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 D1.2.2.1

Ontology for Acquisition and Reconstruction Processes 1st version

	Circulation: [1]CO
Partner(s):	**DISI (Leader), IMATI, ITI, WEIZMANN, MPII, UNIGE, TECHNION, UU**
Authors:	L. Papaleo, R. Albertoni, F. Robbiano, S. Marini, G. Vasilakis, M. Pitikakis, M. Spagnuolo, L. Moccozet, T. Hassner
Version:	04
Stage:	DRAFT (100%)
Date:	Wednesday, December 21, 2005

---

[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).

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 Institut – 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

Document History

Vers.	Issue Date	Stage	Content and changes
1	30 October 2005	20%	Initial structure and content.
2	20 November 2005	50%	First draft version. Diagrams and References to Protégé file added.
3	8 December 2005	75%	Second draft version. Ontology design section refined and first draft of the Results Section added.
4	15 December 2005	100%	Document completed

Executive Summary

This document contains a description of the deliverable **D1.2.2.1** of the IST NoE AIM@SHAPE.

The deliverable ***D1.2.2.1 – Ontology for Acquisition and Reconstruction Processes 1st Version*** – is intended to provide a first version of the ontology the AIM@SHAPE partners have developed for formalizing the knowledge related to the Acquisition and Reconstruction of Shapes. The task leader is **DISI** and has been supported by all the involved partners.

The deliverable is organized as follows: first a brief introduction to the approached problem is sketched (Section 1), then a section on Ontology design is presented where the OntoKnowledge Approach is briefly described (Section 2). Section 3 presents the details of the Acquisition and Reconstruction ontology showing the differences between this version and the previous one (introduced in Deliverable 1.2.2). Section 4 shows some initial results regarding the introduction of reasoning in the ontology and Section 5 gives some concluding remarks. Section 6 lists the publications related to the Acquisition and Reconstruction ontology published within the network.

Table of Contents

1 INTRODUCTION

The knowledge related to the acquisition of a shape characterizes the lifecycle of the shape itself. The acquisition has to be planned (e.g. the devices have to be chosen taking into account their availability and the characteristics of the object to be acquired) and the data coming from the acquisition systems have to be processed in the reconstruction phase to produce a shape. Further processing has to be done to fulfil the users' needs, and documentation has to be produced in order to enrich the semantic impact of the shape.

The Ontology for Shape Acquisition and Reconstruction is intended to produce knowledge support for researchers who deal with the aforementioned steps.

Due to the intrinsic complexity of shapes, ontology-driven metadata are necessary in order to reach a sufficient level of expressiveness. Metadata should provide a thorough characterization of shapes (**Figure 1**) by storing:

a. the information related to its history, such as the acquisition devices and techniques for creating it or the tools for transforming it (its *past*, e.g. for documentation),

b. the information intrinsically held by the shape itself (its *present*) and

c. the information related to its capabilities and potential uses, such as the possible steps that can be performed or the tools that can be used (its *future*, e.g., for acquisition/process planning).

Moreover, ontology-driven metadata can capture different levels of granularity describing a shape as a *simple resource* (e.g. for cataloguing) and characterizing it according to its *geometry* (e.g. for rendering), to its *structure* (e.g. for matching and similarity), and to what it *represents* (e.g. for recognition or classification). **Figure 3** gives an example of a digital shape and its intrinsic characteristics: it can be seen as a simple resource (e.g. name and URL), or can be considered by its geometric characteristics (e.g. a set of triangles and normals). It has a structure (e.g. the skeleton of a teapot) so it can be viewed as a teapot composed by a handle, a spout, a body and a tip. It is also important to take into account the different environments where the shape can be used since the specific application determines relevant characteristics. For example, if the main purpose is to build a teapot, the identification of parts by which a teapot is composed is fundamental, while if the purpose is to let a robot grasp it, the localization of the handle is the only necessary task.

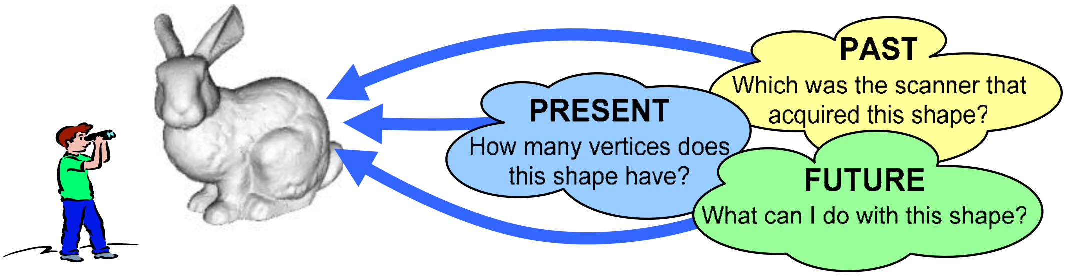


Figure 1 - An expressive characterization of a shape is made up by the information related to its history, the information intrinsically held by the shape itself and the information related to its capabilities.

In the context of the AIM@SHAPE network of excellence, it was decided to enhance the **semantic aspects** of shapes using two different and coexisting approaches.

(1) The first strategy is ***analytic*** and acts on the side of Shape Modelling: development of tools and methods to extract morphological structures from low-level geometry (e.g.: find the skeleton of a shape), and semantic information from structures (e.g. trying to understand where is the handle of a door or the tip of a teapot).

(2) The second strategy is ***synthetic***, and acts on the side of Knowledge Technologies: the domain knowledge and the shape semantics are encoded in context-dependent ontologies, and are used, for example, to annotate and retrieve shapes.

Concerning the synthetic strategy, this document will present the first version of the Ontology for Shape Acquisition and Reconstruction.

In particular, two main paths are considered within the so-called AIM@SHAPE digital shape lifecycle (see **Figure 2**). In the first path, the shape has roots in the real world, from which it is acquired and analysed, in order to produce digital data and semantic knowledge related to them. In the second path, the semantics drive the production of a shape: an idea is handled at the semantic level and then tools are used to concretize it in a real or virtual object. The Ontology for Shape Acquisition and Reconstruction focuses on a part of the first path, i.e. on the phase in which a real object is used to produce a digital shape.

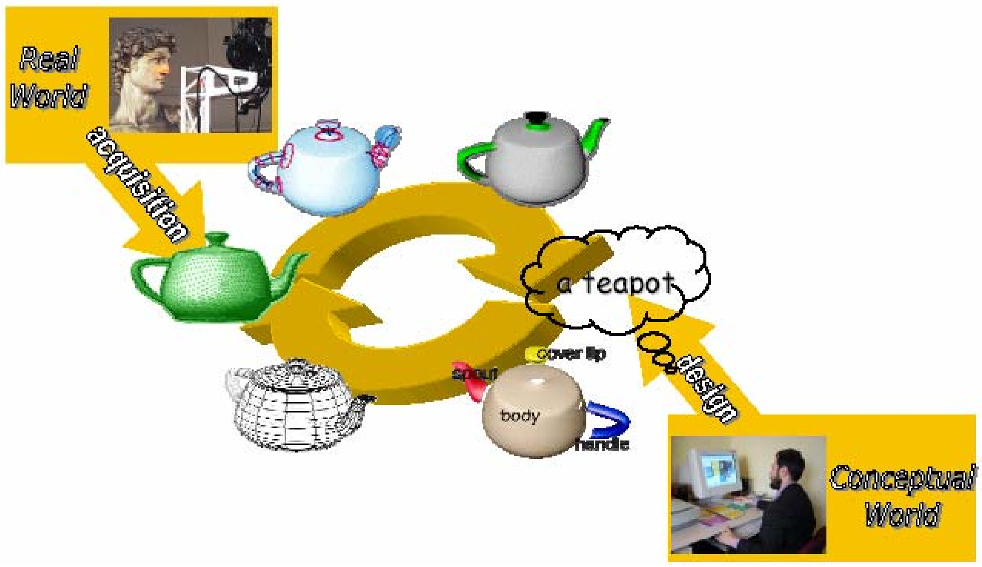


Figure 2 – the digital shape lifecycle.

Deliverable 1.2.1.1 and Deliverable 1.2.3.1 will present the ontologies for Virtual Humans and Product Design, respectively. These ontologies will be used in the DSW to browse the collected resources according to context-dependent views.

2 ONTOLOGY DEVELOPMENT

With “ontology development” is meant the process that produces an explicit and formal specification of a domain of knowledge. As shown in deliverable D1.2.2, the motivation behind the development of an ontology for a particular domain, is to share a common understanding of the knowledge belonging to the domain, while at the same time improving the interoperability among applications that use that knowledge. An ontology is also useful for making explicit assumptions on the domain of knowledge, so that applying changes, as these assumptions evolve, becomes easier and it also enables the re-use of the knowledge extracted from the domain.

To develop the ontology for Shape Acquisition and Reconstruction we have decided to follow the *OntoKnowledge* methodology[2]. This methodology defines an iterative process for the analysis of the domain of knowledge, comprised of four phases, as detailed in the D1.2.2:

(i) Specification of the requirements;

---

[2] http://www.ontoknowledge.org

(ii) Design;

*(iii)*Refinement;

*(iv)*Evaluation and Maintenance

At this version of the ontology for Shape Acquisition and Reconstruction, the requirements *(i)* have been defined and the design phase *(ii)* is advanced with respect to the final goal. At the same time the “Acquisition” part of the ontology has been implemented and populated by several instances providing information on real scanning sessions actually performed by some AIM@SHAPE partners. These instances are used in order to evaluate the ontology with respect to the competency questions defined in the specification of the requirement phase.

The following sections of this deliverable summarize the details of the implementation of the ontology obtained by using the Protégé ontology editor and the first results in answering some basic competency questions.

3 AN ONTOLOGY FOR SHAPE ACQUISITION AND RECONSTRUCTION (V1.0)

The ontology for Shape Acquisition and Reconstruction defines and formalizes concepts and relations among them. **Shape models** and **tools** are treated as resources that can be uploaded and downloaded together with their metadata in the shape and tool repositories respectively.

The domain of the ontology has been defined as the *development, usage and sharing of hardware tools, software tools and shape data by researchers and experts in the field of acquisition and reconstruction of shapes*. The domain was already defined in the preliminary draft of the ontology presented in the Deliverable D1.2.2. It has not changed and the development of the ontology has been based on this domain and the requirements that have been defined.

To specify the Acquisition and Reconstruction pipeline the following macro-steps have been defined:

- *Shape Acquisition* (and Registration): the phase in which sensors capture measurements from a real object;
- *Shaping*: the phase in which all acquired data are merged to construct a single shape;
- *Shape Processing*: the phase in which further computations on the shape may be done (e.g. smoothing, simplification, enhancement, and so on).

As stated earlier, the Acquisition and Reconstruction ontology is intended to be targeted to the scientific community. For this reason experts of the field were interviewed, within AIM@SHAPE, to understand the requirements and to create the competency questions. The initial non-formal competency questions have been presented in deliverable D1.2.2 (see also Table 1). In this deliverable we will present the refined competency questions (see Section 3.3).

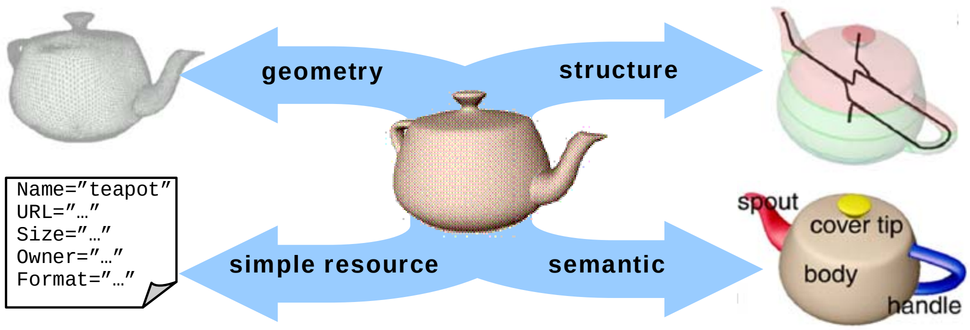


Figure 3 - A shape is described as a simple resource, or by its geometry, its structure, its semantics, depending on the application domain

From the feedback obtained, it was clear that an important landmark of this ontology would have been the conceptualisation of the *Acquisition Session*, with the main aim of planning the acquisition of real objects and annotating the shapes by documenting their acquisition. That is why, during this first phase of activities, we have been mainly concentrated in the formalization of the knowledge related to Shape Acquisition.

Moreover, it is important to remind that a proper conceptualization of shapes, tools, and publications is fundamental not only for their own characterization but also to provide meaningful cross-correlations.

In the following we will present the preliminary work (see deliverable D1.2.2 for more details) and the first version of the Acquisition and Reconstruction ontology. During the ontology design we started concentrating in the Shape Acquisition part which is the first out of three steps that have already been identified. Shaping and Shape Processing are steps in which conceptualization among shapes, tasks and processes are necessary. This part of the ontology is still under development and will be completed in the next version of the ontology [see Section 3.4.3].

The consortium has decided to specify all the cluster ontologies in the Web Ontology Language (OWL[3]). 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[4] plug-in.

3.1 Target Applications

The target applications of the Ontology for Shape Acquisition and Reconstruction are initially related to:

- *Acquisition Planning*: the process of selecting and interconnecting the most suitable acquisition devices to the purpose of obtaining a satisfactory 3D acquisition. Properties of the object to be acquired and possible requirements on the quality of the acquisition can eventually influence the decision.
- *Data validation*: Determine if data are accurate, complete, or meet specified criteria.
- *Benchmarking*: Activities related to the comparison of aspects of performance (functions or processes) with best practitioners; identifying gaps in performance; following through with the implementation of improvements; and following up by monitoring progress and reviewing the benefits.
- *Testing*: Testing the functionality and correctness of a process (algorithm, method, approach) by executing it. Testing is usually performed for one of two reasons: defect detection or reliability estimation.
- *Multi-sensors Data Fusion*: The study of the means and tools for the integration of data coming from multiple sensors.
- *Data enhancement* (automatic Recovery): the process of enhancing the quality of the data with respect to some specific constraints and requirements.

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

The identification of several key entities during the first stages of the development of the ontology for shape acquisition and reconstruction [see deliverable D1.2.2] has supported the definition of concepts, relations and attributes of the ontology. Many of the key entities identified have not become concepts in the first version of the ontology also because their relevance with respect to the refined competency questions was not so high. Instead, they

---

[3] http://www.w3.org/TR/owl-features/

[4] http://protege.stanford.edu/plugins/owl/

have been added as attributes of concepts or relations among concepts.

What follows is a detailed description of all the concepts present in the first version of the ontology. These concepts comprise the evolution of the preliminary concepts presented in deliverable D1.2.2. They have been refined and better structured, looking also at the Shape Repository metadata. For each concept, the definition has been validated with respect to the one present in the AIM@SHAPE Glossary (deliverable D1.2.1).

AcquisitionConditions
It models the conditions under which a given acquisition session has been performed.

AcquisitionDeviceType
It models the different types of acquisition devices.

CameraDevice
It models the characteristics of the camera device.

CTScanDevice
Computer Tomography Scan Device. Applications: Medicine. (see DICOM as standard output shape).

HapticDevice
A haptic device is a device which allows a user to interact with a computer by receiving tactile feedback. This feedback is achieved by applying a degree of opposing force to the user along the x, y, and z axes. There are two main types of haptic devices: glove or pen-type devices that allow the user to "touch" and manipulate 3-dimensional virtual objects and devices that allow users to "feel" textures of 2-dimensional objects with a pen or mouse-type interface.

LaserDevice
Any of several devices that emit highly amplified and coherent radiation of one or more discrete frequencies.

LinearCamera
A video camera where the CCD is a linear array sensor, i.e. a sensor where the sensing elements are in a 1D line.

StructuredLightDevice
A device which uses a light pattern projected at a known angle onto a scene. Observing the lateral position elements of the light pattern in the image reveals depth information.

TimeofFlightDevice
This scanning technology refers to device that calculate distance (3d point position in space) by measuring the time of flight of very short pulses of light.

AcquisitionSession
It identifies the set of conditions, acquisition system and other necessary information used for a session of scanning.

AcquisitionSystem
The set of acquisition devices used for a session of scanning.

RealObject

AcquisitionDevice
A system of sensors connected to a storage device (usually a PC), designed for acquiring data.

RealObjectforAcquisition
any real object is eligible to be scanned.

BoundingBox
In computer graphics and computational geometry, a bounding volume for an object is the minimal closed volume that completely contains the object.

Error
Class which represents the different Types of error coming up from the process.

ErrorDeclaredByProducer
Scan error declared by the scanner producer under given conditions.

Institution
The general definition of an institute, department, company and so on.

Person
The general definition of a person.

ShapeData
Shape Data are created, modified, built, converted, managed during the three main steps of our pipeline: Acquisition (and Registration), Shaping and Post-Processing.

ImplicitSurface
In contrast to parametric surfaces, an implicit surface is defined as the set of points P in space verifying an implicit equation (f (P) – constant = 0). f is called the “field function” (and sometimes the “implicit function”, which is improper since this function is explicitly given by its parametric equation).

LevelSet
The level set of a real function f is the pre-image through f of a constant value t in the domain of f. The level sets are also called contours or iso-levels. Level sets may be non-connected (synonymous: contour).

LinearScan
The output of a linear camera acquisition device.

Mesh
An Euclidean cell complex such that any k-cell of G, with k<d, bounds at least one d-cell of G. A --simplicial-- mesh is a mesh in which all cells are simplices. A d-dimensional simplicial mesh is a simplicial mesh which contains simplices of dimension d or lower.

SurfaceMesh
A surface mesh is a two-dimensional cell complex embedded in the three-dimensional Euclidean space. Usually, it is a complex with a manifold domain, and the 2-cells are either triangles or quadrangles.

VolumeMesh
A volume mesh is a three-dimensional cell complex embedded in the three-dimensional Euclidean space. Usually, it is a complex with a manifold domain, and the 3-cells are either tetrahedra or cubes. It is used to denote the decomposition of the volume of a solid 3D object or of the domain 3D scalar field.

PointsCloud
A set of uncorrelated points, usually in 3D, that has to be further elaborated to obtain a

3D model.

EnrichedPointsCloud
A set of uncorrelated points, usually in 3D, that has to be further elaborated to obtain a 3D model (as a point cloud) plus some additional information associated to each point. Additional information can be color, confidence of the measure, texture, normals and so on.

RangeImage
A grid of distances (range points) that describe a surface in Cartesian (height field) or cylindrical coordinates. A range scanner senses 3D positions on an objects surface and returns an array of distance values.

Raster Data
A n-dimensional grid of measures (generally it could be 2D or 3D).

RasterData2D
A raster data made of XxY positive integers.

Image
An image is a 2D raster data that have in general RGB trio values.

DepthMap
A depth map is a 2D raster data that can have non-integer and negative values (float) which can use more than 8-bits.

RasterData3D
A raster data made of XxYxZ positive integers.

Video
The electronic representation of a sequence of images, depicting either stationary or moving scenes.

MRIVolume (DICOM)
Short for Digital Imaging and Communications in Medicine, a standard in the field of medical informatics for exchanging digital information between medical imaging equipment (such as radiological imaging) and other systems, ensuring interoperability. The standard specifies: a set of protocols for devices communicating over a network, the syntax and semantics of commands and associated information that can be exchanged using these protocols, set of media storage services and devices claiming conformance to the standard, as well as a file format and a medical directory structure to facilitate access to the images and related information stored on media that share information.

Weight
It identifies a weight in a given unit measure.

Pixel
The unit of RasterData.

RasterDataAggregateHierarchy
Represents the full hierarchy of aggregates, and the parameters used to create them.

RasterDataAggregateList
List of aggregates of pixels in the current level of the hierarchy.

RasterDataAggregate
A list of pixels (aggregate of pixel). Note that at the finest level of the pyramid, an aggregate represents a single pixel; at the coarsest scale only a single aggregate exists – the whole RasterData.

3.3 Refinement of the competency questions

Starting from the initial CQs obtained from the partners involved (see Table 1 and Deliverable D1.2.2) we refined them, checking consistency of terms and names with respect to the refined and new concepts modeled in the ontology. Some of the initial CQs were too vague and too ambitious, some of them were not related to what an ontology can do (e.g. *Do two shapes have compatible representations?*). Ths activity has produced a list of more precise competency questions.

Initial Informal Competency Questions

1. Do two shapes have compatible representations?
2. If not how should one shape be converted in the other shape?
3. In what respect are two solid objects similar in shape?
4. Should the ontology classify shapes according to the shape or to the representation type of the shape, or both?
5. Which are the characteristic datasets that can be *problematic or significant* for a specified algorithm?
6. This dataset is problematic or significant for this algorithm/process?
7. Is This algorithm performant?
8. Given an algorithm and a Dataset, is there a Process able to use the algorithm with the dataset?
9. Given this model, from where does its information come from?
10. Given this dataset, how are its data distributed?
11. Given this dataset, which is the spatial distribution of its data?
12. Given this dataset, is there a process able to get information on the distribution of its data?
13. Which are the processes that are able to integrate data coming from multiple sensors?
14. Which are the possible open research fields in relation to this process?
    a. Are there any methods without algorithmic implementations?
    b. Is there a path (simple or composite) between these two concepts in the ontology?
    c. Is there a non-instantiated concept?
    d. Given this ontology-state is there a next step able to bring me to this other state?
15. Can I have global information on this composite process? (Price, performance, global error, etc.)
16. Are there any processes or algorithms able to recover or preserve this characteristic (sharp features, etc.)?
17. Given these initial conditions which scanning devices can be used?
18. Given this geometric model, do I have enough information on its accuracy in order to be able to simplify it at an *x-level* without losing information?
19. Can I convert this representation to this other?
20. Is this method/process applicable?
21. Similarity estimation (can this model possibly represent a human being?)
22. Has the model any proprietary rights?
23. Is the model representing a real world object or a synthesized object?
24. Is the model manifold and/or orientable?
25. Can the model be converted into a volumetric representation?

Table 1 – Initial Informal Competency questions.

The refined CQs are listed in the following:

1. **Questions regarding the Acquisition phase**
    a. What is the minimal price of an Acquisition System able to scan the Real Object "little bird"?
    b. What are the Acquisition Systems able to scan a Real Object which is transparent?
    c. What are the Acquisition Systems able to scan the Real Object "little bird"?
    d. What is the price of the Acquisition System "IMPACT"?
    e. What is the cheapest Acquisition System in the KB?
    f. What is the price of the Acquisition Session "Pisa Miracles Square, May 10"?
    g. What are the Acquisition Systems that are made up by more than one Acquisition Device?
    h. Is it possible to scan the "marble horse" within the Logistic Conditions "DISI LAB"?
    i. How is it possible to scan the "marble horse" within the Logistic Conditions "DISI LAB"?
    j. Are the laser scanners able to scan transparent objects?
    k. What tricks have to be performed in order to be able to scan the Real Object "little bird" with the Acquisition Systems available at DISI?
    l. What Acquisition Systems are available at DISI?
    m. What Acquisition Devices constitute the Acquisition System "IMPACT"?
    n. What tricks have to be performed in order to be able to scan the Real Object "little bird" with the Acquisition Device "Minolta XYZ"?

2. **Questions regarding tools / algorithms using Shape Data**
    a. What is the complexity (time) of this algorithm?
    b. What is the complexity (space) of this algorithm?
    c. Does a convertor exist able to convert the format PLY in the format OFF?
    d. What characteristics of the Shape Data "horse.ply" are relevant for the algorithm "MC" (Marching Cube)?
    e. Given the algorithm "Algo1" (with input constraints such as format, water tightness) and the Shape Data "tire.off" is there a process able to apply the algorithm to the Shape Data satisfying the constraints?
    f. What processes are able to integrate data coming from multiple sensors?
    g. What open research fields are related to this process?
    h. Are there algorithms able to preserve sharp features?
    i. Given the Shape Data “little bird”, is the information on its accuracy sufficient to simplify it up to a given level and without losing relevant information?
    j. What are the possible steps to do starting with a points cloud?
    k. What are the possible steps to do starting with the points cloud "cloud1.txt"?
    l. Through what steps is it possible to reach a mesh?
    m. Through what sequence of steps is it possible to reach a mesh starting from a

points cloud?

n. Through what sequence of steps is it possible to reach a mesh starting from the points cloud "cloud1.txt"?
o. What parameters were used to segment this image with the “SWA” tool?

3. **Questions regarding Shape Data and Real Objects**
    a. Does the Shape Data "Pisa Cathedral" have any proprietary rights?
    b. Who is the owner of the Real Object "Black Tire"?
    c. What is the history related to the model "littlebird.wrl"?
    d. Is the Shape Data "little bird" representing a Real World Object or a Synthesized model?
    e. Is the Shape Data "little bird" manifold?
    f. Is the Shape Data "little bird" orientable?
    g. Can the Shape Data "bird.ply" be converted into a volumetric representation?

    Questions regarding images as single objects:

    h. Which are the colored images?
    i. Which images have dimensions M by N?

    Questions regarding the contents of images using both automatic and user specified information:

    j. What images contain “horses” (faces, cars, etc.)?
    k. What other shapes do images with “horses” contain?
    l. What sub-shapes are contained in a “face” shape? What sub-shapes are contained in the sub-shapes? Etc.
    m. What images contain “elongated objects”?

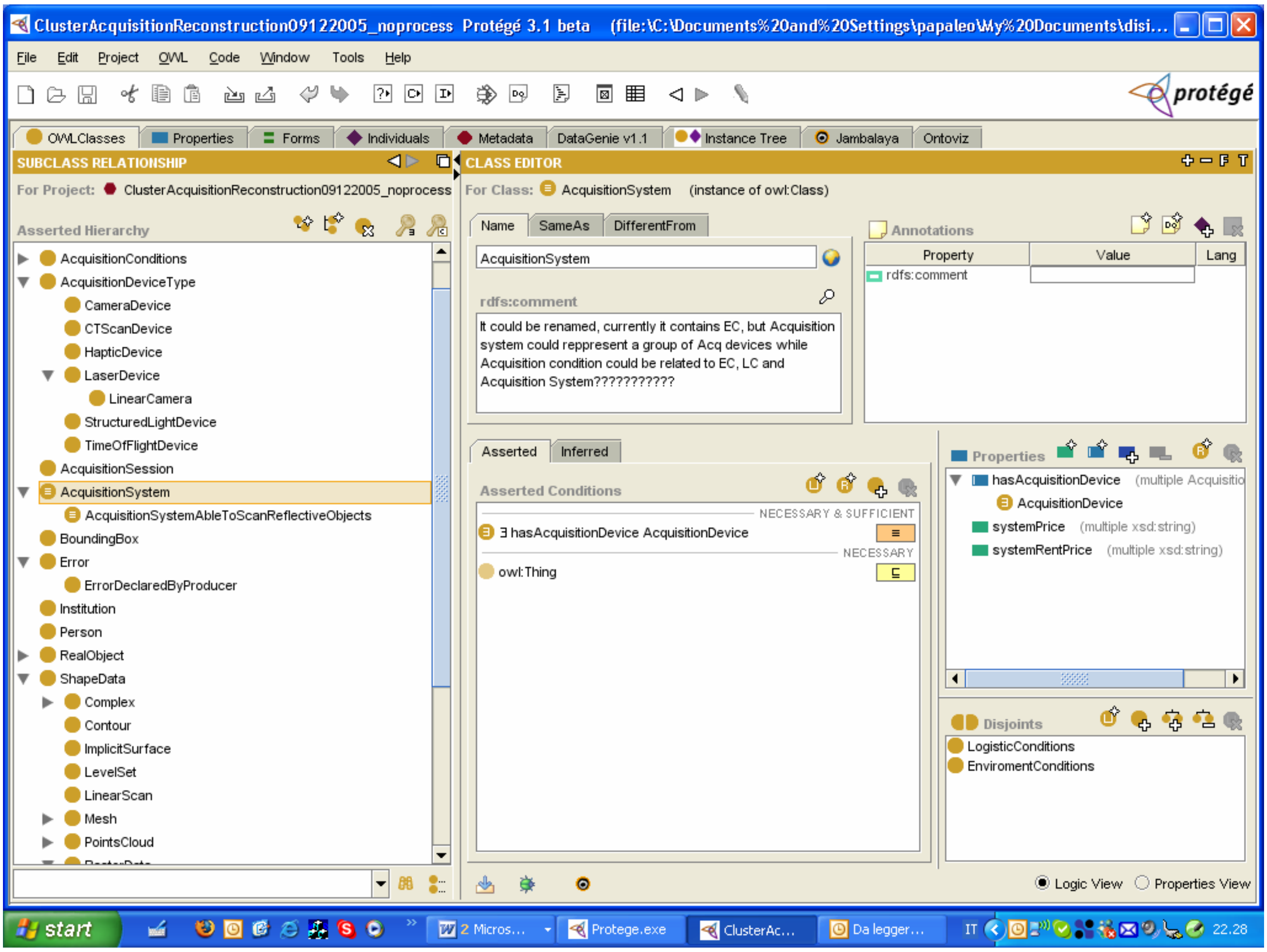
3.4 Ontology Design and Evolution

In this section we will present in more detail some basic concepts defined in the first version of the ontology, presenting also relations and attributes. In particular we will present concepts aiming at modelling the knowledge related to an Acquisition Session, those aiming at modelling the knowledge related to digital shapes and the initial work done for modelling the knowledge related to tasks and processes involving shapes. All these concepts and relations have been inserted in an OWL file using Protégé. **Figure 4** shows the Protégé interface and the classes already created. The concept *AcquisitionSystem* is highlighted.

3.4.1 Modeling the knowledge related to the Acquisition Session

The *AcquisitionSession* has been modelled as a concept in the AR ontology. It is related to an *AcquisitionSystem* (which is made up by one or more *AcquisitionDevices* – via the *hasAcquisitionDevice* relation - e.g. scanners) with a *hasAcquisitionSystem* relation and to the *AcquisitionConditions* in which the acquisition is performed. These *AcquisitionConditions* can be *LogisticConditions* (they include the presence of lights, if there exist any obstacle between the real object and the scanning device and so on) via the *hasLogisticConditions* relation or

EnvironmentConditions (which include the information on where the real object is – indoor or outdoor or underwater - or the level of humidity or even the weather) via the relation *hasEnvironmentConditions*.



AcquisitionSystem

It could be renamed, currently it contains EC, but Acquisition system could reppresent a group of Acq devices while Acquisition condition could be related to EC, LC and Acquisition System???????????

∃ hasAcquisitionDevice AcquisitionDevice

owl:Thing

hasAcquisitionDevice (multiple Acquisitio
AcquisitionDevice
systemPrice (multiple xsd:string)
systemRentPrice (multiple xsd:string)

LogisticConditions
EnviromentConditions

Figure 4 - The ontology for shape acquisition and reconstruction loaded in Protégé

Moreover, some attributes are directly related to the *AcquisitionSession* (e.g. the price for renting the technological devices), while other are related to the different entities in the framework (e.g. the price of a scanning system, or the person/institute responsible for it). An overview on the conceptualisation of the Acquisition Session is given in **Figure 5** where each rectangle represents a concept. The rows in each concept represent a slot which can be either an attribute or a relationship. For each attribute the type is specified, while for each relationship the range is indicated. Whenever a symbol '*****' appears next to the name of an attribute or a relationship, the cardinality can be more than 1.

An *AcquisitionSession* basically documents the acquisition of a *RealObject* and the production of a *ShapeData* (a digital shape), using a particular *AcquisitionSystem*.

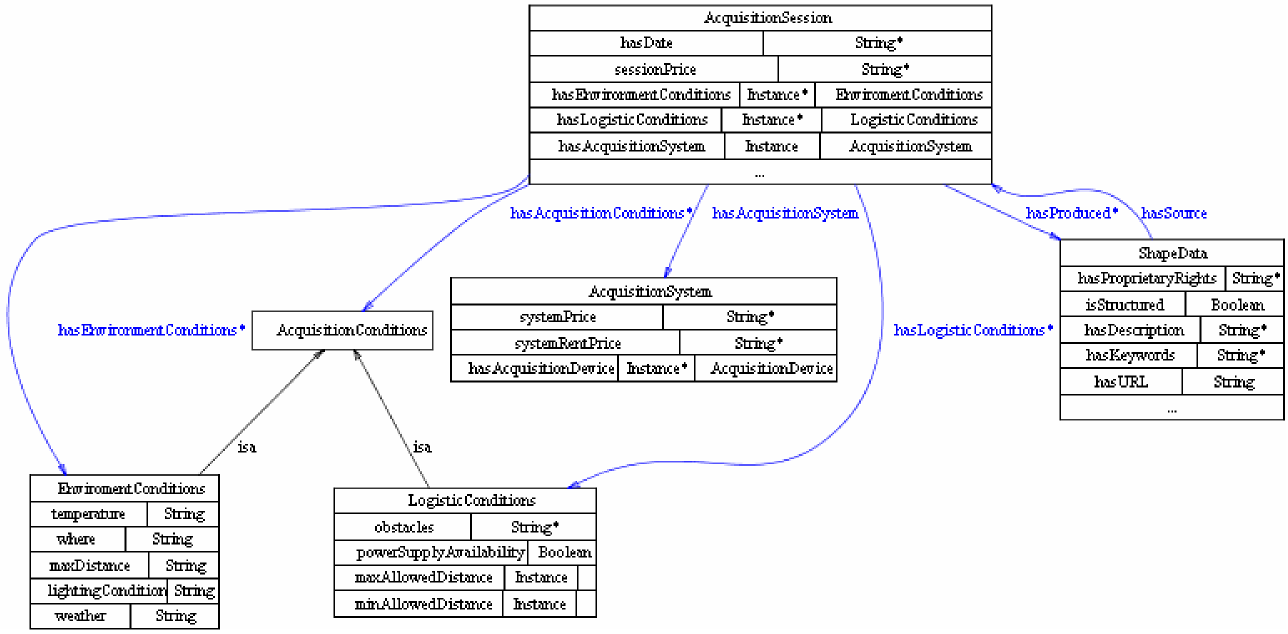


Figure 5 – a diagram showing the relations among different concepts in the Acquisition and Reconstruction ontology related to the acquisition session.

3.4.2 Modeling the knowledge related to Shape Data

ShapeData has also been modelled as a concept in our ontology, with some properties, such as its format or its URL, its Description, but also the information on the source from which it has been generated (through the slot *hasSource*) or the information on the owner of the particular shape (through the slot *hasOwner*): the owner can be identified via an *Institution* or a *Person*.

A ShapeData can be based on another ShapeData (also more than one) or a ShapeData can be used to generate a new one. The relation *isDerivedFrom* formalizes the knowledge related to the **History** of a given shape.

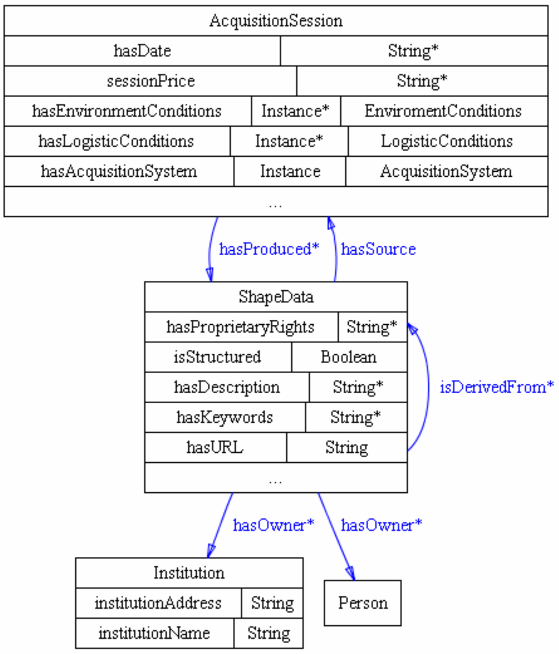


Figure 6 – ShapeDataConcept and its relation with the AcquisitionSession concept.

The metadata of each shape data and the associated definition have been validated with respect to the Glossary and the Shape Repository metadata. In particular three subclasses that were present in Deliverable 1.2.2 (*AcquisitionInput*, *AcquisitionOutputShapingInput* and *ShapingOutputPostProcessingIO*) have been removed because they resulted to be not useful for the overall ontology. **Figure 7** shows a portion of the taxonomy of shapes present in the ontology for shape acquisition and reconstruction.

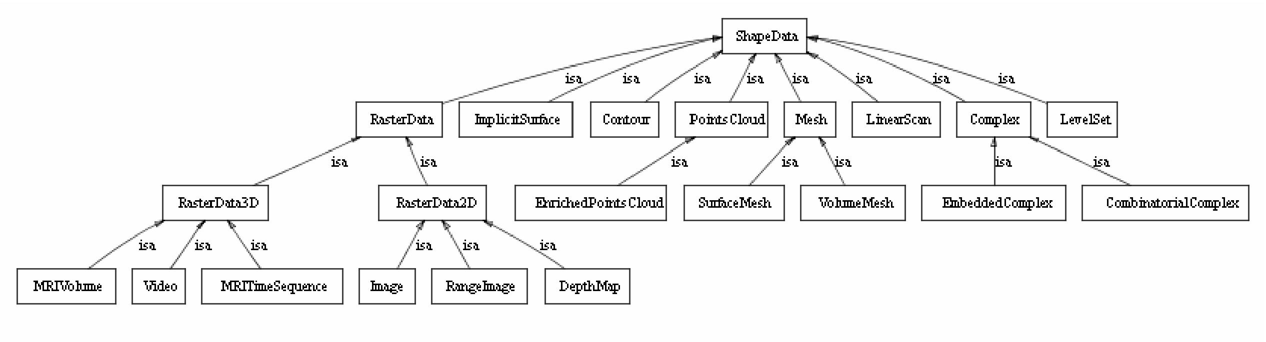


Figure 7 - ShapeData Taxonomy, validated with respect to the shape repository metadata

In this first version of the ontology, particular attention has been given also to the *RasterData* class which, with respect to the CQs and to the interests of the involved partners, should support the knowledge related to a *hierarchy* of *RasterData* associated to a *RasterData*. This schema was not present in the previous ontology presented in Deliverable D1.2.2.

In particular, a *RasterData* can have a *RasterDataAggregateHierarchy* associated (via the relation *hasRasterDataAggregatehierarchy*). A *RasterDataAggregateHierarchy* is made up of a list of aggregates (set of aggregate pixels). It has a *RasterDataAggregateList* (via the *hasRasterDataAggregateList* relation) which is composed by *RasterDataAggregate* (via the *hasRasterDataAggregate* multiple relation). A *RasterDataAggregate* can have neighbours *RasterDataAggregate* (via the multiple relation *hasRasterDatAggregateaNeighbour*) and can be related to its children and its parents in the hierarchy (via the *hasRasterDatAggregateaChild* and *hasRasterDatAggregateaParent* multiple relation).

The relation *hasRasterDatAggregateaNeighbour* captures the relationship between two aggregates in the same scale, the relation *hasRasterDatAggregateaChild* captures the relationship between current aggregate and aggregates in finer scale and the *hasRasterDatAggregateaParent* relation captures the relationship between current aggregate and aggregates in coarser scale The data class hierarchy and its relationship to the segmentation classes are described in **Figure 8**. We are currently validating this schema and inserting it in the ontology protégé file.

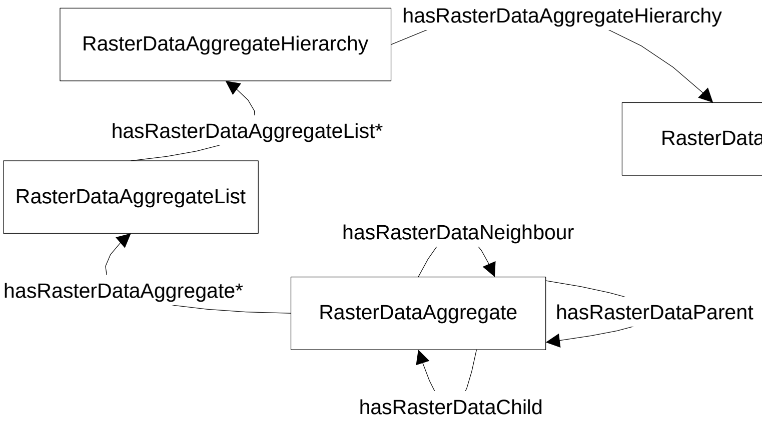


Figure 8 – Class relationship for the RasterData Hierarchy Part

Taken the ontology fragment related to *AcquisitionSession* and *ShapeData* as an example, it can be shown that our ontology is able to support in obtaining the knowledge associated to a digital shape, such as the description of what we called its past, its present and its future. For example, an instance of *AcquisitionSession* includes information about the scanner used to acquire a real object constituting important documentation about *ShapeData*'s past. Assuming that the type of the produced Shape Data is a *SurfaceMesh,* we can focus on some information intrinsically held by itself (and so related to its present)*, e.g.* the number of vertices or faces.

At the same time, the number of vertices of a *SurfaceMesh* can be useful to plan future steps. For example, if we are interested in a surface mesh with at least 10.000 vertices and we found a surface mesh with 3.000 vertices, we could decide to plan a new *AcquisitionSession* increasing the accuracy of the *AcquisitionSystem*. In this case, the ontology supports the planning of a new *AcquisitionSession* providing information such as which *AcquisitionSystem*s are available (indicating also the owners of them), the prices to rent these systems, and so on. The concepts represented in the ontology, being selected according to the experts' skills, provide the right expressiveness to describe and to gather the resources.

3.4.3 Modeling the knowledge related tasks, tools and shapes

Some of the competency questions identified by the involved partners pointed out that a formalization of the relations among tasks, tools and shapes is necessary, such as for example:

1. What are the possible steps to do starting with a points cloud?
2. What are the possible steps to do starting with the points cloud "cloud1.txt"?
3. Through what steps is it possible to reach a mesh?
4. Through what sequence of steps is it possible to reach a mesh starting from a points cloud?
5. Through what sequence of steps is it possible to reach a mesh starting from the points cloud "cloud1.txt"?

An initial solution to formalize this knowledge was already present in the initial version of the ontology (see deliverable D1.2.2) but it was insufficient. We have developed new diagrams and conceptualizations that are actually under development and validation.

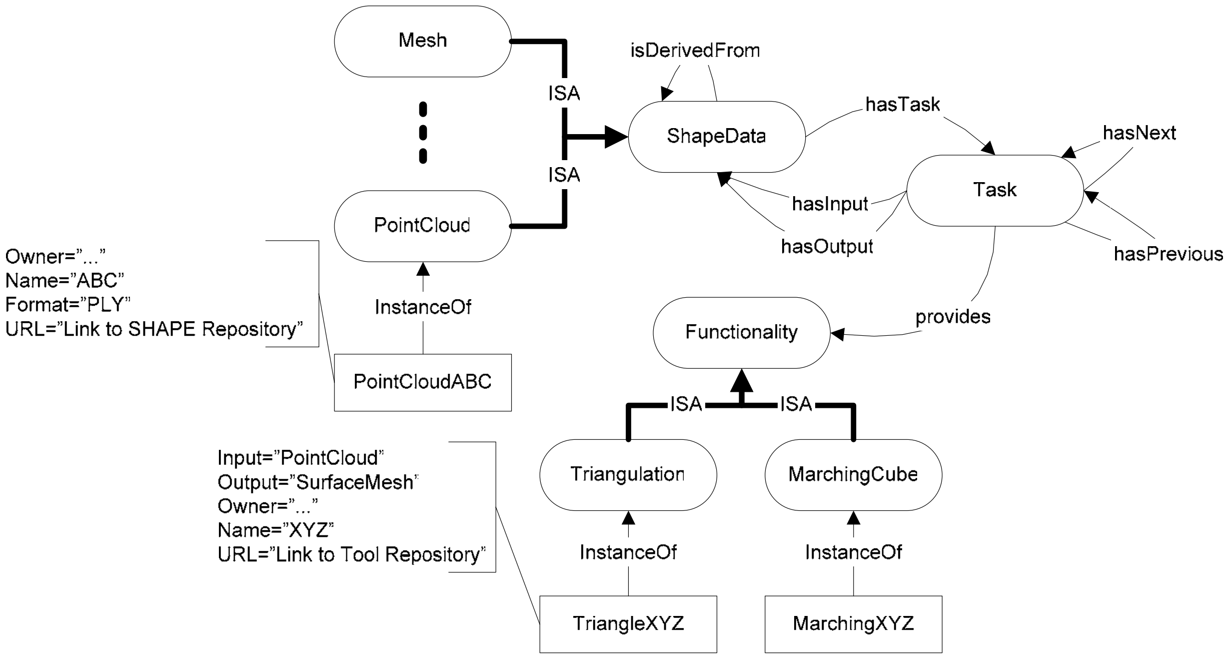


Figure 9 – A diagram showing the initial tentative to formalize the knowledge related to shapes and tools.

Figure 9 shows the first diagram we have identified. In this diagram, *ShapaData* is connected to the concept *Task* via the *hasTask* relation. Each *Task* has, as input or output a *ShapeData* (via the relation *hasInput* ad *hasOutput*, respectively). Each *Task* also is connected to a previous or next task (via the *hasPrevious*, *hasNext* relations) and provides a well-defined functionality (via the relation *provides*). The concept *Functionality* represents all the possible different tasks a tool can provide (Triangulation, Marching Cube and so on…). Instances of these functionalities can have references to specific tools in the tool repository.

4 FIRST RESULTS BY USING REASONING

In this paragraph we are going to illustrate a simple example of how the reasoning can be adopted within the Acquisition and Reconstruction cluster. The adoption of OWL as ontology language allows taking advantage of the more expressive modelling features related to Description Logic. Different reasoners are available for OWL (e.g. Pellet[5], RACER[6]) and, currently the acquisition and reconstruction ontology has been tested using RACER 1.7.16.

Protégé-OWL can play as front-end to an OWL reasoner in order to:

- check the ontology consistency,
- classify the taxonomy,
- compute inferred types.

In particular, by checking the ontology consistency, it is possible to figure out if any defined class results to be incosistent, while the classification of the taxonomy allows to automatically classify the ontology structure. This could be useful in order to infer that a class is a subclass of some other class, without this relation being explicitly stated in the ontology. The computation of inferred types accomplishes the same when instances have been inserted in the Knowledge Base.

In order to practise with the reasoning capabilities, a simple example has been developed

---

[5] http://www.mindswap.org/2003/pellet/index.shtml

[6] http://www.sts.tu-harburg.de/~r.f.moeller/racer/

based on the classes *AcquisitionDevice* and *AcquisitionDeviceType*. As illustrated in **Figure 10**, different acquisition devices are represented as instances of the class *AcquisitionDevice* whereas the acquisition device characteristics are represented as instances of *AcquisitionDeviceType*. An acquisition device can be adopted to scan a object essentially if its characteristics match to the Object properties.

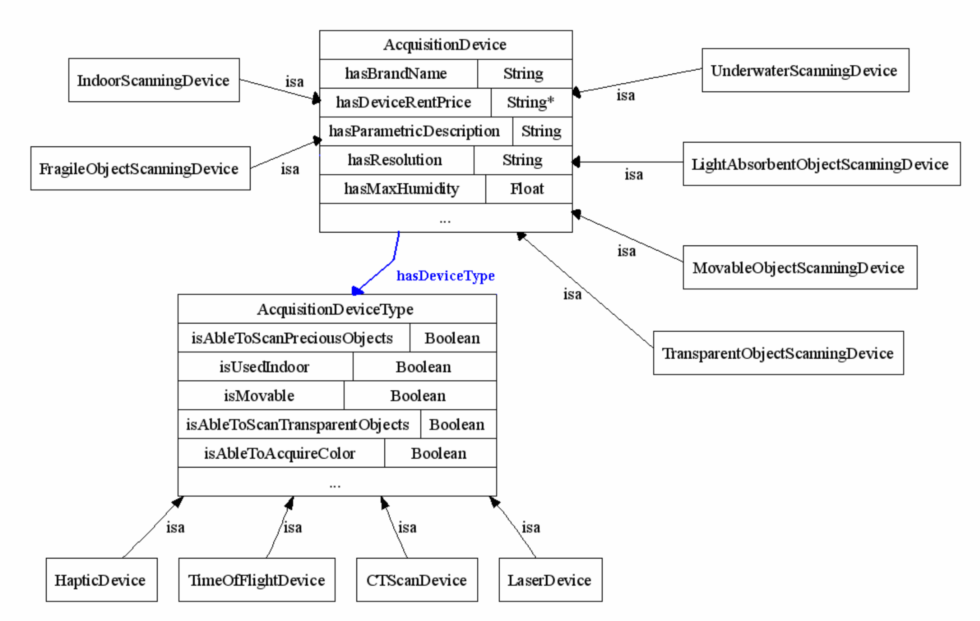


Figure 10 – The ontology schemata for the classes AcquisitionDevice and AcquisitionDeviceType.

The developed example aims at exploiting the reasoning for classifying the scanners according to the characteristics which affect the adoption of particular devices. In order to group the acquisition device instances into a helpful acquisition devices structure, different subclasses of *AcquisitionDevice* are defined and characterized by OWL axioms.

- **Class ColorAcquiringScanningDevice** It represents the class of devices which are able to acquire object shapes as well as their colours.
    - Sufficient and Necessary Conditions:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToAcquireColor ∋ true))

We explain in detail now the axioms characterizing this first class above: the meaning of axioms for the other classes listed below is quite similar and the explanation rather intuitive. The two above axioms state that an instance should be classified as instance of the class *ColourAcquiringScanningDevice* if it is an instance of *AcquisitionDevice* and it has an *AcquisitionDeviceType* able to acquire colors.

- **Class FragileObjectScanningDevice** It represents the class of devices which are able to acquire object that are fragile.
    - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToScanFragileObjects ∋ true))

- **Class IndoorScanningDevice**: It represents the class of devices which can be used indoor.
    - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isUsedIndoor ∋ true))

- **Class LightAbsorbentObjectScanningDevice**: It represents the class of devices which can be used to acquire light absorbent Objects.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToScanLightAbsorbentObjects ∋ true))

- **Class MovableObjectScanningDevice** It represents the class of movable devices.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isMovable ∋ true))

- **Class OutdoorScanningDevice**: It represents the class of devices which can be used outdoor.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isUsedOutdoor ∋ true))

- **Class PreciousScanningDevice**: It represents the class of devices which can be used to acquire objects which are precious.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToScanPreciousObjects ∋ true))

- **Class ReflectiveObjectcanningDevice**: It represents the class of devices which can be used to acquire objects which are reflective.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToScanReflectiveObjects ∋ true))

- **Class TrasparentObjectScanningDevice**: It represents the class of devices which can be used to acquire objects which are transparent.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToScanTransparentObjects ∋ true))

- **Class UnderWaterScanningDevice**: It represents the class of devices which can be used to acquire objects laying under the water.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isUsedUnderwater ∋ true))

- **Class WaveAbsorbentObjectScanningDevice**: It represents the class of devices which can be used to acquire wave absorbent objects.
  - Sufficient and Necessary Condition:


AcquisitionDevice
∃ hasDeviceType (AcquisitionDeviceType ⊓ (isAbleToScanWaveAbsorbingObjects ∋ true))

For each instance of acquisition device, the reasoner is able to infer the proper classes it will belong to, thanks to the axioms presented above and by exploiting the class inference. In particular, in our example, all the defined axioms are sufficient and necessary conditions: this means that given a class **C** and an instance **I**, where **I** is an instance of *AcquisitionDevice*, if **I** satisfies the axioms used to characterize **C**, then **I** can be automatically asserted to be an instance of **C**.

In the ontology for shape acquisition and reconstruction, some acquisition devices and their related acquisition device types have been inserted as instances of *AcquisitionDevice* and *AcquisitionDeviceType,* respectively. The insertion has been done according to the characteristics illustrated in the following two tables (**Table 2** and **Table 3**).

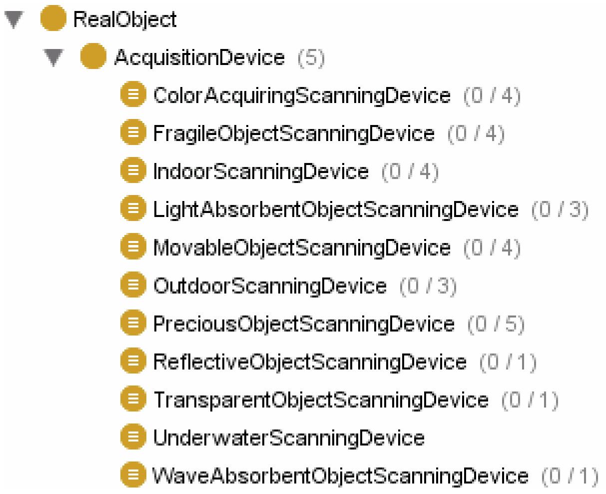
Name	***hasBrandName***	***HasMechanicalSupport***	***hasDeviceType***	***hasOwner***
IMATI_Scanner	Konica Minolta	Tripod	MinoltaVivid910	IMATI
UU_Scanner	Qubic	-	Roland_LPX_250	UU
DISI_Scanner	DISI-SPLIGURU	Conveyor-belt	IMPACT	DISI
Ad1	-	-	HSD_2005	Fake
Ad2	Philips	-	Philips1	Fake

Table 2 -The instances inserted in AcquisitionDevices together with some attributes hasDeviceType and hasOwner which are relationships

AcquisitionDeviceType	**IsAbleToAquireColor**	**IsAbleToScanFragile Object**	**IsUsedIndoor**	**IsAbleToScanLightAbsorbentObject**	**IsMovable**	**IsUsedOutdoor**	**IsAbleToScanPreciousObjects**	**IsAbleToScanReflectiveObjects**	**IsAbletoScanTransparentObjects**	**IsUSedUnderware**	**IsAbleToscanWaveAbsorbingObjects**
MinoltaVivid910	true	True	true	false	true	true	true	false	-	false	-
Roland_LPX_250	false	True	true	true	false	-	true	false	false	false	-
IMPACT	true	True	true	true	true	true	true	false	false	false	true
HSD_2005	true	-	false	-	true	true	true	-	-	-	-
Philips1	true	True	true	true	true	false	true	true	true	-	-

Table 3 - The instances inserted in *AcquisitionDevicesType* and the values of the relative attributes.

The available Acquisition Devices are classified as shown in **Figure 11**.



RealObject
AcquisitionDevice (5)
ColorAcquiringScanningDevice (0 / 4)
FragileObjectScanningDevice (0 / 4)
IndoorScanningDevice (0 / 4)
LightAbsorbentObjectScanningDevice (0 / 3)
MovableObjectScanningDevice (0 / 4)
OutdoorScanningDevice (0 / 3)
PreciousObjectScanningDevice (0 / 5)
ReflectiveObjectScanningDevice (0 / 1)
TransparentObjectScanningDevice (0 / 1)
UnderwaterScanningDevice
WaveAbsorbentObjectScanningDevice (0 / 1)

(a)

IMATI	UU	IMPACT	Ad1	Ad2
X		X	X	X
X	X	X		X
X	X	X		X
	X	X		X
X		X	X	X
X		X	X	
X	X	X	X	X
				X
				X
		X		

(b)

Figure 11 – (a) The hierarchy resulting from the reasoning to infer the instances type; (b) how the instances are distributed in the classes by the type inference, when a 'x' is appear it means that the instance belong to the correspondent class .

We would like to outline that such an example helps answer some of the competency questions (CQs) posed by the domain experts e.g.:

- [1.b] What are the Acquisition Systems able to scan a Real Object which is transparent?
- [1.j] Is a laser scanner able to scan transparent objects?

In case of the CQ number [1.b], the answer can be found browsing the *TransparentObjectScanningDevice* class, whereas in the second case (question [1.j]) a positive answer could be obtained by checking if any instance of *TransparentObjectScanningDevice* is also a Laser Scanner. If such an instance is not available the answer is inconclusive (there is not enough knowledge to answer the query). In the ontology related to the acquisition and reconstruction reasoning has been exploited both during the Ontology development as well as to provide answer to some CQs.

5 CONCLUDING REMARKS

In this deliverable, we have presented the first version of the Ontology Shape Acquisition and Reconstruction. It is intended to produce knowledge support for the researchers who have to face the lifecycle of the shape in general. We have identified that the acquisition of a shape is fundamental in this sense. It has to be planned and the data coming from the acquisition systems have to be processed in the reconstruction phase. Moreover, further processing has to be done to fulfil the users' needs, and documentation has to be produced in order to enrich the semantic impact of the shape.

The first version of the ontology supports these requirements. In the second version we will formalize and integrate the knowledge related to shapes, tools and tasks.

Here, we briefly outline the major changes and improvements with respect to the previous version of the ontology presented in Deliverable D1.2.2:

1. **Refinement of the competency questions** some of the initial CQs were too vague and too ambitious; some of them were not related to what an ontology can do. The result of this activity has produced a list of refined competency questions (see Section 3.3).
2. **Restructuring of the *ShapeData* Taxonomy**: In particular, the creation of the

isDerivedFrom relation for *ShapeData* class in order to model the history of a shape (see Section 3.4.2), the *validation of the metadata* with respect to the Shape Repository metadata and the validation of the class definition with respect to the AIM@SHAPE Glossary.

3. **RasterData Hierarchy** Formalization of the knowledge related to RasterData and their hierarchy (see Section 3.4.2).
4. **Restructuring of the *AcquisitionSession* and the related classes** : In particular the creation of owl axioms for the *AcquisitionDevice* and *AcquisitionDeviceType* classes (see Section 3.4.1).
5. **Tasks, tools and shapes** Creation of the initial diagram for modelling the knowledge related to tasks, tools and shapes (looking also at the Tool Repository metadata) (see Section 3.4.3).
6. **Initial reasoning on the ontology**. At this stage of the ontology development, we are able to answer to provide answer to some CQs (see Section 4).

Looking at some requests from the involved partners (e.g. questions regarding MRI volumes, such as “*which anatomical region of interest (e.g., hip, knee, liver etc.) is this?*” or “*find more examples of a selected body part*”, or “*Is this a healthy organ?*”) it was quite clear that a formalization of the knowledge related to the shape analysis could be useful in order to “close” the digital shape lifecycle.

We need to understand how to integrate this knowledge in this cluster or maybe if another cluster (or sub-cluster) could be necessary in the future. We are planning to discuss this within the Network in order to understand possible interests and to plan future activities.

6 REFERENCES AND PUBLICATIONS RELATED TO THE CLUSTER

(1) R. Albertoni, L. Papaleo, M. Pitikakis, F. Robbiano, M. Spagnuolo, G. Vasilakis, *Ontology-Based Searching Framework for Digital Shapes*, OTM Workshops on Web Semantics (SWWS'05) Agia Napa, Cyprus, 31 Oct-4 Nov, pp. 896-905, 2005.

(2) L. Moccozet, N. Magnenat-Thalmann, L. Papaleo, *Representing 3D Knowledge with Ontology,* International Workshop on Multi-modal wearable interface, Geneva; July 25-26, 2005.

(3) L. Papaleo, R. Albertoni, S. Marini, F. Robbiano, *An ontology-based Approach to Acquisition and Reconstruction*, Collaboration Workshop for the Future Semantic Web at ESWC 2005 Heraklion, Greece, 29th-30th May, 2005.

(4) R. Borgo, P. Cignoni, M. DellePiane, L. Papaleo, M. Spagnuolo, *Extracting Meta-Information From 3-Dimensional Shapes With Protégé*, Poster to the 8th Intl. Protégé Conference - July 18-21, 2005 - Madrid, Spain, 2005.

(5) Papaleo L., Albertoni R., Marini S., Robbiano F., *An Ontology-based Approach to Shape Acquisition and Reconstruction*, Workshop towards Semantic Virtual Environments' (SVE 2005), Villars (Switzerland) March 16th-18th, 2005.