Università degli Studi di Genova

Scuola di Dottorato: Science and Technology for the Information Society
Ph.D. Course in
Electronic and Computer Engineering, and Telecommunications - Cycle XXI

Descriptions of 3D objects based on concepts, content, and context

Francesco Robbiano

Coordinator:
Prof. Bruno Bianco
Relators:
Dr. Michela Spagnuolo
Dr. Bianca Falcidieno

SSD INF/01

Riassunto della tesi

Questo lavoro è un percorso all'interno del mio approccio personale ed originale al processo di descrizione ed annotazione di oggetti tridimensionali, considerando il ruolo e il significato dei concetti, del contenuto e del contesto. Ho provato a organizzare una sorta di viaggio all'interno dell'argomento della descrizione di oggetti durante il quale ho appuntato sul mio taccuino di viaggio osservazioni, necessità, vincoli e principi. Questi appunti riflettono un approccio personale, basato su riflessioni circa l'intero processo. Un contributo importante è anche dato in termini di implementazione e sviluppo di strumenti software che rispecchiano e concretizzano il mio approccio personale all'annotazione e alla descrizione di oggetti 3D. Nella fattispecie, consistono in un insieme di ontologie che si giovano di strumenti tipici delle Tecnologie della Conoscenza per la concettualizzazione di oggetti 3D da diversi punti di vista, e lo strumento ShapeAnnotator, un sistema flessibile e modulare per l'annotazione di oggetti 3D basata sulla loro scomposizione in parti.

La scrittura della tesi segue questa traccia. Nel Capitolo 2 sono analizzate le problematiche relative alla descrizione di oggetti 3D: dove e come sono usate le descrizioni, su quali aspetti cognitivi umani si basano, quale è il loro ruolo nel contesto di una prospettiva semiotica che comprende anche le definizioni e le rappresentazioni. Nel Capitolo 3 l'attenzione si focalizza sugli aspetti cognitivi attraverso i quali gli oggetti 3D vengono percepiti in forma concettuale. Quindi viene approfondito il ruolo dei *concetti*, insieme ai possibili paradigmi per ottenere concettualizzazioni (es. ontologie, folksonomie). Inoltre, sarà presentato il mio contributo alla definizione di ontologie per forme 3D. Nel Capitolo 4 ci si concentra ancora maggiormente sulla portata informativa degli oggetti 3D ed è studiato appunto il loro *contenuto*: qual è l'informazione che può essere estratta dai modelli digitali di un oggetto 3D, qual è l'impatto di tale informazione sulla caratterizzazione semantica dei modelli stessi, come può essere utilizzata,

quali sono i difetti delle attuali metodologie di descrizione. Nel Capitolo 5 l'attenzione si focalizza sul ruolo dell'utente: non solo si considera il comportamento umano in maniera statica, ma si considerano le sue attitudini e i suoi requisiti all'interno di ogni singola query (oppure di ogni altra fase di fruizione), ed è ciò che chiameremo il *contesto* con il quale l'utente si relaziona al mondo esterno: in diversi contesti potrebbe essere interessato a diversi aspetti di uno stesso oggetto. Il Capitolo 6 è dedicato alla descrizione di uno dei principali prodotti implementativi della tesi: lo *ShapeAnnotator*, uno strumento per aiutare l'utente a segmentare ed annotare oggetti secondo la loro scomposizione in parti. Infine, nel Capitolo 7 verranno discusse alcune conclusioni relative al lavoro svolto, anche in prospettiva di sviluppi futuri.

Abstract

This work reflects a personal and original approach to the whole process of 3D object description and annotation, considering the role and the meaning of concepts, content and context. I tried to set up a sort of journey inside the issue of object description during which I have been collecting observations, needs, constraints and principles. In most of the cases they reflect my personal ideas, which are based on solid reflections about the whole process. An important contribution has been also given in terms of implementation and development of software tools which reflect and concretize my personal approach to the subject of 3D object annotation and description. Namely, they consist of a set of ontologies, which make use of Knowledge Technologies for a conceptualization of 3D objects from a number of diverse points of view, and the ShapeAnnotator tool, a flexible and modular system for part-based annotation of 3D objects. The outline of the thesis is as follows. In Chapter 2, the issues related to the *description* of 3D objects are analyzed: where and how descriptions are used, what cognitive aspects of the human attitude they take into account, what is their semiotic role with respect to definitions and representations. In Chapter 3 I focus on the cognitive attitude of a human user towards a 3D object, which is often characterized by a conceptual approach. Therefore the role of *concepts* will be deepened as well as the possible paradigms for conceptualization (e.g. ontologies, folksonomies). Moreover, my contribution to the definition of ontologies for 3D shapes will be described. In Chapter 4 the focus passes to 3D models and the issue of *content* is studied: what is the information that can be extracted from the digital models of 3D object, what is the impact of this information on the semantic characterization of the models, how it can be handled, what are the shortcomings of the current methodologies for description. In Chapter 5 the focus shifts towards the user: not only we consider the human behavior in a static way, but we consider his/her attitude and his/her specific needs in a given query (or any other consumption modality), and we call it the *context* in which the user relates to the external world. In different contexts he/she can be interested in different peculiarities of the object itself. Chapter 6 is devoted to the description of one of the main implementative outcomes of the thesis: the *ShapeAnnotator*, a framework for helping the user to segment and annotate objects according to their decompositions in parts, is presented and discussed. Finally, in Chapter 7, I draw some conclusions about the work done and put it in perspective for future developments.

Acknowledgements

I start this section in English because I want to thank my international contacts. Among them, all the partners of the AIM@SHAPE Project, which has been a sort of big family, and the hosts of the interesting schools held in Tallinn, Chalkidiki, Genova and Utrecht respectively. Particular insight on the topics of my work has been achieved through the interaction with Simone Santini, Ivan Herman and the whole group headed by Lynda Hardman at CWI in Amsterdam, where I recently spent three weeks. Now I move within the national boundaries, where I can switch to italian. Ringrazio il Professor Bianco per la grande disponibilità dimostrata in tutti questi anni, sempre condita da una buona dose di entusiasmo e curiosità. Ringrazio i co-autori dei miei lavori scientifici. E poi, finalmente, prendo spunto da un argomento di cui si è discusso in una recente riunione dello Shape Modelling Group dell'istituto IMATI-CNR, per considerare me e i miei colleghi non solo individualmente ma come una *squadra*. È il complesso organismo della squadra, oltre che i singoli individui, che voglio ringraziare. È lo spirito di squadra a galvanizzarmi e a motivarmi. Quindi ringrazio in primis Bianca e Michela, che oltre ad essere le Relatrici di questo lavoro sono anche l'anima di questa nostra squadra, e hanno saputo entrambe aiutarmi e capirmi anche nei momenti difficili. Poi ringrazio sentitamente tutti i colleghi: ricercatori, tecnici e amministrativi, con una menzione speciale per il contributo "dietro alle quinte" di Marinella e Sandra. Tra i colleghi più vicini a me, voglio dedicare un grazie "un po' più speciale" a Daniela e Marco, il cui impatto su questa mia tesi è stato, a dispetto del titolo, davvero *indescrivibile*, oltre alle mie lontane ma sempre vicinissime compagne di ufficio Chiara e Icchia. E poi, c'è la vita al di fuori del lavoro. Altre storie, altre squadre, composte da persone meravigliose, a cui in questa sede mi limito a dedicare un ultimo e collettivo *grazie*.

Contents

List of Figures

Chapter 1

Introduction

Narrami, o Musa, dell'eroe multiforme, che tanto
vagò, dopo che distrusse la rocca sacra di Troia;
di molti uomini vide le città e conobbe i pensieri,
molti dolori patì sul mare nell'animo suo,
per acquistare a sé la vita e il ritorno ai compagni.
(Omero, Odissea)

Nowadays, 3D content is widely recognized as the upcoming wave of digital media and it is pushing a major technological revolution in the way we see and navigate the Internet. The success of 3D communities and mapping applications (e.g., Second Life, GoogleEarth) and the decreasing costs of producing 3D environments are leading analysts to predict that a dramatic shift is taking place in the way people see and navigate the Internet. The key elements in this landscape are digital 3D resources and users dealing with them.

Having digital resources recorded and stored in order to be accessed by users is just one of the issues to be dealt with. One of the most important problems to be overcome is coupling the resources with an interpretation suitable to human cognition and expressable by some formal language. In this setting, according to the semiotic approach, we can assume the digital encoding of a resource as a syntactic issue, the meaning carried by this encoding as a semantic issue, and its ultimate interpretation by the user as a pragmatic issue [108].

Addressing the semantics of digital 3D objects has a key value in a number of applications, such as the search and retrieval of objects according to their high-level characterisation (useful in diverse areas, e.g. Cultural Heritage, Gaming and Simulation, Virtual Tourism, Medicine), and the exploitation of them in the Semantic Web

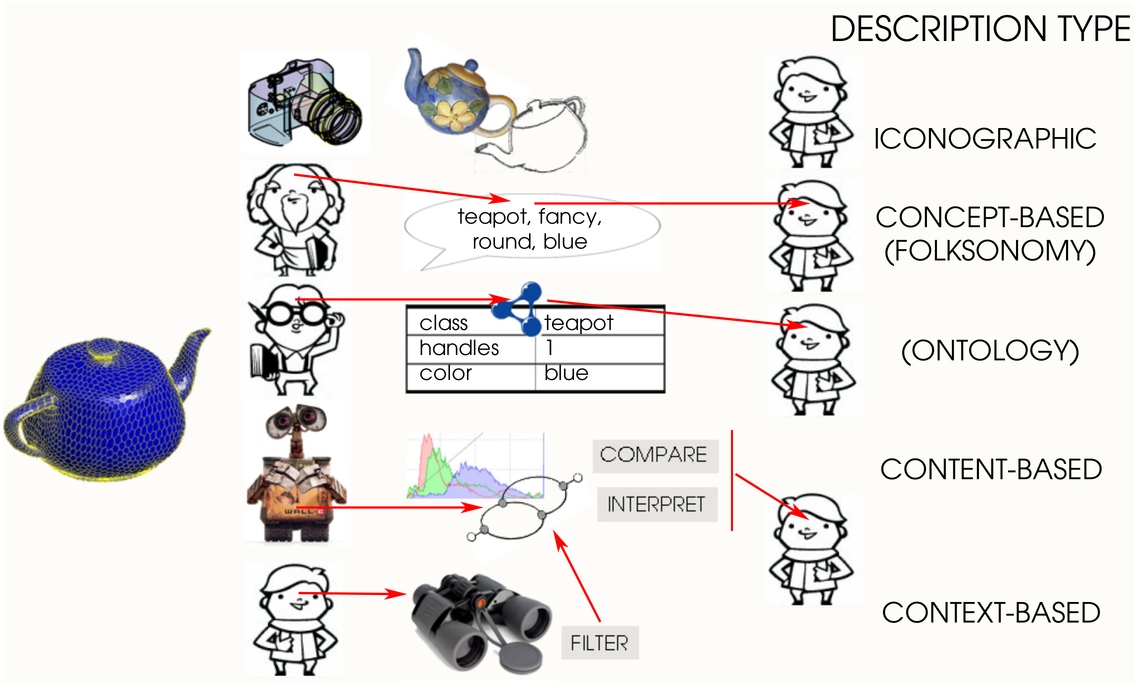

Figure 1.1: Different approaches to description. Concept-based description is directly driven by the user's perception, while content-based is driven by automatic analysis for comparison and interpretation. In context-based description the user has an important role in setting up the relevant features to be used.

framework, by exposing descriptive metadata comprehensible both from humans and from machines. If we look at the big picture of this "quest for semantics", it is interesting to consider the inter-relationships among the objects living in a real or virtual world (as the source of the signs), the digital resource (as their encoding), and its meaning for the user.

Two gaps are known from the literature and need to be filled, or at least reduced: the *sensory gap* and the *semantic gap*. Borrowing the definitions from [99], the *sensory gap* is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene, while the *semantic gap* is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.

Filling these gaps is one of the major targets addressed by multimedia community. Some of the approaches are common to all the multimedia typologies (i.e. images, audio, video, 3D), nevertheless each domain has its own specificities.

In the case of 3D models, it is possible to observe that the sensory gap is often not so large. For most application domains the accurate representation of the surface of an object holds most of the spatial information related to it. It is not affected by occlusion, distortion, or skewing effects, as it may happen in the case of bidimensional images of the same object.

The semantic gap is harder to face. What is missing from the bare representation of the object surface is non-spatial information, i.e. the information not related to the shape of the object (e.g. ownership, material, price), and the information related to the high-level conceptualization of the object, either expressed as features (e.g. size, volume, compactness, presence of holes) or as synthetic concepts (e.g. what it is, what it is used for). To tackle this problem we may follow two complementary paths: on the one hand it is possible to integrate the missing information through the artificial injection of *concept-based* metadata (either by following a formal conceptualization encoded in an ontology or by letting a versatile tagging of the resources), on the other hand it is possible to exploit state-of-the-art descriptors and analysis tools to extract *content-based* information from spatial information.

Both the mentioned approaches concur to narrow the semantic gap: the design of ad-hoc ontologies and the development of a tool for a versatile annotation of 3D objects are part of the plan, yet another important element will be considered crucial for the description process: the role of the user (see Figure 1.1). In fact, reading carefully the above definition of the semantic gap, the interpretation of the data for a user *in a given situation* is mentioned. This means that it is not sufficient to provide a generic interpretation of the data, but the interpretation should take the situation, or *context*, into account. The awareness of the context should enable a dynamic and user-oriented description in which only the most relevant content-based descriptors are activated and combined. The aforementioned considerations constitute the backbone of my approach, which is driven by the centrality of the user in the description process. The integration of concept-based approaches and content-based approaches, and their embedding in a context dynamically defined by the user will be the final goal.

1.1 Summary of the Contributions

In this work, my main contribution is to cast a personal and original approach to the whole process of 3D object description, considering the role and the meaning of concepts, content and context. At the beginning my original interest was about object annotation, but I did not mean just to provide another technical solution to the problem by diving directly in the implementation issues. My approach was much broader, as I tried to set up a sort of journey inside the issue of object description during which I have been collecting observations, needs, constraints and principles. In most of the cases they reflect my personal ideas, which are based on solid reflections about the whole process and will be explained throughout the thesis. An important contribution has been also given in terms of implementation and development of software tools which reflect and concretize my personal approach to the subject of 3D object annotation and description. Namely, they consist of:

- a set of ontologies built for a conceptualization of 3D objects from a number of diverse points of view, presented in Chapter 3;
- the ShapeAnnotator tool, a flexible and modular system for part-based annotation of 3D objects, described in Chapter 6.

This work found an ideal cradle in AIM@SHAPE [79], a Network of Excellence which put together thirteen European excellence partners under the leadership of CNR-IMATI-Genova. The Mission of AIM@SHAPE was to foster the development of new methodologies for modelling and processing the knowledge related to digital shapes. I had the opportunity to deepen the relationship between 3D objects and semantics, not only at a theoretical level but also at a practical level. I had the chance of facing issues related not only to Shape Modelling but also to Knowledge Technologies. Indeed, I was involved in the design and development of the so-called Digital Shape Workbench [77], which made and still makes use of the aforementioned ontologies, and of a Digital Library of references that was developed under my responsibility. This was an important experience also for enhancing my familiarity with metadata and the Semantic Web, both in general and applied to the field of 3D objects. From the project a lot of international collaborations stemmed. The aspects related to the Semantic Web have been further deepened during a Short Mobility period spent in November 2009 in Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands.

I had the chance of producing twelve international scientific publications related to the topics of the present thesis, four of which where published in international journals. Details about these publications can be found in the related chapters.

1.2 Outline of the thesis

The outline of the thesis is as follows. In Chapter 2, the issues related to the *description* of 3D objects will be analyzed: where and how descriptions are used, what cognitive aspects of the human attitude they take into account, what is their semiotic role with respect to definitions and representations.

In Chapter 3 I will focus on the cognitive attitude of a human user towards a 3D object, which is often characterized by a conceptual approach (e.g. the user will more likely describe an object as a “thick book” than as a “compact parallelepiped”). Then, the role of *concepts* will be deepened as well as the possible paradigms for conceptualization (e.g. ontologies, folksonomies). Moreover, my contribution to the definition of ontologies for 3D shapes will be described.

In Chapter 4 the focus will pass to 3D models and the issue of *content* will be studied: what is the information that can be extracted from the digital models of 3D objects, what is the impact of this information on the semantic characterization of the models, how it can be handled, what are the shortcomings of the current methodologies for description.

In Chapter 5 the focus will shift towards the user: not only we consider the human behavior in a static way, but we consider his/her attitude and his/her specific needs in a given query, and we call it the *context* in which the user relates to the external world. In different contexts he/she can be interested in different peculiarities of the object itself.

Chapter 6 will be devoted to the description of one of the main implementative outcomes of the thesis: the *ShapeAnnotator*, a framework for helping the user to segment and annotate objects according to their decompositions in parts.

Finally, in Chapter 7, I will draw some conclusions about the work done and put it in perspective for future developments.

Chapter 2

Description

E come sempre sei la descrizione di un attimo per me
e come sempre sei un'emozione fortissima
e come sempre sei bellissima
(Tiromancino, La descrizione di un attimo)

The main target of this Chapter is to introduce the topics of the thesis and to clarify the general issues and requirements that will be discussed: in a sort of "organized brainstorming", a lot of issues will be triggered. I will start from general assessments on the need of sharing digital resources, pointing out the important role of description in this process. Then, starting from ordinary life examples, the challenge will be to understand what are the requirements for a description that is perceived as a "good one". These requirements will guide us in the process of breaking down the concept of description itself: a description may be seen as a loose form of *representation* in which a *shift of language* and an *elaboration layer* are allowed. Such "constituting bricks" are analyzed and specialized in the context of our interest, i.e. 3D objects. An overview on the actual usage of description will be given, in the more general framework of information handling. Finally, the duality of the *concept-driven* approach versus the *content-driven* approach in description of 3D objects will be introduced.

2.1 Where the sharing takes place

Any kind of digital resource is created and stored in order to be shared. *Where* does the sharing take place? According to its scope (i.e the potential audience of users), the resource may belong either to a narrow domain (e.g. local network, personal computer) or to a broad domain (e.g. Internet). Indeed, the resource sharing, which can be seen as a multiple access to information, occurs even in very narrow domains, e.g. resources accessible by one single person: they are meant to be preserved in time, and so they are meant to be accessed in different moments and used in different contexts. Recently, the life cycle of a digital resource is getting longer and longer, and it is more likely that the *same* resource is going to be used in *different* application domains. This aspect may look quite straightforward but it is very important: we will learn to appreciate the fact that a resource is not only characterized by its own peculiarities, but also from whom and for what purposes it is used. The *same* resource can be used in *different* moments and in *different* contexts.

2.2 What is shared: levels of granularity

Now let us focus the attention on *what* is shared. There are two main levels of granularity through which the user can deal with the resources and share them: he/she can considerthe resources one by one, or refer to them as a collection. In the case of visual resources, and in particular 3D objects, which can be often perceived as a composition of simpler parts, it is possible to point out a third level of granularity, i.e. dealing with parts of objects (see Figure 2.1).

This third level of granularity plays a key role both in the synthesis phase (modelling) and in the analytic phase, and it is important to understand the relevance of shape components in both phases. From the synthesis point of view, an object may be modelled by assembling its building blocks: to make a chair, some components are needed, e.g. the chair base, the seat, the arms and so on (see Figure 2.2). This approach is adopted, for instance, in the *modeling by composition* modality of object design [53].

From the analytic point of view, it is important to note how objects can be perceived by humans. It is widely thought that perception is connected with an instinctive segmentation task: research work tackling the segmentation problem with the objective of *understanding* a shape [7] is mostly inspired by studies on human perception, that loosely couples semantics to geometry. For example, there are theories that received a large consensus [20, 55] and that indicate how shapes are recognized and mentally coded in terms of relevant parts and their spatial configuration, or structure. This

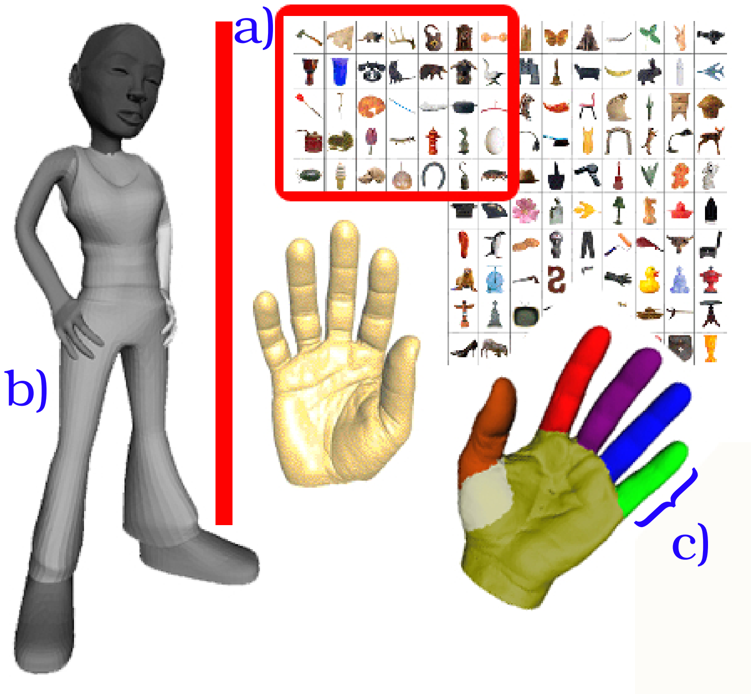

Figure 2.1: Different levels of granularity in the cognitive relationship between user and objects: a) groups of objects, b) objects, c) part of objects

Figure 2.2: Constituting components to make a chair: Back Inner Shell, Back Foam, Back Cover, Adjustable Arms with Pads, Chair Base, Casters, Seat Mechanism with Back Click System, Seat Foam, Seat Cover, Seat Plywood

aspect will be dealt with in Chapter 6, where the task of annotating object parts with concepts is connected with the task of segmenting the original object.

2.3 Ordinary life examples

To guide our insight into the role of description, it is possible to start from examples of descriptions taken from ordinary life, and detect patterns and keywords within them that need deeper analysis. In our ordinary life (both real and digital) we often deal and interact with descriptions of resources. When we read a book we are entertained by the author's description of a landscape or of a feeling; when we look for a hotel room we browse its online description through text, pictures, comments and maps; when we want to have an overview of a scientific paper we check its abstract; when we want to get some basic information of a person we may be interested to look at his/her identity card or at the information in his/her homepage.

Often the description of a resource is reduced to the description of (some of) its structurally relevant subparts; when a girl is described as "blonde and blue-eyed" a two-step process is involved. In the first step a structural conceptual decomposition is (implicitly, in this case) assumed: a human being is characterized, among the other things, by its hair and its eyes. In the second step a separate description of each of these components is used in order to describe, or to enrich the description, of the girl.

When in a conversation some opinions about an event or a person are shared, very often several descriptions are used to communicate: what happened, what was the reactions of people, how they looked like. These descriptions in many cases are subjective and can also rely on relationships of comparison with other things or events, which may be homogeneous with respect to the original resource (similarity) or heterogeneous (metaphor). In the first case it can be possible to state "John's face is similar to his father's", in the second case it is possible to say that "her eyes look like a mountain's lake".

2.4 Requirements for a good description

From the previous paragraph it is possible to identify a number of requirements that have been implicitly triggered. The first quality that a description should have to be a good one is being *icastic* In the case of a visual description (e.g. based on images), it should be as vivid as possible, in order to convey the most prominent features of the resource directly to the users' perception, "advertising" at its best the described resource. In the case of non-visual description (e.g. text, metadata) the most prominent quality of a description is to be properly *informative*. This concept breaks down into two requirements that need to be calibrated: descriptions should be *essential* (synthetic, only important features have to be kept, the focus is on forgetting as much as possible) yet *expressive* (complete, all the important features have to be kept, the focus is on remembering as much as possible). If the description is too essential the risk is being too generic; for instance, "this is a resource" is a true and very synthetic description, but does not yield any information about the actual resource. In Information Theory [36] it could be stated that the self-information related to this description is zero. When, on the other hand, the description is too expressive the risk is to lose track of the important information by diluting it in unnecessary details that can hide the relevant features. This is one of the turning points of the whole process: which portion of the available information has to be kept? And how can it be separated from the unnecessary information?

Finally, I cite two views about descriptions, and their role, that were presented in the past. The first one, due to D'Arcy Thompson in his seminal work on "*On growth and form*" [104] where he emphasized evolution as the fundamental determinant of the form and structure of living organisms and gave mathematicians the task to extract the very essence of form, or shape:

> "we must learn from the mathematician to eliminate and discard; to keep in mind the type and leave the single case, with all its accidents, alone; and to find in this sacrifice of what matters little and conservation of what matters much one of the peculiar excellencies of the method of mathematics".

The second interesting aspect which can be referred to the role of descriptions has been noted by Nackman et al. in [6], in which a clear and more pragmatic distinction is made between representation and description:

> "an object representation contains enough information from which to reconstruct (an approximation to) the object, while a description only contains enough information to identify an object as a member of some class of objects".

Both of these approach looks fine, but in my opinion do not hit the key of the general problem. In D'Arcy Thompson's approach it seems that *discarding* is a necessary step of description but still, using his own words, a "sacrifice". This sacrifice helps to eliminate all the details that make the *single cases* different among one another, in the effort of capturing the common characteristics, i.e. the *type*. The hint is very precious, as it focuses on the importance of *forgetting* some characteristic of the object to be described, yet it suggests an interpretative structure (i.e. single cases vs. type) which is not always there: the same principles can be used even when dealing with objects that are not of the same type. The semiotic approach should be more flexible, as it is not possible to state that meaning pre-exists the resources (as it will be discussed in detail in Chapter 5), and no interpretative structure can be cast a priori.

In Nackman's words the focus is on the definitions of *representation* and *description*, but again it presupposes that the membership to some *class* has to be identified or extracted, and this is an assumption that we cannot afford to make. What I think it is important is to provide definitions that are targeted to the *use* that it can be made of descriptions rather than trying to capture their absolute and irrefutable meaning.

2.5 Going to the roots of description

Describing can be etymologically traced back to Latin “de+scribere”, i.e. “writing about”. Describing is representing something just with the help of words, or in a wider sense, with the help of a different language. It is already possible to make important considerations. In any description process the aim is representation, but a shift of language is intended, and therefore the aim is not to present the target of the description exactly as it was originally: a layer of elaboration is allowed. In few words, my claim is that a description is a *looser form of representation* in which a *shift of language* and an *elaboration layer* are allowed. Different layers of elaboration give the freedom of dealing with the described object in different contexts.

A lot of keywords popped out from this early statements: representation, shift of language, elaboration layer, context. We will discuss these keywords in general scenarios and show how these can provide suggestions on the development of effective computational solutions in the domain of 3D objects.

2.5.1 Representation

The etymology of the word “representation” suggests that something is re-presented, i.e. presented again. Presented to whom? In the first place, it is important to understand which is the targeted audience of the presentation, and eventually to decide how to reproduce the original reaction in the audience itself. Nowadays in the majority of the scenarios the role of a human user is more and more important and active, and the requirements of a representation have to take into account his/her attitudes towards the 3D object. The intended audience behaves actively, and far from being ruled only by a sensorial activity, is characterized also by a mental attitude. When a human is facing a 3D object, he/she perceives it (at least) in two ways: he/she stores not only its visual appearance, but also a conceptual picture of it. Such conceptual characterization does not only involve sensorial information (e.g. soft, perfumed, shiny) but also the information achieved after an immediate elaboration (e.g. heavy, small, nice).

The first conclusion is therefore that, when human users are involved, a faithful representation of a 3D object should include also a conceptual picture of it. Its representation should be enriched with high-level information, such as the linguistic category of the considered object, along with some of its main perceptual features (e.g., “ball, large”, “table, elongated, low, with four legs”).

However, in other applications fields (e.g. medicine, engineering) the most important issue is not directly related to addressing the cognitive attitude of a human user, rather, it is related to the ability of calculating quantitative information about the involved resource. In these cases a good representation is the one that eases this kind of computations.

In the domain of 3D objects, the polygonal mesh (often a triangular mesh) is a representation format that satisfies both the “visual” requirement of simplifying the rendering task and the requirement of allowing for easy computation on top of it. It represents the surface enclosing the object, while the volume is often implicit. In some cases (e.g. Magnetic Resonance Images, Finite Element Analysis), conversely, a representation involving explicitly the enclosed volume is needed.

There are a lot of other representation models: in the late eighties, the use of computers has revolutionized the approach to 3D object modelling, opening new frontiers in new research and application fields: Computer Aided Design, Computer Graphics and Computer Vision, whose main goal is to discover basic models for representing and generating shapes. At the beginning, this effort gave rise to research in Geometric Modelling, which sought to define the abstract properties which completely describe the geometry of an object (geometric model) and the tools to handle this embedding into a symbolic structure.

Terminology and definitions for the foundations of geometric modelling were first introduced in Requicha’s seminal 1980 article [90], whose basic notions have shaped the whole field to this day. Requicha’s paradigm uses four levels of abstraction, called universes (see Figure 2.3). The first level is used to understand the problem from the point of view of the universe to be modelled, made either of real or virtual objects (the *physical* universe); the second level is used to study the problem from the mathematical point of view (the *mathematical* model); the third allows us to understand the various issues of discretising the elements from the mathematical universe

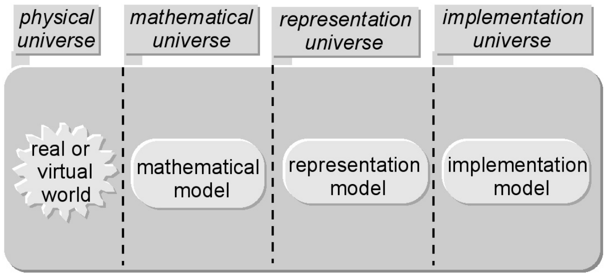

Figure 2.3: The modelling paradigm proposed by Requicha.

(the *representation* or *computational* model); and the fourth is used to map the discretised elements from the representation universe into the structures of a computer language (*implementation* model). Following this paradigm, considerable research activity has been developed in the two most well-known representation schemes for solids: CSG (Constructive Solid Geometry) and BRep (Boundary Representation), which have deeply influenced current commercial geometric modelling systems. Nowadays, there are many more representations schemes that are relevant in specific domains (e.g. implicit surfaces, physically-based models, point sets) and some, such as polygon soups, that are common in generic user domains and are not based on solid mathematical foundations. The discussion presented in this thesis applies to 3D objects in general, whatever representation scheme is used for them.

2.5.2 The shift of language

The description may be seen as an alternative way of expressing the original resource. A shift of language may be useful, mainly for two reasons: the original language can not be used, i.e. there are limitations in information *availability*; the original language is not easy to grasp from the cognitive point of view, i.e. there are limitations in information *consumption*. If a user is shopping online and looking for a dress, he/she would be very glad to have the original resource (the dress) in order to try it on, but since it is not possible he/she has to rely on a as-detailed-as-possible description. In this case the problem is availability, and the description process is intended to fill this gap. On the other hand, suppose that a person is undergoing a radiological investigation such as a study of the intracranial arteries in which the actual data is represented by a 3D representation of the vessels. In this case the representation itself, without a proper description (i.e. the neuroradiological written report) cannot be interpreted by the reader (patient, general practitioner, other specialists). The lack of information consumption is overcome by the neuroradiologist who provides a shift of language by means of a written report. In some sense, it can be said that the information has to be metabolized before being consumed: in this case the problem is information consumption.

Disregarding the reason *why* the shift of language takes place, it is interesting to note that the use of another language opens the description (and indirectly also the original resource) to all the analytic tools typical of the new language. For instance, if something is described through *images*, the use of all the tools making use of *image processing* techniques will be triggered.

2.5.3 The elaboration layer

The elaboration layer is what makes a description different from a mere reduplication of the original information. The aims of the elaboration can be diverse, ranging from a radical simplification for immediate informative use to a complete and realistic portrait for aesthetic or communicative reasons. Data expressed in an Identity Card capture just some key features of the person to describe, while Hemingway's description of oysters gives account of a personal attitude towards eating them. In this case the elaboration is personal:

> As I ate the oysters with their strong taste of the sea and their faint metallic taste that the cold white wine washed away, leaving only the sea taste and the succulent texture, and as I drank their cold liquid from each shell and washed it down with the crisp taste of the wine, I lost the empty feeling and began to be happy and to make plans.

Also a caricature is a form of description: the details to be emphasized are even iper-realistic, to be sure that the attention is conveyed on them.

What is common to all these steps is that they are all *about* the original resource, and targeted to capture (and to communicate) some of its peculiarities and to discard some other peculiarities.

In the previous examples the elaboration step is performed by a functionary, by a writer or by a painter. In the case of digital 3D objects the elaboration can be performed not only manually, but also via a computation. Any manual assessment is done arbitrarily, while any computation can be performed through an analysis of the content of the resource. In the following Chapters we will consider and discuss about both paths.

2.6 Three phases for sharing 3D objects

In all the situations in which the sharing of digital resources occurs, there are three main phases which need to be taken into account: the *storage* (push phase), the *retrieval* (pull phase) and the actual *usage* (consuming phase). Descriptions of resources play a key role in sharing, as they summarize the resources pointing to the inmost relevant and prominent features. Descriptions can be regarded to as the most generic form of meta-data, as they are data about the data they describe. The thesis will investigate the role of description in these phases.

Until this point, I mentioned generically "digital resources", but our attention is mainly directed to 3D models, and so I will focus on the specificities of this kind of resources on each of the three phases that occur in any kind of resource sharing, as stated above.

In the *push phase* (i.e. storage), it is important to capture all the information that is not extractable from the resource itself. It might be important to encode the knowledge of the creator: if the digital model is a copy of a 3D object, provide information about the original real object and about the acquisition phase; if it is artificially produced, provide information about the intent of the designer or about the related sources used in the process of building the model.

In the *pull phase* (i.e. retrieval), the object(s) have to be retrieved, and therefore all the information related to them is important: not only the information that has been attached in the push phase, but also the information that can be extracted by their content, i.e. that is intrinsic to the representation of the models. These content-based descriptions are important, still, we can observe that some characteristics of the objects are relevant in some contexts and negligible in others. In the pull phase the user is embedded into a context, i.e. he/she has in mind requirements about the object to be pulled. Moreover, in this phase it is important to understand which are the specific requirements of the user interested in the retrieval, in order to privilege those descriptions which are more aligned to the actual needs. For instance, if the query ultimate goal is to find humans among other objects, the user will be very much interested in the structure (e.g. legs, torso, arms, head), which well discriminates humans with respect to other kinds of objects, but he/she will be interested in discarding information about the pose, for he/she will not be interested to discriminate standing humans from sitting humans. Therefore it is very important that descriptions of objects can take into account the context (i.e. the user's need in a specific pull phase).

Especially in the field of 3D objects, also in the usage phase (i.e. consumption) the use of descriptions may be very important. In fact, one of the distinctive features is that the digital models of 3D objects are often encoded into very large files. Often the download is slow and band-consuming, therefore it is important to download only the models which we are really interested in. A throughout description of the models can help the user to make up his/her mind before the actual download of the model. Another prominent feature of most 3D objects is their complexity: a faithful representation of the shape of an object encodes also the shape of its constituting parts (e.g. the model of a teapot will bear information about the shape of its spout, its handle and its tip). Thus, a conceptually driven description of the object would be very welcome, but it should take into account also the information about its constituting parts.

Just to summarize in few words, often the life cycle of a 3D digital model is constituted by one push phase and several pull phases and consumption phases; in each of them, descriptions play a fundamental role, and should take into account the *concepts* connected with the resource (both as a whole and considering its constituting parts), the *content* intrinsically carried by it, and the *context* according to which the user is interested to interact. More insight about these modalities will be given in the next Chapters.

A separated consideration should be devoted to browsing. In this case, it could be recalled that the important is not to search, but to find information. In fact, it happens that the user has no precise idea about what to search, but nevertheless he/she just browses a repository (like a catalogue) and may end up retrieving something he/she is interested in. Browsing can be considered as a phase in which the user consumes the information related to the whole set of displayed resources, but also as a search modality, much like a catalogue: it can be seen as a source of information itself or just as a means to detect the linked resources. Browsing has also an added value in information serendipity, as the path from what a user searches to what he/she finds is not always so linear: searching for one resource may be helpful in *finding another one.* In this phase, the icastic requirement of the presented descriptions is the most important, because it raises the cognitive impact of the description of the resource, perceived as a mock-up of the resource itself. To conclude, browsing should be considered both in the pulling phase and in the consuming phase.

2.7 The target of description

Let us see what are the main goals of information description in the more general framework of information handling. The classification of information tasks can be traced back to [49] and synthesized in the list below (see Figure 2.4):

1. Fact Finding: users ask goal oriented and focused questions; they look for specific factual pieces of information.

2. Information Gathering: users carry out several search tasks to fulfill a higher level goal, such as writing a report, preparing an event, or collecting information to make a decision.

3. Keeping Up-to-date: this search task is generally not goal-driven, other than to "keep up-to-date", "just browsing", "see what is new or interesting", recreational searching or even "passing time".

4. Communication: an information exchange task, either face-to-face or through technology, such as email.

5. Transaction: an information exchange task, e.g., online auction, banking, downloading multimedia documents. Transactions are often associated with a user name and password combination.

6. Maintenance: a task which involves organizing information, e.g., updating bookmarks or organizing email in the appropriate folder.

Descriptions play an important role in two ways. In the first place, they can be used as mock-ups when the whole resource is either not available or not immediately

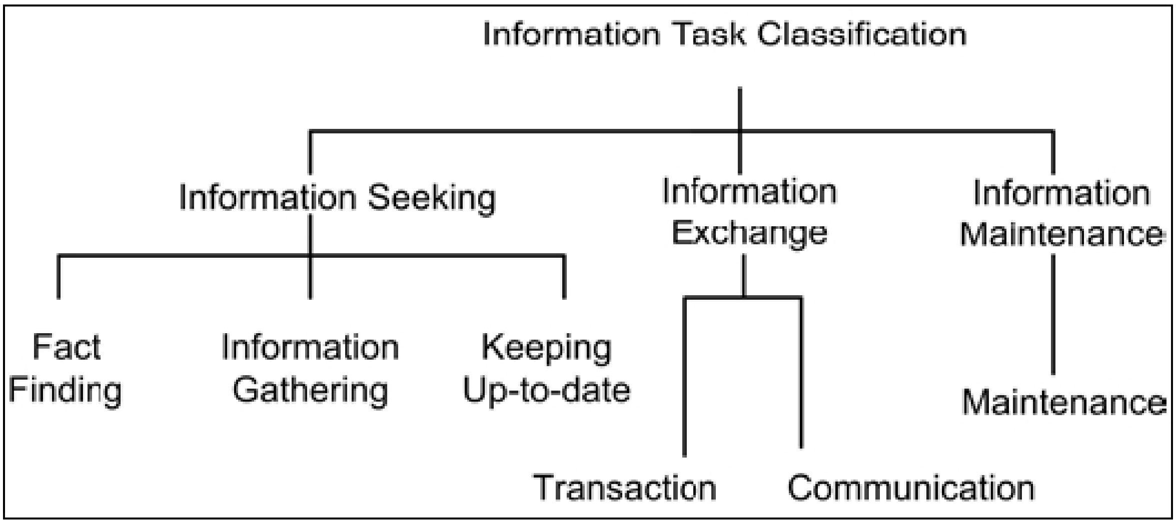

Figure 2.4: Classification in Information Tasks, taken from [4]

understandable at a first glance, much like thumbnails are often used as quick references temporarily substituting an image. This is important in the information seeking activities only in the cases in which there is no precise goal in the keep-up-to-date modality (i.e. to enhance browsing in such a way to maximize serendipitous effects), possibly in the early phases of information exchange (i.e. to provide an early preview of the information to be exchanged), and in information maintenance (i.e. to help the organization of the information).

In the second place, descriptions are very important when they structure the information carried by the resources in such a way to be matched against the user needs. The match can be performed either on the semantic side (e.g. keyword matching, ontology searching) or on the structured content itself (e.g. shape recognition, shape matching). In the former case the conceptual characterizations will be privileged, while in the latter case the content-based descriptors will be taken into account.

Matching is the main task in any resource retrieval framework. A query is submitted by the user of a retrieval system, and matched against the eligible resources in order to find among them the closest to the query, i.e. the most interesting one(s). If a precise match is required, it is possible to say that the task is a *recognition* problem. The estimation of the precise matching or of the proximity between two resources corresponds to the cognitive attitude of similarity assessment. As usual, there are two interweaving perspectives. In the case of conceptual characterizations, the similarity assessment is purely performed by comparing the metadata describing the query and the involved resources, as it happens in keyword-based search or in finer forms of reasoning. In the case of content-based descriptors, the similarity assessment is performed on the space of descriptors.

Generally speaking, shape matching methods rely on the computation of a shape description, also called signature, that effectively captures some essential features of the object. The shape descriptions are then compared using an appropriate computational technique able to translate the similarity between objects into some distance between descriptors. Often the concept-based and the content-based modalities can be fruitfully integrated.

One of the possible way to integrate these aspects is to pursue an automatic *classification* of a set of resources. The goal is to subdivide a set of resources in (homogeneous) categories which are known a priori. The categories are concept-based (e.g. *humans*, *animals*, *vehicles*), while the automatic estimation of the membership can be left to an automatic analysis of the resources. Beside the difficulty of the task, it is important to understand how the subdivision in categories casts a "homogeneity measure" on the set. Resources that can be clustered together from one point of view (pencils and erasers, both "desktop objects") can be considered very separated if we are interested in sharpness (pencils and swords would be more similar and possibly would fall under the category "sharp objects"). Therefore, unless there is a clear a priori evidence about the context, no automatic methodology can be applied effectively.

2.8 Concept or content?

Wrapping up, it is important to point out that the ultimate goal of a description is to have an impact on its receiver (the human user or the computer). The description can be achieved directly by addressing human perception through something that looks like the described object or resembles it vividly (hence the icastic requirement), or it can be achieved indirectly by addressing the conceptual area, i.e. with the help of a language, or it can be conceptualized in a formal fashion in order to be assimilated by a machine.

In the case of 3D objects, for describing objects and their parts we could assume that naming them could suffice, that is, shifting to the language of words and using them as tags to describe the objects. Following this approach, the idea is to use concepts, expressed by words, for describing objects by attaching annotations to them. The formalization of the vocabulary of eligible concepts may be bond to a specific conceptualization, such as an ontology, or to versatile user-driven folksonomies. This approach, that will be considered in detail in Chapter 3, is very straightforward and effective, yet it has several drawbacks and pitfalls, mainly due to the intrinsic limitations of converting concepts into words, and back. According to a real-world example, suppose that the task is to communicate the description of an object between two people (from a human source to a human destination). The describer sees the object, acquires a cognitive-based information about it and tries to synthesize the information into words. He/she describes the object, for instance, as a tall, blonde and blue-eyed man. This description is effective because the source and the destination of the communication are both humans, and thus are supposed to perceive the objects in a similar way. This kind of information does not require further elaboration: it is encoded in such a way that is immediately understandable by the user. For the user the words "man, tall, blonde, blue-eyed" make sense, and are often meaningful: the communication is effective.

The dual approach is building descriptions on top of quantitative data (descriptors) extracted from the available information: the descriptions are objective because they are based on precise measurements performed directly on the available content. They are not dependent on any language or formalism, but are bond to the tools used to compute the descriptors, each of which highlights specific characteristics of the objects, and is invariant with respect to other characteristics. In this case, which will be further analyzed in Chapter 4, the information is not directly accessible by the user. In order to access the meaning of the extracted measurements an interpretation layer is needed. In some cases this step is straightforward while in some others it is complex, arbitrary or simply not so effective to actually describe the most prominent features of an object. The quantitative nature of the measurements is sometimes an advantage rather than a drawback: their main goal may not be the direct insight of a human user, but to ease the automatic calculation of similarity with existing objects. In this case the insight is indirect, as an object is characterized by its proximity to a (possibly known) group of objects. Clustering and classification tasks may be performed on these bases.

From the above statements it is easy to conclude that neither the conceptual nor the content-based approaches are “the solution”: the strengths of both of them should be exploited to boost the effectiveness of objects description. In the next Chapters they will be studied separately, to highlight their specificities, but also conjunctly, as possible approaches to be integrated under the direct control of the user. The observations, the requirements and the needs that were mentioned throughout this Chapter will be taken into account while facing the actual issues of how to describe 3D objects.

Chapter 3

Concept

L'aquila descriveva delle volute nel cielo.
Nessuno ascoltava le sue descrizioni.
Del resto nessuno le aveva volute.
E a dirla tutta non e' che lei fosse un'aquila.

This Chapter is about the role of concepts in 3D object description. At the first place, the role of annotation as a particular kind of description is investigated. The specificities of the case of 3D objects are considered, trying to investigate on the conceptual annotation frameworks which are most effective in describing 3D objects. Advantages and drawbacks are analyzed.

My contribution to this chapter is twofold. On the one hand, I have analyzed the approach of the past and current multimedia description modalities with respect to the issue of semantics, highlighting the need of both structured descriptions (to fulfill the Semantic Web requirements) and unstructured description (to reflect better the cognitive attitude of regular users): most of the difficulties are not on the technological side, but on the socio-cultural side. Therefore, it is important to understand how, why and in which domains the two modalities can interoperate. On the other hand, from the technological point of view, I contributed to the definition of the ontologies developed in the context of the AIM@SHAPE Network of Excellence: with the goal of encoding semantic characterizations in the formalization of 3D shapes, some dedicated ontologies have been developed and used in the general framework of the Digital Shape Workbench [77].

My papers referring to this chapter ([1, 107, 84, 3, 2, 106]):

- Albertoni R., Papaleo R., Pitikakis M., Robbiano F., Spagnuolo M., Vasilakis G., "*Ontology-Based Searching Framework for Digital Shapes*", Lecture Notes in Computer Science Vol.3762, Springer, pp. 896-905, 2005.

- Vasilakis G., Garcia-Rojas A., Papaleo L., Catalano C.E., Robbiano F., Spagnuolo M., Vavalis M., Pitikakis M. (in press, to appear in May 2010). "*Knowledge-Based Representation of 3D Media*". International Journal of Software Engineering and Knowledge Engineering.

- Papaleo L., Albertoni R., Marini S., Robbiano F.: "*An ontology-based Approach to Acquisition and Reconstruction*". In Proceedings of workshop toward Semantic Virtual Environment 2005, Villars, Switzerland, (2005).

- Albertoni R., Papaleo L., Robbiano F., Spagnuolo M.: "T*owards a Conceptualization for Shape Acquisition and Processing*". In Proceedings of 1st International Workshop on Shapes and Semantics, Matsushima, Japan, (2006)

- Albertoni R., Papaleo L., Robbiano F.: "*Preserving Information from Real Objects to Digital Shapes*". In Proceedings of Fifth Eurographics Italian Chapter Conference, Trento, Italy (2007)

- Vasilakis, G., Garcia-Rojas, A., Papaleo, L., Catalano, C.E., Robbiano, F., Spagnuolo, M., Vavalis, M., Pitikakis, M. "*A common ontology for multi-dimensional shapes*", In Proceedings of MAReSO workshop, Genova, Italy (2007)

3.1 Semantics and the need of metadata

According to the definition within linguistics, anything that associates a sign (in our case, a 3D object) to any interpretation of it can be regarded as semantics [108]. Semantics is intended as the association of a resource to a meaning, but as the context in which resources are considered evolves (e.g. new research goals are set), the targeted meaning evolves accordingly. Thus, there cannot be any precise definition of "meaning".

Let us focus on the domain of 3D Shape Modeling. At an early stage of this domain, the late seventies, the focus hinged on the representation schemes, and the semantics was intended as the connection between a syntactically valid representation to a semantically sound mathematical model of a solid. The addressed semantics did not go beyond geometry [90]. Since the late eighties, especially in the CAD domain, the necessity of detecting geometric patterns and reusing them for flexible design tasks was felt.

The introduction of another semantic layer was addressed, one of whose aims was to provide fixed taxonomies of form features (feature definition), based on specific meanings and purposes (e.g. machining), and to match geometric patterns in the digital models with them (feature extraction) [96]. In the engineering community *feature-based modelling* technology was then identified as the solution for associating *functional* information to geometric data, and consequently for integrating design and downstream applications [96, 19]. The central goal was to *conceptualize* shape design with classified elements, i.e. features, and add information related to design intent and other specific contexts. Most of current CAD systems are now supporting feature-based modelling, and they provide a better understanding of the relationships between the functional aspects of engineering and part shape.

Recently, with the explosion of 3D data over the internet and the wide availability of 3D repositories, new user communities require to interact with 3D objects, going well beyond the boundaries of restricted and highly-specialized engineering domain. Since the interaction needs of users with 3D media is increasing, it is important to characterize objects also by properties perceivable by humans. Therefore, the target is to couple resources with informative descriptions that can express their meaning. The encoding of such descriptions is often regarded as metadata.

3.2 The evolution of metadata

Metadata is often defined as “data about data”. In the setting of digital resources, metadata is intended as structured information about resources. The focus of this chapter is on descriptive metadata. This kind of metadata (for any kind of resources, including digital objects) were traditionally created by dedicated professionals (catalogers), yielding accurate descriptions by following elaborate rules and schemes for cataloging, often in the form of Machine-Readable Cataloging (MARC) especially regarding books or other intellectual creations [56].

The very low scalability of this approach induced an alternative: letting authors themselves to create metadata, as happened for instance in the late nineties with the Dublin Core Metadata Initiative [62]. Author-created metadata may help with the scalability problems in comparison to professional metadata, but in both cases the eligible users are completely disconnected from the process of metadata creation [56].

3.2.1 Free tagging and taxonomies

An evolution with respect to the aforementioned approaches implies the inclusion of the user in the process, at first only by providing unstructured comments to web links within so-called *weblogs* [22], and then by directly and explicitly tagging content with keywords. This pattern is followed in many systems involving social networks and quick, light interchange of metadata. For instance, the online photo management and sharing application *Flickr* [63] is based on letting anyone tag with free text any available picture; pictures end up being described by means of attached tags. Also the bookmarks manager *delicious* [61] allows users to categorize sites with keywords chosen by the users. The organic system of organization developing in *delicious* and *flickr* was called "folksonomy" for the first time in 2004 by Thomas Vander Wal to intend a combination of the term "folk" and "taxonomy" [111]. This kind of approach is highly scalable, and its large consensus derives from its effectiveness and flexibility achieved through a limited cost in time and effort for the user.

In the case of 3D objects we could assume that to describe objects and their parts it would be sufficient naming them, i.e. shifting to the language of words and using them as tags to describe the objects. Coming to the pitfalls, since a human describer is involved, this approach to the description process is doomed to be manual. This means that an explicit description effort is needed for any involved resource, and the resources that are not annotated are just not described: they can be found only by stumbling against them. The description may be *inaccurate*: the object that was described as a tall man could actually be a small apple: it is not possible to verify the trustworthiness of the describer. The description is *language-dependant* because it uses the encoding of concepts into words typical of a given (natural) language: "apple" and "mela", which means apple in italian, are two different tags, even if they refer to the same concept. This problem occurs also within the same language with *ambiguity* (e.g. same term used with different meanings) and *synonyms* (i.e. different tags used for the same concept, precluding collocation). Moreover, the description is *flat*, because no prior structuring of the terms and no relationships among them is expected.

3.2.2 Structured metadata and ontologies

The use of closed vocabularies and ontologies overcomes some of the aforementioned pitfalls (i.e. language dependance, flatness of description). Ontologies are formal specifications of conceptualizations [39]: in this case a domain is described a priori and the concepts are organized in a precise structure, including subsumption relationships (is-a) between concepts, other relationships between them, and precise attributes for each concept encoded. In order to describe resources the user needs to know the structure of the conceptualization in advance, but is rewarded with a richer expressivity. A concept is unambiguously expressed by a class of the ontology, and so no ambiguity or synonym clash can occur; in addition, since the structure is formally defined, some inference can be performed on the descriptions: knowing that "an object represents a dog" and that "a dog is an animal" can be used to infer that "the object represents an animal". Moreover, since the attributes can be tailored to the concepts, it is possible to introduce further details: beside stating that a person is "tall" or "short" (qualitative characterizations), the user can fill the "height" attribute of the concept "person" and state that the person's height is 180 cm; this would be very difficult to be expressed only with the help of tags, and no inference could be done on the quantitative information.

It is interesting to refer to well-known criticisms that have been made towards a wide adoption of ontologies. Shirky criticizes the fixed categorization that is necessarily cast by any a priori conceptualization. In [98] he claims that the strategy of designing categories to cover possible cases in advance is widely used and badly overrated in terms of its value in the digital world, especially when the considered domain is as broad as the internet itself. Shirky claims that ontological classification works well in some cases, of course:

> You need a card catalog if you are managing a physical library. You need a hierarchy to manage a file system. So what you want to know, when thinking about how to organize anything, is whether that kind of

classification is a good strategy. Here is a partial list of characteristics that help make it work:

Domain to be Organized

- Small corpus
- Formal categories
- Stable entities
- Restricted entities
- Clear edges

This is all the domain-specific stuff that you would like to be true if you're trying to classify cleanly. The periodic table of the elements has all of these things – there are only a hundred or so elements; the categories are simple and derivable; protons don't change because of political circumstances; only elements can be classified, not molecules; there are no blended elements; and so on. The more of those characteristics that are true, the better a fit ontology is likely to be.

The other key question, besides the characteristics of the domain itself, is "What are the participants like?" Here are some things that, if true, help make ontology a workable classification strategy:

Participants

- Expert catalogers
- Authoritative source of judgment
- Coordinated users
- Expert users

DSM-IV, the 4th version of the psychiatrists' Diagnostic and Statistical Manual [60], is a classic example of a classification scheme that works because of these characteristics. DSM-IV allows psychiatrists all over the US, in theory, to make the same judgment about a mental illness, when presented with the same list of symptoms. There is an authoritative source for DSM-IV, the American Psychiatric Association. The APA gets to say what symptoms add up to psychosis. They have both expert cataloguers and expert users. The amount of 'people infrastructure' that's hidden in a working system like DSM-IV is a big part of what makes this sort of categorization work.

This *people infrastructure* is very expensive, though. One of the problem users have with categories is that when we do head-to-head tests – we describe something and then we ask users to guess how we described it – there's a very poor match. Users have a terrifically hard time guessing how something they want will have been categorized in advance, unless they have been educated about those categories in advance as well, and the bigger the user base, the more work that user education is.

It is also possible to turn that list around. You can say "Here are some characteristics where ontological classification doesn't work well":

Domain to be Organized

- Large corpus
- No formal categories
- Unstable entities
- Unrestricted entities
- No clear edges

Participants

- Uncoordinated users
- Amateur users
- Naive catalogers
- No Authority

If you've got a large, ill-defined corpus, if you've got naive users, if your cataloguers aren't expert, if there's no one to say authoritatively what's going on, then ontology is going to be a bad strategy.

Dramatically, this is often the case when we talk about internet-based applications.

3.2.3 Tradeoff

To conclude, there is no precise answer to the question whether it is better to rely on structured or on unstructured conceptualizations. The tradeoff folksonomy/ontology is mostly a tradeoff between flexibility and expressive power; the former is immediate, it does not require an a priori definition of the domain and can be easily exploited by any user (very high scalability); the latter is structured, expressive but implies an agreement on the domain definition and a higher accuracy by the user in describing the resources. For these practical reasons, the best known systems for sharing resources on the web currently rely on folksonomies, but recently a lot of effort is being put to reduce the harshness of ontologies by avoiding complex and monolithic conceptualizations and by gathering as many resource descriptions as possible (lightweight ontologies, RDF composability, identity consolidation, linked data), trying to fulfill the Semantic Web objectives. In some cases hybrid approaches are adopted, e.g. in [35] the proposed method relies on tagging systems like folksonomies, but structured information is extracted and ontologies are derived. The use of controlled vocabularies triggering hints to the annotating user [59] go in the same direction.

A lot of research can be done to exploit and combine the strengths of the folksonomy and ontology approaches, but there are intrinsical limitations related to the assertive nature of concept-based descriptions: the resources to be described have to be perceived by someone who translates his/her own cognitive reactions into a synthetic description: the process is mainly manual, and the description is arbitrary.

Mind that arbitrariness is not a drawback per se, as it requires a cognitive filtering and interpretation of information from the describer; this is helpful when the synthetic effect of description is focused at conveying features that are either subjective or not intrinsic to the digital resource itself: "beautiful", "light", "myself" are examples of the former type, while "belonging to John", "golden", "fragile" are examples of the latter type.

3.2.4 Requirements for the Semantic Web

A certain degree of structuring appears to be necessary, especially when the requirements of the *Semantic Web* are mentioned. The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation [105]. It enables people to share content beyond the boundaries of applications and websites. The Semantic Web has been described in rather different ways: as a utopic vision, as a web of data, or merely as a natural paradigm shift in our daily use of the Web [74]. Beside its most proper definition, one of the most important purposes is to share data, and not just documents. "Now I want you to put your data on the Web" – Tim Berners-Lee said at a talk hosted by the Technology, Entertainment, Design organization in 2009, introducing the concept of Linked Data.

In few simple words the main need is to share data, and not only documents, the difference being a degree of structure enabling automatic understanding of the pushed content. If I write the sentence "Francesco Robbiano was born in Genova, Italy" almost any human user would understand what I mean, but if I can rely on a solid structure I can share the same information through a triple: {`FrancescoRobbiano`, `birthplace`, `Genova`}; the first being an individual, the second a relation connecting individuals to place, the third a place. This is much more, because in the latter case the information is ready to be consumed, processed, federated and reused automatically, just paying the fee of an a priori conceptualization. Whether it is a high or a low fee is still a matter of debate. Surely the original enthusiasm towards complex ontologies is being recently substituted by fancying lightweight conceptualizations. As soon as information is encoded in the clear, structured manner of RDF [72], also lighter conceptualizations are welcome. Actually, nowadays they are often even more appreciated: lighter ontologies means less a priori categorization, which means more flexibility, which means more chance to have data interconnected and fruitfully shared on the (semantic) web.

This is the rationale behind Linked Data, whose main focus is not to have complex and separated conceptualizations but simpler and cross-linked ones.

Also for this reason the document-sharing activities (more familiar for users) and the data-sharing activities (more useful for the Semantic Web) are not intended as separated tasks, with the risk of still having loads of contributions in the former case but much less in the latter. As a matter of fact, there are strategies to embed structured (i.e. "semantic") data within plain HTML. One example is the use of *microformats* [68], a sort of *semantic markup* based on existing XHTML and HTML tags to convey metadata and other attributes which allows for the encoding and extraction of events, contact information, social relationships and so on. Another example is *RDFa* [65], a

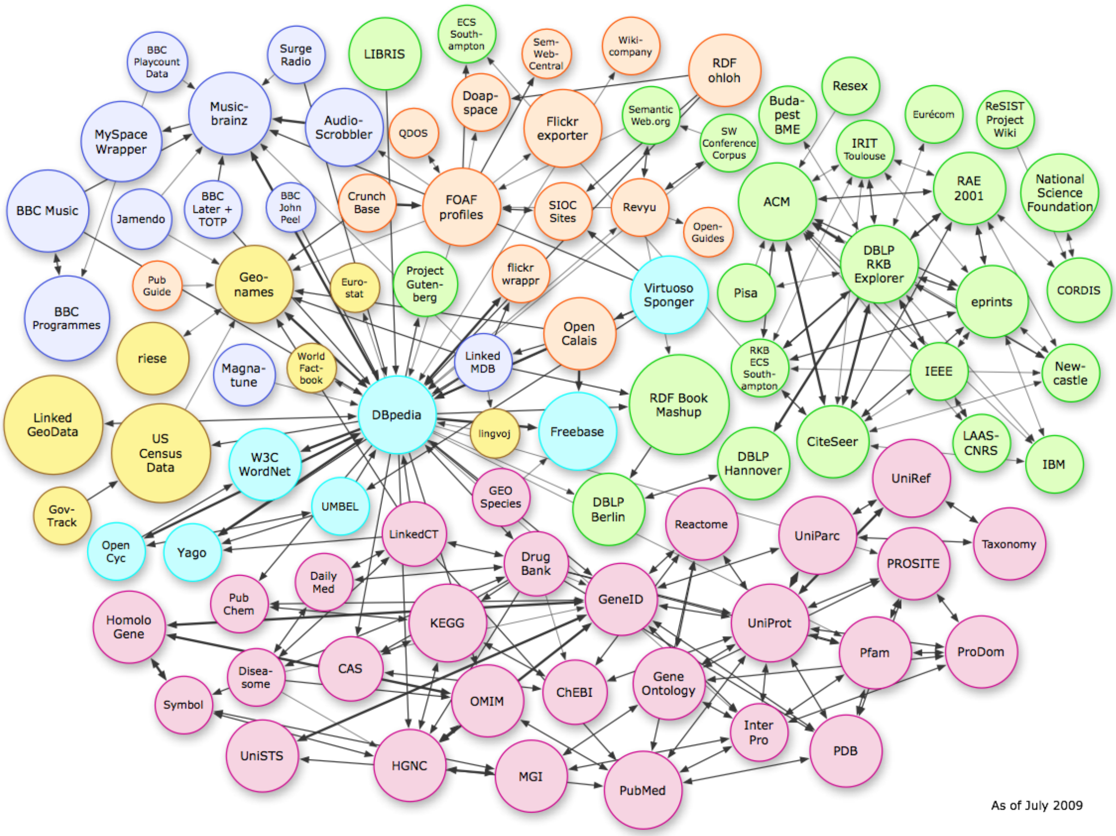

Figure 3.1: Linked Data Cloud as of July 2009

mechanism for embedding RDF triples in HTML but also for enabling the extraction of those RDF triples by compliant user agents.

3.3 The AIM@SHAPE Ontologies

The above considerations stemmed also from the personal experience that I had in developing of shape-related ontologies within the AIM@SHAPE Network of Excellence [79], which I will introduce and present in the following.

AIM@SHAPE made a pioneering effort towards the promotion of a new semantics-oriented approach to model, retrieve, process and share knowledge related to multi-dimensional content in order to facilitate its re-use for producing new content. AIM@SHAPE addressed a wide domain of media by focusing on digital shapes as a generalized concept which encapsulates any multi-dimensional media characterized by a visual appearance in a space of 2, 3, or more dimensions [34], with a particular emphasis on 3D models and animations. The general approach of the work done in AIM@SHAPE consisted in decoupling the formalization of knowledge related to the geometric representation of digital shapes from the formalization of knowledge about them in specific application domains. In this context, it has been decided to develop dedicated ontologies to this aim. The development of the AIM@SHAPE ontologies has been made according to the OntoKnowledge approach [100] and they are expressed in OWL-DL [70]. Note that, in the following, what is written like `This` is actually an entity in the ontologies that have been developed using Protégé [81].

The knowledge related to the application domain in which shapes are manipulated is another key ingredient, as the application context has a significant influence in the way the shape is processed and interpreted. Therefore, the formalization of the geometric knowledge ensures scalability in the process of building application-specific conceptualizations.

The entities within the ontologies have to provide a thorough characterization of shapes (see Figure 3.2) by storing:

- 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),

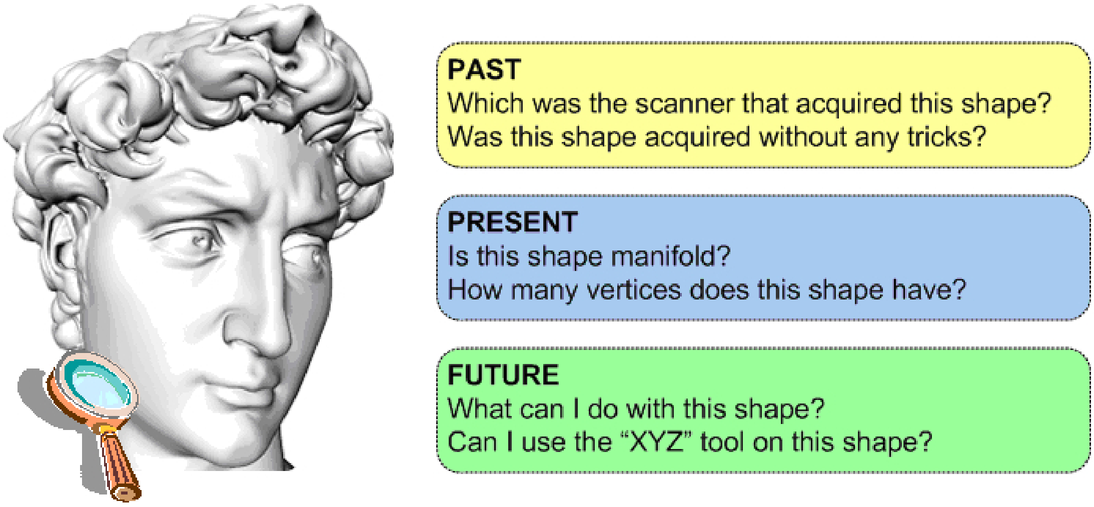

Figure 3.2: An expressive characterization of an object is made up by the information related to its history, the information intrinsically held by its shape and the information related to its capabilities

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

Our desired ontologies should be able also to represent different levels of sophistication describing a 3D object 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). The digital model of a teapot can be seen as 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) or it can be seen a teapot composed by a handle, a spout, a body and a tip. It is important also to take into account the different contexts where the shape can be used since the specific application determines relevant characteristics. For example, if the main purpose is to create 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. The existing branches of research in the field of Shape Modeling are interested in one or more of the above mentioned characterizations, but also on the conditions and the tools to pass from one characterization to another.

Finally, it is important to note that 3D objects play a central role in Shape Modeling, but they do not represent the only kind of resource that must be characterized in the common framework. Every day, scientists work with shapes, tools and publications. It is important to devise the role of these resources in different conceptualizations, making relationships among them explicit. For example, a scientist may want to evaluate his/her latest implementation of a method. In this case, it is interesting to figure out which are the tools providing other implementations of the same method, the publications related to the above tools and methods, or the shapes used as tests for the other implementations (e.g. for testing/benchmarking activities).

According to the above considerations, within AIM@SHAPE we developed two domain-independent ontologies, the Tool Common Ontology and the Shape Common Ontology [107], and three domain-dependant ontologies which rely on the common concepts and can interoperate thanks to them: the Shape Acquisition and Processing Ontology [3], the Virtual Human Ontology [41] and the Product Design Ontology [26].

The common ontologies have the potential to achieve interoperability across domains and encapsulate the fragmented knowledge, captured by the different ontologies, under a unified framework. In the AIM@SHAPE network, the common goal is the ability to reason, to re-use existing knowledge and to extend and create new knowledge about shape resources and tools. In essence, the common ontologies integrate all the metadata information from the Shapes Repository and the Tools Repository within the so-called Digital Shape Workbench [77]. The primary goal is thus to provide an ontology-driven specification for objects representing a shape or a tool, including other helpful information (like information related to the creation of a resource, usability information, etc.) as well. In the specific domains, captured by the dedicated ontologies, a shape or a tool is expected to be characterized by this common metadata information and potentially some domain-specific metadata information as well. My main contributions have been in the design and the development of the Shape Common Ontology and of the Shape Acquisition and Processing Ontology, even if minor contributions where given in all the aforementioned ontologies. In the following, more details on these conceptualizations is given.

3.3.1 The Shape Common Ontology

The Shape Common Ontology (SCO) covers the most essential aspects of knowledge pertaining to the geometric representation of digital shapes, while the full spectrum of information carried or implied by digital shapes is expressed by domain-specific ontologies for which the SCO works as a shared keystone. An overview of the Common Shape Ontology structure, where the most important concepts are shown, is given in Figure 3.3. As it can be seen, the basic structure of the SCO ontology is simple enough to promote reusability. In order to understand the rationale behind the choices made for the conceptualization, it is important to keep in mind that the intended target of the ontology are the scientific researchers, and that the information is not only related to the shapes themselves, but also to their role inside the AIM@SHAPE Shape Repository [78]. A domain application will use the SCO to handle geometric or structural characterization of the shapes as well as information about their storage, grouping, provenance and ownership. This will become clearer after the description that follows, which identifies the most important aspects to potential application scenarios described later in this Chapter.

The most important concepts in the ontology are the `Shape Representation` class and its specializations, whose instances are the actual digital shapes. First of all, a digital shape can be regarded as a generic resource; thus, a `ShapeRepresentation` is an abstract concept encapsulating information that is inherent to the shape model itself. The users are typically interested in getting information about the contact person or institution associated with a shape, and therefore specific relations address the creator, the owner, the contact and the uploader of a digital shape. Another simple yet important way to look at a digital shape is to consider it as a file. For this reason each shape can be related to a `FileInfo` instance, in which the information about the name, the size, the format and the URL of the file are stored. In our conceptualization different shapes can be clustered in a single group, and each group may be characterized by a representative shape (mainly for visualization purposes, the shape which stands for the whole group). Furthermore, subdivisions in subgroups may take place, which reflect a possible hierarchy or generation order between the models. There are different reasons for the need of creating the `Group` concept. For instance, possible reasons for grouping different shapes are:

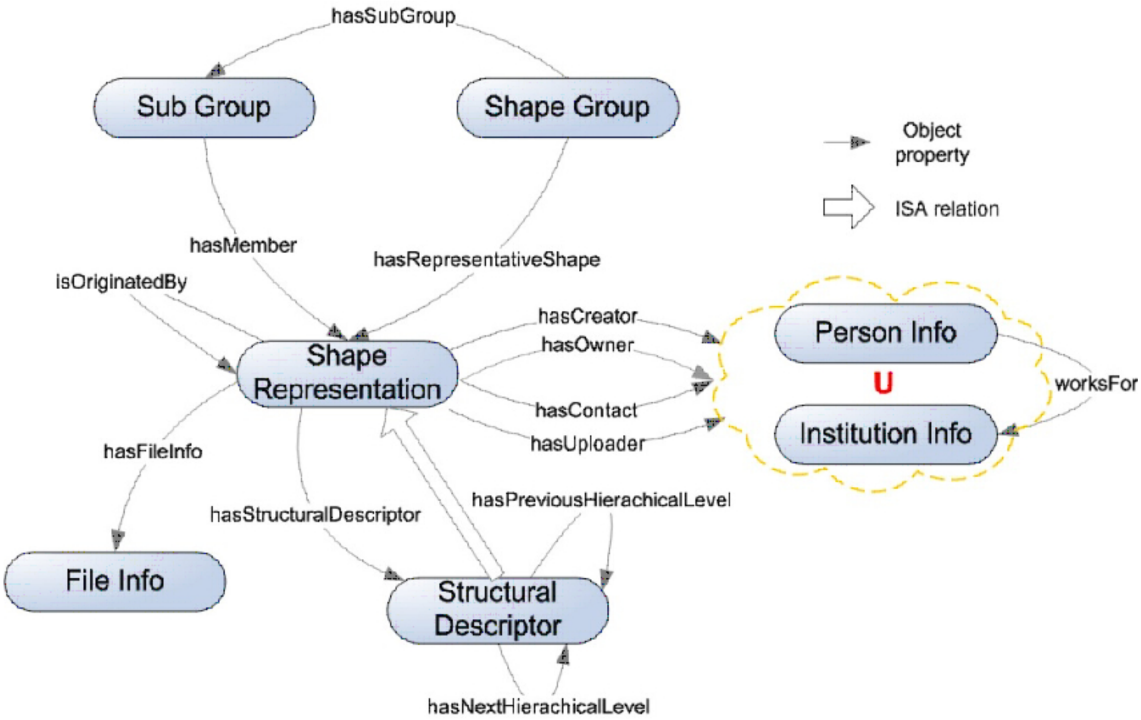

Figure 3.3: An overview of the structure of the Shape Common Ontology.

- they are all parts of a more complex CAD model (in this case the representative shape could be the entire CAD model);
- they constitute the benchmark eligible for running tests on specific algorithms;
- they represent variations, products or by-products of the processing stages of an original shape;
- they are the results of different scans in an acquisition phase, which will possibly be registered, combined, and merged in a unique 3D shape. In this last case it is likely that the representative shape of the group would be the final shape.

The core of the SCO is the conceptualization of the `ShapeRepresentation` concept. It should be noted that the goal is not only to provide a useful categorization of the digital shapes, but also to provide each category with its own specific attributes and relations. An overview of the hierarchy rooted in the `ShapeRepresentation` class is shown in Figure 3.4. The different class levels are drawn in different color in the

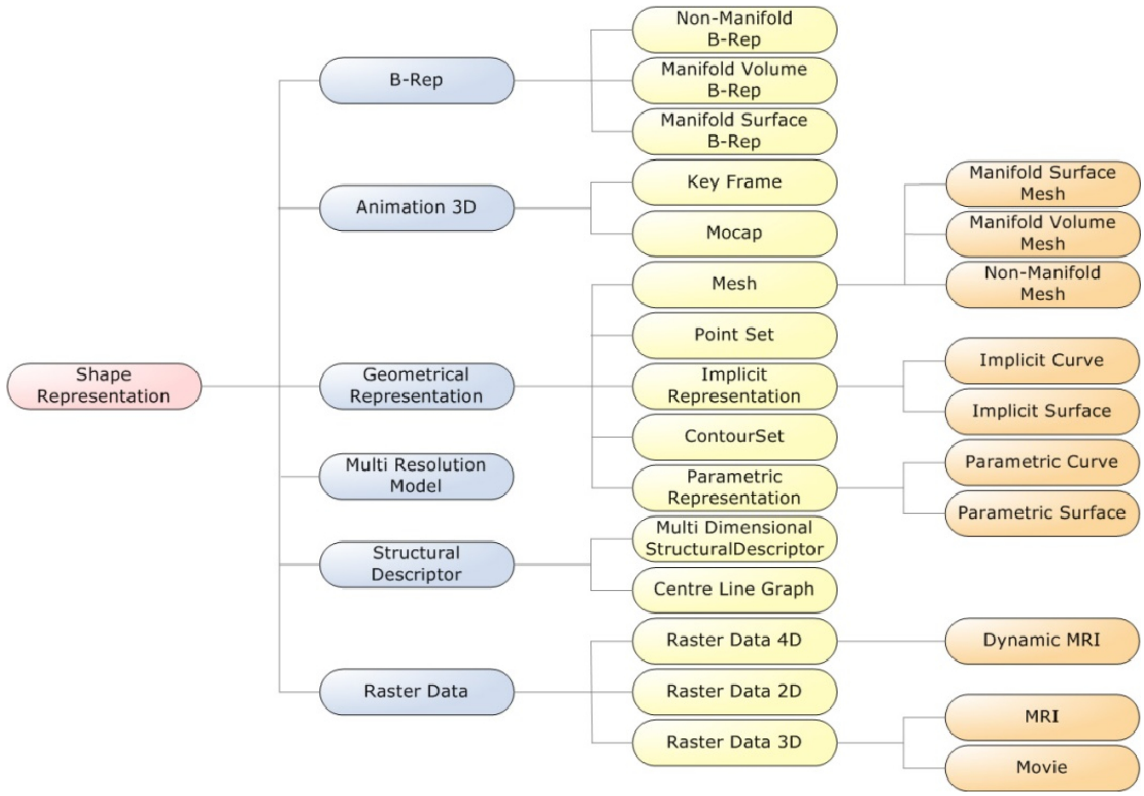

Figure 3.4: An overview of the hierarchy rooted in the Shape Representation class.

diagram. First level classes are shown in light blue color, second level classes are shown in yellow color and third level classes are shown in orange color in the hierarchy.

An overview of the first level follows. Firstly, the `GeometricalRepresentation` class includes shape descriptions based on geometry, while the `B-Rep` class gives more emphasis to the topological information of the shape. The two classes are not disjoint, since formally a mesh is a B-Rep (boundary representation) and the choice of classifying a shape as belonging to one class or to the other depends mainly on the application context. In fact, the Computer Graphics community commonly adopts a mesh description for shapes and the terminology is definitely standard today, while other fields such as CAD traditionally prefer to use the more general B-Rep description. The boundary representation defines objects in terms of faces, edges and vertices which make up their boundary. The properties identified in the SCO favor the topological aspect, considering for example the continuity degree between the faces and its topological complexity.

The attributes defined for the different subclasses of `GeometricalRepresentation` focus on geometrical aspects. The `MultiResolutionModel` class formalizes models represented in a way which allows for a manipulation of geometry at different resolutions, enabling both local and global modification, and modulation of details at different frequencies. The main properties here are related to the granularity of the model, to the minimum and maximum resolution of the models contained and to the method used to simplify recursively the original shape. The `Animation3D` class collects information related to the animation of a shape and can have relationships with the geometrical and structural representation of the shape. The `StructuralDescriptor` class models the structural views of 3D shapes and refers to decompositions of a shape into its relevant parts, together with the adjacency relationships among them. Structural descriptors can be used for an efficient classification, recognition, comparison, and retrieval because they provide a meaningful abstraction of a shape. One property of this class refers to the creation method of the specific instance and, in case a center-line graph is obtained, the information related to the number of arcs and nodes as well as other typical properties of graphs are included as properties. Finally, the `RasterData` class formalizes the information stored in a grid of cells; raster data are commonly used to represent images (2D raster grids), videos and MRI volumes (3D raster grids). The properties related to this class include information about the grid and the single cells, such as dimension, intensity values and RGB values. More specialized classes and their corresponding attributes have been modeled in the subclasses, which are not reported here. For a complete overview of the ontology and the meaning of the different concepts, the Digital Shape Workbench (DSW) [77] can be browsed, which also includes a glossary with short descriptions for relevant concepts and terms.

3.3.2 The Tool Common Ontology

The common ontology for tools (TCO) captures tool-related metadata and constitutes an ontology-driven evolution of the metadata in the Tools Repository. A high level view of the ontology is shown in Figure 3.5.

The concept `PersonInfo` is shared with the SCO and captures information regarding a person that is in some way (creating, ownership, uploading, etc.) associated with the various tools that are stored in the repository.

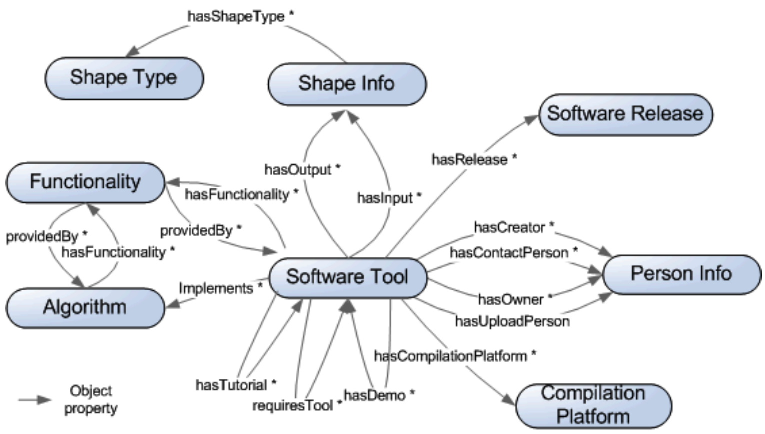

Figure 3.5: The main concepts in the Tool Common Ontology.

The central concept in the ontology is `SoftwareTool`, which describes the various tools that are stored in the repository. A tool can be associated with other tools, through the relation `requiresTool`. It has functionality, specified by the concept `Functionality`, and it may implement one or more `Algorithms`.

A tool can be further related to a `CompilationPlatform`, under which it can be operated, and has several `SoftwareReleases`. Finally, each tool accepts specific input and provides specific output. The input and output of a tool can be one or more shapes, which are described by the concept `ShapeInfo` and are associated with a specific `ShapeType`.

`SoftwareTool` constitutes the base concept in the ontology which is further extended by related concepts defined in the domain ontologies. A tool in the ontology can be further specialized to the concepts of a `Library`, a `Macro`, a `Plugin`, and an `IndependentApplication`.

3.3.3 The Shape Acquisition and Processing Ontology

The purpose of the Shape Acquisition and Processing (SAP) ontology is to describe the development, usage and sharing of hardware tools, software tools and shape data by researchers and experts in the field of acquisition and processing of shapes. In the creation of the SAP ontology, the following macro-steps have been considered: Shape Acquisition (and Registration): that is the phase in which sensors capture measurements from a real object; Shape Processing, that is the phase in which all acquired data

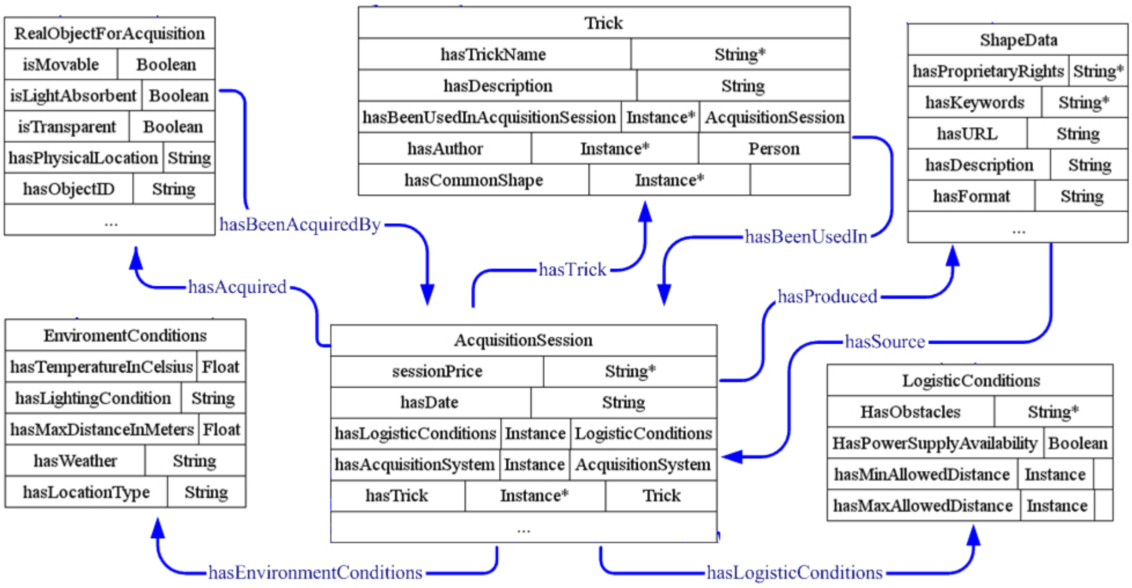

Figure 3.6: A zoom on the AcquisitionSession entity in the SAP ontology.

are merged to construct a single shape; or in which further computations on the shape may be done (e.g. smoothing, simplification, enhancement, and so on). Thus, the target applications are related to acquisition planning, data validation, benchmarking, testing and data enhancement (e.g. automatic recovery). In the following, more details will be given on the conceptualization, presenting a sort of zoom on the acquisition phase.

The acquisition process basically deals with an acquisition session which takes place considering a particular real object and producing a digital shape on the basis of certain conditions. In SAP, the `AcquisitionSession` has been modeled as an entity and an overview of this entity is given in Figure 3.6. The `AcquisitionSession` is related to an `AcquisitionSystem` (which is made up by one or more `AcquisitionDevices` - e.g. scanners) and to the `AcquisitionConditions` in which the acquisition is performed.

The most significant relations are highlighted by arrows. Each rectangle represents an entity. The rows in each entity 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. 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) or `EnviromentConditions` (which include the information on is the type of environment - indoor, outdoor or underwater, the level of humidity or even the weather). Moreover, some attributes are directly related to the `AcquisitionSession` (e.g. the price for renting the technological devices), while others are related to the different entities in the framework (e.g. the person/institute responsible for a scanning system). An `AcquisitionSession` basically documents the acquisition of a `RealObject` and the production of a `ShapeData` (a digital shape), using a particular `AcquisitionSystem`. A `RealObject` has also been modeled as an entity, and the knowledge related to it and to its context is thus preserved: in the ontology are recorded the location of the object, the possibility to move it, whether or not it is transparent or light-absorbent, and so on.

Note that the mentioned characteristics (e.g. transparency and being or not light-absorbing) have immediate impact on the `AcquisitionPlanning`. For instance, a `Trick` can be used when there is a problem of compatibility between the `AcquisitionSystem` and the `RealObject` to be scanned: a light absorbent object and a laser scanner might be incompatible, but if we need to perform the scanning, it is possible to avoid the problem by spreading powder over the object before scanning. Otherwise, it can also be possible to plan the acquisition with another (compatible) `AcquisitionSystem`.

`ShapeData` (which identifies a digital shape) has also been modeled as an entity in our ontology, with some specific 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 has-Source) or the information on the owner of the particular shape (through the slot `hasOwner`): the owner can be an Institution or a Person.

A `ShapeData` can be based on another (or more than one) `ShapeData`, 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.

The ontology introduced so far, even if here only partially described, is already sufficient to describe the macro-step of the acquisition of a real object. Such a simple description provides the basics to embed in the digital shapes information that usually gets lost after acquisition; this information might result important for comparing shapes coming from different providers, for improving the assessment about their quality and for better understanding the results arising from further processing.

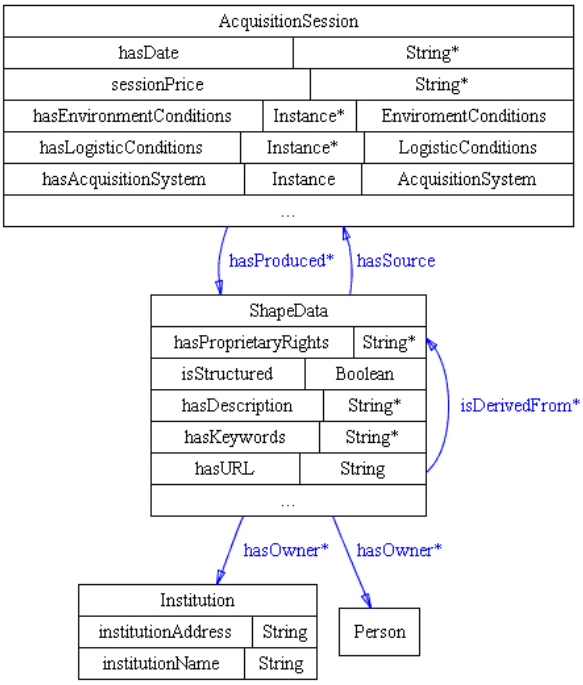

Figure 3.7: `ShapeData` entity and its relation with the `AcquisitionSession` entity.

3.3.4 The Virtual Human Ontology

Virtual Humans, as 3D graphical representations of human beings have a large variety of applications. Within inhabited Virtual Environments, Virtual Humans (VHs) are a key technology that can provide virtual presenters, virtual guides, virtual actors, and be used to show how humans behave in various situations [103].

The infrastructures that would be required for sharing large databases of Virtual Human models and populating Virtual Environments are still not mature; moreover, a common understanding to share these models and their information does not exist. AIM@SHAPE contribution is a semantics-based method for organizing the various types of data that constitute a Virtual Human, in order to foster a common understanding and sharing of such complex 3D entities. The knowledge related to the synthesis, animation and functionalities of VHs is formally specified in the form of an ontology. The Ontology for Virtual Humans aims at organizing the knowledge and data of three main research topics and applications involving graphical representations of humans:

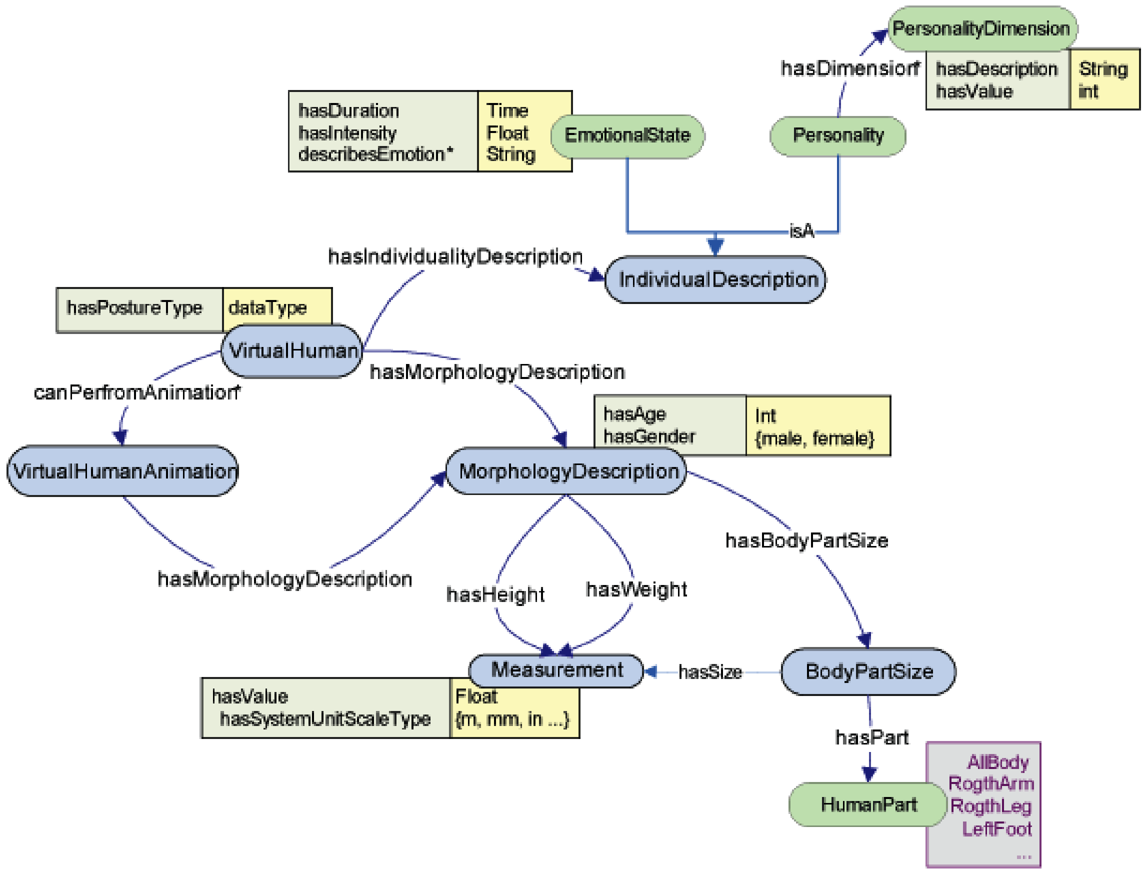

Figure 3.8: Overview of the Virtual Human Ontology

- Human body modeling and analysis: morphological analysis, measuring similarity, model editing and/or reconstruction.
- Animation of virtual humans: autonomous or pre-set animation of VH.
- Interaction of virtual humans with virtual objects: virtual -smart- objects that contain the semantic information indicating how interactions between virtual humans and objects are to be carried out.

The following are the concepts that have been defined to express in a formal way the information and knowledge associated to Virtual Humans (see Figure 3.8):

Geometry: The geometry is the physical visual representation of the Virtual Human and is composed of two parts; the primary is body shape, which refers to the SCO through the relation `GeometricalRepresentation`, and the secondary part is composed by accessories, garments, etc.

Animation: Virtual Human's animation should distinguish between facial and body animation because the way of animating each part is different. Body animation can be `KeyFramed` or `MotionCaptured`. In order for an animation to take place, the character should have a skeletal structure. This structure is defined in the `StructuralDescriptor`, based on the H-Anim specification [67].

Morphology: `MorphologicalDescriptor` contains information like: Age, weight, height, gender.

Behavior: `IndividualDescriptor` and `BehaviorController` are for describing the behavior: the Individual descriptor contains the constant definition of the behavior of the Virtual Individual like his personality or cultural identification, background; the behavior controllers are algorithms that drive the behavior of the character considering the emotional state and its individuality.

3.3.5 The Product Design Ontology

Product design is the first phase of the overall product development process, which deals with all the aspects concerning the realization of an artifact. Due to worldwide competition and technological improvements in the last years, product time-to-market has been reduced and specialization in the Product Development Process (PDP) has been growing. PDP is a very complex process which requires different expertise, according to the specific activity considered. Due to such change of mentality in the design activity, companies and actors of the PDP need to have access to the right information at the right time in a usable format in order to perform an efficient job. It follows that PDP requires not only a large number of information and data, suitable for any specific application, but also a strong interaction among the actors to share and retrieve product data. The Product Design Ontology (PDO) focuses on the annotation and retrieval of shape information in two specific tasks of the PDP, namely the free-form modeling and the engineering analysis. Therefore, it is strictly interconnected with the SCO since the goal in this ontology is to assist researchers who need information related to the shapes and tools intervening in the two mentioned tasks.

The core components of the PDO are (see Figure 3.9):

- shape types and shape representations with product design related metadata;
- shape processing tools and algorithms employed in the product design context;

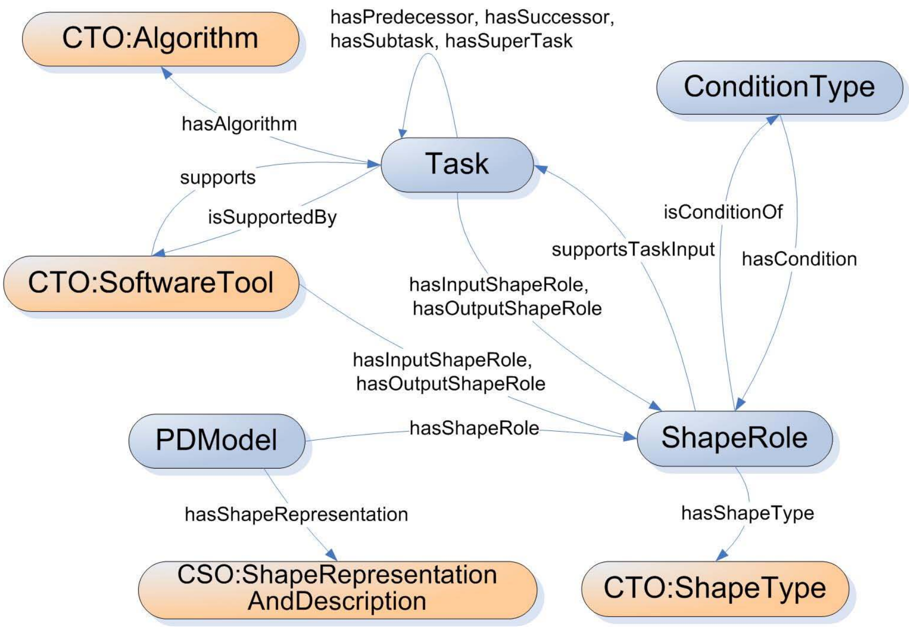

Figure 3.9: Product Design Ontology: core concepts

- tasks accomplished by software tools applied to shapes, which are both data sources and results of product design tasks along the product design flow;
- conditions, either geometric or not, which are related to a shape model when performing a specific task;
- groups, which permit to gather together digital models that share some interrelation.

In particular, the different tasks involved in the Product Design workflow have been formalized, together with the different roles a shape can play within the shape life cycle (e.g., the input and output of the simulation task are surface or volume meshes enriched with proper simulation data and properties), and the requirements and conditions a shape should satisfy to fulfil a given task.

The `Task` concept is the central unit of any pipeline within the PDP and is strictly connected with shapes and tools. Each instance of `Task` represents an activity in the PDP, which involves the application of shape processing tools on some shape model. When performing a specific a task, the digital representation of the shape is always equipped with additional context-dependant information. The concepts of

`ShapeRole` and `PDModel` have been introduced to model respectively shape types and shape representations, enhanced with extra data, for instance regarding conditions to be applied on the shape to fulfil the task activity.

The `ShapeRole` concept enriches shape types description (e.g., volume mesh, BRep) with additional information intervening in a specific task of the Product Design workflow. It is particularly useful for inexperienced users, because it provides the user with all the expertise necessary to execute the whole development process. For each task in the PDP, the user can learn about: the shape type required; the conditions to be verified in order to complete it; the type of results returned and the additional information provided.

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

As mentioned above, the concept `ShapeRole` specialises shape types related to a specific task of the Product Design workflow. This specification includes information about the conditions a shape of the corresponding type has to satisfy in order to perform that task, hence modelled through the concept `ConditionType`. Several types of conditions can be used for enriching a shape, that is, characterising a shape role. We concentrated on geometric conditions, which have been further specialised to distinguish the geometric properties applying to different shape types and boundary conditions, which are associated to a mesh during the analysis stage.

3.3.6 Usage Scenarios

It is interesting to demonstrate that the Shape Common Ontology can be successfully used in domain-specific ontologies. To this aim, in this section I will present two specific user scenarios. The first one is related to the acquisition of a human shape and the production of an animated virtual human while the second one concerns the product development process.

Acquisition of a Human Body

The first scenario on which I will focus is related to a human body acquisition for creating an animated virtual human starting from a real person. A scenario like this is crucial for those applications aiming at making virtual simulations involving humans, such as the population of Virtual Environments, where one of the main challenges is to create a large diversity of human characters to fulfill the demand of a large amount of users. This example requires the organization and maintenance of information at different levels from the geometrical aspects up to the description of abstract concepts such as the personality and emotional traits to individualize Virtual Humans. The scenario is based on two domain-specific ontologies: the Shape Acquisition and Processing (SAP) and the Virtual Human (VH) and it uses also concepts from the SCO ontology, extending the SAP and VH ontologies the SCO. In the description of the scenario, I will refer to concepts and instances belonging to the three ontologies using prefixes. In particular:

- "SCO:" when a concept belongs to the Shape Common Ontology;
- "SAP:" when concept belongs to the Shape Acquisition and Processing Ontology;
- "VH:" when the concept is modeled in the Virtual Human ontology.

Note that, the SAP and the VH formalize concepts that are relevant for them without neglecting information in SCO, which is also relevant for domain applications. In Figure 3.10 the different concepts and relations involving the human shape acquisition are depicted. The scenario is presented as a workflow of actions (scanning, reconstruction, analysis and synthesis) for obtaining an animated virtual human (instance of VH:`VirtualHuman`) from a real object (the human person - instance of the concept SAP:`RealPerson`). Every action is foreseen in the conceptualization of the corresponding domain ontology.

Focusing on each specific action, the process starts with the scanning session (instance of SAP:`AcquisitionSession`), where we acquire a points cloud (instance of

SCO:`PointSet`), which is a set of points in a 3D space, from the real person. This acquisition can be performed with a dedicated scanner, a set of cameras or any other suitable acquisition system (instance of SAP:`AcquisitionSystem`). The acquisition session modeled in the SAP formalizes all the necessary knowledge related to the acquisition phase, including the logistic and environmental conditions under which the scanning has been performed. Furthermore, detailed information about the acquisition system is maintained. Following the acquisition session, and starting from the points cloud produced, a surface reconstruction session is started (instance of SAP:`ToolSession`). The reconstruction is carried out with specific software tools (instances of SAP:`SoftwareTool`), which performs meshing, merging and hole filling operations. Finally, a non-manifold surface mesh (instance of SCO:`NonManifoldMesh`) is created. At this step, we already have a geometrical digital representation of the real person. However, we still need to analyze the shape in order to create the attributes that will allow us to generate the virtual representation of the real person. This means that we need to add an internal structure so that the mesh may be deformed and an animation may be applied. This step requires making an analysis of the shape for its segmentation, annotation and mapping. A phase of analysis and mapping is therefore started (again, an instance of SAP:`ToolSession`) which uses a specific tool (e.g. Plumber, instance of SAP:`SoftwareTool`). From this step, we obtain as output a structural representation of the shape (EllaBody, instance of SCO:`MultiDimensionalStructuralDescriptor`) which can be represented, for example, in an h-Anim format as defined in [21]. In the phase of synthesis, the intervention of an expert is necessary, in this case a designer, who can create a 3D character from the previous annotated shape and add the needed features such as an skeleton and textures to be able to use the virtual human inside a 3D environment.

We can further describe this final shape object with respect to another specific domain, which is captured by the VH ontology. The `VirtualHuman` concept is a human shape that has a geometry and a skeletal structure (VirtualHumanElla becomes an instance of VH:`VirtualHuman` because it has `Geometry` and `Skeleton` in its EllaBody). This final geometry with skeletal structure allows to populate a 3D environment with this character and to apply animations on her.

During this creation pipeline the history of the shape is stored in the SCO. This allows us to answer queries such as: What shape originated from shape "EllaMesh"?,

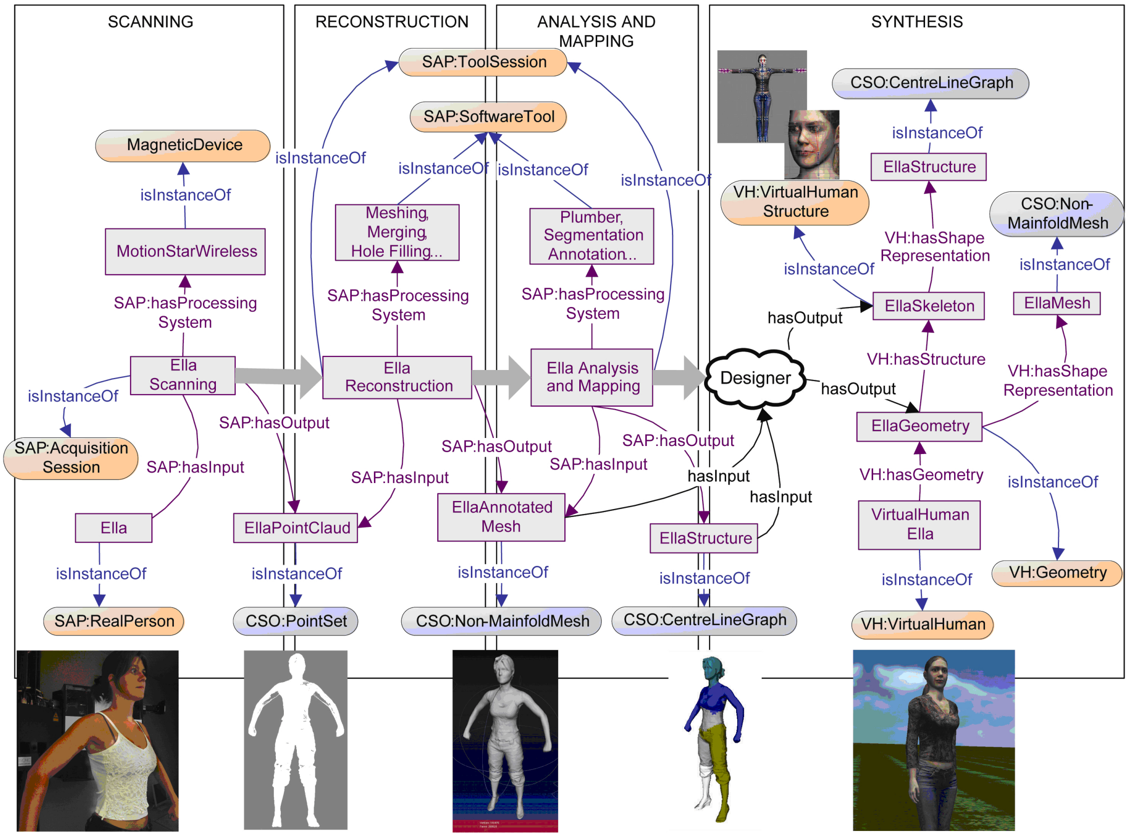

Figure 3.10: Acquisition and Processing of a Human Shape.

What kind of structure conforms the skeleton of this Virtual Human? Which shapes were generated from the shape “EllaPointCloud”? Who is the owner the shape “Frog”?. Furthermore, the SAP and VH also serves in answering domain specific questions, e.g: Which software was used to annotate the shape “EllaAnnotatedMesh”? Which is the real person used to create the animated virtual human “Michela”?, Under which lighting conditions did the real person create this virtual human?, What animations can be used by this virtual human?

Digital Product Workflow in Simulation

Here I will present a typical usage scenario of the PDO related to the engineering analysis, also mentioned as simulation phase. It evaluates the physical behavior of any engineering component of a product, which is subject to various kinds of loads and conditions, ranging from structural analysis to thermal and electrical analysis, and so on. As in the case of the previous scenario, I will use prefixes in the concepts and instances in order to describe their belonging to one of the two ontologies involved. In particular:

- “SCO:” when a concept belongs to the Common Shape Ontology;
- “PDO:” when the concept belongs to the Product Design Ontology.

The CAD model used to design the product is usually represented by parametric surfaces, which are suitable for manufacturing purposes, but not for performing a Finite Element Analysis (FEA). Therefore, the initial design model generated by a CAD system (in the picture the initial model is an instance of SCO:`ManifoldBRep`) needs to be converted into a FE mesh, the model required to run a simulation. In the PDO the input of the simulation is a digital shape, instance of PDO:`SimulationModel`. Consistently, the role (PDO:`ShapeRole`) of a simulation model is PDO:`FiniteElementMesh`, in particular, PDO:`PreSimulationMesh`, and has a shape representation that is an instance of SCO:`Mesh`.

More precisely, a FE mesh is a mesh which satisfies typical geometric conditions. Then, the subtask PDO:`GeometricDesignEvaluation` is dedicated to the verification of the geometrical model. In fact, through a specific attribute of PDO:`ShapeRole` for PDO:`PreSimulatioMesh`, the necessary geometric properties for the specific simulation are listed, while through the metadata associated to the SCO:`Mesh`, it is possible to check if the mesh representing the engineering component has the required properties. If it does not, dedicated software tools included in the DSW can be acquired and utilized for correcting the mesh.

To reduce the complexity of the simulation it often happens that the design model is simplified, removing shape details which do not influence the results of the engineering analysis. Such operation can be applied both on the design model (as in Figure 3.11 where the small holes disappeared) and on the FE mesh after the conversion. If a simplification is required, the role of the design model (or FE mesh) becomes PDO:`SimplificationModel` and a simplification task (PDO:`ShapeSimplification`) appears in the PDO: it mainly consists of a suitable editing and rearrangement of the geometric elements in the shape and therefore all the properties required to perform the simplification correctly and the associated queries refer directly to the SCO scheme. Once the suitable model for simulation has been set, specific boundary conditions have to be imposed. They are physical conditions which describe the interactions of the component at the boundaries of the simulation region. In the PDO such activity corresponds to the task PDO:`DefinitionOfBoundaryConditions` and a taxonomy of Boundary Conditions, that is PDO:`BoundaryConditionType`, has been included, which subdivide them according to the specific simulation type (e.g., structural mechanical, electromagnetic, thermal analysis).

Now the simulation can be executed in the task PDO:`Solving` and the output shape is an instance of the PDO:`SimulationMesh` with the role of a PDO:`PostSimulationMesh`. Belonging to such class implies that the simulation results are associated to the geometric part. In the task PDO:`SimulationPostProcessing` the simulation outcome is interpreted considering also the influence of the shape details removed in the first phases of the process (in Figure 3.11 the small holes have been included again), and finally decisions are made about the suitability of the design with respect to its engineering specification. This conceptualization allows us to answer to a large set of queries such as: What type of conditions should the model 'Carter' have to fulfill before performing the Solving task?, Which kind of geometric checks do we have to consider when performing the `ShapeSimplification` task?, Which software tools are helpful to detect possible self-intersections on the model 'Carter'? What are the PDModels whose `ShapeRole` is `PostSimulationMesh`?

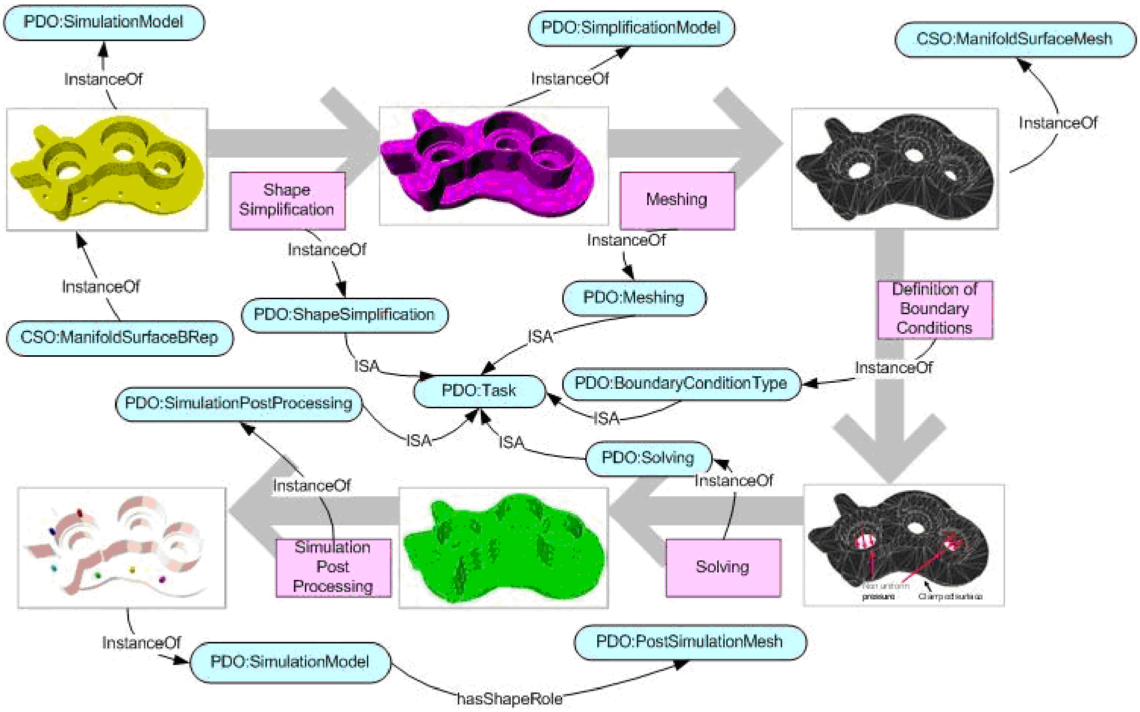

Figure 3.11: Tasks of the product design simulation process. Some elements in the diagram refer to concepts in both the PDO and the SCO. All the boxes are instances of concepts which are sub-concepts of PDO:`Task`.

Chapter 4

Content

Gli chiesero di descrivere se stesso in due caratteri.
Inizialmente pensò: IO.
Poi ripensò: SCHIZOFRENICO.

The term *content-based* retrieval has been introduced in the early 90thies to indicate a class of methods to search for visual content using features, or descriptors, that can be derived automatically from the resource itself without resorting to manual inspection of the content or to concept-like annotations such as captions or keywords. Some of these features have been used also to describe resources using standards such as MPEG-7 [69], which are nowadays broadly used.

The role of this Chapter is to discuss content-based descriptions for 3D objects and to analyse a number of techniques that have been developed in the literature.

My contribution in this chapter is, first of all, to consider some pros and cons of the adoption of a widespread standard for the purpose of describing 3D objects, i.e. MPEG-7. Afterwards I will present a quick overview on the most used 3D object descriptors. As a matter of fact, no single descriptor is capable of capturing all and only the meaningful content. The state of the art presents several methods, and among them a number of geometric and topological descriptors, each of them focusing on specific features: they capture some geometric or topological characterization of the considered objects and discard everything else.

The perspective of my survey, though, is not the classical one taking into account mainly the technical features of such methodologies: the emphasis is on the high-level characteristics that are preserved or discarded by each description technique. This allows for an explicit qualitative approach to the mentioned methodologies: *which kind of information of the original asset does this technique capture?* And, on the other hand, *which kind of information does this technique discard?* This approach will be the basis on top of which an explicit context-based framework could be developed (details in Chapter 5).

4.1 Content-based descriptions for 3D objects

For several kinds of resources, 3D objects making no exception, there are a lot of features which can be extracted from the digital resource itself. In some cases, these features are available off the shelf (they are explicitly encoded, e.g. the name of a song in a Mp3 file), and in other cases they are the outcome of an analytic process (they have to be calculated, e.g. the saturation of an image or the volume of the bounding box of a digital object).

When the description of resources is based on the latter type of features, it is known as *content-based*, as it is based on the intrinsic content of the digital resource, without the need of a human mediation [99]. As it may be noted, there are pros and cons with respect to the concept-based descriptions: content-based descriptions do not need to be synthesized by a human annotator, and therefore the whole process can be automated. Nevertheless, the features extracted automatically from the digital resource need to be reinterpreted to be offered to the user's cognition; also, the features that can be extracted are bond to the type and to the power of the tools used to analyze them.

Note that content-based descriptors are very powerful especially in the domain of 3D objects. This kind of resources is highly informative (there are no problems regarding occlusion, distorsion, skewing, as the spatial information is complete), and therefore analytic tools may extract very useful information.

As written in Chapter 2, the effective sharing of resources is intimately connected with the ease of accessing them. When resources are approached in browsing modality, it may be helpful to expose effectively the relevant information in such a way that they are immediately perceived by the users. In the other cases, the access to information is achieved through search functionalities. The typical paradigm involves a *query formulation* phase, a *matching* phase, and a *result presentation* phase. Let us focus on the core of this paradigm, i.e. the matching phase: the candidate resources are matched against the query and a measure of their similarity is computed.

In the concept-driven case, the matching is performed on the textual metadata characterizing the resources. In the content-driven case, an elaboration layer is added. The objects are processed and their so-called *signatures* are computed, so that the signatures can abstract the content of the original resources. If I call this computation D and the resources $x_1 \ldots, x_n$, every x_i will be actually represented by its signature $D(x_i)$.

In an ideally simple case, it would be possible to infer some conceptual information directly from $D(x_i)$ and fall back into the concept-driven case. Whenever a underlying conceptualization is present, some ad-hoc rules can be created (a priori or on the fly) in such a way to establish connections between computed characterizations (i.e. extracted by the $D(x_i)$) and higher-level concepts. I will show an example of this in Chapter 6, where we make use of feature descriptors to let the user connect low-level measures with the value of certain attributes of the considered models. For instance, the user will be allowed to rely on a descriptor calculating the *compactness* of an object using the formula $\texttt{volume}^2/\texttt{area}^3$ and connect it with the conceptual attribute `compactness`. This approach is also possible when the conceptualization is fixed and the interpretation rules are hard-coded a priori. [57, 43].

In most of the cases, though, the content-based paradigm is not so straightforward (i.e. resource - signature - explicit conceptual meaning): at least from an historical perspective, its main role was not to provide directly an interpretation suitable for a human user, but to allow for the computation of proximity between different resources (i.e. resource - signature - comparison with other signatures - implicit semantic interpretation).

In other words, we move to the space of the $D(x_i)$, and perform the matching there. In this case, the similarity of two objects x_i, x_j is connected with the computation of a *dissimilarity measure* $d(D(x_i), D(x_j))$ between their respective signatures. The goal is to move to a domain in which the essence of the object is captured (possibly in an implicit and machine-understandable manner), and encoded in such a way to allow comparisons and similarity assessment procedures.

For instance, a descriptor of a 3D object based on the distribution of the surface elements with respect to their distance from the center of mass (as well as a descriptor of an image based on a statistic on the colors of the pixel of the image) may not have a direct cognitive interpretation, but still it is based on a meaningful content of the resources and can be used to compare them. The idea is that from a given description the user may not directly understand that the resource is an orange, but he/she can learn that it is similar to other objects representing oranges, and indirectly draw the conclusion that the resource may be an orange, or at least orange-like.

Note that sometimes the lack of a precise binding to a concept may be a strength rather than a weakness: imagine that the user has a model of a human in a given pose (e.g. standing with a lifted arm), and would like to search for a monster in the same pose. In this case he/she would look for a similarity in the content (the specific content expressing pose) rather than in the concepts representing the objects.

4.2 MPEG-7 towards the role of content

So far, we have stressed the importance of sharing data on the web, and in the specific case of 3D resources this target is tightly connected with providing a thorough, formal, and shareable conceptual description of them. Yet, in this framework resources are solely referred to as sort of “black boxes”: any concept or tag could be possibly attached to the 3D objects, disregarding of what they *actually* represent. The actual situation is not so dramatic, as it is often possible to capture some characteristics of the objects themselves directly by analyzing them, i.e. by exploiting their intrinsic content, as it will be explained later in this Chapter. Recalling what have been said in Chapter 2, content-based descriptions are objective because they are based on precise measurements performed directly on the available content. Note that the concept-based and the content-based approaches are not mutually exclusive: a measurement can be performed through an analysis of the content of a resource and encoded in a formal description. This is the key idea behind MPEG-7, which has been originally designed to handle audio and video information, but still being applicable to 3D data. MPEG-7 supports descriptions related to: still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation (scenarios).

MPEG-7 is defined as a Multimedia Content Description Interface and has been released as a ISO standard in late 2001. As it is possible to evince from the official web page [69, 54], this standard provides a rich set of standardized tools to describe multimedia content. Both human users and automatic systems that process audiovisual information are within the scope of MPEG-7. It offers a comprehensive set of audiovisual Description Tools (the metadata elements and their structure and relationships, that are defined by the standard in the form of Descriptors and Description Schemes) to create descriptions, which will form the basis for applications enabling the needed effective and efficient access (search, filtering and browsing) to multimedia content. It is intended to capture concept-based and content-based information. In fact, the MPEG-7 descriptions may include:

1. Information describing the creation and production processes of the content (e.g. director, title, short feature movie);

2. Information related to the usage of the content (e.g. copyright pointers, usage history, broadcast schedule);

3. Information of the storage features of the content (e.g. storage format, encoding);

4. Information about low level features in the content (e.g. colors, textures, sound timbres, melody description) but also

5. Conceptual information of the reality captured by the content (e.g. objects and events, interactions among objects).

All these descriptions are of course coded in an efficient way for searching, filtering, etc. The first three items mentioned are clearly concept-based, as the actual content of the resource is not addressed. The fourth item is content-based, as it directly addresses the low-level features of the resource. The last item can be directly filled via a manual tagging or via an automatic extraction of higher-level features starting from low-level ones, as it may occur in learning; it is a good example of how concepts and content can intertwine yet cooperate.

The strength of this approach is that it is possible to rely on a shared standard which is very flexible and can be easily coupled with search and retrieval facilities, as it happens with the MPEG 7 Query Format [25]. It includes concept- and content-based information and its descriptive power can be easily enhanced by composing higher-level descriptors starting from low-level ones.

I am discussing MPEG-7 because it shows how the problem has been tackled by the multimedia community and because it is sufficiently extensible also to the domain of 3D objects. Nevertheless, it is *applicable* but not originally intended for 3D objects, and therefore the low-level features on top of which it is possible to build descriptors are quite limited. Some 3D frameworks based on MPEG-7 are currently present [21, 38], and some ad-hoc descriptors have been developed [51].

The attention of the 3D media community, however, is more devoted to unify the 3D object representation formats, because a lot of 3D models are currently available, but are encoded in proprietary formats. [116]. In order to ease the deployment of 3D objects on the web, the X3D format [66] was introduced and is nowadays widely adopted. Also, in order to ease the versatility of the creation of 3D assets across different applications the standard interchange format Collada [5] is emerging. Both X3D and Collada are designed in such a way to ease search and retrieval procedures, but they are not designed purposely for descriptive reasons, as MPEG-7 is.

It is still an open issue whether to reinforce existing 3D standards in such a way to enrich their descriptive functionalities, or to exploit the MPEG-7 framework by extending it with flexible 3D descriptors, or even to think about new descriptive standards specific for the 3D but still compliant to the above formalisms.

4.3 3D object description methods

There exist a number of surveys which take very thoroughly into account a number of diverse 3D object description methods. In particular, I mainly rely on the surveys [102] and [114]. The intent in this section is not to provide another state of the art report; in fact, the overview will be very quick and with the aim of letting the reader understand *what* the descriptors can capture rather than *how* they do it. First of all, any description process is characterized by what is *preserved* and what is *discarded* about the original resource in the process of description. Other interesting features that have to be taken into account while characterizing a description methodology are:

- its *discriminative power*, i.e. the ability to capture properties that discriminate objects well;
- its *robustness*, i.e. the ability to map small changes in the object (noise, cracks,

topological degeneracies) into small changes in the descriptor.

Both of these properties should be considered within the usage context. In the case of the discriminative power, a property that is very discriminative for recognizing humans within a database of animals might be not discriminative at all in a database of mechanical pieces. As to the robustness, it is very important to understand what is meant by a "small change" in the object. Is a little hole a small change in an object? It might be, but it might also be the most important difference between a cup and a funnel.

I want to remark that a rigourous evaluation of 3D object descriptors with respect to the object types and contexts of use is still missing. 3D content-based retrieval is relatively a young field, and benchmarking initiatives started only recently (e.g., SHREC [6], since 2006). During the work of this thesis, we have indeed recognized the great value that a methodological evaluation of the performance of the descriptors might have on the development of future search engine and I will discuss again this issue in the closing Chapter of the thesis, pointing to future development of my thesis work.

4.3.1 Representation format

Without digging for the details of the description methodologies, it is still important to point out that the format to represent the considered object, as well as some properties of the coded representation, do affect the choice of the eligible descriptors. Most of the 3D models found on the World Wide Web are meshes stored in a file format supporting visual appearance. Currently, the most common format used for this purpose is the Virtual Reality Modeling Language (VRML) format [42]. Other formats common for meshes are PLY, OFF or STL. With variations, they simply list vertices and then a list of triangles or polygons.

Since most of these models are mainly designed for visualization, they sometimes contain only geometry and appearance attributes. They are often not even meshes but so-called *polygon soups*, consisting of unorganized sets of polygons. In addition, these models are generally not *watertight* meshes, i.e. they do not enclose a volume. This may cause problems, especially when the connectivity and the topology is relevant.

In many cases, though, it is possible to fix this kind of "poor representation" problems through ad-hoc pre-processing tools able to repair the original 3D models. Among these tools it is possible to mention [8]. In any case, the point is that the representation format and some qualitative characterization of the model (e.g. watertightness) are relevant aspects that need to be taken into account.

4.3.2 Invariance and normalization

Discarding one characteristic (feature, property) means that all the objects which differ *only* with respect to that specific characteristic should be considered equivalent, hence their computed mutual distance should be 0. This is strictly connected with the notion of *invariance* with respect to a transformation.

If C is a characteristic (e.g. orientation in space), let us couple C to a family of transformations $\mathcal{T}_C$ which change *only* the characteristic C (e.g. rotations). If a descriptor D discards the characteristic C of a resource, it means that its interpretation of the resource does not vary under the transformation $\mathcal{T}_C$, and it is possible to say that D is invariant with respect to $\mathcal{T}_C$. In other words, $D(x_i) \equiv D(\mathcal{T}_C(x_i))$.

Invariance can be achieved in two ways: the good case is when the description method is insensitive to the characteristic that we would like to forget; if this is not the case it is often possible to set up a preprocessing phase called *normalization*, in order to lose the tracks of the characteristic that we want to forget. For instance, if we want to forget about the orientation of an object we might use a description method that is insensitive to rotation or instead normalize its orientation in a preprocessing phase (e.g. aligning the axes of the reference system along the principal components of the object). Note that neglecting the orientation is often a good choice, as rotation in most of the cases does not affect the semantic interpretation.

Here are examples of simple normalization steps that can be performed [102].

To normalize a 3D model with respect to the scale, the average distance of the points on its surface to its center of mass should be scaled to a constant. Note that normalizing a 3D model by scaling its bounding box is sensitive to outliers. To normalize with respect to translation the center of mass is translated to the origin. To normalize a 3D model with respect to rotation usually the principal component analysis (PCA) method [87] is applied. It aligns the principal axes to the x-, y-, and z-*axes* of a

Figure 4.1: Rotation invariance is *very often* a good quality, but sometimes it may discard the information which separates a *yes* from a *no*.

canonical coordinate system by an affine transformation based on a set of surface points, e.g. the set of vertices of a 3D model. After translation of the center of mass to the origin, a rotation is applied so that the largest variance of the transformed points is along the *x-axis*. Then, a rotation around the *x-axis* is carried out such that the maximal spread in the *yz-plane* occurs along the *y-axis*. To address robustness issues some variants on classical PCA have been proposed, such as *Continuous PCA* [109].

In my opinion, special attention should be paid when facing any preprocessing phase. In other words, whenever a normalization is performed, some information is discarded once and for all (see Figure 4.1). Therefore, we should clarify beforehand if our intention is really to discard that information. Only in that case, we should apply the normalization process. Frequently, a much more brute-force approach is adopted: if a characteristic C is considered not relevant *in most of the cases*, then the methodologies are created in such a way to have an invariance with respect with the related transformation $\mathcal{T}_C$, which often means to force some normalization steps. This is perfectly fine when an all-round methodology is looked for, i.e. a technique which suits *sufficiently well* most of the cases. The approach that I suggest to take into account is exactly the opposite: the goal is not to find a unique all-round methodology, but different techniques, each of them suiting *very well* some specific cases. I will analyze this inclination to versatility in Chapter 5.

4.3.3 Characteristics to be captured

I will start this section with a well-known little story: [64]

> Once upon a time, there lived six blind men in a village. One day the villagers told them, "Hey, there is an elephant in the village today." They had no idea what an elephant was. They decided, "Even though we would not be able to see it, let us go and feel it anyway." All of them went where the elephant was. Everyone of them touched the elephant.
>
> "Hey, the elephant is a pillar," said the first man who touched his leg.
>
> "Oh, no! it is like a rope," said the second man who touched the tail.
>
> "Oh, no! it is like a thick branch of a tree," said the third man who touched the trunk of the elephant.
>
> "It is like a big hand fan" said the fourth man who touched the ear of the elephant.
>
> "It is like a huge wall," said the fifth man who touched the belly of the elephant.
>
> "It is like a solid pipe," said the sixth man who touched the tusk of the elephant.
>
> They began to argue about the elephant and everyone of them insisted that he was right. It looked like they were getting agitated. A wise man was passing by and he saw this. He stopped and asked them, "What is the matter?" They said, "We cannot agree to what the elephant is like." Each one of them told what he thought the elephant was like. The wise man calmly explained to them, "All of you are right. The reason every one of you is telling it differently because each one of you touched the different part of the elephant. So, actually the elephant has all those features what you all said."

In this section, I will consider 3D object descriptors as sorts of high-level sensors enabled to capture some characteristics of the objects they describe. Much like the story of the blind men and the elephant, the descriptions come out from different perspectives and in general cannot agree with one another (see Figure 4.2). My claim is that the only way to follow is the one of the wise man, i.e. not trying to argue about

Figure 4.2: Six blind men perceive different aspects when asked to describe an elephant

what is the best descriptor, but trying to understand and exploit the power of each of them.

In the following, I will consider different high-level characterizations of the 3D objects that can be captured by different descriptors. Global measures are a sort of snapshot of the object just from one single narrow point of view: they are useful only when that specific measure is very important. They can measure the volume of the object, the area of the surface of the object, its length along the principal components, and so on. For a finer description of a shape it might be useful to capture the spatial distribution of its surface elements (i.e. where the object elements are located), the connection among its parts (i.e. how the object elements are connected, whether holes are present), or some non-rigid information (i.e. the information that is persistent even after bending parts of the object). The analysis could be more fine-grained, yet it shows that description methodologies are designed to respond to high-level needs which are mapped to low-level technical details. My emphasis is on the high-level needs that have to be taken into account.

Descriptors capturing global measures

The most coarse-grained object descriptors are the ones that yield global measures. A global measure is a sort of snapshot of the object, which is considered as a whole just from one single narrow point of view: they can measure the *volume* of the object, the *area* of the surface of the object, its lengths along the *principal components*, its *compactness*, its *sphericity*, and so on.

Some of these descriptors (e.g. volume, area) are the outcome of direct computation, some of them can be obtained as a combination of the previous ones within formulae. Some examples can be $\texttt{volume}^2/\texttt{area}^3$, which measures the compactness of a segment and is invariant to its uniform scaling, or $\texttt{genus} > 0$, which can be connected with the presence of holes.

Just to name a few of them, Zhang and Chen [117] propose methods to compute efficiently global measures such as volume, area, statistical moments. Corney et al. [31] propose methods for calculating convex-hull-based indices to carry out a preliminary coarse filtering of candidates prior to more detailed analysis.

These measures refer precisely to a specific feature and for this they can have a high value in an (automatic) annotation phase: they can be matched with high level attributes and computed automatically. This is the case of the so-called *segment descriptors* in the ShapeAnnotator tool, which I will describe thoroughfully in Chapter 6.

For retrieval purposes, they might be useful only when that specific measure is very discriminative. In general, this is not true, but it might fit very specific targets. For instance, let us suppose that the scenario is a game and a monster is approaching. I want to throw something at the monster. In this specific case, the focus is not on the local features of the object to be thrown: any object with a quite small volume and a high compactness could work fine. In other situations, this kind of descriptions are not used as the main methods, but mainly as a preliminary filter to reduce the number of eligible resources.

Descriptors capturing spatial distribution

The need for capturing spatial information means that we are mostly interested in *how* the object occupies the space.

To give a flavor of the methodologies that are used for this purpose, Osada et al. [83] present the so-called *Shape Distributions*, which measure the distribution of properties based on distance, angle, area, and volume measurements between random surface points. In their experiments, the *D2* shape distribution, which measures distances of random points, is most effective (see Figure 4.3). Vranic et al. [110] introduce a ray-based descriptor that describes a surface by associating to each ray from the origin the distance to the last point of intersection of the ray with the model. For this spherical extent function its spherical harmonics are computed, which form a Fourier basis on a sphere much like the sine and cosine do on a line or a circle. Many variants of this method have been proposed [115, 48]. Another completely different class of methods, but still designed to capture the spatial distribution of object are the *view-based* descriptors. Different techniques are set to describe 3D objects through multiple 2D views of it. For instance Chen et al [28] consider similar two models if they look similar from all the viewing angles. A *lightfield* descriptor is introduced, which compares ten silhouettes of the 3D shape obtained by ten viewing angles distributed evenly on the viewing sphere.

Even if in the practical case one of the mentioned methods may end up outperforming the others, the main point that I want to stress is their common intent: these

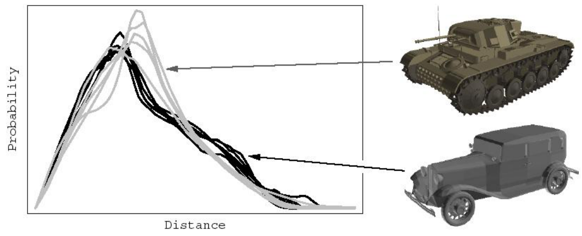

Figure 4.3: D2 shape distributions of five tanks (gray curves) and six cars (black curves) [83]

kind of techniques are designed to capture, and reflect well, the spatial distribution of the considered shape.

This kind of requirement is very proper when the overall shape is actually relevant. A very simple example: if I have to discriminate airplanes from other objects, methods which rely on spatial distribution will be likely to be very good, because airplanes have a typical, rigid shape and are usually well characterized by this.

Examples of objects whose main characteristics can be effectively captured by this kind of descriptors: *airplane*, *screwdriver*.

Examples of objects whose main characteristics are not effectively captured by this kind of descriptors: *human*, *octopus*.

Descriptors capturing connections among parts

The need of capturing information about the connection among parts means that we are mostly interested in the question about *how* the relevant parts of an object are interconnected. As a matter of fact, in some cases an object is better characterized by the interconnection of its constituting parts than by its overall shape.

The interconnection among parts is often recorded in the form of graphs, which encode parts of the objects as vertexes and their connections as edges.

One of the most intuitive ways of capturing the structure of an object is to rely on its decomposition in parts, also known as *segmentation*. In fact, a part decomposition not only provides semantic information about the underlying object, but also can be used to guide several types of mesh processing algorithms, among which skeleton extraction [16]. As I will show in further details in Chapter 6 there are a number of eligible ways to perform a segmentation. Recently, Chen et al. presented a benchmark for evaluation

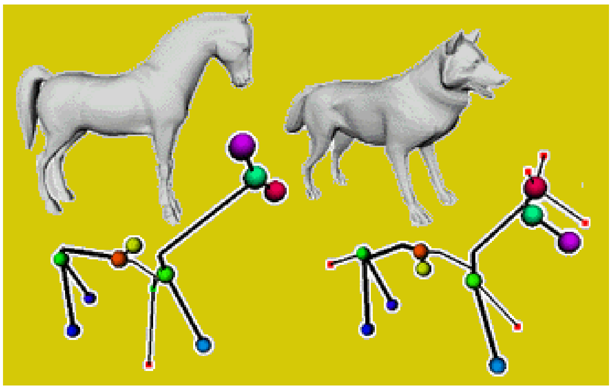

Figure 4.4: Structural descriptors based on Reeb Graphs

of 3D mesh segmentation algorithms [29]. Nevertheless, not all the descriptors enabled to capture the connection among parts rely on segmentations.

Other methods capturing connections among parts may directly derive from skeletonization processes. For instance, Sundar er al. [101] use a skeletal graph that encodes also geometric and topological information.

An important family of part-based descriptors is based on the computation of *Reeb Graphs* 4.4. Given a surface S and a real function $f : S \rightarrow R$ (the so-called *quotient function*), the Reeb graph represents the topology of S through a graph structure whose nodes correspond to the critical points of f, which are located in correspondence of topological changes of S, such as birth, join, split and death of connected components of the surface. [89, 15]

The choice of f determines how the descriptors follows the shape of the object and what are the characteristics, or parts, to highlight in the object. For instance, if f is the *height function*, the descriptors follows the shape of the object as a sort of "scanning" from bottom to top and will capture protrusions and wells in the vertical direction; if f is the *integral geodetic* distance the descriptor scans the object by following its surface, and will capture protrusions, even if bent. Biasotti et al. [18] compare Reeb graphs obtained by using different quotient functions f and highlight how the choice of f determines the final result.

This kind of methodologies is well suited when the objects to describe are well characterized by their parts and their mutual interconnections. A teapot is well characterized by the presence of a handle (which has to be grasped), of a body (which has to hold liquids) and a spout, and by their mutual relations (the handle should be adjacent to the body, the body has to be adjacent to the spout). Note that the connection among parts may not be the only relevant information: the shape of the handle may be important, the body has usually a quite large volume, and the curvature of the spout should ease the pouring task.

Examples of objects whose main characteristics can be effectively captured by this kind of descriptors: *teapot*, *horse*.

Examples of objects whose main characteristics are not effectively captured by this kind of descriptors: *skyscrapers*, *swords*, *balls*.

Descriptors capturing non-rigid information

In some cases the essence of objects is just loosely connected with the location of their parts, often because we are dealing with articulated objects with a lot of degrees of freedom. The Euclidean space itself does not look like a good space where to set up a proper description of this kind of objects. Let us consider a simple yet effective example. We have many 3D models of the *same* virtual human in different position: standing, sitting, jumping, walking, ducking. In the Euclidean space these models are very different, yet they represent the same human. If the aim is to pay attention to the pose of the human, a descriptor preserving the information about the spatial distribution of the object would be proper. But in some other cases the aim is to forget about the pose, and perceive all the mentioned models as *equivalent*.

Recently, several techniques which create descriptors invariant with respect to *isometry* have been created. Two object are isometric (i.e. invariant with respect to isometry) if a homeomorphism from one to the other exists preserving geodesic distances, i.e. mapping curves to curves with equal arc length [91]. This is connected to the invariance with respect to pose. Reuter et al [92] built a methodology to extract what they call the *Shape DNA* from the spectrum of the *Laplace-Beltrami operator*, which is invariant with respect to isometry.

Most of the methods which capture non-rigid information rely on the computation of *geodesic distances* on the surface of the considered objects. Geodesic distances are calculated following the shortest path connecting points on a surface rather than considering their distance in the Euclidean space.

Elad and Kimmel [23, 33] construct bending invariant signatures (see Figure 4.5), by

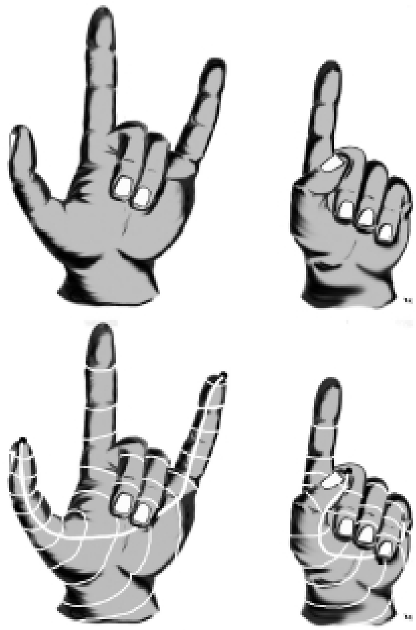

Figure 4.5: Bending invariant signatures are invariant with respect to isometry [33]

creating an embedding of the geometric structure of the surface in a small dimensional Euclidean space where geodesic distances are best approximated by Euclidean ones. Also the Reeb Graphs which rely on integral geodesic distance [45, 14] as the real function are enabled to capture non-rigid information.

This kind of methodologies are more appropriate for non-rigid objects, i.e. objects which are characterized by their structure and their flexibility more than by their rigid shape.

Examples of objects whose main characteristics can be effectively captured by this kind of descriptors: *human*, *rope*.

Examples of objects whose main characteristics are not effectively captured by this kind of descriptors: *airplane*, *fork*

4.4 Discussion

Wrapping up, the main point to stress as a conclusion of this Chapter is that different content-based description methods can be effectively used, and in some cases it is perfectly legitimate to compare their results on shared benchmarks. The approach that I have been proposing in this Chapter yields a coarse characterization, which needs to be further refined and enriched. However, what I think is most important is to provide a characterization of the description methodologies in order to understand which peculiarities of the objects are captured and which are discarded. Following this approach, state-of-the-art descriptors, as well as new ones, will be estimated according to a shared characterization, and a sort of *identity card* of each of them will be provided. There will be no point in using the signature based on the Shape Distributions for detecting humans in different position, simply because their identity card will warn in advance about their extreme weakness for this kind of task. This issue will be further analyzed in Chapter 5.

Besides, in an optimistic perspective, strategies for sharing the outcome of the description processes need to be developed, in such a way to enrich the Open Linked Data with valuable content-based information. The ShapeAnnotator described in Chapter 6 has been designed with this ultimate goal in mind. However, for a wide adoption of this approach, the scientific community should decide how such information has to be coded. MPEG-7 will be the answer or a dedicated standard will be more proper? There is still no answer to this question.

Chapter 5

Context

Seth: What's that like? What's it taste like? Describe it like Hemingway.
Maggie: Well, it tastes like a pear. You don't know what a pear tastes like?
Seth: I don't know what a pear tastes like **to you**.
Maggie: Sweet, juicy, soft on your tongue, grainy like a sugary sand that dissolves in your mouth. How's that?
Seth: It's perfect.
(City of Angels)

As a sort of conclusion from what we learned in the previous chapters, it is possible to claim the following: concept-based descriptions rely on (un)structured conceptualizations to annotate a resource, i.e. they *create* synthetic information about it; content-based descriptions constitute signatures of a resource, i.e. they *extract* analytic information about it. These are two perfectly legitimate ways of providing high-level information about the resource, and in this sense they address the general issue of semantics. Still, it is important to understand that they can never encode the *actual meaning* of the resource, simply because there is no actual meaning *of* a resource (i.e. intrinsic to the resource, innate, independent of the language it may be expressed). Most of the linguistics and philosophy of language of the XX century concluded that there can be no meaning before language and independent of it: meaning can only exist within the categories and the structures of language [46]. This is the basis of a fierce criticism by Santini towards the idea of the Semantic Web itself. In [95] he claims that context is somehow *downplayed* in the most common of the current approaches to semantics, that is the *Semantic Web*. The Semantic Web approach is based on two assumptions regarding signification, namely that the meaning of a document can be represented as a logical theory, and that meaning is a property of the document. He criticizes the wide approach to web data as data that are not so different from the ones in databases: they are looked as carriers of an inherent meaning that can be made sense of by an algorithm. The problem is that these data are expressed in semiotic systems (e.g. natural languages, images, video, 3D objects) that make it difficult to *extract* this meaning. This is the core of the criticism: it is possible to extract meaning only if it is assumed that meaning pre-exists the resource.

A resource (when it is shared) can be intended as a message in a given language towards a plausible audience. Its meaning is a by-product of the signification process, and any sufficiently fine-grained semiotic system cannot make so drastic simplification to underrate the importance of the destination of the message (i.e. the user) in it. The signification process is forcefully an interpretation process, which is inseparable from the context in which it takes place. In other words, it is impossible to set the role of the context aside.

Nevertheless, I share Santini's concern about the limitations of the Semantic Web, but far from considering it the ultimate solution, I still appreciate it as a tool for providing more and more descriptive information. In fact, even the very supporters of the Semantic Web do not conclude that ontologies *represent meaning* (even if sometimes they push in that direction, considering the semantic web approach as a sort of panacea). Nonetheless, especially in the latest years, they push the user communities to share more and more data in structured formats in order to enable automatic harvesting, gathering, analysis and federation of meaningful information. At the same time, while the Semantic Web community asks to assert conceptual information, the content-based information retrieval community tries to pull other pieces of information out of existing data. New descriptors are built, new features are extracted, new information is available. The outcome is the same: more and more descriptive data can be associated to the original resources, sometimes even *too many* descriptive data. But whenever it is possible to speak about information overload, it is also possible to speak about (the lack of) filtering this information.

If you think about having the whole WWW as a single huge resource, what any user wants to do is to get only the important information and filter out the information that is not relevant for his/her in a given context. The same pattern occurs if a plethora of descriptions of the same 3D object is provided: the user wants to focus only on the descriptive feature that are meaningful for him/her in that very context.

After some general considerations about the meaningfulness of descriptive features and some intrinsic problems, such as their mutual separability, I will focus on the actual paths that can be followed for expressing the contexts. In the following, I will mention two complementary approaches: the one carried out by Giorgi et al. [37] which aims at an implicit representation of the context and my personal approach [94], targeted at an explicit representation of the context. My contributions in this chapter are the high-level analysis on the role of contexts in the description process and the approach to an explicit encoding the contexts.

My paper referring to this chapter [94]:

- Robbiano F., Spagnuolo M., Falcidieno B. (2008). "The Role of Contexts and Descriptors for Expressing Semantics in 3D Media". In Proceedings of the SAMT Workshop on Semantic 3D Media, pp. 53-57. Koblenz, Germany, December 3rd, 2008.

5.1 Meaningful features

Now it is the right time to try and give an answer to the questions posed earlier: which features can be considered "meaningful"? And how the meaningful information can be separated by the "useless details"? Regarding the first question, some basic examples can show that the answer, as we can expect, is not unique. Some information can be considered meaningful in a usage domain and useless in another usage domain: it is a matter of the context in which the user relates to the resource. Any context could be seen as a selector which can distinguish the important features from the useless ones and which should ideally act as a sieve to provide a meaningful description of the resource. As for the second question, it depends on the tools enabled to perform analysis and description over the resources: can they discriminate the different features? Can they provide information only regarding the meaningful ones in a given context?

Let us suppose a simplified scenario in which a user is interested in retrieving square objects (e.g. disregarding their size and their color) within a repository in which the objects are described through a simple feature vector expressing: [*shape, size, color*]. In this case the context induces the selection of the feature "shape" and then the ideal descriptor would be a simplified feature vector with just the element *shape* (forgetting information about size and color, not relevant for the given context). But note that this passage is trivial only because the features where clearly separated in the descriptor, the important feature was very clear a priori, and the selection could be performed in a straightforward manner.

When the situation is not so ideal, the most problematic issues are the context-definition step and the determination of the proper descriptor, or combination of descriptors, which fit the requirements of the context. For the first issue it is important to understand how the user can assess his/her viewpoint on the resources: when he/she has a clear a priori idea about the features that can be considered meaningful (as in the simple scenario above), it is likely that he/she would prefer to express the context *explicitly/directly*: "In this context shape is meaningful, size is not meaningful, color is not meaningful". But, if he/she has no way to name the meaningful features a priori, he/she could still be enabled to express the context *implicitly/indirectly*, by grouping resources which share similar characteristics and by separating resources which are different in the given context. In the example above the user could state:"In this context `object1` and `object2` are similar, while `object3` and `object4` are dissimilar". If `object1` is a small, red square and `object2` is a big, blue square, then the implicit information corresponds to the explicit one in the previous case, and this assessment can be further strengthened by the fact that `object3` is a small, white circle and `object4` is a small, white square. Each of these approaches has advantages and drawbacks: the explicit approach is bound to the expressiveness of the known features (if the considered features are just overall shape, size and color they would be not so helpful in the majority of the considered contexts; something more is needed) and to a precise a priori assessment by the user, but has the main advantage of being independent of a specific repository of resources and thus it can be reused easily. The implicit approach can be achieved, for instance, through relevance feedback techniques, but is bound to a specific repository and thus cannot be reused as an extension of the query.

5.2 The role of context in 3D object description

As mentioned earlier, the role of the user in the interaction with digital resources is fundamental. It is possible to assume that any user approaching a 3D object, both in real life and in digital applications, is intrinsically bound to the context in which he/she lives. His/her interpretation of the object can vary a lot depending on the

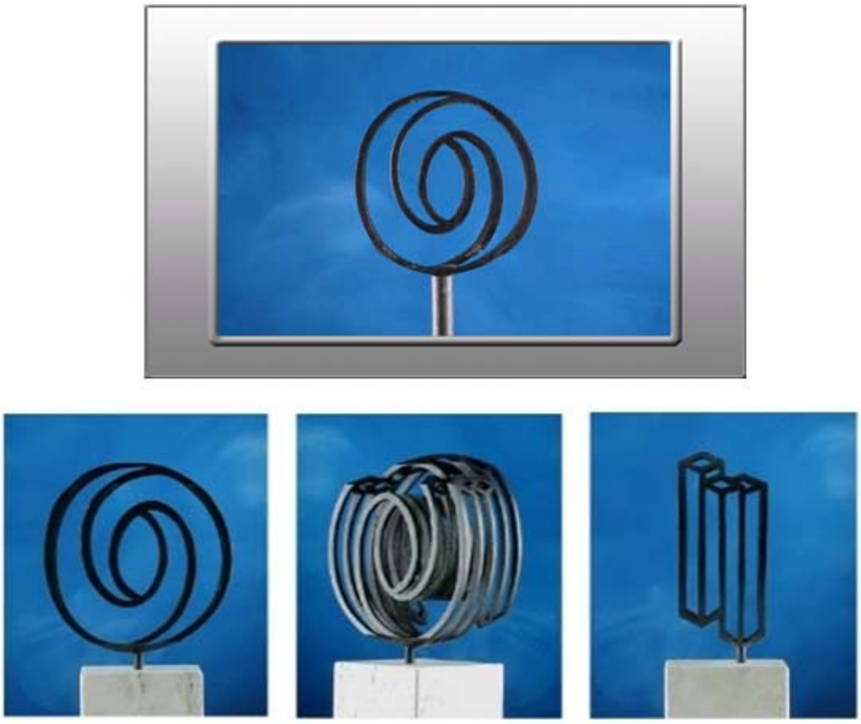

Figure 5.1: Impossible Ring and Pillars by Guido Moretti.

viewpoint, much like the interpretation of the “Impossible Ring and Pillars” by Guido Moretti depends on the perspective it is looked from (see Figure 5.1).

If a user has to fight monsters he/she would examine an object to decide if it can be thrown or used as a weapon; if he/she has to hold liquids, then he/she will care about the presence of holes in the object; finally, if the importance is the resemblance of the object to a given model, then the overall shape would be relevant. Even when the objects and the users are the same, the context may change, and the semantics that we would like to use for enriching the representation of the 3D media would change accordingly. Therefore, it is almost immediate to understand that no single way of encoding semantics is appropriate unless we are bound to a unique and fixed context. This is true also in general. Let us make an example in order to better clarify the point.

> Phone call between Ann and Bob.
> Step 1: Ann: “Tell me something about sage”
> Step 2: Bob: “Where are you?”

Ann might answer that she is at the restaurant, in a greenhouse, in a coffee-shop or in a library. According to her answer, Bob will talk about the taste of sage and its usage in the Italian or French cuisine, the flowers and the specific properties of the Salvia Officinalis, the psychoactive effects of Salvia Divinorum, or the etymology of the word coming from the Latin "salvus" (i.e. "healthy"), respectively.

In our metaphor, Ann is the user, Bob is the computer, and "sage" is the resource. At Step 1 the user asks for a description of the resource. The crucial point is that at Step 2 Bob does not talk about the resource at all. It does not even talk about the user, as he knows perfectly Ann. Nevertheless the question ("where are you?") is fundamental in delivering the most proper description. Context is the key element. Context induces a targeted selection of the view points, and it is useful especially when the possible descriptions of the resources are numerous and diverse.

It is possible to claim that, whenever resources are used in just one application domain, it would be not so useful to address a context-based description of them. Some years ago this could be the case of 3D objects: their importance was high, but often they were intended just within a single usage domain, e.g. product design or game developing. This is also the reason why the role of context-based descriptions has not been felt so important. Nowadays, the situation is changing as users are often interested in exploiting the same resources across different application domains: monuments scanned for cultural heritage reasons may be used for realistic environments in games, humans scanned for simulations may be used as starting points for designing avatars, objects modeled for design reasons may be used in virtual realities, or also the other way around: user-generated content may be taken as inspiration and reused within a new life cycle of product design in a CAE/CAM environment. Moreover, context-dependance can occur also within the same application domain: within a virtual environment a bed can be considered quite similar to a chair when the context requires something to seat on (i.e. the focus is on the main *function* of the chair, which is "something you can seat on"), while in the same environment a bed can be considered very dissimilar to a chair if the context is about design (i.e. the focus is on the *appearance* of the chair).

5.3 How the context can be formalized

The issue of context formalization is very close to the issue of term definition. Let us see in what sense.

5.3.1 Intension and extension of definitions

A rather large and especially useful portion of our active vocabularies is taken up by general terms, words or phrases that stand for whole groups of individual things sharing a common attribute. But there are two distinct ways of thinking about the meaning of any of such terms [50].

The *extension* of a general term is the collection of individual things to which it is correctly applied. The *intension* of a general term, on the other hand, is the set of features which are shared by everything to which it applies. Thus, the extension of the word “chair” includes every chair that is (or ever has been or ever will be) in the world and the intension is (something like) “a piece of furniture designed to be sat upon by one person at a time.”

Clearly, these two kinds of meaning are closely interrelated. We usually suppose that the intension of a concept or term determines its extension, that is, we decide whether or not each newly-encountered piece of furniture belongs to the chairs by seeing whether or not it has the relevant features. Thus, as the intension of a general term increases, by specifying with greater detail those features that a thing must have in order for it to apply, the term extension tends to decrease, since fewer items now qualify for its application.

A *denotative* definition tries to identify the extension of the term in question. Thus, we could provide a denotative definition of the phrase “this logic class” simply by listing all of our names. Since a complete enumeration of the things to which a general term applies would be cumbersome or inconvenient in many cases, we commonly pursue the same goal by listing smaller groups of individuals or by offering a few examples instead. In fact, some philosophers have held that the most primitive denotative definitions in any language involve no more than pointing at a single example to which the term properly applies.

But there seem to be some important terms for which denotative definition is entirely impossible. The phrase “my grandchildren” makes perfect sense, for example, but since it presently has no extension, there is no way to indicate its membership by enumeration, example, or ostension. In order to define terms of this sort, and in order to define more conveniently general terms of every variety, we naturally rely upon connotative definitions.

A *connotative* definition tries to identify the intension of a term by providing a synonymous linguistic expression or an operational procedure for determining the applicability of the term. Of course, it is not always easy to come up with an alternative word or phrase that has exactly the same meaning, or to specify a concrete test for applicability. But when it does work, connotative definition provides an adequate means for securing the meaning of a term.

I previously claimed that the context should be modeled as an ideal sieve enabled to tell the relevant features of a resource from the irrelevant ones. Analogously to the case of term definition, it is possible to follow two main paths: an *explicit/denotative* way and an *implicit/connotative* way, sharing the pros and the cons mentioned above. The only difference being that what needs to be modeled is not the membership assessment to a given set (e.g. "is this object a chair?"), but the mutual similarity among objects within a data set.

Recalling what was said in Chapter 4, in most of the cases the aim of descriptions is not directly to provide an interpretation suitable for a human user, but to allow for the computation of proximity between different resources through an estimation of the mutual similarity of their signatures. The context will cast a specific view point and will influence the assessment of such similarity: in the example of the colored squares and circles, referring to the specific case in which shape was relevant and size and color were irrelevant, the context *induces* the similarity measure to consider a big red square *very similar* to a small white square, and a big red circle *not similar* to a big red square. Other contexts would cause different assessments.

An *implicit* definition of context has the major advantage of not relying on any a priori conceptualization: the user that wants to express the context does not need to know anything about shape, color and size, because he/she would directly act on the similarity assessment of the individuals. Nevertheless he/she is bound to a closed data set, and as the data set changes a lot of undetermined relations arise. Obviously, if the system is flexible enough, the assessment done on few samples could be enough to let the system behave "by example": the system could be designed in such a way to be smart enough to learn just from few assessments. However, the available data set could be not rich enough to express fruitfully the required context, and in this case the nature of the problem would be intrinsic.

Nevertheless, an *explicit* definition of context would require an a priori conceptualization of the features that may be considered meaningful. The user would be requested (directly or through a hidden interface) to express his/her high-level requirements, for instance explicitly stating that shape is relevant, color is irrelevant, size is irrelevant. This requirement of an a priori conceptualization may be a burden, but would be rewarded with a full reusability of the context: even in a completely different data set, if the focus of the user is the same, he/she can still reuse the formalized context. Moreover, it can be applicable even to an empty or unknown data set, as it does not rely at all on the knowledge about the individuals.

What is common to both approaches is the need of exploiting the descriptive power of a variety of *high level shape descriptors* and comparison methodologies, which are able to synthesize a variety of high-level perceptual shape properties, so as to approximate the perceptual variety of humans. Then, we have to develop smart techniques for *adapting, selecting and combining* the descriptors and the similarity they induce, so that they fit the subjective ideas of the observer. This requires "including a human in the loop", that is, to make the user an active player in the search process [32].

5.3.2 An implicit expression of context

The Multilevel relevance feedback technique described in [37] is a very good example of implicit expression of the context. It aims at the interactive approximation of a pseudodistance δ on a dataset

$$\Sigma = \{x_1, \ldots, x_N\}$$

based on the user's feedback. The pseudodistance δ represents the dissimilarities between the objects of Σ with respect to the subjective judgement of the user. A pseudodistance δ is a metrics without the property assuring that $\delta(x_i, x_j) = 0$ implies $x_i = x_j$ (in other words, two objects can have vanishing pseudodistance without coinciding).

Figure 5.2: A continuous scale for relevance assessment

The system can rely on a family of 3D shape descriptors, producing a family

$$\mathcal{G} = \{d_1, \ldots, d_n\}$$

of pseudodistances between the objects in the database Σ. The way humans perceive and recognize things suggests that the recognition of any object requires a plurality of different recognitions, according to different object properties. As discussed above, in the context of 3D shape searching the same concept turns into the need of describing and comparing 3D objects according to different shape properties, i.e. according to different descriptors and comparison methodologies.

Once a query is submitted, the database objects are sorted in order of decreasing similarity to the query. The system then returns a first list of answers. The method proceeds by asking the user to give a feedback about the relevance of some answers through a numerical scale [32, 112] expressing the value of the pseudodistance δ on the pair of objects. δ is the (unknown) pseudodistance the user refers to for comparing the objects in the dataset Σ, and is the target of the approximation. This knowledge, expressed through the user feedback, is used to inhibit the role of the pseudodistances in the family $\mathcal{G}$ that are not compatible with the user's judgement, meaning that they perceive as different those objects that the user considers as similar. Note that the dataset Σ is used just as a form of "training": after the user teaches to the system his/her own requirements about the context by assessing similarities within Σ, the system is enabled to extend what it learned to different and possibly wider scopes.

This technique relies on a continuous numerical scale for relevance judgements, instead of the traditional binary classification which enables the user to consider object either "relevant" or "not relevant" (see Figure 5.2).

5.3.3 An explicit expression of context

Different layers of elaboration give the freedom of dealing with the object in different contexts, and with this approach it is possible to exploit the expressive power of several descriptions on top of the same object, considering their characterizations explicitly.

As we could observe in the previous Chapters, different signatures computed over a 3D object can take into account global or local measurements, statistics for these measurements throughout the whole object (feature distribution), graphs representing their structure, 2D projections recording the view of the objects from different directions, and so on. Each signature follows a precise description scheme, designed to capture specific characteristics from a 3D object. Different description schemes cast different viewpoints on the objects, being *sensitive* to some characteristics and *invariant* to others. Therefore, it is impossible to state that one descriptor scheme is better than another, unless a context is provided. Thus, far from considering the different techniques as competitors in a race, it could be possible to exploit the distinctive feature of each of them to use only the one(s) which are most suitable in a given situation. The target is to match contexts with the proper description schemes, and the approach involving an explicit expression of the context avoids to overlook the knowledge about how the description schemes actually behave. This approach can be regarded to as a "white box" approach, because the characteristics of the descriptors are used a priori to be explicitly matched against the characteristics of the contexts. The following is a preliminary description of the setup of a system that could support the exploitation of multiple descriptions and the selection of the most suitable in a given context. This system may also serve as the basis for the query formulation and the matching phases in a search engine for 3D objects. The system conceived follows the explicit approach, and its goals are to characterize the descriptors, to characterize the contexts, and to provide the information necessary to match them. These characterizations have to be encoded in formalized conceptualizations, in such a way that they can be matched against the contexts. The conceptualizations of the descriptor schemes have to keep information about which characteristics are preserved and which are discarded. Accordingly, the conceptualizations of the contexts have to encode their desiderata about the characteristics that are important in those contexts and the ones that are not. Some examples of these characteristics (see also Chapter 4 for insight)

Characteristics	Length	Overall shape	Structure	Pose	Orientation
Contexts					
Short and long pencils	relevant	not relevant	not relevant	not relevant	not relevant
Detection of humans	not relevant	not relevant	relevant	not relevant	relevant
Resemblance of generic objects	not relevant	relevant	not relevant	relevant	not relevant
Descriptors					
Length of 1st Principal Component	preserved	discarded	discarded	discarded	discarded
Reeb Graph – Geodesic function	discarded	discarded	preserved	discarded	discarded
Shape Distribution	discarded	preserved	discarded	preserved	discarded

Figure 5.3: An example of matching between contexts and descriptors, considering some relevant high-level characteristics. Descriptors that match very well some contexts may be highly inadequate in other contexts.

are: overall shape, structure, pose, orientation, volume, presence of holes.

When descriptors and contexts are expressed in the same way, as shown in Figure 5.3, it will be simpler to match them and select the most suitable description scheme, or even to combine a number of them to fulfil the requirements of a single context. Clearly, this approach can be fruitfully combined with the a posteriori approach, as the actual performance of the description methods has to be tested and evaluated over shared benchmarks [6].

The actual implementation of this framework goes far beyond the scope of this thesis as several research and technological issues need to be solved before implementing this advanced modality of interaction with 3D objects in an object retrieval pipeline. In addition, the handling of multiple and heterogeneous indices needs to be solved and efficient solutions to this problem found. Last but not least, thorough experiments and analysis on the descriptions methods and their outcomes need to be done in order to have a clear characterization of them, both from the perspective of cognitive processes and computational effectiveness. The innovation and potential of this approach, however, are perceived in a variety of application field and this direction of research will be further explored as a future work (see Chapter 7).

Chapter 6

The ShapeAnnotator

E non m'annoto
e no che non m'annoto e non m'annoto
io no che non m'annoto e non m'annoto no che non m'annoto no che non m'annoto.
(Jov'annoti)

In this Chapter I will describe the software system called the *ShapeAnnotator* that has been implemented in the framework of my PhD activities. The ShapeAnnotator allows to segment and annotate 3D objects and demonstrates how the issues discussed in the previous chapters may contribute to the development of innovative systems for building and managing 3D objects in a semantics-oriented fashion. The use of the ShapeAnnotator will be discussed in two scenarios of application: virtual character design and reverse engineering. In the first scenario, we will make use of an ontology that specifies the structuring of a human body into its main component, while in the second scenario we will show how a feature ontology can be used to attach information about relevant manufacturing parts to an object represented as a plain surface mesh.

My papers referring to this chapter ([93, 10, 9, 27, 11]):

- Attene,M., Robbiano,F., Spagnuolo,M., Falcidieno,B. *Part-based Annotation of Virtual 3D Shapes.* In Proceedings of Cyberworlds '07, spec. sess. on NASAGEM workshop, Hannover, Germany, IEEE Computer Society Press, pp. 427-436, 2007.

- Attene M., Robbiano F., Spagnuolo M. and Falcidieno B., *Semantic Annotation of 3D Surface Meshes based on Feature Characterization*, In Proceedings of

SAMT, Genova, Italy, Lecture Notes in Computer Science Vol.4816, Springer, pp. 126-139, 2007

- Attene M., Robbiano F., Patanè G., Mortara M., Spagnuolo M., Falcidieno B. (2007). *Semantic Annotation of Digital 3D Objects.* In Posters and Demos Proceedins of The second International Conference on Semantic And digital Media Technologies (SAMT 2007), Genova, Italy, December 5th-7th, 2007, pp. 25-26

- Catalano C.E., Falcidieno B., Attene M., Robbiano F., Spagnuolo M. (2008). *Shape knowledge annotation for virtual product sharing and reuse.* In Proceedings of ASME Conference on Engineering Systems Design and Analysis. Haifa, Israel, 2008

- Attene M., Robbiano F., Spagnuolo M., Falcidieno B. *Characterization of 3D Shape Parts for Semantic Annotation.* Journal of Computer Aided Design, Vol. 41, No. 10, pp. 756-763, 2009.

6.1 Goals and Functionalities

Key to an effective sharing of 3D media is the possibility to annotate 3D models with information about the meaning carried by them. Existing 3D repositories, however, are based on metadata that describe the object as a whole, possibly indicating the membership to some semantic class (e.g. vehicles, aircraft, humans) that is still manually attached to objects. Note that, since this association is typically done manually, a finer classification would be very demanding. Nonetheless, the ever-growing size of 3D repositories is making the retrieval hard even in a single bounded category. Thus, characterizing shapes of a given domain is becoming more and more important, and specific annotation approaches that require minimal human intervention must be devised.

Moreover, to the extent of our knowledge, there is currently no repository offering the possibility to browse or search model collections according to annotation attached to object *parts*.

At the same time, the problem of automatic segmentation of 3D objects is still an open research challenge, and the solutions offered by the computer graphics community are decoupled from any actual semantic contexts, providing segmentations that are in most of the cases meaningful only in a geometric sense.

The first attempt of adopting knowledge technologies in the 3D shape modeling was pursued by the AIM@SHAPE project [79]. In AIM@SHAPE, 3D content is organized by making a distinction between knowledge pertaining to a purely geometric level (e.g. type of representation, number of vertices, genus), to a structural description of the objects shape(e.g. skeletons, segmentations), and finally to a semantic annotation of objects and objects parts. The structural subdivision of an object into subparts, or *segments*, has proven to be a key issue to devise effective shape annotations as it provides a way to identify relevant parts of an object geometry in an automatic manner. Also at a cognitive level, in fact, the comprehension of an object is often achieved by understanding its subparts [55, 20]. For instance, if we consider an object which has a *human face* it is likely that the object is a *human being* (the presence of a subpart influences the interpretation of the whole object), and if we want to retrieve a human which is *strong* we can search for humans whose arms *volume* is big (here the quantitative characterization of a part influences the qualitative interpretation of the whole).

In the context of the AIM@SHAPE programme of activities, the ShapeAnnotator had the key role of a "proof of concept" demonstrator of the possibility to couple semantics to 3D objects: to our knowledge extent, the ShapeAnnotator is the first system proposed in the literature to support feature-based annotation of 3D shapes. The ShapeAnnotator is the result of a joint work carried out with colleagues at IMATI-CNR and within an international research context well appreciated by the European Commission.

The claims discussed in the previous Chapters and realized by the functionalities of the ShapeAnnotator are summarized by the following statements. First of all, semantics is the discipline studying the relations between signs (expressed by some syntax) and meaning. This triggers two main requirements: the first is the need of a proper syntax determining the signs, the second is the need to have some means to express the meaning, e.g. through the help of a conceptualization. Another important issue that has been raised is about context: the user should be enabled to express explicitly his/her own view point over the resources, in order to let a specific context cast an interpretation layer. The last remark that I want to recall is about content: it is valuable, when possible, to exploit the intrinsic content of the resources, that can be extracted through proper analytical processes.

The general framework of the ShapeAnnotator does take into consideration these requirements. The syntactic atoms (e.g. the "words" of the considered syntax) derive from a generic 3D segmentation process, which is intended to be adapted to the user's perception of the object to be annotated. The connection to a meaning is obtained by binding the object parts to concepts in a given ontology. The choice of the ontology to be used expresses the freedom of the user to contextualize the knowledge to represent. Moreover, the automatic computation of feature descriptors enables the system to encode some relevant content-driven information and delegate some annotation effort to an automatic process.

6.2 Related Work

The two main tasks addressed by our work are related to 3D feature extraction and indexing, which are supported by *segmentation* and *annotation* respectively.

A lot of research that deals with the integration of these two aspects is related to traditional visual media, images and videos in particular. In the last years, indeed, research in multimedia and knowledge technologies demonstrated the potential of semantic annotations to support advanced sharing and interactions with multimedia, for example for search, reuse and repurposing of multimedia resources. The power of annotation relies in the possibility to create correspondences between resources, or parts of them, and conceptual tags: once the media and/or their parts are annotated, they can easily match textual searches. Stated differently, advanced and semantic-based annotation mechanisms support content-based retrieval within the framework of standard textual search engines.

A variety of techniques have been developed in image processing for content analysis and segmentation, and the available annotation techniques are mainly based on the computation of low-level features (e.g. texture, color, edges) that are used to detect the so-called regions-of-interest [44, 113, 85]. For video sequences, keyframe detection and temporal segmentation are widely used [24].

Although the general framework of content analysis and annotation developed for images or videos may be adapted to 3D shapes, the two domains are substantially different. For images and videos, indeed, the objective is to identify relevant objects in a scene, with the inherent complexity derived by occlusions and intersections that may occur. In the case of 3D, information about the object and its features is complete, and the level of annotation can be therefore more accurate. For this reason, geometric measures, such as bounding box length or best-fitting sphere radius of relevant parts, are not biased by perspective, occlusions or flattening effects.

In the following, I will briefly review relevant work in the area of annotation and segmentation, pointing out the issues that are more relevant to the features of the ShapeAnnotator. It is also worth to mention the work presented in [21] as an interesting example of how semantic annotations could be embedded in an MPEG-7 framework. The ShapeAnnotator nicely complements the work in [21] by providing the tools to actually interact with the geometric representations of object and realize the annotation in practice.

In chapter 3 I already mentioned the dualism between ontology and folksonomy, which ends up in a trade-off between flexibility and meaningfulness. In the first case users are tied to a precise conceptualisation, while in the second case they are users are free to tag the considered resources with any keyword they can think of. The trade-off is between flexibility and meaningfulness. In fact, in the case of free keyword annotation users are not forced to follow any formalized scheme, but the provided tags have a meaning just for themselves: since no shared conceptualisation is taken into account, the association of the tag to a precise semantic interpretation can be only accidentally achieved. Well-known examples, that were also discussed before, of this kind of annotation for 2D images are *FLICKR* [63] and *delicious* [61]. In M-*OntoMat-Annotizer* [86] the ontology-driven approach is chosen, the user is allowed to highlight segments (i.e. regions) of an image and to browse specific domain ontologies in order to annotate parts of the former with specific instances of the latter. Similarly, *Photostuff* [71] provides users the ability to annotate regions of images with respect to an ontology and publish the automatically generated metadata to the Web.

The ShapeAnnotator approach falls in the class of the ontology-driven annotation approaches. Specifically, we tackled the problem of annotating shapes belonging to a specific category which is described by an ontology (e.g. human bodies, cars, pieces of furniture). Each of these ontologies should conceptualize shape features characterizing the category, their attributes and their relations. In a *human body*, for example, *head*, *arms* and *legs* are relevant concepts, relations such as *arm* `is_a` *limb* hold, and attributes such as the *size* of the *head* are helpful to infer higher-level semantics (e.g. ethnic group, gender, age-range).

Given an ontology that describes a specific class of shapes, the optimal solution for annotating 3D models would be to use a shape segmentation algorithm able to automatically detect all the features conceptualized by the ontology. This approach is far from being feasible, as existing segmentation algorithms hardly target semantic features and usually follow a pure geometric approach. Recent surveys of these methods can be found [97], while a comparison of the segmentation results is addressed in [13].

Much of the works tackling the segmentation problem with the objective of *understanding* a shape [7] are inspired by studies on human perception, which loosely couples semantics to geometry. For example there are theories that received a large consensus [20, 55] and that indicate how shapes are recognized and mentally coded in terms of relevant parts and their spatial configuration, or structure.

In another large class of methods, the focus is mainly on the detection of geometrically well-defined features. These segmentations do not produce natural features but patches useful for tasks that require different and possibly non-intuitive schemes (e.g., approximation, remeshing, parameterization) [12, 30, 118]. Generally speaking, this approach is feasible when the features have some formal structure that can be associated to a mathematical formulation. In natural domains, for example human body models, there is no clue on how to define relevant features, and only few methods in the literature tackled a semantics-oriented segmentation in these kind of domains [57].

6.3 The ShapeAnnotator tool

The ShapeAnnotator is a flexible and modular system for part-based annotation of 3D objects. In our settings, 3D shapes are represented by surface meshes while annotation domains are formalized by ontologies: these are mainly implementation choices, while the whole framework has a larger applicability and is independent of the specific representation used both for geometry and knowledge.

The novelty of the *ShapeAnnotator* relies on the concurrent use of a variety of shape segmentation tools to offer a rich set of operators by which the user can interact with the objects shape and easily select the parts he/she wishes to link to relevant concepts expressed by the ontology. The ShapeAnnotator acts therefore at two levels:

it helps the user in the identification of relevant parts, or features, in the model, and it provides the software environment to annotate the parts with concepts that express their semantics. Moreover, since the formalization of a concept may also involve the specification of metric parameters of the part (e.g. dimensions, length), the annotation step implements also a number of automatic services for the computation of these quantitative properties.

To summarize, in this section I will describe how the ShapeAnnotator makes use of the following solutions:

- A *multi-segmentation* framework to specify complex and heterogeneous surface segments (the *features*);
- A set of operations called *segmentmeters* to calculate geometric and topological characterizations of the segments;
- An ontology module which allows the user to browse the domain ontology and to create instances *describing* the features;
- A mechanism to *teach* the system how instance properties can be computed automatically based on segmentmeters.

With regard to implementation details, the ShapeAnnotator is written in C++ and provides an interactive GUI for the inspection, segmentation and semantic annotation of 3D surface meshes. Segmentation tools have been added as plugins to the system, thus allowing for flexibility in the choice of the most appropriate segmentation tools. The conceptualization of the expertise domain is loaded as an OWL ontology, and each feature identified by the user can be annotated using concepts coded by ontology classes. The set of instances resulting from this process are saved as an OWL file. The software provides a set of tools to create and edit the surface features, and it features also a fully functional ontology browser to navigate the ontology and select the desired class to be instantiated to annotate each feature. In September 2007 the ShapeAnnotator has been released as a GPL Open Source software project and hosted by sourceforge.net (http://shapeannotator.sourceforge.net/).

Finally, ontology editing was done using the Protégé resource, which is supported by grant LM007885 from the United States National Library of Medicine.

6.3.1 Multi-segmentation and Part-based Annotation

In the case of 3D shapes, the identification of relevant features is substantially different from the corresponding 2D case. For 2D images, segmentation algorithms are not always considered critical to define features for annotation; on a flat image, in fact, useful features may be even sketched by hand [86]. In contrast, a 3D shape may be very complex and drawing the boundary of a feature might become a rather time-consuming task, involving not only the drawing stage, but also rotating the scene, translating and zooming in and out to show the portions of the surface to draw on. Moreover, while on a 2D image a closed curve defines an inner area, in 3D this is not always true. Hence, for the 3D case, using segmentation algorithms to support feature detection is considered a mandatory step.

Nevertheless, the huge amount of different and specialized works on mesh segmentation indicates that satisfactory results are missing. The majority of the methods used in Computer Graphics are not devised for detecting specific features within a specific context, as for example is the case of form-feature recognition in product modeling and manufacturing. The shape classes handled in the generic segmentation contexts are broadly varying: from virtual humans to scanned artefacts, from highly complex free-form shapes to very smooth and feature-less objects. Moreover, it is not easy to formally define the meaningful features of complex shapes in a non-engineering context and therefore the capability of segmentation methods to detect those features can only be assessed in a qualitative manner [13].

Hence, due to intrinsic limitations, no single algorithm can be used to provide rich segmentations, even within a single domain. This motivates the introduction in the ShapeAnnotator of a theoretical framework for working with multi-segmentations, that allow for a much more flexible support for semantic segmentation. The intuition behind multi-segmentation is that a meaningful shape segmentation is obtained by using in parallel a set of segmentation algorithms and by selecting and refining the detected segments.

This approach realizes part of the discussion presented in Chapter 4: no single descriptor can be used to characterize an object in all domains and for all purposes. Therefore, the strategy that we think is at the basis of true innovation in this field is to move towards a toolbox of segmentation algorithms that give the users the possibility

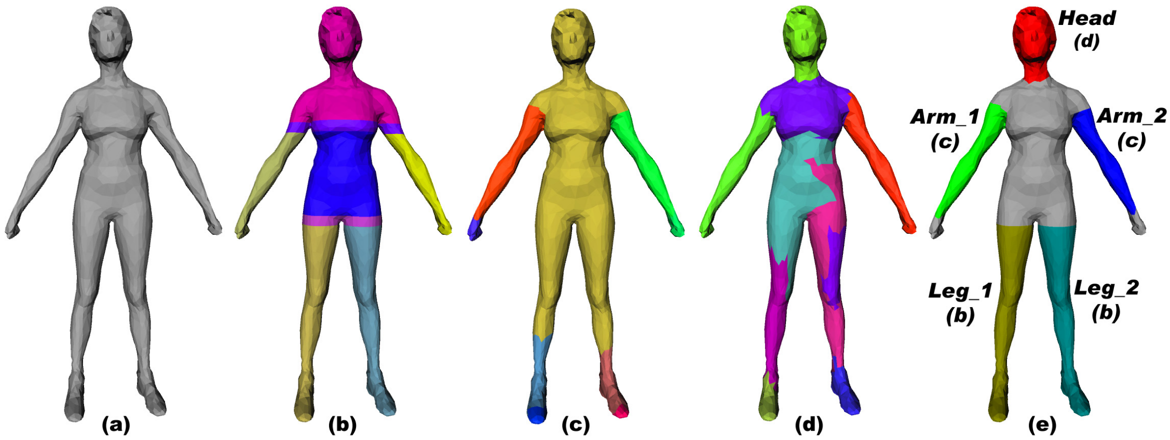

Figure 6.1: An original mesh (a) has been partitioned using different segmentation algorithms: [17] in (b), [58] in (c) and [12] in (d). Only the most relevant features taken from (b), (c) and (d) have been selected and annotated in (e).

to drive the segmentation towards his/her specific segmentation requirements.

Most segmentation algorithms proposed in the literature [13] strive to subdivide the surface into non-overlapping patches forming an exhaustive partitioning of the whole model. Our proposition is that even this assumption is too restrictive: following the claim that the segmentation has to reflect the cognitive attitude of the user, the detected parts do not necessarily have to constitute a partition of the model, as some segments may overlap, and some surface parts may not belong to any significative segment at all.

In addition, also curves and points can be perceived as meaningful parts of an object: sharp edges in a CAD model, the roof line in a car design model, or the tip of the nose in a virtual human carry a relevant meaning per se, and should be treated as independent lower-dimensional features.

Therefore, it is often possible to design a proper technique for the identification of a particular class of features [58, 12] and, if there is the need to identify features of different classes, it is possible to use different segmentation algorithms and take the features from all of their results. In some cases, moreover, there is an intrinsic *fuzziness* in the definition of the boundaries of a feature (i.e., in a human body model the neck may be considered part of both the head and the torso). This is another reason to avoid the restriction of using a sharp partitioning of the whole to identify all the relevant segments.

Due to these observations, we introduce the concept of *multi-segmentation* of a 3D surface represented by a triangle mesh, and say that in a multi-segmented mesh, the results of several segmentation approaches may overlap (see Figure 6.1(b) -(d)).

When a multi-segmented mesh is interpreted within a specific context, some of the segments can be considered particularly *meaningful*. Such meaningful segments (i.e. the features) can be annotated by specific conceptual tags describing their meaning within the context. We refer to an *annotated mesh* as to a multi-segmented mesh in which some of the segments have been annotated (see Figure 6.1 (e)).

Having established *what* do we mean by an annotated mesh, it remains to explain *how* to produce it out of an existing triangle mesh. In principle, an expert in a particular domain should be able to identify significant features and to assign them a specific meaning. As an example, an engineer should be able to look at a surface mesh representing an engine and identify which parts have a specific mechanical functionality. Unfortunately, to the best of our knowledge, today there is no practical way to transform such expertise into usable content to be coupled with the plain geometric information.

To bridge this gap, we defined an annotation pipeline and developed a prototype graphical tool to support it. This tool has been specifically designed to assist an expert user in the task of annotating a surface mesh with semantics belonging to a domain of expertise.

6.3.2 Feature identification

After loading a model and a domain ontology, the first step of the annotation pipeline is the feature identification, i.e. the execution of segmentation algorithms to build the multi-segmented mesh. Once done, interesting features can be interactively selected from the resulting multi-segmented mesh. Each interesting feature can then be annotated by creating an instance of a concept described in the ontology. Optionally, the system may be also programmed to automatically compute attributes and relations among the instances to significantly enrich the resulting knowledge base.

Hereafter a $feature$ is a part of the shape that is worth an annotation, and can be either a connected component, a part of a connected component, a set of connected components or even the whole input mesh.

In order to identify surface features, the ShapeAnnotator provides a set of mesh segmentation algorithms. Our prototype has a plugin-based architecture so that it is possible to import proper segmentation algorithms according to the requirements of the specific class of features. In the current implementation, we have chosen a number of algorithms that cover quite a wide range of feature types. In particular, it is possible to capture:

- *Planar features* through a clustering based on a variational shape approximation via best-fitting planes [30];
- *Generalized tubular features* with arbitrary section computed by the *Plumber* algorithm introduced in [58];
- *Primitive shapes* such as planes, spheres and cylinders through a hierarchical segmentation based on fitting primitives [12];
- *Protrusions* extracted through shape decomposition based on Morse analysis using the height function, the distance from the barycenter and the integral geodesics [17].

The selection of these segmentation tools was motivated, at some extent, by the complementarity of the properties that the tools are able to detect: even if it is just a preliminary example, this is in line with what has been discussed in Chapter 4.

The first step is to use some of these tools and capture, at least roughly, some features of the object. Then it is possible to refine the selected segments through morphological operators. Up to now we set up some operators which act on a segment, determining the growth (shrinkage) of it by adding (removing) a strip of triangles to (from) its boundary. But, since the above operators act blindly only on the geometry of the segment, we devised a set of more meaningful and useful operators, that take into account the presence of other influencing nearby segments. These operators make it possible for instance to:

- Merge two surface segments.
- Grow or shrink a surface segment until its boundary is shared with another surface segment, or coincides with a curve or reaches a point

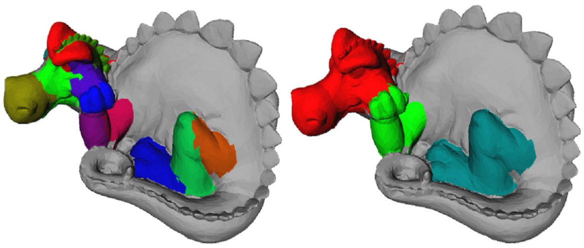

Figure 6.2: Definition of non-trivial features starting from a raw segmentation. On the left, the fitting primitive algorithm [12] could not capture the features properly. On the right the features computed have been edited to obtain a more useful segmentation.

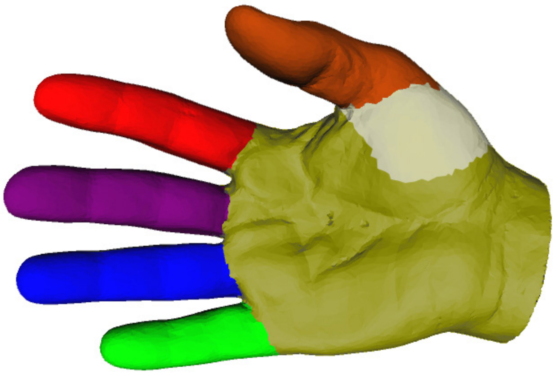

Figure 6.3: Example of segmentation of a natural shape. The bright color on the hand surface indicates that the corresponding part is an overlap of segments (in this case the thumb and the palm). Model courtesy of AIM@SHAPE Shape Repository.

- Make a curve segment from the boundary of a surface segment
- Make a point segment from the intersection of two curves
- Split a surface segment in two using a curve as the new common boundary

It is also possible to remove a segment or to add a new one from scratch (i.e., a single triangle), and edit it through the above operators.

These operations make it possible to refine raw segmentations and properly define useful non-trivial features within a few mouse clicks, as shown in Figure 6.2. Further examples of segmentations are shown in Figures 6.3 and 6.4.

In the current implementation of the ShapeAnnotator and its plugins, it is assumed

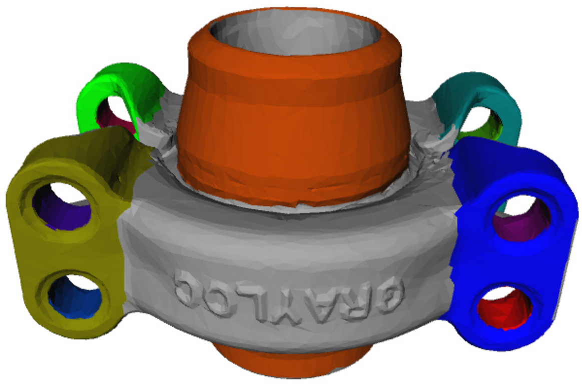

Figure 6.4: Example of segmentation of an artificial shape. Model courtesy of AIM@SHAPE Shape Repository.

that the mesh representing the object is a manifold surface, that is, a topologically well-defined surface.

Some of the existing modelers, however, allow to create shapes with much less quality, for instance polygon soups with components which self-intersect. Also, other modelers offer the possibility to create "concise" or "iconic" meshes by using non-manifold configurations of the faces; it is possible, for example, to replace the wings of an aircraft (which are solid, though relatively thin) with sheets of triangles with zero-thickness. In this kind of models, the decomposition in manifold components often corresponds to the identification of regions which are actually meaningful; in the example of the aircraft, such a decomposition would lead to one fuselage and two wings.

To handle properly these situations, during the model loading stage, the ShapeAnnotator creates a "default" segmentation by grouping together triangles forming connected components. If the triangles of one of such components do not constitute a combinatorial manifold, the loader automatically runs a conversion algorithm [40] that properly duplicates singular elements so that the component is replaced by a set of connected combinatorial manifolds with the same geometry. Eventually, each of these connected manifolds becomes a single segment of the default segmentation.

6.3.3 Manual Annotation

To annotate the features, the user may select proper conceptual tags within a domain of expertise formalized as an OWL [80] ontology. Strictly speaking, for the current functionalities of the ShapeAnnotator, a simpler language could be sufficient, as long as the user is prompted with the chance of selecting among relevant concepts; the choice of OWL, however, has been driven by the potential evolution of the ShapeAnnotator, which is foreseen to become more *intelligent* in the sense of providing inference functionalities (see Section 6.5), and by the fact that OWL is supported by popular ontology editors [81].

Non trivial ontologies may be huge [76], and effective browsing facilities are fundamental to reduce the time spent to seek the proper concept to instantiate. In our approach, the ontology is depicted as a graph in which nodes are classes and arcs are relations between them (see Figure 6.5, middle).

Browsing the ontology consists of moving along paths in the graph, which means jumping from a concept to another across relations. The navigation may be customized by the user and, while the simplest way of browsing is across relations of type `subClassOf` or `superClassOf`, it is possible to select any combination of properties that will be shown by the browser (see Figure 6.5, right top). Once a proper concept has been identified, the ShapeAnnotator provides the possibility to create an instance, which means providing a URI (Universal Resource Identifier, that is, a unique name) and setting the value of the properties (attributes and relations) defined in the ontology for the class being instantiated (see Figure 6.5, right bottom).

Each instance may be further modified in order to make it possible to assert relations between instances of the knowledge base (i.e., `myHead isAdjacentTo myNeck`).

6.3.4 Automatic Annotation

Currently, our system requires the user to manually select the concepts to instantiate; for attributes and relations between instances, however, there is the possibility to *tell* the ShapeAnnotator how these properties can be calculated without the user intervention. The ShapeAnnotator, in fact, comes with a set of functionalities to measure geometric aspects of shape parts (e.g. bounding box length, radius of best-fitting cylinder) and to establish topological relations among the parts (e.g. adjacency,

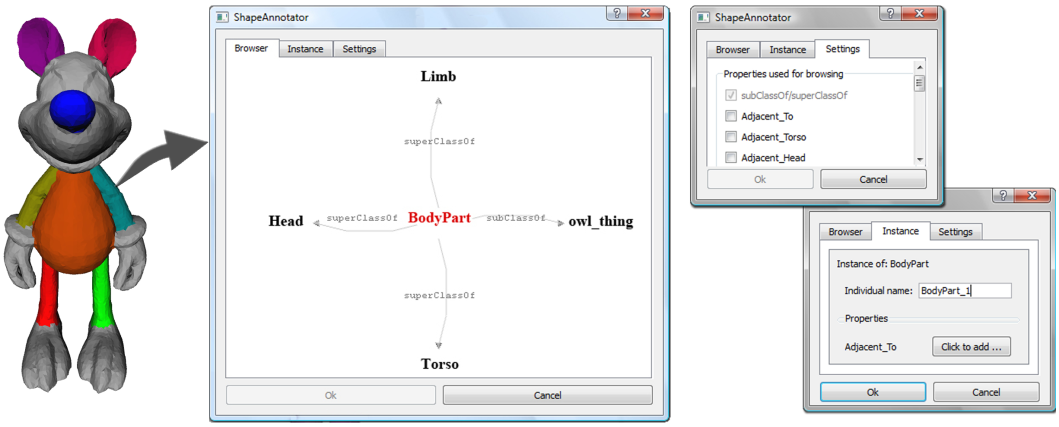

Figure 6.5: The ontology browser, the selection of navigation settings and the creation of an instance.

containment, overlap). Each of these measures is focused on a specific aspect of the segments, either intrinsic or in their relationship with other connected segments, and for this reason we call them *segment descriptors*.

Currently, they belong to the following two groups:

- **Topological relations between segments** consisting of *adjacency*, *overlap*, *disjointness* and *containment*;
- **Geometric aspects of a segment** consisting of *oriented bounding box length*, *width* and *height*, *best-fitting sphere radius*, *best-fitting cylinder radius*.

Some of these descriptors are immediately connectable with important properties of the segment (e.g. area, enclosed volume), but we also allow their combination within formulae, in order to obtain higher-level descriptors. Some examples can be $\texttt{volume}^2/\texttt{area}^3$, which measures the compactness of a segment and is invariant to its uniform scaling, or $\texttt{genus} > \texttt{0}$, which can be connected with the presence of holes. Anyhow, since segment descriptors are calculated independently of any ontology, the user may define their *interpretation* within each specific domain of annotation. The user may establish a set of *connections* between topological relations and conceptual descriptors (e.g. "segment adjacency" $\leftrightarrow$ `is_connected_to`) and between geometric descriptors and class attributes (e.g. "radius of best-fitting cylinder" $\leftrightarrow$ `through_hole` :: `radius`, "genus > 0" $\leftrightarrow$ `pierced`).

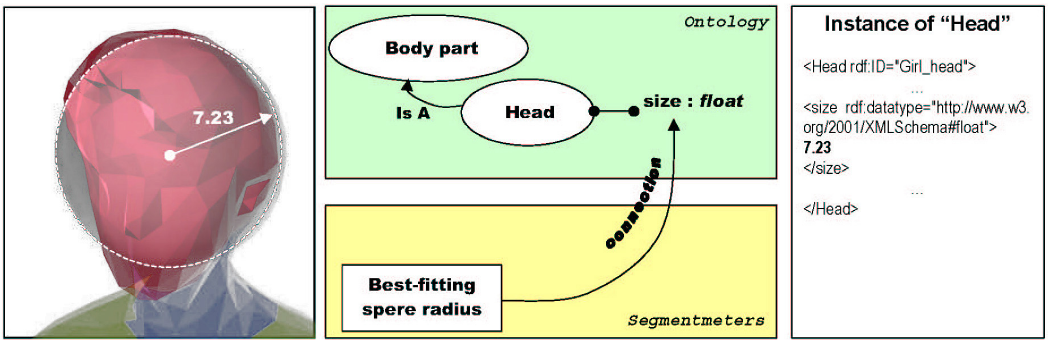

Figure 6.6: The attribute *size* of the instance *Girl_head* is automatically set to the value 7.23 because this is the value computed by the connected segment descriptor.

After having established such connections, the instance properties are transparently computed by the system. For example, when annotating a reverse engineered mechanical model, a part may be manually annotated as a `Through_hole`, while its parameter `radius` is automatically computed by the ShapeAnnotator as the radius of the cylinder that best fits the geometry of the part; if two adjacent segments are manually annotated as instances of the class `stiffener`, the relation `is_adjacent_to` is automatically set to conceptually link the two instances. An example of connection is shown in Figure 6.6.

Since we believe that modularity is crucial to provide a flexible annotation framework, we made our system able to load additional descriptors implemented externally as plug-ins, just as we have done for the segmentation algorithms, and to combine the descriptors available off the shelf in order to produce higher-level ones, through the help of a small editor in which the user can write formulae and save them for later use.

At this point, the connections can be established through a proper dialog in which all the segment descriptors are available in a drop-down menu; when one of them is chosen, a list of properties defined in the domain ontology is shown and the user may select some of them to establish connections. The list of properties shown is filtered so that only admissible connections can be picked; this avoids, for example, the connection of a property with more than one descriptor, or between non-compatible descriptors and ontology properties (e.g. "segment adjacency" ↔ `through_hole` :: `radius`).

The connections can be established either before the creation of instances or afterwards. In the former case, for each newly created instance the properties are computed on the fly based on the existing connections; in the latter case, the values of the properties of the existing instances are (re)computed according to the newly established connections.

To allow their easy reuse when annotating several models in the same domain, the connections can also be saved as an XML-based file and loaded later on.

The formulae used to combine simple descriptors into higher-level ones will be extended in the future to include conceptual characterizations in their bodies. For instance the formula "is-human AND height<120" could be used to create a (boolean) segment descriptor, connectable for instance with "is-children", allowing the construction of semantics on top of other forms of semantics in an automatic and layered way through the use of simple yet powerful forms of inference.

6.3.5 Resulting Knowledge Base

The result of the annotation process is a set of instances that, together with the domain ontology, form a knowledge base. Each instance is defined by its URI, its type (i.e., the class it belongs to) and some attribute values and relations that might have been specified/computed. In its current version, the ShapeAnnotator saves the multi-segmented mesh along with the selected, and possibly edited features as a single PLY file [75]. The instances are saved as a separate OWL file that imports the domain ontology. Additionally, the OWL file contains the definition of two extra properties:

- `ShannGeoContextURI`, whose value is the URI of the multi-segmented mesh (typically the path to the PLY file saved by the ShapeAnnotator);
- `ShannSegmentID`, whose value is an index that specifies a segment in the multi-segmented mesh.

All the instances produced during the annotation pipeline are automatically assigned values for the above two properties, so that the link between semantics and geometry is maintained within the resulting knowledge base (see Figure 6.7).

Note that the OWL files produced by the annotation of several models in the same domain can constitute a unified knowledge base; this would contain all the instances describing the models in terms of their meaningful parts, allowing unprecedented levels of granularity for query formulation. Once an instance has been located, for example, it is possible to retrieve the geometry of the corresponding part and, possibly, to extract it without the need to download and process the whole model.

6.4 Application Scenarios

In the following sections, I present two scenarios that describe how the part-based annotation framework might be used.

6.4.1 Virtual character design

In several application domains, ranging from simulation to entertainment and gaming, the problem of designing virtual characters arises. Whereas in the case of simulation the virtual characters, along with their motion capabilities, are often modeled after real humans [47], in the case of entertainment and gaming avatars most often come from imaginative inspiration. The design is done most of the times from scratch or through the personalization of a given set of parameters. This is the current status for the avatar creation in many *MMORPGs* and in online virtual worlds such as Second Life [73], where avatars can be created constructively, by selecting from predefined sets of values the shape of the body, the skin, the hair, the eyes, the clothes, and so on. Thus, the actual creative freedom is limited to the selection and combination of predefined attributes and parameters. As an example, we consider the path of a user interested in harvesting and selecting digital models of human *heads* having specific high-level characteristics (e.g. large, narrow, belonging to a male, belonging to a given ethnic group, with a long nose, with distant eyes), starting from a repository containing virtual humans. Currently he/she should browse the repository, calculate the parameters he/she is interested in (e.g. distance between eyes, length of nose), download each interesting model, and perform editing operations in order to get the parts of the models corresponding to *heads*. All of these steps should be performed manually. Thanks to the ShapeAnnotator the resources in the repository will be annotated (the parameters will be calculated automatically), and a dedicated knowledge base will contain direct instances of *heads*, corresponding to portions of the original models in the repository. Each *head* will be also conceptualized via attributes and relations, and so the proper selection could be performed independently of the specific geometry. Moreover, when

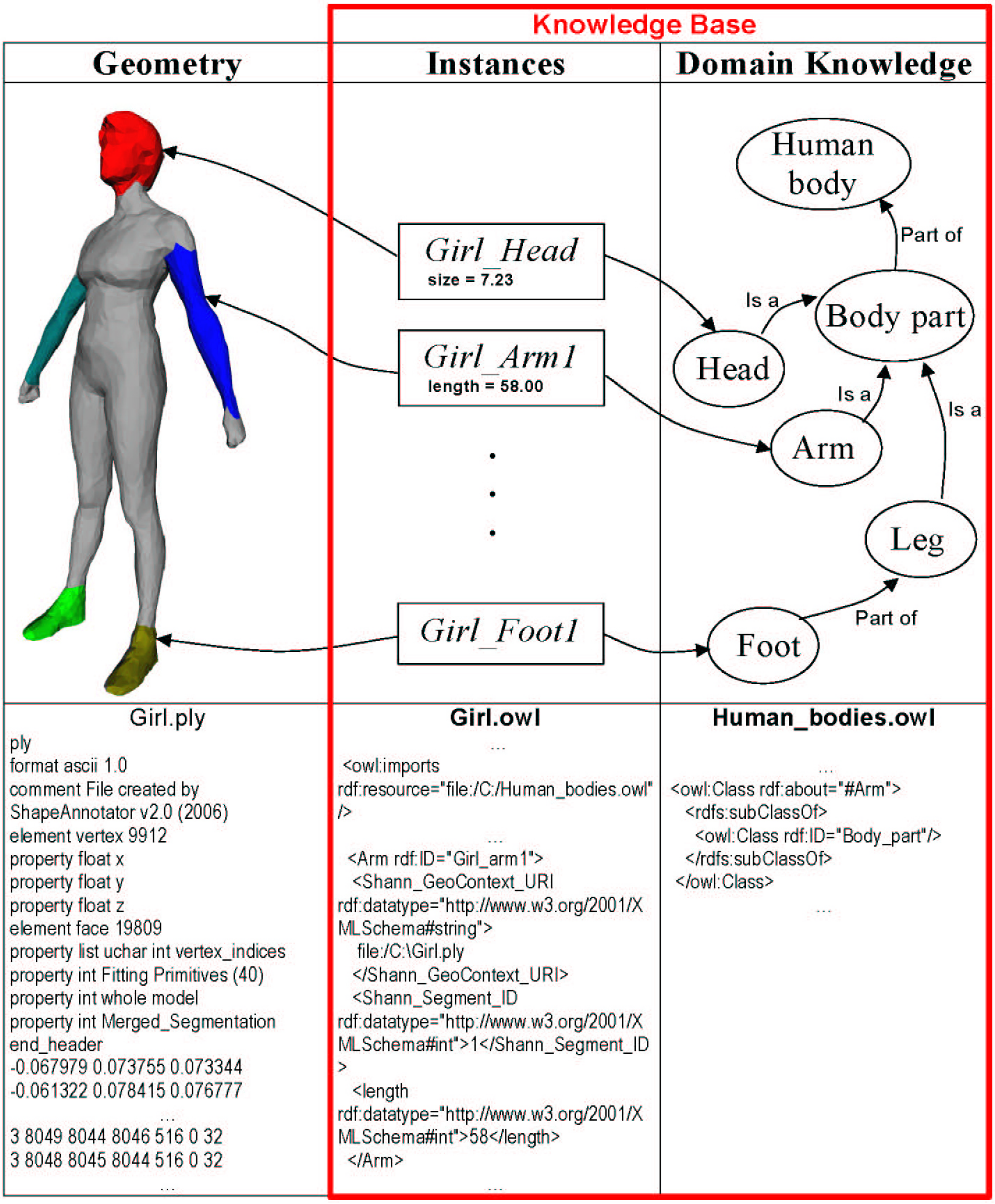

Figure 6.7: The instances along with their specific relations represent a formal bridge between geometry and semantics.

some interpretation rules will be coded, as it is foreseen in the ShapeAnnotator, the values of the parameters could be connected with some higher level characterization. Thus, the user will be able to search the repository directly for “heads of Caucasian adult male” or “heads of children”, or “heads of black women”. The virtual character creation effort could then be significantly eased.

6.4.2 E-manufacturing and product (re)design

The importance of annotating 3D data has been largely addressed in industrial manufacturing, where interoperability becomes a crucial issue and different solutions have been proposed to face it efficiently. For example, the introduction of the so-called Group Technology [82] in industrial product modelling allowed to model and code classes of objects that exhibit design and manufacturing similarities, with a favorable impact on design and production practices. This approach to design, however, did not provide means to explicitly support the annotation and indexing of “pieces” of the objects at hand, which is an important issue, already raised in Chapter 2 while talking about the levels of granularity in the domain of 3D object. Also, knowledge technologies have been proposed in industrial manufacturing, as an efficient methodology to formalize and share semantics in different contexts of the product development process. Product Lifecycle Management, or PLM, systems are the most common tool used in this domain, and such formalizations are mainly document-oriented and provide a low-level description of the product data with no special customized view. Also, there are examples of ontologies for the formalization of CAD/PDM/PLM knowledge (e.g. [52, 88]). However, there is still a lack of tools which exploit knowledge technologies for part-based annotation and retrieval of shape data, even if these are perceived as key issues by the community. In e-manufacturing, designers and engineers may collaborate remotely using Internet technologies to devise new products. In this field, a common practice for approaching the creation of new parts is to adapt existing ones to the new specifications or constraints: this is done either to support creativity in the early phase of the design or for cutting costs by reusing parts. The search is therefore the first step of the design pipeline: even a simple speed up of finding relevant 3D content has a direct monetary impact in itself, and, more importantly, it has a direct economic impact on manufacturing industry as locating precisely existing content that can be reused reduce dramatically the variety of parts to be modelled, managed, produced, and maintained.

Companies that adopt an automated part reuse strategy have the potential for significant cost savings. These parts, however, are often poorly classified or classified in different languages (for example, where acquisitions of companies in different countries has occurred). This means that textual searching alone is not sufficient to find the parts. This is where content-based searching becomes key, coupled with concept-based annotations.

To summarize, there are two main problems with current technologies:

- Metadata in state-of-the-art repositories are typically related to the whole shape. Thus, retrieving suitable parts that may belong to composite models in the repository is an extremely difficult task.
- Even if one assumes that a model can be retrieved, adapting it to the needs of the new product may be another hard task. In particular, if the model is reverse engineered, its parts are neither explicit nor annotated, and turning the polygon mesh into a feature-based model to be edited becomes a long and tedious operation.

Through the ShapeAnnotator, each model in the repository can be *abstracted* and represented as a combination of its meaningful features. Each such feature is described by an instance in the knowledge base in which attributes and relations with other features are explicit. Thus, all the relevant parts of all the models in the repository can be easily indexed and effectively retrieved. Moreover, once a reverse engineered polygon mesh has been retrieved, its features are explicit and their parameters properly instantiated, making the translation into an editable feature-based model much easier.

6.5 Discussion

In this Chapter, I have illustrated how to decompose the shape into interesting features within the multi-segmentation framework, and introduced the annotation pipeline to attach a formal semantics to the features and the whole shape.

The ShapeAnnotator demonstrates how the discussion on the role of concepts, context and content can concur to the realization of a prototype system for annotating 3D objects.

The formalization of concepts into an ontology and the selection of the ontology as a key element for the annotation points out the key role of the context in the annotation pipeline. The same object may be annotated according to different ontologies, that is, during different annotation sessions where the ontology loaded changes accordingly.

The role of content and related descriptors is also made clear by the strategy adopted to include segmentation plugins into the system. We have pointed out that the segmentation process is unfeasible using only state-of-the-art approaches, and the lack of really semantics-driven segmentation tools is mediated by an intelligent mix of tools, aiming at segmenting the shape according to complementary properties, and by the human intervention which determines the final segmentation. Moreover, the introduction of segment descriptors along with their context-based interpretation represents a first step towards automatic annotation methods for the 3D domain.

The ShapeAnnotator provides, obviously, just a partial answer to the many problems discussed and opens up new issues and perspectives.

In its current version, the ShapeAnnotator has minimal inference capabilities which have been implemented just to provide a flexible browsing of the ontology. This means that input ontologies are assumed to be mostly asserted; if not, the user can use an offline reasoner to produce the inferred parts. Future developments are targeted to this aspect, and internal inference capabilities are foreseen. Besides simple deductions on the input ontology, inference will also be used to (partially) automate the whole annotation pipeline. Although the process can be completely automated in rather few domains, in many others the user might be required to contribute only to disambiguate few situations. In future developments, we plan to treat also lower-dimensional features (i.e. curves and points) with a twofold benefit: they will be eligible for annotation, just as the other segments, and they will be useful to edit other segments (e.g. a curve may be used to split a segment into two subsegments).

Chapter 7

Conclusions

Di questo potrei parlare all'infinito,
ma odio Leopardi e tutti i poeti da pelliccia.
(Alessandro Bergonzoni)

This thesis has been a journey inside the concept of description itself. A radical approach to this issue implied a general rethinking of the whole subject. The impact of the assessments that have been made throughout the thesis is twofold. On the one hand I used these observations as building blocks for the design and the development of the original tools I presented, namely:

- a **set of ontologies** built for a conceptualization of 3D objects from a number of diverse points of view, presented in Chapter 3;
- the **ShapeAnnotator** tool, a flexible and modular system for part-based annotation of 3D objects, described in Chapter 6.

On the other hand I had the chance of setting up requirements and guidelines for next-generation tools, in particular search engines dealing (also) with 3D objects. Throughout the journey a lot of interesting points popped out. Among them it is possible to recall the following.

1. **The general issue of semantics.** The correct perspective to look at semantics, in my opinion, is not to look for "the semantics" of an object as a sort of Holy Grail to be searched and unveiled, but to cast an approach capable of associating a meaning to a sign: semantics is the discipline studying this very association.

This approach has major advantages. In the first place it is possible to import considerations made in the fields of linguistics and semiotics without the need of reinventing the wheel. In the second place, if a meaning has to be associated to a sign, there should be formal clearness about what are the eligible signs: am I referring to objects, to object parts, to collections of objects? In the third place it is very important to understand how should we express the targeted meaning: in natural language, through concepts, through signatures, through links to other objects? There is no single answer to this question, as any application domain has its own peculiarities and needs. There is not even a strict rule telling that metadata has to be necessarily textual. The versatility has to me maximized.

2. **The role of descriptions.** As I wrote in Chapter 2 in my opinion a description is a *looser form of representation* in which a *shift of language* and an *elaboration layer* are allowed. Different layers of elaboration give the freedom of dealing with the object in different contexts. In next-generation tools the user, whose role is more and more focal, should be enabled to activate different description layers, in order to search, retrieve and perceive objects through different perspectives. Moreover, flexible semiotic systems should allow the description of 3D objects not only via textual tags or numerical descriptors, but also via their connection with other resources, i.e. other 3D objects but also images, videos and so on.

3. **Concepts, the Semantic Web, and the Linked Data.** The acquisition of meaning by interlinking resources with one another is one of the aims of the Semantic Web. This is certainly an important point to be taken into account in the design of next-generation tools. Keeping in mind that this is not the only way to address semantics, as I have discussed in Chapters 3 and 5, it is still fundamental that objects and object parts can be connected with the ever-growing cloud of Open Linked Data. In particular, when ontologies are included in the signification loops, they should be as "live" as possible: a neat and perfectly tailored concept characterizing a 3D object is not so useful in the Semantic Web perspective if it is not linked, or easily linkable, with other shared data.

4. **Content-based annotation.** The fascination about the Semantic Web should not let us underestimate the power of content-based annotation. If we pursue the

aim indicated in Chapters 4 and 5 of characterizing shape descriptors in terms of a useful conceptualization, we will be enabled to extract information directly from the content-based signatures. This is a path already followed in the development of the ShapeAnnotator tool described in Chapter 6, but the need is to extract more fine-grained information and most of all to have the chance to integrate it with concept-based information. It is very desirable to have the objects belonging to the digital collection of a museum annotated with all their interesting characteristics. It is even more desirable to link these characteristics with the Linked Data out there. It would be awesome to let a system automatically produce some of those annotations, i.e. tagging an object part with its volume, its compactness, its sphericity, or even with higher-level (and possibly domain-dependent) concepts such as its architectonic style, its use or the name of the part it represents. To be really effective in practice, the formalism to describe the objects should realize a bridge between the properties of its shape and the computational methods that are able to handle them in the digital world.

5. **The user and the context.** The user is more and more important in the actual multimedia landscape. Therefore, he/she should be granted with a leading role in any process involving annotation and retrieval. He/she should be enabled to easily cast different viewpoints on the same object, in order to let an intelligent reuse of information and a fine tuning on the methods to be used, thus strengthening any annotation or retrieval system. This is a matter of correct conceptualization of the whole process but also of an ambitious high-level design of a powerful user interface easily allowing the user to express his/her context, his/her perspectives, his/her needs but also to assess his/her personal approach to the considered resources and to give feedback to the system.

Any system considering all the points exposed above will be endowed with the versatility necessary to stand out. It is possible to foresee scenarios in which users contribute in the semantic annotation of 3D objects, and then are enabled to pose queries such as "find the objects that represent a baroque vase with handles, and whose handles are globally similar in shape to the ones of the Champions League Cup". In the example *vase* and *handle* could refer to conceptual tags and be resolved via a semantic search, *baroque* could be a high-level concept extracted by an intelligent automatic system, *globally similar in shape* will be resolved by applying a geometric search using as a query by example the model of the handles of the *Champions League Cup* (available through an effective connection with the Linked Data cloud) and as signatures the ones produced by methods which emphasize the spatial distribution of the parts selected by the semantic search.

Bibliography

[1] R. Albertoni, L. Papaleo, M. Pitikakis, F. Robbiano, M. Spagnuolo, and G. Vasilakis. Ontology-based searching framework for digital shapes. In *On the Move to Meaningful Internet Systems 2005: OTM Workshops*, volume 3762 of *Lecture Notes in Computer Science*. Springer.

[2] R. Albertoni, L. Papaleo, and F. Robbiano. Preserving information from real objects to digital shapes. In R. De Amicis and G. Conti, editors, *Eurographics Italian Chapter Conference*, pages 79–86, feb 2007.

[3] R. Albertoni, L. Papaleo, F. Robbiano, and M. Spagnuolo. Towards a conceptualization for shape acquisition and processing. In *Proceedings of the 1st International Workshop on Shapes and Semantics, Matsushima, Japan*, pages 85–91, 2006.

[4] Alia Amin, Jacco van Ossenbruggen, Lynda Hardman, and Annelies van Nispen. Understanding cultural heritage experts' information seeking needs. In *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 39–47, New York, NY, USA, 2008. ACM.

[5] Remi Arnaud and Mark C. Barnes. *Collada: Sailing the Gulf of 3d Digital Content Creation*. AK Peters Ltd, 2006.

[6] M. Attene and S. Biasotti. Shape retrieval contest 2008: Stability of watertight models. In *Proc. IEEE SMI 2008*, pages 219–220, 2008.

[7] M. Attene, S. Biasotti, M. Mortara, G. Patane, and B. Falcidieno. Computational methods for understanding 3D shapes. *Computers&Graphics*, 30(3):323–333, 2006.

[8] M. Attene and B. Falcidieno. Remesh: An interactive environment to edit and repair triangle meshes. In *Proc. SMI 2006 – IEEE International Conference on Shape Modeling and Applications*, pages 271–276. IEEE Computer Society Press, 2006.

[9] M. Attene, F. Robbiano, G. Patanè, M. Mortara, M. Spagnuolo, and B. Falcidieno. Semantic annotation of digital 3D objects. In Riccardo Albertoni and Michela Mortara, editors, *Posters and Demo Proceedings of Second International Conference on Semantic and digital Media Technologies*, 2007.

[10] M. Attene, F. Robbiano, M. Spagnuolo, and B. Falcidieno. Semantic annotation of 3D surface meshes based on feature characterization. In *Proceedings of SAMT, Genova, Italy*, volume 4816 of *Lecture Notes in Computer Science*. Springer.

[11] M. Attene, F. Robbiano, M. Spagnuolo, and B. Falcidieno. Characterization of 3d shape parts for semantic annotation. *Computer Aided Design*, 41(10):756–763, 2009.

[12] Marco Attene, Bianca Falcidieno, and Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.

[13] Marco Attene, Sagi Katz, Michela Mortara, Giuseppe Patanè, Michela Spagnuolo, and Ayellet Tal. Mesh segmentation - a comparative study. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, page 7, Washington, DC, USA, 2006. IEEE Computer Society.

[14] S. Biasotti. *Computational Topology Methods for Shape Modelling Applications.* PhD thesis, Universitá degli Studi di Genova, May 2004.

[15] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1-3):5–22, 2008.

[16] S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview on properties and efficacy of topological skeletons in shape modelling. In *SMI '03: Proceedings of*

the Shape Modeling International 2003, page 245, Washington, DC, USA, 2003. IEEE Computer Society.

[17] Silvia Biasotti. *Computational Topology methods for Shape Modelling Applications*. PhD thesis, University of Genoa, Italy, 2004.

[18] Silvia Biasotti, Simone Marini, Michela Mortara, Giuseppe Patanè, Michela Spagnuolo, and Bianca Falcidieno. 3d shape matching through topological structures. In *DGCI*, pages 194–203, 2003.

[19] Rafael Bidarra and Willem F. Bronsvoort. Semantic feature modelling. *Computer-Aided Design*, 32(3):201–225, 2000.

[20] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[21] I.M. Bilasco, J. Gensel, M. Villanova-Oliver, and H. Martin. 3DSEAM: a model for annotating 3D scenes using MPEG-7. In *Proc. 7th IEEE International Symposium on Multimedia (ISM'05)*, pages 310–319. IEEE Computer Society, 2005.

[22] Rebecca Blood. Weblogs: A history and perspective, 2000.

[23] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of nonrigid shapes*. Springer, 2008.

[24] Paul Browne and Alan F. Smeaton. Video retrieval using dialogue, keyframe similarity and video objects. In *ICIP 2005 - International Conference on Image Processing*, 2005.

[25] A. Carreras, G. Cordara, J. Delgado, F. Dufaux, G. Francini, T. M. Ha, E. Rodriguez, and R. Tous. A search and retrieval framework for the management of copyrighted audiovisual content. In *Proc. of the 50th International Symposium ELMAR 2008*, 2008.

[26] C.E. Catalano, E. Camossi, R. Ferrandes, V. Cheutet, and N. Sevilmis. A product design ontology for enhancing shape processing in design workflows. *Journal of Intelligent Manufacturing*, 20(5):553–567, 2009.

[27] C.E. Catalano, B. Falcidieno, M. Attene, F. Robbiano, and M. Spagnuolo. Shape knowledge annotation for virtual product sharing and reuse. In *Proceedings of ASME Conference on Engineering Systems Design and Analysis, Haigfa, Israel*, July 2008.

[28] D. Chen, M. Ouhyoung, X. Tian, and Y. Shen. On visual similarity based 3D model retrieval. *Computer Graphics and Applications*, (22):223–232, 2003.

[29] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. *ACM Trans. Graph.*, 28(3):1–12, 2009.

[30] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 905–914, New York, NY, USA, 2004. ACM Press.

[31] J. Corney, H. Rea, D. Clark, J. Pritchard J, M. Breaks, and R. Macleod. Coarse filters for shape matching. *IEEE Comput. Graph. Appl.*, 22(3):65–74, 2002.

[32] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: ideas, influences and trends of the new age. *ACM Computing Surveys*, 40(2), 2008.

[33] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.

[34] B. Falcidieno, M. Spagnuolo, P. Alliez, E. Quak, M. Vavalis, and C. Houstis. Towards the semantics of digital shapes: The aim@shape approach. In *Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, 2004.

[35] Alberto Córdoba Francisco Echarte, José J. Astrain and Jesús Villadangos. Ontology of folksonomy: A new modeling method. In *Semantic Authoring, Annotation and Knowledge Markup (SAAKM)*, 2007.

[36] Robert G. Gallager. *Information Theory and Reliable Communication.* John Wiley & Sons, Inc., New York, NY, USA, 1968.

[37] D. Giorgi, P. Frosini, M. Spagnuolo, and B. Falcidieno. Multilevel relevance feedback for 3d shape retrieval. In *Proceedings 3DOR'09: Eurographics Workshop on 3D Object Retrieval*, pages 45–52, 2009.

[38] C. Grana, M. Davolio, and R. Cucchiara. Similarity-based retrieval with MPEG-7 3D descriptors: Performance evaluation on the Princeton Shape Benchmark. In *Digital Libraries: Research and Development*, volume 4877 of *Lecture Notes in Computer Science*, pages 308–317. Springer Berlin – Heidelberg, 2007.

[39] N. Guarino. Formal ontology and information systems. In N. Guarino, editor, *Formal Ontology in Information Systems*, pages 3–18. IOS Press, Amsterdam, 1998.

[40] A. Guéziec, G. Taubin, F. Lazarus, and B. Horn. Cutting and Stitching: Converting Sets of Polygons to Manifold Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, pages 136–151, 2001.

[41] Mario Gutierrez A., Alejandra Garcia-Rojas, Daniel Thalmann, Frederic Vexo, Laurent Moccozet, Nadia Magnenat-Thalmann, Michela Mortara, and Michela Spagnuolo. An ontology of virtual humans: Incorporating semantics into human shapes. *Vis. Comput.*, 23(3):207–218, 2007.

[42] J. Hartman and J. Wernecke. *The VRML 2.0 handbook–building moving worlds on the web.* Addison Wesley Developers Press, 1997.

[43] S. Hassan, F. Hetroy, and O. Palombi. Ontology-guided mesh segmentation. In *Proc. FOCUS K3D Conference on Semantic 3D Media and Content*, volume LNCS 2728, pages 33–37, 2010.

[44] Th. Hermes, A. Miene, and P. Kreyenhop. On textures: A sketch of a texture-based image segmentation approach. In R. Decker and W. Gaul, editors, *Classification and Information Processing at the Turn of the Millenium*, pages 210–218, 10 - 12 March 2000. Proc. 23rd Annual Conference Gesellschaft für Klassifikation e.V. 1999.

[45] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH 2001: Proceedings of the* 28^{th} *Annual Conference on Computer Graphics and Interactive Techniques*, pages 203–212. ACM Press, 2001.

[46] F. Jameson. *The prison-house of language.* Princeton, NJ, 1972.

[47] P. Kalra, N. Magnenat-Thalmann, L. Moccozet, G. Sannier, and A. Aubel. Real-time animation of realistic virtual humans. *IEEE Computer Graphics and Applications*, 18(5):42–55, 2006.

[48] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of SGP 2003: Eurographics Symposium on Geometry Processing*, pages 156–165, 2003.

[49] Melanie Kellar, Carolyn Watters, and Kori M. Inkpen. An exploration of web-based monitoring: implications for design. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 377–386, New York, NY, USA, 2007. ACM.

[50] G. Kemerling. Definition and meaning. http://www.philosophypages.com/lg/e05.htm.

[51] Duck Hoon Kim, In Kyu Park, Il Dong Yun, and Sang Uk Lee. A new mpeg-7 standard: Perceptual 3-d shape descriptor. In *Advances in Multimedia Information Processing – PCM 2004*, volume 3332 of *Lecture Notes in Computer Science*, pages 238–245. Springer Berlin – Heidelberg, 2005.

[52] Y. Kitamura and R. Mizoguchi. Ontology-based systematization of functional knowledge. *Journal of Engineering Design*, 15:327–351, 2004.

[53] Jeehyung Lee and Thomas Funkhouser. Sketch-based search and composition of 3D models. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, 2008.

[54] B.S. Manjunath, P. Salembier, and T. Sikora, editors. *Introduction to MPEG-7: Multimedia Content Description Interface.* Wiley, 2002.

[55] D. Marr. *Vision - A computational investigation into the human representation and processing of visual information: An invitation to Cognitive Science.* W. H. Freeman, 1982.

[56] A. Mathes. Folksonomies-cooperative classification and communication through shared metadata. *Computer Mediated Communication*, 2004.

[57] M. Mortara, G. Patane', and M. Spagnuolo. From geometric to semantic human body models. *Computers & Graphics-Uk*, 30(2):185–196, 2006.

[58] Michela Mortara, Giuseppe Patanè, Michela Spagnuolo, Bianca Falcidieno, and Jarek Rossignac. Plumber: A method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In *SM '04: Proceedings of the ACM Symposium on Solid Modeling*, pages 339–344, New York, NY, USA, 2004. ACM Press.

[59] A. Noruzi. Webology: Editorial, volume 4, number 2. http://www.webology.ir/2007/v4n2/editorial12.html, 2007.

[60] American Psychiatric Association. Task Force on DSM-IV. DSM-IV: diagnostic and statistical manual of mental disorders. 1994.

[61] [online]. Delicious. http://delicious.com/.

[62] [online]. The dublin core metadata initiative. http://dublincore.org/.

[63] [online]. Flickr. http://www.flickr.com/.

[64] [online]. http://www.jainworld.com/literature/story25.htm.

[65] [online]. http://www.w3.org/tr/rdfa-in-html/.

[66] [online]. http://www.web3d.org/.

[67] [online]. Humanoid animation working group. http://www.h-anim.org/.

[68] [online]. Microformats. http://microformats.org/wiki/main_page.

[69] [online]. Mpeg-7. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[70] [online]. Owl, web ontology language overview. http://www.w3c.org/2004/owl/.

[71] [online]. Photostuff. http://www.mindswap.org/2003/photostuff/.

[72] [online]. Rdf semantics. http://www.w3.org/tr/rdf-mt/.

[73] [online]. Secondlife. http://www.secondlife.com.

[74] [online]. The semantic web. http://semanticweb.org/.

[75] [online]. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3dscanrep.

[76] [online]. The NCI cancer ontology. http://www.mindswap.org/2003/cancerontology, 2003.

[77] [online]. AIM@SHAPE digital shape workbench. http://dsw.aimatshape.net/, 2004.

[78] [online]. AIM@SHAPE shape repository. http://shapes.aimatshape.net/, 2004.

[79] [online]. Network of excellence AIM@SHAPE, EU FP6 IST NoE 506766. http://www.aimatshape.net, 2004.

[80] [online]. OWL web ontology language guide. http://www.w3.org/tr/2004/rec-owl-guide-20040210/, Feb 2004. W3C Recommendation.

[81] [online]. The Protégé ontology editor. http://protege.stanford.edu, 2006.

[82] H. Opitz. *A Classification System to Describe Workpieces*. Pergamon Press, NY, 1970.

[83] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002.

[84] L. Papaleo, R. Albertoni, S. Marini, and F. Robbiano. An ontology-based approach to acquisition and reconstruction. In *Proceedings of Workshop Towards Semantic Virtual Environments, Villars, Switzerland*, 2005.

[85] T. Pavlidis and Y.-T. Liow. Integrating region growing and edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(3):225–233, 1990.

[86] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, I. Kompatsiaris, and S. Staab. M-OntoMat-Annotizer: Image annotation. linking ontologies and multimedia low-level features. In *KES 2006 - 10th Intntl. Conf. on Knowledge Based, Intelligent Information and Engineering Systems*, oct 2006.

[87] M. Petrou and P. Bosdogianni. *Image processing: the fundamentals.* John Wiley, 1999.

[88] J. Posada, C. Toro, S. Wundrak, and A. Stork. Using ontologies and step standards for the semantic simplification of cad models in different engineering domains. *Applied Ontologies*, 1(3-4):263–279, 2006.

[89] G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 222:847–849, 1946.

[90] Aristides G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Comput. Surv.*, 12(4):437–464, 1980.

[91] M. Reuter. *Laplace Spectra for Shape Recognition.* Books on Demand, ISBN 3-8334-5071-1, 2006.

[92] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as Shape-DNA of surfaces and solids. *Computer Aided Design*, 38(4):342–366, 2006.

[93] F. Robbiano, M. Attene, M. Spagnuolo, and B. Falcidieno. Part-based annotation of virtual 3D shapes. In *Proceedings of the 2007 International Conference on Cyberworlds, spec. sess. on NASAGEM workshop, Hannover, Germany*, pages 427–436, Washington, DC, USA, 2007. IEEE Computer Society.

[94] F. Robbiano, M. Spagnuolo, and B. Falcidieno. The role of contexts and descriptors for expressing semantics in 3d media. In *Proceedings of the SAMT Workshop on Semantic 3D Media*, pages 53–57, 2008.

[95] S. Santini. An oddly-positioned position paper on context and ontology. In *Proc. 2nd IEEE international conference on semantic computing*, 2008.

[96] J.J Shah and M. Mäntylä. *Parametric and feature-based CAD/CAM.* John Wiley & Sons, 1995.

[97] Ariel Shamir. Segmentation and shape extraction of 3D boundary meshes. In *Eurographics 2006 - State of the Art Reports*, pages 137–149, 2006.

[98] C. Shirky. Ontology is overrated: Categories, links, and tags. http://shirky.com/writings/ontology_overrated.html.

[99] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[100] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies.* International Handbooks on Information Systems. Springer, 2004.

[101] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings of SMI 2003: International Conference on Shape Modeling and Applications*, pages 130–139. IEEE Computer Society, 2003.

[102] J.W.H. Tangelder and R.C. Veltkamp. A survey of content-based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.

[103] Daniel Thalmann. The virtual human as a multimodal interface. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 14–20, New York, NY, USA, 2000. ACM.

[104] D'Arcy Wentworth Thompson. *On growth and form.* Cambridge University Press, Cambridge, Eng., 1942.

[105] James Hendler Tim Berners-Lee and Ora Lassila. The semantic web. *Scientific American Magazine*, 2001.

[106] G. Vasilakis, A. Garcia-Rojas, L. Papaleo, C.E. Catalano, F. Robbiano, M. Spagnuolo, M. Vavalis, and M. Pitikakis. A common ontology for multi-dimensional shapes. In *Proceedings of 1st International Workshop on Multimedia Annotation and Retrieval enabled by Shared Ontologies, Genova, Italy*, 2007.

[107] G. Vasilakis, A. Garcia-Rojas, L. Papaleo, C.E. Catalano, F. Robbiano, M. Spagnuolo, M. Vavalis, and M. Pitikakis. Knowledge-based representation of 3d media (in press, to appear in april 2010). *International Journal of Software Engineering and Knowledge Engineering*, 2010.

[108] Patrizia Violi. *Meaning and Experience*. Bloomington: Indiana University Press, 2001.

[109] D. Vranic. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[110] D. Vranic, D. Saupe, and J. Richter. Tool for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In J.-L. Dugelay and K. Rose, editors, *Proc. of IEEE 2001 Workshop on Multimedia Signal Processing*, pages 293–298, 2001.

[111] T. Vander Wal. Folksonomy coinage and definition. http://vanderwal.net/folksonomy.html, 2004.

[112] Hong Wu, Hanqing Lu, and Songde Ma. Multilevel relevance judgement, loss function, and performance measure in image retrieval. In *CIVR*, pages 101–110, 2003.

[113] Yu Xiaohan, Juha Ylä-Jääski, and Yuan Baozong. A new algorithm for texture segmentation based on edge detection. *Pattern Recognition*, 24(11):1105–1112, 1991.

[114] Y. yang, H. Lin, and Y. Zhang. Content-based 3D model retrieval: a survey. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and reviews*, 37(6):1081–1098, 2007.

[115] M. Yu, I. Atmosukarto, W.K. Leow, Z. Huang, and R. Xu. 3d model retrieval with morphingbased geometric and topological feature maps. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE Computer Society Press, 2003.

[116] T Zaharia and F Prêteux. 3d versus 2d/3d shape descriptors: A comparative study. In *Proc. SPIE Conf. on Image Processing: Algorithms and Systems*, 2004.

[117] C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *Proc. IEEE International Conference on Image Processing*. IEEE Computer Society Press, 2001.

[118] Eugene Zhang, Konstantin Mischaikow, and Greg Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics*, 24(1):1–27, 2005.