

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

IST NoE No 506766



a project financed by the
Sixth Framework Programme

Deliverable D1.2.3.1



Ontology for Product Design - 1st version

Circulation: ¹CO

Partner(s): **IGD (Leader), IMATI, INPG, INRIA, ITI, SINTEF**

Authors: N. Sevilmis, G. Brunetti, C. Catalano, E. Camossi, V. Cheutet, T. Dokken, R. Ferrandes, F. Giannini, J.-C. León, M. Pitikakis

Version: 02

Stage: Completed

Date: Thursday, March 2, 2006

¹ Please indicate the dissemination level using one of the following codes:

PU = Public

PP = Restricted to other programme participants (including the Commission Services).

RE = Restricted to a group specified by the consortium (including the Commission Services).

CO = Confidential, only for members of the consortium (including the Commission Services).

Copyright

© Copyright 2006 The AIM@SHAPE Consortium

consisting of:

CNR-IMATI-GE	C.N.R. – Istituto di Matematica Applicata e Tecnologie Informatiche Dept. of Genova, Italy
DISI	Università di Genova – Dipartimento di Informatica e Scienze dell'Informazione, Italy
EPFL	École Polytechnique Federale de Lausanne, Switzerland
FhG/IGD	Fraunhofer Institut für Graphische Datenverarbeitung, Germany
INPG	Institut National Polytechnique de Grenoble, France
INRIA	Institut National de Recherche en Informatique et Automatique, France
ITI-CERTH	Informatics and Telematics Institute – Center for Research and Technology Hellas, Greece
UNIGE	Université de Genève, Switzerland
MPII	Max-Planck-Institut für Informatik, Germany
SINTEF	Stiftelsen for industriell og teknisk forskning ved Norges Tekniske Høgskole, Norway
TECHNION	Technion – Israel Institute of Technology, Israel
UU	Utrecht University, Netherlands
WEIZMANN	Weizmann Institute of Science, Israel

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the AIM@SHAPE Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

All rights reserved.

This document may change without notice.

Document History

Vers	Issue Date	Stage	Content and changes
01	16 January 2006	Draft	Initial structure and content.
02	01 March 2006	Completed	Scenarios included

Executive Summary

This document contains the deliverable **D1.2.3.1** of the IST NoE AIM@SHAPE.

The deliverable ***D1.2.3.1 – Ontology for Product Design - 1st Version*** – is intended to provide a first version of the new Product Design Ontology, which is part of the ontology development process in the network.

The task leader is **IGD** and has been actively supported by all involved partners.

Following the structure of the Deliverable: Chapter 1 introduces the motivation for the development of an ontology and its purpose in the field of Product Design. Chapter 2 summarizes the need of ontologies as a means to support information management and knowledge sharing within the network. Chapter 3 discusses the main purpose (target applications, competency questions) and key elements of the Product Design Ontology, before discussing the design of the actual Product Design Ontology in Chapter 4.

Table of Contents

1	INTRODUCTION	7
2	ONTOLOGY DEVELOPMENT	9
3	AN ONTOLOGY FOR PRODUCT DESIGN (V1.0)	11
3.1	Target Applications	12
3.2	Initial Key Entities and Creation of the Concepts of the Ontology	12
3.3	Usage Scenarios and related Ontology Queries	13
3.3.1	Free-form deformation on NURBS models (INPG-IMATI-IGD)	13
	Scenario	13
	Competency Questions	14
3.3.2	Quality check of CAD-models	15
	Scenario	15
	Competency questions	15
3.3.3	FEA model preparation and analysis of the details effects	16
	Scenario	17
	Competency questions	18
3.3.4	Acquisition of test data for FEA post-processing	20
	Scenario	21
	Competency questions	21
4	ONTOLOGY DESIGN AND EVOLUTION	23
4.1	Modeling Knowledge about Product Design Processes	23
4.1.1	The Task Concept	23
4.1.2	Tasks of the Product Design Ontology	25
4.2	Boundary Condition Type Hierarchy	28
4.3	CAD Representation Type	29
4.4	Tools Concept Extensions	30
4.5	Shape Concept Extensions	30
5	FUTURE ONTOLOGY EXTENSIONS AND OPEN ISSUES	31
	REFERENCES	33

Table of Figures

Figure 1: Shape life-cycle in a typical Finite Element Analysis (FEA) design evolution loop.16

Figure 2: Inter-relations between Task, ShapeRepresentation and SoftwareTool23

Figure 3: Protégé screenshot of the Product Design Ontology: sub-classes of the Task concept and their instances24

Figure 4: Task concepts and instances with relationships in the Product Design Ontology.27

Figure 5: Product Design Tasks as modeled in the Product Design Ontology. Main flow is taken from ISO 10303 (STEP), AP214. The Shape Acquisition and Reconstruction process as well as possible relationships to the Virtual Humans Ontology are included.28

Figure 6: Boundary Condition Type Hierarchy29

1 INTRODUCTION

Effective and efficient information management and knowledge sharing has become an essential part of more and more professional tasks and workflows in the Product Development Process. Semantic Web technologies and Ontologies are quite promising in this field and offer new possibilities regarding knowledge management, retrieval effectiveness and online collaboration compared to conventional technologies and techniques.

We have developed a Product Design Ontology to be applied in e-science scenarios especially useful for researchers working in the development of tools and methods supporting industrial product design and engineering analysis. The main purpose of the Ontology for Product Design is to formalize semantic concepts of interest and interlink them through semantic relationships in order to make them usable for computer-assisted tools and stylists (e.g. annotation of a 3D model by the character lines entity).

The Ontology for Product Design comprises relevant knowledge in the following areas:

- Knowledge about the representation of shape semantics in industrial design and analysis (e.g. expressing, preserving and retrieving the design intent; better integrating the styling activity in the design activity by exploiting related contextual knowledge; providing hints and information to different points of view (manufacturing, simulation, ergonomics, etc.) and support the early computer-assisted design activity.)
- Knowledge about the functionality and usage of shape processing methods and algorithms
- Knowledge about processes and workflows in the respective product development phases.

Altogether we are testing both, the expressiveness of standardized ontology description languages (W3C standards: RDF(S)/OWL) that are used to formalize the semantic of product design data and 3D models, and the Product Design Ontology for advanced and reasoning-based information management and retrieval to assist the development of shape processing tools for design, but also for the product design phase itself.

2 ONTOLOGY DEVELOPMENT

Ontologies are designed to be used in a highly decentralized manner, appropriate for global information networks, such as AIM@SHAPE, and are well suited for information integration tasks in distributed environments, such as the DSW (Digital Shape Workbench of AIM@SHAPE).

Ontologies are a means to formalize explicit knowledge related to a specific domain. Explicit knowledge can be used as input for reasoning algorithms in order to deduce implicit knowledge, i.e. statements that have not been explicitly modeled. For instance, in the context of shape modeling the domain of interest would cover engineering knowledge on different levels ranging from basic geometrical and topological relationships to form features and higher-level application-specific features that carry semantics. Reasoning amounts to conclude implicit topological relationships as well as higher-level relations holding between features and can be applied to CAx or shape retrieval. That is, through the combination of ontologies and their inference mechanisms we are able to model, represent and better organize all the knowledge that is required to answer competency questions (see Chapter 3) and to process information-intensive tasks in the field of Product Design.

The predecessor to this Product Design Ontology addressed just styling aspects. More specifically, the previous version was also focused on how shape semantics can be made explicit in computer-aided styling (CAS) systems. For this new Product Design Ontology the cluster focused the target applications and target user group to model process, tool and shape know-how relevant to the two phases of the product development process addressed in WP7, i.e. the free-form styling and the engineering simulation. In fact, formalizing stylists' knowledge when creating new products is a challenging issue which requires more research activity; then, it will be faced extensively in WP7 and possibly integrate here in the future. The previous version of the design ontology as well as the ISO STEP-based geometry ontology have been incorporated into the product design ontology and the common shape and tool ontologies where applicable.

The Product Design Ontology is designed following the methodology proposed in the OnToKnowledge project (IST-1999-10132) (compare with D1.2.2.1). In addition, the consortium has decided to specify all the cluster ontologies in the Web Ontology Language (OWL). OWL is a W3C Recommendation and constitutes the most expressive ontology language that is available today. It has also been decided to use Protégé as a support tool for the ontology design and development, and to adopt the related OWL Protégé plug-in.

3 AN ONTOLOGY FOR PRODUCT DESIGN (V1.0)

The domain of the design ontology has been defined as the development, usage and sharing of software tools and shape data by researchers and experts in the field of industrial design and engineering analysis.

Three main concepts have to be elaborated to address the needs of the product design ontology: **Shape Representations**, shape processing **SoftwareTools** and **Algorithms**, and the product design **Tasks**, which communicate via shapes and related metadata and that are supported by software tools that consume or produce shapes. Further design specific concepts enrich the ontology with relevant knowledge for software tools, algorithms, and shapes. Examples are information about the types of boundary conditions applied for an engineering simulation, the role of a shape in the design process, the owners of specific instances, related literature, file information, etc.

Shapes in different **shape representations** and shape processing **software tools** and **algorithms** are expected to be either resources that can be uploaded and downloaded together with their metadata in the shape and tool repositories, respectively, or they represent metadata about respective tools and algorithms, possibly along with an URL linking them to further information or for getting into contact with the owner. Especially algorithms might also be interrelated with information stored in the Digital Library.

Shapes are data sources or results of specific **tasks** along the product development process. Software tools and algorithms, in turn, support one or more specific **tasks** along this process and consume and/or produce shapes as well as related metadata. To specify the product design process the following main process tasks have been identified. They represent a typical workflow of subsequent product development activities and will be further detailed:

1. "Product Styling": Process of specifying the shape of a product in consideration of aesthetical and visual aspects, which is addressed in the Task 7.2.
2. "Product Design": Definition of the functionality, geometry, and technological properties of a product, including the product analysis addressed in Task 7.3.

For the sake of completeness the ontology also contains the predecessor task "Product Definition", where the essential properties and characteristics of a product are defined.

Furthermore, the successor task "Product Evaluation" is included, on the one hand again for the sake of completeness; on the other hand, because the **Acquisition and Reconstruction Ontology** developed in T1.2.2 (see D1.2.2) is embedded in this main task. In addition there is a sub-task relationship between the Acquisition and Reconstruction Ontology and the "Product Styling" task.

In a similar way there is a relationship between the Product Design Ontology and the **Virtual Humans Ontology** (T1.2.3/D1.2.3), as the generation of virtual humans and the related smart objects are applied for ergonomics analysis being sub-task of the major "Product Design" task (see Figure 5).

The first two levels of the product design tasks (compare with Figure 5, Chapter 4) are

based on the Application Activity Model (AAM) of AP214 of the ISO Standard 10303 (STEP – Standard for the Exchange of Product Information) [ISO 10303] that provides a well studied process model from product definition, styling, design, to engineering analysis and simulation including information about data flow and involved tools in the process. In STEP the AAM is meant – as it is the case with the tasks in this product design ontology – to define the context and workflow where the different shape representations are used and the different shape processing tools are applied.

Note that ISO STEP separates between the conceptual **Product Styling** phase and the **Product Design** phase, where all shape and functional aspects are detailed, e.g. using computer-aided design (CAD) systems, and analyzed (simulated), e.g. using computer-aided engineering (CAE) systems applying, for instance, applying finite element analysis methods. It was decided, however, to name the ontology described in this document **Product Design Ontology**, because product design is generally used as term covering the whole product development process from definition to physical evaluation. In this sense the product design task of this ontology is meant as the task where initial functional and form concepts are detailed until the final product specification is achieved.

3.1 Target Applications

The target group of the Ontology for Product Design is initially related to researchers and software developers developing methods and tools for styling and industrial design as well as for engineering analysis, respectively. Furthermore, the ontology can be used by stylists/industrial designers, analysis engineers and other experts to gather domain knowledge about shape processing workflows and related tools.

The ontology, therefore, is designed to support those that either use shape processing tools applied in the product design process or those that develop new tools applicable in this process. For the first group the ontology is intended to provide domain knowledge, for instance, about a pipeline of tools that can be applied to achieve a certain result, e.g. a shape in a specific representation for a given application along the product development process.

For the second group, the developers of shape processing methods and tools, the ontology intends to provide information about related methods, the application context of certain methods within the product design process, access to suitable test shapes for benchmarking of shape processing tools, etc. Obviously, especially researchers and developers might belong to both groups. For instance, performing research in the area of simulation post-processing it might be of interest to find a way about how a specific CAD model available in the shape repository might be processed to become a mesh with certain characteristics to be used as test data for the own tools or to benchmark it with other post-processing tools.

Target applications of the Product Design Ontology, therefore, are any activities dealing with shape processing (acquisition, creation, reconstruction, conversion, processing etc.) relevant within the product design process.

3.2 Initial Key Entities and Creation of the Concepts of the Ontology

Key entities of the Product Design Ontology are, as already presented in the introduction to this chapter:

- **Shape Representations** (references to the common shape ontology) as well as

design specific shape related metadata like for instance, Boundary Conditions Types applied in simulation tasks;

- **SoftwareTools** and **Algorithms** (references to the common tool ontology) used to process shapes in the design context;
- Product design **Tasks**, which communicate via shapes and related metadata and that are performed by software tools and algorithms that consume or produce shapes.

3.3 Usage Scenarios and related Ontology Queries

This section describes cluster-specific scenarios with associated queries for the Product Design Ontology. The queries identified significantly influenced the ontology development.

The scenarios address two main phases of the product development, namely the product styling (refers to T7.2) and the simulation (refers to T7.3) performed during the detailed product design.

For product styling two typical scenarios are selected. One is the deformation of free-form surfaces in NURBS (Non-Uniform Rational B-Spline) representation. The second is also dealing with shapes in NURBS representation and addresses model quality issues to provide adequate data for down-stream applications.

For simulation also two scenarios have been selected. The first addresses Finite Element Analysis (FEA) model preparation including model simplification and meshing aspects. The second deals with the simulation post-processing to examine and interpret simulation results.

In the discussion of the scenarios, Competency Questions (CQ) are classified in two groups:

- CQs with a tick (✓) are the CQs that the current version of the ontology can already answer;
- CQs with a cross are the ones that can not be answered yet, but that will be addressed by future versions of the ontology.

3.3.1 Free-form deformation on NURBS models (INPG-IMATI-IGD)

Based on a product definition, often referred to as packaging, an early step in the product development is the conceptual definition of the functional and aesthetical characteristics of a new product. Especially in the case of styling, defining the aesthetical and stylistic characteristics of a product involves modelling with free-form geometry, typically using the parametric NURBS (Non-Uniform Rational B-Splines) representation for curves and surfaces to achieve compatibility with down-stream applications like, for instance, Computer-Aided Design (CAD).

Besides providing a sketched-based interface to define these main characteristics of a new shape, free-form deformation methods are important for Computer-Aided Styling (CAS), e.g. for achieving high quality surfaces with smooth transitions between adjacent surface patches and to introduce free-form features to define local details of the shape.

Scenario

The scenario addresses the need of developers of free-form deformation tools, e.g. e-scientists, to test the own tool with appropriate models and to compare it with the

capabilities and results of other free-form deformation tools. For this it is necessary to find appropriate NURBS models as well as information about other NURBS deformation algorithms and software tools.

In the scenario, the e-scientist performs the following steps, each corresponding to a task in the product design ontology:

- Step 1. **Search** for a CAD model with NURBS patches of a given category, e.g., automotive, that represents a real-object. If nothing is found, he/she queries for a model of a given category, that represents a real-object, and then, he/she queries for tools to transform the geometry of the found models into the NURBS representation.
- Step 2. *Analyse* the model, for checking the connectivity between patches.
- Step 3. *Prepare* the model, correcting the possible errors on the shape, e.g. performing a healing process.
- Step 4. *Perform the deformation* by using the developed NURBS deformation tool.
- Step 5. *Search* for other NURBS deformation tools to apply them for the deformation on the same models for comparison of results.
- Step 6. *Perform a quality check* by comparing the map of Gaussian curvature of the results of steps 4 and 5.

Competency Questions

In the following all CQs in the form of, for instance,

“Find the tools performing the quality check?”

can have two different interpretations:

- Find the software tools that implement the **algorithm** of category quality check algorithm
or(/and)
- Find the software tool that supports the **task** quality check.

In the list that follows, CQs are classified in two groups: CQs with a tick are the CQs that the current version of the ontology can already answer; CQs with a cross are the ones that can not be answered yet, but that will be addressed by future versions of the ontology.

Step 1.

- ✓ Find a CAD model, which product category is automotive and is a real object.
- ✓ Find a model, which product category is automotive and is a real object.
- ✓ Find a tool that performs conversion, and has as input a given shape type (see previous CQ), and as output a CAD model.

Step 2.

- ✓ Find the tools that check for connectivity between patches and have as input a NURBS model.

Step 3.

- ✓ Find the tools that perform healing, and have as input a NURBS model.

Step 4.

- ✓ Find the tools that perform deformation, and have as input a NURBS model.

Step 5.

- ✓ Find the tools that build Gaussian curvature map and have as input a NURBS model.

3.3.2 Quality check of CAD-models

All current CAD-systems (year 2006) are based on the technology embedded in the STEP standard from the early 1990s. Consequently the inherit qualities of STEP and of current CAD-systems are aimed at what was computationally feasible in 1990 and the qualities required by industries in the early 1990s. The principles for geometry representation in STEP are the boundary representation and the assumption that edges and faces are represented in exact way. In the case of surfaces of low algebraic degree (planes, spheres, ...) we get a near exact representation of the curves representing topological edges. However, for NURBS represented geometries this is not the case. There will always be gaps between the faces, these are often repaired within algorithms inside a given CAD-systems. However, they will be present in exported CAD-models. Multiple design methods can produce self-intersecting surfaces in a CAD-model. However, in cases where the self-intersections are not visually detected by the CAD-user they will in many CAD-systems remain undetected.

This scenario outlines future developments of the product design ontology. Though, some CQs can be answered we the current version of the ontology.

Scenario

A CAD-model is received from a CAD-systems and the feasibility of the use of the CAD-model for down stream processes or long time storage is to be determining. Two levels of problems are to be addressed:

- Defects that makes the CAD-model non-manifold (within specified tolerances) and consequently invalid.
- Configurations that make down stream applications difficult.

Competency questions

Competency questions related the manifold requirement:

- ✗ Is the CAD-model 3-manifold within specified tolerances?
- ✓ How is the model connected (C0,C1,C2,G0,G1,G2) within specified tolerances?
- ✗ Is the CAD-model tested for self-intersections?
- ✓ Is the model without self-intersecting faces?
- ✗ Is the single surfaces/patches of model without ridges or vanishing normals?
- ✗ Do any hull in the CAD -model self-intersect?
- ✗ Do different hulls in the CAD-models intersect?

Competency questions that might make down stream applications work better

- ✗ Is there large variation in patch size within the model?
- ✗ Are there long and very narrow patches within the model?
- ✗ Are there extremely small patches in the model?
- ✗ Are there extreme curvatures in the model?
- ✗ Is each single faces of the model 2-manifold?

- ✗ Do any of the surfaces used for describing the faces have self-intersections outside of the part used by the patch description?
- ✗ Are self-intersections in the part of a surface used for describing a patch trimmed away edges?

3.3.3 FEA model preparation and analysis of the details effects

Finite element analysis was first developed for use in the aerospace and nuclear industries where the safety of structures is critical. Today, the growth in usage of the method is directly attributable to the rapid advances in computer technology in recent years. As a result, finite element packages exist that are capable of solving the most sophisticated problems, not just in structural analysis, but for a wide range of phenomena such as steady state and dynamic temperature distributions, fluid flow and manufacturing processes such as injection molding and metal forming. A typical shape evolution cycle for FEA is illustrated in Figure 1.

FEA consists of a computer model of a material or design that is loaded and analyzed for specific results. It is used in new product design, and existing product refinement. It is possible to verify that a proposed design will be able to perform to the products specifications prior to manufacturing or construction. Modifying an existing product or structure is utilized to qualify the product or structure for a new service condition. In case of structural failure, FEA may be used to help determine the design modifications to meet the new condition.

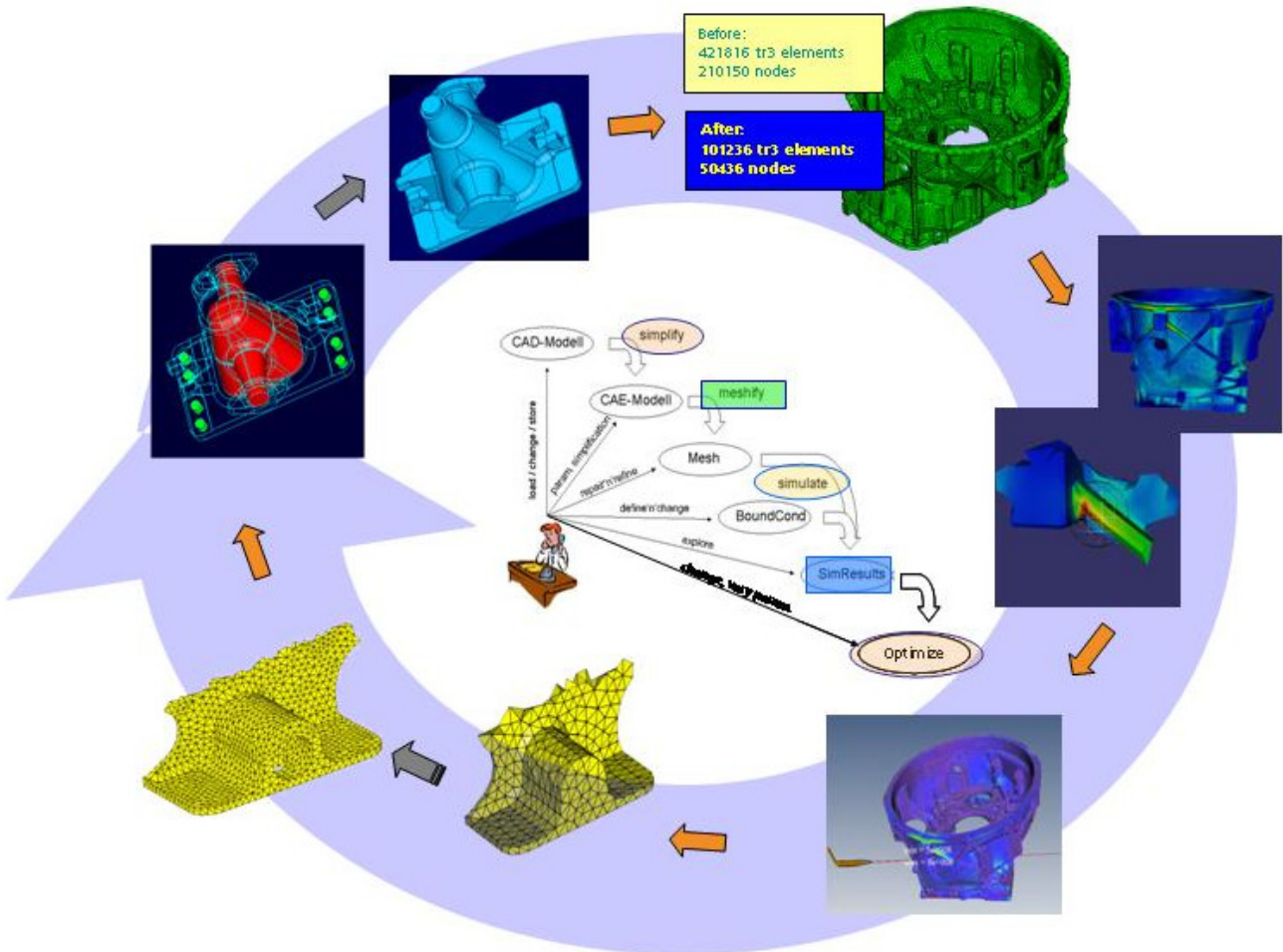


Figure 1: Shape life-cycle in a typical Finite Element Analysis (FEA) design evolution loop.

Mathematically, the structure to be analyzed is subdivided into a mesh of finite sized elements of simple shape. Within each element, the variation of displacement is assumed to be determined by simple polynomial shape functions and nodal displacements. Equations for the strains and stresses are developed in terms of the unknown nodal displacements. From this, the equations of equilibrium are assembled in a matrix form which can be easily be programmed and solved on a computer. After applying the appropriate boundary conditions, the nodal displacements are found by solving the matrix stiffness equation. Once the nodal displacements are known, element stresses and strains can be calculated

Within each of these modeling schemes, numerous algorithms (functions) can be inserted, which may make the system behave linearly or non-linearly. Linear systems are far less complex and generally ignore many subtleties of model loading and behavior. Non-linear systems can account for more realistic behavior such as plastic deformation, changing loads etc. and is capable of testing a component all the way to failure.

Today FEA is routinely used to solve problems in areas like:

- Structural strength design
- Structural interaction with fluid flows
- Analysis of Shock (underwater & in materials)
- Acoustics
- Thermal analysis
- Vibrations
- Crash simulations
- Fluid flows
- Electrical analyses
- Mass diffusion
- Buckling problems
- Dynamic analyses
- Electromagnetic evaluations
- Metal forming
- Coupled analyses

The typical workflow for FEA is to tessellate (meshing) the original CAD model applying boundary conditions, e.g. forces, displacements, temperature, etc., to perform FEA on this mesh and to post-process the simulation result for interpretation of the simulation outcome. As simulation is computational expensive and to keep the simulation task manageable, the CAD models typically are simplified for simulation, either before the meshing in the CAD system or after tessellation in the resulting polyhedron. Simplification typically means to reduce the complexity by suppressing design details (defeaturing) as blendings, holes, etc. that are considered not to influence significantly the simulation outcome. The influence of shape details on the simulation result and the decision, which shape details can be suppressed for a specific simulation, however, needs to be carefully analysed and requires significant expert knowledge.

Scenario

An e-scientist wants to test his/her tool for FE model preparation (shape simplification) on tessellated CAD models. During the shape simplification process shape details are

suppressed. It becomes necessary to analyse the influence of each suppressed detail on FE simulation results.

In the scenario, the e-scientist performs the following steps:

1. *Search* for a CAD model of a given category, e.g., mechanical part, that represents a real-object in a given format, e.g., STEP or stl. The e-scientist can provide information about boundary conditions required by a further process. Moreover, he/she can provide information about the requested complexity of the model, e.g., number and type of features, number of faces.
2. *Tessellation* of the model to transform the initial CAD geometry into a polygonal model.
3. Analysis of the model for *checking the connectivity* between patches.
4. Perform *conformity set-up* to make the polyhedron conform, if necessary.
5. Perform *defeaturing (or simplification)* of the polygonal model.
6. Perform *3D meshing* of the simplified model or of the suppressed details.
7. Perform the *FE analysis* of the simplified model.
8. *Determine the influence* of each suppressed detail on the simulation results (e.g., for linear static problems). *Apply one indicator* for each removed detail, where an indicator is a tool that evaluates the accuracy of the FE results in accordance to the shape simplifications performed.
9. Perform *refeaturing* of the simplified model

Competency questions

Step 1.

- ✓ Find a CAD model in a given format, which product category is mechanical part, and is a real object.
- ✗ Find a mesh model of a given format, which product category is mechanical part, and is a real object, that has certain boundary condition types. *This CQ will be addressed through the concept Role of a shape in the future version of the ontology.*
- ✓ Find a CAD model of a given format, which product category is mechanical part, and is a real object, that has a maximum number of faces.
- ✓ Find a CAD model of a given format, which product category is mechanical part, and is a real object, that has through holes.
- ✓ Find a CAD model of a given format, which product category is mechanical part, and is a real object, that has a minimum/maximum number of through holes.

Step 2.

- ✓ Find a tool that performs conversion that has as input type a CAD model of a specific type (see previous point), which output type is a polygonal triangulated model, and which execution platform is Windows 2k.

Step 3.

- ✓ Find the tools, which execution platform is Windows 2k, that checks for connectivity between patches and have as input a polygonal triangulated model with format stl.

Step 4.

- ✓ Find the tools which execution platform is Windows 2k that perform conformity set-up, and have as input a polygonal triangulated model with format stl.

Step 5.

- ✓ Find the tools which execution platform is Windows 2k that perform defeaturing (or simplification), and have as input a polygonal triangulated model with format stl (or iges, STEP).
- ✓ Find the tools which execution platform is Windows 2k that perform defeaturing (or simplification) of skin details and have as input a polygonal triangulated model with format stl.
- ✓ Find the tools which execution platform is Windows 2k that perform defeaturing (or simplification) of topological details (e.g., through holes) and have as input a polygonal triangulated model with format stl.
- ✓ Find the tools which execution platform is Windows 2k that is able to identify and select details (e.g., through holes) and have as input a polygonal triangulated model with format stl.
- ✓ Find the tools which execution platform is Windows 2k that save suppressed details (e.g., through holes) and have as input a polygonal triangulated model with format stl.
- ✓ Find the tools which execution platform is Windows 2k that save suppressed details (e.g., through holes) and have as input a tessellated model with format stl.
- ✓ Find the tools which execution platform is Windows 2k that save suppressed details (e.g., through holes) and have as input a tessellated model with format stl and have as output model with format stl (or iges, STEP)
- ✓ Find the tools which execution platform is Windows 2k that save suppressed details (e.g., through holes) and have as input a tessellated model with format stl and have as output a model with format stl.

Step 6.

- ✓ Find the tools which execution platform is Windows 2k that perform 3D meshing, and have as input a CAD model of a give type (see previous point) with format stl (or iges, STEP) have as output a 3D Mesh, which elements are tetrahedrons.
- ✓ Find the tools which execution platform is Windows 2k that perform 3D meshing, and have as input a CAD model of a give type (see previous point) with format stl (or iges, STEP) have as output a 3D Mesh, which elements are tetrahedrons, with format abaqus (or ansys, unv).
- ✓ Find the tools which execution platform is Windows 2k that perform 3D meshing, and have as input a CAD model of a give type (see previous point) non-regular with format stl (or iges, STEP) have as output a 3D Mesh, which elements are tetrahedrons, with format abaqus (or ansys, unv).

Step 7.

- ✓ Find the FE analysis tools which execution platform is Windows 2k, which have as output a model with format abaqus (or ansys, unv).
- ✗ Find the FE analysis tools which execution platform is Windows 2k, which have as output a model with format abaqus (or ansys, unv) and the some information of the simulation like displacement of nodes; strain energy of

tetrahedral elements. *This CQ will be addressed through the concept Role of a shape in the future version of the ontology.*

Step 8.

- ✓ Find a tool which determines the influence of each suppressed detail on the simulation results.

Step 9.

- ✓ Find the tools which execution platform is Windows 2k that perform refeaturing.

3.3.4 Acquisition of test data for FEA post-processing

The result of a FEA, however, needs to be studied in a post-processing to able to make decisions regarding the suitability of the design with respect to its engineering specification. Post-processing, therefore, is an important tool for a graphical interactive exploration and interpretation of static and dynamic simulation results.

Post-processing begins with a thorough check for numerical problems that may have occurred during solution. Once the solution is verified to be free of numerical problems, the quantities of interest may be examined. Many interactive visualization techniques are available, the choice of which depends on the mathematical form of the quantity as well as its physical meaning. For example, the displacement of a solid linear brick element's node is a 3-component spatial vector, and the model's overall displacement is often displayed by superposing the deformed shape over the undeformed shape. Dynamic viewing and animation capabilities aid greatly in obtaining an understanding of the deformation pattern. Stresses, being tensor quantities, currently lack a good single visualization technique, and thus derived stress quantities are extracted and displayed. Principal stress vectors may be displayed as color-coded arrows, indicating both direction and magnitude. The magnitude of principal stresses or of a scalar failure stress may be displayed on the model as colored bands.

An area of post-processing that is rapidly gaining popularity is that of adaptive remeshing. Error norms such as strain energy density are used to remesh the model, placing a denser mesh in regions needing improvement and a coarser mesh in areas of overkill. Adaptivity requires an associative link between the model and the underlying CAD geometry, and works best if boundary conditions may be applied directly to the geometry, as well. Adaptive remeshing is a recent demonstration of the iterative nature of h-code analysis.

Optimization is another area enjoying recent advancement. Based on the values of various results, the model is modified automatically in an attempt to satisfy certain performance criteria and is solved again. The process iterates until some convergence criterion is met. In its scalar form, optimization modifies beam cross-sectional properties, thin shell thicknesses and/or material properties in an attempt to meet maximum stress constraints, maximum deflection constraints, and/or vibrational frequency constraints. Shape optimization is more complex, with the actual 3D model boundaries being modified. This is best accomplished by using the driving dimensions as optimization parameters, but mesh quality at each iteration can be a concern.

Another direction clearly visible in the finite element field is the integration of FEA packages with so-called "mechanism" packages, which analyze motion and forces of large-displacement multi-body systems. A long-term goal would be real-time computation and display of displacements and stresses in a multi-body system undergoing large displacement motion, with frictional effects and fluid flow taken into account when necessary.

Scenario

The discussed scenario addresses e-scientist in the area of FEA post-processing that seek for adequate shapes, e.g. volume and surface meshes, that are simulation results of a FEA. Main issue for the test of FEA post-processing tools is the acquisition of relevant, if possible real-world, test data. Formats of interest include: CGNS (Computational Grid Notation Standard, an industrial standard), flt (Fluent), simr (SimReal), Ensignt, STEP AP237 (ISO STEP draft), Cfx (CFX), Modlflow, Pamflow, STEP AP209 (ISO standard), Patran, Nastran, Ansys, Adams, Abagus, Ideas, CATIA, etc.

An addition, it is of interest to find other post-processing tools applicable to the investigated type of simulation (related to the Boundary Condition Type) to compare the own tool, e.g. for benchmarking.

Competency questions

The CQs are built around the attributes describing the type and content of such simulation results and post-processing tools:

- ✗ Find shapes that are simulation results of a FEA of given Boundary Condition Type, that are possibly real-world objects, and that are in a given file format. *This CQ depends on the development of the Shape Role concept planned to be introduced in a follow-up version of the Product Design Ontology.*
- ✓ Find conversion software tools that convert shapes being simulation results between two given formats.
- ✗ Find post-processing algorithms or post-processing software tools that work with a given type of simulation results (*depending on Boundary Condition Type, which needs to be related with the post-processing task in a follow-up version of the Product Design Ontology*).
- ✓ Given a CAD model in a specific format, find a workflow and/or a sequence of software tools that generate a simulation result of the type (Boundary Condition Type) and format requested.

Substituting the respective boundary conditions types and formats lead to the single concrete CQs relevant to this scenario.

4 ONTOLOGY DESIGN AND EVOLUTION

The Product Design Ontology is being developed using the Protégé v3.1.1 ontology editor and its OWL plug-in.

4.1 Modeling Knowledge about Product Design Processes

4.1.1 The Task Concept

As already described in chapter 3, the main concept in the Product Design Ontology is the **Task** concept (see Figure 2). It has four auto-relations (*hasPredecessor*, *hasSuccessor*, *hasSubtask*, *hasSuperTask*). In this way, we are able to model via the four auto-relations on the instance level any sequences of tasks that constitute a specific process or process step. The relations (*hasPredecessor*, *hasSuccessor*) and (*hasSubtask*, *hasSupertask*) are defined to be inverse relations.

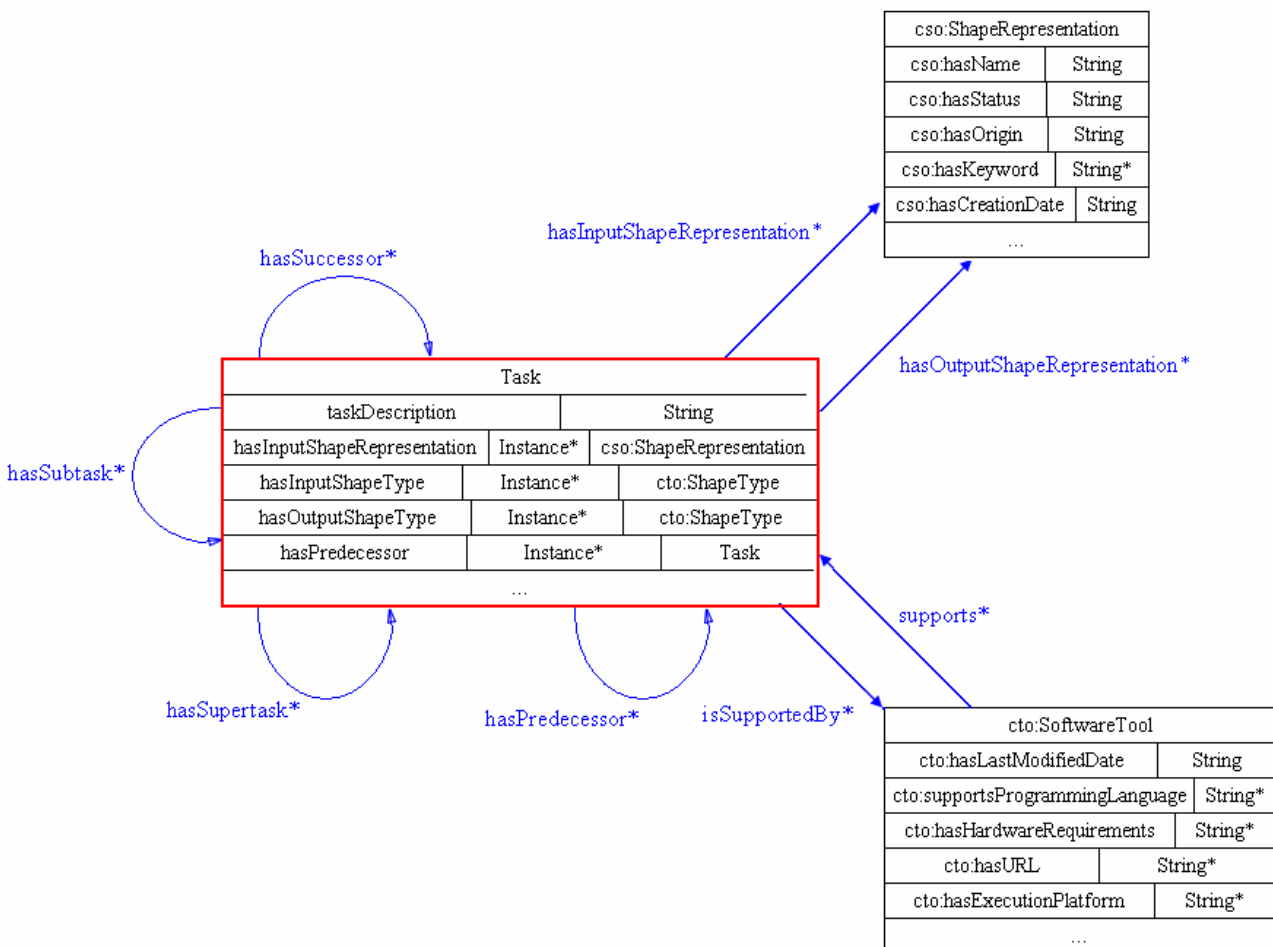


Figure 2: Inter-relations between Task, ShapeRepresentation and SoftwareTool

Each task is related to input and output *ShapeTypes* through the relations *hasInputShapeType* and *hasOutputShapeType*. These entities represent the types of the shape representations which intervene in that specific task and contribute to express the knowledge of the shape processing within the design workflow. Moreover, each *Task* has also input and output *ShapeRepresentations* (via the relations *hasInputShapeRepresentation* and *hasOutputShapeRepresentation*, respectively) to provide concrete examples of task I/O models included in the Shape Repository.

Analogously, each task is related to some algorithm categories (*hasAlgoCategory*) to represent the types of computations that are used in the task. Again, the relationships *supports* and *isSupportedBy* between the concepts *Task* and *SoftwareTool* are inverse relations, which permit to retrieve the tools, possibly present in the Tool Repository, suitable for the task activities.

The current version of the Product Design Ontology contains several sub-classes of the *Task* concept. Section 4.1.2 describes the specific task instances. On the one hand the structure of the ontology must be rich enough in order to be able to understand and process queries (concept level), on the other hand the knowledge base must be able to return results (instance level).

Figure 3 shows the new current of the Product Design Ontology with respect to the *Task* concept. The *Task* concept has been specialized by different sub-classes conceptualizing specific tasks such as *Meshing*, *ShapeSimplification*, *PartDesign*, *ShapeStyling*, etc. Generally, such sub-classes represent the main tasks of the process.

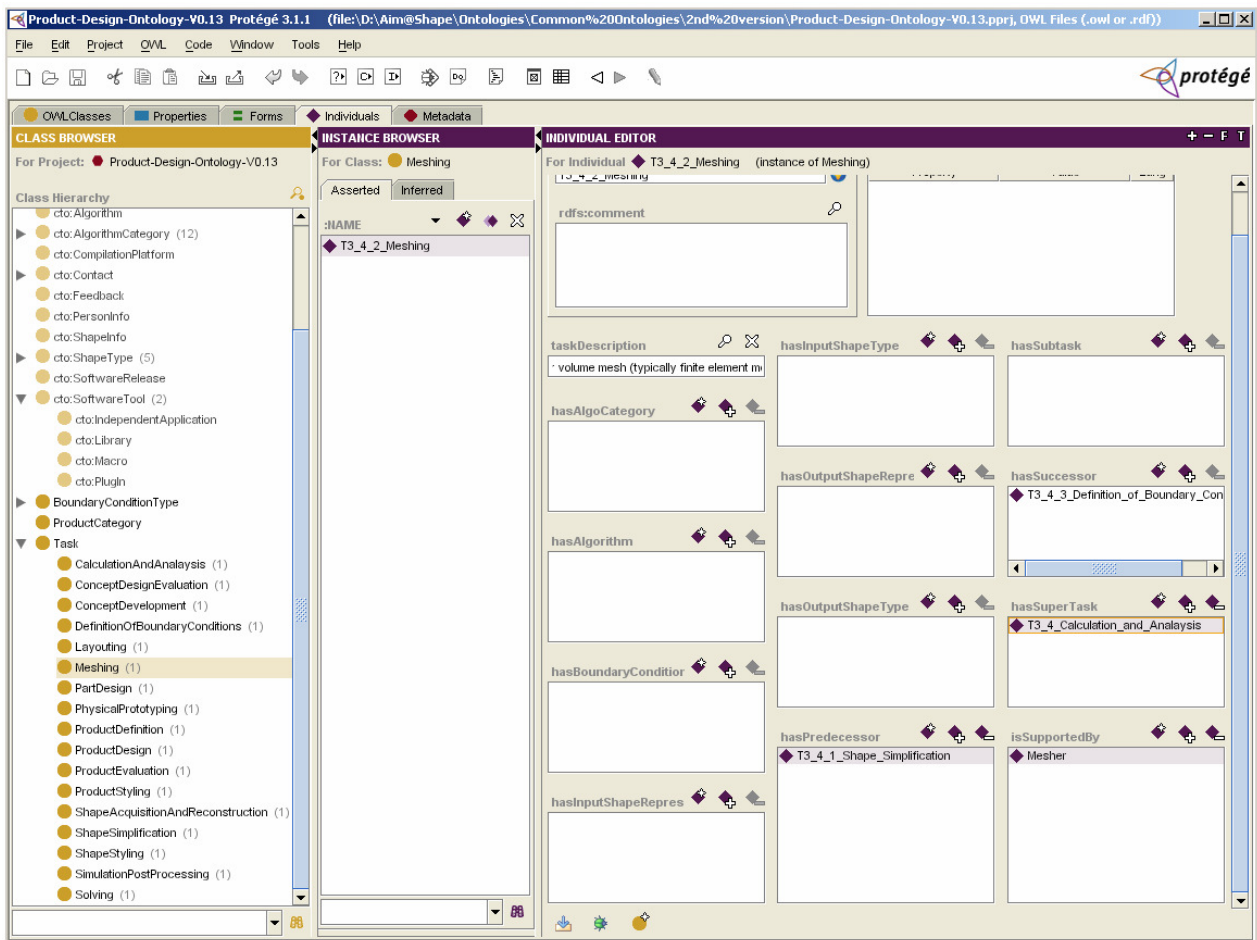


Figure 3: Protégé screenshot of the Product Design Ontology: sub-classes of the Task concept and their instances

4.1.2 Tasks of the Product Design Ontology

Note that the currently modeled and instantiated tasks and task interrelationships have been included just as an initialization of the Product Design Ontology, because they are part of the ISO Standard STEP [ISO 10303] and therefore valid. The current ontology should not be considered as fully defined with relation to design tasks, but to be further extended and detailed as long as new shapes and shape processing tools and methods are inserted to the domain in general and to the respective AIM@SHAPE repositories in particular.

The current **Tasks** in the Product Design Ontology are described below:

- *T1_Product_Definition*: Task in which the essential properties of a product in the form of lists of essential characteristics such as function, size, material, weight, etc. are specified. Even if this task is currently not part of the AIM@SHAPE Product Design Ontology, it is however included for completeness.
- *T2_Product_Styling*: Process of specifying the shape of a product in consideration of aesthetical and visual aspects.
 - *T2_1_Concept_Development*: Elaboration of the idea of a product developed with respect to a list of specifications.
 - *T2_2_Physical_Prototyping*: Process of building a physical prototype based on 3D shape models. Physical prototyping is not target of the AIM@SHAPE Product Design Ontology. However, as physical prototypes are input to subsequent quality insurance activities involving *acquisition and surface reconstruction (measurement) methods* (see D1.2.2), it is included for completeness.
 - *T2_3_Shape_Styling*: Process of generating the 3D shape description of a product.
- *T3_Product_Design*: Task in which the functionality, geometry and technological properties of a product are defined in detail.
 - *T3_1_Layouting*: Task for developing the layout of the whole product including all major dimensions. Often called packaging layout definition.
 - *T3_2_Concept_Design_Evaluation*: Task verifying the feasibility of the design of the major parts (components) of a product.
 - *T3_3_Part_Design*: Process of designing the details of the single parts of a product. Detailed design - CAD - is not in the focus of the Product Design Ontology. However, CAD models, being modeled within this task, are major input for several shape processing tools relevant to AIM@SHAPE.
 - *T3_4_Calculation_and_Analysis*: Task to determine and analyze the properties and expected behavior of a product and its components/parts using various methods, such as Finite Element Analysis (FEA), Computational Fluid Dynamics (CFD), vibrational behavior analysis, strength and stress analysis, etc. in order to optimize material and dimensions.
 - *T3_4_1_Shape_Simplification*: Task simplifying Shape Models (typically CAD models) as preparation for a simulation. Simplification is performed to reduce complexity and improve performance of simulation tasks.
 - *T3_4_2_Meshing*: Calculation of a surface or volume mesh (typically

finite element mesh). Surface meshing in other contexts is also called tessellation or triangulation.

- *T3_4_3_Definition_of_Boundary_Conditions*: When performing a FEA/CFD analysis, boundary conditions on the finite elements need to be defined to define the target of a simulation (force distribution, temperature, flow, ...).
 - *T3_4_4_Solving*: Task performing the actual simulation taking a mesh and the boundary conditions on nodes as input.
 - *T3_4_5_Simulation_Post_Processing*: Post processing of simulation results to support its interpretation. Input are the original shape model, its mesh representation and the simulation results, typically being scalar or vector values on finite element nodes.
- *T4_Product_Evaluation*: Process of executing tests with the prototype of a product or a component. Most of the subtasks of *Product_Evaluation* are not relevant for the AIM@SHAPE Product Design Ontology. However, surface acquisition (measurement) of surface for product evaluation provides a link to the *Acquisition and Reconstruction Ontologies* (see D1.2.2), which is why it is included.
 - *Shape_Acquisition_and_Reconstruction*: See Shape Acquisition and Reconstruction Ontology.

The processes of the *Product Design Ontology*, i.e. the workflow it is modeling, is shown in Figure 5. As stated before, just the first two levels in the task hierarchy have been taken from the STEP standard. To demonstrate the interrelationship with the other AIM@SHAPE ontologies, also the *Shape_Acquisition_and_Reconstruction* task has been incorporated at suitable place within the workflow, also to demonstrate that specific (sub-) tasks can be used within the domain of different (super-) tasks.

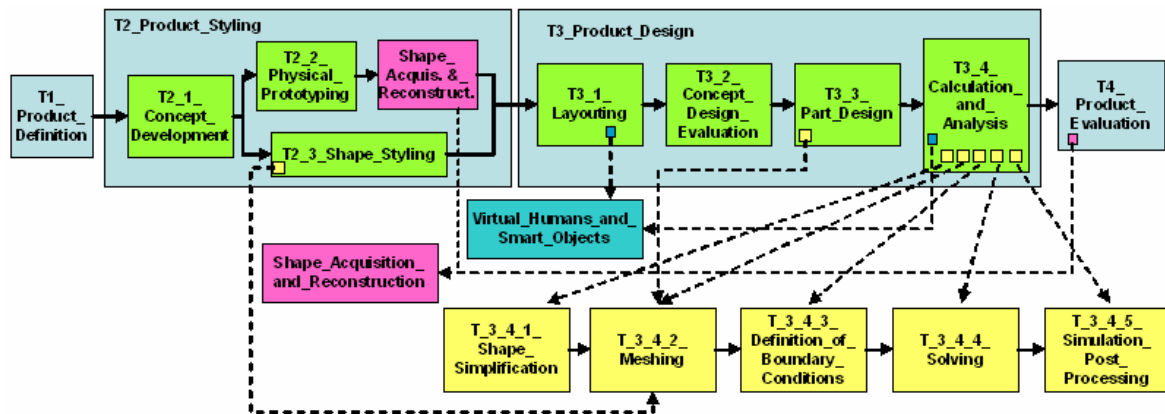


Figure 5: Product Design Tasks as modeled in the Product Design Ontology. Main flow is taken from ISO 10303 (STEP), AP214. The Shape Acquisition and Reconstruction process as well as possible relationships to the Virtual Humans Ontology are included.

Other tasks used within the context of different other tasks are *T3_4_2_Meshing*, which might be a sub-task also of *T2_3_Shape_Styling* or *T3_3_Part_Design* (not yet modeled in detail in the ontology), and the use of concepts of the *Virtual Human and Smart Objects Ontology*, for instance, in *T3_1_Layouting* or *T3_4_Calculation_and_Analysis*, e.g. to describe shape processing pipelines when performing ergonomics analysis.

In the case of the *T3_4_2_Meshing*, the task is currently mainly inserted to describe the activity of surface or volume meshing (to generate finite element meshes) of Shapes given in Boundary Representation or Surface Representation. In comparison to the role of this sub-task in *T2_3_Shape_Styling* or in *T3_3_Part_Design*, meshing is typically a surface meshing to provide a shape representation suitable for visualization, for instance, for Design Review. In this role similar tools might be used for meshing, but e.g. with other surface quality criteria. Also, the terminology of typical users changes from Meshing to Tessellation or Triangulation. With these observations it becomes feasible to distinguish between the different roles of Meshing by inserting a new separate task called, for instance, Tessellation. As any task can be related to any tools and any shapes this extension would be possible without change of the ontology itself, but just by adding a new Task instance and its interrelationships to predecessors, successors, and super-tasks.

Rounding up the presentation of the current Product Design Ontology, Figure 4 shows the interrelationships between the currently inserted task instances. The representation is equivalent to that shown in Figure 5.

4.2 Boundary Condition Type Hierarchy

In the context of simulation, for the meshing task *T3_4_2_Meshing* it is of interest to identify the type of boundary condition guiding a specific mesh generation. The definition

of boundary condition types for the task *T3_4_2_Meshing* is optional, whereas for the task *T3_4_3_Definition_of_Boundary_Conditions* it is mandatory.

However, in order to conceptualize this coherence, the Boundary Condition Type Hierarchy has been introduced into the Product Design cluster to structure the relevant Boundary Condition Types (see Figure 6). Note that several meshing task instances may refer to different boundary conditions type to distinguish specific meshing types on the instance level.

The root concept in this hierarchy is the *BoundaryConditionType* concept and has four direct sub-classes: *StructuralMechanics*, *CFD*, *Electromagnetic* and *Symmetry*. Each one of these sub-classes is sub-divided through further sub-classes that are graphically represented in Figure 6.

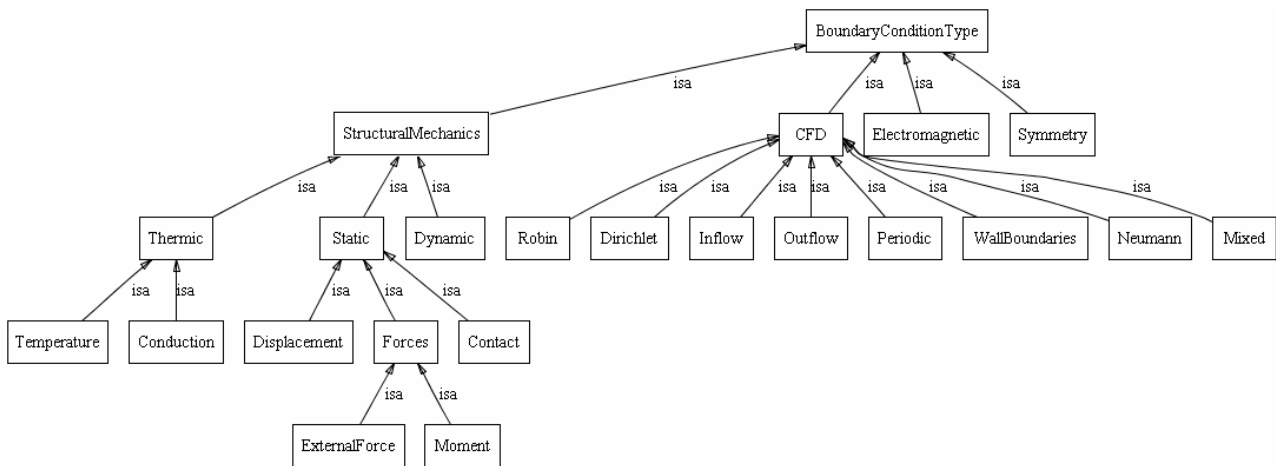


Figure 6: Boundary Condition Type Hierarchy

Each one of these concepts must be instantiated in order to be able to create instance graphs. This is due to the current limitations of the ontology description language OWL, since it does not permit to relate instances and concepts via user-defined properties. There is only one standard property, the *rdf:type* property, that allows to link an instances to its class. However, if this modeling approach were supported, ontology query languages such as nRQL [HMW04] or OWL-QL [FHH03] should be extended in order to be able to manage queries.

4.3 CAD Representation Type

Within the Product Design Ontology, the *CAD Representation* type *CADRepType* has extended the commonly used ShapeType Hierarchy, which has been firstly introduced in Deliverable 1.5.1 “Report on an integrated view of the domain ontologies”. In fact, the *CADRepType* taxonomy is required to model the I/O shape types of tools. In this way, a CQ like “Give me the tools that take manifold non-selfintersecting models as input” can then be answered.

The main properties characterizing CAD models that we want to include are: *Closed*, *Manifold*, *Trimmed*, *Non-SelfIntersecting*, *C1*, *C2*, *G1*, *G2*. In the next version of the ontology, the meaningful subset of all the combinations of these properties will be defined as sub-classes of *CADRepType*; among them, *Manifold_CADRep*, *Non selfintersecting_C1_CADRep*, *Closed_Manifold_C1_Trimmed_CADRep*.

4.4 Tools Concept Extensions

Tools for the Product Design Ontology are any shape processing tools either consuming or producing shape data and supporting a product design task. Classification of metadata about tools (currently meaning tools of Tool Repository, other tools, methods, algorithms, etc.) should be those modeled in the **Tools Repository**.

The only structural extension to the common tools ontology requested is to add the attribute *supports* to the *SoftwareTool* concept, in order to associate tools with tasks.

The proposal is, for instance, that every time a tool is added to the tool repository, also the relationship to tasks supported by the tool should be explicitly inserted, either by selecting one or more existing tasks or by adding a new task instance, if the task context of the tool needs to be more specific.

Having tools and tasks interrelated in this way, competency questions about the relationship of tools and task/workflows can be answered.

4.5 Shape Concept Extensions

Currently no structural extension to the Common Shape Ontology is needed by the Product Design Ontology.

5 FUTURE ONTOLOGY EXTENSIONS AND OPEN ISSUES

This deliverable just provides a snap-shot on the current status of the product design ontology, which will be further developed in the upcoming project period.

Main concepts already identified to be further developed include the following:

- The cluster is planning to introduce the concept of **ShapeRole** for the design ontology to related shape representation instances with its role within the product development workflow, e.g. simulation result. With this it will be possible to associate queries with the occurrence and relevance of a shape within a given workflow. For instance, it is not sufficient to know that a shape is in CAD representation to understand that one is the simplified version of the other. Likewise, both the input and the output to simulation are 2D or 3D meshes. Shape role is intended to distinguish between both.
- The Product Design cluster also identified the need for sophisticated **grouping mechanisms** as they are currently present. Several shape grouping concepts are thought off: shapes that represent the same object, shapes via a chain of shape processing tools have been derived from each other, shapes belonging to the same product category, shapes representing the same object in different versions, different formats, etc.
- The concepts for realizing the scenario on **quality check** of CAD models (see section 3.3.2) need to be developed.

Furthermore, most design **task concepts** may need to be further specialized with specific attributes to extend the knowledge base about the design workflows addressed as target applications of the product design ontology.

An open issue is how to represent the results of **queries on design tasks and workflows**. Currently, query results on tasks and workflows are an unordered list of involved tasks. If multiple workflow paths are returned, all tasks – still unsorted – will be the search result.

What needs to be addressed in the next project period, is how to provide a sorted list of tasks as result of workflow queries. So far the cluster discussed to possible approaches:

1. Extension of the ontology by a workflow concept with sub-concepts like 1-Step workflow, 2-step workflow, ..., n-step workflow, with 1, 2, ..., n attributes to identify the 1st, 2nd, ..., nth task of the workflow.
2. An alternative is an extension of the search engine to adequately present the workflow. All required workflow information is already included in the task concept on the instance level.

REFERENCES

- [FHH03] R. Fikes, P. Hayes, I. Horrocks
OWL-QL - A Language for Deductive Query Answering on the Semantic Web, Knowledge Systems Laboratory, Stanford University, Stanford, CA, 2003, http://ksl.stanford.edu/KSL_Abstracts/KSL-03-14.html
- [HMW04] V. Haarslev, R. Möller, M. Wessel
Querying the Semantic Web with Racer + nRQL
<http://www.sts.tu-harburg.de/~r.f.moeller/racer/papers/2004/HaMW04.pdf>
- [ISO 10303] 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 processes
- [RQL] The RDF Query Language (RQL), FORTH Institute of Computer Science,
<http://athena.ics.forth.gr:9090/RDF/RQL/>