole product lifecycle (Amann, 2002) (Antti and Anselmi, 2005) (Stark, 2006) and aims at enabling them to handle company projects.

Unfortunately, commercial PDM and PLM systems are not able to support efficiently the design phase, which is dynamic, as well as the information generated and used in the PDP. Especially in digital mock-up, a specific digital product representation fits each phase. Moreover, specific shape information is relevant for each actor (Gero and Rosenman, 1996), and specific interface workflows between two different actors are needed in order to transform the product from one description to another, taking into account both the geometry and the usage context (Hamri et al., 2006). Commercial PDM and PLM systems are not effective in assisting such a crucial phase, because they are mainly document-oriented and provide a general description of the product with no special customised view. Furthermore, PDM/PLM systems, which rely on Database Management Systems, opportunely structure the information, but fail in encoding the relationships among the engineering data and their evolution in the PDP.

To overcome this lack, we herein present an ontology for product design, the *Product Design Ontology* (PDO). The PDO is centred around shape processing, and its aim is to assist a PDP actor who needs information related to the shapes and tools intervening in the PDP different tasks; in this respect, we formalised the interpretation of the information associated to a shape and the functionalities and usage of shape processing methods.

The PDO was originated in the framework of the Network of Excellence AIM@SHAPE (AIM@SHAPE), whose intent was the integration of research on digital shapes modelling and processing with knowledge technologies and semantic web tools. Among the objectives of the project, there were the definition of a common framework for formalising, processing and sharing shape knowledge, and the set up of a shape-related knowledge base, the Digital Shape Workbench (DSW) to be coupled with efficient communication and collaborative working infrastructures. The DSW permits to store, integrate, adapt, enhance and retrieve shapes and shape processing tools, together with relevant scientific publications. In particular, two Common Ontologies (COs) - one for *shapes* and the other one for *shape processing tools* - were proposed: they allow for respectively a generic (i.e., non-domain related) specification of shape representations and shape processing tools. The COs have been integrated and complemented according to a bottom-up approach, relying on the development of three ontologies, which address the need of particular shape application domains (i.e., Product Design,Virtual Humans, and Shape Acquisition and Processing) (Vasilakis et al., 2005).

Product Design has been selected as one of the shape processing sub-domains of interest because PDP is one of the application contexts where semantic web techniques and ontologies are promising for retrieval and online collaboration support. In particular, ontologies, which formalise a common understanding on symbols and terms used, enable us to share a common vocabulary, thus to support communication and transfer of knowledge among different product stakeholders (Giménez et al. 2008). Indeed, the scientific literature reports on several proposals where ontologies are applied for the formalisation of CAD/PDM/PLM knowledge (Horvàth et al., 1998) (Jun et al., 2005) (Kitamura and Mizoguchi, 2004) (Kopena and Regli, 2003) (Li et al., 2005) (Patil et al., 2005) (Posada et al., 2006). In particular, (Horvàth et al., 1998) presented an ontology-based approach towards the definition of design features, while (Kitamura and Mizoguchi, 2004) and (Kopena and Regli, 2003) investigated the modelling of functional knowledge for Product Design. In (Li et al., 2005), the authors propose the application of a knowledge-based formalisation of conceptual design know-how and intentions to improve the design retrieval and reuse. In (Posada et al., 2006), the authors use an ontology to support the semantics-driven simplification of CAD models applied to the visualisation and the design review of large plant models. In (Giménez et al. 2008), the authors propose a product ontology to share information on complex product models, defined according to different levels of detail, related to both the structure (with respect to a bill of material decomposition) and the abstraction (i.e., Family, Variant Set, Product). Another work addressing the exchange and reuse of product models knowledge is (Gupta and Gurumoorthy, 2008), where the Domain Independent Form Features (DIFF) Model, describing the operational semantics of features, is used to define an ontology of form features.

Since the last ten years the National Institute of Standards and Technologies (NIST), has worked on a Core Product Model (CPM) (Fenves et al. 2008) to support intelligent design repositories for sharing and reusing the knowledge on design artifacts, focussing on the aspects of *form, function* and *behaviour*. More recently, the University of Missouri-Rolla (UMR) followed the same approach to enhance their design repository (Bohm et al. in press). Our work is complementary with respect to the CPM because we give a geometric description of shape models, and we investigate the aspects of behaviour according to particular subprocess of the PDP. The traditional functional knowledge, as addressed by (Fenves et al. 2008), may be easily integrated in the PDO to complement the product description.

However, the interest of the research community in the integration of product semantics using knowledge technologies does not usually focus on the annotation and retrieval of shape data as addressed in this paper. Moreover, even if the general approach we adopted is applicable to the whole PDP, we especially investigated the requirements of freeform modelling and engineering simulation, proposing a very detailed representation of the knowledge entailed by the related design workflows, and ad hoc solutions for their enhancement. In particular, the PDO addresses two general scenarios:

1) the user, asking queries either to the DSW or to any repository, retrieves application-driven information about shape processing methods and tools according to his/her point of view on the product; besides, he/she may access to suitable test shapes for benchmarking his/her shape processing tools;

2) the user may ask for a pipeline of tools with a specific functionality; besides, a software developer may benchmark his/her own tool searching for tools that perform the same tasks.

The benefits of the use of ontologies for these scenarios are twofold. On the one hand, they are able to make the communication among researchers working in different areas or addressing different aspects of PDP straightforward; in addition, the development of further software is simplified because developers may refer to the well-formalised Web Ontology Language (OWL) syntax we used for defining the PDO. On the other hand, since the formalisation we propose was designed in collaboration with experts in the field, the proposed terminology and processes are trustworthy.

Although this work was conceived in a research context, its scope may be extended to industrial applications. In fact, the PDO, queried according to the former scenarios, recalls PDM capabilities which are strictly focussed on the digital process of a product. Furthermore, it can assist the training of unexperienced engineers or new partners of a company, who need to learn commonly used design workflows in the PDP, as well as the particular procedures adopted in a company.

Finally, we point out that the work presented in this paper does not contrast with initiatives for the development of workflow languages, e.g., (YAWL) (XFlow), workflow patterns (Workflow Pattern), and workflow ontologies as that proposed in (OWL-WS). The methodologies and the software solutions they provide can be applied to handle the general aspects and the execution of PDP tasks. Therefore, they could be applied to complement this work. By contrast, the PDO supports and enhances the specific domain requirements of researchers dealing with design workflows. These workflows are executed by developing the two investigated PDP tasks.

The paper is organised as follows. In Section 2, a description of the subset of the PDP we selected as application context is provided and three usage scenarios are detailed. In Section 3, we describe the development of the PDO by detailing the main concepts and relationships related to the representation and the exploitation of workflows within the Product Development Process. In Section 4 a significant validation scenario for the PDO is discussed, while Section 5 concludes the paper with some final remarks.

2 Product design user scenarios

The Product Design Process formalised in the PDO is consistent with the Application Activity Model (AAM) of AP214 of the ISO Standard 10303, STEP – Standard for the Exchange of Product Information (STEP); therefore it provides an agreed model of the Product Design workflow (see Fig. 1). The essential characteristics of a new product to be put on the market are expressed in terms of functions, size, materials, weight, etc (*T1_Product_Definition*). Starting from them, designers work on the creation of the object shape (*T2_Product Styling*). When the final shape has been defined, a first three dimensional (3D) digital model is built. Then, it is translated into a more detailed and precise one, which is useful for the engineering phase of the PDP (all these stages are included in *T3_Product_Design*). In particular, the evaluation of the product from the engineering point of view is performed transforming the model in a representation suitable to the execution of an engineering analysis (i.e., Finite Element mesh). It often happens that modifications on the original model are required because of unsatisfactory results (*T3_4_Calculation and Analysis*).

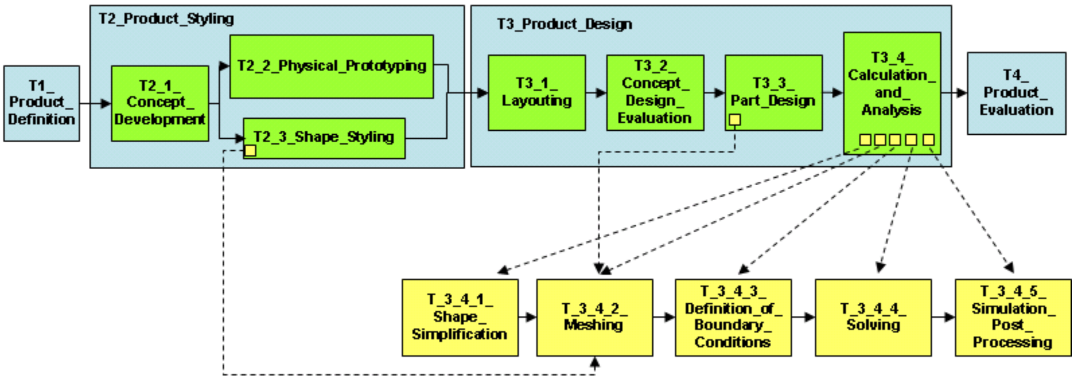


Fig. 1. Product Design Flow as modelled in the ISO 10303, AP214 (STEP)

In the following, we describe the most relevant activities of the Product Design Process that drove the PDO design, i.e., freeform modelling and engineering analysis, focussing in particular on three different user scenarios. The scenario settled for freeform modelling concentrates on the deformation of freeform surfaces represented with NURBS (Non-Uniform Rational B-Splines). It includes quality checks of the digital model to provide adequate data for down-stream applications. Feature-based modelling has been also taken into account.

By contrast, two scenarios of engineering analysis are considered: the first addresses the preparation of a design model to perform a behaviour simulation through a Finite Element Analysis (FEA), including the process of shape simplification; the second deals with the simulation post-processing, where simulation results are examined and interpreted.

Finally, the formalisation of grouping mechanisms is discussed apart. From the analysis of these scenarios, we derived meaningful Competency Questions (CQs), which guided the design of the PDO.

2.1 Freeform modelling

Based on the product definition, the first step of the product design is the creation of a new object taking into account functional and aesthetic characteristics. Stylists translate such constraints into some alternatives shapes, which can be very complex from the geometric point of view thanks to the advanced manufacturing technologies. This requires the employment of freeform modelling techniques, typically working on the parametric NURBS representation for curves and surfaces to be compatible with down-stream applications. The scenario we consider mainly addresses developers of freeform deformation tools who have to test their own tool with several suitable models and to compare it with the capabilities and results of other existing tools. In this case, the connection with the DSW is very strict: in this repository, in fact, it is possible to search for NURBS models enriched with significant geometric metadata, and software tools with specific functionalities. Similarly, the same scenario can be envisaged in an industrial context, where a CAD user searches for past projects in the company repository, while the formalisation of the early design workflow can be used for training purposes.

In Fig. 2, the typical steps of the freeform deformation scenario are shown. After searching for suitable models (Fig. 2(a)), possibly belonging to some specific industrial category, the first action is verifying if the properties of the model fit the input requirements of the modelling tool to be tested (Fig. 2(b)); in case they did not, the model has to be corrected (Fig. 2(c)). Then, the software can be run (Fig. 2(d)) and the results evaluated (Fig. 2(e)).

For instance, a good connectivity between patches is commonly required in CAS/CAD applications. The deformation tool under test would reasonably require a smooth surface in input and should provide a smooth surface in output. The input model selected in the repository has to fulfil such a property. Specific evaluation tools –which can be retrieved also in the DSW– permit to check the regularity of the shape, detecting the areas where geometric constraints fail and have to be re-established; they can be used before and after the deformation process.

Examples of CQs associated to this scenario are: *search for a design model represented as a BRep with NURBS patches and belonging to the automotive category; search for tools that check the connectivity between patches of the design model.*

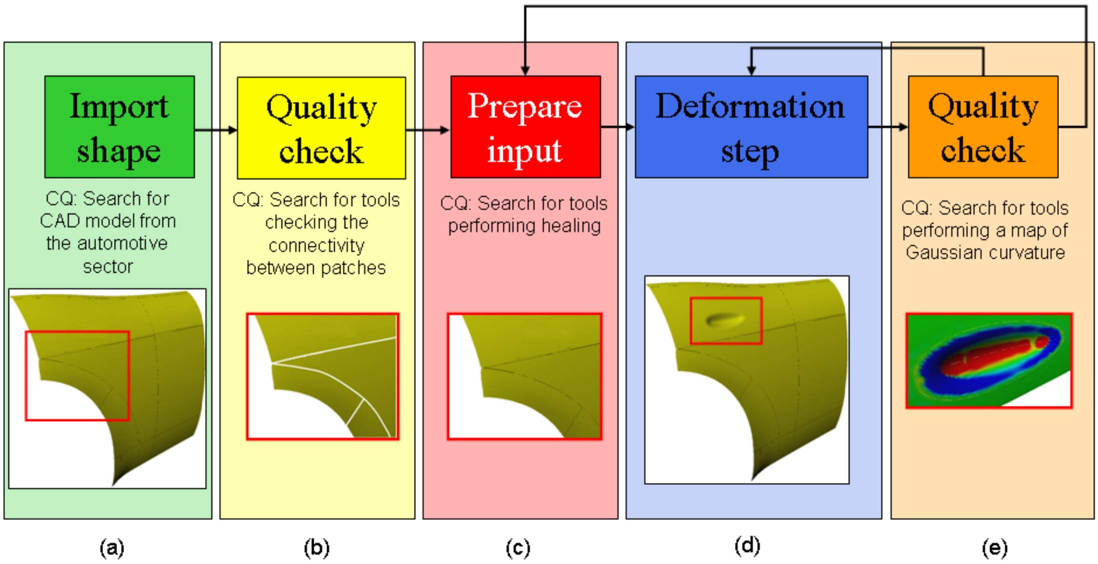


Fig. 2. Description of the scenario for the freeform modelling task.

Quality check of digital models is a crucial task when transferring from one system to another: due to the different tolerances used, a CAD model can be invalidated and then become useless in the downstream processes. As a consequence, quality checks occurring more frequently in some type of models transfer have been identified and included in this scenario. We consider in particular: checks for avoiding non-manifold models (within specified tolerances), which are important for all the transfers (e.g., check for duplication of elements; check for self-intersections and singularities); checks for preventing configurations that make downstream applications difficult, especially for geometric computation (e.g., tests for mini-elements; tests for edges lops and for the orientation of the surface normal).

In this context, examples of CQs that can be successfully answered are: *find software tools that support the specific quality check of self-intersection; find the types of geometric conditions that are necessary for a FEA; find the types of checks to consider when performing the meshing task.*

Finally, the CAD models we refer to in our PDO may reasonably contain not only geometry, but also some well-defined form features, whose number and type is a further indication of the complexity of the shapes. In fact, traditional 3D shape representations are generally able to support only geometric information; thus, features were introduced in product design as a modelling approach aiming at bringing a specific kind of semantics in geometry (Shah and Mantyla, 1995). They carry a different semantics according to the area of process under observation; in particular, form features are strictly connected to the shape of the object. The benefits of feature-based modelling have been well recognised and CAD systems largely adopt it both in the mechanical and freeform domains.

2.2 Engineering Analysis

During the PDP, the simulation stage evaluates the physical behaviour of any engineering component constituting the whole product, which is subject to various kinds of loads and conditions. Finite Element (FE) approaches are widespread in industry to analyse the mechanical behaviour of a component. They apply mechanical models of behaviour on a appropriate model, i.e., a FE mesh. A FE mesh represents the discretisation of a continuous geometric model, and is obtained by decomposing it into small elements having a simple but arbitrary size (the finite elements).

FEA ensures that design requirements are feasible. It can be useful both when designing a new product before its manufacturing and when refining an existing product, e.g., when creating a product design variant or when the original one is subjected to new loading conditions. In case of analysis failure, FEA may be used to determine which design modifications better meet the mechanical conditions.

A typical shape evolution cycle for FEA is illustrated in Fig. 3. The CAD model used to design the product (Fig. 3(a)) is usually represented by parametric surfaces, which are suitable for manufacturing purposes, but not for performing a FEA. Therefore, the initial design model generated by a CAD system needs to be converted into a FE mesh.

Actually, the FE mesh is generated on a simplified model of the component (Fig. 3(b)). In fact, the initial one usually contains several shape details which complicate and could compromise the FE mesh generation and solving. Therefore, several steps of shape adaptation are required, where the shape details (e.g., form features) considered as not influent on the FEA have to be removed. However, shape simplifications performed on the design model could affect the accuracy of FE results. Their choice and evaluation have to be carefully analysed and require significant expert knowledge.

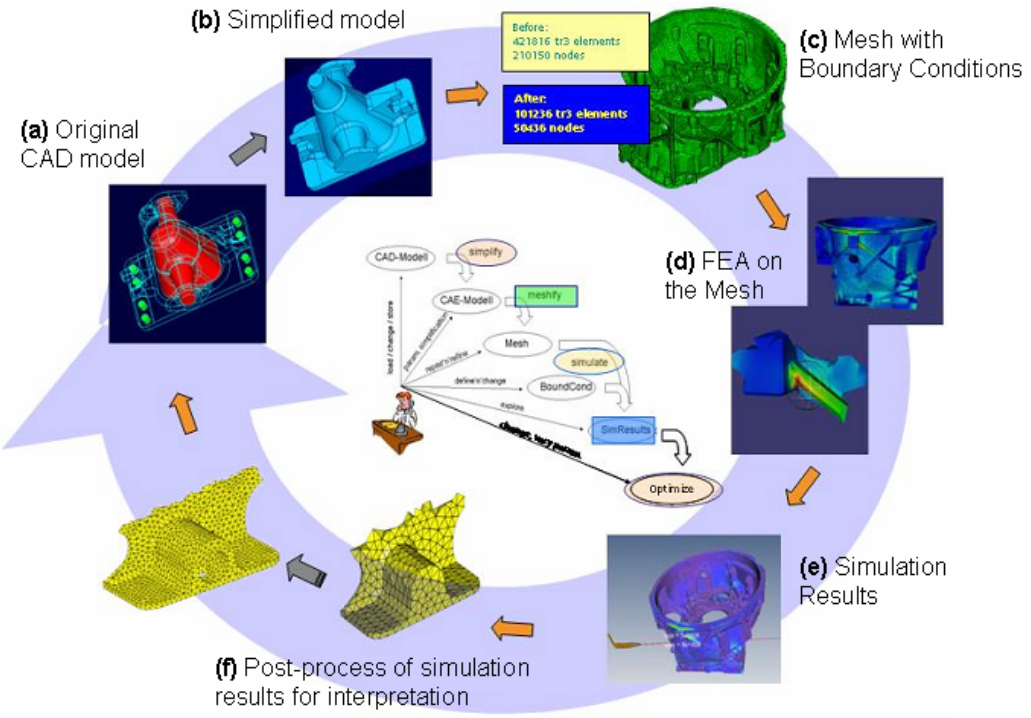


Fig. 3. Shape life-cycle in a typical FEA design evolution loop.

Once the FE mesh is created on the simplified shape, boundary conditions are added (Fig. 3(c)). They describe the interactions of the component with its neighbouring environment (e.g., forces, displacements, temperatures over defined model's areas) and are related to the specific mechanical problem to be studied, e.g., structural mechanical, electromagnetic, thermal analysis.

Then, the FEA is executed (Fig. 3(d)). It returns information about the behaviour of the component when subjected to the loads and constraints specified by the boundary conditions (Fig. 3(e)). The kind of information included in the simulation results depends on the type of physical problem that has been simulated. For example, a thermal analysis can return information about temperatures, while a structural mechanical analysis can provide a stress field as a possible result.

To address engineering analysis in the PDO, we considered two different user scenarios. The first focuses on the removal of shape details when generating a simplified version of the design model. A user could be interested in testing tools, able either to suppress shape details during the simplification of the initial design model or analyse the influence of the performed shape simplification on FE simulation results.

The second scenario is related to the post-processing stage. Indeed, behaviour simulation results need to be post-processed for interpreting the simulation outcome and making decisions about the suitability of the design with respect to its engineering specification. The first post-processing phase is a check on the occurrence of numerical problems during the solving. If they did not occur, the entities of interest may be examined thanks to interactive visualisation techniques supporting the FEA.

Similarly to the freeform scenario, a user working in the area of FEA may seek for relevant shapes, possibly real-world test data contained in the DSW. In addition, searching for alternative tools is useful for benchmarking in both pre- and post-simulation scenarios. In an industrial context, these scenarios can be interpreted as the retrieval of past successful company projects and the analysis of such projects for further reuse or training.

Examples of CQs related to the engineering analysis scenarios are: *find the tools that perform Meshing and have as input a design model with format STL and have as output a Volume Mesh; find the models suited for a FEA; find models that are simulation results of a thermal analysis applied to a cylinder head*.

2.3 Grouping mechanisms

Several shape grouping mechanisms are adopted in Product Design. Gathering related models can be useful for their further reuse; for example, it is very common:

- <u>Grouping shapes belonging to the same product category</u>

Still in the design phase, a new product is usually created drawing inspiration from or even reusing past design. In this case, a user needs to retrieve product models belonging to a particular category, e.g., the car light box.

- <u>Grouping shapes belonging to the same assembly</u>

Complex products are not designed as single models: engineers, especially in the automotive and aerospace industry, rather separately design the different parts and assemble them in a second phase.

- <u>Grouping shapes representing the same object in different representations and different formats</u>

Generally, this kind of grouping is useful when passing from one PDP stage to another, where models need to be transferred to other systems or converted into other geometric representations.

- <u>Grouping shapes belonging to specific chains of shape processing operations</u>

In PDP, typical sequences of operations can be identified when performing certain tasks. Consequently, a shape workflow can be recognised, where one shape derives from the previous one after the application of suitable tools. A user may need to retrieve such workflows to interpret and better evaluate the shape under examination. In particular, models of a specific workflow can be searched for; it can be also useful to find out if a given model belongs to different workflows. For example, the grouping of different simplified versions of a model needed during a FEA preparation can support an engineer in a post-processing task.

According to the identified categories of groups, here is a list of CQs that imply the use of some notion of group: *does a given model belong to an analysis group? does it belong to an assembly? does it have a design variant in the repository? find all the elements of a given group*.

3 PDO: The Product Design Ontology

The PDO was modelled according to On-To-Knowledge (Sure et al., 2003), a widely used methodology for ontology development which, at the time of the project, was optimal in terms of power of capturing knowledge domain and simpifying the process of ontology formation.

Few methodologies tailored to product design have emerged recently; for instance, in (Nanda et al, 2006) a formal methodology has been proposed, which, whilst is well structured, is very targeted towards product families rather than shape processing in product design.

On-To-Knowledge does not refer to a specific domain, but may be classified as an application-dependent methodology (Corcho et al., 2003). It defines the ontology design and deployment as an interative process including four phases: kick-off, refinement, evaluation, and maintainance and evolution.

During the first step of the *kick-off phase*, the key concepts and their mutual relations were identified, together with the usage scenarios we described in Section 2, knowledge sources (domain experts, glossaries and dictionaries, etc.), potential users, and the competency questions the ontology would have answered. In the second step, a semi-formal hierarchy of concepts and relations was built by applying a middle-out approach: the most important concepts were identified, while the others were defined either by their specialisation or generalisation. This a very natural approach, which permits to focus on the core concepts and select the level of detail desired for the formalisation.

In the *refinement phase*, we modelled the formal product design ontology using Protégé (Protégé) in combination with the corresponding OWL plug-in. We chose OWL to model the PDO because, differently from other technologies, e.g., the Resource Description Framework (RDF), it provides an improved expressive power for representing (formal) machine-interpretable semantic content. Hence, it allows to represent effectively relationships and modelling constraints among the conceptual entities that describe the Product Design domain. In particular, the use of OWL allowed us to model cardinality restrictions on data properties and subproperties, as well as to define further inference rules on relations. For instance, the object properties *hasPredecessor* and *hasSubTask* of the concept *Task* in Fig. 4 are declared as transitive properties.

During the *evaluation phase*, the ontology was populated and constantly validated to measure its compliance with the initial expectations. Usage scenarios, requirements specifications and CQs were used as a reference for the evaluation.

Finally, the *mantainance phase* took into account the evolution of the ontology according to refined application scenarios.

Guarino proposed in (Guarino, 1998) to develop different types of ontologies to capture knowledge at various levels of detail. In accordance with this work, the two Common Ontologies from the AIM@SHAPE network (cf. Section 1) may be referred to as domain ontologies. In fact, the COs model the generic concepts of *shape representation* and *shape processing tool*, which strongly intervene in the previously described scenarios. The PDO specialises those concepts according to the specific point of view of PDP actors, thus can be seen as an application ontology in the Guarino's classification.

The modelling process of the PDO we described above is also aligned with the specification, conceptualisation, formalisation, implementation and evaluation phases specified by METHONTOLOGY (Fernandez, 1997), even if it does not rely on application scenarios.

The core components of the PDO are (see Fig. 4):

- **shape types** and **shape representations** with product design related metadata;
- **shape processing tools** and algorithms employed in the product design context;
- **tasks** accomplished by software tools applied to shapes, which are both data sources and results of product design tasks along the product design flow;

- **conditions**, either geometric or not, which are related to a shape model when performing a specific task;
- **groups**, which permit to gather together digital models that share some interrelation.

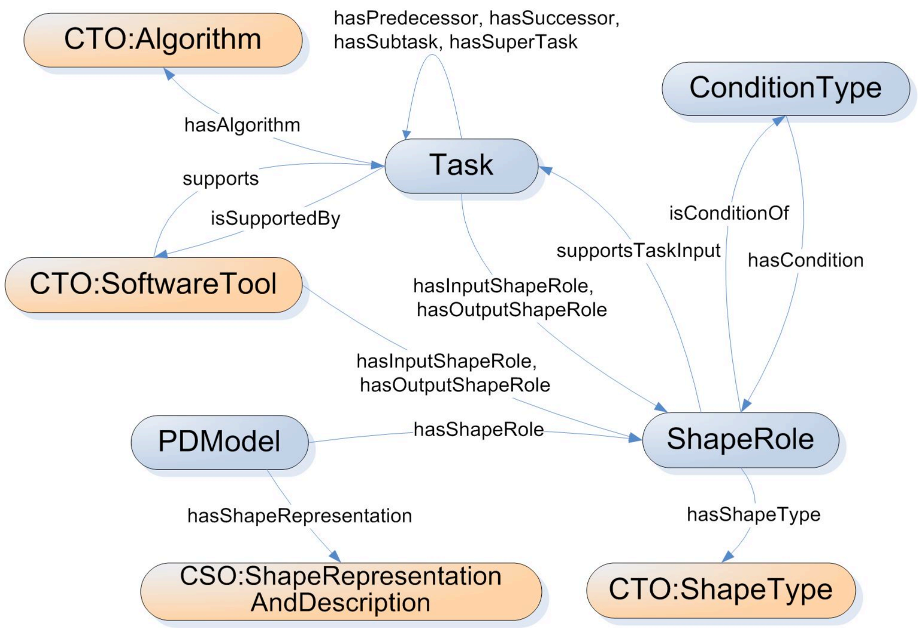


Fig. 4. PDO core concepts

In the following, we focus on the PDO peculiarities, and the choices related to the ontology design are discussed with respect to the specific requirements of the freeform and analysis scenarios. In particular, the different tasks involved in the Product Design workflow have been formalised, together with the different roles a shape can play within the shape life-cycle (e.g., the input and output of the simulation task are surface or volume meshes enriched with proper simulation data and properties), and the requirements and conditions a shape must or should satisfy to fulfil a given task.

3.1 The Task Concept

The *Task* concept (see Fig. 4) is the central unit of any pipeline within the PDP and is strictly connected with shapes and tools. Each instance of *Task* represents an activity in the PDP, which involves the application of shape processing tools on some shape model. An example of task is *simulation post-processing*, as discussed in Section 2.

Task is provided with four auto-relations to model any sequence of tasks constituting a specific process or process step in the PDP. The link with software tools is modelled with specific relations, which enable the answer to some of the competency questions discussed in the previous section (e.g., *search for tools that check the connectivity between patches of the CAD model*). Note that we are able to distinguish among the different functionalities a tool provides, thus retrieving categories of tools, e.g., freeform modelling tools.

3.2 The role of a shape: ShapeRole and PDModel

The scenarios described in Section 3 demonstrate that, when performing a specific activity of the PDP, i.e., a task, the digital representation of the shape is always equipped with additional context-dependent information. The concepts of *ShapeRole* and *PDModel* have been introduced to model respectively shape types and shape representations, enhanced with extra data, for instance regarding conditions to be applied on the shape to fulfil the task activity.

The *ShapeRole* concept enriches shape types description (e.g., volume mesh, BRep) with additional information intervening in a specific task of the Product Design workflow. It is particularly useful for inexperienced users, because it provides the user with all the expertise necessary to execute the whole development process. For each task in the PDP, the user can learn about: the shape type required; the conditions to be verified in order to complete it; the type of results returned and the additional information provided. In particular, *ShapeRole* gives the input and the output shape type for a task. Moreover, it enables the retrievement of the domain knowledge about specific shape processing workflows, answering CQs like: *find the types of model needed as input of a simplification task*.

Fig. 5 shows the complete *ShapeRole* taxonomy provided by the current version of the PDO. Two direct implementations of *ShapeRole* are included: *SimplificationModelRole* and *FiniteElementMesh*. These concepts are employed by the simplification and FEA tasks, respectively. The consistency of the different roles has been maintained: for example, a *FiniteElementMesh* is restricted to have a surface of volume mesh as a shape type, together with suitable boundary conditions. The concept *FiniteElementMesh* is further specialised through the concepts *PreSimulationMesh* and *PostSimulationMesh* in order to distinguish between the two different roles a mesh can play in the context of simulation.

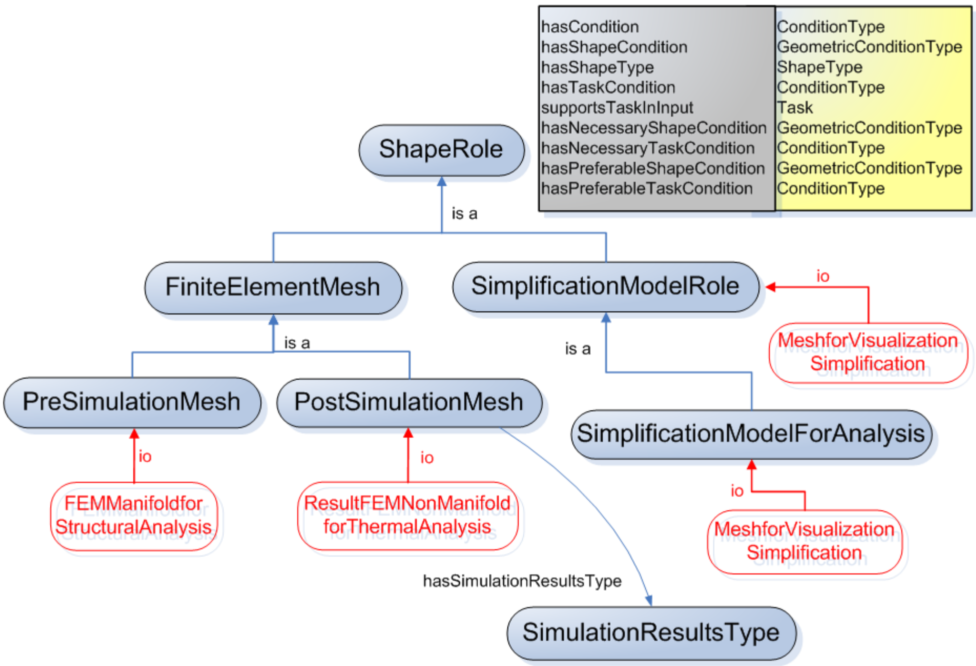


Fig. 5. The ShapeRole taxonomy.

Differently from *ShapeRole*, which models the role of a given shape type, the *PDModel* (Product Design Model) concept (see Fig. 6) has been introduced to model the role of a specific shape model. In fact, while the concept of shape role is useful to find general pipelines in the PDP, the *PDModel* permits to retrieve the flow of a single shape. Thus, it assists the benchmarking and testing activities of specialised researchers on the one hand, and, on the other hand, it provides engineers with the histories of specific digital products. Consistently, a *PDModel* also includes the information related to the corresponding shape role.

PDModel is currently implemented through the concepts *SimplificationModel*, representing the models occurring in simplification, and *SimulationModel*, employed in FEA. The corresponding roles are *SimplificationModelRole* and *FiniteElementMesh*, respectively (cf. Fig. 5).

A *PDModel* includes also the information about the CAD features occurring in the model representation. Indeed, when retrieving CAD models as test data, the presence of metadata about features is a clue to the complexity of the model. The occurrence of each type of feature, e.g., hole and blend, is reported (Fig. 6).

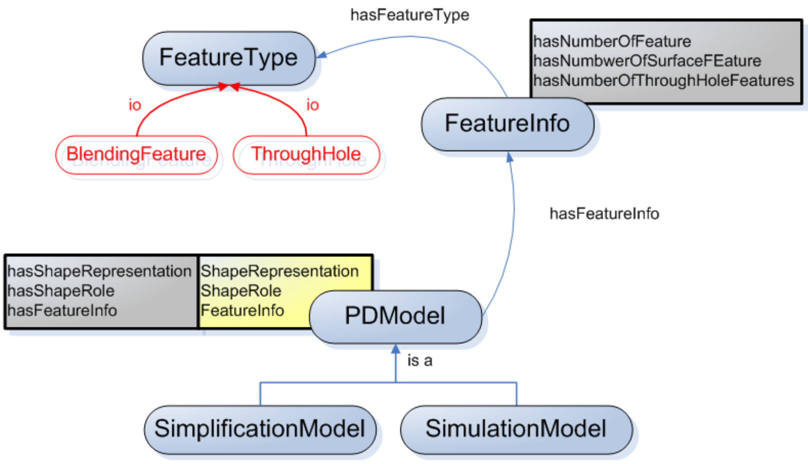


Fig. 6. The PDModel taxonomy and some of its relations

3.3 The Condition Type

As mentioned above, the concept *ShapeRole* specialises shape types related to a specific task of the Product Design workflow. This specification includes information about the conditions a shape of the corresponding type has to satisfy in order to perform that task, hence modelled through the concept *ConditionType* (Fig. 7). The introduction of the *ConditionType* class makes it possible to answer to competency questions like: *find the types of geometric conditions that are necessary for a FEA*.

Several types of conditions can be used for enriching a shape, that is, characterising a shape role. We concentrated on geometric conditions, which have been further specialised to distinguish the geometric properties applying to different shape types (so far, BRep and Mesh), and boundary conditions, which are associated to a mesh during the analysis stage (see Fig. 7).

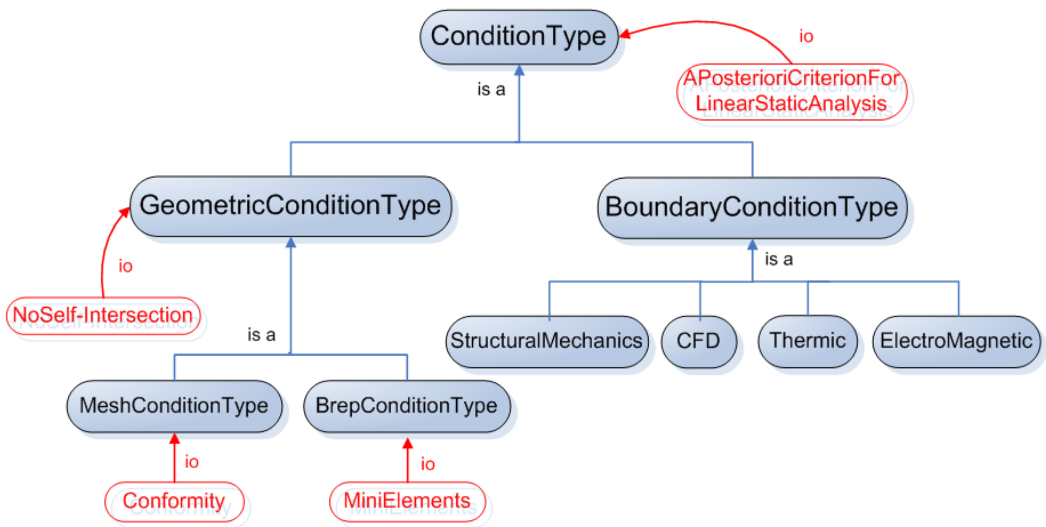


Fig. 7. ConditionType taxonomy

ShapeRole and *ConditionType* are related through the *ShapeRole* property *hasCondition*, referring to the geometric conditions that a shape with the given *ShapeRole* must or should verify. This property is further specialised through sub-properties (see Fig. 8) to distinguish between conditions specifically related to the shape type (e.g., conformity, a condition required to a mesh when its shape role is *FiniteElementMesh*) and to the task while performing it. The conditions related to the task can be either geometric (e.g., edge size), as in case of simplification tasks, or not (e.g., boundary conditions), as it happens, for instance, when performing simulation. Moreover, a further distinction is made to discriminate between conditions that are mandatory for a given role and conditions whose verification is suggested (namely, necessary and preferable conditions, respectively).

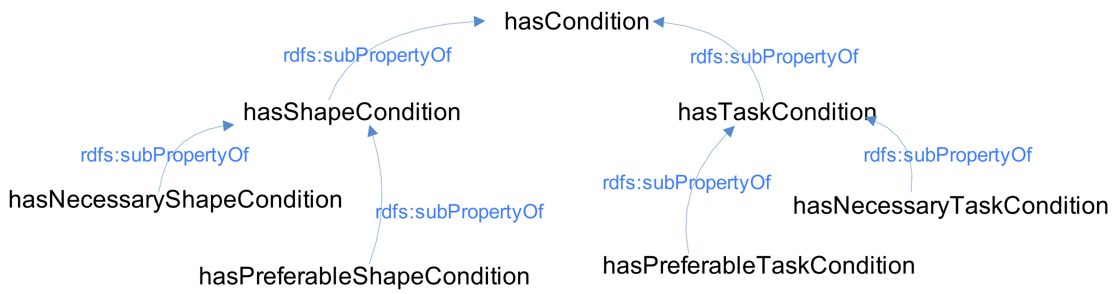


Fig. 8. Relation hasCondition and its subproperties

3.4 Modelling groups

The modelling of the *Group* class enables the creation of a relation between product models of a repository, which have not been stored together. The property *hasElement* has values in *PDModel* and is invertible to answer to CQs asking for retrieving groups for a given element, like: *find the simulation model whose role is PreSimulationMesh and that belongs to an analysis group also containing the post-simulation model of the given one*.

The *Group* class has the subclasses *AssemblyGroup*, *VariantGroup*, *SimplificationGroup*, and *AnalysisGroup*, which correspond to the categories of the groups we introduced in section 2.3 (see Fig. 9).

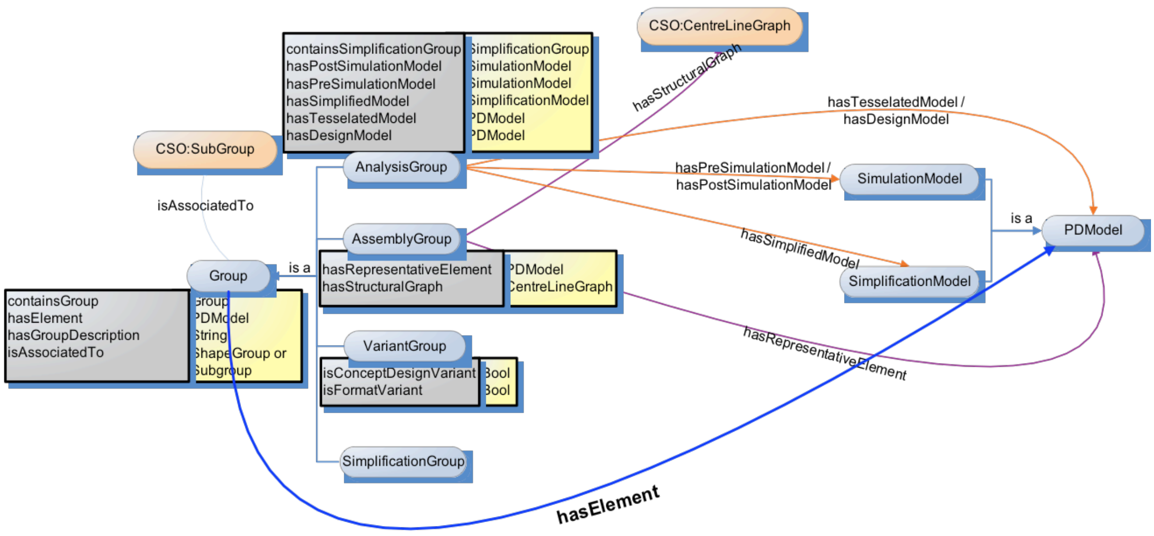


Fig. 9. Groups in the PDO

The *AssemblyGroup* has the restriction that any subgroup contained is an assembly. It is also possible to specify the graph structuring the assembly components through the relation *hasStructuralGraph*. This allows the user to manage complex assemblies and easily retrieve information about the interrelationships among the assembly components.

The *VariantGroup* includes few variants of the same model; in particular, we took into account the concept design variants and format variants.

The *SimplificationGroup* concept specialises the general *Group* class with the constraint of containing at least one *SimplificationModel*. A *SimplificationGroup* gives the possibility of collecting both different simplifications of the same model and models having a different *ShapeRole*. Indeed, the group contains models linked together in a simplification workflow, no matter what their shape role is, as required by the identified user scenarios.

Finally, the *AnalysisGroup* aims at structuring the different *PDModels* related to the simulation phase, requiring that at least one *SimulationModel* belongs to such group. The *AnalysisGroup* specialises the relation *hasElement* with the subproperties *hasDesignModel*, *hasTessellatedModel*, *hasSimplificationModel*, *hasPreSimulationMesh*, *hasPostSimulationMesh*, so that the shape workflow is implicitly modelled and its retrieval becomes easy for the user, once he/she browses the group.

4 The PDO in use

The structure of the PDO has been validated as part of the DSW of the AIM@SHAPE Network of Excellence, both evaluating its semantic correctness and verifying its effectiveness. The first assessment has been done by the researchers involved in the project, who also tested the effectiveness in answering all the CQs of the selected scenarios through the project web portal (AIM@SHAPE).

For this purpose, an on-line Semantic Search Engine (SSE) has been developed within the network (see Fig. 10), providing the means for intuitive search, without dropping the flexibility or expressiveness of the queries. This is accomplished by letting the user interact with the underlying knowledge base rather than simply querying, by using both a graphical user interface and the natural language. The semantic-based natural language query interface of the SSE is able to analyse semantically queries using ontologies and to provide adequate results. In Fig. 10 the user submits the natural languange query *Find a simulation model which role is a post simulation mesh.* Fig. 11 shows the corresponding results, indicating that there are some instances of the concept *SimulationModel* that have the link *hasShapeRole* to some instances of the concept *ShapeRole*.

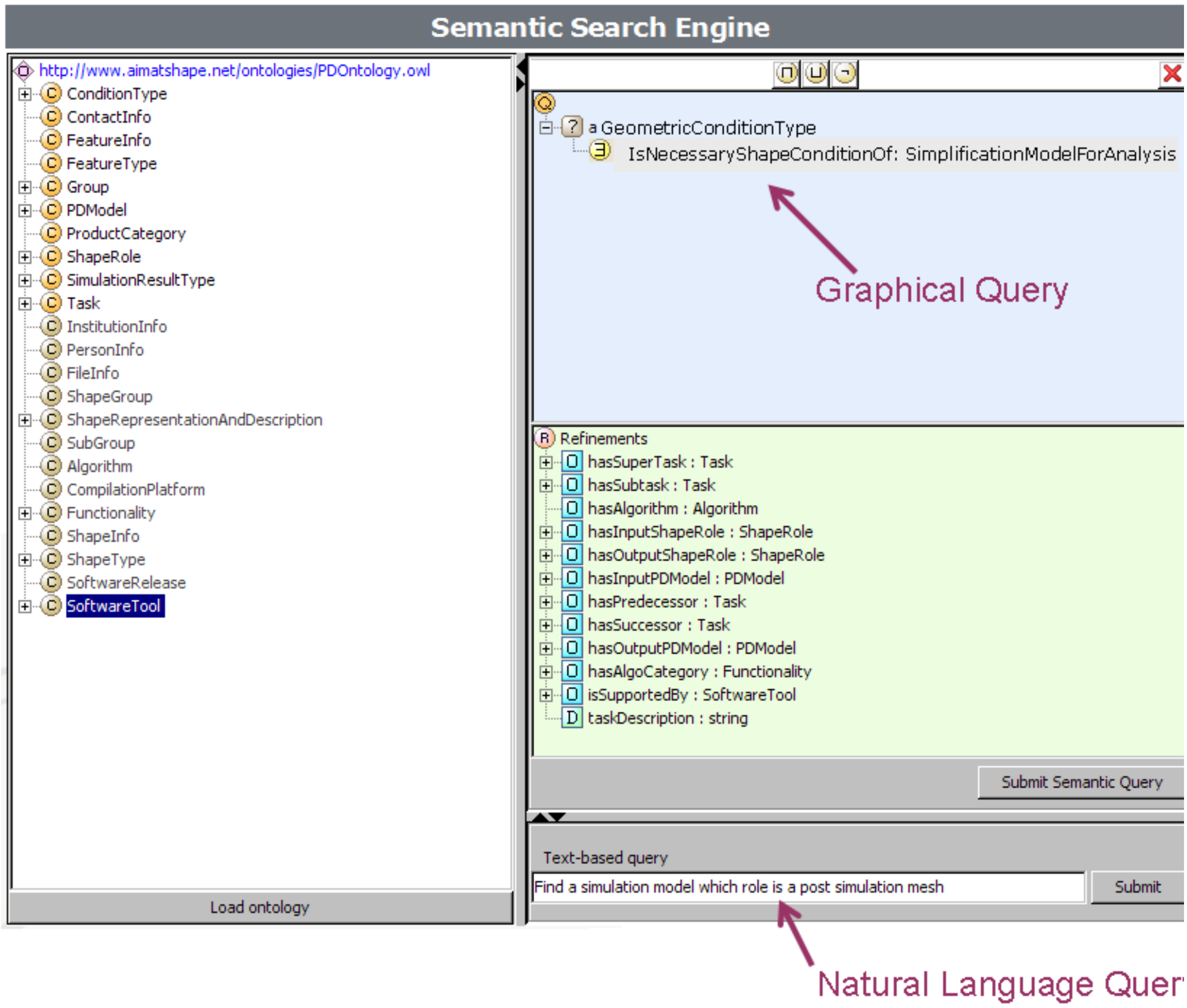
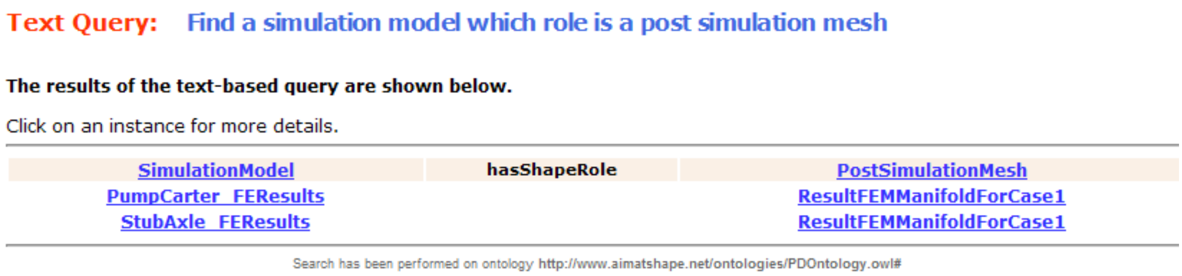


Fig. 10. AIM@SHAPE Semantic Search Engine


Text Query: Find a simulation model which role is a post simulation mesh

The results of the text-based query are shown below.

Click on an instance for more details.

SimulationModel	hasShapeRole	PostSimulationMesh
PumpCarter_FEResults		ResultFEMManifoldForCase1
StubAxle_FEResults		ResultFEMManifoldForCase1

Search has been performed on ontology http://www.aimatshape.net/ontologies/PDOntology.owl#

Fig. 11. Results of the natural language query

The structure of the PDO can also be navigated through a hyperbolic tree. Such viewer allows the user to familiarise with the ontology showing only the class herarchy. Moreover, it is possible to browse directly the metadata that are associated to the ontology instances. Figure 12 shows an example of metadata browsing, where the user begins his/her navigation by searching for an instance of analysis group and then, starting from the obtained results, goes on looking for the metadata related to it.

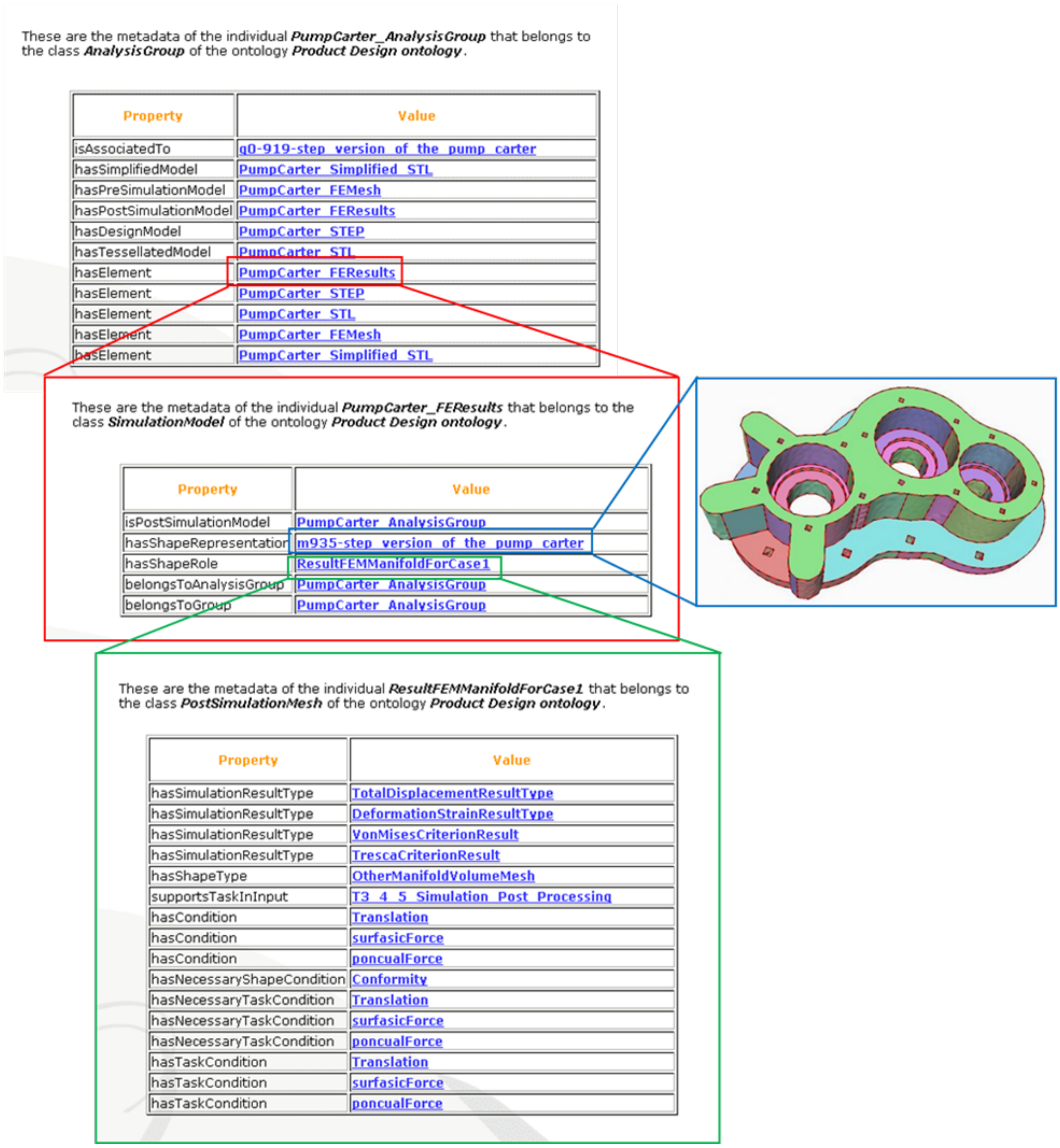


Fig. 12. An example of metadata navigation

Besides simple searches and navigation through ontology classes and metadata, the user may use the SSE in order to ask precise queries to the PDO. In Fig. 13, a typical usage scenario of the PDO within the DSW is shown; it is related to the task *Calculation and Analysis*, further specialised in several subtasks describing the different simulation activities (e.g., *Shape Simplification*, *Meshing*, *Definition of Boundary Conditions*).

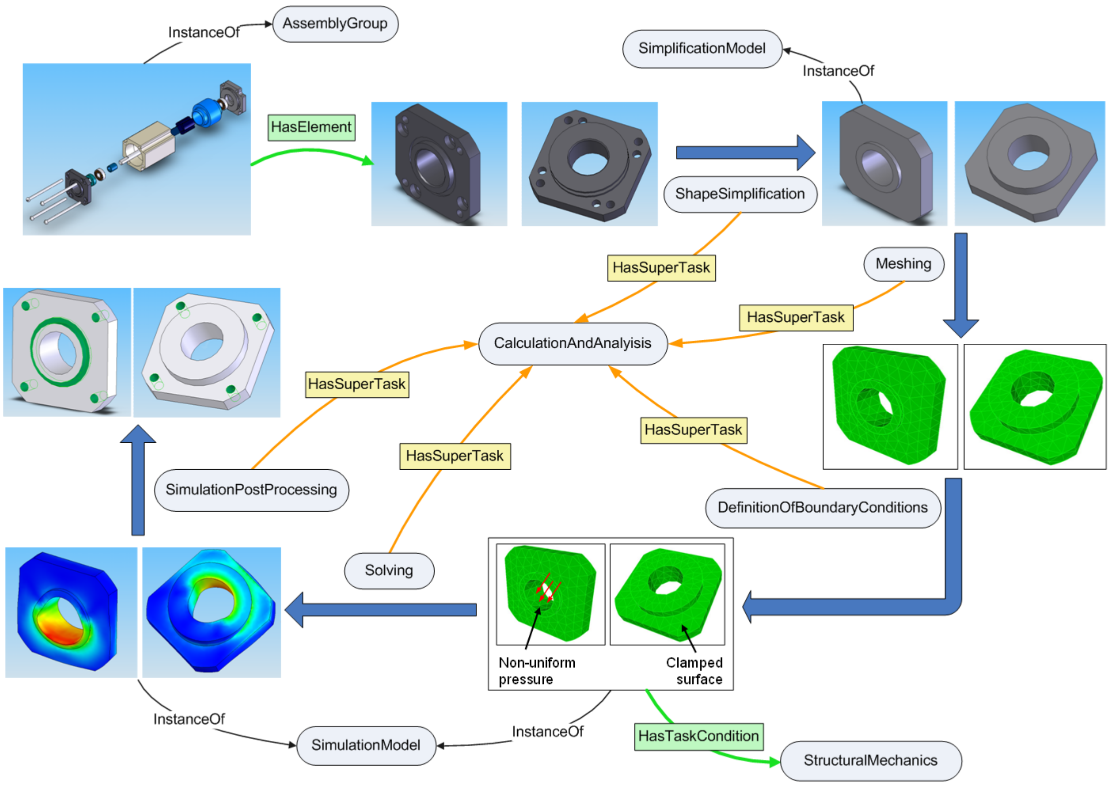


Fig. 13. An example of simulation workflow

A mechanical assembly is selected from the shape repository to perform a structural simulation on one of its components. The mechanical component changes role according to the specific subtask, which constrains the conditions on the shape data to be verified. At the beginning of the workflow, it is an instance of the concept *SimulationModel*, which is associated with a BRep model in the shape repository; it has the role of *SimplificationModel* because some details have to be removed before the *Meshing* task. After meshing the geometric model, the role of the mechanical component becomes *PreSimulationMesh*, because it has to be subjected to a simulation process.

Knowledge about the workflow can be gathered both by browsing metadata, as showed in Fig.12, and by querying the semantic search engine. As an example, the mandatory geometric conditions to perform a FEA may be asked to verify if the mesh obtained is suitable. Fig. 14 shows the answer of the search engine to the CQ "*which types of geometric conditions are necessary for a simplification model for analysis?"*. If the user wants to specialise more his/her query on some constraints that have to be respected during the simplification process, he/she can browse the instances of the *ShapeRole SimplificationModelForAnaysis*: for *SimplificationMeshwithStaticBC*, he/she will find that also *AxisSymmetry*, *Translation* and *VolumetricForce* are necessary task conditions.

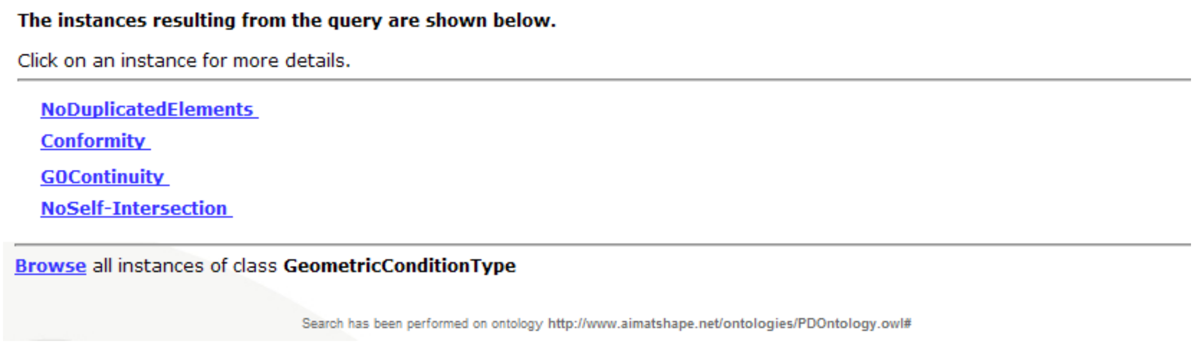


Fig.14. Results of the graphical query

According to the kind of simulation, specific boundary conditions have to be applied in the task *DefinitionOfBoundaryConditions*, which are instances of *BoundaryConditionType* (*StructuralMechanics*, in this case). After performing the simulation, the output shape will be an instance of the *SimulationMesh* with the role of *PostSimulationMesh*: this implies that the simulation results are associated to the geometric part in order to interpret the simulation outcome and to make decisions about the design suitability with respect to its engineering specification.

5 Conclusions

In this paper, an ontology-based formalisation of the Product Design Process has been presented. Herein we focussed on the Product Design workflow, introducing the role of shapes along such workflow and the geometric and the task-oriented conditions involved in the process. The formalisation also includes the representation of the functionality of shape processing methods.

The main effort of this work has been put into designing a flexible and rich ontology scheme, capturing and sharing the knowledge related to the digital shape workflow that is typical of the first phases of PDP.

Although the PDO target was initially the research community, we believe that such a formalisation can be also reused and extended according to industrial requirements, coupling it with the project repository of the company. In fact, an effective retrieval of past solutions and the training of company-specific processes are definitely a key issue of the digital product design phase.

The main limitation of this work is that the semantic annotation required to populate the ontology is performed manually; thus, enriching the model repository becomes an expensive activity. This aspect is even more critical in an industrial context, where past projects are numerous and their storage is different according to the company policy, possibly changing over the years.

Automatic annotation of resources is related to the development of advanced tools and search engines able to deal with such new data, which is an open issue in semantic web and related areas. In the AIM@SHAPE project, the geometric aspect has been tackled developing different automatic annotation tools in the DSW (Borgo et al., 2005), able to extract useful information from a specific digital shape and to translate it according to the metadata defined in the Common Shape Ontology. Similarly, some automation is predictable for the PDO, still referring to the morphological and topological aspect. In fact, several research results related to the automatic extraction of the assembly and feature information from the model evaluation are present in literature. There exist techniques that aim at extracting automatically the assembly graph from a complex model (see, for example, http://www.cadshuttle.org), while feature recognition methodologies are well-known and widely applied in product design.

On the contrary, the PD domain knowledge treated in this paper cannot be automatically extracted. Nevertheless, if we consider the adoption of the proposed methodology in industry, the amount of manual annotation may be limited to the annotation for each single project, once the company repository is structured according to the semantic conceptualisation proposed in the PDO.

More generally, the efficacy of new methodologies for 3D shape modelling and reasoning relies on an effective coupling between geometry and semantics. Such a goal should be obtained minimising the interaction between the user and the system, which has several margins of improvements in all the application areas where standardised procedures exist and can be identified and extracted. Therefore, a closer collaboration with the users for the collection and exploitation of the knowledge is needed in order to encode the semantics and enrich the intrinsic information of the product within the various contexts of the product design.

Acknowledgments. This work has been carried out within the scope of the AIM@SHAPE Network of Excellence supported by the European Commission Contract IST 506766. The work of Elena Camossi is supported by the Irish Research Council for Science, Engineering and Technology. The authors are grateful to the AIM@SHAPE partners, in particular they would like to thank Tor Dokken

(SINTEF), Gino Brunetti (IGD), Jean-Claude Leon (INPG), Franca Giannini (IMATI), Marios Pitikakis (ITI) for their active contribution to the PDO development.

References

AIM@SHAPE: Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects, European Network of Excellence, Key Action: 2.3.1.7 Semantic-based knowledge systems, VI Framework, Contract IST 506766, URL: ***http://www.aim-at-shape.net***; ***http://dsw.aimatshape.net***.

Amann K., (2002) Product lifecycle management: empowering the future of business, CIM Data, Inc.

Antti S., Anselmi. I, (2005) Product Lifecycle Management, Springer.

Bohm, M.R., Stone, R.B., Simpson, T.W., Steva, E.D. (in press). Introduction of a data scheme: To support a design repository. Computer-Aided-Design, In press, doi: 10.1016/j.cad.2007.09.003.

Borgo, R., Delle Piane, M., Cignoni, P., Papaleo, L., Spagnuolo, M. (2005) Extracting Meta-Information From 3-Dimensional Shapes With Protégé. 8th International Protégé Conference Proceedings, Madrid, July 18- 21. http://vcg.sourceforge.net/tiki-index.php?page=TriMeshInfo.

Corcho O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering 46:41–64. 2003.

Fenves, S.J., Foufou, S., Bock, C., Sriram, R.D (2008): “CPM2: A Core Model for Product Data”. Journal of Computing and Information Science in Engineering 8(1), pp. 014502-014508.

Fernández-López, M., Gómez-Pérez, A., Juristo, N.(1997): "Methontology: from ontological art towards ontological engineering". Proc. Symposium on Ontological Engineering of AAAI.

Gero J.S. and Rosenman M.A., (1996) Modelling multiple views of design objects in a collaborative CAD environment, INCIT Proceedings, 49-62.

Giménez, D.M., Vegetti, M., Leone, H.P., Henning, G.P. (2008) PRoduct ONTOlogy: Defining product-related concepts for logistics planning activities. Computers in Industry 59, 231-241.

Guarino, N. (1998) Formal ontology and information systems. In Proceedings of FOIS’98 (Formal Ontology in Information Systems), Trento, Italy, 6-8 June 1998. IOS Press.

Gupta, R.K., Gurumoorthy, B. (2008) A Feature-Based Framework for Semantic Interoperability of Product Models. Journal of Mechanical Engineering 54(6), pp. 446-457.

Hamri, O., Léon, J.C., Giannini, F., Falcidieno, B., Poulat, A., Fine, L. (2006) Interfacing product views through a mixed shape representation. Part 1: Data structures and operators, Proceedings of Virtual Concept.

Horvàth, I., Vergeest J.S.M. and Kuczogi, G., (1998) Development and Application of Design Concept Ontologies for Contextual Conceptualization, Proceedings of ASME Computers in Engineering Conference (DEC98), September 13-16, Atlanta, Georgia, USA.

Jun, H-B, Kiritsis, D., Xirouchakis, P., (2005) Product lifecycle information modelling with RDF. PLM '05, pp. 44-54.

Katzy, B., Schuh, G., (1998) The Virtual Enterprise. Handbook of Life Cycle Engineering: Concepts, Methods and Tools, Dordrecht, Kluwer Academic Publishers, 59-92.

Jun, H-B, Kiritsis, D., Xirouchakis, P., (2005) Product lifecycle information modelling with RDF. PLM '05 44-54.

Kitamura, Y. and Mizoguchi, R., (2004) Ontology-based systematization of functional knowledge. Journal of Engineering Design, 15, 327-351.

Kopena, J.B. and Regli W.C., (2003) Functional Modelling of Engineering Design. IEEE Data Engineering Bullettin, 26(4), 55-62.

Lee, K., (1999) Principles of CAD/CAM/CAE systems. Reading, Mass., Addison Wesley.

Li, Z., Liu, M., Anderson, D.C. and Ramani, K., (2005) Semantics-Based Design Knowledge Annotation and Retrieval. In IDETC/CIE 2005: ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. ASME.

Mukhopadhyay T., Kekre S., Kalathur S., (1995) Business Value of Information Technology: A Study of Electronic Data Interchange. MIS Quarterly, Vol. 19(2), pp. 137-156.

Nanda J, Simpson T. W., Kumara S. R. T., Shooter S. B.(2006), A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language, Journal of Computing and Information Science in Engineering, June 2006, Vol. 6, pp.103-113.

Nonaka, I. and Takeushi, H., (1995) The Knowledge-Creating Company, New York: Oxford University Press.

OWL, The Web Ontology Language, ***http://www.w3.org/2004/OWL/***.

Patil, L., Dutta, D. and Sriram, R., (2005) Ontology-Based Exchange of Product Data Semantics. Transaction on Automation Science and Engineering, 2, 213--225.

Posada, J., Toro, C., Wundrak, S., Stork, A., (2006) Using Ontologies and STEP Standards for the Semantic Simplification of CAD Models in Different Engineering Domains. Applied Ontology Journal, Volume 1, Number 3-4 2006, pp. 263-279.

Protégé, Ontology editor, http://protege.stanford.edu/

RDF, Resource Description Framework, http://www.w3.org/RDF/

OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition (2005). Available at http://www.nextgrid.org/download/OWL-WS%20at%20e-Science2005.pdf. University of Southampton IT Innovation Centre.

Shah, J.J., Mantyla M., (1995) Parametric and feature-based CAD/CAM, Wiley-Interscience Publication, John Wiley & Sons, Inc.

Stark, J., (2006) Product Lifecycle Management (PLM) - Report on State of the Art, John Stark Associate.

STEP, International Standard ISO/DIS 10303-214: Industrial automation systems and integration – Product data representation and exchange – Part 214: Application Protocol: Core data for automotive mechanical design process.

Sure, Y., Staab, S., Studer, R., (2003) On-To-Knowledge Methodology. Handbool on Ontologies. Series on Handbooks in Information Systems, S. Staab and R. Studer eds, pp. 117-132.

Vasilakis, G.; Pitikakis, M.; Vavalis, M.; Houstis, C., (2005) A Semantic-based Search Engine for 3D Shapes: Design and early Prototype Implementation. 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, IEE London.

Workflow Pattern, Web page: http://www.workflowpatterns.com/. Accessed march 2008.

XFlow Process Management System, Web page: http://xflow.sourceforge.net/. Accessed march 2008.

YAWL: Yet Another Workflow Language, Web page: http://www.yawl-system.com/. Accessed march 2008.