Semantics for Virtual Humans

THÈSE N^{O} 4301 (2009)

PRÉSENTÉE LE 19 FÉVRIER 2009

À LA FACULTE INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE RÉALITÉ VIRTUELLE
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alejandra GARCIA ROJAS MARTINEZ

acceptée sur proposition du jury:

Prof. A. Schiper, président du jury
Prof. D. Thalmann, directeur de thèse
Dr L. Papaleo, rapporteur
Prof. C. Pelachaud, rapporteur
Prof. S. Spaccapietra, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

Résumé

Les personnes qui se créent un personnage virtuel comme dans Second Life, par exemple, font accroître la population des mondes virtuels. Peu à peu, l'évolution de la technologie permet d'augmenter encore plus le réalisme des mondes virtuels en créant des Humains Virtuels plus crédibles. Néanmoins, réunir les ressources nécessaires à la création de cette crédibilité n'est une tâche facile. Ceci est en grande partie du à la complexité du processus de création des Humains Virtuels. Et même s'il existe déjà de nombreuses modèles d'Humains Virtuels, leur réutilisation s'avère difficile car nous ne disposons pas de toute l'information permettant de l'évaluer pour savoir s'il possède les caractéristiques nécessaires.

De plus, la connaissance liée à la création d'Humains Virtuel n'est ni connue par tout le monde, ni bien diffusée. Il y a en effet différentes techniques de création, différents composants logiciels, et différents processus pour créer un Humains Virtuel capable d'habiter dans un type d'environnement virtuel. Leur création implique notamment une représentation géométrique avec une structure de contrôle interne, la synthèse du mouvement appropriée utilisant différentes techniques d'animation, des contrôleurs de haut niveau ainsi que des descripteurs pour simuler leur comportement, leur caractère, leur connaissance, ou leurs méthodes d'interaction avec leur environnement. Créer ou simuler ses caractéristiques requière des compétences dans différents domaines tels que les mathématiques, l'intelligence artificielle, l'infographie ou le graphisme.

Par ailleurs, il n'existe ni une méthode commune, ni même une compréhension collective sur la façon dont sont faits les éléments permettant de créer les Humains Virtuels. Par conséquent, il est nécessaire de décrire les ressources existantes (1), de présenter les caractéristiques et la composition des Humains Virtuels (2), de schématiser le processus de création (3), et d'expliquer les différents niveaux et domaines de connaissance qui sont en jeu (4). Cette thèse propose donc une description sémantique des Humains Virtuels et en fournit une représentation explicite, avec leurs caractéristiques propres. Cela afin d'apporter un cadre de travail conceptuel qui retiendra l'attention de toutes les personnes impliquées dans leur création et dans leur développement.

La création d'une description sémantique implique entre autres, une collecte des connaissances, une définition de concepts commune entre les experts et la validation de la conception de l'ontologie. Cette thèse présente toutes ces procédures et fait une description approfondie d'une Ontologie des Humains Virtuels accompagnée de sa validation.

La création de cette ontologie a pour but de promouvoir la réutilisation des ressources existantes, de créer un partage des connaissances sur la création et la composition des Humains Virtuels, et de fournir une base aux nouvelles recherches liées au développement d'Humains Virtuels réaliste et de leurs environnements. Enfin, cette thèse présente plusieurs développements qui ont pour but de démontrer l'utilisation et la réutilisation de l'ontologie. Ces développements serviront notamment à aider la recherche sur les connaissances des Humains Virtuels, à améliorer leur réalisme et faciliter la population des mondes virtuels.

Mots-Clef: Réalité Virtualle, Humains Virtuels, Caractère 3D, Semantique, Ontologies.

Abstract

Population of Virtual Worlds with Virtual Humans is increasing rapidly by people who want to create a virtual life parallel to the real one (i.e. Second Life). The evolution of technology is smoothly providing the necessary elements to increase realism within these virtual worlds by creating believable Virtual Humans. However, creating the amount of resources needed to succeed this believability is a difficult task, mainly because of the complexity of the creation process of Virtual Humans. Even though there are many existing available resources, their reusability is difficult because there is not enough information provided to evaluate if a model contains the desired characteristics to be reused.

Additionally, the knowledge involved in the creation of Virtual Humans is not well known, nor well disseminated. There are several different creation techniques, different software components, and several processes to carry out before having a Virtual Human capable of populating a virtual environment. The creation of Virtual Humans involves: a geometrical representation with an internal control structure, the motion synthesis with different animation techniques, higher level controllers and descriptors to simulate human-like behavior such individuality, cognition, interaction capabilities, etc. All these processes require the expertise from different fields of knowledge such as mathematics, artificial intelligence, computer graphics, design, etc.

Furthermore, there is neither common framework nor common understanding of how elements involved in the creation, development, and interaction of Virtual Humans features are done. Therefore, there is a need for describing (1) existing resources, (2) Virtual Human's composition and features, (3) a creation pipeline and (4) the different levels/fields of knowledge comprehended. This thesis presents an explicit representation of the Virtual Humans and their features to provide a conceptual framework that will interest to all people involved in the creation and development of these characters. This dissertation focuses in a semantic description of Virtual Humans.

The creation of a semantic description involves gathering related knowledge, agreement among experts in the definition of concepts, validation of the ontology design, etc. In this dissertation all these procedures are presented, and an Ontology for Virtual Humans is described in detail together with the validations that conducted to the resulted ontology.

The goal of creating such ontology is to promote reusability of existing resources; to create a shared knowledge of the creation and composition of Virtual Humans; and to support new research of the fields involved in the development of believable Virtual Humans and virtual environments. Finally, this thesis presents several developments that aim to demonstrate the ontology usability and reusability. These developments serve particularly to support the research on specialized knowledge of Virtual Humans, the population of virtual environments, and improve the believability of these characters.

Keywords: Virtual Reality, Virtual Humans, 3D Characters, Semantics, Ontologies.

Acknowledgements

First of all I want to thanks Daniel Thalmann for giving me the opportunity to be part of his team. To the members of the jury for accepting this thesis: Laura Papaleo, Catherine Pelachaud and Stefano Spaccapietra; and to the president of the jury André Schiper.

I would like to mention some people from AIM@SHAPE NoE who was involved in the realization of my research: Bianca Falcidieno, Laura, Chiara Catalano, Michela Spagnuolo, George Vasilakis, Marios Pitikakis, Laurent Moccozet, Francesco Robbiano, and the young researchers Waqar Saleem and Samir Hammann. I learnt so much from this project and I found very nice persons and good friends. Also thanks to the great people I meet in HUMAINE NoE Catherine Pelachaud, Marc Schröder, Roddy Cowie, Tanja Baenziger, Jean Claude Martin; some others who I will keep tracking thanks to Facebook: Claudia Marinetti, Amaryllis Raouzaiou, Lori Malatesta, George Caridakis, Kostas Karpouzis, Mina Vasalou and Etienne Roesch.

I want to give a special thanks to Mario Gutiérrez because he thought of me for this PhD, and to encourage my spirit of research. To Frédéric Vexo for supporting Mario's idea of bringing me to VRLab. An special thanks to Josiane Bottarelli, for worrying about us and for buying me flights tickets - I enjoyed all conferences.

To all VRLab team, who were there for me when I was sick. Xavier and Letitia for sharing with us their friendship and good dinners. To Achille for his funny acid humor. Sylvan for sharing Indy even he pissed in our apartment sometimes. Mathieu for taking care of Cleo. To Damien for sharing good concerts and groupie activities. Jonathan and Barbara for borrowing me their book and sharing tips. To Eshan for coming to all the Mexican parties. To Patrick for making us try wine from Valais. Shubert for making cachaça for us and taking care of the scanner. Pablo and Ivana for letting me be the camera-woman on their wedding. Helena and Benöit for their poker evenings, even I was almost never there, but Renaud enjoy them a lot. Thierry Michellod for those life conversations and for encouraging me to go swimming. To Mireille and Olivier for their kindness and sympathy. To Hector for being positive and loyal person.

When I was not at VRLab, I shared the adventure of being in Switzerland with many people. Thanks to Mila Tevez and Isabel Orduz for adopting me as sister. To Anne-Marie Clerc for being the best host ever. To Daniela Nikolova for sharing crazy thoughts. To my "compadres" Cendrine Saugy and Fernando Unda for their friendship. To my Mexican friends who made great this last year for me: Gaby Quintero, Mario Salgado and Luis Garcia Naranjo for the afternoons we went running and the great weekends. To Luis Miery for his support to go to Satellite after work. To Magali Mapraud for being the best neighbor and for letting me paint her walls as occupational therapy. To the unconditional friendship of Magdalena and Manuel Sanchez.

Finally, I want to recognize the valuable support from my parents, I thank them for everything I've achieved. To my brothers and my family that I missed a lot every Sunday. And to the most important person in these last four years, who has being with me in the good and the bad times: Renaud Ott, my soul mate. Thanks also to Renaud's family Thierry and Loyce for letting me be part of their lives.

Contents

List of Figures

Chapter 1

Introduction

Real-time virtual environments realized over the Internet are gradually becoming more accessible. Interesting role-playing games such as Everquest[1] and World of War Craft[2], allow people to interact and collaborate to achieve goals. More revolutionary environments are social environments such as second life[3], where people can embody a 3D Virtual Human and create a virtual life with other human beings. These environments aim to globally enhance people's communication using the Internet.

Within these worlds people can create their own Virtual Characters and can personalize them by simply choosing the hair color, skin, cloths or accessories. They can also buy and sell characters' features that have already obtained some level of power in the game. There are official web sites such as sonny exchange, or even in eBay, that specialize in such transactions.

The diversity of 3D content inside these virtual environments are limited to the production by specialized companies, and the creation of virtual environments is mainly done by experts. However, new technology such as 3D scanning and motion capture are evolving in a way that the creation pipeline of virtual characters and environments are being automated. Results of these processes are large databases of models and animations that could be better exploited if they were accessible and well described.

3D models and animations are only a part of what its needed to create virtual environments. There are many more resources involved in the process such as artificial intelligence algorithms for controlling characters, motion control techniques, psychological models of behavior, etc. These kinds of resources provide and require extra information that is generally result of research studies and is not shared in a reusable way. The lack of a common framework makes difficult to integrate the achievements made by research communities committed to creating believable agents. Moreover, many 3D and virtual reality applications could be developed if the available resources were better described, and the dissemination of the knowledge were represented in an understandable and computable way.

1.1 Problem Statement

It is a challenge to successfully populate 3D environments with autonomous agents or avatars because of the complexity of simulating Humans. The knowledge of the creation of Virtual Humans has remained for experts. Virtual environments cannot easily integrate or reuse exiting resources because of the lack of common understandable descriptors. Thus, the creation of environments that allow non-experts to reuse existing Virtual Humans and their features, is a complex task.

[1] http://everquest.station.sony.com/
[2] http://www.worldofwarcraft.com
[3] http://secondlife.com/

This dissertation aims to formalize this knowledge to speed up the creation of Virtual Humans and the population of 3D environments. The objective is to find out how Virtual Humans and their features can be “labeled” in computational systems in order to facilitate their interchange, scalability, and adaptability according to specific needs. A semantic description of Virtual Humans will contribute to a common understanding among different research fields that aim at the creation of realistic Virtual Humans.

Thus, the research questions posed in this dissertation are:

- **How to describe Virtual Human components using regular (common) vocabulary or simplified representations?**; and
- **How this representation can support the creation, population and control of inhabited virtual environments?**

A semantic representation of a Virtual Humans must be closely related to a geometrical one. It is particularly required to be able to move from the geometric representation towards the semantic driven one. This is possible with the analysis of 3D graphical representation in order to query the 3D models for semantic information. It is also required to be able to go from the semantic description to the graphical representation. The integration of the semantic descriptors in the modeling and animation process is proposed to be able to construct the graphical representation of a Virtual Human from the semantic descriptors. In this representation not only geometric features are required, but also more meaningful ones that describe believable Virtual Humans.

1.2 Approach

Semantics refers to the meaning involved in any kind of expression, language, or sign. It is the formalization of the meaning of concepts; it makes explicit an assumption that is implicit. The “body” of semantics in computer sciences is usually distinguished by ontologies. When using ontologies for adding semantics, one can represent knowledge with diagrams (human readable), and at the same time use a markup language (machine readable). Providing an ontology for Virtual Humans represents a step towards to a common understanding, and shared knowledge about their creation and composition.

The creation of an ontology requires a common understanding among experts in the domains, and different levels of knowledge need to be covered in order to provide a good foundation of such ontology. Therefore, the creation of an ontology for Virtual Humans was carried out in the context of AIM@SHAPE NoE [112] project. This consortium brought together the collaboration of experts from the fields of artificial intelligence, mathematics, computer engineers, computer graphics and virtual reality. Within AIM@SHAPE several ontologies were created to converged into the propose of describing with semantics the creation of geometrical shapes. Furthermore, a semantic driven shapes and tools repositories called the Digital Shape Workbench (DSW). The DSW provides an infrastructure to support research by offering semantic search of shapes, tools and knowledge.

Thus, one of the ontologies developed within AIM@SHAPE project and for this dissertation is an ontology for Virtual Humans. This ontology is related with the other ontologies; and during its development there was contribution in the design process of the other ontologies and in the development of the DSW.

Thus, the aim of the project is the creation of ontologies that:

- enable the reuse of the domain knowledge to improve interoperability among applications that use the domain knowledge;
- provide to research community specific tools and resources that help to speed up their research;

- provide new resources for testing applications;
- integrate the expertise coming from different communities; and, to disseminate specific context-dependent knowledge.

The methodology proposed for the creation of these ontologies defines first a scope and user scenarios to describe for what the ontology will be used for. The ontology development involved experts from related research fields to interchange and agree in the needed definition of concepts. These consensuses proposed the semantics that needed to be validated by populating and querying the ontologies. Finally, the usage of ontologies is demonstrated with the creation of ontology-based applications.

1.3 Contribution

The main contribution of this work is an ontology for Virtual Humans that serves as a conceptual framework for the creation of virtual humans and population of virtual environments. Together with this ontology the development of several applications demonstrate that a conceptual framework like an ontology can support the creation and research of Virtual Humans and inhabited virtual environments.

1.3.1 Virtual Human Ontology

The Virtual Human Ontology (VHO) specializes in Virtual Human's description at geometric, structural and semantic levels. This ontology aims to provide sufficient information to simplify the comprehension of Virtual Humans composition, and to promote information sharing between creators and users, experts and non experts.

A semantic description of Virtual Humans *per se* should consider the following main fields of knowledge:

- Geometry: The geometry is the visual representation of Virtual Humans. It is composed of a body shape with features, accessories, garments, etc. Relevant information concerning the geometry is the creation process, the integration of features, the information inherent to the shape such as the morphology, anthropometry, etc. The geometry should contain features such as a skeletal structure to be able to animate it.
- Animation: The Virtual Human can be animated using prerecorded movements such as Key Framed or Motion Captured movements. Procedural animation is used, where algorithms drive movements like walking, reaching, taking, etc. There are also motion control techniques that uses hybrid methodologies to control and create new animation sequences. The creation of realistic animations is expensive in time and resources. Most of the animations are not completely exploited (reused) due to the lack of meaningful descriptions.
- Behavior: A good simulation of human-like behavior is highly desirable to create believable virtual humans. Behavior simulation is as complex as the human being, and it requires the knowledge from other disciplines such as Artificial Intelligence, Psychology, Philosophy, etc. To simulate higher level behaviors there are algorithms that drive low level motions in order to create coherent actions and capabilities. It is required that Virtual Humans are capable of perceiving, acting and reacting, and communicating in different modalities. Moreover, all these capabilities should include individualization to provide the same variation like in the real life.

Therefore, the main contribution of this work is the VHO, together with ontology-based application that demonstrate ontology's usability and reusability.

1.3.2 Ontology Usability and Reusability

The goal of the development of these applications is in one part to validate the ontology and to satisfy the user scenarios; but also, a second objective is to contribute to the state of the art of Virtual Human's development. Each application developed for this dissertation is ontology-driven and aims to prove that an ontological approach helps in the creation of believable Virtual Humans and population of 3D environments. The contribution of these application has being interest of the scientific community with the publication of articles in relevant conferences and journals.

The first application scenario is an application ontology that aims to formalize knowledge research in expressive facial [59] and body animation [58]. This ontology was developed in collaboration with experts in gesture recognition and synthesis. The knowledge domain are MPEG-4 facial and body animation concepts and their relationship with emotions through psychological models. This ontology reuses the VHO, and allows storing, indexing and retrieving prerecorded synthetic facial animations that can express a given emotion. Also this ontology can be used as refined knowledge base in regards to the emotional facial animation creation.

The other two applications aim to demonstrate how the VHO can contribute: to the creation of inhabited virtual environments, and to the creation of believable Virtual Humans.

The second application is focused on the creation of inhabited virtual environments [55]. The creation of virtual reality applications and 3D environments is a complex task that requires good programming skills and expertise in computer graphics and many other disciplines. The complexity increases when complex entities such as Human-like characters are integrated in the environment. This application assists in the tasks of setting up a 3D scene and configuring several parameters that affect the behavior of virtual entities (i.e. 3D objects and Virtual Humans). This application uses the visual programming paradigm and a semantic representation: the Virtual Human ontology. The ontology allows for storing and organizing the components of a 3D scene, together with the knowledge associated with them, and it is also used to expose functionalities of the 3D engine. Based on a formal representation of components, the proposed architecture provides a scalable 3D system. Using this system, non-experts can set up interactive scenarios with minimum effort; no programming skills or advanced knowledge are required.

The third application aims to improve realistic behavior of Virtual Humans [57]. One of the problems when creating believable Virtual Humans is the challenge of producing varied animations that are coherent with the context and with the individual traits of each character. A particular situation is when these characters are not able to perform individualized reactions to unexpected events in the environment. In order to reflect the individuality of each Virtual Human, these reactions have to be expressive and unique. This application presents firstly a model of reaction based on personality traits. The model was defined using statistical analysis of real people reacting to unexpected events. Moreover, it is considered that the emotional state intervenes in the modulation of reactions, thus a model of emotion update is integrated. Secondly, the application uses the Virtual Human ontology to store all individual information and to describe the generated reactive animation that is composed of several body part movements synthesized using different animation techniques: Inverse Kinematics and Key Frame. This application presents a scenario where Virtual Humans demonstrate how they can produce different reaction movements to unexpected stimuli, depending on their personality traits and emotional state.

Finally, some other contributions complement this work to facilitate the population of the VHO [56]. One system application and two software tools were developed to support the annotation process of 3D models and animation sequences.

1.4 Outline

This thesis is divided in three parts, organized as follow:

Part I Related Work . This part contains the state of the art on the subject of this thesis: Virtual

Humans. It also describes the approach proposed to represent the knowledge, and the respective related work.

Chapter 2 describes the different levels of knowledge of Virtual Human creation that go from the creation of their geometrical shape up to their conception as autonomous creatures.

Chapter 3 describes different approaches that aim at the representation of knowledge of related multimedia items, virtual environments and Virtual Humans; and presents the methodology used to create the presented ontologies.

Part II Semantics for Virtual Humans . This part firstly describes the context in which the development of this work took place. Secondly, it presents the ontology to describe Virtual Humans and gives some examples used to validate the proposed ontology design.

Chapter 4 overviews the AIM@SHAPE project and the ontologies developed therein. It also presents the Digital Shape Workbench infrastructure, created in the context of AIM@SHAPE NoE with the goal of supporting the storage, classification and retrieval of different resources such as 3D models, animations and software tools.

Chapter 5 describes the proposed semantics for the Virtual Human description. This semantics include the description of the Data Acquisition processes for the creation of a shape and motion captured animation sequences; and the Virtual Human Ontology (VHO) that presents the proposed semantics for each of the Virtual Humans's components presented in chapter 2. The validation examples of the VHO are also described in this chapter.

Part III Ontology Usability and Reusability . This part presents three applications developed to demonstrate the usability and reusability of the VHO. These applications aim also to contribute to the state of the art on the creation of virtual humans. Finally, conclusions and future perspectives are presented.

Chapter 6 presents an application ontology created on the top of the VHO. This ontology aims to represent specific knowledge of the synthesis of expressive body and facial animations using the MPEG-4 standard and models of emotion.

Chapter 7 presents an application that assist on the task of setting up a 3D scene populated with Virtual Humans. This application is supported by the ontology and uses the visual programming paradigm to overcome any deficiencies with the required programming skills when creating a 3D scene.

Chapter 8 presents the VHO used to improve Virtual Humans' behavior. This application aims to simulate spontaneous reactive behavior which is one of the open issues in the creation of believable Virtual Humans.

Chapter 9 summarizes this work, and presents the conclusions and future perspectives.

To complement this work, there are one appendixes that present related developments done for this dissertation (A), and two other appendixes (B and C) that aim to help the reader to familiarize with the notations and concepts used in this thesis.

Appendix A presents some tools developed to populate the VHO, as one of the main challenges to exploit ontologies is their population.

Appendix B presents a Glossary of Terms. This glossary is list of definitions of concepts relevant to the presented ontologies.

Appendix C overviews the Ontology Web Language to familiarize the reader with the terminology and diagrams presented in this dissertation.

Part I

Related Work

Chapter 2

Virtual Humans Creation

Creation of Virtual Worlds inhabited by believable Virtual Humans is a big challenge. Virtual characters should have a pleasant appearance, perform natural movements, be able to interact with users and with their environment, and exhibit intelligence. To be able to create such characters one should have knowledge from different fields such as: 3D modeling, computer graphics, physics, mechanics, artificial intelligence, etc. This chapter explains relevant components in the creation of Virtual Humans, as well as some current research issues. This state of the art aims to describe relevant information to create a semantic representation of Virtual Humans.

The main components that constitute a Virtual Human are described in the figure 2.1. This layered diagram illustrates specialized research topics with blue blocks, and human-like attributes used by those research topics in the yellow blocks.

The geometry is the visual representation of the Virtual Human, which indirectly provides morphological and anthropometrical information. The geometry includes an internal structure conforming a skeleton used to drive the geometry and animate the character. Besides, prerecorded motions are created to animate the geometry through a skeletal structure. Animations can also contain information about the morphology and the individuality that the motion can represent; for instance, the manner of walking, an emotional state, the gender that the animation belongs to, etc. To be able to modify prerecorded motions or create motions on the fly, motion control techniques are used. These motion controllers may or may not use prerecorded animations. There are also higher level controllers for behavioral animation. These controllers provide high level instructions to indicate the character what to do, when to do it and even why to do it. Behavioral controllers make use of lower level controllers to provide human-like capabilities.

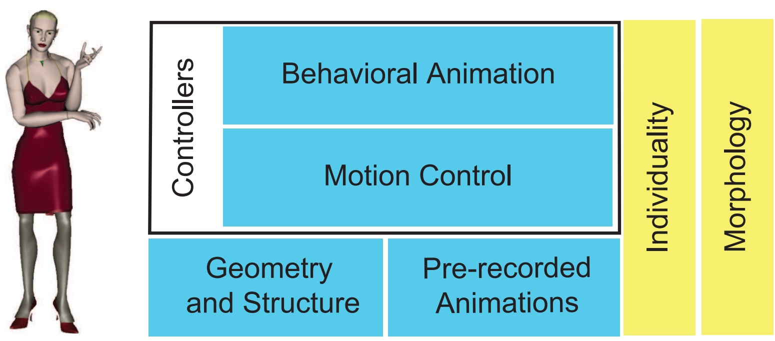

Figure 2.1: Levels of knowledge in Virtual Human creation

Each one of the presented elements contribute to the creation of Virtual Humans and help to conceive individual autonomous characters. The following section presents a bottom-up description of each of these components; trying to emphasize the information that is important to be kept during the creation process.

2.1 Geometry and Structure

The Human shape is very complex to be created from scratch. There are approaches that attempt to extract human shapes from pictures [133], [98] or videos [41]. These approaches make use of different media elements which information is not always kept nor well described. Thus, it is hard to compare methods or to know if some tricks were used.

More recent approaches are scanning technologies, which allow extracting 3D shapes from real people. Body scanning provides very high resolution models that need to be processed for its usage. From the scanning session is obtained a point claud which has to be meshed, merged, segmented, and skeletonized, to be able to animate it.

To automatize this process, some methods make use of specialized domain knowledge. These methods are more accurate because they prevent the surface from being erroneously corrected in under-sampled areas. A common approach is the use of models' templates used to fit the scanned data to the geometry of the template. Kähler et al. [76] proposed a method to reconstruct faces using template-to-data correspondence to easily identify landmarks. Allen et al. [5],[6] proposed to reconstruct bodies using correspondence markers attached to the scanned people. In [13], Ben Azouz et al. proposed an alternative approach for extracting the variations of the human shape from a 3-D anthropometric database using a volumetric representation of human shapes. More recently, Wang [92] described a feature-based parametrization of the human body for constructing mannequins from body scan data.

Part of the geometry description of Virtual Humans is also their internal structure, namely skeletal structure, body segments and feature points. These features are created to drive the geometry to animate the character. The extraction of these features from geometries is normally a manual process, however current research aims to extract them automatically from the geometry. Dellas et al. [45] proposed to extract joints location using priory knowledge of human anatomy.

With the increasing amount of Virtual Human models and animations, the need of reusing animations on different models motivated the standardization of the skeleton. The H-ANIM standard proposes *an abstract representation for modeling three dimensional human figures. This International Standard describes a standard way of representing humanoids that, when followed, will allow human figures created with modeling tools from one vendor to be animated using motion capture data and animation tools from another vendor* [74]. This standard is commonly used in the research field but not so much in the industry.

Presented reconstruction and processing techniques make use of a priori knowledge or formal standards, and only few approaches integrate semantic information related to the human shape reconstruction process. Moccozet et al. [106] proposed a full reconstruction pipeline that produces a close to animatable approximation from the scanned data. This method, presented by Seo et al. [136], is based on fitting a human template to the scanned data to produce the skin surface and the animation control information (skeleton). For this approach, anthropometry measurements and anatomical knowledge are important for the shape reconstruction. It is still almost impossible to completely automatize the creation of human shape without priori knowledge.

This section described the creation processes of virtual human's geometrical models that can be animated. The following section will present how animation of Virtual Humans is done using different animations techniques and motion control techniques.

2.2 Motion Synthesis

Motion synthesis of Virtual Humans is the simulation of human movement, where Virtual Humans are supposed to act and move like human beings in the most realistic possible way. Human movement simulation or animation can be tackled at different levels of abstraction, from skin deformation until behavioral simulation.

Motion primitives are basically mathematical transformations applied to articulations (joints) of the skeletal structure and translations of the 3D vertices of the mesh for deforming skin (skinning). Some of the difficulties to simulate realistic character movements are the high dimensionality of the skeletal structure (186 degrees of freedom), the the complexity of modeling joint limits, joint coupling, skin deformation, self collision avoidance, etc. The creation of these primitives can be computer generated, acquired from the movement of real people, or the combination of techniques. The following sections describe the different techniques to synthesize Virtual Humans motion.

2.2.1 Skinning

Skinning is the process of deforming the surface of the geometry according to the character's movements. There are several approaches used for skinning, and most of them are skeleton driven. The deformation of the geometry has a weight associated respect to the joint(s). This deformation can be computed using a function or interpolating predefined data of defined body postures.

Newer approaches are anatomy-based which offer greater realism than the traditional skeleton-driven. Some researches provide automatic creation of muscles from an already designed body [125],[169]. These approaches consist in segmenting the body and use priori knowledge of a muscles' positions to generate the geometrical muscles. Charbonnier et. al. [33], proposed an ontology for the simulation of the musculoskeletal system. This ontology provides information such the description of which muscles are in which bones, the elasticity of muscles, etc.

The skinning provides the transformations to properly deform the mesh according to skeletal animation. Next section describes the common animation techniques.

2.2.2 Animation Techniques

The easiest way of animating 3D characters is with the help of skilled animators. They manually define key postures of the articulated figure by setting positions and orientations of the joints and using interpolators to create the postures in between. This technique is called **Keyframe** animation. Some other techniques can be used to automatically produce the postures for the key frames.

However, the animation process using only key frames is very long, and it may not produce realistic results. Other popular technique is **Motion Capture**, which provides realistic results, but is very expensive in terms of resources. This technique consists in tracking the motion of a person using markers or sensors in his body and copying the movement into a 3D character. Motion Capture produces data frame by frame with positions and/or orientations of the markers with a very high frame rate producing a large amount of data. Some approaches extract key frames form motion capture sequences, which is useful for compression, retrieval, browse (summarization) and reuse [167].

During the motion capture session, exists relevant information that is not commonly registered, such as actor morphology, number of markers used and their set up position in the actor's body, setting up conditions (e.g. lights, environment), device configuration, etc. The collection of this information can be important for future processes.

There are different motion capture devices, and they can be classified in the following types:

- Optical systems: use track markers (light emitters) and several cameras sensitive to the light.

  - Optical Passive Markers: use markers coated with a Retro-reflective material to reflect light that is generated near the cameras lens.
  - Optical Active Marker: the markers themselves are powered to emit their own light, eg. Leds.
  - Optical Time modulated active marker: Active marker systems can further be refined by strobing one marker on at a time, or tracking multiple markers over time and modulating the amplitude or pulse width to provide marker ID.

- Inertial systems: based on miniature inertial sensors, biomechanical models and sensor fusion algorithms.
- Mechanical motion: uses an exoskeleton that directly track body joint angles
- Magnetic systems: calculate position and orientation with sensors that measure the low-frequency magnetic field generated by a transmitter source.

All these devices have advantages and disadvantages, however these technologies already allow for real time motion capture which is interesting for VR applications.

Another technique of motion data acquisition is video-based systems. The movement of the actor is extracted from the analysis of video. The video analysis is performed by a computer either from a pre-recorded video, or in real-time. This solution is costly in terms of computational resources, but avoids the use of markers on the actor and can lead to several other uses (for example shape recognition, gesture recognition). This solution is also widely used by people working on facial animation or motion synthesis.

Use of motion capture makes possible to create very large repositories of animations. The next sections present some research that deal with the organization of prerecorded animations.

2.2.3 Motion Databases

Motion capture is the animation technique that provides the most realistic results. However, motion acquisition process is very expensive in time and resources, as a consequence, the reuse of animations is highly desirable. To be able to reuse animations, they have to be described for the different needs. The Amoba system [65] proposes a database structure that allows to store and retrieve motion data using different classification schemes. They provide a very open structure for two types of classifications: movement properties and gesture descriptions. This approach requires the help of specialized recognition system to populate the database. The problem of using databases is that the system remains centralized and may be not easy to be scaled.

The description of an animation sequence can be done from very different perspectives depending on the features of interest: action, kinematics, posture, behavioral, emotional content, etc. There are some automatic classification approaches that provide low level features description. Ashraf and Wong [10] presented an analytical method to classify motions. They used low-level kinematic analysis and high-level knowledge-based analysis to create states that provide coherent snapshots of active body-parts during the motion. Assa et al. [11] proposed a method for action synopsis to automatically select representatives poses from a sequence. They decreased the number of dimensions of the movement and represented it in a motion curve to be able to analyze and extract the salient points in the curve. Kovar and Gleicher [85] used logical and numerical comparisons to build a continuous parameterized space of motions to compare animations and extract the similar ones. These types of classification allow for content-based comparisons or to query motions by example.

Other approach that allows the automatic classification of animation into specific actions is presented by Muller et al. [108]. Their method uses motion templates created using learning machine methodologies to identify types of motions using training sets. This allowed to classify motions like cartwheels, lying down, or throwing motions.

All presented approaches perform low level description of animations and make use of previous knowledge for their classification with higher level semantics. Therefore, the more extensively described the animations are, more reusability can be provided.

There are other approaches that deal with the creation of new animation transitions in real time using data bases. One solution introduced by Kovar et al. [86], proposed the construction of motion transition graphs. Motion graphs are built by analyzing a large amount of motion captured sequences. Nodes in the graph are sequences of frames extending between transitions called motion clips. The goal is to be able to make transitions from a clip i_k to j_m (i.e., clip i, frame k to clip j, frame m), where clips are similar enough to create an acceptable motion. Motion graphs have been mainly used to control directed locomotion [91]. The performance of motion graphs are constrained to the environment they are used in, and the desired reactivity capabilities, which tends to be low [132]. Motion graphs are an alternative way for describing animation. They can be used to define how animations are connected/related with each other, however, this more a pragmatic than a semantic approach.

Hence motion capture does not produces animations ready to use, some problems have to be solved such as animation retargeting, filtering and foot planting. Modifying animations is very difficult task and the quality of the motion can be drastically altered. The following section present motion control techniques to modify animations in real time.

2.2.4 Motion Control Techniques

As said before, the animation is basically composed of a set of rotations and some times translations of the joints and/or landmarks in the Virtual Human's skeletal structure. Animation control techniques can be classified depending on the method used to generate animation values:

Data-driven is the use of existing data to produce animations. Motion captured animations and key frame are data driven techniques.

Physics based can be Kinematical or Dynamical. Kinematics deals with physical properties of motion: position, velocity and acceleration. Dynamics are based on Newton's law of motion, relates the causes of motion (forces) taking in to account body's masses.

Hybrid techniques uses data and physics based approaches.

As mentioned in the previous section, motion capture provides satisfactory results, but the lack of control of motion sequences constrains their reusability. Several issues have to be solved to be able to reuse these animations. For example, the motion retargeting problem, which aims to apply an animation to a morphology that is different from the one originally created for. Gleicher et al. [61] stated that the length of the members is very influential on the generated movement. Kupla et al. [90] provided a model to solve morphologies problem by proposing a motion representation independent from the morphology. The following motion control techniques have been used to create procedural animations or to alter the animation sequences and create a new on on the fly.

The first physics based techniques are: Kinematical and Dynamical. Kinematics is divided in forward and inverse. Forward or Direct Kinematics (FK) provides directly the joint angle value. The problem of forward kinematics is to find the corresponding end effector's pose x given the joint positions q (to compute x from q). Meanwhile, Inverse Kinematics (IK) aims to find the corresponding joints positions q given the actual end effector pose x (computation of q from x).

Various solutions have been proposed for both problems. For Inverse Kinematics, analytical approaches like the one of Tolani et al.[150], which worked for a limited number of degrees of freedom. Kallmann [77] presented an special solution for human-like arms and legs, providing a search method for achieving postures avoiding joint limits and collisions. Alongside, there are numerical approaches that can handle more complex structures and constraints [16], [17]. An example of procedural animations solved solely using IK is locomotion control proposed by Boulic [23].

Other application for IK in motion control is the problem of locomotion adaptation of pre-recorded animations, known as *footskating*. This artifact occurs when character's foot slides (skates) when is supposed to remain planted. A solution was proposed using analytical IK methods [86] to adapt motion. Lee and Shin [96] presented a retargeting method to constrain different characteristics of the motion including footplants using IK. More recently, Glardon et al. [60] proposed a solution using numerical IK together with motion anticipation to be able to define footplants and compute the constrained motion.

The second physics based technique is the animation of articulated bodies with Dynamics. This technique uses the laws of physics to generate motion, and some of them use learning machine methodologies. The results are specialized controllers that are able to perform very specific forms of motions such as walking [94], running, cycling [73], acrobatics [165] and swimming [168]. Shapiro et al. [138] presented a physics-based animation controller toolkit; a good review on dynamic controllers can be found therein. Lee et al. [97] created a biomechanical-based human neck simulating the muscles structure to create hierarchical neuromuscular control model that mimics the relevant biological motor control mechanisms. This approach is different from the conventional joint-actuated skeletal models. Finally, hybrid strategies combine the use of motion capture or other data-driven kinematic motion with physical simulation. Zordan et al. [173] applied dynamic forces to a motion captured sequence to achieve physically realistic movements without loosing the details of the original motion.

The described procedural animation techniques offer a nice realistic solutions for character animation. However, it is very hard to create the infinite amount of controllers or motion captured sequences needed to provide all kind of motions humans beings are able to perform.

2.2.5 Facial Animation

Facial animation of Virtual Humans has a big impact in their realism as the face can encode more communicative cues than the body.

Facial animation has different approaches from body animation. Deformation models are used to achieve a realistic performance. These models are commonly parameterized with the Facial Action Coding System(FACS) developed by Paul Ekman and and Wallace Friesen in 1976 [48]. The FACS defines 46 Action Units (AU), and each AU describes an independent movement of a facial muscle or a set of muscles. This system was initially created to describe and distinguish facial movements; but it is widely used to synthesize facial expressions and articulated speech postures using deformation models.

In the animation field a standardization of the FACS has been proposed by the MPEG-4 [109]. This standard specifies 84 feature points on the neutral face, which provides spatial reference for the Facial Animation Parameters (FAPs) definition. The FAP set contains two high-level parameters, visemes and expressions. Facial shape is defined by the Facial Definition Parameters (FDP) and Facial Deformation Tables (FDT) nodes. FDP contains the feature points in the face shape and the FDT contains the deformation of the model face as a function of its animation. To animate the face, the FAP node has the translations of feature points, expressed in FAP Units; they can produce high quality facial expressions and speech articulation. FAPs that may be used together are represented in groups in order to facilitate the animation. This standard has widely applied in the research of recognition, analysis and synthesis. The MPEG-4 standard has served to produce a large number of facial expressions. Hartmann et al. [72] created a gesture engine to plan expressive behavior in a conversational agent. Raouzaiou et al. [130] presented a methodology to generate large combination of expressions to create intermediate ones.

However, the human facial modeling and animation is one of the most big challenges in computer animation. A recent survey in the field is presented in [49]. This survey divides in three kinds the facial animation:

- Simulation-based aims to recreate the anatomical muscle structure such as bones, muscles, skin and connective tissues [148]. Simulation of muscles movements has been presented as free form deformation [80] or spline models to simulate wrinkles [166] [153].

- Performance-driven copies features from real expressions performance and applies them into the model [115]. Guenter et al. [67] captured three-dimensional geometry and color and shading information to reconstruct models from videos. Vlasic et al. [155] mapped video recorded performances to different animatable faces.

- Shape-blend consists in the interpolation between existing model examples to deform the geometry. Blanz et al. [20] reconstructed morphable models from photos using Principal Component Analysis. Zhang et al. [170] used examples of expressions to obtain details of the deformation such as wrinkles.

There exist plausible results in facial animations, however to synthesize in real time realistic facial animations, and to be able to reuse animations, it is necessary to describe them and associate them to their corresponding interpretation (i.e. emotional state, gesture).

The next section describes the final process of the creation of credible Virtual Humans. The behavioral animation is on the top of the simulation of Virtual Humans.

2.3 Behavioral Animation

There are models/algorothms in Computer Animation that provide the basic movements of characters such as locomotion, facial animation, etc. Nevertheless, the control of why, when and what movement to perform belongs to the behavioral animation. This field of research aims not only to program characters that can perform tasks, but to create character that may simulate autonomy like human beings.

Autonomy in Virtual Humans is the conjugation of capabilities where low level motions are governed by higher level behaviors; for instance walking motion is governed by path planning which is governed by a motivation of the agent to go somewhere. Every behavior has its own model that is influenced by the internal state of the virtual human and constraints in the environment.

The contribution to the realism of behavioral animation is to create individual Virtual Humans that may be differentiated one from another, not only in their appearance and movements, but also in their behavior. To achieve this, there are some considerations to take into account:

- The behavior of a Virtual Human should be coherent with its appearance. Thus, a morphological description of the Virtual Human is considered to select the right animation and the right behavior. For example, movements of an elderly person are more slower that those from a young person.

- As all Virtual Humans should behave -act and react- differently, a constant internal configuration (personality) has to be provided in order to account for these differences.

- Over the time the behavior of the Virtual Human should vary, otherwise their behavior look repetitive. The same person may take different decisions to the same situation depending on their internal state (emotional state and mood).

Thus, considerations mentioned above, governs the different capabilities of the Virtual Human such as perception, action, reaction, motivation, learning, interaction, etc. The conjunction of these capabilities aim at creating autonomous virtual characters.

The rest of this section presents each of the described considerations to create behavioral animation: morphological descriptors, psychological models used to create behavioral models, and some examples of behavioral models.

2.3.1 Morphological Descriptors

Character's morphology is inherent to the geometrical shape. As described previously in the subsection 2.1, previous knowledge of the human shape, specifically anthropometric information is very useful for the shape reconstruction. Moreover, morphological information is needed for adapting motions form one body to another, but it is also needed to select the motion that better represents the physical appearance. Pronost et al. [126] presented an study that analyzes and synthesizes locomotion of different morphologies to synthesize locomotion of early hominids. Another example is the motion synthesis of handicapped people [3], where it is expected that their physical abilities change accordingly.

Anthropometric descriptors are one part of the individualization of Virtual Human movement. To create more realistic behaviors in a face to face scenario, it is necessary to consider other morphological aspects such as gender, age or cultural identification. Integration of morphological aspects of Virtual Humans has been considered in locomotion synthesis. Troje et al. [151] performed studies in the perception of locomotion of different genders. Le et al. [95] integrated age attributes to the locomotion considering age-related changes in gait pattern.

Integration of morphological descriptors in Virtual Humans is still open research. However, they are important for creating animations that correspond to the physical appearance of Virtual Humans.

2.3.2 Individuality Descriptors

Individual behavior results from the interaction of three components [124]: (1) personality-based predispositions for behavior, (2) moods/emotions and (3) environmental situations. Personality can be considered as a consistent state over a long period of time, while emotions and moods are short and inconsistent.

The design of a model that translates these individual traits in behavioral cues is a big challenge. In real situations people think and react differently, and this depends on a numerous factors (perception, experience, interpretation, context, cultural background, etc). Moreover, the range of possible behavioral expressions is huge and still unmanageable with existing character animation techniques.

Representation of individual traits commonly uses psychological models. The creation of these models are mainly based on empirical studies that associate actions to personalities, or types of movements to emotional states. The most popular psychological model used to simulate individual traits are described in the following.

Personality Models. There are several psychological models that describe humans'personality, but not all these models are suitable for computational applications. Computational models provide commonly dimensional representations that go from three up to seven dimensions. A well known model is the Big three model [50], which relies on biological basis of socialization-psychoticism introversion-extraversion and neuroticism-stability to describe personality. However, a competing model, the Five Factor Model [105] is the most popular used in computational models.

Emotion Models. To represent emotions, there are also several models that have different objectives, e.g. Eckman's for face expression [46], OCC [114] largely used for emotion synthesis, Whissel's activation-evaluation space [163] to represent emotions in a continuous space, etc. These models are readily amenable to the intentional stance, and are ideally suited to the task of creating concrete representations of personality and emotions with which to enhance the illusion of believability in computer characters.

Thus, these models are to represent individual traits, and they are used to create behavioral models.

2.3.3 Behavioral Models

Models of behavior are used to achieve individual conduct and simulate autonomy and intelligence. A representative example of a semi-autonomous Virtual Humans is the successful video game "The Sims" by Will Wright (1998). The interest in this game is kept due to the dynamics of the character's individuality. Personalities are chosen to fit the role you want your *Sim* to play. This system uses personality descriptors (outgoing, playful, neat, active and nice) which drive the character's behavior. The character is able to simulate an autonomy that is ruled by its personality, emotional state and motivations, but users can alter the *Sim*'s life to try to make them successful in their lives.

As mentioned above, most of the systems that integrate models of behavior include detailed models of personality and emotions [7]. The state of the art in models and methods that cover models of emotion and personality generating expressions and behaviors can be found in [154].

Some models deal with the computational representation of internal parameters of agents. Tanguy et al. [145] proposed a model to simulate the dynamics of emotions, but does not consider that personality can influence the emotional state. Kshirsagar created a model of personality and emotions that synthesizes conversational characters with expressive animations [88], [87]. Marcella and Gratch [102] described how characters can copy emotions and change beliefs from appraisal mechanisms.

Other models are intended to reflect long term behavior. Silva et al. [139] presented a rule-based model for decision making that takes into account individual traits and internal states. Badler et al. [15] presented a parameterized action representation model that allows an agent to act, plan, and reason about its own actions or others' actions. Servin et al. [43] presented a hierarchical model that choose an action depending on the internal variables (needs), motivations and goals.

Finally , simulation of cognition have achieved plausible results in the modeling of learning behaviors of creatures or animals [140], [21], [24]. However, modeling cognition of Human beings is much more complex due to their multiple modalities of interaction. Conde et al. [36], [37] presented a model that aims to provide perceptive behavior to Virtual Humans and learning from the environment.

Summarizing, there are numerous models of behavior that are focused in different capabilities to provide autonomy to Virtual Humans. These models are basically algorithms that control the animation of the character at high level. However, there is still a lot of open issues in behavioral animation.

2.4 Conclusions

This chapter presented some current practices of the creation of realistic Virtual Humans. Within this creation process the relevant knowledge that should be preserved involves:

- Creation history of human 3D shapes with control animation structure: control skeleton and skin binding.
- Integrating knowledge about human shapes, not only relying on the geometric data, but also morphology, anthropometry, individuality descriptors.
- Description of resources, features or tools that help Virtual Humans to accomplish the capabilities for making them autonomous: animations, controllers, behavioral algorithms, etc.

In common practice, the different processes in the creation pipeline do not share information that can be useful for each other. There is a lack of a common framework of Virtual Human creation that allows faster production within different application domains.

This work proposes to organize this information to facilitate the creation of Virtual Humans. Addressing the first research question of this work: *How to describe Virtual Human components using common vocabulary or simplified representations*. Next part explores the existing possibilities to formalize knowledge, and some of formalization approaches that are related with this work.

Chapter 3

Knowledge Representation

Knowledge Representation is a branch of artificial intelligence that aims to analyze knowledge of the world and map it to a computable form. According to Sowa, *"knowledge representation is a multidisciplinary subject that applies theories and techniques from three other fields: (1)* ***Logic*** *provides the formal structure and rules of inference, (2)* ***Ontology*** *defines the kinds of things that exist in the application domains and (3)* ***Computation*** *supports the applications that distinguish knowledge representation from pure philosophy ... Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain"* [141].

A formal representation of knowledge should be understandable by humans, and machines should be able to process it, so they can behave as if they knew. One may have to use a programming language for the computers to process the information, but not all languages fulfill the needs of Knowledge Representation described before. Depending on the language used, it is possible to apply logical operations (reasoning), but sometimes not all programming languages are easy to understand by humans.

This chapter is focused firstly on the problem of how to represent knowledge that can be processed by machines using logical reasoning. Secondly, some specific languages are presented along with the related work that uses them to describe some aspects of Virtual Humans. Finally, this chapter describes the methodology used to develop the ontologies presented in this thesis.

3.1 Background

The definition of **Ontology** comes originally from the philosophy, and means *a systematic explanation of being*. In computer science, ontologies describe the formalization of a knowledge. The most popular definition is the one given by Gruber [64]: *An ontology is an explicit specification of a conceptualization.* Others definitions include that ontologies are the product of a common agreement of a group or community of people about concepts, taxonomies, relation between concepts, properties that describe the concepts, and axioms and constraints.

There are different approaches for modeling ontologies such as software engineering technologies (UML), database technologies (Entity-Relationship diagrams), and Artificial Intelligence based techniques (Description Logics, Frames and First Order Logics) [62]. The formalism adopted to model an ontology constrains the way of defining concepts and constraining relationships between concepts. It also restricts the possible software implementations and the reuse of represented knowledge.

An advantage of software engineering techniques (e.g. UML) is that the graphical representation of objects allows to non experts in informatics to model knowledge, and it also allows different levels of abstraction of components in the final system. In data base modeling it is possible to represent concepts with entities and attributes. It is also possible to describe simple axioms using integrity constraints.

The main drawback of these two techniques is that formal representations of axioms and constraints are limited, thus this prevents from creating formal semantics.

Artificial Intelligence (AI) techniques used to model ontologies are Frames and First Order Logics, and Description Logics (DL). *Frames* represents concepts described through predicates (classes, properties and relations). Using frames it is possible to declare axioms to define rules that restricts relationships in the knowledge base. DL functions using frames as well. It defines *roles* to describe binary relations between concepts. In these kinds of representations it is possible to create axioms and make inferences with them, e.g.: *If any two animals are siblings*, it is possible to infer that *there exists someone who is the mother of both of them.*

Summarizing, mentioned techniques allow representing concepts, taxonomies, relationships between concepts, and properties to describe concepts. However, the integration of constraints and axioms differ form one technique to another; and these axioms and constraints are important to clarify the meaning of concepts. The AI-based ontologies are specialized in knowledge representation and provide the most complete components to create formal semantics.

Current web technologies support the creation of AI-based representations using markup languages. The example of this application is the Semantic Web, which is indeed an extension of the current web; but the information is given in a well-formed meaning, enabling computers and people to work in cooperation using shared knowledge components [18]. Ontologies are an essential component in the Semantic Web.

The special interest in AI based formalisms is that there are techniques and languages used as standards in the Semantic Web [147] such as XML, RDF, RDFS, OWL, etc. These languages are good candidates for representing and implementing ontologies. In life sciences, the adoption of ontologies has proven that it is possible to successfully formalize the knowledge of complex entities and data sets in shared conceptualization. In any case the adoption of standards is important to allow the extension and interpretability of ontologies [137].

The following section presents an overview of these representation languages.

3.2 Representation Languages

The Semantic Web principles are implemented in the layers of Web technologies and standards, presented in figure 3.1. In the lower level layer there are the Unicode and Uniform Resource Identifiers (URI), which make sure to use international characters sets and provide the means for identifying named resources in Semantic Web. The XML layer has the name-space and schema definitions that integrate the Semantic Web definitions with the other XML based standards. The RDF [158] and RDF-Schema [159] layer is to make statements about objects with URI's and define vocabularies that can be referred by URI's. This layer allows to define data types of the resources. The Ontology layer supports the evolution of vocabularies because it can define relations between the different concepts. The upper layers: Logic, Proof and Trust, are currently an open issue. The Logic layer enables to define rules while the Proof layer executes the rules and evaluates them together with the Trust layer mechanism, which defines whether or not to trust the given proof. Finally, the Digital Signature layer is for detecting alterations to documents. These are the layers that are currently standardized in the World Wide Web Consortium (W3C) working groups[82].

The Resource Description Framework Schema (RDF-S) [159] is a simple modeling language on top of the Resource Description Framework (RDF) formalism, both W3C recommendations. RDFS offers primitives for defining knowledge models that are close to frame-based approaches. RDFS introduces classes, subsumption relationships on both classes and properties and global domain and range restrictions for properties as modeling primitives. The resulting formalism allows to model ontologies as a taxonomic structure of concepts with attributes and relations to other concepts as properties attached to each concept.

However, RDFS misses some features that are commonly found in systems developed by the AI

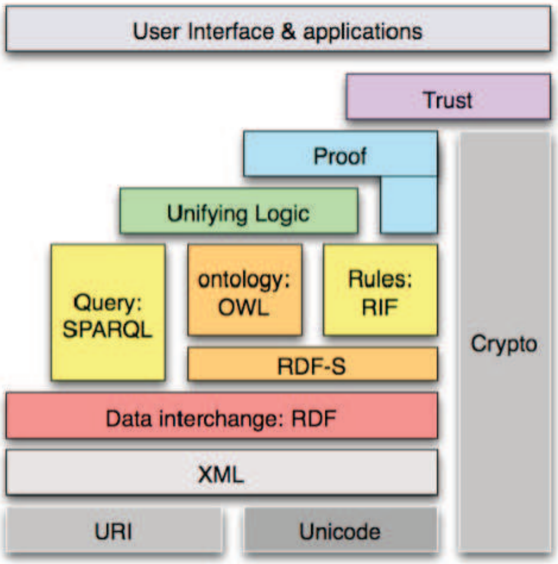

Figure 3.1: Semantic Web Layers

community (e.g. frame-based systems, description logics). It contains some features that makes hard to provide formal semantics for RDF-S, which prevents the creation of automatic and efficient inference engines. In the ontology layer, DAML+OIL [156] addresses this issue by extending RDF-S with concepts commonly found in frame-based languages and description logics. The result is a language that is compliant with RDF and RDF-S. It provides definitions to create formal semantics and efficient inference engines. This allows more advanced querying and the inference engines can also be used to detect contradictions and other errors in a DAML+OIL specification. DAML+OIL is currently used by the W3C Web-Ontology (WebOnt) and it was a starting point for a W3C Ontology Web Language [159] (OWL).

The Ontology Web Language (OWL) is a language inspired by description logics and also developed within W3C. It is designed to be used by applications that need increased expressive power compared to that supported by RDFS. It provides additional vocabulary along with formal semantics. This language is created upon Resource Definition Framework (RDF). OWL is a more expressive language than RFD, it provides relations between concepts in a logic way. What makes a representation language appropriate or not is its expressive power and its support for reasoning.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users:

OWL Lite supports those applications that primarily need a classification hierarchy and simple constraints. It supports cardinality constraints, but only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives. OWL Lite provides a quick migration path for thesauri and other taxonomies.

OWL DL (Description Language) supports those applications that need the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. In OWL Full, a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. However, it is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

These web technologies provide well formed sharable structure using the power of markup languages, they fulfill the requirements to represent knowledge at different levels.

In the computer graphics field, there are attempts to standardize and formalize different aspects of Virtual Humans using the described web technologies. Next section presents the related work relevant in multimedia, virtual environments and Virtual Humans.

3.3 Related work

This section describes firstly approaches that use XML to represent specialized capabilities of Virtual Humans. Secondly, it presents some approaches that propose ontological-based representations for multimedia items. Thirdly, conceptual representations of virtual environments are also presented.

3.3.1 Metamodels: Extensible Markup Languages

XML is used to define a grammar to represent concepts at lexical level [44]. This makes possible to reuse a document or interpret its content using a parser and its definition schema or DTD. Within the studies of the creation of virtual humans, several XML representations have been proposed for specific cases.

Several markup languages have being created to code communicative behaviors (verbal and non-verbal). The Character Mark-up Language [8] proposes to link available engines for generating and controlling believable behavior of animated agents with the corresponding animated representations. The Avatar Markup Language [89] aims at encapsulating the Text to Speech, Facial Animation and Body Animation in a unified manner with appropriate synchronization. De Carolis et al. [28] developed a XML interface to communicate the Mind with Body of an avatar. They proposed to regulate body signals (posture, gaze, speech, etc.) accordingly with agent's internal state using an Affective Presentation Markup Language and a behavior planner called Discourse Plan Markup Language. The Behavior Markup Language [84] represents agent's communicative behaviors in different modalities, and the synchronization of them. It describes attributes of the visual appearance and movement dynamics of expressive effects.

A general descriptive language is the Human Markup Language [116] (HumanML). This language describes characteristics of artificial humans as well as real humans, enabling real-time animated behaviors for 3D representations of humans. HumanML includes basically simple types, such as age, gender or physical descriptors (weight, hair color, eye color) and complex types such as address, human artifact (clothes, jewels), belief, human communication channel, community as an abstract human organization, human culture, human emotion, geolocator, haptic (defined as the strength, location, and body part used in a touching behavior), human intent, kinesic (Human Movements), human personality type, etc.

The Virtual Human Markup Language VHML [101] is intended to be used for controlling Virtual Humans' speech, facial animation, facial gestures and body animation. Authors proposed a XML language that is composed by specialized markup languages for each modality: DMML Dialogue Manager Markup Language, AIML, FAML Facial Animation Markup Language, BAML Body Animation Markup Language, SML Speech Markup Language, EML Emotion Markup Language.

These XML languages offer a description to represent concepts. They are adequate to annotate and to provide a standardization of terms, or to script control sequences. However, the context (meta-information) is not provided. In complex applications, this kind of description risks to be confusing and wrongly interpreted.

XML descriptions have been proposed to provide meta information about digital items' content, but it still cannot be considered as formal semantics. Next subsection describes some ontologies to represent multimedia content and specific application domains.

3.3.2 Multimedia Ontologies

Multimedia is a generalization of different types of communication media that concerns also 3D environments, and as a consequence Virtual Humans. Semantic description of multimedia items has been mainly developed for audio, video and still images. Domain specific ontologies are focused on describing the content and the parts of a multimedia scenario (e.g. elements in a scene, colors, motion duration, etc.). These descriptions are defined in order to categorize, retrieve and reuse multimedia elements. Examples of domain-specific ontologies involves a wide set of applications from Cultural Heritage [143] to Biomedicine [30].

Most of these kinds of ontologies deal with content description, and make complete or partial use of the MPEG-7 standard [107]. The MPEG-7 standard, formally named Multimedia Content Description Interface, provides a rich set of standardized tools to describe multimedia content (still pictures, graphics, 3D models, audio, speech, video, and composition information). It regards how different elements are combined in a multimedia presentation independent of storage, coding, display, transmission, medium, or technology. Furthermore, MPEG-7 has an ontology defined which has a general meta-data description [75]. The Visual Descriptors Ontology [119] aims to offer a more extensive description of the visual part of MPEG-7. It is primarily focussed to support automatic content annotation using reasoning and providing access to specific domains. The Core Ontology for Multimedia (COMM) [9] is another ontology that extends MPEG-7 and provides richer multimedia semantics by using generic software patterns which create a layer between MPEG-7 concepts and domain-specific interpretations. Currently, there are efforts towards a generalized multimedia ontology [111], which aims to unify concepts among domain specific and top-level ontologies.

Multimedia Ontologies are still too general to cover the description of multimedia elements targeted in this work, only small attention has been given to 3D items. OntologyX3D [79] is a dedicated 3D ontology mapped from the X3D standard. It represents graphic elements and virtual reality concepts, and provides concepts for the scene graph. This ontology could be considered as an upper ontology of the ontologies here presented.

Inside virtual environments the 3D items represent more than a geometry. They can contain information about its usage, functionality, interactions, etc. The following chapter describes the semantics that have being developed for Virtual environments, which deals more with the meaning of entities inside a virtual world.

3.3.3 Semantic Virtual Environments

The work done towards a knowledge representation of 3D environments has remained mostly at geometric level. Tijerino et al. [149] presented the very first approach to represent 3D geometries using two ontologies. The objective was to help people to describe mental images through natural language combined with hand gestures. Recently, a more sophisticated description was presented by Bille [19]. This work is focused on the representation of simple geometries like spheres, and basic transformations and interactions. However, to be able to create complex entities such as Virtual Humans the information requires higher level of abstraction.

A more generic representation of virtual environments was proposed by Gutiérrez [68]. This approach proposed a model to define digital items as components that conform the virtual scene (agents, objects, interfaces, etc). This model describes different properties of such items: geometric, semantic and controllers. The objective of this representation is to facilitate the reusability and the scalability of virtual environments. This thesis provided a very first version of an Ontology for Virtual Humans by creating a list of Competency Questions and an initial ontology that covered the basic components of the Virtual Human. However, that structure was still in a early stage because any validation or application scenario was proposed.

For the creation of any ontology, a methodology has to be used to ensure the requirements of common agrement among a group of experts. Next section describes the methodology followed to create the ontology presented in this dissertation. In each development stage, a summary of results achieved during each phase of the methodology.

3.4 Ontology Development Methodology

The development of ontologies is sometimes guided by different methodologies, but it does not exist a perfect formula. There are several proposed methodologies for ontology development, however, the use of a methodology does not guarantee a good representation of the knowledge. What makes a representation of knowledge to become an ontology is the common agreement among a group of people in the definition of concepts of a domain. For the development of the **Virtual Human Ontology** (VHO) presented in this thesis, and other the ontologies described in next chapter, the collaboration between experts in different domains took place in the framework of the NoE AIM@SHAPE [112]. The methodology used is the On-To-Knowledge [142]. This methodology defines four iterative stages, depicted in figure 3.2: Ontology Kickoff, Refinement, Evaluation and Maintenance and Evolution.

The following subsections describe the principles of each stage and summarize the results obtained during the development of the VHO.

3.4.1 Ontology Kickoff

Ontology Kickoff identifies the domain of the ontology, some usage scenarios and knowledge sources (domain experts, glossaries and dictionaries, etc.). It also defines basic questions (competency questions) that the ontology should be able to answer, and identifies potential users and applications.

Thus, the domain of the *VHO is the description of Virtual Humans to support their creation process and the population of virtual environments.* The users of this ontology may be experts that want to specialize the ontology to represent more complex definitions, or non experts that need a higher level of abstraction of Virtual Human's features.

The definition of user scenarios is a very important task because it contains the core data of the ontology. Some examples of the scenarios provided for the VHO are:

Modeling Data Repository Scenario: A place where a designer and animator could find human shapes and use them to model new Virtual Human, or to improve or reconstruct existing ones. The ontology could help to manage an online shop to build custom 3D mannequin and select and fit clothes and accessories for visualizing on the mannequin. It means that it will help to know the suitable clothes and accessories to a specific Virtual Human by matching the corresponding landmarks.

Shape recognition, extraction and analysis: A knowledge base able to answer queries about low level features of the Virtual Human shape (landmarks, topological graphs, and so on). Users are researchers working on algorithms for recognizing features on a shape representing a virtual/real human. Data would be used on ergonomics studies, computer vision algorithms, etc.

Virtual Environment Knowledge Base: The ontology will serve as support for the population of virtual environments, where all information about virtual object as and virtual humans can be extracted in real time for the population of an environment. This means that it would serve as a database but also as a rule database of possible simulation scenarios.

From scenarios collected it is possible to acquire important concepts and relationships for the construction of the ontology. For example, in the second scenario, the word landmark could represent one

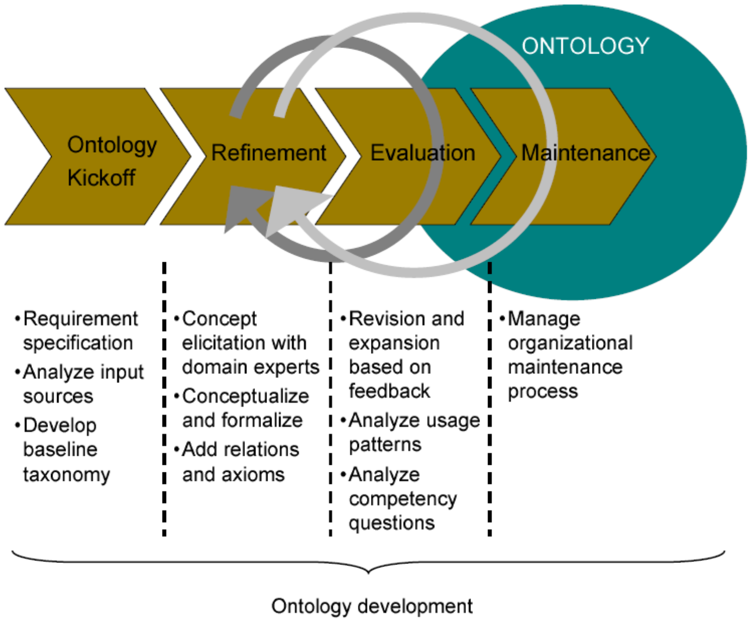

Figure 3.2: On-To-Knowledge methodology process

concept that links a cloth and a virtual human.

Apart from the scenarios, other data source are the competency questions. These questions serve also to validate the ontology because it should be able to answer directly or indirectly these questions. Examples of these questions are: What is the mesh resolution of the Virtual Human shape? What is the level of articulation of the Virtual Human model and is it able to perform a specific given sequence of motion? What is the size, gender, or weight of a Virtual Human model? What are the algorithms to simulate emotional state changing for Virtual Humans? What models of psychological model of emotions it uses? Some other questions were already suggested in [68].

With scenarios and competency questions it is possible the identification of concepts, synonyms, classifications, possible hierarchies, etc. The important concepts are part of a vocabulary used here. A glossary of these concepts is presented in the Appendix B.

Taking into account that Virtual Human creation involves other domains, such as the 3D geometries, software components, shape acquisition, modeling, etc., it should be considered the reuse of other upper and domain ontologies that specialize in the mentioned domains. At this stage, some common knowledge was identified between related domains and specific knowledge of specific domains.

3.4.2 Refinement

The refinement phase is to choose an approach to represent the knowledge in which the ontology is going to be encoded in a formal manner.

The goal of this work is was provide an explicit description of Virtual Human's characteristics and features. The representation proposed is an axiomatic one rather than an operational or procedural one. The VHO does not represent how a rotation of a joint is performed, but it represents that a given rotation belongs to a set of transformations that belong to an animation that has higher level descriptors (i.e. walking animation sequence).

Therefore, the technology used to implement these ontologies was the Ontology Web Language (OWL) [157]. This formal language offers more ways of entity relation, cardinality relation between entities and more others logical operations. Protégé software[127] was used to implement OWL. This open source is an ontology editor and knowledge-base framework. The Racer [129] software was used as reasoner to query and to make inferences in the ontology.

The ontology design process is based on the creation of classes to encapsulate concepts. When defining classes in OWL, each one receives an unique identifier (URI). The classes specify the concept of a "thing", but they are not a thing by themselves. Classes are instantiated to create "things", called Individuals. Relations between concepts (classes) are represented by properties. Those properties describe the kind of associations between classes. Relations can also specify some restrictions as quantifier, cardinality or value restrictions. A detailed description of OWL and the notation used in the following chapters is presented in the Appendix C.

3.4.3 Evaluation

The evaluation of the ontology consist in assessing if the expected requirements are satisfied. Usage scenarios and competency questions are used as reference.

The first way of performing simple evaluations was populating the ontologies and making queries. For the evaluation of the VHO, is was used different query languages such as SPARQL [161] and nRQL [110], both provided within Protégé software.

In this phase, it was noticed that the ontology population is very tedious task because it was performed manually. To overcome this problem, automatic tools were developed to extract meta-data automatically and semi-automatically. These tools are presented in the Appendix A.

With the population of the VHO, it was possible to perform more complicated queries using a specialized search engine tool developed by Vasilakis et al.[152]. This tool creates queries in nRQL [110] language, and uses Racer [129] to perform the reasoning in the ontology.

A second way of evaluating the usability of the VHO was developing applications that use it. These applications are related with the usage scenarios, and they are presented in the Part III of this thesis.

3.4.4 Maintenance and Evolution

The ontology development is a cyclic process because some designs may be not the most optimal ones. The maintenance and evolution stage consists in modifying the ontology to add new concepts or to satisfy competency questions.

The design of the VHO was performed incrementally by the definition of evolution steps. Each evolution step had as objective to describe specific aspects of Virtual Humans such as the description of a shape, structure, data acquisition (Shape and Mocap), animation parameters, animation controllers, etc. During an evolution step the structure of the ontology was modified to add concepts and/or refine existing ones. When an evolution step was completed the evaluation of the design was performed.

3.5 Conclusions

This chapter presented an overview of different methodologies used to represent knowledge. Artificial Intelligence techniques are the most adequate to define concepts and axioms. Moreover, they are considered in the Semantic Web representation languages. Some web languages have been used to represent some aspects of Virtual Humans using basic representations such as the XML. Besides, multimedia ontologies describe some geometrical aspects of the 3D shapes. However, the aim of this work is to represent not only geometrical aspects, but also semantic aspects to support the creation of Virtual Humans and description of their features to populate virtual environments.

The development of the VHO followed the Onto-To-Knowledge methodology described in this chapter. The creation of an ontology requires the collaboration of a group of experts in the domain to agree in the knowledge to represent. In the framework of AIM@SHAPE project different ontologies were developed, including the VHO. The development of the VHO was leaded by this dissertation. Next chapter presets in detail the ontologies created within the AIM@SHAPE.

Part II

Semantics for Virtual Humans

Chapter 4

AIM@SHAPE Project

To exploit the potential of shapes, additional information along with geometrical information should be provided. New research in the creation of digital shapes is slowed down by the need to master or integrate many different theoretical and technological aspects. To overcome this problem the Network of Excellence AIM@SHAPE [112] proposed a framework that aims to integrate semantic description in the shape development. This enhances the value and potential for general reuse of contained shape models and allows the stored knowledge to be retrieved, processed, shared, and exploited to construct new knowledge [51].

AIM@SHAPE's objective was to create a semantic-based representation of digital shapes with the formalization of shape knowledge and the definition of shape ontologies in specific contexts [51]: Virtual Humans, Shape Acquisition and Processing and Product Design.

This project created an e-Science framework called the Digital Shape Workbench (DSW)[1], which is a common environment to share knowledge about shape models and software tools. This framework is a common infrastructure for integrating, combining, adapting, and enhancing existing and new software tools and shape databases. The development of the DSW was carried together with this dissertation, and this application demonstrates the ontology usage inside a web-based repository of geometric shapes and their related knowledge.

4.1 DSW Framework

Figure 4.1 depicts the architecture of the DSW, which is composed of the following components:

Repositories. Shape Repository (SR) and Tools Repositories (TR) store digital resources in the DSW. Besides providing shapes and software tools, they provide a complete descriptive information about them, which is provided by the corresponding ontology (CSO and CTO).

Ontology and Metadata Repository. The Ontology and meta-data Repository (OMR) is the knowledge base back-end, which contains domain-specific and general knowledge related to shapes in terms of instances of the AIM@SHAPE ontology classes. The OMR is an ontology management system that allows storing, editing and accessing ontologies and ontology-driven metadata. The OMR is closely related to the semantic search engine: the expressiveness and flexibility of the search is constrained by the way the information is stored in the repository and the logical formalism that is used to describe this information.

Search Framework The Search Framework involves both a semantic and a geometric approach:

[1] http://dsw.aimatshape.net/

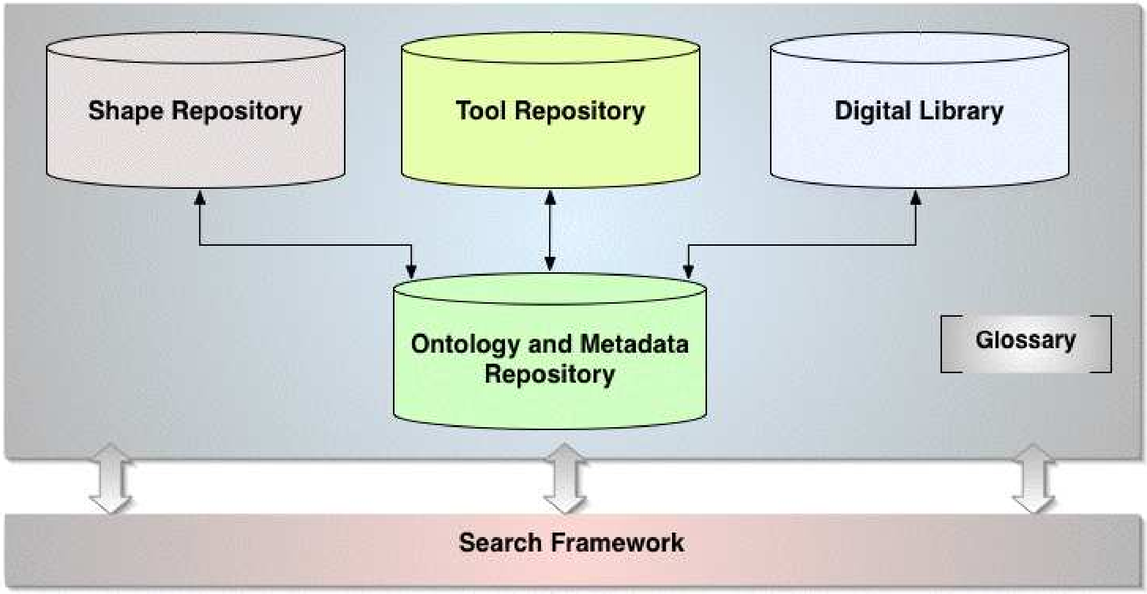

Figure 4.1: Digital Shape Workbench Architecture

Semantic Search Engine. The Semantic Search Engine (SSE) is used for simple search of resources uploaded into the OMR. It also provides implicit knowledge depending on the available resources, by using semantic criteria and an inference engine. Explicit information is stored in the OMR in terms of resources and their metadata, while implicit information can be deduced on the basis of the relationships and rules induced by the ontologies. The inference engine is able to search the SR, the TR and the DL and to answer any query regarding those repositories in a transparent way [152].

Geometric Search Engine. The Geometric Search Engine (GSE) provides content-based 3D shape retrieval mechanisms according to different similarity criteria and matching methods (e.g. global and partial matching, sub-part correspondence, part-in-whole). The GSE adds to the DSW e-Science platform the possibility to act as a 3D retrieval benchmarking system [100].

Digital Library. The aim of the Digital Library is to create a common repository of scientific references and technical reports. It is a reference for publications on the topics of shape modeling, computer graphics, semantics, etc. Each item in the Digital Library is described by meta-data. References are avilable in the most common bibliographic formats, such as Bibtex or RIS.

Glossary. The Glossary aims at the definition of a common vocabulary which will be used by the Consortium to deal with digital shapes. The list of identified terms represents a meaningful and commonly understood set of terms describing different aspects of digital shapes.

Therefore, in the DSW it is possible to upload, search and retrieve geometrical shapes and software tools. These resources are annotated using concepts in the ontologies created for their semantic description. The ontologies were created using the methodology presented in section 3.4, and they are described in the following section.

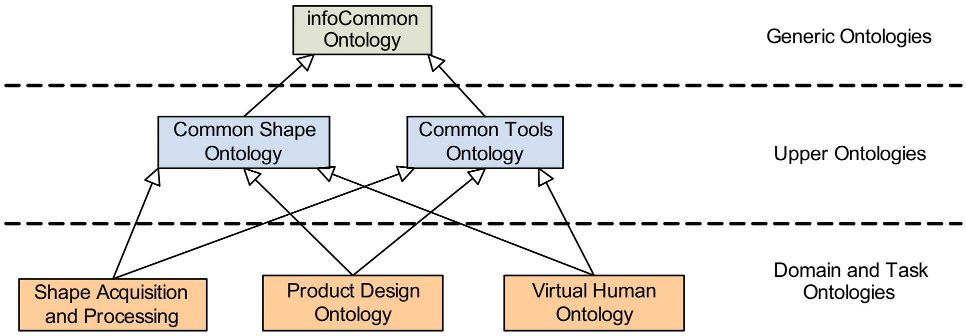

Figure 4.2: Developed ontologies

4.2 AIM@SHAPE Ontologies

The ontologies presented in AIM@SHAPE NoE cover different needs of scenarios and competency questions. To describe each ontology and the relation between them, consider that ontologies can be classified according with their subject of conceptualization [66]:

Generic Ontologies involve concepts or general knowledge are described. These ontologies are highly reusable.

Top-level or Upper Ontologies describe general concepts and provide general notations.

Domain Ontologies provide vocabularies and relationships within a given domain, including activities taking place in it. Concepts in domain ontologies are specialization of those already defined in top level ontologies.

Task Ontologies describe a vocabulary related to a specific task or activity, including the sequences of these tasks.

Application Ontologies contain all information needed for a particular application.

Following this classification, figure 4.2 presents how AIM@SAPE ontologies are related.

A generic ontology called **infoCommon Ontology** is used to describe information about persons and institutions that have the authoring of the digital items described in the upper ontologies.

Top level ontologies are **Common Shape Ontology** (CSO) and **Common Tools Ontology** (CTO). They are dedicated to describe the geometrical aspects of shapes and software tools. The final instances (resources) described in these ontologies are referred by the domain ontologies. These domain ontologies describe the creation processes of shapes, or their features in a more dedicated manner inside the domain ontologies.

The domain ontology **Shape Acquisition and Processing Ontology** (SAP) conceptualizes 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. The fundamental goal of the ontology is to formalize the knowledge related to the Acquisition and Processing of a Shape [4].

The **Product Design Ontology** (PDO) described in detail in [29], focuses on the description of two important phases of the product development processes: styling and simulation. PDO formalizes in a ontology the description of processes, tools and knowledge relevant to the product development, i.e. the free-form styling and the engineering simulation.

Finally, the **Virtual Human Ontology** (VHO) is related to the description of complex 3D entities such as virtual humans, not only at the geometric level, but also at the structural and semantic level.

The goal of this description is to formalize the composition of virtual humans and to facilitate sharing of useful information by experts and non-experts in the domain, in order to promote reusability and scalability [69].

OWL [157] was used to formalize AIM@SHAPE's ontologies, and an extension to support the Semantic Web Rule Language (SWRL) [160] was also developed. RacerPro [129] was used to provide reasoning on the knowledge base.

The rest of this chapter describes each of the mentioned ontologies, paying special attention to those descriptions that are relevant to the Virtual Humans. A detailed description of this notation together with an overview of OWL is given in the Appendix C.

4.2.1 Upper Ontologies

One important recommendation when creating ontologies is to reuse upper ontologies for general conceptualizations. This is to avoid redefinition of concepts and over working. It is also important to try to avoid descriptions inside a domain ontology that are not so relevant for this domain or belongs to a more general conceptualization.

In this work, one general ontology and two upper ontologies were created to be reused by domain ontologies. These top level ontologies are described in the next section.

4.2.1.1 Info-Common Ontology

This small ontology captures information about persons and institutions that are authors or own the different resources (i.e. 3D models, algorithms, libraries). Figure 4.3 presents the ontology diagram with two concepts: *PersonInfo* and *InstitutionInfo*. Each concept has attributes that describe them. A relationship between *PersonInfo* and *InstitutionInfo* is given by the *worksFor* property.

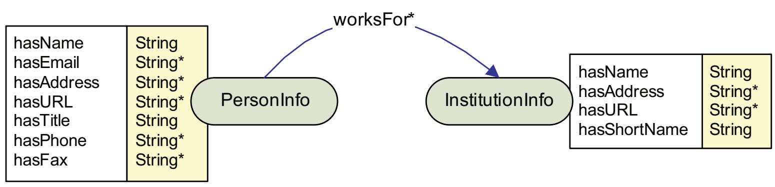

Figure 4.3: General Ontology of Author and Owner information

This general ontology is used by the following upper ontologies.

4.2.1.2 Common Shape Ontology

The Common Shape Ontology (CSO) targets different kinds of multimedia content, ranging from 2D/3D models to videos. This maintains top-level information that can be shared and used in different domains. Figure 4.4 presents the structure of this ontology.

The most important class is the *ShapeRepresentation* together with its specialized subclasses. *Shape Representation* is considered as an abstract concept that encapsulates information inherent to a shape model. The left part of figure 4.4 presents the properties of the *ShapeRepresentation* concept. These properties are inherited to specific concepts which also have specialized properties that describe them.

In this first description, a digital shape is considered as a generic resource. Users are typically interested in getting information about contact person or institution given by *PersonInfo* and *InstitutionInfo* respectively. Therefore, specific properties addresses the creator, the owner, the contact and the uploader person of a digital shape. Another simple yet important way to look at a digital shape is to consider it

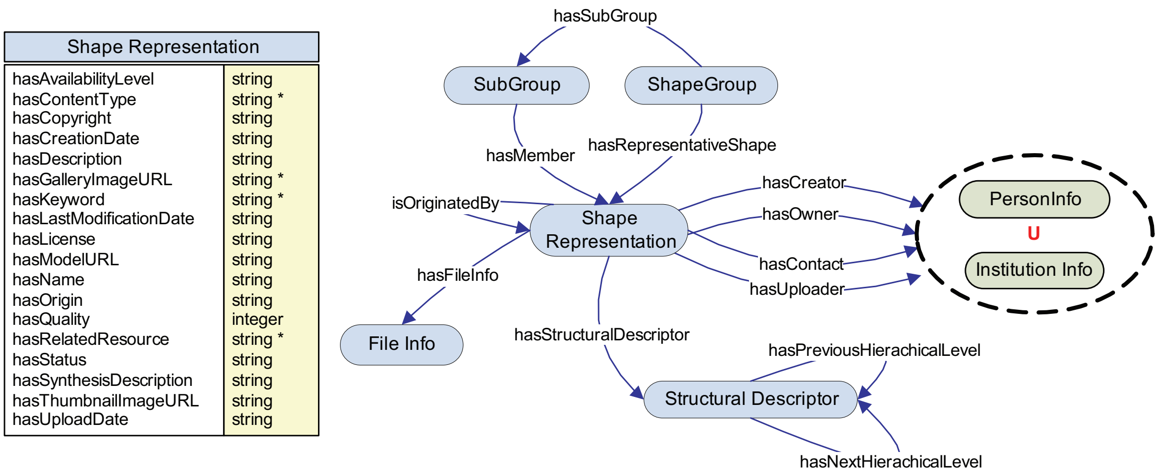

Figure 4.4: Main structure of the Common Shape Ontology

as a file. Thus, each shape can be related to a *FileInfo* instance, where one can find the file's name, size, format, URL, etc.

Shape Representation concept includes the different resources that can be part of a shape or a shape by itself. However, the *Shape Representation* class has more specialized subclasses as presented in the taxonomy in figure 4.5. Concepts in the very first level are: B-Rep, Animation Sequence, Geometrical Representation, Multi Resolution Model, Structural Descriptor and Raster Data. The definitions of these concepts can be found in the glossary in the Appendix B.

The goal of this taxonomy is not only to provide a useful categorization of the digital shapes, but also to provide to each category its own specific attributes and relations. Figure 4.6 presents a description of those concepts that are of special interest of this thesis. This concepts represent some knowledge about Virtual Human creation and composition. As described in section 2.1 a Virtual Human can be created using existing sources such as 2D images, video, or spatio-temporal data. This relationship between different media sources can be conserved using the property *isOriginatedBy*.

Another example is the description of Virtual Human's skeletal structure. A Virtual Human *Mesh* can be described with a *StructuralDescriptor* through the property *hasStructuralDescriptor*, which may be related to a *CentreLineGraph* structure. The Center Line defines complex objects that can be seen as the arrangement of tubular-like components. This is abstracted to a collection of center-lines that split and join following an object's topology, which forms a skeleton. A center-line should satisfy the following requirements: centricity, connectivity and singularity.

The CSO is partially used by the VHO, however this does not limit the reusability of some components for describing elements not covered here.

Next section presents the second top-level ontology used by the VHO.

4.2.1.3 Common Tool Ontology

The Common Tools Ontology (CTO) captures meta-data related to specific software. Figure 4.7 depicts the main concepts of this ontology. Concepts from the Info-Common Ontology are used to relate software tools with their creators, owners, uploaders, etc.

The main concept, *SoftwareTool*, can be associated with other tools through the property *requiresTool*. It has a functionality defined through the concept *Functionality*, and it may implement one or more *Algorithms*.

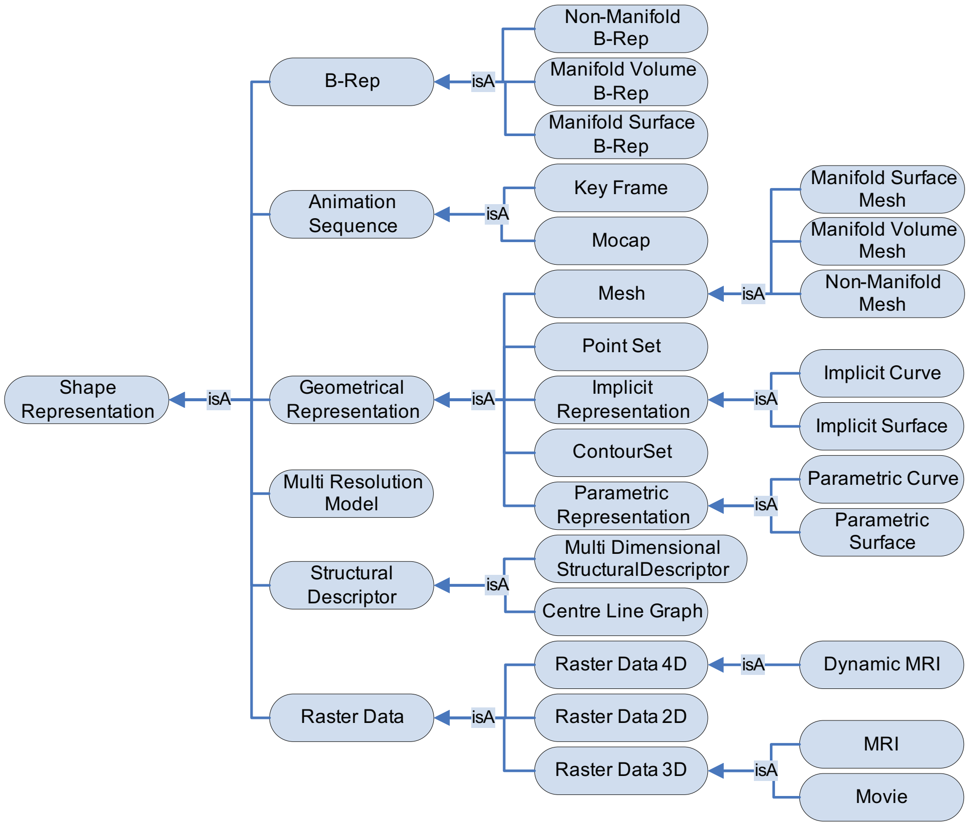

Figure 4.5: Common Shape Ontology Taxonomy.

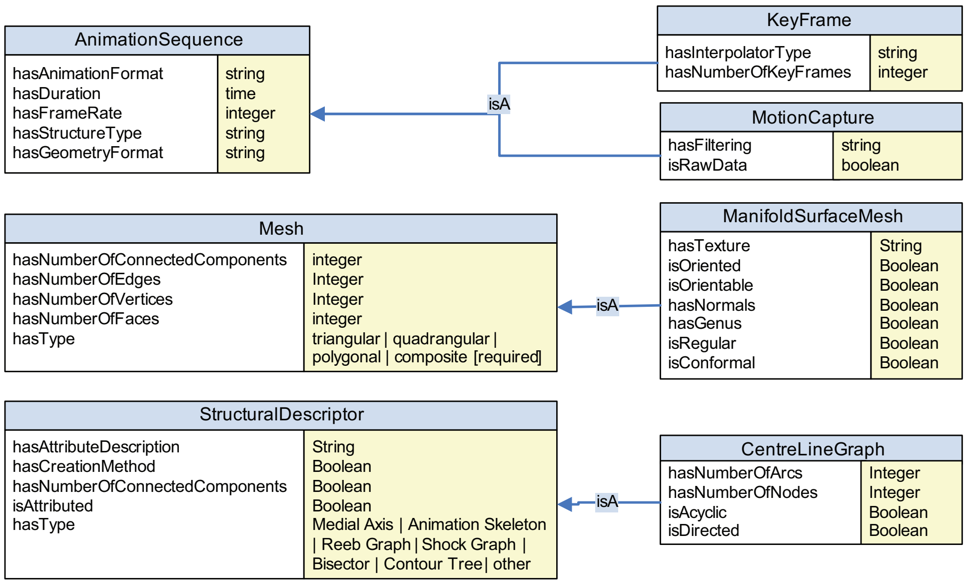

Figure 4.6: Datatype properties for relevant concepts for the Virtual Humans.

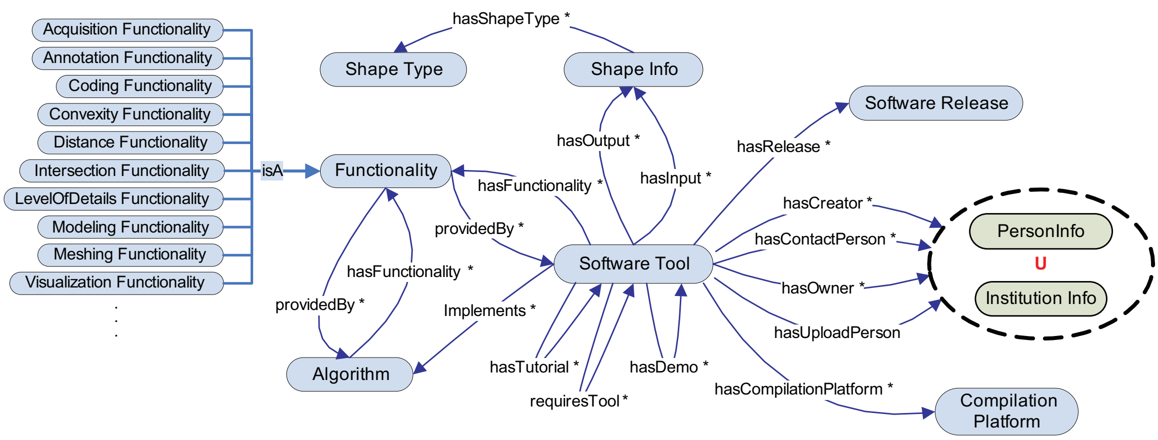

Figure 4.7: Main structure of the Common Tool Ontology

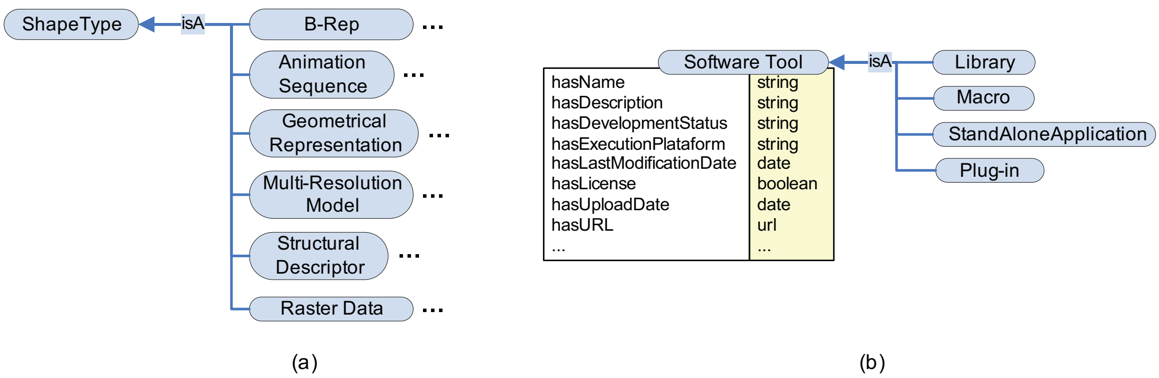

Figure 4.8: (a) Types of Software Tools and their properties. (b) Shape Type topology.

A tool can be related to a *CompilationPlatform* under which it operates, and it can have several *SoftwareReleases*. Each tool accepts specific inputs and provides specific outputs. The inputs and outputs may be one or more shapes, which are described by the concept *ShapeInfo*, and associated with a specific *ShapeType*. This Shape Type has the same topology as the *ShapeRepresentation* of the CSO presented in figure 4.8-a, but it does not describe the shape itself. It functions only to specify the type of shapes that can be processed by a software tool.

Finally, a software tool can be a *Library*, a *Macro*, a *Plug-in*, or a *StandAloneApplication* as depicted in figure 4.8-b. The CTO involves the basic concepts which are further extended by domain and task ontologies. This ontology is used to describe any implementation of the described capabilities or features for virtual humans presented in the chapter 2. For example a walking algorithm, a key frame player, a facial animation synthesizer, etc.

This section described the developed top-level ontologies that are reused by the domain specific ontologies. Next section describes two domain specific ontologies related to the creation process shapes including Virtual Humans.

4.2.2 Domain Ontologies

The domain ontologies capture more specific knowledge. The creation process of digital shapes is covered in two ontologies that have different approaches about their conception. The Shape Acquisition and

Processing (SAP) ontology considers that the creation of a shape comes from date extracted from a real object. While the Product Design Ontology (PDO) is focused in the creation of geometries from scratch and dedicated mainly to the product development process.

4.2.2.1 Product Design Ontology

The Ontology for Product Design (PDO) has the objective of guiding researchers and developers in e-science scenarios in the development of tools and methods that support industrial product design and engineering analysis. It also formalize processes, tools and know-how relevant to the free-form modeling and the engineering simulation phases [29].

The key concepts of the PDO are:

Task : describes the workflow where different shape representations and shape processing tools are used.

ShapeRole : describes the shape conditions and task conditions required to perform a certain task.

Figure 4.9 presents the ontology diagram of the PDO. The *Task* concept represents the workflow. A *ShapeRepresentation* from the CSO represnets the input or output of a *Task*. This workflow based on tasks can describe a hierarchy through the properties *hasSubTask* and *hasSuperTask*. It can also describe a sequential flow with the properties *hasPredecessor* and *hasSuccessor*. A Task is also described by the *ShapeRole* which represents the type of shape that the task needs. The *ShapeRole* specifies certain conditions to accomplish, described in the *ConditionType* concept. A particular 3D model in *PDModel* can be classified inside a *ShapeRole*.

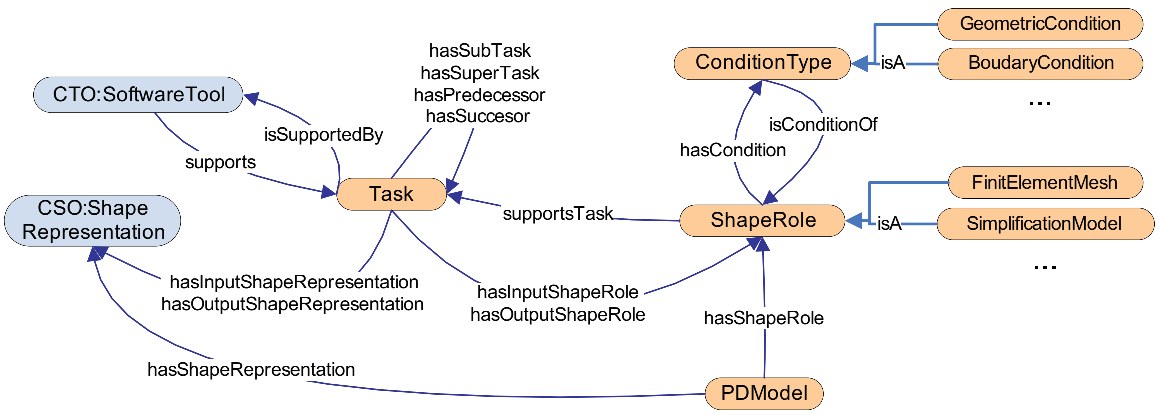

Figure 4.9: Main structure of the Product Design Ontology

4.2.2.2 Shape Acquisition and Processing Ontology

The history of a model can be described in a precise way when specialized technologies such as Motion Capture and 3D scanning are used. This section describes a task ontology that specializes in the description of the methodologies followed for this kind of data acquisition process.

The Shape Acquisition and Processing (SAP) ontology uses the top-level ontologies described before. The domain of this ontology is defined as *the development, usage and sharing of hardware tools, software tools and shape data by researchers and experts in the field of acquisition and processing of shapes* [4].

SAP ontology aim to preserve useful information at different levels (geometric, structural, semantic) when passing from the real world to the digital world or when performing actions in the digital world to make improvements or to add features. For the creation of this ontology, it is considered that a digital shape can be created either from a real object or synthetically. Two critical phases are related to the life cycle of a digital item.

1. **Acquisition Phase**: in this first phase the contextual knowledge includes different conditions and properties related to the object to be digitalized, to the surrounding environment or even to the knowledge of the experts. Most of this information must be preserved and passed to the other steps of often complex modeling pipelines, in order to improve the quality of the results and to open to new research approaches.

2. **Processing Phase**: in this phase different tasks with different objectives can be performed. A digital item can be structured, identifying critical features and/or significant portions, or it can be enhanced, removing for example undesired holes or noise. There are a multitude of different operations that can be applied to a digital shape, all of them depending from different application contexts.

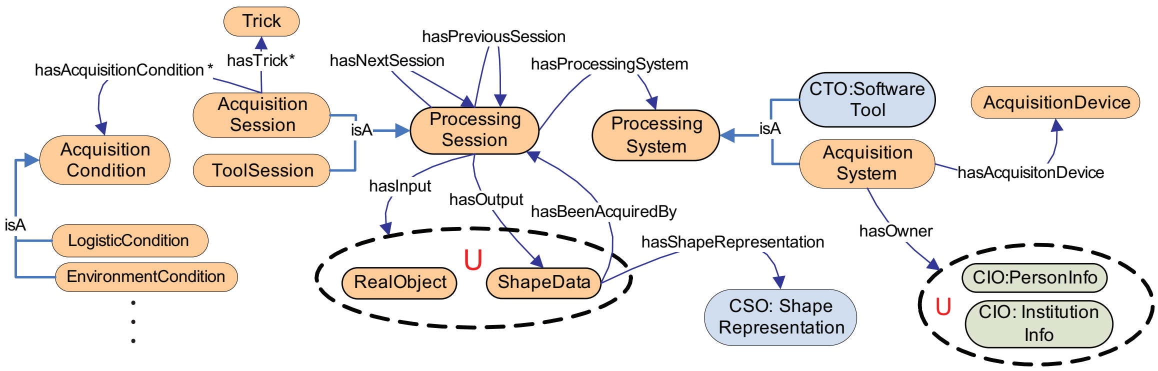

Figure 4.10: Main structure of the Shape Acquisition and Processing Ontology

SAP ontology's main concepts and relations are depicted in figure 4.10. A digitalization process starts with an *AquisitionSession* which is subclass of a *ProcessingSession*. These sessions may have as input a *RealObject* or *RealPerson*, which is described with the property *hasInput*. The output of the process is defined with the *hasOuput* property, and this output is defined in the *ShapeData* class, which makes reference to a digital shape in the CSO (a *ShapeRepresentation*). During the acquisition session the hardware used and the conditions of the sessions are also captured. The hardware used inside an

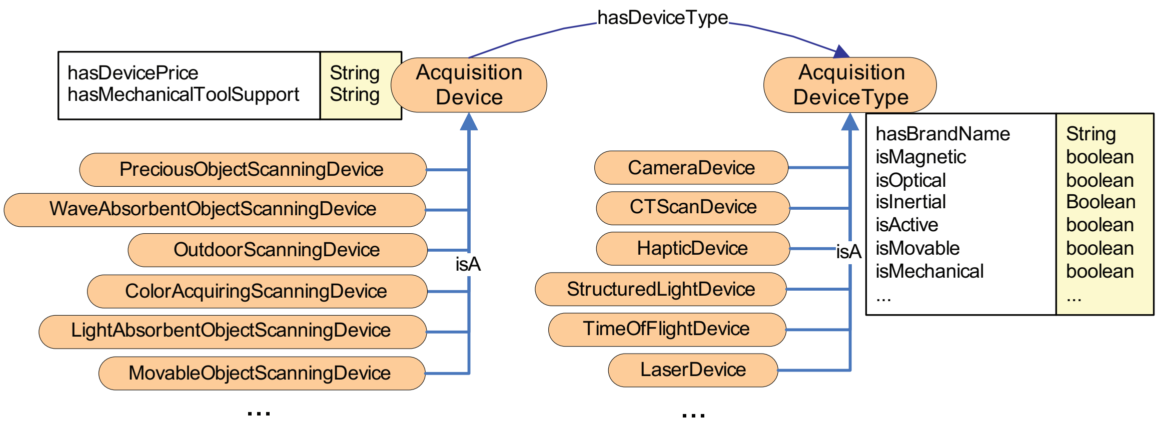

Figure 4.11: Data acquisition devices in the SAP ontology.

acquisition session is described in the *AcquisitionDevice* and *AcquisitionDeviceType* concepts depicted in the figure 4.11.

The *AcquisitionSession* has the property *hasProcessingSystem* that can be either a *SoftwareTool* or an *AcquisitionSystem*. In other words, the Processing Session can use a system that is software or hardware. The *AcquisitionSession* class is also described by *acquisitionCondition* property, which can be: logistic, environmental, or setting up in the case of Motion Capture Session. It is also possible to register any *Trick* performed during a session.

After an acquisition session of a shape, the shape needs to be processed using software tools. Thus, the next *ProcessingSession* is a *ToolSesion*. Some examples of a tool session may be the mesh reconstruction, or the skeletonisation of a human shape. The description of processes like these can be useful for instance, to compare different creation or processing techniques using the same sources.

4.3 Conclusions

This chapter described the AIM@SHAPE project in which context five ontologies were created with the purpose of adding semantics to digital shapes. Inside this project the DSW framework was developed to provide to researchers a search engine where they can find specialized digital shapes, tools and related knowledge using higher level descriptors using ontologies.

The ontologies developed within AIM@SHAPE are:

- a general ontology that describe persons and institutions that related to shapes and tools;
- two upper ontologies, one describes shapes in general and the other software tools related to these shapes;
- tree domain ontologies that focuses in the shape creation and description. Two of them are described in this chapter: the PDO describes shapes created from scratch and used for the product design industry; and, the SAP which describes the creation process of shapes which data is acquired from real world.

The third domain ontology is the Virtual Human Ontology. This ontology was developed for this dissertation and is described in the next chapter.

Chapter 5

Semantics for Virtual Humans

The contribution of this dissertation is a semantics-based method for organizing the various types of data that constitute a Virtual Human. Its main objective is to foster a common understanding and sharing of such complex 3D entities. The knowledge related to the synthesis, animation techniques and functionalities of Virtual Humans is formally specified using semantics.

Semantics for Virtual Humans aims at organizing the knowledge and data of three main research topics and applications involving graphical representations of humans:

- Human body modeling and analysis: morphological analysis, measuring similarity, model editing and reconstruction.
- Animation of virtual humans: Shape deformation may be attached to the shape, or inside an algorithm. To produce animation values there are several techniques, which can be divided into motion capture and key frame.
- Virtual Human behavior: The behavior controllers are algorithms that drive the behavior of the character considering the emotional state and its individuality.
- 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.

This chapter presents the semantics defined for these aspects of Virtual Humans. Next section deals with the description of the creation process of 3D models and Motion Capture animation sequences.

5.1 Semantics for Data Acquisition

Data acquisition consists in the extraction of data from the real world to be used in the creation Virtual Humans. For example, the 3D scanning of a real person to create his/her geometrical model, or the capture of movements to create animation sequences to simulate Virtual Human movement.

Section 4.2.2.2 of the previous chapter presented the Shape Acquisition and Processing ontology (SAP). The domain of this ontology is to describe the creation process of geometrical shapes. Therefore, the creation of human-like shapes and animations fits well in the SAP ontology. However, it is necessary to complement the SAP ontology to fully describe the creation of Virtual Human shapes and animations sequences.

Semantics added to the SAP ontology are for describing the acquisition systems and acquisition conditions related to the acquisition human-like 3D shapes and Motion Captured animations.

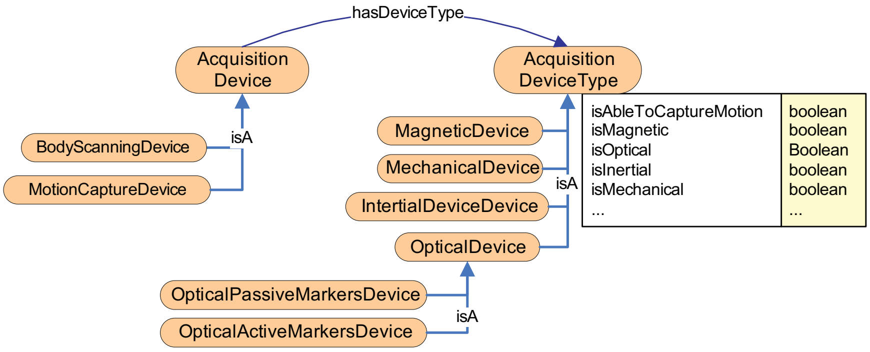

Figure 5.1: Data acquisition devices in the SAP ontology for Human Shape and Motion Capture.

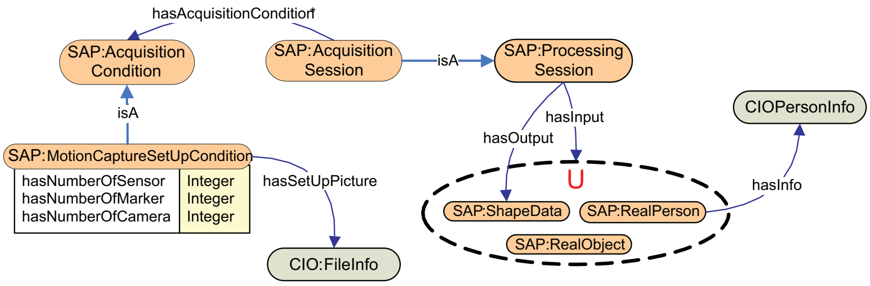

Figure 5.2: Description of the conditions of the Motion Capture sessions.

Figure 5.1 presents the *AquisitionDevice* classification and the description of scanning and Motion Capture devices. These devices can be of different types, as defined in the *DeviceType* class. Using the *hasDeviceType* property it is possible to automatically classify instances by creating a necessary and sufficient condition under each subclass of *AcquisitionDevice*. An example to classify *MotionCaptureDevice* would be through the condition:

$$\exists hasDeviceTypesome(AcquisitionDeviceTypeand(isAbleToCaptureMotionhastrue))$$

The SAP ontology has a good description of scanning sessions, thus there is no need to add more descriptors for these. However, Motion Capture sessions differ, and it is possible to include information about the conditions of the session such as the number of markers used, the number of cameras, how markers were placed (a picture of the actor with the exact position of the markers) and the actor's information (person information).

Figure 5.2 presents some classes from the SAP ontology that are reusable to describe Motion Capture sessions: *ProcessingSession* and *ProcessingSystem*. *ProcessingSessions* class has two subclasses: *AcquisitionSession* and *ToolSession*. These classes are used to differentiate processes that use weather a device or a software tool. The *AcquisitionSession* has the property *hasAcquisitionCondition* that refers to the class *AcquisitionCondition*. The specificities of Motion Captured sessions are described using a class called *MotionCaptureSetUpCondition* under AcquisitionCondition with the properties: *hasNumberOfSensor*, *hasNumberOfMarker*, *hasNumberOfCamera* and *hasSetUpPicture*.

The description of the person that was scanned or Motion Captured is also included in the semantics to complement the description of a session. This is also presented in figure 5.2, where the *ProcessionSession* has the range of the *hasInput* property defined as the union of *ShapeData*, *RealObject* and *RealPerson*. A Real Person is described with the *PersonInfo* concept defined in the Common-Info Ontology (see section 4.2.1.1).

In this way the history of a shape acquisition and Motion Capture sessions can be kept using the processing pipeline proposed in the SAP ontology. Using this semantics it is possible to describe in detail how a shape or an animation was created and processed. The resulting digital items from the creation process are now ready to be used for populating virtual environments.

The descriptors presented in this section complements the SAP ontology to describe the acquisition and processing of human shapes and Motion Capture sessions.

The rest of this chapter focuses in the presentation of the semantics of the Virtual Human's features.

5.2 Virtual Human Ontology

The Virtual Human Ontology (VHO) is an specialized ontology which domain is *the description of Virtual Humans to support their creation process and the population of virtual environments.* This section describes in detail the semantics proposed in this thesis.

5.2.1 General Information

The description of Virtual Humans includes general concepts that share information about their creators and versions. Figure 5.3 presents the main concepts of the VHO. The class *Resource* encapsulates concepts that are described with the properties *hasAuthor*, *hasVersion* and *hasVersion*). The subclasses of the *Resource* class, there are: *VirtualHuman VirtualHumanAnimationSequence*, *VirtualHumanControllers*, *VirtualObjects* (*SmartObjects* and *Garments*). *VirtualHuman* is the main concept of this ontology, and it represents a full-body or partial representation of a Virtual Human.

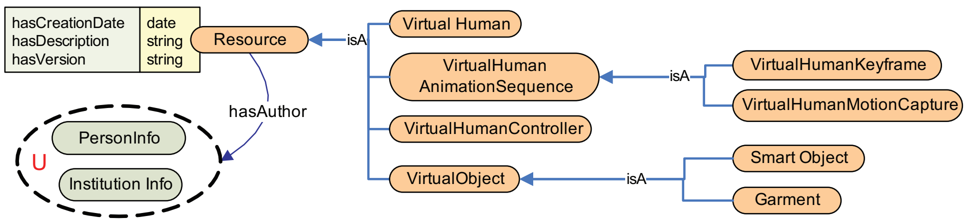

Figure 5.3: Main concepts of the Virtual Human Ontology

A Virtual Human can be synthesized in a variety of ways and can represent a real person through an avatar or a virtual one with autonomous behavior. Virtual Humans are characterized by a set of general attributes (gender, age, height, weight), a geometry, and structural descriptors. A Virtual Human can perform animation sequences, can wear a virtual garment, can use smart objects and can be controlled by Virtual Human Controllers. These descriptions are depicted in figure 5.4. At the same time some conceptualizations are also related with concepts from top-level ontologies. For example, the *Geometry* class *hasShapeRepresentation* is a geometry described in the CSO; the *VirtualHumanController hasTool* described as a *SoftwareTool* form the CTO; and so on.

The following subsections describe in detail the data properties and relation properties of each of the concepts that describe the Virtual Human.

5.2.2 Geometry and Structure

When acquiring a shape from scanning, the results are a single static snapshot of the body shape. For many range applications, this static snapshot is not sufficient because it does not capture all the possible degrees of flexibility of the human shape. To mimic the flexible and dynamic behavior of the

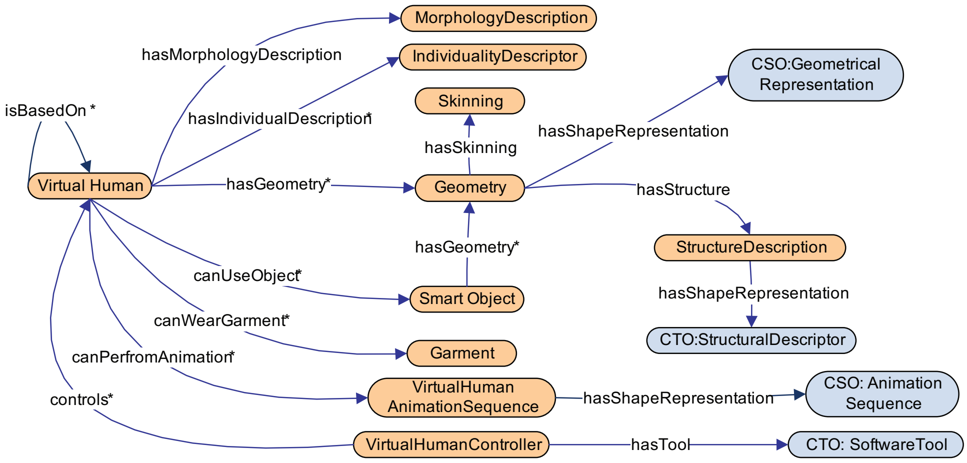

Figure 5.4: Main concepts related to the Virtual Human features description.

human shape, the traditional approach uses skeleton-driven deformations. This classical method for skin deformation is the most widely used technique in 3D character animation. Body animation binds a 3D shape to an articulated control skeleton. Binding information is then used to deform the body shape according to control skeleton motion.

Therefore, a Virtual Human is described by a geometry that can have an internal structure and a type of skinning. The structure represents firstly a skeleton which can be defined as a Center Line Graph. Figure 5.5 presents the diagram of the ontology where the *VirtualHuman* is described by the property *hasGeometry* connected with the *Geometry* class, which has the property *hasStructure* that describes the skeleton. To specify that this structure has to be of Center Line Graph type (described in the CSO), the following necessary restriction is defined in the *StructureDescription* class:

$$\exists\ hasShapeRepresentation\ CSO : CentreLineGraph$$

To distinguish a Virtual Human structure from objects' structures the subclasses *VirtualObjectStructure* and *VirtualHumanStructure* are created, including the following restrictions:

In the *VirtualHuman* class:

$$\exists\ Geometry\ and\ (hasStructure\ some\ VirtualHumanStructure)$$

In the *VirtualObject* class:

$$\exists\ Geometry\ and\ (hasStructure\ some\ VirtualObjectStructure)$$

The description of the Virtual Human's*StructureDescription* starts with the definition its root *Node* (using the property *hasRootNode*). A *Node* is an abstract class that describes a family of graph-like nodes that can constitute diverse structures. A *Node* is defined by its children nodes with the property *hasChild** that points to itself. This property is transitive, which allows for inferring all children nodes that are down in the structure of any *Node*.

Virtual Human's skeleton is close related to the H-Anim [74] standard. This specification considers that the human body is conformed by a number of segments (such as the forearm, hand and foot) which are connected to each other by joints (such as the elbow, wrist and ankle). An H-Anim file should contain a set of *Joint* nodes that are arranged in a hierarchical form. Each *Joint* node can contain other *Joint* nodes, and may also contain a *Segment* node which describes the body part associated with that joint. Each *Segment* can also have a number of *Landmarks* nodes, which define locations relative to the

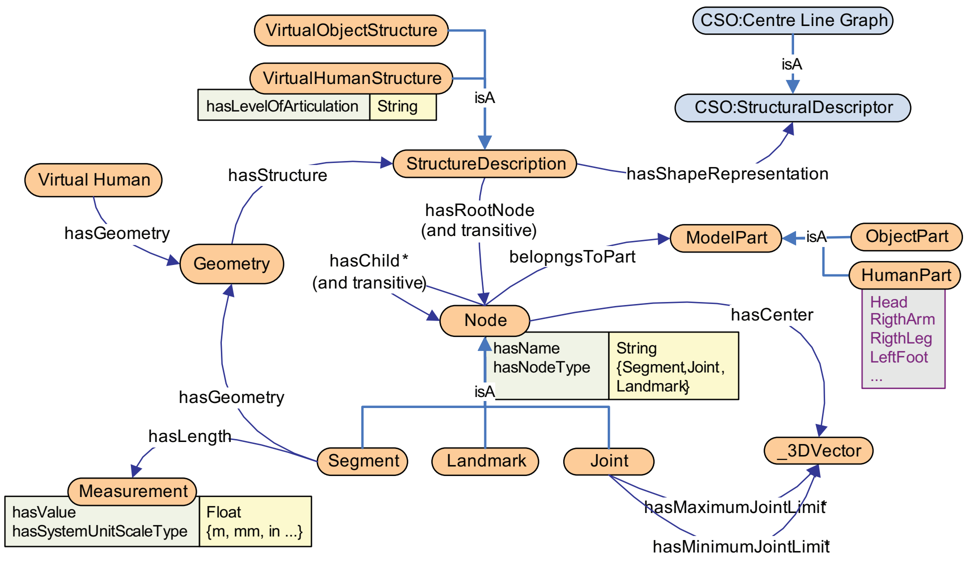

Figure 5.5: Description of the structural features associated to the Virtual Human skeleton

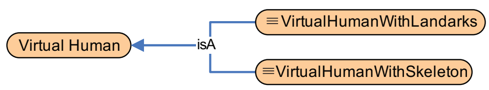

Figure 5.6: Classes to infer Virtual Humans with specific features

segment. *Landmarks* are featured points that store information associated to a particular location in a 3D geometry. This standard gives support to the creation of interchangeable humanoids, behaviors and animations.

The ontology diagram presented in figure 5.5 depicts that a *Node* can be of different kinds. The types of nodes considered are: *Segment* (segmentation of a geometrical shape), *Landmark* (featured points) and *Joint* (articulations). This skeletal structure describes indirectly the human body parts, the property *belongsToPart* connected with the *ModelPart* class represents this association.

Even this representation of the Virtual Human structure is close to the H-Anim, it is also flexible enough to define any other structures that have the centre line graph principle.

To make inferences of Virtual Humans that have features integrated, a set of subclasses were defined. These are subclasses of *VirtualHuman* class. They are presented in the figure 5.6, and denoted by the ≡ symbol. These classes are not meant to be used for creating instances directly, instead they are used to infer which Virtual Human instances accomplish the given conditions (by checking the Necessary and Sufficient restrictions in the class description).

The *VirtualHumanWithSkeleton* class has the necessary and sufficient condition:

$$\exists\ hasVirtualHumanStructure\ some\ VirtualHumanStructure\ \cup \forall\ hasRootNode\ only\ Joint$$

When a classification is performed using a reasoner (i.e. Reacer [129]), those Virtual Human instances that have at least a *VirrtualHumanStructure* and this structure *has a RootNode* of type *Joint*, will be automatically instances of the *VirtualHumanWithSkeleton* class.

The *VirtualHumanWithLanmarks* class has the following necessary and sufficient condition:

$$\exists\ hasVirtualHumanStructure\ some\ (VirtualHumanStructure\ and$$
$$(hasRootNode\ some\ (Node\ and\ (hasChild\ some\ Landmark))))$$

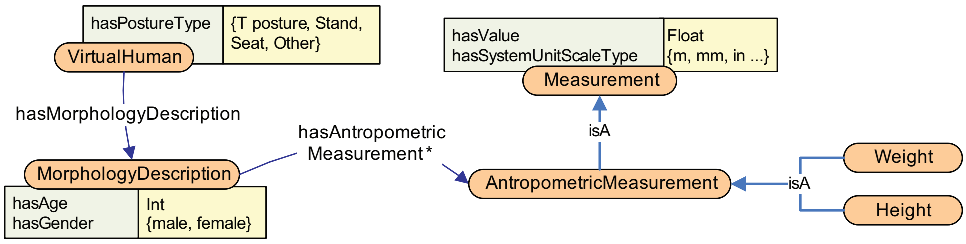

Figure 5.7: Description of morphological characteristics

This condition means that the Virtual Human's instances that will be classified under *VirtualHumanWithLandmarks* class should have at least one instance under the *VirtualHumanStructure* class, and this instance has the property *hasRootNode* related with an instance *Node*, that has the property *hasChild* related with another instance of type *Landmark*.

This subsection has dealt with the geometrical and structural description of Virtual Humans. However, Virtual Humans have other more abstract descriptions that are used to provide the corresponding motion/behavior according with their appearance.

5.2.3 Morphology and Individuality

There are some basic morphological characteristics such as: gender, height, weight and age, that influences each person's movements and behaviors. Some of these characteristics are inherent to the character's shape, and others to individual traits. Knowledge about body's shape morphology and individuality are important to reconstruct a shape, to adapt motions to different characters, and also to parameterize behavioral animations.

Morphological characteristics for the human shape (geometry) and human movement (animation) are described in the ontology. In figure 5.7, Virtual Human is described using the *MorphologyDescription* class. This class aims to encapsulate the human-like shape description, such as gender, age and anthropometric measurements. The *AnthropometricMeasurements* includes specialized subclasses of specific anthropometric measures (i.e. height, weight). This could be extended for more specialized measurements. The Virtual Human has also the *hasPostureType* property to specify the posture at which the model stands.

Besides adequating movements to a morphology, it is also important to create different individualities for Virtual Humans. Psychological models are used to create individual behaviors as presented in section 2.3. These descriptors are mainly for annotating animations or parameterizing behavioral algorithms. In figure 5.8, the concepts created to describe individual traits are grouped in the *IndividualDescription* class. These concepts are described by the subclasses *PersonalityTrait*, *EmotionalState* and *Mood*. The *EmotionalState* is related to one or more emotions, a duration and an intensity; the *Personality* is described by one or more *PersonalityDimensions* and their values; and the *Mood* is described by a single value.

The mentioned individual attributes are important for annotating animations with individual traits of personality or emotions. Animation sequences' semantics are described right after.

5.2.4 Animation Sequences

Since the creation of realistic animation sequences is very expensive, their description is very important to allow reusability. Figure 5.9 presents the defined semantics to describe animations according to the research issues presented in the section 2.2.

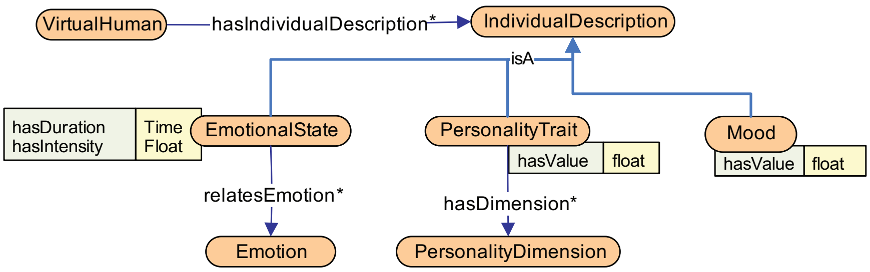

Figure 5.8: Concepts for individuality description

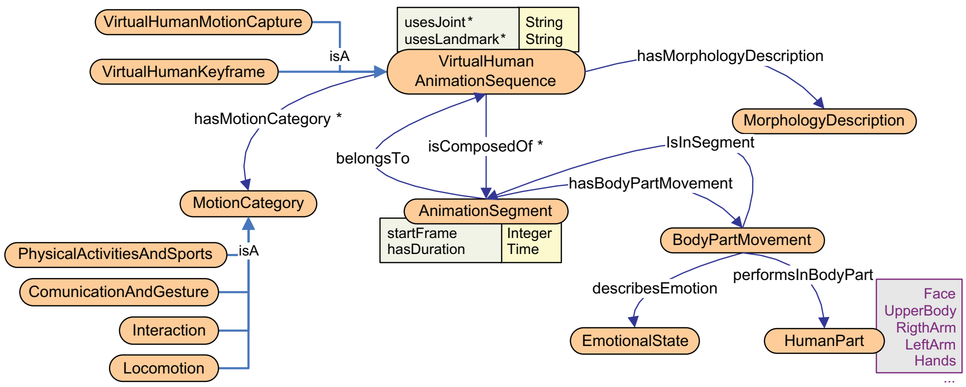

Figure 5.9: Description of Virtual Human Animation Sequences.

A first description of animations is their classification inside the class *MotionCategory*. This classification corresponds to the animation content, and one animation file can be classified in more than one category. *MotionCategory* class contains the subclasses: Interaction, Locomotion, Physical Activities and Sports and Communication and Gestures. An animation can be classified in one or more *MotionCategory*.

A second description of animation includes information about whether or not an animation fits to a specific Virtual Human. To be able to make this comparison, the animation is described in terms of the morphology it was created or adapted for. Then, the animation is related to *MorphologyDescription* class with the *hasMorphologyDescription* property. This description can be used also to describe the morphology of the actor if the animation comes from Motion Capture.

The animation can be segmented to allow describing it more in detail. Specific body part movements can be described, where descriptors for animations are specific gestures, emotional expressions, etc. To describe body part movements the *AnimationSegment* is related to the class *BodyPartMovement* using the property *isComposedOf*. The *AnimationSegment* class has the properties: *hasStartsFrame* and *hasDuration*, and the *BodyPartMovement* has the properties: *describesEmotion*, and *EmotionalState* and *performsInBodyPart*.

An application scenario for animation segment description and its usage is presented in chapter 8.

5.2.5 Animation Controllers

Controllers are algorithms for animating the characters. As depicted in figure 5.10, these controllers are represented by *VirtualHumanController* class. They require inputs to work and produce outputs which are usually animation sequences or specific joint values. Nevertheless, it is possible that the output is also input of other controller.

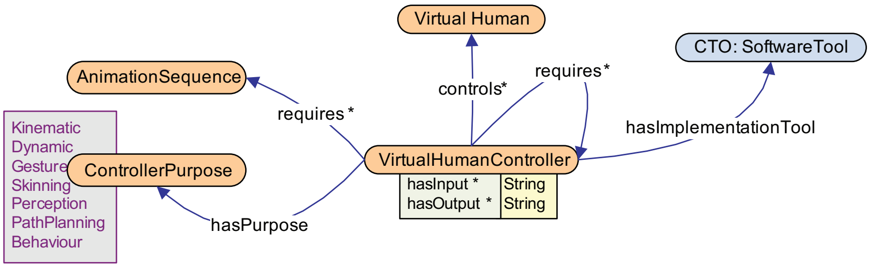

Figure 5.10: Description of Virtual Human Controllers.

*VirtualHumanController*s are described according with their purpose. A *ControllerPurpose* enumeration class is depicted in figure 5.10 with the following instances: Cinematic, Physical, Cognitive, Path Planning, Skinning, Sensorial, Gesture, and so on. Other description of *VirtualHumanController* class is to describe what are the requirements of a controller (*requires** property). These requirements may be other controllers, animations sequences or specific Virtual Humans (*controls** property).

Controllers are normally implemented in a software tool, where relation with the CTO is established with the *hasImplementationTool* property.

Until this part, the description of general features directly related with Virtual Humans are described. However, when considering that Virtual Humans are going to populate a virtual environment, more descriptions for interacting with the environment are needed. Next section presents some of the semantics considered for virtual objects that Virtual Humans can interact with.

5.3 Virtual Objects

This subsection describes briefly two kinds of virtual objects: Smart Objects and Garments. Even if these descriptions are not part of the core of this work, their inclusion is important to prove that the description of Virtual Humans provided in the sections above is compatible and reusable by other elements related with the Virtual Humans.

5.3.1 Smart Objects

The Smart Object approach [78] explores the idea of having a database of agent-object interaction information. For each object modeled, it is possible to include the functionality of its moving parts and detailed commands describing each desired interaction. Giving clues to aid the interaction, the Smart Object encloses all those attributes useful for Virtual Human to interact with an object [2]. Object's attributes are essentially a named value of a primitive type: 3D point, point, list, vector, matrix, boolean, Integer, Float, String, Quaternion or Matrix. These attributes are used as input values of higher level controllers. Grouping related attributes into attribute sets serves to specify a kind of interaction; for example, for the Virtual Human to get close to a specific object, would need: (1) the desired position of the Virtual Human in the end of the walking sequence, in the form of a point attribute; (2) the desired final orientation of the Virtual Human at the final position, in the form of a vector attribute.

In figure 5.11, the concepts to describe Smart Objects are shown. The *SmartObject* class inherits the descriptions of the *VirtualObject* class, a geometry, and a structural description. The attributes of an object are grouped in the *AttributeSet* class. An attribute set can have one or more object attributes where locations, transformations, or others, can be defined.

Information presented in the ontology can help to describe specific usage on an object, for example: the characteristics of an object (structure, physical properties, etc.), or if the object can be grasped.

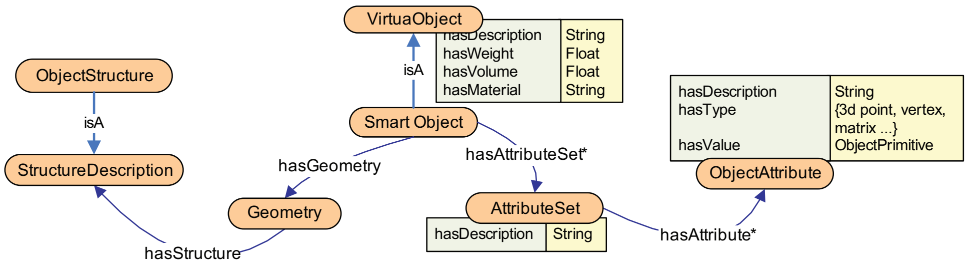

Figure 5.11: Semantics for Smart Objects

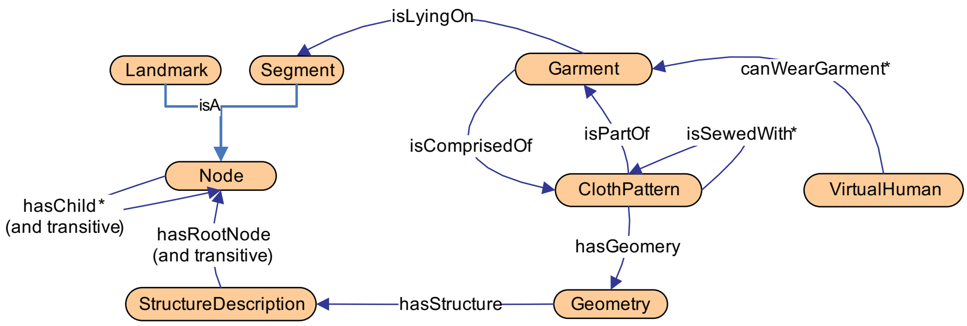

Figure 5.12: Description of Virtual Garments in the VHO

Next section describes virtual garments and accessories. These descriptions specify their composition and correct positioning of these objects into the Virtual Humans.

5.3.2 Garments and Accessories

Garments and accessories are objects that are attached to the human shape, such as clothes, jewels, hats, glasses, etc. Attaching accessories to a human shape involves locating where they should be placed on the body shape and extracting measurements information in order to fit the accessories to the body shape.

A garment is composed by some cloth patterns sewed with each other. A piece of clothing corresponds to a part or segment of the human body, for example the sleeve of a t-shirt corresponds to the arm. To put the cloth in the right position, some landmarks have to be identified in the corresponding human body segment.

Previous work on the ontological description presented by Fuhrmann et al. [53] was considered to integrate these concepts into the VHO. This ontology specifies that garments are composed by a set of Cloth Patterns that are geometries. A Cloth pattern is lying on a Human Segment which has landmarks to identify the right position. The ontology diagram in figure 5.12, presents the described concepts for garments. To define the portioning of the garment in the human segment, the same structural descriptor is used; which also serves to define Landmarks in a Cloth pattern.

Including the definitions for smart objects and garments has been very useful to demonstrate that concepts initially described for Virtual Humans can be easily reused by other concepts that relates with them. Next section presents a scenario that describes the creation pipeline of a Virtual Human form the scanning of a real person until the animation stage. This scenario demonstrates the relation of the ontologies presented in the chapter before and the VHO.

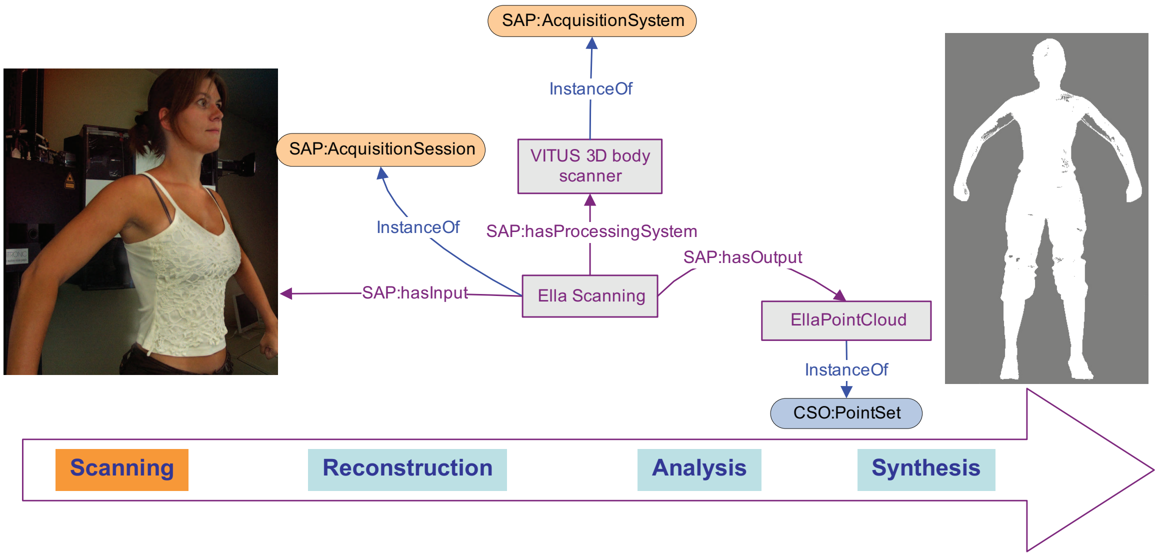

Figure 5.13: Scanning session description.

5.4 Ontology Validation

The semantics presented in this chapter have been validated by queering the ontology and creating user scenarios. This section presents some of the scenarios formulated to validate the VHO. One scenario is the creation pipeline of a Virtual Human from a real person, and two other scenarios queries the ontology searching for specific Virtual Human's features.

5.4.1 Virtual Human Creation Description

This scenario is related to a human shape acquisition to create an animated Virtual Human from a real person. A scenario like this is relevant for Virtual Human applications 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 scenario is based on two upper ontologies (CSO and CTO), and two domain-specific ontologies (SAP and VHO).

The "*SAP*" prefix denotes a concept belonging to the Shape Acquisition and Processing Ontology; the "*VHO*" prefix means that the concept is formalized in the Virtual Human ontology; the "*CSO*" prefix means that the concept belongs to the Common Shape Ontology; and the "*CTO*" to the Common Tools Ontology. Every action which is foreseen in the conceptualization process (e.g. scanning, reconstruction) has specific requirements for input and output. The interconnections between inputs and outputs create the actual workflow to pass from a real person to the animated Virtual Human.

The shape acquisition starts with a scanning session, presented in figure 5.13 (*EllaScanning* instance of *SAP:AcquisitionSession*). After the scanning of the person is completed, a points cloud representation is created, which is a set of points in the 3D space (*EllaPointCloud* instance of *CSO:PointSet*). This acquisition is performed with a dedicated scanner (*VITUS_3D* instance of *SAP:AcquisitionSystem*); detailed information about the acquisition system is also maintained. The acquisition session modeled in the SAP Ontology formalizes all the necessary knowledge related to the acquisition phase, including the logistic and environmental conditions under which the scanning has been performed.

The creation process continues with a reconstruction from the points cloud as presented in figure 5.14. The reconstruction of the surface is considered as a tool session (instance of *SAP:ToolSession*) because it uses a specific software tool (instance of *SAP:SoftwareTool*) that performs meshing, merging and hole

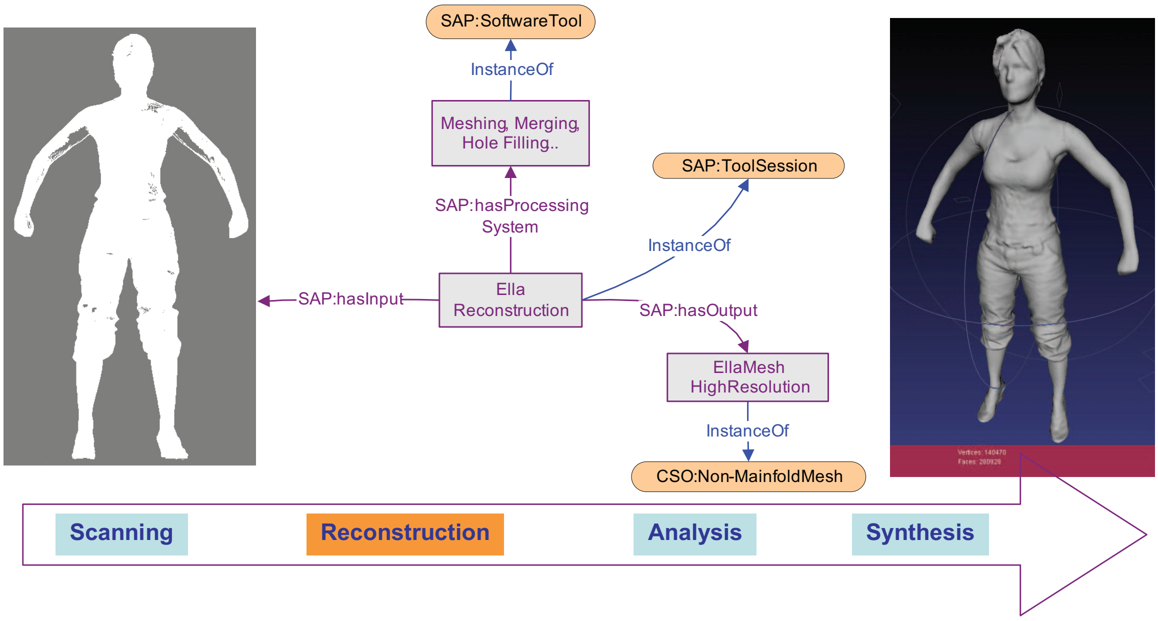

Figure 5.14: Shape reconstruction description.

filling operations. From the reconstruction process, a non-manifold surface mesh is created (*EllaMesh-HighResolution* instance of *CSO:NonManifoldMesh*).

At this step, a geometrical digital representation of the real person is obtained. However, the shape is not ready for a simulation, it still needs to be analyzed to create the features needed for the virtual representation of the real person. This is done by creating an internal structure in the mesh to be able to apply an animation. This step requires making an analysis of the shape for its segmentation, annotation and mapping, depicted in figure 5.15. A phase of analysis and mapping is performed (an instance of *SAP:ToolSession*) which uses a specific tool (*Plumber*[PlumbnberRef], instance of *SAP:SoftwareTool*). The output in this last step is a structural representation of the shape (*EllaBody*, an instance of *CSO:MultiDimensional StructuralDescriptor*).

This structured human shape can be processed by a designer to be described inside the Virtual Human domain, presented in figure 5.16. The Virtual Human concept is a human shape that has a geometry and a skeletal structure (*VirtualHumanElla* becomes an instance of *VHO:VirtualHuman* because it has Geometry, a Skeleton and a morphology description). In the VHO it is possible to describe animations that can be used by a given Virtual Human. For example, an animation (*IdleAnimation* is an instance of *CSO:Animation3D*), and can be assigned to the Virtual Human to be animated.

During this creation pipeline the history of the shape is stored in the CSO. This allows us to answer to competency questions such as: *What shape was originated from shape 'Ella'?, What system was used to generate it?, Does the shape 'EllaMesh' have a structural descriptor? Who produced shape 'EllaPointColud'?*. Furthermore, the VHO is also to answer domain specific questions, e.g., *Who is the real person used to create the animated Virtual Human 'VirtualHumanElla'?, What animations can be used by 'VirtualHumanElla'?*

Scenarios like this one shows the information that can be described in the ontologies and how it is organized. The following two scenarios are more practical in the sense that they aim to demonstrate how competency questions are formulated and answered with the ontology.

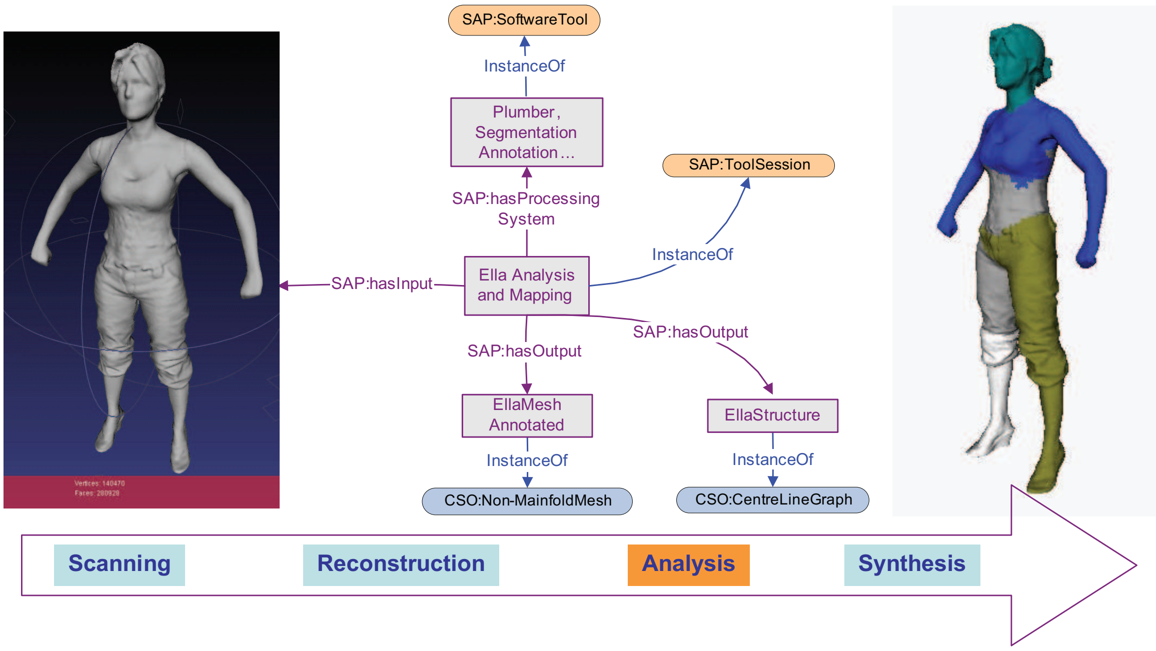

Figure 5.15: Shape analysis description.

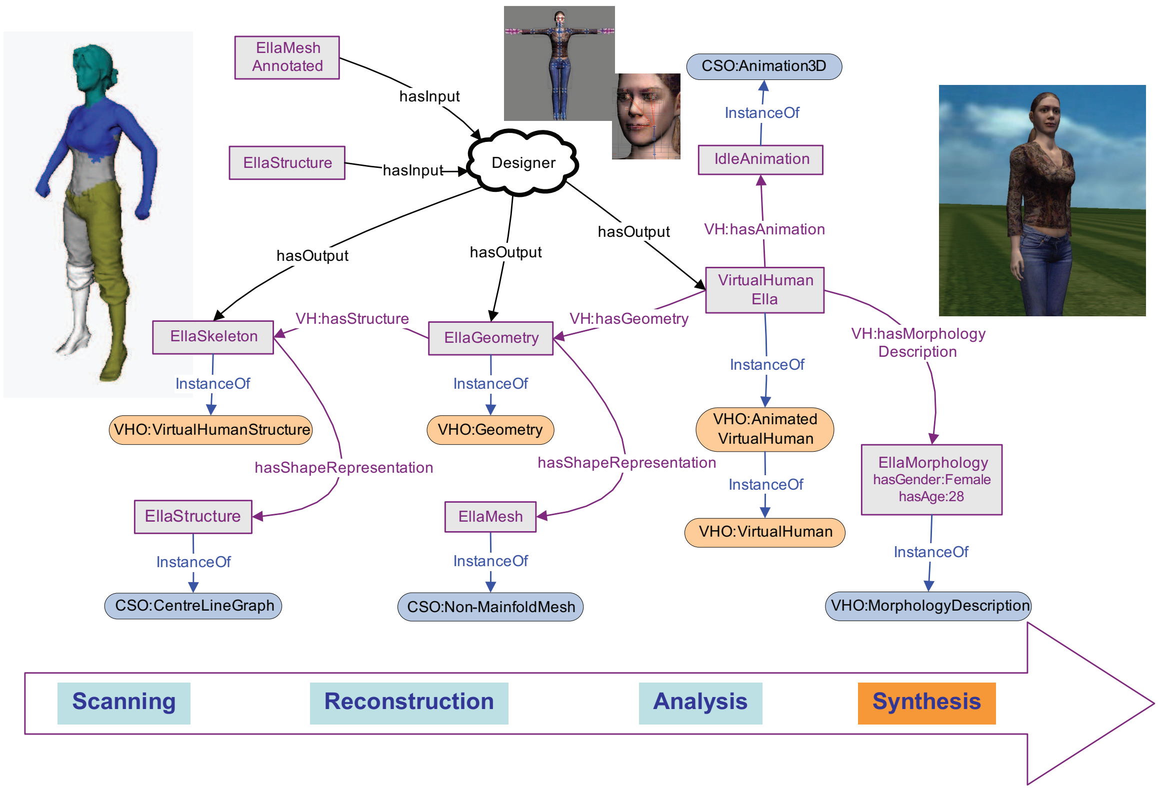

Figure 5.16: 3D simulation of the real person.

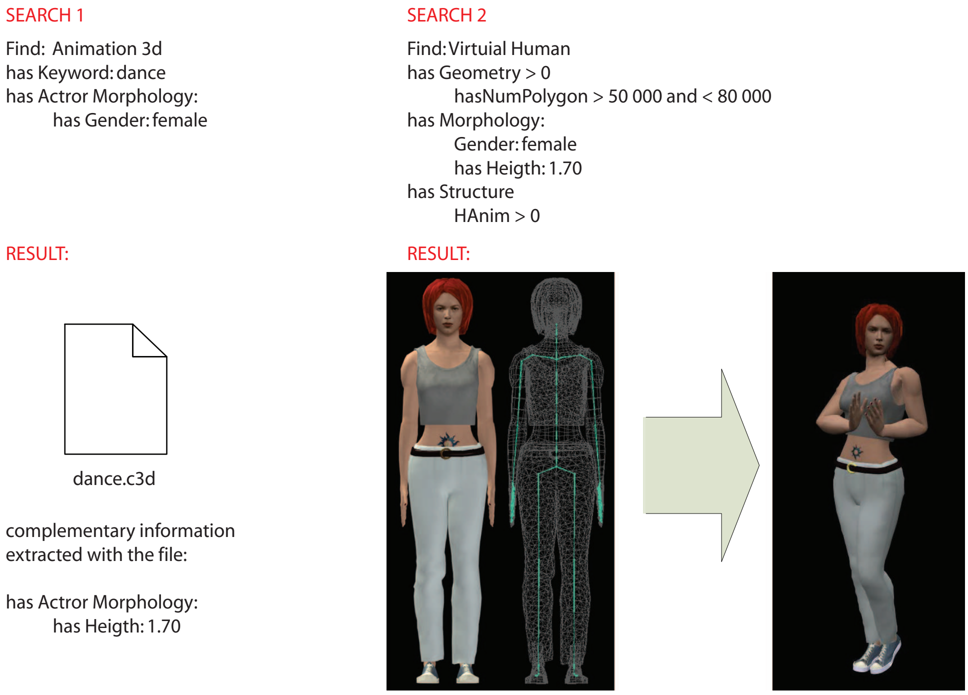

Figure 5.17: Searching process to get Virtual Human dancing.

5.4.2 Queering the Ontology

This section presents two case scenarios where the ontology is queried to answer specific competency questions. The first scenario aims to find an animation sequence that can be reused with a specific human model. The second scenario explores the internal structure of a Virtual Human that will help the user to set up an animation technique (inverse kinematics). For these scenarios, a specialized search engine developed by Vasilakis et al.[152] was used to perform queries in the VHO.

The first scenario considers that a designer wants to have an animation of a woman dancing. He looks in the ontology for animations of women that have the keyword dancing (figure 5.17 search 1). He founds one in C3D [1] format which can be opened in Motion Builder [2]. He makes a retargeting of the Motion Captured animation to the H-Anim skeleton, and he exports the animation to 3D Studio Max [3] He founds that the Motion Captured animation is made for a woman of 1.70 m. Therefore, he needs to look for a woman with a height of 1.70 m, with a H-Anim structure, and also with a number of polygons between 50k and 80k, because that is the range that models can be loaded and animated in the viewer without problems (figure 5.17 search 2). He founds a model in Collada [4] format which can be opened in 3D Studio Max as well. Finally, he can attach the animation to the model and play it. The result of the last search is presented in the figure 5.18, which is a screen shot of the search engine.

This process can take 30 minutes in the best case, and much more depending on how “clean” are the files that the designer found. It is demonstrated that using an optimized search like this simplifies the work of the designers. Normally, a designer would first need to identify the dancing animations by playing each one; then he would have to check if the dancing sequence fits the selected human model, which means a lot of retargeting job because the scale of the model is not the same as the one used for

[1] C3D, Coordinate 3D. http://www.c3d.org/.
[2] Autodesk, Motion Builder Software. http://usa.autodesk.com/adsk/servlet/index?id=6837710&siteID=123112.
[3] Autodesk 3ds Max. http://usa.autodesk.com/adsk/servlet/index?id=5659302&siteID=123112.
[4] COLLADA, Asset Exchange Schema for Interactive 3D. http://www.khronos.org/collada/

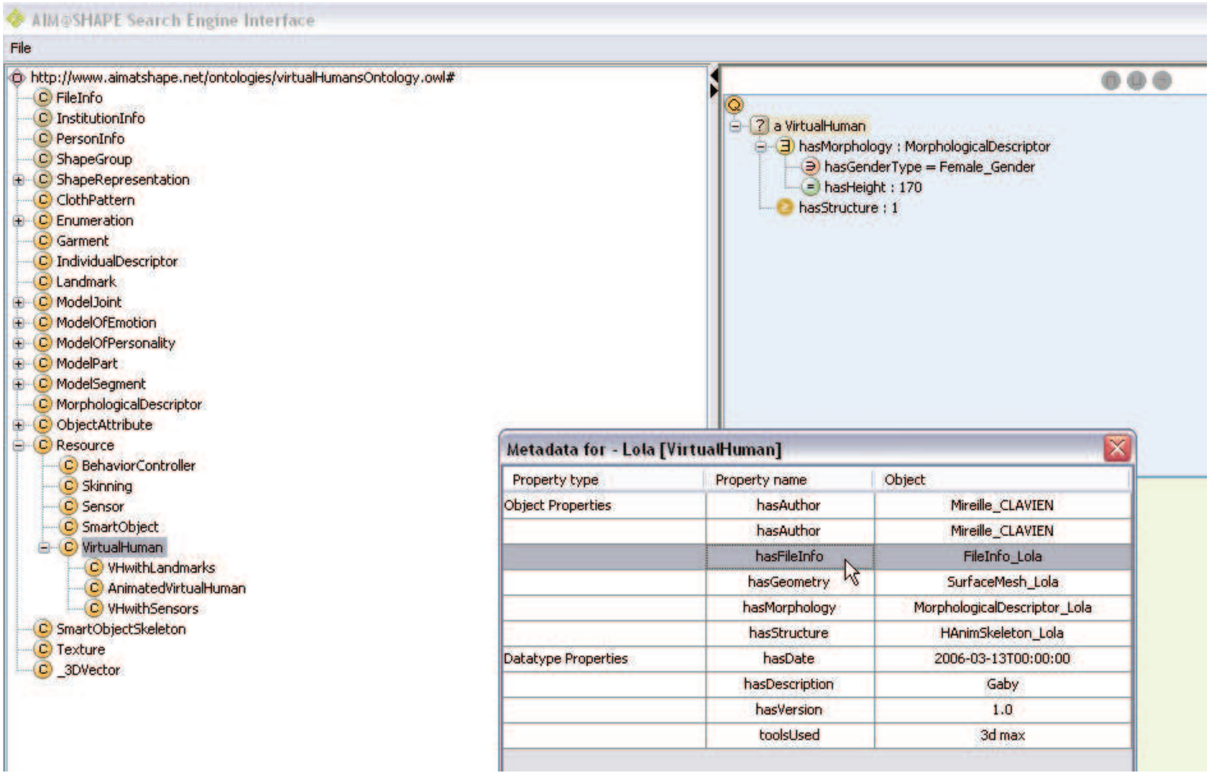

Figure 5.18: Search results of a VH woman using the search engine.

the animation.

The second scenario considers a programmer who wants to create a game using a male character. This character should be an animatable character, which means that should have a skeleton. Therefore, the user performs the search in the Figure 5.19.

From the search of male characters with skeletal structure, he chooses Keith. To make the desired animation of Keith, he wants to use Inverse Kinematics technique. He needs to know names of the Joints that can be used as end-effectors. The end-effectors are Human Joints localized at the end of the skeletal structure, and do not have children. He performs in the ontology the query shown in the Figure 5.20. This query searches in Keith's skeletal structure, for all the joints that are children of Root Joint and do not have children. The results are the names of the joints that will be used in the inverse kinematics as end effectors.

This scenario aims to prove that the previous knowledge of the metadata of the structure of a Virtual Human can help to find specific information used to implement an animation technique. The mentioned queries were executed using the search engine. The result of the second scenario in the search engine is presented in figure 5.21.

The validation of the VHO has been presented with the description of user scenarios and queering the ontology for specific features in the Virtual Humans. However, more population of the ontology is needed to be able to answer all possible scenarios. The ontology population is an important issue because of the granularity of the information provided by the VHO, for instance the skeletal structure of each Virtual Human which can be over a hundred instances of joints, sites, segments, etc. This amount of data cannot be inserted manually in the ontology, thus there are some semi-automatic and automatic metadata extraction tools presented in the appendix A.

SEARCH 1

Find VirtualHuman x?, FileInfo y?
hasMorphology
hasGender: Male
hasSkeleton >1

RESULT:

x? VirtualHuman_Gus
y? FileInfo_Tomy

x? VirtualHuman_Gino
y? FileInfo_Peter

x? VirtualHuman_Keith
y? FileInfo_Keith

Figure 5.19: Search of a Male Virtual Human with H-Anim skeleton.

SEARCH 2

From VirtualHuman
hasName = Keith
hasStructure = HAnimSkeleton
hasRootJoint = ModelJoint
hasChild = HumanJoint
NOT (hasChild = HumanJoint)

RESULT:

HumanJoint_Keith_l_midtarsal
HumanJoint_Keith_r_midtarsal
HumanJoint_Keith_l_wrist
HumanJoint_Keith_r_wrist
HumanJoint_Keith_skulbase

The animation using the l_wrist as end-efector
for raching objects.

Figure 5.20: Queries to search the end-effectors of a Virtual Human.

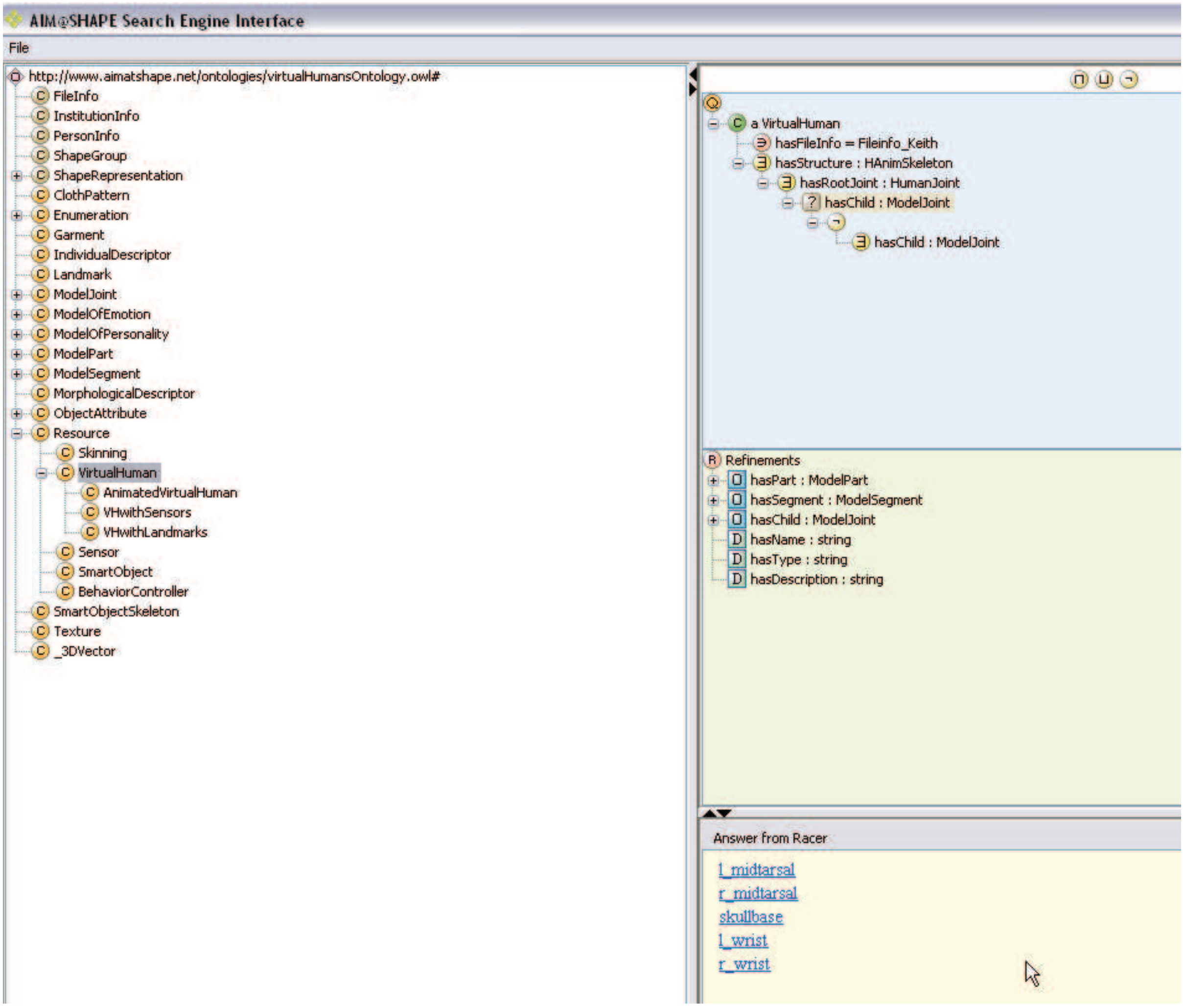

Figure 5.21: Search of end-effectors of a Virtual Human in the search engine.

5.5 Conclusions

This chapter presented the semantics for Virtual Humans that aims to cover the proposed domain: the description of creation of Virtual Humans and their features to populate virtual environments.

The creation process of Virtual Humans was described inside the SAP ontology which is specialized in the acquisition process of data for creating shapes. Thus, this ontology was complemented with concepts to describe the creation of human shapes and Motion Capture animation sequences.

The Virtual Human's semantic description is presented in the VHO. This ontology encloses the concepts and relations proposed to describe geometrical, structural and conceptual aspects of the Virtual Humans, including the software controllers used for their animation and population of virtual environments. Moreover, the proposed descriptions include the interaction of Virtual Humans with specific virtual objects such as garments and smart objects.

This semantic description is a promising alternative for modeling and managing the knowledge related to Virtual Humans. To demonstrate the usability and reusability of this ontology, the next part of this dissertation deals with its second research question: *How this representation can support the creation, population and control of inhabited virtual environments.*

Part III

Ontology Usability and Reusability

Chapter 6

Ontology for MPEG-4 Expressive Animation Synthesis

Authors of human expressive animation synthesis are closely attached to the MPEG-4 standard because it provides a structured control of character's face and body. Recent work has demonstrated that using MPEG-4 it is possible to create realistic facial expressions. Also the complexity of the body gestures can be studied in order to create synthetic animations for Virtual Humans. The knowledge behind this research would be better understood if disseminated using explicit representations such as ontologies.

This chapter presents an ontology that aims to describe this specialized knowledge. This ontology was created in collaboration with experts in the field from the NoE HUMAINE [113] project, and this results were published in [59] and in [58].

The ontology for MPEG-4 Expressive Animation Synthesis is an application ontology that reuses the Virtual Human Ontology (VHO) to create a very specific knowledge base. The domain is defined as the synthesis facial expressions [131] and body gestures [27] using MPEG-4 standard. These approaches are based on discrete and dimensional emotion representation models [39]. The advantage of the mentioned approaches is that real and naturalistic data captured with emotion induction techniques are used. Therefore, the obtained measurements are fairly more realistic than those portrayed in most facial expression databases.

For this ontology two kinds of scenarios are considered. The first one is for retrieving animation files that have been annotated with high level descriptors, e.g. searching for a several animation files that express a particular emotion in different levels. The second scenario is where the ontology can provide a kind of expertise in the knowledge domain. Based on these scenarios, some examples of competency questions that this ontology is able to answer are: What are the face animations for expressing sad emotion? What kinds of anger expression there are? What is the range of FAP values for the following emotions: worried, terrified? Given a set of FAP values what is the emotion that can be produced?

As the MPEG-4 Facial and Body Animation objects is used by this application ontology, the description of the main concepts of this standard are represented in the ontology.

6.1 MPEG-4 Face and Body Animation

MPEG-4 has been exploited because it makes it possible to have a structured control of the character. Some of the advantages of this standard are: very low bit rate coding, high quality and customizable rendering, scalability, etc. This subsection describes the parameters specified for Face and Body Animation in the framework of MPEG-4 standard.

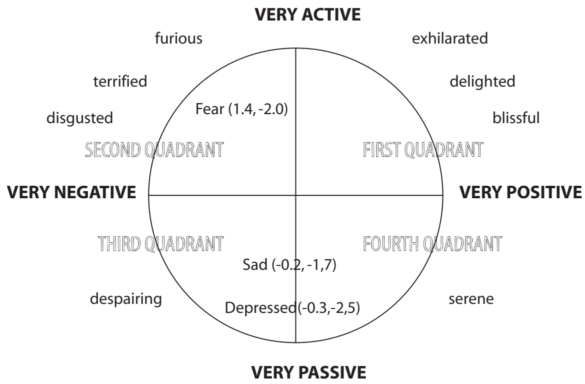

Figure 6.1: Activation-Evaluation space.

For facial definition, MPEG-4 specifies 84 feature points on the neutral face [146], which provides spatial reference for the Facial Animation Parameters (FAPs) definition. The FAP set contains two high-level parameters, visemes and expressions. Facial shape is defined by the Facial Definition Parameters (FDP) and Facial Deformation Tables (FDT) nodes. FDP are the feature points in the face shape and the FDT contains the deformation of the model face as a function of its animation. For animating a face, FAP node has the translations of feature points, expressed in FAP Units; they can produce high quality facial expressions and speech pronunciation. FAPs that can be used together are represented in groups in order to facilitate the animation [42].

To define the body in MPEG-4 the Body Definition Parameters (BDP) has the properties related to the topology, geometry and texture of each character. This definition is based in the hierarchical structure of H-Anim standard. The *Node* concept defined previously for the skeletal structure is similar to the BDP definition, thus the BDP and can be represented in the VHO. To provide the transformation values for the Virtual Human's skeletal structure, the Body Animation Parameters (BAP) are provided for each frame in an animation file.

The objective of this ontology is to describe a relationship between MPEG-4 animation sequences and emotional expressions. Next section describes the representation of emotional models in this ontology.

6.2 Emotion Representation Models

Depending on the context of interaction, one might prefer to choose a discrete emotion representation models, such as Ekman's [46], over a dimensional [163]. For example, if the objective of an application is classification of images or videos into specific categories, it is better to use discrete representations. On the other hand, dimensional models perform better when capturing subtle emotions (everyday human discourse), and can be differentiated between the two principal axes, e.g. in an anger detection application used in automated phone centers. Besides these representations, component process models, such as Marsella's [103] or Scherer's [134] appraisal checks, aim at mimicking human brain processes when confronted with new events. In these situations, several aspects of a stimulus are taken into account, such as its disposition against the aims of the subject, its novelty, etc. Although they are useful in modeling human-human discourse, there are several unclear aspects in these models that hamper the development of related computational models, such as timing or sequencing of processes.

6.3 Expressive Facial Animation Description

Virtual human's face is capable of having explicit synthetic expressions by itself within animations according to a given emotion. Ekman's Action Unit Code System [46] has been widely accepted to describe the six archetypal emotions, and it can be mapped into MPEG-4 FAPs. For each of the six basic emotions, a list of FAPs is associated to produce the specified emotion. However, this model is not sufficient to produce a varied range of emotions; thus, a more transparent and continuous representation of emotions needs to be integrated. Whissel's activation-evaluation space [163] is a simple circular representation that captures a wide range of emotions and simplifies them in two dimensions: activation and evaluation, depicted in figure 6.1. The vertical axis represents the activation value and the horizontal one the evaluation value. This representation is complemented with the Plutchik's [121] angular measures, which runs from Acceptance (0) to Disgust (180).

As general rule, one can define six general categories for facial expressions, each categorized by an archetypal emotion. Within these categories, expressions are described by different emotional intensities, as well as minor variations in expression details. The need for different Expression Profiles arises from the fact that the same emotion may be expressed via the face using different facial deformations, subject to personal traits or different feelings in expressivity. This concept is useful in both analysis and synthesis purposes, since it allows for personalized, diverse expressions, thus reducing the possibility of robot-like faces. The following subsections explains the profiles defined for archetypal and intermediate facial expressions. These profiles are the core knowledge that this ontology aims to capture.

6.3.1 Archetypal Expression Profile

In general, archetypal expressions can be uniformly recognized across cultures and they are therefore invaluable in trying to analyze the user's emotional state. However, these expressions occur rather infrequently, and in most cases emotions are expressed through a variation of a few discrete facial features related to particular FAPs.

Based on elements from psychological studies and from statistical analysis using FAPs, in [131], 25 archetypal facial expressions profiles were defined and grouped into six basic emotions. Table 6.1 presents the profiles Sadness and Fear expressions considering ranges of FAP values for the same expression. The FAPs are represented by the F and the number at which it belongs to, e.g. the FAP F_{19} is the *close_t_l_eyelid* (a complete list of FAP description can be found in [109]). In this case, Sadness has only one profile and Fear has nine. The complete list of all archetypal expression profiles and their FAPs ranges can be found in [131]. For the cases where an emotion has many profiles, it means that this emotion can be expressed differently in those different profiles. For example, the fear emotion profile zero describes an afraid person who has the bottom mid-lip raised, his eyes slightly closed and only the inner and middle part of his eyebrows raised; while the fourth fear profile describes a person who has his mouth and his eyes opened, his eyebrows raised and squeezed.

However, archetypal expressions are still not enough for the wide range of human expressions, and the creation of intermediate expression is proposed to cope this limitation.

6.3.2 Intermediate Expression Profile

The basic idea behind intermediate expressions is to create expressions by combining the FAPs employed in two archetypal ones. Creating profiles for an expression that cannot be clearly characterized as an archetypal one is not straightforward. Apart from estimating the range of variations of FAPs, one should first define which FAPs are involved in the particular intermediate expression.

The method proposed in [131], uses the Whissel's wheel [163] model to define the profiles of intermediate expressions. This model suggests that emotions can be represented as a point in a space. Therefore, FAPs that are common in both expressions are retained during synthesis, while FAPs used in only one

Profiles	FAPs and Range of Variation
Sadness (P_S^0)	F_{19} ∈[-265,-41], F_{20} ∈[-270,-52], F_{21} ∈[-265,-41], F_{22} ∈[-270,-52], F_{31} ∈[30,140], F_{32} ∈[26,134]
Fear ($P_F^{(0)}$)	F_3 ∈[102,480], F_5 ∈[83,353], F_{19} ∈[118,370], F_{20} ∈[121,377], F_{21} ∈[118,370], F_{22} ∈[121,377], F_{31} ∈[35,173], F_{32} ∈[39,183], F_{33} ∈[14,130], F_{34} ∈[15,135]
...	...
Fear ($P_F^{(4)}$)	F_3 ∈[400,560], F_5 ∈[-240,-160], F_{19} ∈[-630,-570], F_{20} ∈[-630,-570], F_{21} ∈[-630,-570], F_{22} ∈[-630,-570], F_{31} ∈[460,540], F_{32} ∈[460,540], F_{33} ∈[360,440], F_{34} ∈[360,440], F_{35} ∈[260,340], F_{36} ∈[260,340], F_{37} ∈[60,140], F_{38} ∈[60,140]
...	...

Table 6.1: Profile for the Archetypal Expression Sadness and Fear, which contains the corresponding FAPs with their range values.

expression are averaged with the respective neutral position. In the case of having FAPs in common, averaging of intensities usually favors the most exaggerated of the expressions that are combined, whereas FAPs with contradicting intensities are canceled out.

This is a rule-based technique for analysis and synthesis to merge profiles of archetypal expressions for creating intermediate ones. An example of the Intermediate expression depressed, which is created from the combination of Sad and Fear archetypal profiles is presented in the Table 6.2. Depressed profile(D_S) values are the range variation resulted from the combination of Sad (P_S) and Fear (P_F) archetypal profiles using the activation-evaluation measures. Not all the FAPs are presented in this table, but for a detailed explanation and reference to intermediate expressions look at [131].

Expression	Activ-Eval	F_3**min**	F_3**max**	F_5**min**	F_5**max**	F_{19}**min**	F_{19}**max**	...
Fear (P_F)	$(1.4, -2.0)$	400	560	-200	-160	-630	-570	...
Depressed (P_D)	$(-0.3, -2.5)$	160	230	-100	-65	-110	-310	...
Sad (P_S)	$(-2.0, 1.7)$	0	0	0	0	-265	-41	...

Table 6.2: Intermediate Profile of Depression and the archetypal expressions used to calculate it using the activation and evaluation measures.

The results of the merge procedure is presented in figure 6.2. A *Sadness* profile (open eyes, inner part of eyebrows slightly raised) is combined with a *Fear* profile (open mouth, wide open eyes, raised eyebrows). The result of such combination is different versions of *Depression* expression. In the expression presented in figure 6.2-(c), the mouth is not as open as in *Fear*, and the eyes and eyebrows values are lying between the values of the above-mentioned profiles, so eyes are more open than in *Fear* and less open than in *Sadness* and the same stands for eyebrows as well.

6.3.3 Ontology of MPEG-4 Facial Expression Profiles

Figure 6.3 presents the concepts for the MPEG-4 Face and Body Animation, where concepts from the VHO are used (*VirtualHumanAnimationSequece* and *VirtualHumanKeyFrame*). The *BodyAnimationParameters* and *FacialAnimationParameters* concepts are used to describe expressive animations. The geometrical aspects of MPEG-4 are already represented in the ontology using the structural description. The *FacialAnimationParameterDefinition* concept describes the necessary information of each FAP, and has the property *affectsLandmark** which links the animation with corresponding landmarks. The same happens with the *BodyAnimationParameterDefinition*, which has the property *affectsJointName* that links the animation parameter with a specific joint.

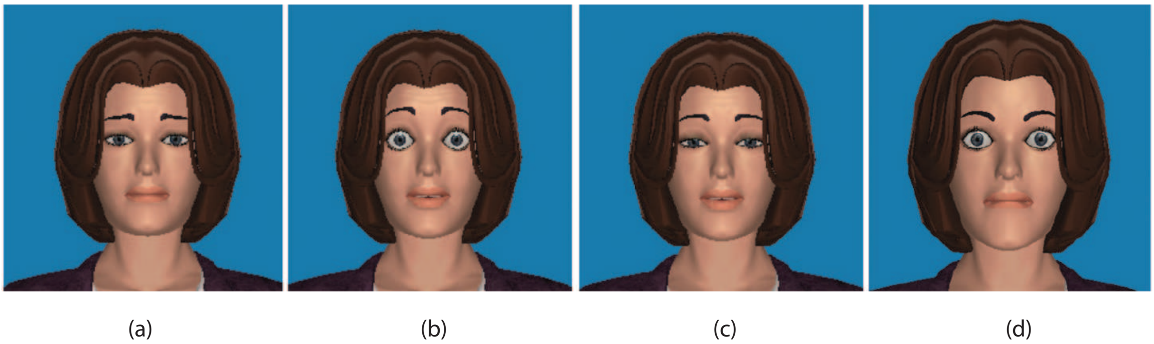

Figure 6.2: Example of graphical results for the expressions: (a) Sadness, (b) Fear, and (c),(d) Depression.

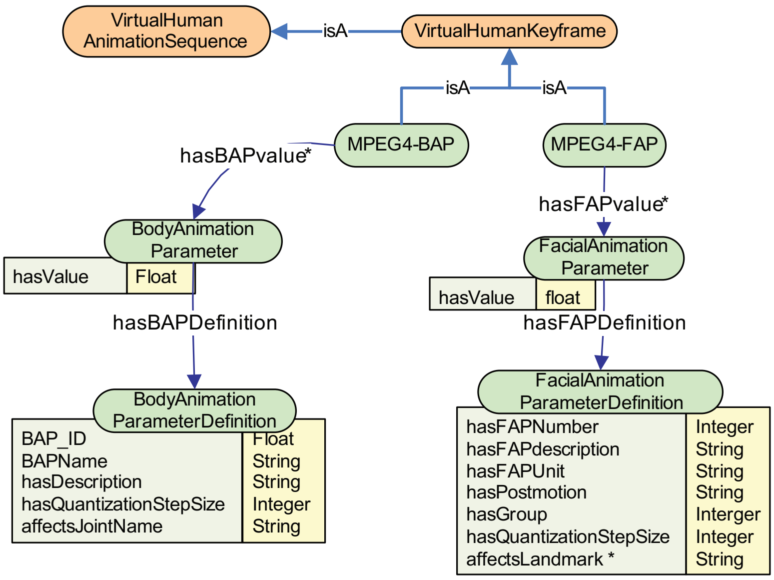

Figure 6.3: Description of concepts for MPEG-4 facial and body animation.

The ontology diagram of the representation of emotions is depicted in figure 6.4. It shows that one emotion can be modeled by one or more models of emotion. One model is the Action Unit Code System that associates an emotion with a set of Action Units that can also be mapped to their FAP sets counterpart. The other emotion model is the Wissel's wheel that gives activation and valence values to a given emotion. This last model is used in the next subsections to create expression profiles to describe animation sequences.

As already described, two types of Profiles are defined for facial expressions: Archetypal and Intermediate. Figure 6.5 presents the ontology diagram of the described profiles. In this diagram, an *AnimationSequence* is described by having an expression using the property *hasExpression* connected to *Expression* class, which also describes an *Emotion*. Facial expressions can be described using an archetypal or intermediate profile. The relation between profiles are defined by the following assertions defined in the ontology:

1. The Face expression can be defined by a profile Archetypal or Intermediate.
2. Each Intermediate Expression should enclose 2 archetypal expressions.
   $IntermediateProfile\ encloses\ ArchetypalProfile = 2$

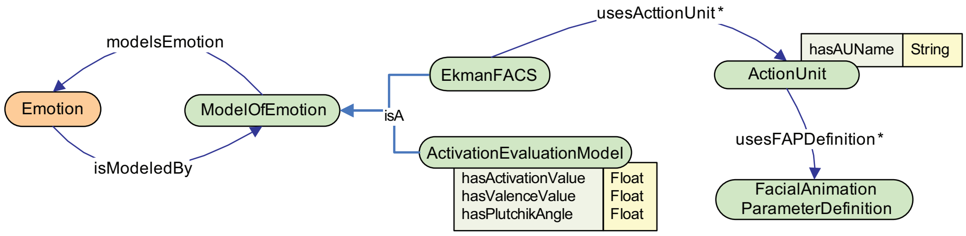

Figure 6.4: Semantics for Models of emotions.

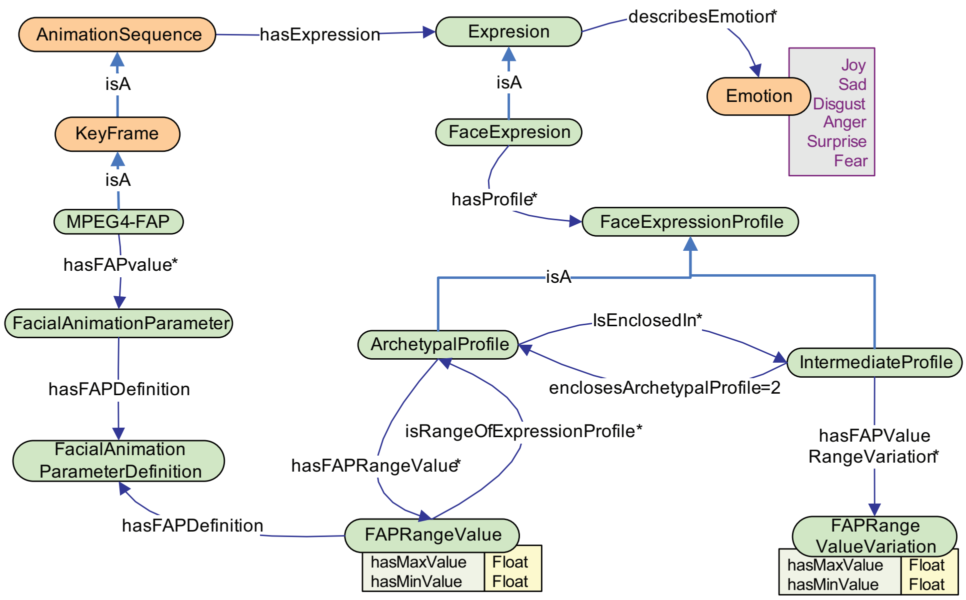

Figure 6.5: Semantics for the Emotional Facial Expression Profiles.

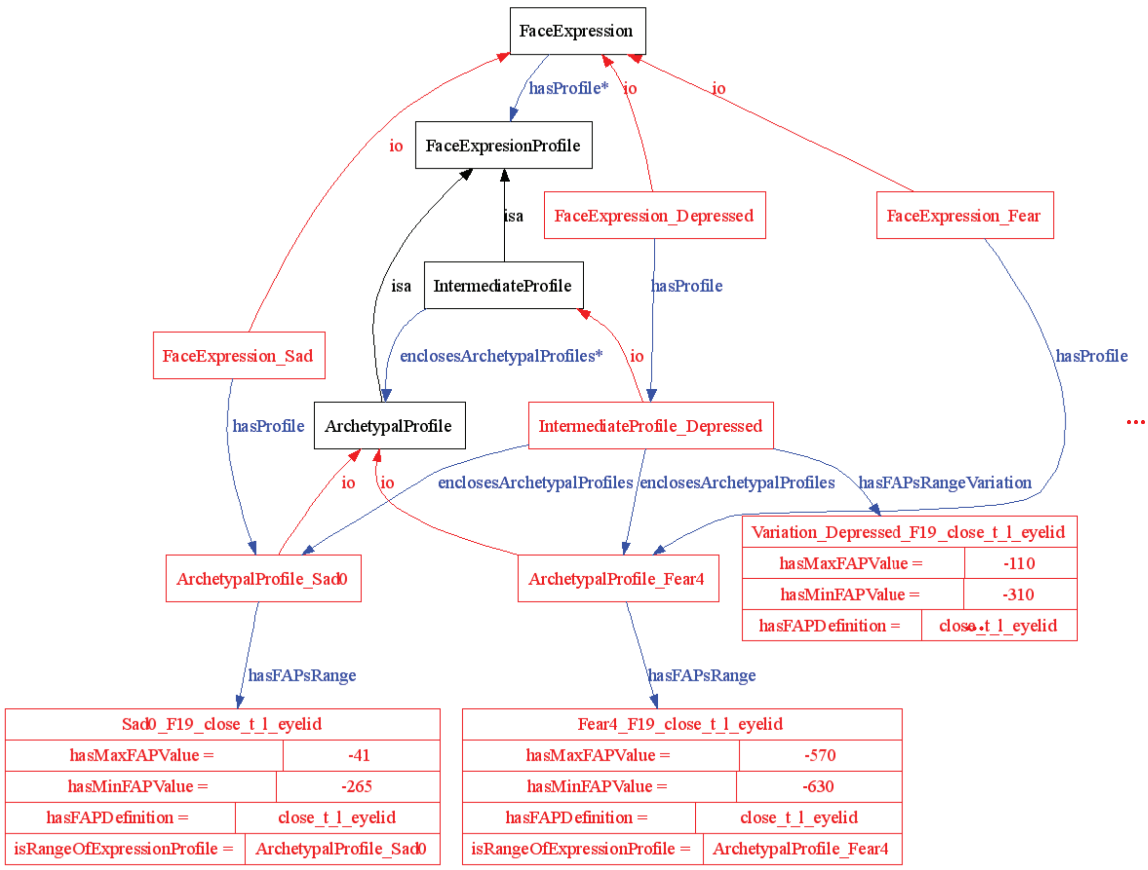

Figure 6.6: Example of profile population in Protege.

3. Each profile has a Range of FAPs associated (*FAPRangeValue*). The *FAPRangeValue* class contains the max and min FAPs values.

4. The Range of FAPs of Intermediate expressions is the union of the Ranges of the Archetypal expression that are enclosed by the Intermediate expression. This can be expressed as a restriction for the *hasFAPRangeValue* property as:

   $\forall\ hasFAPsRangeValue\ (FAPRangeValue\ \cap$
   $((\exists\ isRangeOfExpressionProfile\ ArchetypalProfile)\ \cap$
   $(isEnclosedBy\ IntermediateProfile)))$

5. As described in [130], Intermediate expressions FAP Range Variation is computed considering three possibilities: a) Mutually inclusive FAPs with different signs b) Mutually inclusive FAPs with same sign c) Mutually exclusive FAPs The computation of these values is through the expression generator (this issue will be explained in the next section entitled Rule Population)

An example of population of expression profiles in the ontology is presented in figure 6.6. This is a diagram of some instances for Fear and Sad archetypal emotional expression and Depression as intermediate expression accompanied by graphical results obtained using the presented method for synthesizing face expression.

Fear emotion has a face expression *FaceExpressionFear*. This face expression is defined as archetypal, and as a consequence has Archetypal Profiles defined: *ArchetypalProfile_Fear0*, *ArchetypalProfile_Fear4*.

Each profile has their FAP Range values obtained form [131]. Sad archetypal expression *ArchetypalProfile_Sad0*, is defined in the same way.

To define *Depression* as intermediate expression, in between of *Fear* and *Sadness*, the *Depression* Face Expression is an intermediate expression that encloses the Archetypal Expressions *Fear* and *Sadness*. The *FAPValueRange* of the Depression profile is implicitly declared as the union of the FAP range of Fear and Sad profiles. For this intermediate profile the *FAPValueRangeVariation* contains the FAP range values that were obtained from the intermediate expressions generator presented in [130]. In this example, the FAP 19 *close_t_l_eyelid* is contained in both archetypal expressions and has the same sign, which is the first case of the rules previously specified. In the same way all the possible facial expressions with their profiles should be integrated in the ontology.

The definition of these semantics described in this section allows us to retrieve animations that have been annotated using profiles with FAPs; and also to retrieve the knowledge covered in the synthesis of those facial expressions.

6.4 Expressive Body Animation Description

Body gestures in Virtual Humans are an important element to enhance realism. A good performance of body movements in is achieved by pre-constructed motion captured sequences. However, motion capture is an expensive and exhaustive work, consequently the reusability of animations is highly desired. For reusing specific animations it is necessary to annotate them with extra meta-data. This information should provide not only the kind of movement it performs, but also the way that it is performed in order to fit it inside a context.

Caridakis et al. presented in [27] approaches to synthesize a MPEG-4 based expressions using discrete and dimensional emotion representation models [39]. They have provided high level descriptors to communicative gesture animations and the possibility of enhancing animations with expressivity parameters. This section presents an ontology that helps to classify expressive body animations into the proposed emotional models and parameters that can be used to describe them.

6.4.1 Body Expressive Parameters Extraction

This ontology provides a classification of animation sequences of gestures by associating them to emotion quadrant of Whissel's model. For example, hand clapping gesture can be associated to Joy and excitation, arms crossing can be associated to anger or unconformity, etc. This gesture classification process was derived from captured videos of acted session presented in [27]. Those sessions included 7 participants, where each one performed 7 gestures. Each gesture was performed several times with the student-actor impersonating different situations. After, these gestures are classified as presented in the table 6.3. This classification uses the Whissel's model, where gestures are assigned within one or more quadrants of the activation-evaluation space; this means that those gestures can be considered as part of all that emotional scenarios.

The classification of gestures makes possible to retrieve gestures that can be applied inside an emotional state context.

However, the literature of perception studies [71], [162] shows that expressive communicative behavior can be described in six dimensions; which can be used to enhance animations with more expressive gestures. The dimensions of expressivity are:

Overall activation is considered as the quantity of movement during a conversational turn

Spatial extent is modeled by expanding or condensing the entire space in front of the agent that is used for gesturing.

Gesture	**Whissel Quadrant**
explain	(0,0), (+,+),(-,+),(-,-)
oh my God (both hands over head)	(+,+), (-,+)
leave me alone	(-,+), (-,-)
raise hand (draw attention)	(0,0), (+,+),(-,-)
bored (one hand under chin)	(-,-)
wave	(0,0), (+,+),(-,+),(-,-)
clap	(0,0), (+,+),(-,+),(-,-)

Table 6.3: Classification of gestures in a Whissel's wheel quadrant.

Temporal parameter of the gesture determines the speed of the arm movement of a gesture's meaning carrying stroke phase and also signifies the duration of movements (e.g., quick versus sustained actions)

Fluidity differentiates smooth/graceful from sudden/jerky ones.

Power/Energy is identical with the first derivative of the motion vectors calculated in the first steps.

Repetition is the repetition of a pattern in the gesture.

Each dimension of expression can take place during the different phases of the gesture. For an arm gesture, expressivity works at the level of the phases of the gesture: for example the preparation phase, the stroke, the hold as well as on the way two gestures are co-articulated [71], [54]. For each emotion there are several values of each dimension.

For example, in order to complete the animation of the gesture "wave" with joy emotion, apart from a profile of expression joy, the BAPs values of gesture "wave" are multiplied by the appropriate values of expressivity parameters. The frames resulting of applying this mechanism oar shown in figure 6.7. Some of the values for emotion Joy can be found in Table 6.4.

Overall Activation	Spatial Extent max	Spatial Extent mean	Temporal	Fluidity	Power	Repetitivity
3164.57	2.52	1.8236	71	0.93	44.57	0
2389.68	3.07	1.59	71	1.03	33.66	0
3599.99	8.53	4.14	81	0.91	44.44	0
...						

Table 6.4: Values of expressivity parameters for the first quadrant (joy, surprise) of Whissel's wheel.

The first two frames are the animation without expression, and the third and fourth are the same frames with joy expression. Details about the actual implementation can be found in [99].

6.4.2 Ontology of MPEG-4 Body Expressions

The representation of the animations within their classification and the expressive parameters is presented inside the ontology in figure 6.8. Each animation is categorized inside one or more *WhisselWheelQuadrant*, and it is the same for each emotion modeled in the *WhisselEmotionModel*, where each emotion belongs to a one quadrant. Each modeled emotion has *ExpresivityBodyParameter* values associated. Therefore, an animation that belongs to a quadrant can be enriched with the expressive parameters of the emotions that are in the same quadrant.

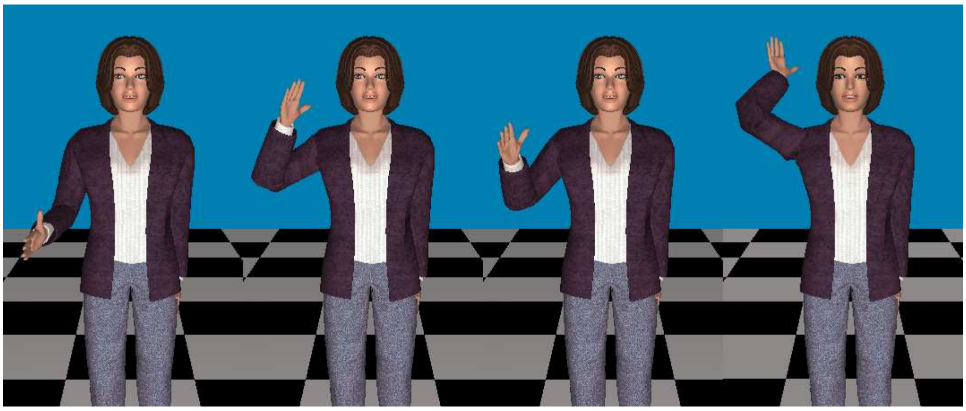

Figure 6.7: Frames of normal “Wave” gesture in the first two frames, and the second two, the same frames with joy expression.

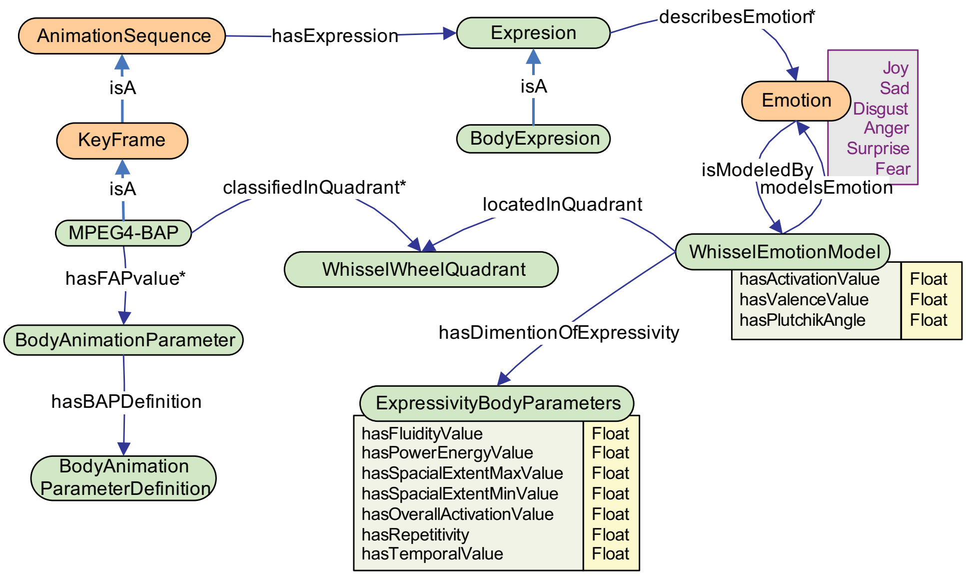

Figure 6.8: Diagram of the body expression ontology.

The population of the ontology included animation gestures presented in table 6.3: explain, "oh my God" (both hands over head), "wave", etc. Each animation is associated to their corresponding quadrant of Whissel's wheel, and their corresponding dimensions of expressivity.

In this ontology, animations can be classified into a model of emotion and describes also parameters of expressivity used to enhance the expressivity of animations. The structure of this ontology allows for retrieving animations that have been annotated with emotional descriptors under the presented psychological model of emotion; and also allows retrieving the knowledge of the emotional expressive enrichment of animations.

6.4.3 Querying the Ontology

To be able to extract information from ontologies is by making queries. This section presents some queries performed using the plug-in new Racer Query Language interface [110] for OWL ontologies in Protégé [127]. RacerPro [129] is used for providing reasoning when querying the ontology using nRQL.

The following are examples of competency questions translated in nRQL and the results obtained after the query.

What are the animations that express joy? (Animations that belongs to the same quadrant as joy emotion)

```
(retrieve (?c) (and (?a |WhisselWheel|)
(?a ?b|isLocatedInQuadrant|)
(?b |WhisselWheelQuadrant|)
(?b ?c |hasAnimationRelated|)
(?c |BodyAnimation|)
(?a (some |modelsEmotion||CONCEPT-FOR-Joy|))))
```

Result:

```
clapping
explain
oh_my_God
right_hand_wave
```

What are the parameters of expressivity to enhance the expressivity of an animation with joy?

```
(retrieve (?b) (and (?a |WhisselWheel|)
(?a ?b |hasDimentionOfExpresivity|)
(?b |ExpressivityBodyParameters|)
(?a (some |modelsEmotion||CONCEPT-FOR-Joy|))))
```

Result:

```
ExpressivityBodyParameters_Joy2
ExpressivityBodyParameters_Joy1 ...

Each of these last results has a set of the six parameters of
expressivity.
```

6.5 Conclusions

This chapter presented an ontology that represents the knowledge of specialized research on facial and body expressive animation within MPEG-4 framework. This application ontology reuses the VHO and provides new concepts for the new domain. Facial expressions have been represented in a form of archetypal and intermediate profiles that describe the synthesis of the emotions for Virtual Humans. The body expressions are classified into a model of emotion representation and their features are extracted.

The structure of this ontology allows retrieving animations that have been annotated under this structure. Moreover, it makes explicit the expertise covered in the synthesis of those expressions, for instance, to search for specific parameter values to generate certain kind of expression, these parameters can be: feature points involved, range of their FAPs values, emotion-activation values, etc.

The main issue experienced when adding semantics to a knowledge domain is the extraction of meta-data and the interconnection with existing knowledge sources, such as audiovisual databases with expressive material. To solve these problem, automatic and semi automatic meta-data extraction tools where developed to be used in specific cases. These tools are described in the appendix A.

Next chapters present a different usability scenarios of the VHO. On is to support the population of virtual environments and the other is to improve the realistic behavior of Virtual Humans.

Chapter 7

Visual Creation of Inhabited 3D Environments

Creating Virtual Reality (VR) applications requires experience on computer graphics, human-computer interaction, artificial intelligence, animation techniques, etc. Moreover, good programming skills are needed to put together all needed components. The fact of being dependent on expert programmers when implementing a VR application can be an obstacle for designers and other creative members of the production team [31].

Various software libraries and development environments have been created to ease this integration task. However, programming skill are still required in order to script animation sequences, set up interaction paradigms or program autonomous characters.

This chapter presents an approach based on a semantic representation of the components of a virtual environment: virtual entities (objects and characters) and animation algorithms (behavior controllers for autonomous Virtual Humans). The semantic representation is an the ontology of Virtual Humans, which functions as knowledge base to retrieve and configure Virtual Humans and their controllers.

In combination with the ontology, a visual programming interface is used to facilitate setting up an interactive virtual environments with autonomous characters without the need to write a single line of code. It also allows to represent spatial relations between objects in the scene, and plug animation controllers to virtual objects.

Next section describes some authoring tools for virtual environments, and how they represent the components of a 3D scene and what kind of information/knowledge is exposed to the user.

7.1 Related work

Some domain specific tools have been created to assist non-experts on the creation of virtual worlds. For example, the authoring tool for 3D online expositions presented by Costagliola et al. [38]. This tool used text based or iconic based interfaces to set up their scenarios.

More generic tools propose the use of scripting and XML languages [81] [117]. However, these systems do not offer scalability. In [63], authors presented an authoring tool that helped to accelerate creation of content using phyton scripting with some extension mechanisms. Ponder et al. [122] presented a development framework that uses python scripts to configure different libraries and extend the system functionality. The developer is able to create scenes populated with autonomous Virtual Humans by means of short programs (python scripts). A problem with scripts when using wrapper functions, like those implemented in Python language, a good documentation of functions is needed, and users have to get familiar with the initialization and internal workflow of the 3D engine. This represents an important learning curve. Moreover, people interested in creating a VR application needs some programming skills.

Commercial systems such as Virtools[1] and Quest 3D[2] offer a graphical schematic interface to navigate and manipulate entities in the 3D scene. This is in fact a visual representation of the scene graph: a hierarchical data structure representing the spatial relationships between 3D objects. 3D Modeling and animation tools such as Maya[3] and 3DS Max[4], also provide this kind of visual scene graphs, which are helpful when managing the components of a complex scene.

Mentioned visual representations focus on the geometric aspects of virtual entities. These representations allow us to see how elements like virtual characters are composed (skeleton, hierarchy of joints and segments, etc.), and they allow to change some of their properties (size, position, texture, etc). Complex information such as animation algorithms to be used can be manipulated only through scripts. In order to setup interactive behavior, once again, users still need a minimum knowledge of programming.

To overcome the necessity of specific knowledge, an intermediary representation that captures relevant concepts and knowledge associated to the creation of a virtual environment. Ontologies have been successfully applied to represent the knowledge and concepts of specific domains. The description of 3D items inside an ontology has being already presented in several works. The OntologyX3D [79] represents very specific and low-level concepts associated to a 3D scene.

Ontologies also have the power to represent high level properties and knowledge associated to the entities composing a VE, including geometric properties. For example, recent applications in product design aim to capture conceptual functionalities associated to 3D shapes, supporting collaboration in the design process [32].

Considering high-level properties is very important when developing a virtual environment. When creating behavioral virtual humans, information such the number of bones and degrees of freedom (DOF) of a character is not of interest, but rather its personality, the description of its emotional state and the animations that can be used to express/reflect these properties.

Next section describes the approach followed to represent high-level properties and knowledge associated to entities in a virtual environment (VE).

7.2 Semantics for Inhabited 3D Environments

The concept of Semantic Virtual Environments [68] has been defined with the goal of creating environments that can reuse digital items, and be scaled in functionality. This approach is very helpful to describe complex environments based on the semantic description of their 3D components. However, the integration of complex entities such a Virtual Humans, requires for more specialized description. The semantics presented in the chapter 5.2, are used to describe Virtual Humans as an active semantic entity with features, functionalities and interaction skills.

Figure 7.1 presents a diagram of the VHO extended with the necessary concepts to develop this application. The main class called *Resource* represents the items that the user can place, translate and rotate in the 3D scene. There are three kinds of Resources: Virtual Humans, Objects and Scenarios. Virtual Humans have three main properties: *hasIndividualDescription* (emotional state and personality), *hasMorphologyDescription* and *hasGeometry*. The description of the *VirtualHuman* is extended with the *hasAction* property, which associates Virtual Humans with *VirtualHumanActions*. These actions can be related to another resource, for example: *VirtualHuman hasAction LookAt targets VirtualHuman*. The *VirtualHumanActions* are functionalities that use specific *VirtualHumanControllers*. Controllers represent an implementation available in the 3D engine; in this case they are specific to the used system [118].

[1] S.A. Virtools. http://www.virtools.com.
[2] Quest 3D. http://www.quest3d.com.
[3] Autodesk Maya. http://www.autodesk.com/maya.
[4] Autodesk 3D Studio Max. http://www.autodesk.com/3dsmax.

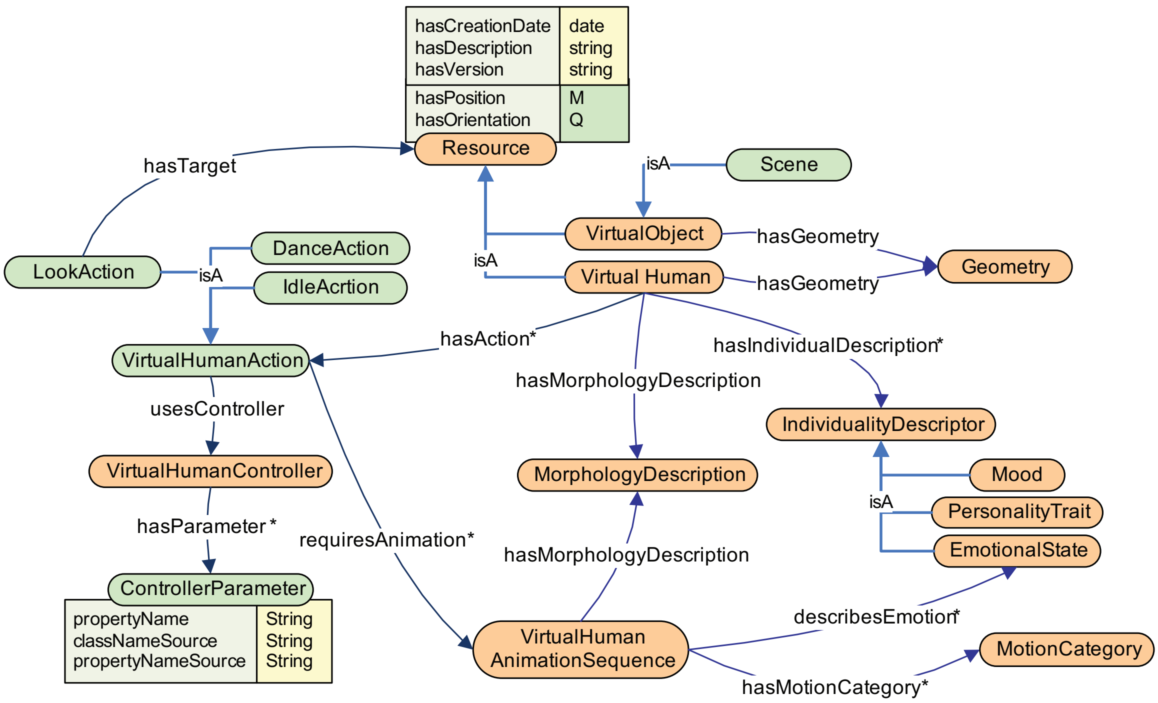

Figure 7.1: Virtual Human Ontology extended.

Figure 7.2 presents an example of the virtual human action *DanceAction*. This action has the property *usesAnimation* with the restriction *some Animation that hasCategory Dance*. The controller that this action uses is *KeyFramePlayer* whose function has the parameters *VirtualHumanName* and *AnimationFile*.

The Parameters of a controller are also described, and have the following properties that are dependent of the Action they are used for:

propertyName gives the name of the action property that establishes the relationship between the instance and the property of interest.

classNameSource is the name of the class from where the parameter will be taken.

propertyNameSource the name of the property that has the value for the parameter.

Parameters' descriptions are used to create messages from the 2D to the 3D engine. This is described in the following section.

The ontology also describes other constrains that can be used as filters in the application. For example, if a Virtual Human has a defined emotional state, then the animations to choose are limited to those that express the same emotion. The same happens with the morphology description, the animations available for a specific character are limited to those suitable for its particular morphology.

Additionally, the ontology is used to define all functionalities provided by the 3D engine under use. Different 3D engines can be utilized, it is necessary just to configure in the ontology the corresponding meta-data to the specific controllers implemented in the engine. The following section describes how these functionalities are exposed and used in the visual programming language.

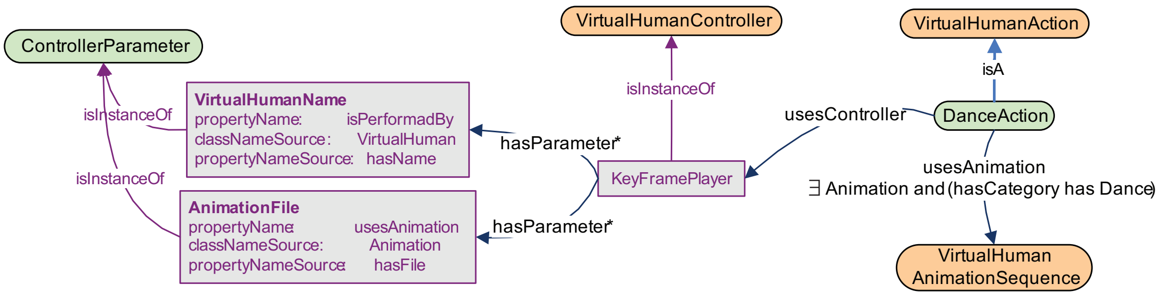

Figure 7.2: Description of Virtual Human Actions and their controllers in the ontology

7.3 System Description

This section describes the visual programming language used to represent the 3D scene with the help of the ontology. Followed by the system architecture.

7.3.1 Visual Programming Language

Visual Programming languages permit to manipulate elements graphically in stead of using text. Elements commonly used include: boxes, arrows, cycles, etc. The implementation proposed is not hierarchical as is the case of the applications described in the section 7.1. This visual representation consists on a 2D projection of the 3D scene and includes a graphical representation of the interactions between elements, which is closer to an ontological representation.

Figure 7.3 shows a screen shot of the interface created using the NetBeans Visual Library[5], which offers support for visual modeling. The main component is the Graph Scene, on top of which several *Widget Layers* are added. Each of those layers have a different purpose: adding widgets, making connections, defining the background, moving widgets, etc.

When the ontology is loaded, the instances of the subclasses of the *Resource* class that have a geometrical representation are shown. These available resources presented inside tree structure corresponds to the class hierarchy (fig. 7.3-1). The user can select a resource from the tree and place it into the scene. This action produces the creation of a widget in the 2D scene, and the resource is loaded and rendered in the 3D scene in real time. The bidimensional space of widgets represents the $[x, z]$ plane of the 3D scene; thus, the user can place objects intuitively as if he was looking at the scene from above. There are more controls provided for controlling the rest of the transformations (y position and rotations) (fig. 7.3-2).

Each subclass derived from the *Resource* class has specialized properties that describe them. When a resource is selected, its properties are displayed in the right box of the interface (fig. 7.3-3).

To define actions between a Virtual Humans and other resources, the user can draw arrows between them (fig. 7.3-4) through the connection layer. Actions that are not related to other resources are displayed when the user double click on them (fig. 7.3-6). Each time the user creates an action for a virtual human, the action is stored in a queue. When the user plays the scene, the scheduled actions are performed (fig. 7.3-5).

Each time the user interacts through the visual programming interface, the system sends messages to the 3D engine in real time. These messages are created using a communication protocol presented in table 7.1. Messages for the programmed actions are created using the controllers' description in the ontology. For example, if the user creates a relationship *VirtualHuman hasAction Interact Gaze with VirtualObject Plant*, the message is the concatenation of the function's parameters described in the ontology.

[5]NetBeans Visual Library. http://graph.netbeans.org/

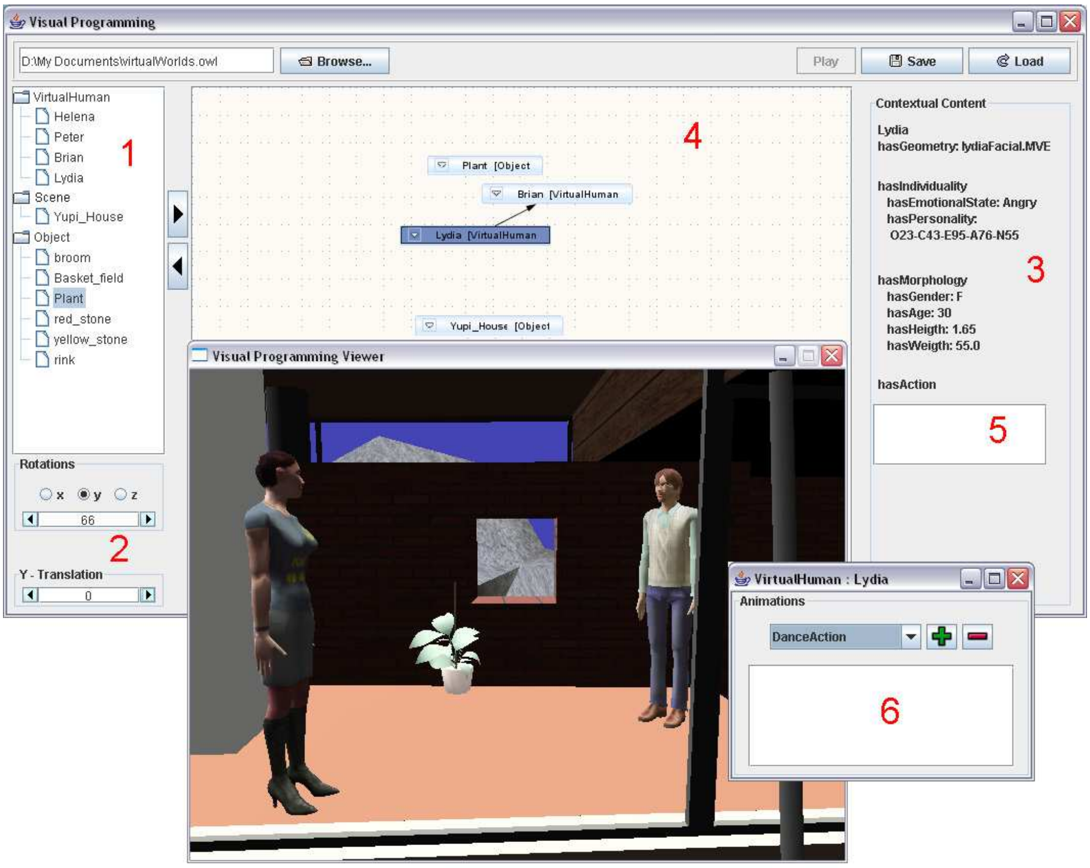

Figure 7.3: 2D representation if the Virtual Environment using Visual Programming Interface

Function	Message
Quit	0
Add Resource	1:name:source: orientation:posx:posz
Remove Resource	2:name
Modify Resource Properties	3p:name:posx:posz 3y:name:posy 3r:(x/y/z):name:angle
Add/Remove Action	4(a/r):functionName:parameters
Play Animation	5

Table 7.1: Communication protocol between visual and 3D applications.

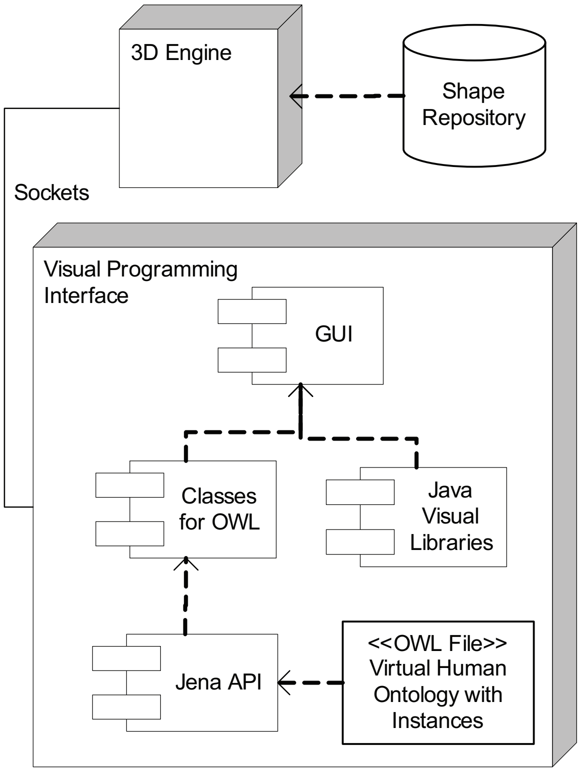

Figure 7.4: System Architecture

The following subsection describes the technical details of the implementation of this application.

7.3.2 System Architecture

The Jena API [104] is used to access to the ontology. Each concept form the ontology (OWL Classes) are implemented using java classes in the Visual Programming Interface. These classes are rendered in the GUI using the open source NetBeans, and can be manipulated with the visual programming Language as described before. The 3D engine used [118] supports loading Virtual Humans, and has implemented the functionalities of Key Frame Player and Gaze Controller. Figure 7.4 presents a diagram of the system architecture.

The modularity of this system makes the 2D interface completely independent from the 3D engine. The only implementation that needs to be done is the communication between the GUI and the 3D engine, where the 3D engine should support the functionalities exposed in the ontology.

7.4 Creating a 3D scene

The creation of a 3D scene requires an ontology populated with the instances of 3D items, such as Virtual Humans and 3D objects available in a data repository. The ontology should also contain instances describing the controllers (functions) provided by the 3D engine to be used.

The 3D engine used has two controllers available: *KeyFramePlayer* and *GazeController*. These controllers cans support the following actions: *TalkAction*, *IdleAction* and *LookAtAction*. These actions can be performed using the defined controllers with the restrictions described in section 7.2. *Talk* and

Figure 7.5: Animation sequence of Virtual Humans. Left:Lydia talks and Peter is Idle. Center: Peter looks at the Plant. Right: Lydia looks at the plant.

Idle actions use the *KeyFramePlayer* controller; and they use animations that belong to the Talk and Idle motion categories respectively. The animation to play is chosen randomly form the available animations that accomplish the defined restrictions.

Once the ontology has been populated, it can be loaded in the application, and the user can start placing objects and Virtual Humans, and programming actions. The figure 7.5, depicts a scene with two Virtual Humans, *Lydia* and *Brian*, and two 3D objets, *YupiHouse* and *Plant*. In this scene *Lydia* has the *TalkAction* and *LookAtAction* to the *Plant*; and Brian has the *IdleAction* and *LookAtAction* to the Plant.

Setting up this scene does not take more than a few minutes. Until now the application allows for creating quite simple scenarios. However, the user can define many different actions depending on the controls he have implemented in the 3D engine and exposed through the ontology.

There are still some issues that can be solved in different ways. For example, handling animation duration, or animation blending, setting up a time-line, or having an event-based animation system, etc.

7.5 Conclusions

This chapter presented an application to easily create virtual environments, based on a Visual Programming paradigms supported by an ontology for Virtual Humans and 3D objects. The system aims to facilitate the creation of inhabited 3D environments, in particular for users with none or minimal programming skills. The visual programming paradigm is used to represent 3D entities and their relationships on a 2D plane, to facilitate their management.

The Virtual Human Ontology is extended to describe 3D scenes, and to describe actions for virtual humans that are implemented inside controllers. The controllers are also described in more detail by defining the functions they have exposed and parameters needed. The information about Virtual Humans given by the ontology is used to filter animations dependent on the morphology and emotional state.

Using the ontology is possible to program Virtual Humans in a higher level of semantics, and focus on higher-level tasks. The drawback of ontology use is that it requires good understanding of various specialized languages (OWL). Another problem is that in order to manipulate and populate an ontology, ad-hoc interfaces should be used (e.g. Protǵé) and they are difficult to understand by non-experts. Moreover, such ontology manipulation systems are not designed with VE development in mind, thus interfacing the knowledge in the ontology with a VE application is no trivial task. However, once the ontology is setup, the application provides a user-friendly tool for the development of VR applications.

Chapter 8

Simulation of Individual Reactive Behavior

The lack of believability of virtual humans (VH) is partially due to the challenge of producing varied animations that are coherent with the context and with the individual traits of each character. A particular situation is when these characters are not able to perform individualized reactions to unexpected events in the environment. In most of the cases VHs are animated with the same animation for one kind of reaction. One solution to individualize and vary animations is the use many animation sequences reproduced by real actors, to avoid having a limited selection of movements. This approach is used in the most popular interactive games. However, large animation repositories are difficult to handle and are not enough to achieve the impression of individuality.

This chapter parents an attempt to enhance the believability of Virtual Humans by enabling them to produce spontaneous reactions to unexpected events in a VE. An empirical experiment was conducted to verify the hypothesis that reactions are highly dependent on personality traits. This experiment consisted in observing peoples' reactions to similar stimuli, and the collection of their personality traits by applying a questionnaire [123]. This questionnaire obtained values for each dimension of the Five Factor Model (FFM) [105] of personality. However, not only personality influences the reaction, but also the emotional state. An emotion can be associated to the motion speed and certain hand or body gestures that reflect the current emotional state. Therefore, the working principle of this proposal is that: the specific reaction to a given event is defined as a function of personality; perception defines (selects) the basic characteristics of the reaction (general orientation of the gesture: gaze direction, body position, etc.); once the basic characteristics of the reaction are defined, emotional clues are applied to the motion to reflect the current emotional state.

Thus, a model of reactions is proposed to select a type of reaction based on personality traits. This is complemented with a computational model of emotion for synthetic characters which simulates changes of the emotional state. In fact, the reaction behavior is a composition of a reaction posture and segments of pre-recorded animations with emotional cues. All this information is organized inside the ontology of Virtual Humans, and extracted for the application.

The synthesis of reactive animation performs at run-time and combines two animation techniques: Inverse Kinematics (IK) and Key Frame Interpolation (KF). IK was used to create desired postures for the reaction, and KF was used to enhance animations with expressiveness. The ontology helps to structure the domain knowledge and serves to annotate all the data needed form the animation sequences. Several animations are classified and, those that contain expressive gesturers are annotated by segmenting movements into body parts. The ontology describes also Virtual Humans' individualities in terms of morphology, personality and emotion. The information in the ontology is extracted with queries and saved in XML for the application usage.

Next section overviews previous work that aim at producing realistic reactive behavior.

8.1 Related work

Since reactive animations are movements difficult to reproduce by hand due to their specificity and variability, they are usually created from motion capture data. To be able to perform a realistic reaction, a large database of motion capture data is required, together with a search engine to find the most suitable animation. For example, to produce realistic reactive animations out of motion capture data, some authors apply motion blending, dynamic constraints [172] or physical constraints [144]. Komura et al. [83] used motion capture mixed with momentum-based IK. IK was used to keep the balance of the character. On these approaches, authors deal with the passive effect of an impact. In [52], the realism of the reaction was improved by making the arms of the character react in a protective way when falling down. These animations are physically realistic, but not individualized.

Imparting individualities in Virtual Humans is by means of integrating computational models of psychology. The following section describes the adopted solution for the representation of these models.

8.2 Defining Individual Reactions for Virtual Humans

As described in the section 2.3, individual descriptors are composed by personality, emotion and moods. FFM has been integrated in several models for synthetic characters, e.g. [139] [88].

the FFM [105] personality model is used for the creation of a reactive model. FFM is one of the most popular personality models used for computation; it defines the following dimensions:

Openness refers to how willing people are to make adjustments in notions and activities in accordance with new ideas or situations

Conscientiousness high in Conscientiousness tend to be organized, thorough, and planful.

Extraversion defined as a trait characterized by a keen interest in other people and external events, and venturing forth with confidence into the unknown

Agreeableness measures how compatible people are with other people, or basically how able they are to get along with others

Neuroticism is a dimension of personality defined by stability and low anxiety at one end as opposed to instability and high anxiety at the other end

FFM is used in the experiment to analyze the influence of each personality factor on the type of reaction, which is explained right after.

8.2.1 Reaction Experiment

The experiment consisted in inducing participants to a spontaneous reaction and applying them a test to obtain their personality profile [123]. This test results in the quantification of previously described personality traits in a scale of 0 to 100. The conditions of the experiment were to have the person standing up with the body free of obstacles (like a back-pack or a purse), in front of a computer answering the personality test. After some time, the subject was called from the back and a ball was thrown towards him/her. Figure 8.1 shows photos from the videos taken during this experiment.

The reactions obtained varies in the same kind of stimuli, such as avoiding the ball, catching the ball, closing the eyes, contracting the upper body (a protective movement), etc. These movements were grouped in three types of reactions to be able to analyze the data:

Figure 8.1: Videos from the reaction experiment.

Description	Intercept/reject	Protect	Avoid
Step towards stimulus position	X		
Step backwards stimulus	X	X	X
Knees flexion		X	X
Pelvis Displacement	X	X	X
Hands position in stimulus	X		
Palm Orientation to stimulus		X	
Palm Orientation against stimulus		X	
Rise Lower arm		X	
Arms extension			X
Column inclination	X	X	X
Column rotation	X	X	X
Hiding Head		X	

Table 8.1: Movements observed for each type of reaction

Intercept : to catch the ball or to use the hands to avoid contact with the ball.

Avoid : to move the body in the opposite direction from where the stimulus is coming from.

Protect : to contract the chest, to protect the body from the impact of the ball, or close the eyes.

After observing the videos, simple movements that compose the types of reaction described above are identified as listed in the table 8.1. Some of them are common within types of reactions, and others are specific to a type of reaction. These movements by themselves do not mean anything; but when they are combined, and inside a context, the complete movement have a meaning.

Unfortunately, important movement's characteristics such as the power of the movement, the speed, etc, were not considered because it should take more specialized studies and a statistical classification would be not possible.

The statistical analysis performed using this data is called J48, which is an implementation of of Quinlan's model C4.5 [128]. This test finds the correlation between personality traits and reaction types. This analysis is used to predict responses on a categorical dependent variable from one or more predictor variables. Decision trees are easy to understand and easy to convert to a set of production rules. However, these algorithms are unstable because slight variations in the training data can result in different attribute selections and the effect can be significant since attribute choices affect all descendant subtrees. This algorithm works as follows:

1. Choose an attribute that best differentiates the output attribute values.
2. Create a separate tree branch for each value of the chosen attribute.
3. Divide the instances into subgroups so as to reflect the attribute values of the chosen node.

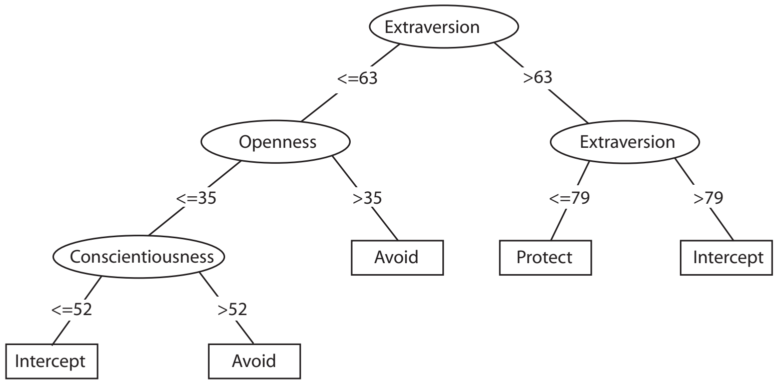

Figure 8.2: Decision tree for reactive motion.

4. For each subgroup, terminate the attribute selection process if:
   (a) All members of a subgroup have the same value for the output attribute, terminate the attribute selection process for the current path and label the branch on the current path with the specified value.
   (b) The subgroup contains a single node or no further distinguishing attributes can be determined. As in (a), label the branch with the output value seen by the majority of remaining instances.
5. For each subgroup created in (3) that has not been labeled as terminal, repeat the above process.

This classification was performed using the data mining WEKA software [164]. The decision tree obtained from this analysis, presented in figure 8.2, could classify correctly the majority of the instances of the experiments' data set (17/20).

This pragmatic solution provides different reactions depending on personality descriptors associated to virtual characters. However, one can argue about the coherent signification of the influence of some personality traits in the reaction, which is the case of conscientiousness. From the previous definition of conscientiousness, one can suppose that this trait does not influence spontaneous reactions because there is not much time to think about making a good or bad decision when performing a reaction. On the other hand, from the statistical point of view, there is a relationship between conscientiousness and the reaction. This is one of the problems when working with psychological approaches, personality is a very diffuse concept and the results obtained may not be applicable to other populations.

This model of reaction provides a type of reaction depending on a personality configuration, but the same reaction has to be also varied according to the emotional state. Next section describes how emotional state is used as modulator of the reaction movements.

8.2.2 Emotion Modulates Motion

The emotion as a dynamic factor in individuals can be used to modulate expressions during the animation. The model implemented in the simulation of reaction is derived from the Generic Personality and Emotion Model (PEModel) [47]. This model proposes to use vectors and matrices to represent the emotion using OCC [114] model of emotion, the Five Factor personality model [105]. Different vectors are considered: one vector for personality p with a dimension m where m is the number of personality dimensions; one vector for emotions e, with a dimension n where n is the number of emotions; another vector a of dimension m that contains emotion information of a desired change in the emotion, called emotion influence; and a vector w_t for the emotional state history. Therefore the model proposed to simulate emotion is given by the formula:

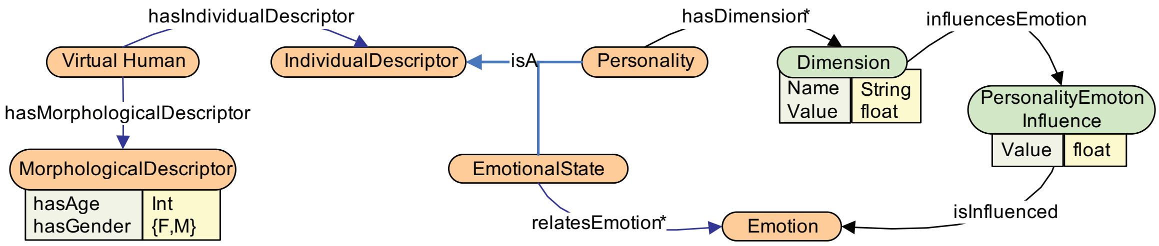

Figure 8.3: Virtual Human individuality.

$$e_{t+1} = e_t + \Psi(p, w_t, a) + \Omega(p, w_t) \tag{8.1}$$

The function Ψ, based on the personality p, the current emotional state history w_t and the emotion influence a, calculates the change of the emotional state as a result of the emotion influence. And the function Ω represents the internal change such as decay of emotional state.

For the implementation a matrix P_0 called *Personality-Emotion Influence Matrix* of dimension $[mxn]$ is created. This means that a given personality has associated different emotional states with varying intensities. The matrix P_0 is defined depending on the model of personality and emotion being used. Then, assuming that personality does not change, the product of P_0 and p give a vector u indicating the importance of each emotion depending on the personality dimensions. This vector is used to construct the diagonal of the matrix P, thus for each emotion the matrix P contains a value that defines how strong an emotion can be given by the personality p. Thus the function Ψ is defined by:

$$\Psi(p, w_t, a) = P \cdot a \tag{8.2}$$

To keep it simple, this implementation do not considering the emotion history w_t, nor the internal change decay (Ω function). The impact of not consider this would be a not so realistic performance in a long term in the animation.

The models described in these sections are the basis to create individualized Virtual Humans. A formal representation using ontologies is created to be able to practically implement these models. Ontologies fill the gap between the theory and the practice because they express the concepts and use this as a database for feeding the required information by the system. Next section describes this semantic layer.

8.3 Semantics for Reactive Motion Synthesis

As observed in the experiment, differences in reactions came out because of external and internal parameters. In this case, the external parameters are the direction and velocity of the stimulus. Internal parameters are comprised in the individuality of the person. Therefore, according to the external information and internal state of a person (individuality), an evaluation process takes place and decides the movement to perform.

However, most of the times it is not clear how to apply individual parameters into animations. Thus, the approach followed here is to express in a formal language how parameters that define personality and emotional states affect animation synthesis. These semantics has been formally represented in the ontology presented in the section 5.2.3 and reused for the reactive behavior simulation.

8.3.1 Representation of Individual Virtual Humans

The individuality of a virtual human can be given by visible and not visible aspects. The visible one is the appearance which is described in its morphology; and the not visible is its internal "configuration" described by the personality and emotional state. These aspects provide a semantic definition of virtual humans and should be taken into account at animation-time. Character's morphology constrains the animation in terms of anatomical measurements. Personality traits can determine the action to perform under certain circumstances. The emotional state defines a particular way of performing an action, and can communicate character's feelings. Finally, character morphology confines motion to some physical capabilities.

Personality can be described in terms of several dimensions. Each dimension represents the tendency of characters to perform some actions. Emotional state is related with at least one emotion. Figure 8.3 depicts these descriptions and the relationship of influence between personality and emotion.

8.3.2 Representation of Reaction Movement

To represent reactive animations it is proposed that an animation is composed of different movements applied in the different body parts, as represented in figure 8.4. The movements can be a segment of a Key Frame (KF), or an Inverse Kinematic posture (IK). The Virtual Humans geometry is driven by a skeletal structure, which has joints that act as articulations. The Skeleton can be segmented in body parts, and each body part has their own joints (i.e. the arm has shoulder, elbow and wrist joints). For each body part, different transformation can be applied from one of the two animation types: IK or KF.

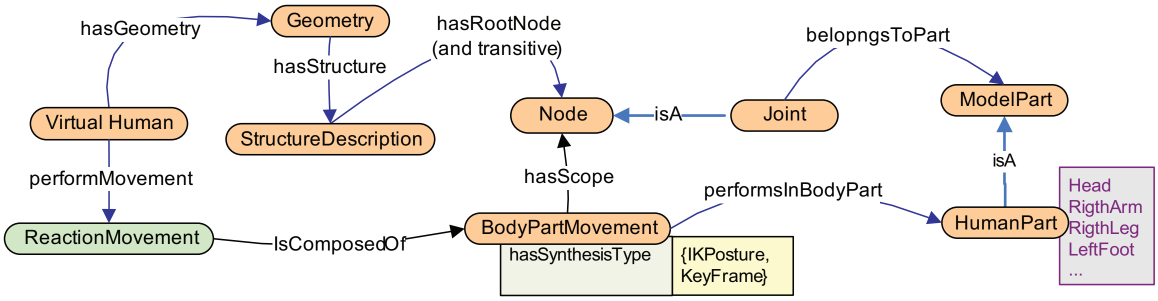

Figure 8.4: Character Animation Composition.

As presented in figure 8.5, the final posture of the reaction depends on the type of reaction: Intercept, Avoid or Protect. Each type of reaction is composed of different body movements with mutually exclusive body parts. The implementation of these IK postures are function of anatomical measurements and stimulus position and direction (e.g. to synthesize a step backwards one needs to know the legs' length in order to compute for example how much the root should move). Therefore, IK will produce a rough version of the intended animation. As IK-based animation tends to be very robotic, the reactive animation is complemented with KF animation segments that to enhance the expressivity and realism. These animations are applied to the body parts that are not used by the IK animations (i.e. Face, hands).

8.3.3 Representation of Expressive Animations

As expressive animation is very hard to create, KF animations that contain specific motions with emotional meaning are annotated and reused. Figure 8.6 describes the semantics used to annotate animations segmented in the body parts: face, head, upper body, lower body and hands. A *VirtualHumanKeyFrame*

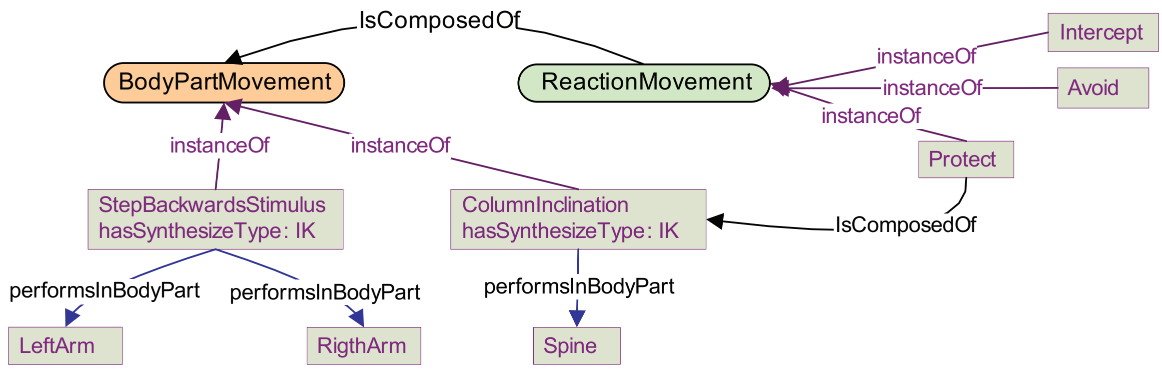

Figure 8.5: Semantics for Reactive synthetic movement using IK.

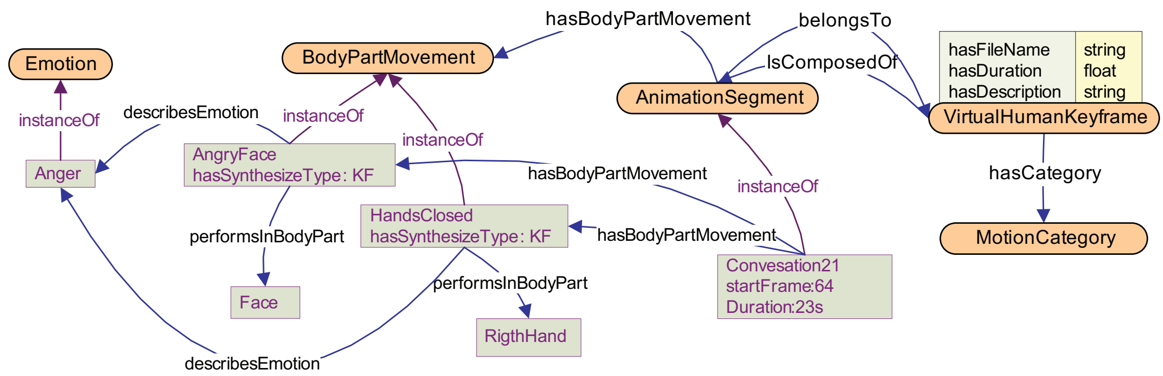

Figure 8.6: Description of emotional expressions using FK.

isComposedOf several *AnimationSegments* that *hasBodyPartMovement* that *describesEmotions*. An example of this relationship is an animation that *hasBodyPartMovement HandsClosed* that *describesEmotion Angry*. Moreover, KF animations are classified in motion categories as presented in the section 5.2.4.

The annotation criteria is based on body motion studies. Darwin started to study how emotions are communicated through the body [40], he said that there are specific movements in human beings that are performed depending on a specific emotional state. This approach has been supported by experimental research in body expression [162]. In table 8.2, there is the description that defines body part movements and their relation with an specific emotion. Thus, animation annotation is done by observing at specific body parts, looking for these specific movements.

Next section presents the implementation of a virtual environment. Virtual Humans are characterized by a specific personality and emotional state, which will define the way they react to different stimuli.

8.4 Application scenario

This scenario aims to illustrate the different reactions that Virtual Humans can perform accordingly to presented reaction model. The description of individualities is by means of the FFM personality model and six emotional states: Sad, Angry, Happy, Disgust, Fear and Surprised. Thus, arbitrary values are defined for each character and stored them in the ontology. This ontology contains also the annotation of 120 animations with body part movements that express any of the six emotional sates described above.

Emotion	Head	Upper Body	Hands	Movement Quality
Sad	Down, hanging	Contracted	-	Slow, passive
Angry	Erect	Expanded, rigid	Opening and closing	High movement
Happy	Up	Erect	-	
Disgust	Downward	Contracted, Arms crossed, rise shoulders	- movements	No expansive
Fear	-	Arms over the head, rise shoulders, arms against chess	Opening and closing	-

Table 8.2: List of emotions associated with specific body part movements.

Figure 8.7: Simulation of view, hearing and touch sensors with bounding boxes.

Protégé software[1] is used to retrieve this information from the ontology. This software allows making queries using SPARQL [161]. An example of a query is to extract the body part animations that have been annotated with a given emotion. The query results also include the file name, where the animations are contained, the start time and the duration. Query results are parsed by the 3D application which also extracts and uses the data.

The 3D engine used is MVISIO [118], where it is possible to visualize and animate H-Anim compliant characters. Basic perception sensors were developed to let characters perceive events and interact with nearby objects, shown in figure 8.7. These sensors are composed of bounding boxes attached to a specific joint: vision is the trapezoid, audition is the circle, and touch is the yellow rectangle. For the collision detection and physical simulation of some objects Aegia Physics Software [120] is used.

The character animation is performed with the use two animation libraries, a Key Frame player, and an Inverse Kinematics controller [25]. With IK the postures for reactions are synthesized (intercept, avoid and protect), and with KF the key frame animations are applied.

IK postures are composed by simple movements presented in the table 8.3. These simple movements were identified in the reaction experiment, and they are used to synthesize the animation. For each simple movement exist a function that modifies the IK parameters in order to synthesize it. These movements depend on some parameters, like a distance or direction; also they use some body parts and a type of transformation (P-position, O-orientation).

For example, the movement "hands position in stimulus", is function of the stimulus position. In this case, the end effectors of the right and left hand are used to set the position of the hands where the stimulus will be. Moreover, an aperture between the hands is given by the size of the stimulus. This

[1] Protégé (c) 2005. Stanford Medical Informatics. http://protege.stanford.edu/index.html

	Description (Parameters)	Body Parts (transformation)
Lower Body	Step (direction, distance)	Right Ankle (P), Left Ankle (P)
	Knees flexion (distance) - in Y axes	Pelvis (P), Right Ankle (P), Left Ankle (P)
	Pelvis displacement (direction, distance)	Pelvis (P), Right Ankle (P), Left Ankle (P)
	Pelvis rotation (angle)	Pelvis (O), Right Ankle, Left Ankle (P)
Upper Body	Hands position in stimuli (stimuliPosition, stimuliSize)	Right wrist (P), Left wrist (P)
	Palm Orientation to stimuli (stimuliPosition)	Right wrist (O), Left wrist (O)
	Palm Orientation against stimuli (stimuliPosition)	Right wrist, Left wrist (O)
	Rise Lower arms (stimuliPosition)	Right wrist (P), Left wrist (P)
	Arms extension (distance)	Right wrist (P), Left wrist (P)
	Column inclination (direction, distance, recruitment)	Skullbase (P)
	Column rotation (degrees, recruitment)	Skullbase (O)
	Hiding Head (distance)	Skullbase (P)

Table 8.3: Parametrization and scope of synthesized movements

movement is illustrated in figure 8.8-left. Other example is a movement that does not uses the hands position, for example "column inclination". For this movement, there is an end-effector in the spine. The spine position is function of a vector direction of the stimuli and a distance. A combination of "hands position in stimuli" and "column inclination" combination is depicted in figure 8.8-middle. Finally, to add movements to the lower body, for example the "knees flexion", end effectors on the legs are set, and in the pelvis for positioning the body. The parameter to this function is the distance that the pelvis has to go down. "Knees flexion" movent is integrated with the other described movements in figure 8.8-right.

However, the expressive animation composition consists in having one main animation in the Upper and/or Lower body and secondary animations segments (KF) in the hands, head and face. The secondary animations are for performing those animations that were annotated with emotional content. The main issue to solve when playing different animations in parallel is the coordination and synchronization - duration- of animations. Thus, the duration of the main animation is taken as the main duration. When

Figure 8.8: Blending simple movements to create reaction postures: positioning the hands in the stimuli (left), Inclination of the column (middle), Knees flexion movement (right).

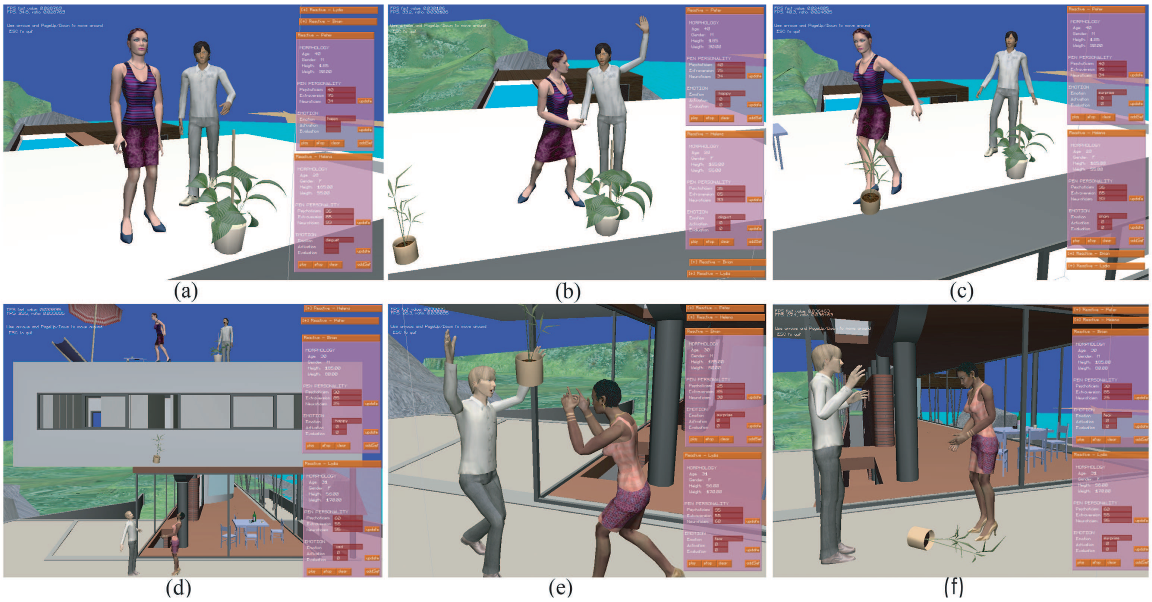

Figure 8.9: Reactive scenario sequence.

blending the two kinds of animation: KF and IK, there is no problem of coordination, it is possible to set a transition value from the actual posture to the final posture of the IK.

When characters are not reacting, they perform a predefined stack of actions: walk, talk, wait, etc, which are defined using the motion category. These actions can be also enhanced with expressive movements (animation segments that have emotional meaning). Spontaneous reactions happen when unexpected events -stimuli- occur in the environment. Each VH has a reactive controller, which performs the following sequence:

1. perceive an event, turn the head to look at it;
2. get a type of reaction depending on its personality using the decision tree model described in figure 8.2;
3. compute the reaction posture according to the stimulus position and set the end effectors of the arms and/or legs (body parts affected by IK);
4. update emotional state with the intensity of the predominant emotion depending on the personality of the VH, as described in section 8.2.2;
5. get expressive KF animations for the remaining body parts according to the current emotional state.

The scenario presents four Virtual Humans: Helena, Peter, Brian and Lydia (from top left to bottom right); they have different personalities and emotional states. The ideal[2] scenario is presented in figure 8.9. The animation starts when Peter comes behind Helena (a) and screams to say “hi!” to her (b). She reacts and becomes angry, turns away from Peter and kicks a pot on the roof (c). The pot falls down. As Brian and Lydia hear something they look up and realize that a pot is falling over them (d). Brian gets surprised and tries to catch the pot (e). Lydia feels fear and tries to protect her self (f).

[2] “Ideal”, because it does not happen like this all the time. VH actions depend on their current personality and emotion configurations

Figure 8.10: Results of different reaction postures and expressions.

To create a continuous animation each character has a list of actions to perform during the sequence, i.e. while any reactions are performed. All animations, including the reaction, are enhanced with expressiveness by adding secondary expressive animation segments, where selected segments correspond to the actual emotional state.

Figure 8.10 depict results when changing the personality values. Moreover, as the stimulus comes from a different direction the reaction simulation looks different every time.

8.5 Conclusions

This chapter presented an approach to improve believability of Virtual Humans by implementing a reactive behavior to unexpected events. This approach makes use of the Virtual Human Ontology to describe reactive animation composition.

This work proposes to individualize Virtual Humans' reactions by using a reaction model that consists on a decision tree based on a statistical analysis of real people reacting. This model is complemented with a model of emotion that updates emotional state. The synthesis of reaction animations is the combination of two animation techniques Key frame interpolation and inverse Kinematics. IK was used to produce reaction postures: intercept, avoid and protect; and KF was used to enhance animation with expressiveness in the body parts that were not used by IK.

The ontology is used to store and retrieve Virtual Human individual descriptors, reaction movements composition, and information about animation segments annotated with emotional meaning. Animation annotation resulted to be a very exhaustive work, thus future work should contemplate the annotation and classification of animations in an automatic or semi-automatic way. This may require pattern recognition techniques and the definition of specific criteria to qualify movements.

The reactive model presented is good enough to provide a simulation of different personalized reactions. However, the actual reaction types and animation sequences should be validated by psychological theories. Thanks to the semantic representation in the ontology, this work provides a generic platform that can be further parameterized to better fit different psychological models; and render virtual characters more and more expressive.

Chapter 9

Conclusions

The creation of Virtual Humans is a complex task because it involves several fields of knowledge. There are many existing technologies and techniques that have improved and automated the creation of these characters, and as a consequence produced large amounts of resources. The creation and population of new 3D environments has been somewhat hindered by the complexity of integrating and reusing geometrical models, animations and software tools.

The aim of the present work was to provide a formal description of Virtual Humans, their creation process, and their needed features to populate virtual environments. The approach followed was the integration of semantics by means of ontologies. This approach appeals the importance of human knowledge representation to solve current problems with complexity, reusability, and productivity of software applications.

9.1 Summary

The Part I of this dissertation summarized the knowledge that the an ontology for Virtual Humans aims to represent. This is the creation of human-like geometries and structural components, the animation methodologies, some aspects that should be considered to create believable characters such as individual traits, behavioral algorithms, etc. This part also presented the existing approaches to represent the knowledge, and some related work that aimed at representing aspects related to the Virtual Humans. The approach choose for this thesis is the creation of an ontology to represent the knowledge, and a methodology for ontology development was also presented.

The Part II of this work presented a context in which the Virtual Human Ontology (VHO) was developed in. This context was the AIM@SHAPE project, which allowed to bring together a group of experts from different fields related to shape development. This condition is important for the ontology development because the agreement among a group of experts is required for the definition of concepts and relationships among them. Several ontologies were developed within the AIM@SHAPE project: Common Shape Ontology (CSO), Common Tools Ontology (CTO), Shape Acquisition and Reconstruction Ontology (SAP), Product Design Ontology (PDO) and Virtual Human Ontology (VHO). A relevant performance indicators of these ontologies are the number of triples achieved which is 45000; and the number of instances per ontology: CSO: over 2500, CTO: over 700, VHO: over 200, SAP: over 150 and PDO: over 100.

Particulary the VHO developed for this dissertation shares specific and general knowledge with the other ontologies. The VHO is presented in detail in this part, and it describes the Virtual Human's characteristics described in the Part I. Also some validation scenarios of the VHO are described.

The Part III presented the developments to demonstrate the usability and reusability of the VHO. The first development is an application ontology that reused the VHO and represented more specific knowledge relates with the expressive facial and body animation of Virtual Humans. The second development used the VHO for an application that supported the creation of populated Virtual Environments. The third development used the ontology to organize the Virtual Human's individual descriptors to describe animation segments in order to improve the simulation of realistic behavior of Virtual Humans.

9.2 Semantics for Virtual Humans Overview

This section presents an overview about the information that can be found in the presented ontologies, trying to make a special emphasis in the Virtual Human description.

9.2.1 General Information of Virtual Humans

The common ontologies (CSO and CTO) describe general information about shapes, animations and software tools. This information includes authors, visions, files, owners, copyrights, etc.

The VHO provides a description of features needed by Virtual Humans to populate virtual environments. The Virtual Human class can be a description of a human-like geometry model, or can also be an abstract description of a Virtual Human. However, this ontology does not provide information about the creation process of its shape or its animations. This information is targeted by the SAP Ontology. The way to find the creation process of a Virtual Human in other ontologies is through the property *hasShapeRepresentation*, which points to its resource in the CSO (upper ontology), shared among the domain ontologies. Once the resource is identified, it is possible to search for it in the domain ontologies.

The following competency questions are some examples of the general aspects of Virtual Humans covered by the ontologies. The questions are followed by concepts and properties defined in the ontologies used to provide an answer.

- Is this model obtained by editing another model? In VHO: *VirtualHuman isBasedOn VirtualHuman*. In CSO: *ShapeRepresentation isOriginatedBy ShapeRepresentation*.
- What features have been changed on model X? This is a comparison of shape data 1 and 2 in SAP: *ShapeData(1) hasBeenAcquiredBy ProcessingSession hasPreviewSession hasInput ShapeData(2)*.
- What tools where involved in the synthesis/modification of this Virtual Human? In SAP: *ShapeData hasBeenAcquiredBy ProcessingSession hasProcessingSystem*.
- What is the height of the model? In VHO: *VirtualHumans hasMorrphology* that *hasAnthropometricMeasurement Height*.
- Is the model male or female?In VHO: *VirtualHumans hasMorrphology hasGender*.
- Is this VH a child or an adult? In VHO: *VirtualHumans hasMorrphology hasAge > 18*.
- Which are the Virtual Human that are fat/slim/short? In VHO: *VirtualHumans hasMorrphology hasWeight and hasHeigth*.
- Which are the standing (seating, walking, etc.) Virtual Human? In VHO: *VirtualHuman hasPostureType*.

9.2.2 Geometrical Aspects

General descriptions about the geometry are also found in the Common Shape Ontology. The Virtual Human ontology contains descriptions that refer to special features in the geometry, mainly for animation purposes. These features are the skeleton description, landmarks, body segmentation, body part measures, etc.

This ontology can describe any human like skeleton that has a graph-like structure, such as the one defined by the H-Anim standard. This kind of structure is described in the CSO as graph-like structure. If an H-Anim structure is defined, it is possible to infer extra information thanks to the priori knowledge of H-Anim skeleton. It is possible to infer, for example: the hierarchy of the structure, the body parts, the Level of Articulation, classification of complete/incomplete body structure, etc.

Some examples of competency questions related to geometrical and structural aspects are:

- What is the body model geometry? (a mesh, a point set, etc.) In VHO: *VirtualHuman hasGeometry hasShapeOntologyRepresentation GeometrcalRepresentation (CommonShapeOntology).*
- Is the Virtual Human complete (does it have a skeleton, a hierarchy of body parts, a set of landmarks attached to it)? In VHO: *CompleteVirtualHuman, VirtualHumanWithLandmarks, VirtualHuman-WithSkeleton.*
- Which VH have a landmark description? In VHO: *VirtualHuman hasStructure hasNode nodeType Landmark.*
- Which are the available structural descriptors for a particular Virtual Human? In VHO: *Virtual-Human hasStructure hasNode nodeType.*
- Which aspects of the shape are described by the structural descriptor related to a particular VH? In VHO: *VirtualHuman hasStructure StructureDescription hasShapeRepresentation hasDescription.*

9.2.3 Animations

The VHO describes two types of prerecorded animation sequences: Motion Captured and Key Frame Interpolation. To classify them, the Motion Category class defines the following categories: Communication and Gesture, Interaction, Locomotion, and Physical activities and Sports. For Motion Capture animations, it is possible to describe morphological characteristics of the person who performed the Motion Capture. Key Frame animations can also have a morphology description associated to a specific Virtual Human model.

In order that an animation can be performed by a Virtual Human, the Virtual Human's structure should have all nodes of the Animation's structure. Even in this ontology is possible to extract structures from the Virtual Human and the Animation, but it is not possible to make comparisons.

Higher semantic descriptions of animations are included with emotional expression descriptors. Firstly, by segmenting animations into body part movements descriptions that can have emotional meaning; for example facial expressions or hands gestures. Secondly, a more specialized ontological description of emotional expressions by using Facial and Body Animation object of MPEG-4 standard. This classification consists in describing emotions according to animation range values.

Some competency questions that were answered related to animations are:

- What are the joints affected by this animation sequence? In VHO: *AnimationSequence hasStructuralDescriptor hasNode nodeType Joint.*
- Are there any animation sequences lasting more than 1 minute that are suitable for this Virtual Human? In VHO: *AnimationSequence hasDuration > 1min.*

- Are there any running/football playing animation sequences for this kind of Virtual Human? In VHO: *AnimationSequence hasMotionCategory PhysicalActivityAndSport*.

- What are the facial animations for expressing the emotion of sadness? In VHO: *AnimationSequence isComposedOf BodyPartMovement performInBodyPart Face hasEmotionDescription Sad*.

- What kinds of anger expressions do we have? In VHO: *AnimationSequence isComposedOf BodyPartMovement hasEmotionDescription Anger*.

9.2.4 Virtual Human Controllers

Virtual human controllers can be classified in the VHO according to their purpose: Cinematic, Dynamic, Cognitive, Gesture, Path Planning, Skinning, Perception, etc. Controllers are described by their inputs and outputs, and also their special requirements such as other controllers, animations, specific virtual humans etc. Thus, for example, it is possible to describe a behavioral controller that controls a Virtual Human that has an individual description with specific dimensions of personality (e.g. Extraversion/Introversion).

The following list of competency questions represents the related description of Virtual Human Controllers:

- What are the input and output channels of a particular Behavior controller (animation algorithm)? In VHO: *VirtualHumanController hasInput? hasOutput*

- What are the models suitable to be animated with this algorithm? In VHO: *VirtualHumanController controls VirtualHuman*.

- Can this Virtual Human react to sound events in its virtual environment? In VHO: *VirtualHumanController hasControllerPropuse SensorialHearing*

The general information of Virtual Human controllers is found in the CTO. The Controller has the property *hasImplementationTool*, which points to a Software tool, if implemented. Software tools can implement algorithms, this is described with the property *hasAlgorithm*, which points to the *Algorithm* class in the CTO.

9.2.5 Additional Virtual Human Features

Additional to the Virtual Human description, the VHO allows describing accessories, garments and Smart Objects. These elements are geometries of objects with additional features such as skeleton and landmarks; and can be used by Virtual Humans.

The ontology of virtual garments presented in [53], has been included in this ontology. This ontology describes how garment's parts are linked together, their landmarks, and what is their corresponding position in a human body. Finally, the Smart Objects are also described as featured objects that contain information for their manipulation [2] by the Virtual Humans (i.e. grasping, manipulation). The questions answered related to Smart Objects are:

- What capabilities does an object provide? In VHO: *SmartObject hasDescription*.

- What are the actions the human can execute on the object? In VHO: *SmartObject has Attribute ObjectAttribute*.

- What are the characteristics of an object (structure, physical properties, etc.)? In VHO:*SmartObject hasSmartObjetStructuire hasGeometry*.

- How can the object be grasped? In VHO: *SmartObject hasAttribute HandPosture*.

This section presented the main components related to the Virtual Human simulation considered in the ontologies presented in this thesis.

9.3 Limitations and Perspectives

The competency questions are used to validate the ontology. However there are some limitations in the ontologies to answer some other questions. For example, several questions posed at the beginning of this work, need extra processing to be answered because their answer is the result of a comparison of two questions. An example of these questions are:

- Does this Virtual Human match another Virtual Human (or how much do they match)? In particular: are they in the same posture? Do they have the same structure? Do they have similar parts (same arm length, same fatness, similar nose)? Do they have similar anthropomorphic measures (in terms of landmarks)?
- How will this Virtual Human look like after 20 years? With 20 kg more? With another nose?
- Does this model fit this cloth?
- What Virtual Human do I get if I put the head of VH1 on the body of VH2?
- Can the animation sequence X be applied to the Virtual Human Y (in the case of key-frames for skeleton-based animation this would basically depend on the possibility to match the key-frame data to the skeleton of the Virtual Human)?

Other example is the search for specific Animations, Garments or Objects that a given Virtual Human could use. This information was represented with the properties:

- *canWearGarment*: if a Virtual Human has all the landmarks that the garment has, and the corresponding body part size.
- *canUseObject*: if the Virtual Human has the joints and landmarks that the object needs.
- *canPerformAnimation*: if the Virtual Human has all the joints that the animation needs, and the joint's names are the same; if the morphology description for the animation corresponds to the Virtual Human ones.

These relationships cannot be directly inferred by the ontology, one needs to perform some comparisons of different queries to be able to know the answer. This could be solved by including classes with special restrictions to be able to infer the relationships.

A possible solution to these problems is the implementation of additional functions after making a query to compare results. For example, to answer the question: can Virtual Human X wear the Garment Y? It is possible to implement the rule:

$$\forall\, Y: \; (hasVHLandmark(X,\, Y)\; AND\; hasGarmentLandmark(Z,\, Y)$$

$$AND\; isVH(X)\; AND\; isGarment(Z)\; AND\; isLandmark(Y)$$

This rule is indeed useful to infer the property *canWearGarment* described before. The implementation of such rules remain in the future work. However, an alternative solution to these questions is the use of processing tools or services.

In summary, this ontology is useful to provide featured or not Virtual Humans, a variety of well described animation files, raw data of motion captured sequences, animation algorithms, etc. If the population of this ontology continues, it will provide to research communities tools and resources to speed up the research process and provide new resources for testing their applications. Moreover, the ontology can be used as a complete knowledge base to drive the data of real time applications.

However, the usability of the ontology is constrained to its population. The need for automatic or semi-automatic annotation tools remains as important future work for the ontology usage. Additionally, it is important to mention that the developed technologies such as ontology editors and reasoning systems are still not mature. There are not so many technologies that help the exploitation of ontologies, nevertheless with emerging field in the semantic web this will certainly evolve.

This ontology is a step towards to the formalization and documentation of virtual human creation. This evolution is important because with the knowledge of virtual human creation process, it is possible to speed up the population of Virtual Environments for example by cloning real people. It is possible to anticipate in the acquisition and reconstruction process, and the information required in the simulation.

Finally, the description of Virtual Humans can also be complemented with the semantic formalization of virtual environments, Interfaces and so on. This is a very important future direction because virtual environments are becoming very popular, thus to making them scalable, reusable and adaptable is a need for an important benchmark. This is for example the direction of other new NoE such as SALERO [1] and FOCUS K3D [2].

[1]Semantic AudiovisuaL Entertainment Reusable Objects. http://www.salero.eu/
[2]Foster the Comprenhention and Usie of Knowledge intensive 3D media. ICT-2007-214993

Appendix A

Annotation Tools

Annotation of existing and new digital items is a very important task for ontology exploitation. This is usually a very hard task because of the granularity of information to be extracted and the different file format of the resources. Moreover, some semantic descriptions relay on the users' observations and criteria.

This appendix presents some tools developed to support annotation tasks and make easier the population of the Virtual Human Ontology. The first tool supports the manual annotation of Motion Captured sequences using a multimodal approach. The second tools are for automatic extraction of meta-data from geometry and animation files. The information to be extracted has the purpose of populating the ontologies presented in the previous chapter.

A.1 Multi-modal Annotation Tool for Motion Capture

Motion Capture is the extraction of real world motion and transferring it to the digital one. This technique is the most used for character animation, and produces large amounts of data. Commonly, animation databases are used to store and organize this data. However, the reuse of animations is not always possible because it is hard to find exactly what is needed in numerous large sequences if they are not properly classified or described. Therefore, the better description is provided for animations at the moment they are created, as there are more opportunities for reusability.

The Motion Capture process is inherently multimodal because it involves several inputs such as Motion Capture (mocap) sensors and user control in an animation software. This process can be complemented with more elements that facilitate annotation tasks. Multimodal interfaces are implemented in Virtual Environments (VE) because they help to produce the effect of immersion. This immersion is provided through a natural interaction between the user and the environment. One pioneer multimodal application is the "Media Room" by Bolt [22]. This application combined images projected on a screen and user gestures. Positioning one or more users in front of a large rear-projection screen displaying the virtual world is an approach in semi-immersive VE that has given encouraging results. Examples of systems implementing the semi-immersive approaches are based on: large screens, or cave system, handheld devices to navigate and interact in the environment [1][34], and 3D sound rendering [135]. Handheld devices innovate interfaces for controlling higher level features [70].

Immersive environments are more likely to provide semantic information about captured movements. For example, if the goal of the Motion Capture session is to create emotional animations, using an immersive environment the criteria for relating a movement with an emotion is less subjective.

The tool proposed here integrates the required inputs into a multimodal interface composed of a handheld device (a mobile GUI), Motion Capture equipment and a large projection screen; providing an immersive multimodal environment for animating characters. This approach is to animate a character from "inside", having the actor (animator) immersed in a Virtual Environment and looking at the world through the eyes of the character.

The tool proposed here, intends to facilitate both the data acquisition and organization. This tools targets the information acquisition during the creation process of animations.

A.1.1 Extracted Meta-data

During the Motion Capture process, the descriptors that can be extracted involves different levels of knowledge. They can be general (e.g. hardware used), or they can be specific (e.g. the movements recorded during the session). Some of this data can be automatically extracted, and some other manually filled by the user using the handheld device.

Form the ontologies described in the previous chapter, this is the list of the meta-data that can be used to describe the Motion Capture session:

- Motion Capture session is described in the SAP ontology. The meta-data associated to the session is: the motion capture device type, the acquisition conditions (logistics, environment, setting up pictures, any trick used), the system used to process the animations (i.e. Maya Software), etc.
- Each animation is described more in detail in the CSO; where the corresponding meta-data associated to a Motion Capture sequence is: animation format, duration, frame rate, if it has filtering, if it is raw data or it has been processed, the acknowledgement of the creator and owner, etc.
- As the animation belongs to a Human motion, it can be also described in the VHO. The corresponding meta-data for this ontology is: motion category, morphological description of the actor, and individual description of the movement, etc.

This information is generated in a XML format, and is inserted in the ontologies. Next subsection gives more details on the components of the multimodal interface and how they are interconnected.

A.1.2 System Description

This multimodal annotation tool makes use of Motion Capture, a large projection screen and a handheld device. The Motion Capture system, as main input device, allows the animator to acquire high quality animations and give the appropriate intention and expressiveness to each movement. The second input modality is the handheld device that will work as a remote control. In combination with the PDA, a large screen will provide visual feedback to the user. With the PDA device the amount of people required in the production process is reduced, which makes a more interactive interface; the same person using the Motion Capture can drive the annotation process. Moreover, this mobile interface allows for previewing the animation and accessing the database without the need to be in front of a PC.

The interaction of the elements above mentioned with the knowledge-based system is illustrated in figure A.1. These elements are described as follows:

- Main control of this tool resides in the Manager system. This Manager receives commands from the PDA device and executes an action in response; it can also send information to the PDA, such as metadata (attributes) or animation data (the actual animation).
- The Motion Capture system continuously sends information from the sensors in the actor and the Manager system reads them depending of the command in process; it could be reading only the right hand orientation or all the sensors information (recording an animation).

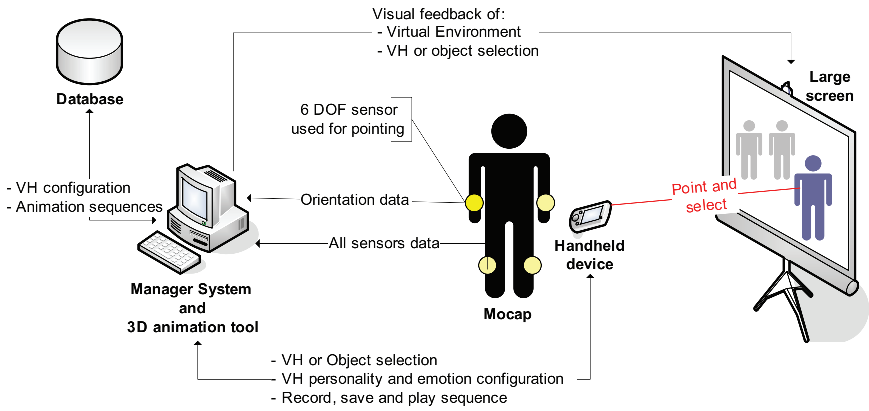

Figure A.1: Multimodal interaction between elements.

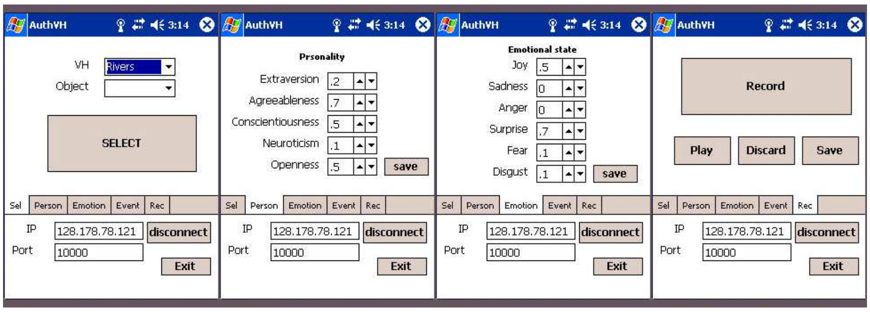

Figure A.2: Screen shots of the GUI displayed on the PDA.

- The Manager makes transactions in the knowledge-base system, can store or retrieve information of the metadata or animation sequences.
- The manager system communicates with the 3D viewer in which the VE is represented (Maya). This VE contains Virtual Humans and objects that can trigger events, the scene is projected on the large screen.

The annotation process is driven through the PDA device. First, the user selects a Virtual Human or object by pointing at the large screen and pressing a button in the PDA. Objects can be associated to an event. Events can be triggered from the PDA. Depending of the event some of its parameters can be modified, for example a ball can be thrown in different directions.

From the PDA, personality traits and emotion parameters of a Virtual Human can be configured for each sequence to be recorded. Many combinations of parameters of emotional states can be recorded for one event. These configurations are saved in the knowledge base system. The GUI in the PDA with this functionality implemented is illustrated in figure A.2.

Animating a Virtual Human can be done in two modalities: watching the Virtual Human mimic the user movements, or viewing the VE through the Virtual Human eyes (see figure A.4). The character moves in real time according to the data acquired by the Motion Capture system.

For recording, there is a mechanism similar to a VCR with a big button to start and stop recording. When start recording button is pressed, two seconds are given to the user to get an initial posture. After those two seconds the sequence starts to be recorded until the user presses stop. When storing the recorded animation, the last two seconds are removed to avoid storing undesired motions due to the movement for pressing the stop button on the PDA. The user can Save or discard a sequence. The anima-

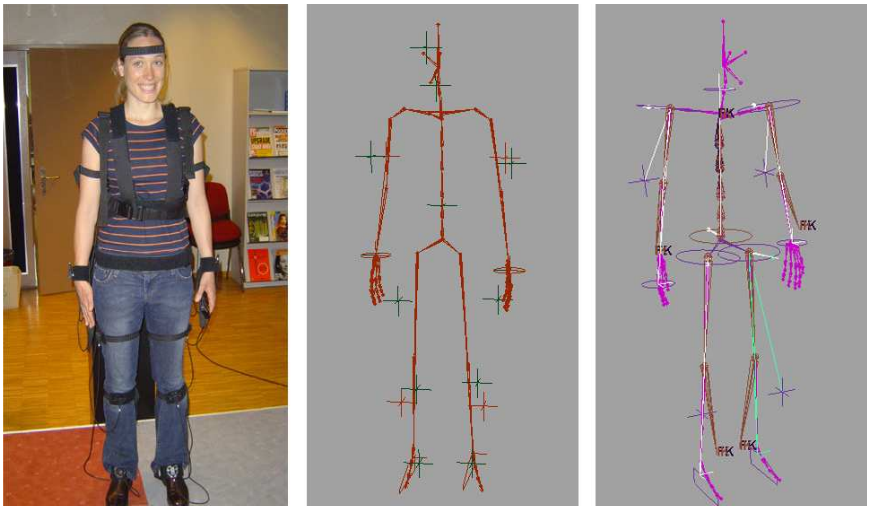

Figure A.3: Association of the mocap's sensors with the skeleton effectors/joints.

tion is stored with the current configuration of emotional states, event, personality and Virtual Human. Finally, "Stop animating" option stops animating from the Motion Capture and "Stop annotating", the Virtual Human is deselected.

To reproduce animations the user can Play a sequence. It consists on reproducing the last sequence recorded in the large screen, or any of the other saved sequences. This last option will only work when the Virtual Human is neither being animated nor in pause. User can also preview recorded sequences in the PDA.

Next section provides technical details concerning the implementation.

A.1.3 Implementation

Motion Capture is performed using the Motion Capture system from Ascension Technology[1], composed of 13 6-DOF magnetic sensors. Data from Mocap sensors are sent to the Motion Capture API of Maya using Maya's 5.0 [12]. Using this API the right transformation to the raw sensor data is applied and calibrated. Calibration was implemented following the method described in [14]; this method computes the correspondence between the initial orientation of each sensor and the default initial orientation of each body part of the character's H-Anim skeleton. To pass the information into the Maya's scene, several MEL (Maya Embedded Language) scripts were implemented. These commands are executed by the Manager system. The Manager system was implemented in C++. The communication with the PDA is done through sockets. Interaction with maya is done through the command port interface.

Virtual Humans modeled in Maya are animated through their H-Anim skeleton. Inverse kinematics was used to compute proper joint rotation values for the Virtual Human limbs, and applied orientation constraints to some joints (root, column and skull) with the suitable weights. The association of the sensors in the actor with the Motion Capture and the locators in Maya and the skeleton are shown in figure A.3.

The handheld device is a PDA iPAQ HP 4700. The GUI for the PDA developed in C# provides the user with the controls to drive the animation process. The user can switch application to visualize the recorded animations on the PDA using the 3D viewer developed in [70].

[1] Ascension Technology Corporation. http://www.ascension-tech.com/

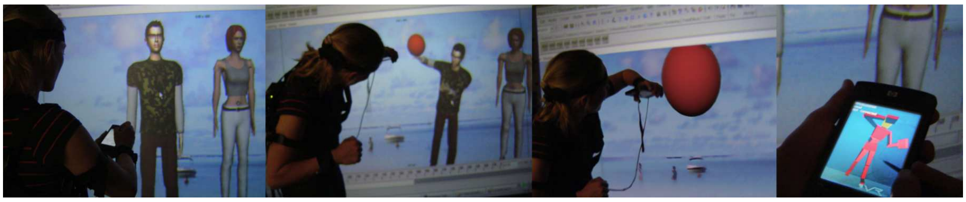

Figure A.4: Using the multimodal annotation tool.

For each animation created during a Motion Capture session, an XML file with the described metadata is created; another XML for the whole session contains the system configuration.

User's pointing gesture is used to select Virtual Humans or objects in the scene. This is by reading the position and orientation of the 6-DOF sensor on the right hand. In this way the user can naturally interact just by pointing at the screen.

A MEL script is used to save animations, and the animation is exported to MPEG-4 BAP format using a plug-in. The BAP file created is placed into a shared directory in the host machine and the path saved in the database. If the user wants to play the animation on the PDA, the Manager sends the path and name of the animation and the Mobile animator is able to preview the animation.

A.1.4 Results

Figure A.4 shows an Motion Capture session using the described multi-modal tool. The user configures a Virtual Human, performs an animation, and previews the saved animation on the PDA.

The combination of Mocap with a large projection screen proved to be an efficient and intuitive way to produce multiple sequences of realistic animation and had good feed-back to the user. The lightweight interface (PDA) was more comfortable than using a PC, but disturbed the user because he had to decide between keeping the PDA in the hand while recording or leaving.

The multimodal interface provided a useful mechanism for capturing relevant information about the Motion Capture session and produced animations. However, the handheld device could be enhanced with speech recognition, in particular for the start/stop recording functionality.

The following code in XML is an example metadata generated in a Motion Captured session which belongs to the Virtual Human Ontology. Tags in the first level are instances for the concepts in the ontology (i.e. *MotionCaptureSequence*) and the following sequence of tags are the properties that belogs to those concepts (i.e. *hasMotionCategory*). The *individualID* attribute corresponds to the URI for each individual in the ontology.

```
<?xml version="1.0" encoding="UTF-8"?>

<VHOIndividual xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="VHOIndividual.xsd">

<BodyPartMovement individualID="vh_m998-alkSlope2">
    <performInBodyPart>UpperBody</performInBodyPart>
    <hasEmotionDescription>Disgust</hasEmotionDescription>
</BodyPartMovement>
<AnimationSegment
    individualID="vh_m999-alkSlope2">
    <startFrame>549</startFrame>
    <hasDuration>10s</hasDuration>
    <hasBodyPartMovement>vh_m998-alkSlope2</hasBodyPartMovement>
</BodyPartMovement>
```

```
<MotionCaptureSequence
    individualID="vh_m1000-walkSlope2">
    <hasShapeRepresentationAndDescription>m1000-walkSlope2</hasShapeRepresentationAndDescription>
    <hasMotionCategory>walk</hasMotionCategory>
    <hasActorMorphology>shubertMorphology</hasActorMorphology>
    <isComposedOf>vh_m999-alkSlope2</isComposedOf>
</MotionCaptureSequence>

<MorphologyDescription individualID="shubertMorphology">
    <hasAge>32</hasAge>
    <hasGenderType>Male</hasGenderType>
    <hasHeigth>heightShubert</hasHeigth>
    <hasWeight>weightShubert</hasWeight>
    <hasBodyPartSize>LeftArmShubert</hasBodyPartSize>
    <hasBodyPartSize>RightArmShubert</hasBodyPartSize>
    <hasBodyPartSize>LeftLegShubert</hasBodyPartSize>
    <hasBodyPartSize>RightLegShubert</hasBodyPartSize>
</MorphologyDescription>

<Measurement IndividualID="heightShubert">
    <hasSystemUnitValue>centimerters</hasSystemUnitValue>
    <hasValue>1.66</hasValue>
</Measurement>

<Measurement IndividualID="ShubertLegMeasure">
    <hasSystemUnitValue>centimerters</hasSystemUnitValue>
    <hasValue>80.6</hasValue>
</Measurement>

<Measurement IndividualID="ShubertArmMeasure">
    <hasSystemUnitValue>centimerters</hasSystemUnitValue>
    <hasValue>70.59</hasValue>
</Measurement>

<BodyPartSize IndividualID="LeftLegShubert">
    <hasSize>ShubertLegMeasure</hasSize>
    <belongsToModelPart>LeftLeg</belongsToModelPart>
</BodyPartSize>
    ...
</VHOIndividual>
```

Describing animation sequences by means of meta-data introduces a semantic layer that promotes the reuse of animations. Next section describes different tools to automatically extract meta-data from existing models and animation files.

A.2 Parser-based Annotation Tools

The tools presented in this section were developed with the objective of populating the ontologies described in this dissertation. However, the task of extracting meta-data is very complex due to the large diversity of files formats. Two types of formats were targeted in this tools, the VRML and the MPEG-4 BAP. These two types were choose because they are standardized, which gives an advantage of having a priori knowledge of the content.

These formats and the meta-data extracted are briefly explained below. These tools are available in

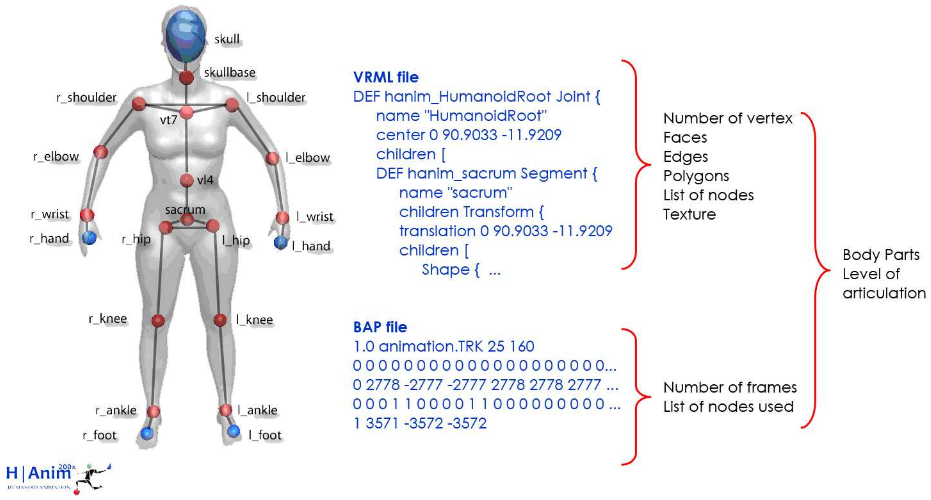

Figure A.5: Files examples and the extracted data using using automatic tools

the tools repository of AIM@SHAPE[2].

A.2.1 MPEG-4 Body Animation Parameters Meta-data Extractor

MPEG-4 is an international standard developed by the Moving Picture Experts Group (MPEG) which defines a set of tools for compressing multimedia at very low bit rates. MPEG-4 Face and Body Animation (FBA) is a part of MPEG-4 standard which defines human like virtual character representation and their animation.

The Body Animation Parameters (BAP) format of MPEG-4 specifies the joint transformations to modify virtual body postures during animation. It contains 296 parameters that describes the topology of the skeleton. A BAP file contains several lines that correspond to the frames. Each frame is composed of two lines: one line with 296 values of corresponding joints (with 1 or 0 to specify if the joint is used or not); and a second line that has another 296 masks indicates actual transformation of the joint.

The parser created to read BAP files extracts the number of frames in the animation, and the list of used joints. Part of the BAP file and meta-data extracted is depicted in figure A.5.

The parser generates two XML files for two of the ontologies presented in this dissertation. Examples of these files generated from the parser are the following:

```
<?xml version="1.0" encoding="UTF-8"?> <SRindividuals
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SR_2.05b.xsd">

<FileInfo individualID="vrlab_boy.bap_fileinfo">
    <hasFileFormat>BAP</hasFileFormat>
    <hasFileName>boy.bap</hasFileName>
    <hasFileSize>97</hasFileSize>
    <hasFileURL>http://shapes.aimatshape.net</hasFileURL>
</FileInfo>

<KeyFrame individualID="cso_vrlab_boy.bap">
    <hasInterpolatorType>none</hasInterpolatorType>
```

[2] http://dsw.aimatshape.net/tools/

```
    <hasNumberOfKeyFrames>160</hasNumberOfKeyFrames>
    <hasAnimationFormat>BAP</hasAnimationFormat>
    <hasDuration>6</hasDuration>
    <hasGeometryFormat>none</hasGeometryFormat>
    <hasName>boy.bap</hasName>
    <hasStatus>1</hasStatus>
    <hasStructureType>h-anim</hasStructureType>
    <hasAnimationMethod>keyFrame</hasAnimationMethod>
    <hasContactPerson>Mireille_Clavien</hasContactPerson>
    <hasCreator>Mireille_Clavien</hasCreator>
    <hasFileInfo>vrlab_boy.bap_fileinfo</hasFileInfo>
    <hasOwner>EPFL</hasOwner>
    <hasUploadDate>Mon Nov 12 15:21:20 CET 2007</hasUploadDate>
    <hasAvailabilityLevel>project-only</hasAvailabilityLevel>
</KeyFrame>
</SRindividuals>
```


This first example presents some instances for the *KeyFrame* class in the Common Shape Ontology. Some of the values extracted from the animation file are: the file format, the number of key frames, the FileInfo; other values are fixed such as the owner, creator, contact person, etc.

More specific metadata extracted from the same animation file belongs to the Virtual Human Ontology. This metadata is presented in the following XML:


```
<?xml version="1.0" encoding="UTF-8"?> <VHOIndividual
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="VHIndiciduals.xsd">

<KeyFrameSequence IndividualID="vh_vrlab_boy.bap">
    <usesJoint>l_sternoclavicular</usesJoint>
    <usesJoint>r_sternoclavicular</usesJoint>
    <usesJoint>l_acromioclavicular</usesJoint>
    <usesJoint>r_acromioclavicular</usesJoint>
    <usesJoint>l_wrist</usesJoint> <usesJoint>r_wrist</usesJoint>
    <usesJoint>l_hip</usesJoint> <usesJoint>r_hip</usesJoint>
    <usesJoint>l_knee</usesJoint> <usesJoint>r_knee</usesJoint>
    <usesJoint>l_shoulder</usesJoint> <usesJoint>r_shoulder</usesJoint>
    <usesJoint>r_elbow</usesJoint> <usesJoint>l_elbow</usesJoint>
    <hasShapeRepresentationAndDescription>cso_vrlab_boy.bap</hasShapeRepresentationAndDescription>
</KeyFrameSequence>
</VHOIndividual>
```


This example presents an instance for the *KeyFrameSequence* class, together with the set of properties to be inserted in this class such as the Joints used in the animation and the corresponding link to the description of the *KeyFrame* in the Common Shape Ontology described above.

A.2.2 VRML Meta-data Extractor

The VRML format follows the H-Anim standard. The analyzed files are Humanoid models. The body of a character is built as a series of nested Joints. Each joint may have a Segment associated with it. The total list of the joints and associated segments are presented in section. Each Segment of the body is defined by a mesh of polygons. Each segment can be mapped to the 2D texture bitmap, which allow extracting associated textures. In VRML there is a node that represents a 3D shape formed by constructing faces (polygons) from listed coordinates. The coordinates are indexed to define the polygonal faces.

Figure A.5 presents an example of a VRML file. The specific metadata extracted from VRML files is stored in two XML files that correspond to the Common Shape Ontology and to the Virtual Human Ontology. An example of these metadata is presented in the following XML:


```
<?xml version="1.0" encoding="UTF-8"?>

<SRindividuals xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SR_2.05b.xsd">

<FileInfo  individualID="vrlab_natasha.wrl_fileinfo">
    <hasFileFormat>VRML</hasFileFormat>
    <hasFileName>natasha.wrl</hasFileName>
    <hasFileSize>198308</hasFileSize>
    <hasFileURL>http://shapes.aimatshape.net</hasFileURL>
</FileInfo>

<CentreLineGraph individualID="m1016-natasha">
    <hasNumberOfArcs>70</hasNumberOfArcs>
    <hasNumberofNodes>141</hasNumberofNodes>
    <isAcyclic>true</isAcyclic>
    <isDirected>false</isDirected>
    <hasAttributeDescription>Position, Orientation, weigth, size</hasAttributeDescription>
    <hasCreationMethod>Designed</hasCreationMethod>
    <isAttributed>true</isAttributed>
</CentreLineGraph>

<Mesh individualID="m1013-natasha">
    <belongsAlsoToClass>NonManifoldMesh</belongsAlsoToClass>
    <hasNumberOfEdges>2822</hasNumberOfEdges>
    <hasNumberOfFaces>1436</hasNumberOfFaces>
    <hasNumberOfVertices>1128</hasNumberOfVertices>
    <hasType>triangular</hasType>
    <hasName>Natasha</hasName>
    <hasFileInfo>vrlab_natasha.wrl_fileinfo</hasFileInfo>
    <hasOwner>EPFL</hasOwner>
    <hasUploadDate>2007-11-28 10:37:15</hasUploadDate>
    <hasStructuralDescriptor>m1016-natasha</hasStructuralDescriptor>
</Mesh>

</SRindividuals>
```


This metadata for the Common Shape Ontology creates instances for a geometrical mesh, a structural descriptor and a file information. Some of the metadata is automatically extracted form the file (i.e. number of vertices, faces, etc) and other is fixed such as some characteristics of the structural descriptor which are fitting of H-Anim. The second XML file is is Virtual Human specific:


```
<?xml version="1.0" encoding="UTF-8"?> <VHOIndividual
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="VHIndiciduals.xsd"> <Measurement
individualID="hanim_l_forefoot_natasha_Size">
   <hasSystemUnitScale>m</hasSystemUnitScale>
   <hasValue>0.060195785</hasValue>
</Measurement> &lt;Segment individualID="hanim_l_forefoot_natasha">
   <hasName>l_forefoot</hasName>
   <nodeType>Segment</nodeType>
   <hasSize>hanim_l_forefoot_natasha_Size</hasSize>
```

```
&lt;/Segment&gt;
    ...
<Measurement individualID="hanim_l5_natasha_Size">
   <hasSystemUnitScale>m</hasSystemUnitScale>
   <hasValue>0.048758827</hasValue>
</Measurement> &lt;Segment individualID="hanim_l5_natasha">
   <hasName>l5</hasName>
   <nodeType>Segment</nodeType>
   <hasSize>hanim_l5_natasha_Size</hasSize>
&lt;/Segment&gt; <Joint individualID="hanim_vl5_natasha">
   <hasName>vl5</hasName>
   <nodeType>Joint</nodeType>
   <beolongsToModelPart>Trunk</beolongsToModelPart>
   <hasChild>hanim_l5_natasha</hasChild>
   <hasChild>hanim_vl3_natasha</hasChild>
</Joint> <Joint individualID="hanim_HumanoidRootnatasha_Root">
   <hasName>HumanoidRoot</hasName>
   <nodeType>Joint</nodeType>
   <hasChild>hanim_sacroiliac_natasha</hasChild>
   <hasChild>hanim_vl5_natasha</hasChild>
</Joint> <VirtualHumanStructure individualID="natasha_Structure">
   <hasShapeRepresentation>m1013-natasha</hasShapeRepresentation>
   <hasRootNode>hanim_HumanoidRootnatasha_Root</hasRootNode>
   <hasLevelOfArticulation>LOA1</hasLevelOfArticulation>
</VirtualHumanStructure> <Geometry individualID="natasha_Geometry">
   <hasShapeRepresentation>m1013-natasha</hasShapeRepresentation>
</Geometry>

<Measurement individualID="natasha_Heigth">
   <hasSystemUnitScale>m</hasSystemUnitScale>
   <hasValue>1.51</hasValue>
</Measurement>

<MorphologyDescription individualID="natasha_Morphology">
   <hasGenderType>Female</hasGenderType>
   <hasHeigth>natasha_Heigth</hasHeigth>
</MorphologyDescription>

<VirtualHuman individualID="natasha_VH">
   <hasGeometry>natasha_Geometry</hasGeometry>
   <hasMorphologyDescription>natasha_Morphology</hasMorphologyDescription>
   <hasPostureType>Stand</hasPostureType>
   <hasVirtualHumanStructure>natasha_Structure</hasVirtualHumanStructure>
   <hasAuthor>EPFL</hasAuthor>
</VirtualHuman>
</VHOIndividual>
```


This example presents the metadata generated for a VRML model which is a *VirtualHuman* instance. The generated descriptors of this Virtual Human are the geometrical information and structural information.

A.3 Conclusions

This chapter presented different tools to annotate animations and models of Virtual Humans. However, it is easy to realize that there is a lot of work to do to annotate digital items. The proposed solutions are not very powerful, but they accomplished their objective to facilitate the population of the ontology.

From a review in the literature, there is indeed ongoing work on animation categorization, and extraction of movement's features. As mentioned in [65], each categorization of animation databases uses their own process for acquiring the respective data. These can be divided into two categories: acquiring the data automatically from the recorded movement of the skeleton, acquiring additional data from separate data sources (i.e. classification by an expert).

An example for the acquisition of data from an expert would be a Laban Movement Analyst, who describes Laban's Shape and Effort of a given motion [171]. But there are also some experiments for retrieving such data automatically. Camurri [26] analyzed the expressivity in dance in Laban's Term of Effort by measuring the changes of the Kinesphere, i. e. the expansion and contraction of the space surrounding the dancer. Some others examples for automatic data acquisition that seems more promising, were presented in section 2.2.3. Those deal with human gesture recognition, template comparison, etc.

Appendix B

Glossary Of Terms

Acquisition Device : A system of sensors connected to a storage device (usually a PC), designed for acquiring data. Related terms: stereo vision.

Acquisition Planning : The design of the strategy for the data acquisition. For instance, it decides object/sensor position(s), number of acquisitions. Related terms: stereo vision.

Animation Controller : Algorithms used to produce animations. The class specifies the inputs required for the algorithm to work and the outputs (usually animation sequences or specific joint values) it is capable to produce. Related terms: Virtual Human, Behavioral Animation, Animation Sequence.

Animation Sequence : Pre-recorded animation sequences (key-frame animation, etc.). In general it contains the joint angle values and/or vertex displacements corresponding to the key animation frames. Different interpolation and codification methods can be used. Such sequences can be applied to one or many VH depending on the codification and technique being used. Related terms: Virtual Human, Motion Capture, Key frame Animation.

Articulated character : 3D animation object made of a geometric skin attached to an articulated skeleton. An articulated character animation is driven by skeleton animation and performed with Skeleton driven deformation. Related terms: Skeleton, Articulation, Joint, Virtual Human.

Articulated skeleton : See Control skeleton. Synonyms: Skeleton. Related terms: Joint, Articulation.

Behavioral Animation : Methods for animating characters by specifying their behavior. For Virtual Humans behavioral animation is expected to exhibit an autonomous believable anthropomorphic behavior, where Virtual Humans perform specific actions under pre-specified conditions. Related terms: Virtual Human, Animation Controller, Animation Sequence, Individual Descriptor.

Blending : Surface connecting smoothly two 3D curves, with prescribed tangent planes along these curves.

Body animation : Methods for animating the body of a Virtual Human. Body animation is usually performed with Skeleton Driven Deformations. Synonyms: Animation Controller. Related terms: Virtual Human, Animation Sequence, Motion Capture, Key frame Animation, Skinning

Body posture : Specification of joint values describing a virtual human body posture. A body posture can be defined using forward/inverse kinematics, forward/inverse dynamics or even motion capture. Related terms: Virtual Human, Animation Controller, Forward Kinematics, Inverse Kinematics, Forward Dynamics, Inverse Dynamics

Calibration : Measuring the parameters appearing in the equations ruling the acquisition process of an acquisition device.

Center-line skeleton : The concept of center-line is strictly related to that of skeleton. Complex objects can be seen as the arrangement of tubular-like components, and abstracted to a collection of center-lines which split and join, following the object topology, and which form, actually, a skeleton. A center-line should satisfy the following requirements: centricity, connectivity and singularity. Related terms: Skeleton.

Control skeleton : A connected set of segments, corresponding to limbs, and joints, corresponding to articulations. Related term: Skeleton, Joint, Articulation, Segment.

Degrees of freedom : (1) The variables one can employ in manipulating geometry. Typically, these are the vertices of polygonal meshes and the control points and knots, in case of B-splines. (2) Possible axis of rotation within a skeleton joint.

Direct kinematics : See Forward kinematics. Synonyms: Forward kinematics

End-effector : The free extremity of an end segment in a control skeleton. Related Terms: Inverse Kinematics, Animation Controller.

Facial animation : Methods for animating the face of a Virtual Human. Facial animation is usually performed with Skin Interpolation. Related terms: Virtual Human, Animation Sequence, Animation Controller, Skinning.

Forward dynamics : In forward dynamics, a skeletal motion is determined from input torques and forces. Related terms: Inverse dynamics, Animation Controller, Body Posture.

Forward kinematics : In forward kinematics, a skeletal posture is determined by assigning input joint angles individually for all the joints of a chain. Synonyms: Direct Kinematics. Related terms: Inverse kinematics, Animation Controller, Body Posture.

Hand Posture : Specification of joint values describing the hand posture required to grab or manipulate a Smart Object. Related terms: Smart Object, Body Posture.

H-Anim Skeleton : The H-Anim skeleton is an animation-oriented structural descriptor. It is an efficient representation of an articulated skeleton for Virtual Humans. It was initially proposed by the Humanoid Animation Working Group (http://www.h-anim.org). Now it has been adopted by the MPEG-4 specification as one of the standard structural descriptors for animatable Humanoid Virtual Characters. Related terms: Virtual Human, Skeleton, Node, Joint, Segment, Site

Hierarchical Model : Organization of data records as a collection of trees, rather than arbitrary graphs. With respect to geometry a rough description of the geometry is located at the highest level, and more and more details are added descending the tree. Related term: Graph.

Human joint : Joint node, basic building block of a Virtual Human skeleton, representing the actual joints on a human being: shoulders, wrists, vertebrae, etc. A joint can have different kinds of children: they can be segments (typically used to store the actual geometry of each limb), or other joints. Under certain cases (not H-Anim 1.0 compliant) a joint could directly store a geometry node. Related terms: Virtual Human, Skeleton, Joint.

Human segment : Node specialisation for H-Anim Segments, typically used as the container of a 3D shape representing a given human body part. Synonyms: Segment. Related terms: Skeleton, Virtual Human.

Individual Descriptor : Human like traits that can be used to describe virtual human characters; these traits can be personality models, emotion, mood, etc. These descriptors are considered to create behavioural animations. Synonyms: Individuality. Related terms: Virtual Human, Behavioral Animation.

Inverse dynamics : It consists in determining the forces and torques required to produce a prescribed motion. Related terms: Forward dynamics, Animation Controller, Body Posture.

Inverse kinematics : In inverse kinematics, a skeletal posture is defined by specifying target location to end-effectors. The joint angles of the control skeleton are defined so that the end-effectors reach their targets as close as possible. Related terms: Forward kinematics, Animation Controller, Body Posture.

Joint : A point of articulation between two or more bones, especially such a connection that allows motion. Synonyms: Articulation Related. terms: Skeleton, H-Anim, Skeleton Joint.

Joint limits : Maximum and minimum values that restrict the angle range of a joint. Related terms: Joint, Skeleton joint angle.

Key-frame Animation : Type of animation that is defined by a set of frames, where each frame contains a set of key frames which indicate the position and orientation of defined objects in the animation. Each key frame includes a key time which orders the set of key frames. Related terms: Animation Sequence, Motion Capture.

Landmark : A place holder to store information associated to a particular location on a 3D geometry. Landmarks can be anatomical structures used as a point of origin to locating other anatomical structures, or points from which measurements can be taken. Synonyms: Site, Feature Point. Related terms: Segment.

Level of Articulation : Term used in the H-Anim standard related to the degrees of freedom of a skeleton chain. LOA is also used relatively to LOD, with respect to multi-resolution for bone-based animation. Synonyms: LOA. Related terms: Level of Detail, LOD.

Level of Detail : It is a compact description of several representations of a shape, from which representations of a shape at different levels of resolution can be obtained. Synonyms: LOD. Related terms: Level od Articulation, LOA.

Marker : See Sensor(1). Synonyms: Sensor. Related terms: Motion Capture.

Mesh : An Euclidean cell complex such that any k-cell of Γ, with $k < d$, bounds at least one d-cell of Γ.

Mesh merging : The combination of meshes resulting from different viewing directions into a single mesh. Related terms: model composition, ICP.

Model Composition : A process by which a new 3D model (e.g. a mesh) is constructed by the seamless composition of two or more existing models. Related terms: Mesh merging, constraints.

Morphological Descriptor : Describes the morphology of a human, such as age, weight, height, gender, etc. Also anatomical and anthropometric infestation related with the human shape. Synonyms: Morphology. Related terms: Virtual Human.

Morphing : The process of making a smooth transition between two shapes.

Motion Capture : Methods for capturing movement data from a live source. The data are filtered and processed in order to replicate the same motion as the one performed by the live source on a control skeleton. Related terms: Animation Sequence, Animation Controller, Key Frame Animation.

Node : An abstract class used to describe a family of ”graph-like” nodes that can constitute diverse structural descriptors such as articulated skeletons. Related terms: Skeleton, H-Anim, Joint, Segment, Site, Landmark.

Object Attribute : Synonyms: Abstract class for Smart Object attributes which can be divided into two main classes: Hand Posture and Location. Related terms: Smart Object, Hand Posture.

Object joint : Analogous to the joint node (Human joint), but adapted to describe smart objects. Related terms: Joint, Skeleton.

Object segment : Node specialization for Smart Objects' structural descriptors. It contains information about each object part, including the geometry and a pointer to additional smart object attributes. Related terms: clustering, aggregate, object segment, segment, segmentation, Skeleton.

Scanning : To record or measure an object by sensing a systematically radiated beam of (laser) light. Sensing is typically done with a device such as a camera, from which color or depth information is acquired.

Segment : (1) Portion of an H-Anim figure (Virtual Human) or Smart Object that represents its geometry and appearance. (2) A collection of pixels that have similar properties (e.g., similar intensity values or texture) which differ from their surrounding pixels. Synonyms: object segment, Human Segment. Related terms: aggregate, segmentation, image segmentation, pixel, image, cluster, Skeleton

Segmentation : To split a data set into smaller subsets based on given classifications or criteria. Frequently used in image processing to analyze pictures and for detection. When reconstructing geometry objects from sampled points, segmentation techniques can be used to detect subsets of points belonging to the same mathematical surface. Related terms: object segment, segment.

Sensor : (1) An electronic device used to measure a physical quantity such as temperature, pressure or loudness and convert it into an electronic signal of some kind (e. g. a voltage). Sensors can be classified in passive (not interacting with the scene) and active (interacting with the scene). Sensors are normally components of some larger electronic system such as a computer control and/or measurement system. (2) Virtual entities that give to Virtual Humans the possibility to acquire information (stimuli) coming from their surrounding virtual environment. Virtual sensors let a VH see, hear and touch its virtual environment and react in consequence. The information acquired is then analyzed with different algorithms and/or Behavior Controllers that produce animation as output (the VH reacts to stimuli). Related terms: Animation Controller, Virtual Human, Behavioural Animation, Motion Capture.

Site : See Landmark. Synonyms: Landmark. Related terms: H-Anim.

Skeleton : (1) In algebraic topology, a p-skeleton is a simplicial sub-complex of a simplicial complex K that is the collection of all simplices of K of dimension at most p. (2) Related with the notion of medial axis, the skeleton of a bounded open subset X is the set of centers of maximal balls, where an open ball B is maximal if every ball that contains B and is contained in X equals B. (3) The skeleton of a shape is the reduced object representation that conforms to human visual perception and preserve the salient shape features. (4) Implicit skeleton: in the field of implicit modeling, it is used for a set of geometric primitives to which a distance can be computed. These primitives are used for generating the field function that defines the surface. Skeleton-based implicit modeling can also be referred to as ”Structural Implicit modeling”, since the skeleton defines an internal structure for the model. Related terms: Virtual Human, Joint, Articulation, Segment.

Skeleton animation : Methods to animate an articulated skeleton by changing the values of the skeleton joints angles over time. The main methods are inverse and forward kinematics and Motion Capture. The result of a skeleton animation is stored as an animation sequence. Related terms: Skeleton joint angle, Animation Sequence, Animation Controller.

Skeleton articulation : An articulation is the intersection of two limbs, which means it is a skeleton point where the limb which is linked to the point may move. Synonyms: Join, Skeleton Joint, Articulation. Related terms: Skeleton.

Skeleton-driven deformation : It consists of deforming the skin to match the current posture of the control skeleton for articulated object animation or deformation. The basic approach consists in assigning a set of joints with weights to each vertex in the skin. The location of a vertex is then calculated by a weighted combination of the transformation of the influencing joints. Various extensions and enhanced complex methods have been proposed. Related terms: Skinning.

Skeletonisation : The operation of extracting a skeleton. Related terms: Skeleton.

Skeleton joint : A joint is the intersection of two segments, which means it is a skeleton point where the limb which is linked to the point may move. Skeleton joints are usually 3D Degrees of Freedom (DoF) rotational joints: flexion, pivot and twist. Synonyms: Joint. Related terms: Skeleton.

Skeleton joint angle : The angle between two segments (limbs) connected by a joint is called the joint angle. Related terms: Joint, Skeletal Animation.

Skeleton skin : A geometric shape that represents the outer shape of an articulated object. This shape is attached to the articulated skeleton and animated with skeleton driven deformations according to the skeleton animation. Related terms: Skinning, Skeletal Animation.

Skeleton skinning : (1) Applying skeleton driven deformation to the skeleton skin of an articulated character in order to adjust it to the current skeleton pose. (2) Attaching geometric primitives or volumes to the joints and limbs of a control skeleton in order to define and control the shape of the skin. This information is further used in animation to control the skin deformation with respect to the control skeleton motion. Skinning Related terms: Skeletal Animation.

Skin Interpolation : It consists in deforming a geometric surface (the skin) according to the animation of an underlying associated animation structure (skeleton, muscles) or to morph targets. Related terms: Skinning, Animation Controller, Morphing.

Skin mapping : It consists of mapping a geometric surface called skin to a control skeleton in order to establish a direct correspondence between the control skeleton and the skin. This information is further used in animation to control the skin deformation with respect to the control skeleton motion. Related terms: Skeletal Animation, Skinning.

Skinning : Refer to the surface representation used to draw a character in skeletal animation. Deformations can be considered as vertex displacements driven by a function of rotation angles belonging to one or more joints. This allows for more aesthetical visual results when animating VH. Skinning algorithms can be used to obtain anatomically correct skin deformation by simulating the interaction between bones, muscles and skin. Synonyms: Skeleton Skinning. Related terms: Skeletal Animation, Skinning Mapping, Skinning Rigging.

Skinning file : It describes a particular type of data used for animating deformable VH. A skinning file contains weights assigned to each vertex of the VH geometry. Such weights indicate the amount of deformation to apply when changing the posture. Related terms: Skinning, Skeletal Animation.

Skin rigging : See Skin mapping. Synonyms: Skin Mapping. Related terms: Skeletal Animation, Skinning.

Smart Object : Virtual object with which virtual humans are capable of manipulations. To implement this, the environment should be extended with some form of knowledge on how interactions between virtual humans and objects are to be carried out. Typical interactions are grasping and manipulation operations. A Smart Object is constituted by a hierarchical collection of nodes. The hierarchical organization specifies the relations between different Geometry and Attribute Sets composing an object. Related terms: Hand Posture, Smart Object Skeleton, Object Segment, Object Joint, Object Attribute.

Smart Object skeleton : Hierarchical structure of object joints and/or object segments used to describe a smart object which usually including mobile parts. Related terms: Skeleton, Smart Object, Object Joint.

Texture : In computer graphics, the digital representation of the characteristic appearance of a surface having a tactile quality. Related terms: image, RGB

Twist : A torsion of the limb which is influenced by the joint. Related terms: Joint, Skeletal animation, Skeletal Joint Angle.

Virtual Human : Specialized instance of an articulated character. The model can be synthesized in a variety of ways and can represent a real or a virtual person. VHs are characterized by a set of general attributes (gender, morphology, ...), and structural descriptors (skeleton, geometry, landmarks, etc..). Synonyms: Virtual Character. Related terms: Individual Descriptor, Morphological Descriptor, Articulated Character, Animation Sequence, Animation Controller, Smart Object.

Appendix C

Ontology Web Language

This appendix presents an overview of the Ontology Web Language (OWL) [35]. This overview aims to help the reader to understand the ontology representation and the notation used in this dissertation.

C.1 Encoding the OWL document

As mentioned in the section 3.2, OWL is created on a layered representation of the Semantic Web. The components on the bottom of the OWL are: URIs and NameSpaces, XML and XMLS Datatypes, RDF and RDF/XML, and RDF Schema and Individuals.

Thus, the header of an OWL document is composed by the needed name spaces and the Ontology element. The header of the Virtual Human Ontology (VHO) has the following elements:

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:cso="http://www.aimatshape.net/ontologies/shapeCommonOntology.owl#"
    xmlns:cto="http://www.aimatshape.net/ontologies/toolCommonOntology.owl#"
    xmlns:ico="http://www.aimatshape.net/ontologies/infoCommonOntology.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   ...
    xml:base="http://www.aimatshape.net/ontologies/virtualHumansOntology.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="file:/C:/infoOntology_v1.0.owl"/>
    <owl:imports rdf:resource="file:/C:/toolOntology_v1.12.1.owl"/>
    <owl:imports rdf:resource="file:/C:/shapeOntology_v2.3.1.owl"/>
  </owl:Ontology>
  ...
</rdf:RDF>
```

Name spaces declaration may also include a reference to upper ontologies that need to be imported, where each name space has a prefix.

Along with name spaces declaration, there is the definition of *xml:base* which contains the documents' URI. This definition serves to have the same URI in different documents in different locations; which allows also to share the same concepts but to have different instances. Also if the name of the file changes the URI is not affected.

After name spaces the *owl:Ontology* element is defined. This element contains the information of the versions, such as the prior version, compatibility and incompatibility with other versions, and deprecated classes and properties. It may also include the location of the imported ontologies through the property *owl:imports*. This specify the URIs of external ontologies referenced in the current ontology.

After the header, the body of the document is defined. This body contains the definition of the OWL components: Classes, Properties and Individual Statements.

C.2 OWL Components

OWL is a vocabulary extension of RDF, which is based on a graph representation. A RDF graph, or simply a graph, is composed of a set of RDF triples. Triples expresses simple assertions or statements, and consist of a subject, a predicate, and an object. The subject and the object represent individuals, and they are related each other through the predicate.

To be able to represent these assertions in OWL, there are three main components defined:

Classes are sets of individuals that share the same description.

Properties represents the predicate

Individuals are subjects and objects

Figure C.1 presents a visual representation of theses components. In figure C.1-(a) classes are in a rounded box and an arrow with the legend *IsA* means the hierarchy among classes. Figure C.1-(b) presents the properties of a class, there are two kinds of properties: data type properties and object properties. Class properties are a relationship between two classes. And, figure C.1-(c) presents the individuals that are instances of a class.

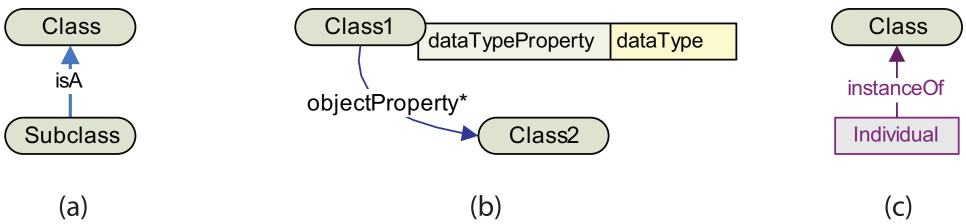

Figure C.1: Nomenclature of visual aid for describing the ontologies

This is the basic notation used to represent the domain, but more specific notation is presented in the rest of the appendix.

C.2.1 Classes

The Class in OWL can be seen as a Class in object oriented paradigm, but in OWL a Class is sometimes treated as synonym of concept. The classes can be referenced internally using *#classname*, and externally using the complete *URIrefs*. Depending on the domain some individuals can be present in more than one class (they can be two things). To express the opposite, classes have to define their *Disjoint* classes.

The following example presents the *VirtualHuman* Class:

```
<owl:Class rdf:about="#VirtualHuman">
```

```
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Resource"/>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
     The main class of this ontology. A Virtual Human is a full-body or
     partial representation of a Human being. The model can be synthesized
     in a variety of ways and can represent a real or a virtual person. VHs
     are characterized by a set of general attributes (sex, nationality, race),
     and structural descriptors. A VH can have animation sequences, behavior
     controllers or smart objects associated to it.
    </rdfs:comment>
    <owl:disjointWith>
      <owl:Class rdf:about="#VirtualHumanController"/>
    </owl:disjointWith>
    <owl:disjointWith rdf:resource="#VirtualHumanAnimationSequence"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#VirtualObject"/>
    </owl:disjointWith>
</owl:Class>
```

A class can be inherit from another class. Thus, the *VirtualHumans* class inherits the properties from *Resource* class, and it is a disjoint with *VirtualHumanController, VirtualHumanAnimationSequence, and VirtualObject* classes.

Therefore, the ontology is the description of concepts of a domain of knowledge, and concepts are represented by classes an described through properties.

C.2.2 Properties

Classes are described using properties, which can be one of the following types:

owl:DatatypeProperty relates properties to data values.

owl:ObjectProperty relates properties to other concepts.

owl:AnnotationProperty describes additional information about concepts.

owl:OntologyProperty relates ontologies with other ontologies.

The most used ones are the *DatatypeProperty* and *ObjectProperty*. For each property there should be a *rdfs:domain* to define the class that the property applies to; and there should be a *rdfs:range*, which is the range of values or objects that the property can have.

Datatype property is associated to a data value which is can be strings or simple XMLS. Datatypes are referenced using the corresponding URI to the XMLS: *http://www.w3.org/2001/SMLSchema*. The following example presents the property *hasInput* as a *DataTypeProperty* that have as domain the *VirtualHumanController*, and as range a *XMLSchema#string*:

```
<owl:DatatypeProperty rdf:ID="hasInput">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#VirtualHumanController"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        List of requirements considered as Input of a controller.
    </rdfs:comment>
</owl:DatatypeProperty>
```

Opposed to Datatype property, Object property is to reference the property to an individual. Which means that the value is a resource rather than a string literal. The following example presents the Object property *hasGeometry* that relates the *Geometry* with the union of concepts:

```
<owl:ObjectProperty rdf:ID="hasGeometry">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#VirtualHuman"/>
          <owl:Class rdf:about="#SmartObject"/>
          <owl:Class rdf:about="#ClothPattern"/>
          <owl:Class rdf:about="#Segment"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="#Geometry"/>
  </owl:ObjectProperty>
```

This example shows that any of the *VirtualHuman*, *SmartObject*, *ClothPattern* and *Segment* concepts may have at least one geometry associated.

These examples are graphically described in figure C.2, (a) depicts the Datatype and (b) the Object properties.

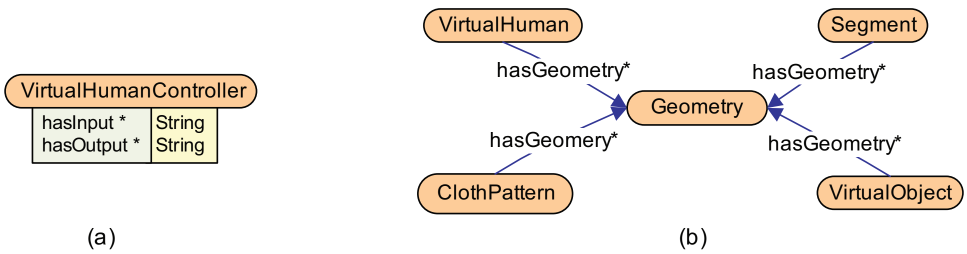

Figure C.2: (a) Data properties of the *VirtualHumanController* class. (b) Object properties of several classes to Geometry class.

The following sections describes different characteristics that properties can have to describe the type of relation or restriction when relating concepts.

C.2.2.1 Functional and Inverse Functional Properties

By default, properties accept the association of multiple resources, represented with an asterisk (*) in figure C.2. The Functional property associates a class with a single value only. This can be applied to both Data and Object properties. An example of this type of properties in the VHO is the *hasStructure* property, where Geometry can have at most one *StructuralDescriptor*:

```
<owl:FunctionalProperty rdf:ID="hasStructure">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#Geometry"/>
    <rdfs:range rdf:resource="#StructureDescription"/>
</owl:FunctionalProperty>
```

Inverse Functional properties define the opposite relation between two concepts that have already a Functional property. This means that if a value of an inverse property is know, it is possible to know to which subject it belongs to. This is exactly what is presented in the following example:

```
<owl:InverseFunctionalProperty rdf:about="#isComposedOf">
    <rdfs:domain rdf:resource="#VirtualHumanAnimationSequence"/>
    <rdfs:range rdf:resource="#AnimationSegment"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <owl:inverseOf rdf:resource="#belongsToAnimation"/>
  </owl:InverseFunctionalProperty>

<owl:FunctionalProperty rdf:ID="belongsToAnimation">
    <owl:inverseOf>
        <owl:InverseFunctionalProperty rdf:ID="isComposedOf"/>
    </owl:inverseOf>
    <rdfs:range rdf:resource="#VirtualHumanAnimationSequence"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#AnimationSegment"/>
 </owl:FunctionalProperty>
```

The *VirtualHumanAnimationSequence* can be composed of at least one *AnimationSegment*, and it is also possible to know to which *VirtualHumanAnimationSequence* a given *AnimationSegment* belongs to. The examples described are illustrated in figure C.3.

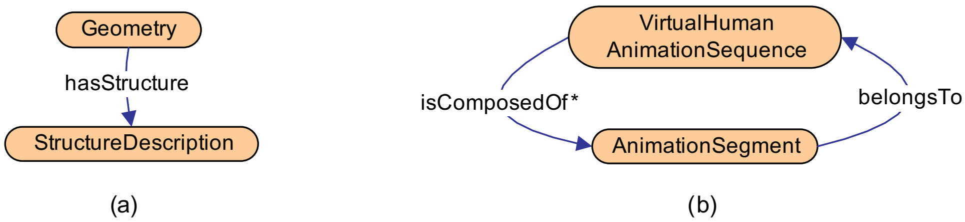

Figure C.3: (a) Functional property *hasStructure*. (b) Invese Functional property *isComposedOf*.

C.2.2.2 Transitive Property

Transitive properties are Object properties whose subjects and values can be chained together. This means if a transitive property relates **a** to **b** and **b** to **c**, then **a** is related to **c** via the same property.

An example of this property is the description of the *StructuralDescriptior* concept. This structure constitutes a graph where nodes are connected with the *hasChild* property. This property is transitive so it is possible to access to all children nodes of a given node in the structure:

```
<owl:TransitiveProperty rdf:about="#hasChild">
    <rdfs:domain rdf:resource="#Node"/>
    <rdfs:range rdf:resource="#Node"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:TransitiveProperty>

<owl:Class rdf:ID="Joint">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Node"/>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:ID="Segment"/>
    </owl:disjointWith>
    <owl:disjointWith>
```

```
      <owl:Class rdf:ID="Landmark"/>
    </owl:disjointWith>
  </owl:Class>
```

This code is represented in the diagram in the figure C.4. For instance the class Joint is subclass of Node and in has the instances: *r_shoulder*, *r_elbow* and *r_wirst*. Thus, it is possible to infer all children of *r_shoulder* which will bring *r_elbow* and *r_wirst* because of the transitivity of the *hasChild* property.

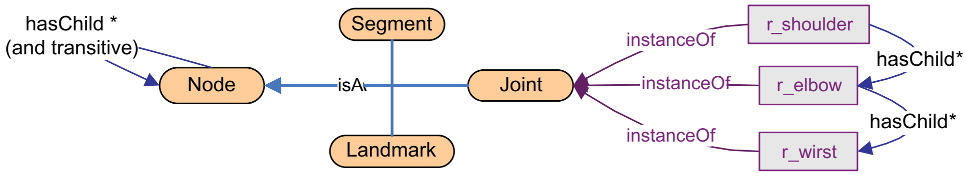

Figure C.4: The transitive property *hasChild* is used to access all children nodes in the structure.

C.2.2.3 Symmetric Property

The Symmetric property functions as as both directions property, but only one is required. This means that the domain and the range of the property must be the same. There is no symmetric property defined in the ontologies represented in this dissertation, but an example would be to define two nodes that may be siblings:

```
  <owl:SymmetricProperty rdf:ID="isSibilingOf">
    <rdfs:range rdf:resource="#Node"/>
    <rdfs:domain rdf:resource="#Node"/>
    <owl:inverseOf rdf:resource="#isSibilingOf"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </owl:SymmetricProperty>
```

The code above presents the property *isSiblingsOf* which have the same concept *Node* as range and domain. This means that a Node that is sibling of other Node, the property applies for the other node as well.

C.2.3 Individuals

The third and last component of the ontologies are the Individuals. Individuals are instances of a given class. Individuals are named using URIs. The individuals that are a particular member of a class are declared using as a tag the name of the class they belongs to. This explicitly specifies the type which the individual belongs to. An example of Virtual Humans individual is the following:

```
<VirtualHuman rdf:ID="vh_m1007-brittany">
    <hasDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
       Female Character</hasDescription>
    <hasGeometry rdf:resource="#vh_m1007-brittany_geometry"/>
    <hasMorphologyDescription>
      <MorphologyDescription rdf:ID="MorphologyDescription_vh_m1007-brittany">
        <hasGenderType rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
           female
        </hasGenderType>
      </MorphologyDescription>
    </hasMorphologyDescription>
```

```
    <hasPostureType rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Neutral
    </hasPostureType>
</VirtualHuman>
```

This example, illustrated in the figure C.5, presents the definition and properties description of the Virtual Human individual with the URI: *vh_m1007-brittany*. This description is very similar to simple XML document, and it has its semantic interpretation in the Virtual Human Ontology.

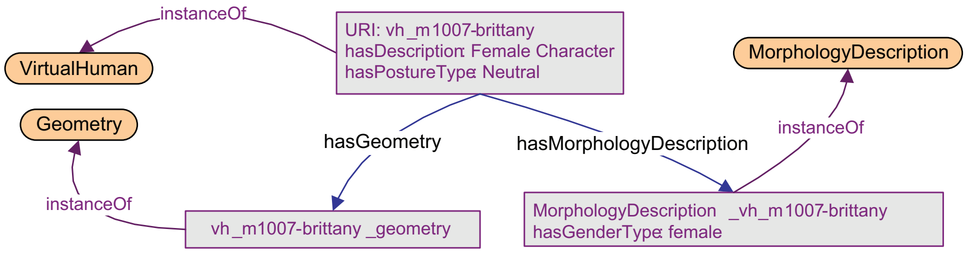

Figure C.5: Graphical representation of instances of a Virtual Human.

C.3 OWL Dialects Comparison

OWL can be supported by three different dialects described in the section 3.2: OWL Lite, OWL DL and OWL Full. These dialects offer different capabilities and apply different restrictions to some of the constructs described in this chapter. Some of these differences are described in table C.1.

Depending on the requirements of the knowledge to model one of these dialects may be chosen. Table C.2 present the advantages and disadvantages of each dialect. For a complete reference of OWL dialects see [93].

This appendix presented a small introduction to OWL with the objective to serve as a reference to the reader of this dissertation.

The complete definition of the OWL standard can be found in the W3C web site [1].

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

Construct Group	**Language constructs**	**Differences**
Class Expressions	rdfs:Class and owl:Class	**Lite and DL:** OWL:class is subclass of RDFS:class. **Full:** RDFS:class and OWL:class are equivalent. Classes can be instances or properties at the same time.
	rdfs:subClasssOf	**Lite**: Subject and Object of statements must be named classes
	owl:equivalentClass	**Lite**: **DL:** Can be applied to class expressions
	owl:disjointWith	**Lite**: N/A **DL:** Relating classes
Derived Classes	owl:intersectionOf	**Lite:** The only available class description only for named classes **DL**: Boolean combination
	owl:unionOf	**Lite**: N/A **DL**: Boolean combination
	owl:complementOf	
	owl:oneOf	
Property characteristics	owl:inverseOf	**DL**: cannot have owl:cartinality nor be functional.
	owl:TransitiveProperty	
	owl:SymmetricProperty	
	owl:FunctionalProperty	
	owl:InverseFunctional Property	**Lite**: Canot be defined for datatype properties **DL**: cannot have owl:cartinality nor be functional.
Property Expressions	rdfs:Property	**DL**: Must be either: ObjectProperty, DatatypeProperty, AnnotationProperty or Ontology Property
	owl:ObjectProperty	**Full**: owl:ObjectTypeProperty is considered equivalent to rdf:Property
	owl:DatatypeProperty	**DL**: Must be disjoint from object properties
	rdfs:doamin	**Lite**: Object of statements must be named class or datatype
	rdfs:range	
Value Constraints	owl:allValuesFrom	**Lite**: Object of statements must be named class or datatype
	owl:someValuesFrom	
	owl:hasValue	**Lite**: NA **DL**: Filler information
	owl:dataRange	**Lite**: NA **DL**: Enumerated data values
Cardinality	owl:minCardinality	**Lite**: (only 0 or 1) **DL**: any integer = 0 and cannot be used on a TransitiveProperty
	owl:maxCardinality	
	owl:cardinality	
Individual Equivalency	rdfs:Individual	**DL**: Must belong to a class
	owl:sameAs	**DL**: Must reference a named individual
	owl:differentFrom	

Table C.1: OWL constructors differences

Dialect	**Advantages**	**Disadvantages**
OWL Lite	Easy to support with software	Limited expressiveness
OWL DL	Decidable	Restricted
OWL Full	Very expressive	Not decidable

Table C.2: OWL dialect comparison

Bibliography

[1] T. Abaci, R. de Bondeli, J. Ciger, M. Clavien, F. Erol, M. Gutierrez, S. Noverraz, O. Renault, F. Vexo, and D. Thalmann. Magic wand and enigma of the sphinx. *Computers & Graphics*, 28(4):477–484, August 2004.

[2] T. Abaci, M. Mortara, G. Patane, M. Spagnulo, F. Vexo, and D. Thalmann. Bridging Geometry and Semantics for Object Manipulation and Grasping. In *Proceedings of Workshop towards Semantic Virtual Environments' (SVE 2005) workshop*, 2005.

[3] A. Aguiló, P. Martinez, J. Buades, F. Perales, and M. Gonzlez. Human motion analysis and synthesis using graphical biomechanics models applied to disable swimming people. In *in the 3rd International Workshop on Virtual Rehabilitation*, pages 16–17, Lausanne, Switzerland, 2004.

[4] A. Albertoni, L. Papaleo, R. Robbiano, and M. Spagnuolo. Towards a conceptualization for shape acquisition and processing. In *Proceedings of the 1st International Workshop on Shapes and Semantics*, 2006.

[5] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 587–594, New York, NY, USA, 2003. ACM.

[6] B. Allen, B. Curless, and Z. Popović. Exploring the space of human body shapes: Data-driven synthesis under anthropometric control. In *Proceedings of Conference on Digital Human Modeling for Design and Engineering. SEA International.*, 2004.

[7] E. Andre, M. Klesen, P. Gebhard, S. Allen, and T. Rist. Integrating models of personality and emotions into lifelike characters. *Proceedings International Workshop on Affect in Interactions. Towards a New Generation of Interfaces*, 453, 1999.

[8] Y. Arafa and A. Mamdani. Scripting embodied agents behaviour with cml: character markup language. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 313–316, New York, NY, USA, 2003. ACM.

[9] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, chapter COMM: Designing a Well-Founded Multimedia Ontology for the Web, pages 30–43. SpringerLink, 2007.

[10] G. Ashraf and K. C. Wong. Semantic representation and correspondence for state-based motion transition. *Visualization and Computer Graphics, IEEE Transactions on*, 9(4):481–499, Oct.-Dec. 2003.

[11] J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: pose selection and illustration. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 667–676, New York, NY, USA, 2005. ACM.

[12] Autodesk. Maya. http://www.autodesk.com/maya, 2006.

[13] Z. B. Azouz, M. Rioux, C. Shu, and R. Lepage. Analysis of human shape variation using volumetric techniques. In *Proc. International Conference on Computer Animation and Social Agents*, pages 197 – 206. IEEE Press, July 2004.

[14] C. Babski and D. Thalmann. Real-time animation and motion capture in web human director (whd). In *VRML '00: Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, pages 139–145, 2000.

[15] N. Badler, J. Allbeck, L. Zhao, and M. Byun. Representing and parameterizing agent behaviors. *Computer Animation*, pages 133–143, 2002.

[16] P. Baerlocher. *Inverse kinematics techniques of the interactive posture control of articulated figures*. PhD thesis, EPFL, Lausanne, 2001.

[17] P. Baerlocher and R. Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *Vis. Comput.*, 20(6):402–417, 2004.

[18] T. Berners-Lee. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor.* Harper San Francisco, 1999.

[19] W. Bille. *Conceptual Modeling of Complex Objects for Virtual Environments: A Formal Approach.* PhD thesis, Vrije University of Brussel, Lausanne, 2007.

[20] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[21] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. P. Johnson, and B. Tomlinson. Integrated learning for interactive synthetic characters. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 417–426, New York, NY, USA, 2002. ACM.

[22] R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980. ACM Press.

[23] R. Boulic, N. Magnenat-Thalmann, and D. Thalmann. A global human walking model with real-time kinematic personification. *Vis. Comput.*, 6(6):344–358, 1990.

[24] R. Burke, D. Isla, M. Downie, Y. Ivanov, and B. Blumberg. Creaturesmarts: The art and architecture of a virtual brain. In *The Game Developers Conference*, 2001.

[25] B. L. Callennec and R. Boulic. Interactive motion deformation with prioritized constraints. *Graph. Models*, 68(2):175–193, 2006.

[26] A. Camurri and R. Trocca. Analysis of expressivity in movement and dance. In *XIII Colloquium on Musical Informatics*, L'Aquila, Italy, September 2000.

[27] G. Caridakis, A. Raouzaiou, K. Karpouzis, and S. Kollias. Synthesizing gesture expressivity based on real sequences. In *Workshop on multimodal corpora: from multimodal behaviour theories to usable models, LREC 2006 Conference*, Genoa, Italy, May 2006.

[28] B. D. Carolis, V. Carofiglio, M. Bilvi, and C. Pelachaud. APML, a Markup Language for Believable Behavior Generation. *Life-Like Characters: Tools, Affective Functions, and Applications*, 2004.

[29] 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*, 2008.

[30] C. Catton, S. Sparks, and D. Shotton. The imagestore ontology and the bioimage database: semantic web tools for biological research images. *Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the (Ref. No. 2005/11099)*, pages 257–264, 30 Nov. - 1 Dec. 2005.

[31] A. Celentano and F. Pittarello. A content-centered methodology for authoring 3d interactive worlds for cultural heritage. In *ICHIM (2)*, pages 315–324, 2001.

[32] C. Cera, W. Regli, I. Braude, Y. Shapirstein, and C. Foster. A collaborative 3d environment for authoring design semantics. *Computer Graphics and Applications, IEEE*, 22(3):43–55, May/Jun 2002.

[33] C. Charbonnier, B. Gilles, and N. Magnenat-Thalmann. A semantic-driven clinical examination platform. In *In Surgetica'2007, Computer-Aided Medical Interventions: Tools and Applications*, pages 183–189, September 2007.

[34] J. Ciger, M. Gutierrez, F. Vexo, and D. Thalmann. The magic wand. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 119–124, New York, NY, USA, 2003. ACM Press.

[35] R. Colomb. Ontology and the semantic web. 2007.

[36] T. Conde and D. Thalmann. Autonomous virtual agents learning a cognitive model and evolving. pages 88–98, 2005.

[37] T. Conde and D. Thalmann. Learnable behavioural model for autonomous virtual agents: low-level learning. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 89–96, New York, NY, USA, 2006. ACM.

[38] G. Costagliola, S. D. Martino, F. Ferrucci, and F. Pittarello. An approach for authoring 3d cultural heritage exhibitions on the web. In *SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 601–608, New York, NY, USA, 2002. ACM Press.

[39] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. Taylor. Emotion recognition in human-computer interaction. In *IEEE Signal Processing Magazine, no. 1, January 2001, pp.32-80*, 2001.

[40] C. Darwin. *The Expression of the Emotions in Man And Animals.* Digireads. Com, 2005.

[41] E. de Aguiar, V. Theobalt, M. Magnor, and H.-P. Seidel. Reconstructing human shape and motion from multi-view video. In *The 2nd IEE European Conference on Visual Media Production*, pages 44–51, 2005.

[42] F. de Rosis, C. Pelachaud, I. Poggi, V. Carofiglio, and B. D. Carolis. From greta's mind to her face: modelling the dynamics of affective states in a conversational embodied agent. *International Journal Human-Compuer Studies*, 59(1-2):81–118, 2003.

[43] E. de Sevin and D. Thalmann. A motivational model of action selection for virtual humans. In I. C. SocietyPress, editor, *Computer Graphics International*, New York, 2005.

[44] S. Decker, S. Melnik, F. V. Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The semantic web: The roles of xml and rdf. *IEEE Internet Computing*, 04(5):63–74, 2000.

[45] F. Dellas, L. Moccozet, N. Magnenat-Thalmann, M. Mortara, G. Patan"e, M. Spagnuolo, and B. Falcidieno. Knowledge-based extraction of control skeletons for animation. In *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, pages 51–60, Washington, DC, USA, 2007. IEEE Computer Society.

[46] P. Eckman. Facial expression and emotion. *American Psychologist*, 48:384–392, 1993.

[47] A. Egges, S. Kshirsagar, and N. Magnenat-Thalmann. Generic personality and emotion simulation for conversational agents. In *Journal of Computer Animation and Virtual Worlds*. John Wiley and Sons, Ltd., 2004.

[48] P. Ekman and W. Friesen. *The Facial Action Coding System: A Technique for The Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto,. CA, 1978.

[49] N. Ersotelos and F. Dong. Building highly realistic facial modeling and animation: a survey. *Vis. Comput.*, 24(1):13–30, 2007.

[50] H. Eysenck. *Handbook of personality: Theory and research*, chapter Biological dimensions of personality, pages 244–276. New York, Guilford., 1990.

[51] B. Falcidieno, M. Spagnuolo, P. Alliez, E. Quak, E. Vavalis, and C. Houstis. Towards the semantics of digital shapes: The aim@shape approach. In *EWIMT*, 2004.

[52] P. Faloutsos, M. van de Panne, and D. Terzopoulos. The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers and Graphics*, 25(6):933–953, 2001.

[53] A. Fuhrmann, C. Groß, and A. Weber. Ontologies for virtual garments. In *Proceedings of Workshop towards Semantic Virtual Environments' (SVE 2005) workshop*, Villars Switzerland., 2005.

[54] P. Gallaher. Individual differences in nonverbal behavior: Dimensions of style. *Journal of Personality and Social Psychology*, 63, 1992.

[55] A. Garcia-Rojas, M. Gutierrez, and D. Thalmann. Visual creation of inhabited 3D environments: An ontology-based approach. *Visual Computer*, 2008.

[56] A. Garcia-Rojas, M. Gutierrez, D. Thalmann, and F. Vexo. Multimodal authoring tool for populating a database of emotional reactive animations. In *Second International Workshop Machine Learning for Multimodal Interaction.*, volume 3869, pages 206–217, 2006. Virtual Reality Lab., Ecole Polytech. Fed. de Lausanne, Switzerland.

[57] A. Garcia-Rojas, F. Vexo, and D. Thalmann. Semantic Representation of Individualized Reaction Movements for Virtual Human. *International Journal of Virtual Reality*, 6(1):25–32, 2007.

[58] A. Garcia-Rojas, F. Vexo, D. Thalmann, A. Raouzaiou, K. Karpouzis, and S. Kollias. Emotional Body Expression Parameters In Virtual Human Ontology. In *Proceedings of 1st Int. Workshop on Shapes and Semantics*, pages 63–70, 2006.

[59] A. Garcia-Rojas, F. Vexo, D. Thalmann, A. Raouzaiou, K. Karpouzis, S. Kollias, L. Moccozet, and N. Magnenat-Thalmann. Emotional face expression profiles supported by virtual human ontology. *Computer Animation and Virtual Worlds*, 17(3-4):259–269, 2006. VRLab. - EPFL, CH-1015 Lausanne, Switzerland.

[60] P. Glardon, R. Boulic, and D. Thalmann. Robust on-line adaptive footplant detection and enforcement for locomotion. *Visual Computer*, 22(3):194–209, 2006. Virtual Reality Lab., Ecole Polytechnique Federate de Lausanne, Switzerland.

[61] M. Gleicher. Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, New York, NY, USA, 1998. ACM.

[62] A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[63] M. Green. Towards virtual environment authoring tools for content developers. In *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 117–123, New York, NY, USA, 2003. ACM.

[64] T. R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.

[65] S. Grunvogel, J. Piesk, S. Schwichtenberg, and G. Buchel. Amoba: A database system for annotating captured human movements. In *CA '02: Proceedings of the Computer Animation*, pages 98–102. IEEE Computer Society, 2002.

[66] N. Guarino. Formal ontology in information systems. In I. P. Amsterdam, editor, *Proceedings of FOIS'98*, pages 3–15, Trento, Italy, 6-8 June 1998.

[67] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 55–66, New York, NY, USA, 1998. ACM.

[68] M. Gutiérrez. *Semantic virtual environments*. PhD thesis, Lausanne, 2005.

[69] M. Gutiérrez, A. García-Rojas, D. Thalmann, F. Vexo, L. Moccozet, N. Magnenat-Thalmann, M. Mortara, and M. Spagnuolo. An ontology of virtual humans: Incorporating semantics into human shapes. *The Visual Computer*, 23(3):207–218, 2007.

[70] M. Gutierrez, F. Vexo, and D. Thalmann. The mobile animator: Interactive character animation in collaborative virtual environments. In *VR '04: Proceedings of the IEEE Virtual Reality 2004 (VR'04)*, page 125, Washington, DC, USA, 2004. IEEE Computer Society.

[71] J. A. Harrigan. Listener's body movements and speaking turns. In *Communication Research*, volume 12, pages 233–250, April 1985.

[72] B. Hartmann, M. Mancini, and C. Pelachaud. Formational parameters and adaptive prototype instantiation for mpeg-4 compliant gesture synthesis. In *CA '02: Proceedings of the Computer Animation*, page 111, Washington, DC, USA, 2002. IEEE Computer Society.

[73] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78, New York, NY, USA, 1995. ACM.

[74] T. humanoid animation working group. H-anim. http://www.h-anim.org.

[75] J. Hunter. Adding Multimedia to the Semantic Web-Building an MPEG-7 Ontology. *International Semantic Web Working Symposium (SWWS)*, pages 261–281, 2001.

[76] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: generating animated head models with anatomical structure. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 55–63, New York, NY, USA, 2002. ACM.

[77] M. Kallmann. Analytical inverse kinematics with body posture control. *Computer Animation and Virtual Worlds*, 2007.

[78] M. Kallmann and D. Thalmann. Direct 3d interaction with smart objects. In *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 124–130, New York, NY, USA, 1999. ACM.

[79] E. Kalogerakis, S. Christodoulakis, and N. Moumoutzis. Coupling ontologies with graphics content for knowledge driven visualization. In *VR '06: Proceedings of the IEEE Virtual Reality Conference (VR 2006)*, page 6, Washington, DC, USA, 2006. IEEE Computer Society.

[80] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. volume 11, pages 59–69, 1992.

[81] M. Klesen, M. Kipp, P. Gebhard, and T. Rist. Staging exhibitions: methods and tools for modelling narrative structure to produce interactive performances with virtual actors. *Virtual Reality*, 7(1):17–29, 2003.

[82] M. Koivunen and E. Miller. W3C Semantic Web Activity. *Semantic Web Kick-Off in Finland*, pages 27–44, 2001.

[83] T. Komura, E. S. L. Ho, and R. W. H. Lau. Animating reactive motion using momentum-based inverse kinematics: Motion capture and retrieval. *Comput. Animat. Virtual Worlds*, 16(3-4):213–223, 2005.

[84] S. Kopp, B. Krenn, S. Marsella, A. Marshall, C. Pelachaud, H. Pirker, K. Thórisson, and H. Vilhjálmsson. Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. *IVA*, 2006.

[85] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.*, 23(3):559–568, 2004.

[86] L. Kovar, J. Schreiner, and M. Gleicher. Footskate cleanup for motion capture editing. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 97–104, New York, NY, USA, 2002. ACM.

[87] S. Kshirsagar. A multilayer personality model. In *SMARTGRAPH '02: Proceedings of the 2nd international symposium on Smart graphics*, pages 107–115, New York, NY, USA, 2002. ACM Press.

[88] S. Kshirsagar and N. Magnenat-Thalmann. Virtual humans personified. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 356–357, New York, NY, USA, 2002. ACM Press.

[89] S. Kshirsagar, N. Magnenat-Thalmann, A. Guye-Vuillème, D. Thalmann, K. Kamyab, and E. Mamdani. Avatar markup language. In *EGVE '02: Proceedings of the workshop on Virtual environments 2002*, pages 169–177, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[90] R. Kulpa, F. Multon, and B. Arnaldi. Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum, Eurographics 2005 special issue*, 24(3):343–352, 2005.

[91] T. Kwon and S. Y. Shin. Motion modeling for on-line locomotion synthesis. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 29–38, New York, NY, USA, 2005. ACM.

[92] C. C. L. Parameterization and parametric design of mannequins. *Computer-Aided Design*, 37(1):83–98, January 2005.

[93] L. W. Lacy. *Owl: Representing Information Using the Web Ontology Language.* Trafford Publishing, Victoria, BC, 2005.

[94] J. Laszlo, M. van de Panne, and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. *Computer Graphics*, 30(Annual Conference Series):155–162, 1996.

[95] D. Le, R. Boulic, and D. Thalmann. Integrating Age Attributes to Virtual Human Locomotion. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV-5/W10, 2003.

[96] J. Lee and S. Yong. A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 39–48, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[97] S.-H. Lee and D. Terzopoulos. Heads up!: biomechanical modeling and neuromuscular control of the neck. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1188–1198, New York, NY, USA, 2006. ACM.

[98] W.-S. Lee, J. Gu, and N. Magnenat-Thalmann. Generating animatable 3D virtual humans from photographs. In M. Gross and F. R. A. Hopgood, editors, *Computer Graphics Forum (Eurographics 2000)*, volume 19(3), 2000.

[99] M. Mancini, B. Hartmann, and C. Pelachaud. Non-verbal behaviors expressivity and their representation. Technical report, PF-star report 3.

[100] S. Marini, M. Spagnuolo, and B. Falcidieno. Structural shape prototypes for the automatic classification of 3d objects. *IEEE Computer Graphics and Applications*, 27(4):28–37, 2007.

[101] A. Marriott, S. Beard, J. Stallo, and Q. Huynh. VHML-Directing a Talking Head. *Active Media Technology*, 2001.

[102] S. Marsella and J. Gratch. A step toward irrationality: using emotion to change belief. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 334–341, New York, NY, USA, 2002. ACM Press.

[103] S. Marsella and J. Gratch. EMA: A computational model of appraisal dynamics. *Cybernetics and Systems*, pages 601–606, 2006.

[104] B. McBride. Jena: a semantic web toolkit. *IEEE Internet Computing*, 6(6):55–59, Nov/Dec 2002.

[105] R. McCrae and O. John. An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215, 1992.

[106] L. Moccozet, F. Dellas, N. Magnenat-Thalmann, S. Biasotti, M. Mortara, B. Falcidieno, P. Min, and R. Veltkamp. Animatable human body model reconstruction from 3d scan data using templates. 2004.

[107] MPEG-7. Moving picture experts group iso/iec. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, October 2004. version 10.

[108] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 137–146, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[109] I. J. N4668. Overview of the mpeg-4 standard, March 2002.

[110] nRQL. new racer query language (nrql) interface to owl ontologies in protégé. http://www.cs.concordia.ca/k_bhoopa/nrql.html.

[111] E.-F. I. N. of Excellence aceMedia. Towards a common multimedia ontology framework. Technical report, contract number 001765, April 2007.

[112] E.-F. I. N. of Excellence AIM@SHAPE. contract number 506766 (official web site), http://www.aimatshape.net.

[113] E.-F. I. N. of Excellence HUMAINE. contract number 507422 (official web site), http://emotion-research.net.

[114] A. Ortony, G. Clore, and A. Collins. *The cognitive structure of emotion.* Cambridge University Press., Cambridge, UK, 1988.

[115] F. I. Parke and K. Waters. *Computer facial animation.* A. K. Peters, Ltd., Natick, MA, USA, 1996.

[116] J. Peltz. HumanML: The Vision, Jay Peltz. *DM Direct Newsletter*, 2005.
[117] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. *Computer Graphics*, 30(Annual Conference Series):205–216, 1996.
[118] A. Peternier, D. Thalmann, and F. Vexo. Mental vision: a computer graphics teaching platform. In *proceedings of the 2006 Edutainment Conference*, pages 223–232, 2006.
[119] K. Petridis, I. Kompatsiaris, M. Strintzis, S. Bloehdorn, S. Handschuh, S. Staab, N. Simou, V. Tzouvaras, and Y. Avrithis. Knowledge Representation for Semantic Multimedia Content Analysis and Reasoning. *Knowledge-Based Media Analysis for Self-Adaptive and Agile Multi-Media, Proceedings of the European Workshop for the Integration of Knwoledge, Semantics and Digital Media Technology, EWIMT*, 2004:0–902, 2004.
[120] A. PhysX. Ageia physx system software. http://www.ageia.com/drivers/drivers.html.
[121] R. Plutchik. *Emotion: a psychoevolutionary synthesis.* Harper and Now, New York, 1979.
[122] M. Ponder, G. Papagiannakis, T. Molet, N. Magnenat-Thalmann, and D. Thalmann. Vhd++ development framework: Towards extendible, component based vr/ar simulation engine featuring advanced virtual character technologies. *cgi*, 00:96, 2003.
[123] J. Potter. The big five personality test. http://www.outofservice.com/bigfive/, (c) 1996-2007 Atof Inc.
[124] M. Poznanski and P. Thagard. Changing personalities: towards realistic virtual characters. *J. Exp. Theor. Artif. Intell.*, 17(3):221–241, 2005.
[125] M. Pratscher, P. Coleman, J. Laszlo, and K. Singh. Outside-in anatomy based character rigging. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 329–338, New York, NY, USA, 2005. ACM.
[126] N. Pronost, G. Dumont, G. Berillon, and G. Nicolas. Morphological and stance interpolations in database for simulating bipedalism of virtual humans. *The Visual Computer*, 22(1):4–13, 2006.
[127] Protégé. (c) 2005 stanford medical informatics. http://protege.stanford.edu/index.html.
[128] J. Quinlan. *C4. 5: programs for machine learning.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1993.
[129] RacerPro. Renamed abox and concept expression reasoner. http://www.racer-systems.com.
[130] A. Raouzaiou, E. Spyrou, K. Karpouzis, and S. Kollias. Emotion synthesis: an intermediate expressions generator system in the mpeg-4 framework. International Workshop VLBV05, 15-16 September 2005, Sardinia, Italy, 2005.
[131] A. Raouzaiou, N. Tsapatsoulis, K. Karpouzis, and S. Kollias. Parameterized facial expression synthesis based on mpeg-4. *EURASIP J. Appl. Signal Process.*, 2002(1):1021–1038, 2002.
[132] P. S. A. Reitsma and N. S. Pollard. Evaluating motion graphs for character animation. *ACM Trans. Graph.*, 26(4):18, 2007.
[133] B. Rosenhahn, L. He, and R. Klette. Automatic human model generation. In *The 11th International Conference on Computer Analysis of Images and Patterns ICAṔ05*, pages 41–48, 2005.
[134] K. SCHERER. Appraisal considered as a process of multilevel sequential checking. *Appraisal Processes in Emotion: Theory, Methods, Research*, 2001.
[135] S. Schertenleib, M. Gutierrez, F. Vexo, and D. Thalmann. Conducting a virtual orchestra. *IEEE MultiMedia*, 11(3):40–49, 2004.
[136] H. Seo and N. Magnenat-Thalmann. An example-based approach to human body manipulation. *Graph. Models*, 66(1):1–23, 2004.
[137] N. Shadbolt, T. Berners-Lee, and W. Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.
[138] A. Shapiro, D. Chu, and P. Faloutsos. The dynamic controller toolkit. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation - Posters and Demos*, pages 25–26. Eurographics Association, 2006.
[139] D. R. D. Silva, C. A. Siebra, J. L. F. Valadares, A. C. Frery, J. R. Falcão, and G. L. Ramalho. Synthetic actor model for long-term computer games. *Virtual Reality*, 5:1–10, 2001.
[140] K. Sims. Evolving virtual creatures. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22, New York, NY, USA, 1994. ACM.
[141] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Course Technology, August 1999.
[142] S. Staab, R. Studer, H.-P. Schnurr, and Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26–34, 2001.
[143] C. D. Y. K. M. Strintzis. Ontology-based access to multimedia cultural heritage collections - the reach project. *Computer as a Tool, 2005. EUROCON 2005.The International Conference on*, 1:151–154, 2005.
[144] B. Tang, Z. Pan, L. Zheng, and M. Zhang. Simulating reactive motions for motion capture animation. In *Advances in Computer Graphics, LNCS 4035*, pages 531–537. Springer, 2006.

[145] E. Tanguy, J. J. Bryson, and P. J. Willis. A Dynamic Emotion Representation Model Within a Facial. Animation System. *The International Journal of Humanoid Robotics*, 2006.

[146] A. Tekalp and J. Ostermann. Face and 2-D mesh animation in MPEG-4. *Signal Processing: Image Communication*, 15(4-5):387–421, 2000.

[147] E. Terzi, A. Vakali, and M.-S. Hacid. *Web Technologies and Applications*, volume 2642 of *Lecture Notes in Computer Science*, chapter Knowledge Representation, Ontologies, and the Semantic Web, page 594. 2003.

[148] D. Terzopoulos and K. Waters. Physically-based facial modelling, analysis, and animation. In *The Journal of Visualization and Computer Animation*, volume 1(2), pages 73–80, 1990.

[149] Y. Tijerino, M. Yoshida, S. Abe, and F. Kishino. A shape knowledge representation scheme and its application on a multi-modal interface for a virtual space teleconferencing system. In *Proceedings 4th IEEE International Workshop on Robot and Human Communication*, pages 259–264, 1995.

[150] D. Tolani, A. Goswami, and N. I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5):353–388, 2000.

[151] N. F. Troje, J. Sadr, H. Geyer, and K. Nakayama. Adaptation aftereffects in the perception of gender from biological motion. *Journal of Vision*, 6(8):7:850–857, 2006.

[152] G. Vasilakis, M. Pitikakis, M. Vavalis, and C. Houstis. A semantic based search engine for 3d shapes: design and early prototype implementation. *Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the (Ref. No. 2005/11099)*, pages 391–397, Nov. - 1 Dec. 2005.

[153] M.-L. Viaud and H. Yahia. Facial animation with wrinkles. 1992.

[154] V. Vinayagamoorthy, M. Gillies, A. Steed, E. Tanguy, X. Pan, C. Loscos, and M. Slate. Building expression into virtual characters. In *Eurographics Conference State of the Art Reports*, 2006.

[155] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 24, New York, NY, USA, 2006. ACM.

[156] W3C. Daml+oil (march 2001) reference description. http://www.w3.org/tr/daml+oil-reference.

[157] W3C. Owl web ontology language. http://www.w3.org/tr/owl-features/.

[158] W3C. Resource description framework (rdf): Concepts and abstract syntax. http://www.w3.org/tr/rdf-concepts/.

[159] W3C. Resource description framework schema (rdfs). specification 1.0. http://www.w3.org/tr/2000/cr-rdf-schema-20000327/.

[160] W3C. Semantic web rule language . http://www.w3.org/submission/swrl/.

[161] W3C. Sparql query language for rdf. http://www.w3.org/tr/rdf-sparql-query/.

[162] H. G. Wallbott. Bodily expression of emotion. *European Journal of Social Psychology*, 28(6):879–896, December 1998.

[163] C. M. Whissel. *The dictionary of affect in language R. Plutchnik and H. Kellerman (Eds) Emotion: Theory, research and experience: vol 4, The measurement of emotions.* Academic Press, New York, 1989.

[164] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* 1999.

[165] W. L. Wooten and J. K. Hodgins. Animation of human diving. *Computer Graphics Forum*, 15(1):3–14, 1996.

[166] Y. Wu, N. M. Thalmann, and D. Thalmann. A dynamic wrinkle model in facial animation and skin ageing. *The Journal of Visualization and Computer Animation*, 6(4):195–206, 1995.

[167] J. Xiao, Y. Zhuang, T. Yang, and F. Wu. *Advances in Computer Graphics*, volume 4035/2006 of *Lecture Notes in Computer Science*, chapter An Efficient Keyframe Extraction from Motion Capture Data, pages 494–501. Springer, 2006.

[168] P.-F. Yang, J. Laszlo, and K. Singh. Layered dynamic control for interactive character swimming. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 39–47, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[169] X. Yang and J. J. Zhang. Automatic muscle generation for character skin deformation. *Comput. Animat. Virtual Worlds*, 17(3‐4):293–303, 2006.

[170] Q. Zhang, Z. Liu, G. Quo, D. Terzopoulos, and H.-Y. Shum. Geometry-driven photorealistic facial expression synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, 12(1):48–60, Jan.-Feb. 2006.

[171] L. Zhao. *Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures.* PhD thesis, CIS, University of Pennsylvania, 2001.

[172] V. B. Zordan and J. K. Hodgins. Motion capture-driven simulations that hit and react. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 89–96, New York, NY, USA, 2002. ACM Press.

[173] V. B. Zordan, A. Majkowska, B. Chiu, and M. Fast. Dynamic response for motion capture animation. *ACM Trans. Graph.*, 24(3):697–701, 2005.

Personal information

Surname(s) / First name(s): **García Rojas M., Alejandra**

Address(es): Rue Cité Devant 6, 1005-CH, Lausanne, Switzerland

Telephone(s): +41 79 811 5839 Mobile: +41 21 311 5838

Email(s): alegrm@gmail.com

Nationality(-ies): Mexican

Date of birth: 20th November 1977

Desired employment/ Occupational field: **Computer Graphics and Virtual Reality, Semantic Web, Web design and development.**

Work experience

Since Feb 2005: **Doctorate Assistant**

Employer: Virtual Reality Laboratory at EPFL. Lausanne, Switzerland. http://vrlab.epfl.ch

Main activities and responsibilities: Collaboration to European projects NoE AIM@SHAPE and HUMAINE. Research on the creation of 3D animated virtual humans using Ontology Web Language. Thesis title "Semantics for Virtual Humans".

Apr 2004 to Dec 2004: **System Administrator and Web Developer**

Employer: Travelweb SL. Madrid, Spain. http://www.travelweb.es

Main activities and responsibilities: Administration of the selling system in Citrix of four travel agencies. Development of an Intranet web site in JSP and MySQL for communication and administration proposes.

Jan 2001 to Oct 2003: **Systems Analyst**

Employer: Electronic Data Systems of México. Mexico City, México. http://www.eds.com.mx

Main activities and responsibilities: Analysis, development and implementation of reports of the different incomes of Aeromexico S.A. for the finances department. Tools used: Tomcat, JSP, Oracle and PL/SQL.

Jan 2000 to Dec 2001: **Programmer Analyst**

Employer: DigiPro S.A. de C.V. Mexico City, México. http://www.digipro.com.mx

Main activities and responsibilities: Development in VB and ADO of a client-server application to capture data to digitalize documents.

Education

Since Feb 2005: **Ph.D. in Computer Science, specialization in Virtual Reality**.

University: Virtual Reality Laboratory at EPFL. Lausanne, Switzerland.

Oct 2003 to Jul 2004: **Master in Interactive Digital Technologies**.

University: Universidad Complutense de Madrid, Spain.

Jun 1995 to Dec 1999: **Bachelor of Computer Systems Engineering**.

University: ITESM Campus Toluca, Toluca, México

Language Skills

Mother tongue(s): **Spanish**

Other language(s): English and French

Self-assessment
European level(*)

Understanding		Speaking		Writing
Listening	Reading	Spoken interaction	Spoken production	

English	C1	C1	B2	B2	C1
French	C1	B2	B2	B2	A2

(*) *Common European Framework of Reference (CEF) level*

Information Technologies Skills

Programming Languages C, C#, C++, Java and VB.

Data Bases MySQL, Oracle, PL/SQL, SQL Standard.

Web Development J2EE, JavaScript, VBScript, PHP, ASP, HTML, XSLT, XPath, CSS, XML, XMLSchema Web 2.0 (AJAX, RSS, XML, JSON) and Web 3.0 (OWL)

Modeling Tools UML, Three-tier architecture.

Specialized Programs Microsot .Net Framework, Eclipse, Protǵé (Ontology editor), Alias Maya, Adobe Illustrator, Photoshop, Flash, Microsoft Office.

Additional information

Publications

Journal Articles

G. Vasilakis, A. Garcia-Rojas, L. Papaleo, C. E. C. F. Robbiano, M. Spagnuolo, M. Vavalis, and M. Pitikakis, "Knowledge-based representation of 3d media," *International Journal of Software Engineering and Knowledge Engineering*, 2008

A. Garcia-Rojas, M. Gutierrez, and D. Thalmann, "Visual creation of inhabited 3D environments: An ontology-based approach," *Visual Computer*, 2008

A. Garcia-Rojas, F. Vexo, and D. Thalmann, "Semantic Representation of Individualized Reaction Movements for Virtual Human," *International Journal of Virtual Reality*, vol. 6, no. 1, pp. 25–32, 2007

M. Gutiérrez, A. García-Rojas, D. Thalmann, F. Vexo, L. Moccozet, N. Magnenat-Thalmann, M. Mortara, and M. Spagnuolo, "An ontology of virtual humans: Incorporating semantics into human shapes.," *The Visual Computer*, vol. 23, no. 3, pp. 207–218, 2007

A. Garcia-Rojas, F. Vexo, D. Thalmann, A. Raouzaiou, K. Karpouzis, S. Kollias, L. Moccozet, and N. Magnenat-Thalmann, "Emotional face expression profiles supported by virtual human ontology," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 259–269, 2006. VRLab. - EPFL, CH-1015 Lausanne, Switzerland

Conference Proceedings

A. Garcia-Rojas, M. Gutierrez, and D. Thalmann, "Simulation of Individual Spontaneous Reactive Behavior," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 143–150, 2008

A. Garcia-Rojas, F. Vexo, and D. Thalmann, "Individualized Reaction Movements For Virtual Humans," in *4th International Conference on Computer Graphics and Interactive techniques in Australasia and Southeast Asia*, pp. 79–85, 2006

A. Garcia-Rojas, F. Vexo, D. Thalmann, A. Raouzaiou, K. Karpouzis, and S. Kollias, "Emotional Body Expression Parameters In Virtual Human Ontology," in *Proceedings of 1st Int. Workshop on Shapes and Semantics*, pp. 63–70, 2006

L. Moccozet, A. Garcia-Rojas, F. Vexo, D. Thalmann, and N. Magnenat-Thalmann, “In Search for your own Virtual Individual,” in *Semantics And digital Media Technology conference (SAMT) 2006*, vol. 4306/2006, pp. 26–40, Springer Berlin / Heidelberg, 2006

A. Garcia-Rojas, M. Gutierrez, D. Thalmann, and F. Vexo, “Multimodal authoring tool for populating a database of emotional reactive animations,” in *Second International Workshop Machine Learning for Multimodal Interaction.*, vol. 3869, pp. 206–217, 2006. Virtual Reality Lab., Ecole Polytech. Fed. de Lausanne, Switzerland

A. Garcia-Rojas, D. Thalmann, F. Vexo, L. Moccozet, N. Magnenat-Thalmann, M. Spagnuolo, and M. Gutiérrez, “An Ontology of Virtual Humans: Incorporating Semantics into Human Shapes,” in *The 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT 2005)*, pp. 7– 14, 2005

Personal interests | Sports, Painting and Reading.