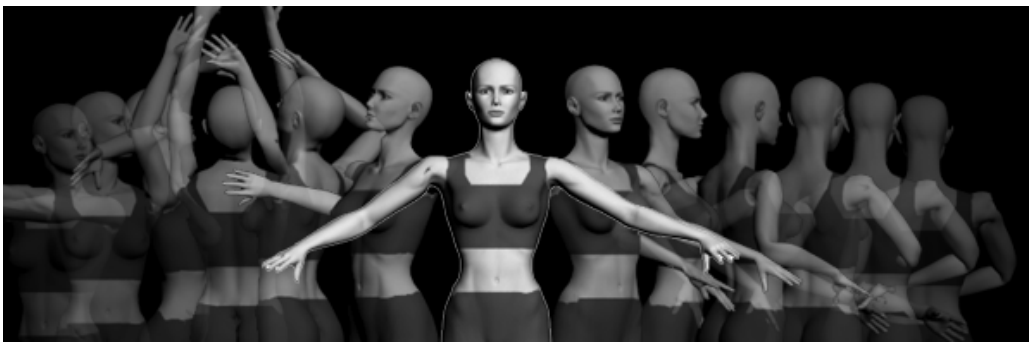
UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO



Gesticulation Expression in Virtual Humans

Celso Miguel de Melo
(Licenciado em Engenharia Informática e de Computadores)

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Orientador:	Doutora Ana Maria Severino de Almeida e Paiva
Co-Orientador:	Doutor Mário Rui Fonseca dos Santos Gomes
Júri	
Presidente:	Doutor Mário Rui Fonseca dos Santos Gomes
Vogais:	Doutora Ana Maria Severino de Almeida e Paiva
	Doutor Stefan Kopp
	Doutor João António Madeiras Pereira

Dezembro/2006

Título: Expressão por Gesticulação em Humanos Virtuais

Nome: Celso Miguel de Melo

Mestrado em: Engenharia Informática e de Computadores

Orientador: Prof. Ana Maria Severino de Almeida e Paiva

Co-orientador: Prof. Mário Rui Fonseca dos Santos Gomes

Resumo

Os humanos expressam-se através dos seus corpos. Em particular, pensamento e emoções reflectem-se em gesticulação. Gesticulação é o tipo de gesto inconsciente, idiossincrático e não convencional que os humanos realizam em conversação ou narração. Este trabalho contribui para o esforço de captura da expressividade da gesticulação em mundos virtuais.

Em concreto, este trabalho propõe um modelo, fundamentado em psicolinguística, para expressão por gesticulação em humanos virtuais que suporta: (a) animação de gesticulação em tempo-real descrita por sequências de restrições estáticas (posições, orientações e formas de mão da Linguagem Gestual Portuguesa) e dinâmicas (perfis de movimento); (b) sincronização entre gesticulação e discurso sintetizado; (c) reprodução automática de anotações em GestuRA, um algoritmo de transcrição de gestos; (d) inspirando-se nas artes, expressão de emoções através de três canais do ambiente circundante – câmara, iluminação e música; (e) controlo de expressão através de uma linguagem abstracta, integrada e síncrona – Expression Markup Language.

Para avaliar o modelo, dois estudos em contexto de narrativa de histórias, envolvendo 147 pessoas, foram realizados. Estes comparam a expressão de um narrador humano com um virtual. Os resultados indicam que os gestos sintéticos comparam positivamente relativamente aos sintéticos e que a interpretação da história não varia significativamente entre narradores. Contudo, o narrador humano foi, ainda assim, preferido. Um terceiro estudo, envolvendo 50 pessoas, avaliou a expressão de emoções pelo ambiente. Os resultados indicam que os canais de iluminação e música são eficazes na expressão de emoções contudo, o canal da câmara pode ser melhorado.

Palavras-Chave:

Expressão por Gesticulação, Humanos Virtuais, Expressão pelo Ambiente, GestuRA, Expression Markup Language

Title: Gesticulation Expression in Virtual Humans

Name: Celso Miguel de Melo

Supervisor: Prof. Ana Maria Severino de Almeida e Paiva

Co-supervisor: Prof. Mário Rui Fonseca dos Santos Gomes

Abstract

Humans express themselves through the affordances of their bodies. In particular, thought and emotion are expressed through gesticulation. Gesticulation is the kind of unconscious, idiosyncratic and unconventional gestures humans do in conversation or narration. This work contributes to the effort of harnessing the power of gesticulation in digital worlds.

In concrete, this work proposes a psycholinguistics-based virtual human gesticulation expression model which supports: (a) real-time gesticulation animation described as sequences of constraints on static (Portuguese Sign Language hand shapes, orientation palm axis, orientation angle and handedness) and dynamic (motion profiles) features; (b) synchronization between gesticulation and synthesized speech; (c) automatic reproduction of annotations in GestuRA, a gesticulation transcription algorithm; (d) inspiring on the arts, expression of emotions through three environment channels – camera, illumination and music; (e) expression control through an abstract integrated synchronized language – Expression Markup Language.

To evaluate the model, two studies, involving 147 subjects, were conducted in storytelling contexts where a human storyteller is compared to a virtual one. Results indicate that synthetic gestures fare well when compared to real gestures and story interpretation does not differ significantly between storytellers. However, the human storyteller was still preferred by most subjects. A third study, involving 50 subjects, was conducted to evaluate expression of emotions through the environment. Results indicate that the illumination and music channel are effective but, the camera channel can be further improved.

Keywords:

Gesticulation Expression, Virtual Humans, Environment Expression, GestuRA, Expressive Markup Language

Acknowledgments

Universal education, sustainable development, climate change, alternative energy sources, pollution, hunger, poverty, gender equality, ageing, diseases, crime, terrorism, technology transfer, capacity building, globalization, etcetera. There is so much in the world agenda, it's overwhelming. To take on such a burden, the next generation relies on previous ones. As we, young adults, prepare to undertake such an endeavor, your guidance, encouragement and knowledge is invaluable. To you, angels in my life, I am thankful.

For her dedication and passion to research, for breeding talent in such an unusual and promising field, for believing and encouraging me, for giving me space to freely explore my ideas, Professor Ana Paiva deserves my sincere gratitude. To the members of the GAIPS research group, thank you for providing me with a sense of community and for making these years, altogether, more enjoyable. Finally, I thank all Professors who thought me so much.

A very special thanks to my mother, for her dedication to family and unconditional love, and to my father, for being proud of me, for companionship and unconditional support. I also thank my family and friends for all those shared moments which give meaning to the very vision this work lives by.

To the stranger with the look. The subtle smile. The silent words. Every day, you're kindness and respect gave me strength. To the woman with the voice. To the fearless leader. To the one with the expressive clothing. And to all Portuguese man and women who, through ambition, creativity and dedication, have shown that we can do great things, inspiring and pushing me to the limit.

Contents

List of Figures

List of Tables

1 Introduction

“After millions of years of natural selection, human beings have some serious competition for their lofty perch on the evolutionary ladder—and the challenger has only been evolving for a few decades. Some contend that anything we can do, ‘virtual humans’ can do better."

Anonymous (adapted)

1.1 The Vision

Humans express themselves through the affordances of their bodies. Thought is verbalized through speech and visualized in gesticulation. Emotion reflects in the face, voice and body. Personality and culture distinguish the individual and community style of body, facial and vocal expression. While conversing, interactional cues, such as giving floor, are conveyed through gaze, nods and gesture. Furthermore, not only is communicative intent distributed across modalities, it is also expressed in a synchronized fashion. Such is the case, for instance, for a gesture which complements the co-occurring verbal message. In effect, multimodal expression endows humans with the ability to efficiently communicate complex messages.

It is the power of human multimodal expression that we wish to capture in digital worlds. This is the field of virtual humans. The purpose is to simulate in digital technology the properties of nature and affordances of the human body which support expression. Sound is no more than the disturbance of air particles. Voice is the exercise of control over such a disturbance. Facial expressions are just configurations of the muscles of the face. Gestures are no more than sequences in time of body postures with certain dynamics properties. All these things can be simulated by a machine and some, with varying degrees of success, already have.

The field’s short- and mid-term applications are vast: analysis and validation of human sciences’ theories; new human-computer interface paradigms such as interface agents, with whom a user dialogues and delegates tasks; emotional agents, endowed with emotional intelligence, which are sensitive to the user’s needs and capable of executing social tasks; pedagogical agents capable of motivating and rewarding students through multimodal expression; entertainment, including computer games, cinema, etc.

1.2 The Problem

Multimodal expression is a broad concept. This work focuses on a single aspect of it: humans express thought through gesticulation. Gesticulation is the kind of unconscious, idiosyncratic and unconventional gestures humans do in conversation or narration. They tend to focus on the arms and hands, though other body parts may be involved. Furthermore, gesticulation and speech, which are believed to be different sides of the same mental process, co-express the same underlying idea unit and synchronize at various levels. [1][2]

The problem of modeling gesticulation can be divided into the sub-problems of generation and execution. Gesticulation generation concerns with the simulation of the speech and gesture production process, i.e., the distribution of communicative intent across modalities and selection of proper surface realizations which, in the case of gestures, correspond to constraints on static and dynamic features of the arms and hands. Gesticulation execution is more akin to the body and concerns with the actual animation, in a synchronized fashion, of the static and dynamic constraints which define the gesture. This work focuses on the execution sub-problem.

Gestures, including gesticulation[1], also reflect emotions. For instance, a sad person gestures slowly, while a happy one energetically. Thus, a secondary problem addressed by this work is that of conferring emotion qualities to gesticulation without altering its semantic meaning.

1.3 The Contribution

Precisely, the contribution of this work is a gesticulation expression model which supports:

- Real-time gesticulation animation described as sequences of constraints on static (Portuguese Sign Language hand shapes, orientations and positions) and dynamic (motion profiles) features;
- Multimodal synchronization between gesticulation and speech;
- Automatic reproduction of annotated gesticulation according to GestuRA, a gesture transcription algorithm;
- Emotion expression through three environment channels – camera, illumination and music;
- Expression control through a markup integrated synchronized language – Expression Markup Language (EML).

[1] Chapter 2 clarifies that gesticulation is only one form of gesture, namely, the kind which accompanies speech in conversation or narration contexts.

1.4 Overview

The rest of this document is organized as follows:

- **Chapter 2, Background and Related Work.** This chapter starts by describing psycholinguistics gesticulation research which is essential to understand the requisites for the proposed model; overviews related work in computer graphics and computational psycholinguistics; overviews related work in the expression of emotions through gesture; and, finalizes with an overview of markup languages for multimodal expression control;
- **Chapter 3, The Model.** This chapter details the proposed gesticulation expression model. First, the virtual human three-layer architecture is introduced. Then, deterministic expression, which supports keyframe animation, non-deterministic expression, which supports robotics-based procedural animation, and vocal expression, which integrates a text-to-speech system, are explained. These modalities support gesticulation expression which is described next. Here, supported static and dynamic features, gesture-speech synchronization, automatic reproduction of GestuRA annotations and expression of emotions through the environment are presented. Finally, the chapter ends with the description of a markup integrated synchronized language for expression control;
- **Chapter 4, Evaluation.** This chapter reports three studies, involving 197 subjects, conducted to evaluate the model. The first two compare a human storyteller with a virtual one in the narration of the Portuguese traditional story “The White Rabbit”. The third evaluates this work’s approach for the expression of emotions through the environment;
- **Chapter 5, Conclusions and Future Work.** This chapter draws some conclusions, discusses this work’s contribution and proposes future work;
- **Appendices.** Several appendices are attached at the end of the document. These present, successively, the formal definition of the virtual human skeleton, anthropometry data for robotics manipulators, the GestuRA transcription algorithm and the Expression Markup Language specification.

2 Background and Related Work

This chapter overviews background research and related work, focusing on the following:

- Gesticulation research in psycholinguistics including gesture continua, the speech-gesticulation relation and gesticulation structure, dimensions and models;
- Gesticulation computational systems in computer graphics and psycholinguistics;
- Previous work on expression of emotions through gestures;
- Markup languages for multimodal expression control.

Human gestures, in a broad sense, range from goal-oriented *voluntary gestures* and subconscious *involuntary gestures*. Voluntary gestures include, for instance, walking towards some place or talking. Involuntary gestures occur for biological reasons including, for instance, eye blinking and breathing. Between these extremes, several subclasses have been studied [4]: gestures of deception, gestures of seduction, gestures in the classroom, power gestures in business environments, emotional gestures, etc. This work focuses on one which could be named *communicative gestures* subclass.

In the communicative gestures subclass, Kendon ([5] in [1]) distinguishes four types: gesticulation; pantomime; emblems; and sign language. *Pantomime* relates to gestures which occur without conversation. *Emblems* are culturally dependent gestures which have conventionalized meaning. An example is the American V (of victory) gesture, executed with the palm facing the listener. *Sign languages* consist of communication languages expressed through visible hand gestures. Examples are languages used by the deaf. Finally, *gesticulation*, which is the focus of this work, is the kind of gestures humans do in narrations or conversations. McNeill further characterizes these types according to four *continua*: [6]

- The first *continuum* defines the relation between gestures and speech:

Gesticulation →	Emblems →	Pantomime →	Sign Language
Obligatory presence of speech	Optional presence of speech	Obligatory absence of speech	Obligatory absence of speech

- The second continuum defines the relation between gestures and linguistic properties:

Gesticulation →	Pantomime →	Emblems →	Sign Language
Obligatory absence of linguistic properties	Obligatory absence of linguistic properties	Some linguistic properties present	Linguistic properties present

- The third *continuum* defines the relation between gestures and conventions:

Gesticulation →	Pantomime →	Emblems →	Sign Language
Not conventionalized	Not conventionalized	Partially conventionalized	Totally conventionalized

- The fourth *continuum* defines the gestures' semiotic properties. Here, being *global* means that gesture meaning is interpreted in a top-down fashion, i.e., the meaning of the parts results from the meaning of the whole. This property contrasts with the segmentation of verbal language. A *synthetic* gesture means that it concentrates in unique symbolic form distinct meanings which expand throughout the accompanying verbal expression.

Gesticulation →	Pantomime →	Emblems →	Sign Language
Global and synthetic	Global and analytic	Segmented and synthetic	Segmented and analytic

2.1 Gesticulation Research

Gesticulation is the kind of idiosyncratic, unconventional and unconscious gestures humans do in narrations or conversations [1]. They tend to focus on arms and hands, though other body parts may be involved [2]. Furthermore, they intimately relate to the accompanying speech.

2.1.1 Gesticulation and Speech

Gestures which occur when a person is speaking are a manifestation of *verbal thought*. Verbal thought, which does not include all forms of thought, nor all forms of speech, is the kind of thought which resides in the intersection between thought and speech. [2]

It is believed that speech and gesticulation are manifestations of the same underlying process. Thus, gesticulation and speech co-express the same underlying idea unit possibly in non-redundant ways. McNeill justifies as follows: (a) gesticulation occurs only in conversation; (b) both coordinate at the semantic and pragmatic levels; (c) both are synchronous; (d) they develop together in childhood; (e) they deteriorate together in aphasia. [1]

Through gesticulation, however, information is conveyed in a fundamentally different way than through speech: (a) gesticulation is not combinatoric – two gestures produced together do not combine to form a larger one with a complex meaning; (b) there is no hierarchical structure in gesticulation as in language; (c) gesticulation does not share linguistic properties found on verbal communication. [1]

2.1.2 Gesticulation Structure

According to how it unfolds in time, gesticulation can be structured hierarchically into *units*, *phrases* and *phases* ([7][8] in [1]). A unit, which is the highest level in Kendon's hierarchy, is the time interval between successive rests of the limbs. A unit may contain various phrases. A phrase is what is intuitively called 'gesture' [2]. A phrase consists of various phases: (a) *preparation*, where the limbs position themselves to initiate the gesture; (b) *pre-stroke hold*, where a hold occurs just before the stroke; (c) *stroke*, which is the only obligatory phase, is where actual meaning is conferred. The stroke is synchronous with its co-expressive speech 90% of the time [9] and, when asynchronous, precede the semantically related speech; (d) *post-stroke hold*, where a hold occurs after the stroke, before initiating retraction; (e) *retraction*, where the limbs return to rest. Preparation, stroke and retraction were introduced by Kendon ([10] in [1]) and the holds by Kita ([11] in [1]).

2.1.3 Gesticulation Dimensions

McNeill and colleagues characterize gesticulation according to four dimensions: [1][2]

- *Iconicity* – which refers to gesticulation features which demonstrate through its shape some characteristic of the action or event being described. An example is as follows: "They [arrived][2] through the Milky Way together / Invoked on behalf of Tonante"[3]. The narrator while pronouncing the phrase slides the right arm to the left with palm facing down, as suggesting that "they" floated. Notice that the manner of motion was not explicit in speech;
- *Metaphoricity* – which is similar to iconics however, referring to abstract concepts. Representative is the *process* and *conduit* gestures which represent, respectively, ongoing processes and abstract entities "objectification". An example of the latter is "I shall [give you], my Lord, relation / of myself, of the law and the weapons I brought" while the speaker forms a "box" gesture which it hands over to the listener;
- *Deixis* – which refers to features which situate in the physical space, surrounding the speaker, concrete and abstract concepts in speech. An example is "Who are you, which land is [this] you inhabit / Or have you [of India] any signs?" where the narrator points to the floor while pronouncing "this" and to the back while pronouncing "of India";

[2] In this document, straight parentesis in the text represent co-occurence of gesture with speech.

[3] All examples in this section come from L. Camões, *Os Lusíadas*.

- *Beats* – which refer to small baton like movements that do not change in form with the accompanying speech. They serve a pragmatic function occurring, for instance, with comments on one's own linguistic contribution, speech repairs, and reported speech.

According to McNeill ([2], p.42), "multiplicity of semiotic dimensions is an almost universal occurrence in gesture". Thus, it makes more sense to speak of dimensions and saliency rather than exclusive categories and hierarchy.

2.1.4 Gesticulation Models

The *growth point model*, proposed by McNeill [1][2], claims that language is inseparable from imagery. Imagery, here, are gestures. This language-imagery dialectic combines the static and dynamic dimensions of language. The former sees language as an object and focuses on the study of linguistic forms. The later sees language as a process and focuses on the mutual impact of language and thought as utterances unfold. The dialectic combines these dimensions through the concept of a *growth point*, which represents a single idea unit which unfolds as utterance and gesture. In the growth point two modes of thinking – linguistic and imagistic – are active and this coexistence of unlike modes results in instability which must be resolved by accessing forms on the inherently stable static dimension. The resolution *materializes* as utterance and gesture. Furthermore, materialization increases with the unpredictability of the idea unit, i.e., with its opposition to the current context. Thus, the less predictable the idea, the more complex the gesture will be. Finally, context manifests in gestures through *catchments*, i.e., recurring forms which refer to the discourse unit's theme.

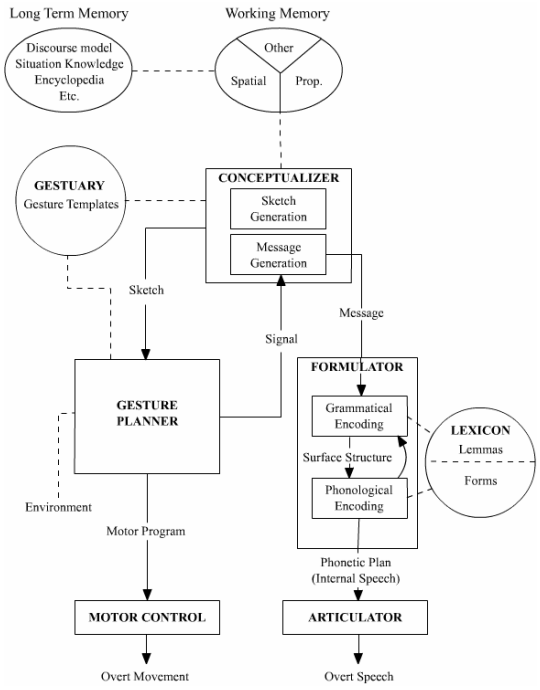
Contrasting to the non-modular growth point model, various modular *information processing models* have been proposed. An information processing model assumes the brain functions by processing information and, furthermore, may assume information is stored, in some representation, in modules on which cognitive processes operate [13]. Relevant models were proposed by de Ruiter [13], Krauss [14], Kita [15] and Cassell [16] – see Fig. 2.1.

A common trace in de Ruiter's, Krauss' and Kita's information processing models is that they extend Levelt's *speaking* model [17] for speech production with gesture production. The speaking model is structured into modules. The first is the *conceptualizer* which transforms communicative intention into a propositional form called the *preverbal message*. This message is, then, passed to the *formulator*, which has two components: a *grammatical encoder* that creates the utterance surface structure; and a *phonological encoder* that produces a phonetic plan. The formulator accesses a *lexicon* which stores lexical items' semantic and syntactic properties. Finally, the phonetic plan is fed into the *articulator* which converts it to audible speech. This model is shown on the right side of Fig. 2.1-(b).

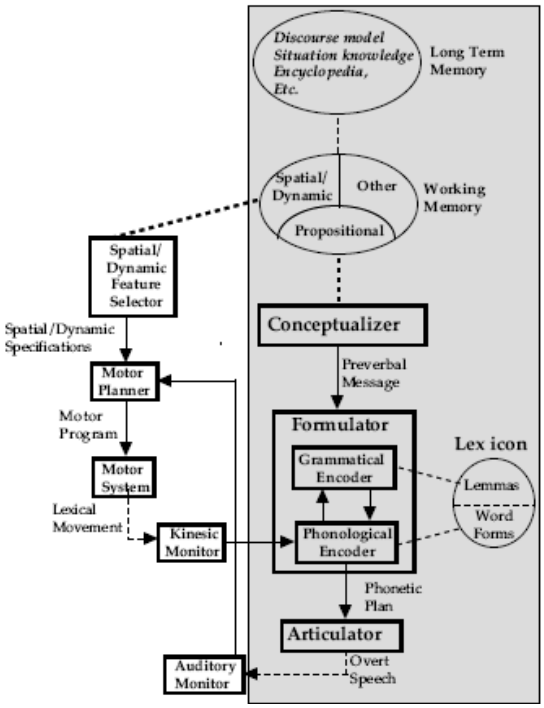
The *sketch model* by de Ruiter [13] – Fig. 2.1-(a) – explains comprehensively how communicative intent leads to gesture form. Gesture production starts in the conceptualizer as it handles similar problems for speech and has access to working memory which holds imagistic information. Gestures occur when there is something new to communicate, when information is hard to encode verbally, when it is necessary to enhance communication or when speech fails. Regarding form, iconics are generated from features extracted from imagery. Emblems, which are lexicalized, are retrieved from a *gestuary* which stores predefined gestures. However, instead of storing complete gesture specifications, the gestuary holds *templates* which constrain only the necessary features. Pantomimics, or "enactment of a certain movement performed by a person or animate object in imagistic representation", are generated from *action schemas* ([18] in [13]) in the gestuary. Deictics are generated from a direction vector and conventionalized hand shapes ([19] in [13]) in the gestuary. In this way, parallel to the preverbal message, the conceptualizer outputs a *sketch* which holds gesture form information. The sketch is fed into the *gesture planner* which converts it into motor programs for execution. In concrete, the gesture planner handles: (a) *body-part allocation* where, for instance, for a deictic if both hands are occupied, the head might be selected; (b) *environment influence*, which refers to further constraints imposed by the surrounding environment; (c) *gesture fusion*, which refers to gesture combination. Finally, synchronization, which is achieved through signal passing between modules, handles the following phenomena: (a) onset of gesture usually precedes the onset of speech; (b) gestural holds, including repetitive gestures; (c) gesture interruption.

Krauss's model [14] – Fig. 2.1-(b) – describes gesture production and its effects on lexical retrieval. The model focuses on *lexical gestures* which are similar to McNeill's iconics and metaphorics. One important assumption is that information conveyed through gestures is not necessarily part of the communicative intention. Thus, gestures originate in the speaker's working memory and not in the conceptualizer as in de Ruiter's model. Contrasting to de Ruiter's and McNeill's assumption of an imagistic component in knowledge, Krauss' model is a *featural model*, i.e., concepts are represented as sets of elementary features. In concrete, concepts are represented through propositional and non-propositional features. Non-propositional features refer to visuospatial aspects. Features in one format can be converted into the other. Regarding gesture production, the s*patial/dynamic feature selector* selects a subset of the concept's non-propositional features to pass down to the *motor planner* which generates form. Regarding speech production, the conceptualizer selects relevant features to express verbally, translating non-propositional into propositional features if necessary. Gestural facilitation of speech is achieved through the *kinesic motor* which sends visuospatial information to the phonological encoder and, thus, facilitates word retrieval by a process of

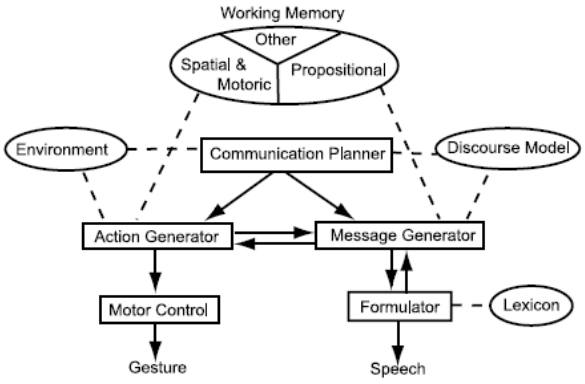
cross-modal priming. Finally, regarding synchronization, when the lexical affiliate concludes, the auditory monitor explicitly sends a signal to the motor planner for gesture termination.



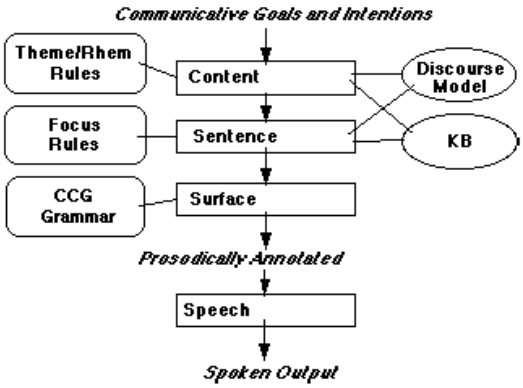
(a) de Ruiter model [13]



(b) Krauss model [14]



(c) Kita & Özyürek model [15]



(d) Cassel & Prevost model [16]

Fig. 2.1 Information processing models for speech and gesture production.

Kita and Özyürek [15] – Fig. 2.1-(c) – propose a class of models based on the *interface hypothesis* [20] which states that gestures are culturally dependent in that they are influenced by the speaker's language. This hypothesis contrasts to Krauss' and de Ruiter's models which say that gestures result from imagery and are not influenced by language. In this class of models, Levelt's conceptualizer is split into two modules: (a) the *communication planner* which generates communicative intention and defines which modalities will be involved; (b) the *message generator* which, considering the communicative intent and context, formulates

propositions to verbalize. Regarding gesture production, content is influenced by: (a) communicative intent; (b) visuospatial information in working memory; (c) "language formulation possibilities" constraints via an online feedback from the formulator through the message generator. The idea is that context dependent communicative intent, as defined by the communication planner, is passed to the *action generator*, responsible for general action planning including gestures, which interactively with the message generator distribute content across the two modalities. Interplay between these generators is similar to the imagery-language dialectic in McNeill's growth point model. As this class of models focus only on gesture content, synchronization is not explained.

Cassell and Prevost [16] – Fig. 2.1-(d) – propose a speech and gesture production model which expands on Prevost's model [21]. Here, speech production, similarly to Levelt's speaking model, follows three stages: (a) *content planning*, where high-level propositions are defined to meet speaker's goals and *theme/rheme* distinctions specified. The theme refers to contextual information in an utterance and the rheme to new or contrastive information; (b) *sentence planning*, where high-level propositions are translated into language specific sentences; (c) *surface realization*, where syntactic and prosodic constituents are defined for text-to-speech rendering. Regarding gesture production, alignment with rhematic material occurs in content planning, while alignment with intonation occurs in sentence planning. Distribution of features across modalities, as it is argued to be language specific, occurs in sentence planning. Finally, selected features are realized into gestural forms in the surface realization stage.

McNeill ([2], pp.132-136, and [22]) has been very critical of modular information processing models for gesture and speech production. First, he criticizes a gesture production model based on Levelt's speaking model:

> "These extensions share the same limitation, which derives ultimately from Speaking itself: they do not combine imagery and language into single units (growth points), and they are unable to describe incorporation of context into verbal thought." ([2], p.132)

Furthermore, McNeill argues that modular information processing models fail to model context. As context can only be represented as background external input to modules representing speech and gesture production, these models fail to capture the dynamic dimension of context on the imagery-language dialectic of the growth point model. In fact, he claims that in all of the aforementioned models "context has been excluded". Additionally, he claims that a symptom of this problem is that synchronization is achieved through signal passing and, thus, does not arise from thought itself.

2.1.5 Summary

Even though the speech and gesture production process is beyond the scope of this work, it is necessary to understand it in order to retrieve the necessary requisites for a gesticulation animation model. In concrete, the psycholinguistics gesticulation research presented in this section leads to the following requisites which are implemented in this work:

- *Gesticulation should, at least, span arms and hands*, as it tends to focus in these body parts;
- *Gesticulation and speech should be able to synchronize at the sub-second time granularity*, as they are believed to be different sides of the same underlying mental process and synchronize at the semantic and pragmatic levels;
- *It should be possible to describe gesticulation at the phase level*, as they distinguish parts which are motivated by physical, synchronization or meaning constraints. Furthermore, phases introduce a hierarchy of relevance which is used to choose which parts should adapt so as to conform to time warping or co-articulation synchronization constraints;
- *Gesticulation can be described through constraints on its features*, in concrete, as sequences of static (hand shape, orientation and position) and dynamic (motion profiles) constraints. The feature-based approach is justified for several reasons. First, describing gesticulation according to dimensions and saliency suggests that meaning distributes across the affordances of the upper limbs and hands and thus, rather than overall form a more granular (or feature-based) description is possible. Second, a feature-based approach is compatible with most speech and gesture production models: the imagistic component in McNeill's growth points ultimately materializes into gesture features; de Ruiter's sketch model revolves around the concept of gesture templates (in a gestuary) which correspond to constraints on features; Krauss actually considers knowledge representation as feature-based; finally, Kita & Özyürek and Cassel & Prevost even though not detailing gesture morphology, motivate their models with motion gestures described according to features.

2.2 Gesticulation Computational Systems

This section surveys relevant work in computer graphics and computational psycholinguistics.

2.2.1 Computer Graphics

Realistic hand simulation is about modeling the mechanical complexity and visual aspect of the hand. With this aim, various approaches have been followed. Thompson [23] proposes a system for medical professionals where multiple computerized tomographic scans are used to produce anatomically accurate kinematic models of the patient's hand. A good review of hand anatomy, biomechanics and anthropometry literature is described in [24]. Furthermore, this model

recognizes that hands differ among individuals. Wagner [25] also reached this conclusion by comparing pianist's hands anthropometry to regular people. Magnenat-Thalmann [26] explores realism through *joint-dependent local deformation* operators which are parameters influenced by specific joints that deform a subset of the hand surface. These operators simulate rounding at the joints and muscle inflation. Kunni et al ([27] in [28][24]) explore manifold mappings, consider inter-joint dependencies and animate simple gestures. Ip ([29] in [30]) proposes a hierarchical four-layer model: (a) *anatomy*, which models the hand anatomy; (b) *physics*, which uses *dynamics*, i.e., the control of motion based on forces, for physically-based motion; (c) *geometry*, which defines the hand skin mesh; (d) *image*, which correspond to low-level animation sequence. Representing learning-based approaches, neural networks have been applied to generate natural hand motion based on real-life examples [31][32]. Moccozet [33] proposes a three-layer hand model – *skeleton*, *muscle* and *skin* – where Dirichlet free-form deformations are used to simulate muscle and realistic skin deformation. Sibille [35] proposes a real-time generic anatomical hand model. Hand motion is based on dynamics, mass-spring meshes are used to calculate soft tissue deformations and, finally, the system handles collision detection. Albrecht [24] also proposes a real-time anatomical human hand model. Even though it does not handle collision detection as Sibille's model, motion is based on a more realistic hybrid muscle model: *pseudo-muscles* control motion of bones based on anatomical data and mechanical laws; *geometric muscles* deform the skin tissue using a mass-spring system. Furthermore, a deformation technique is proposed to create individual specific hand models. Tsang [36] proposes a skeletal musculo-tendon model of the human hand and forearm. The model "permits direct dynamics simulation, which accurately predicts hand and finger position given a set of muscle activations" and a "solution to the inverse problem of determining an optimal set of muscle activations given a pose or motion; muscle fatigue, injury or atrophy can also be specified, yielding different (…) solutions".

Due to its goal, simulation inherently contributes to gesticulation models. There is growing evidence that the body shapes and constraints thought [37]. There is growing evidence that gestures reveal the imagery of thought [1][2]. Thus, anatomical physically-based realistic hand simulation models, more than producing aesthetically pleasant animation, are necessary for the embodiment of the complex idiosyncratic gesticulation we see in humans.

Focusing on task oriented behavior simulation, many *grasping* solutions have been proposed. Grasping refers to the simulation of a natural hand grasp of an arbitrarily complex object. Magnenat-Thalmann [26] presented one of the first computational models. The model proposes a semi-automatic solution where angles are automatically calculated, but the animator has to explicitly position the hand and define contact points in both hand and object. Rijpkema

[38] proposes a knowledge-based solution inspired on the notion that how and where a grasp takes places is a function of the person's familiarity with the object. This familiarity is defined as similarity to standard shapes – blocks, spheres, torus, cones, etc. – to which parameterized canonical holds apply. Hand motion results from the combination of inverse kinematics, high-level behaviors (opening, closing and spreading groups of fingers) and predefined postures from a library. Sanso [39] proposes a heuristic approach which varies hand grasp according to the object's shape and dimension. Huang [28] extends [39] with multi-sensors, inspired on proximity sensors in robotics, placed at appropriate places in the hand to obtain more natural grasps. Furthermore, to improve on linear interpolation arms motion is based on the Lagrange-Euler equation. Pollard [40] proposes a grasping solution which combines physical and motion capture animation. This kind of hybrid solution has been explored before in other domains, as it brings together flexibility from physics and naturalness from motion capture [41].

Grasping solutions may contribute to iconics and metaphorics simulation. These correspond to gesture expression of the imagery of the concrete and abstract. However, how is this imagery to be represented and expressed? A solution could represent imagery has combinations of simple three-dimensional shapes (cubes, pyramids, spheres, etc.) and expression would be similar to a grasp of this shape. This idea has been explored at least in [42].

Music fingering task simulation has also been widely explored [25][31][43]. Here, the idea is to simulate proper finger placement and the complex coordination required to play musical instruments. Proposed solutions serve other tasks, including gesticulation.

Human-computer interaction sees in multimodality the next step towards easier and more natural human-computer interaction. A survey of multimodal interfaces can be found in [44][45]. Gesture plays a key-role in these systems. Here, focus lies on gesture recognition and interpretation. Several approaches for gesture recognition have been presented [46][47][48][49]. These can be divided at least according to two dimensions. The first distinguishes *three-dimensional* from *appearance-based* models. In the former case, gestures' three-dimensional features are explicitly recognized through a hand glove while in the latter case, gesture *images* are compared against patterns for feature extraction. The second dimension distinguishes *global* from *local* models. A global model focuses on wrist position and orientation while a local model on individual finger motions. The study of hand constraints [50] and relevant gesture features [49] are usually associated with this field. Knowledge from both these aspects contributes to realistic gesticulation simulation.

Presently, however, multimodal systems are far from being able to recognize and interpret gesticulation. First, taxonomies tend to focus on simple pointing, manipulative and conventionalized gestures as opposed to complex, idiosyncratic and unconventional. Second,

most of the existent work performs late *fusion* of multimodal input where, for instance, in a gesture and speech system, interpretation would occur only at the utterance level. Gesticulation requires fusion to be much earlier. Third, most systems ignore two-handed gestures. Naturally, researchers have realized these limitations and began to address them in recent systems.

Finally, a remark is due regarding the contribution of robotics to the field. Indeed, a lot of the theory used to build a dexterous robotic hand underlies the aforementioned computer graphics models. However, while computer graphics focuses on models which exist in virtual worlds, robotics focuses on robots which act in the real world. A survey of hand research in robotics can be found in [51].

2.2.2 Computational Psycholinguistics

Animated Conversation [52], developed by Cassell and colleagues, was the first computational system to try to simulate idiosyncratic, unconventional and unconscious gesticulation. Based on research in psycholinguistics, the model proposes a rule-based system capable of synchronizing gestures of the right type with co-occurring speech. However, the system has weaknesses [16][53]: first, gestures are selected from a predefined library; second, communicative intent is not distributed across modalities and thus, too many, and redundant, gestures are generated; finally, it is not in real time. In *Real Estate Agent (Rea)* [54][55][56][57][58][59], an *embodied conversational agent* capable of multimodal input recognition and output synthesis, is presented. Two kinds of information are distinguished: *propositional*, referring to communicative intent; and *interactional*, referring to conversation mediation. The process of speech and gesture production follows the theory described in [16] (see subsection 2.1.4). In concrete, distribution and realization of communicative intent across speech and gesture resorts to an extension of the *Sentence Planning Using Description (SPUD)* generator [60]. Here, *lexical descriptors* – lexical items and gesture features – are selected and organized into a grammatical structure that manifests “the right semantic and pragmatic coordination between speech and gesture”. Thus, Rea, solves the second of the aforementioned Animated Conversation’s weaknesses. Regarding architecture, the system is structured according to various modules. The *Input Manager* recognizes and fuses input modalities. Together, the *Understanding*, *Decision* and *Generation Modules* analyze input according to dialogue state and context, plan output behaviors and distribute them across modalities. Besides this deliberative behavior, the system also supports fast hardwired interactional reactions. Finally, the *Action Scheduler*, to which this work relates, executes and synchronizes behaviors. Synchronization is based on combinations of modality specific events – like the occurrence of a phoneme or word – and temporal constraints. Events may also be sent to upper modules. Furthermore, behaviors competing for body degrees-of-freedom are arbitrated according to priorities.

In [61] Cassell et al propose the *Behavior Expression Animation Toolkit (BEAT)* which receives as input text and automatically generates linguistically motivated synchronized nonverbal behavior and synthesized speech. The system is composed of a set of knowledge bases and modules. Regarding knowledge, there is a domain knowledge base with entries about objects and actions. Each entry may also define a predefined gesture which depicts it. Regarding modules, each successively receives as input and generates as output XML tagged text. First, the *Language Tagging* module annotates input text with linguistic and contextual information, namely: theme/rhemes; word newness; and contrasts. Then, the *Behavior Suggestion* module applies rule-based nonverbal generators to produce all possible nonverbal behaviors which are, then, selected using filters. Finally, the *Behavior Scheduling and Animation* module, converts the input text with linguistic and gesture annotations into a format which can be rendered in a text-to-speech and animation system. Synchronization between speech and nonverbal behavior can be based on an *absolute time animation plan*, which relies on the text-to-speech to obtain phoneme times to schedule the animation, or an *event-based plan*, which generates rules to be triggered in runtime as phonemes are rendered. However, BEAT is limited in two important ways: (a) the gesture production process is incompatible with most gesticulation models (see subsection 2.1.4) in the sense that it infers gestures from text instead of generating both text and gesture from communicative intent; (b) it is limited to predefined gesture forms.

Leveraging on the previous systems, Cassell et al [62][63] developed *Media lab Autonomous Conversational Kiosk* (*MACK)*, a mixed reality multimodal embodied conversational kiosk. MACK uses speech, gesture and instructions on a map placed between the agent and user to convey directions about locations. Gesture synthesis is based on a library of direction deictics and landmark iconics. Gesture selection and synchronization with speech relies on the aforementioned BEAT system.

Kopp and colleagues [64][65][66] developed a comprehensive model for gesture animation based on research in psycholinguistics and motor control theory. Based in [13], a knowledge base – the g*estuary* – holds gesture templates which consist of hierarchical trees where leaf nodes are gesture features and internal nodes represent parallel, sequential, symmetry and repetition constraints. Gesture features refer to the stroke phase and include static constraints (hand shape, orientation and location) and dynamic constraints (hand motion). Gesture production is structured in two steps: *gesture planning* and *generation and execution of motor command*. Gesture planning starts with selection of appropriate templates from the gestuary according to communicative intent, proceeds by instantiating the templates according to context and concludes by assigning appropriate temporal constraints. The gesture plan is then fed into a motor planner for animation. Motor planning automatically appends preparation and retraction

motion considering co-articulation effects and distributes constraints to motor subsystems according to affected body parts. The hand motor subsystem supports *HamNoSys* ([67] in [65]) hand shapes and stereotypical transitions. The arm subsystem animates sophisticated arm trajectories defined in Cartesian coordinates and smoothly interpolated resorting to non-uniform cubic B-splines which support specific velocity profiles such as, for instance, slowing down at turning points. Furthermore, the system supports keyframe body animation, muscle based facial animation and text synthesis parameterization through SABLE [68] tags.

Finally, Cassell, Kopp and colleagues brought together the best from the aforementioned systems in *Northwestern University Multimodal Autonomous Conversational Kiosk (NUMACK)* [69][70][71], a system capable of synthesizing in real-time co-verbal context-sensitive iconic gestures without relying on an underlying library of predefined gestures. Building on the aforementioned MACK system, it aims at providing appropriate multimodal directions to certain locations. Generating iconics on-the-fly relies on two assumptions: (a) contrasting to words which are arbitrarily linked to concepts, iconics depict visual information about the object or action being referred to; (b) empirical studies provide evidence for patterns in how features of shape, spatial concepts and relationships are conveyed in gesture. Thus, they propose the introduction of a new semantical level in the gesture production process where *image description features*, referring to the aforementioned shape and spatial concepts, are mapped into morphological gesture form features. Regarding the natural language and gesture generation process, extending the work in Rea, a *gesture planner* is integrated with *SPUD* at the microplanning stage. Microplanning is where communicative intent is distributed and recoded into linguistic and gesture form. In concrete, the gesture planner converts communicative intent into appropriate image description features which are, then, iteratively mapped to specific form features which fill a *gesture features structure* that represents the whole gesture. This structure is integrated with SPUD's language resources and the algorithm proceeds, then, as in Rea. Surface realization, where the output from microplanning is converted into synthesized speech and gesture animations, relies on BEAT to add further nonverbal behavior and Kopp's system to schedule and animate both verbal and non-verbal behavior.

2.2.3 Summary

While computer graphics focuses on realism, computational psycholinguistics focuses on believability. Though the aim of this work is more in line with the latter, both contribute with techniques which support effective implementation of the psycholinguistics requisites imposed on a gesticulation animation model.

The kind of anatomical physically-based hand models explored in computer graphics, contribute with techniques to control and structure the hand model. Regarding structure, the

prevalent technique is to use a hierarchical link structure with appropriate joint limits. Regarding control of static features, hand position and orientation use robotics-based techniques and hand shapes are commonly achieved through predefined libraries. Regarding control of dynamic features, such as motion profiles, physically-based interpolation techniques are used to generate specific velocity and acceleration profiles.

Relating to efforts in automatic gesture recognition, this work supports automatic reproduction from a gesture transcription algorithm. However, reproduction from transcription is more flexible than through recognition. Recognition systems accurately capture form but, lag with respect to meaning. Thus, reproduction is limited to captured form. In contrast, transcription algorithms rely on knowledge from (human) analysts to interpret meaning and, thus, reproduction, as long as it conforms to the meaning, need not have the same form.

Computational psycholinguistics systems usually build on top of computer graphics models adding techniques for gesticulation generation from communicative intent, higher-level execution control and multimodal synchronization. This work does not explore the gesture and speech production process however, shares several aspects with the underlying animation models seen in the aforementioned systems, namely: their requisites are strictly based on psycholinguistics research and similar static and dynamic features are explored.

2.3 Expression of Emotion through Gesture

Several researchers have explored *motion modifiers* which add emotive qualities to existent motion data. Signal-processing techniques [72][73][74] were used to extract information from motion data which is used to generate emotional variations of neutral motion. Rose and colleagues [75] generate new motion with a certain mood or emotion from motion data interpolation based on radial functions and low order polynomials. Chi and colleagues [76] propose a system which adds expressiveness to existent motion data based on the effort and shape parameters of a dance movement observation technique called Laban Movement Analysis. Finally, Hartmann [77] draws from psychology six parameters for gesture modification: overall activation, spatial extent, temporal extent, fluidity, power and repetition.

However, in digital worlds, motion modifiers need not be limited to the body. In this sense, this work proposes a new kind of motion modifier which goes beyond the body to express emotions through three channels of the surrounding environment – camera, lights and music. This idea draws from the arts where it is common to explore the affordances of the environment to reflect the characters' emotional state. For instance, in theatre, which is one of the most complete forms of expression, dramatic expression, text, sceneries, illumination, make-up, sound and music work together to tell the story [78].

2.4 Multimodal Expression Control

The problem of controlling and integrating gesticulation expression with other modalities is usually solved through markup languages [79]. The idea is that the gesticulation production process communicates the gesticulation plan, created from the communicative intent, to the gesticulation execution process, which animates it, through this language. The language, thus, supports a convenient clear-cut separation between these processes.

However, at the time of this writing, no standard markup language exists for multimodal expression control. The community has acknowledged this need and has begun to address it. A promising effort is the *SAIBA* framework [80] which is bringing together several research groups from the community. Unfortunately, this standard is still in its infancy and, therefore, this work proposes, for the time being, yet another control language.

In this context, this work proposes the *Expression Markup Language (EML)* control language. This language is particularly influenced by: VHML [81], SMIL [82] and MURML [83]. Regarding *Virtual Human Markup Language (VHML)*, this work reuses the notion of organizing control according to modality specific modules. Regarding *Synchronized Multimedia Integration Language (SMIL)*, which is oriented towards audiovisual interactive presentations, this work uses a similar modality synchronization mechanism. Regarding *Multimodal Utterance Representation Markup Language (MURML)*, this work defines a similar notation for gesture specification and synchronization with co-verbal speech. Finally, in contrast to high-level languages such as GESTYLE [84] which tries to capture the individual's expression style and APML [85] which represents, among others, communicative intent, emotions, interaction and cultural aspects, the proposed language focuses on low-level body control such as gesticulation animation as sequences of constraints on static and dynamic features and the generation of speech in a text-to-speech system.

3 The Model

This chapter describes the gesticulation expression model focusing on the following:

- The virtual human three-layer architecture;
- Deterministic expression, i.e., keyframe animation and combination mechanisms;
- Non-deterministic expression, i.e., robotics-based procedural animation;
- Vocal expression, including speech synthesis and parameterization;
- Gesticulation expression, including animation described as sequences of constraints on static (hand shape, position and orientation) and dynamic (motion profiles) features, synchronization with speech, automatic reproduction of GestuRA transcriptions and expression of emotions through the environment;
- An abstract integrated synchronized language for multimodal expression control.

This work proposes a gesticulation expression model which supports:

- Real-time gesticulation animation described as sequences of constraints on static (Portuguese Sign Language hand shapes, orientations and positions) and dynamic (motion profiles) features;
- Multimodal synchronization between gesticulation and speech;
- Automatic reproduction of annotated gesticulation according to GestuRA, a gesture transcription algorithm;
- Emotion expression through three environment channels – camera, illumination and music;
- Expression control through an abstract integrated synchronized language – Expression Markup Language (EML*)*.

The gesticulation expression model builds on top of a virtual human architecture and several other expression modalities, Fig. 3.1. First, a general representation for virtual humans is based on the *three-layer architecture*, described in section 3.1, which defines at its core a hierarchical skeleton. Second, in order to control the virtual human, two expression modalities are proposed: *deterministic expression*, described in section 3.2, which defines basic keyframe animation; *non-deterministic expression*, described in section 3.3, which defines robotics-based procedural animation. Third, gesticulation expression is tightly related to speech. Thus, *vocal expression*, described in section 3.4, which integrates a text-to-speech system, is defined. Fourth, in order to implement the psycholinguistics requisites described in subsection 2.1.5,

gesticulation expression, described in section 3.5, is created. Finally, so as to provide an abstract control interface for a mind, a markup integrated synchronized control language, *Expression Markup Language (EML)*, described in section 3.6, is proposed.

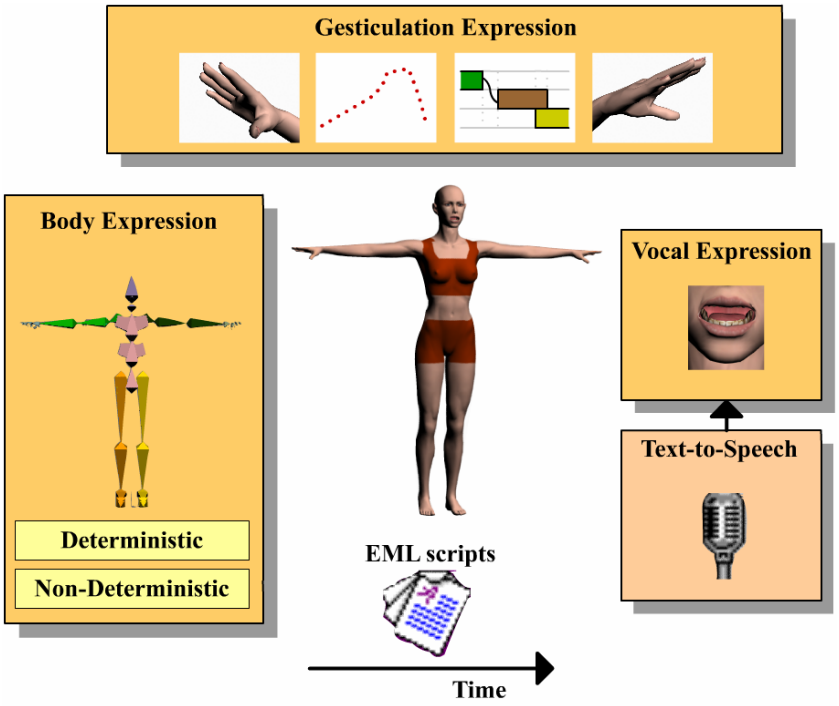


Fig. 3.1 Model overview.

3.1 Virtual Human Architecture

In this work, virtual humans are structured according to the three-layer architecture [86][87] which is organized as follows:

- *Geometry layer*, which defines the underlying skeleton, maintains geometry information (skin, bone's frames, etc.) and renders the virtual human in some graphics API[4];
- *Animation layer*, which builds on top of the geometry layer and is responsible for animation, updating in each frame the underlying skeleton pose;
- *Behavior layer*, which builds on top of the animation layer and is responsible for high-level control. In theory, the virtual human mind should interact only with this layer.

3.1.1 The Skeleton

At the core of the geometry layer lays the skeleton. The skeleton is an *articulated structure* [88], i.e., a *link* hierarchy connected by *joints*. Animating the virtual human corresponds to animating

[4] API stands for *Application Programming Interface*.

the skeleton which will, then, deform the polygonal mesh defining the skin. Besides being computationally efficient, this technique affords animation reutilization. As long as the skeleton is equivalent, different virtual humans may share animations.

So, what is the ideal skeleton for virtual humans? How many bones (or links) should it have? This work proposes a human-based skeleton, Fig. 3.2. It is a logical first approach if we interpret the human skeleton as a locomotion system which benefited from several million years of evolution subjected to the forces of natural selection. In concrete, the proposed skeleton has 54 bones. All skull bones are compacted into a single one and the ribs are omitted altogether. Due to the complexity of their human counterparts, the vertebral column is simplified to 4 bones and only one of the shoulder complex's articulations is modeled. In practice, this model sufficed to simulate intended motion. Appendix A defines the skeleton formally.

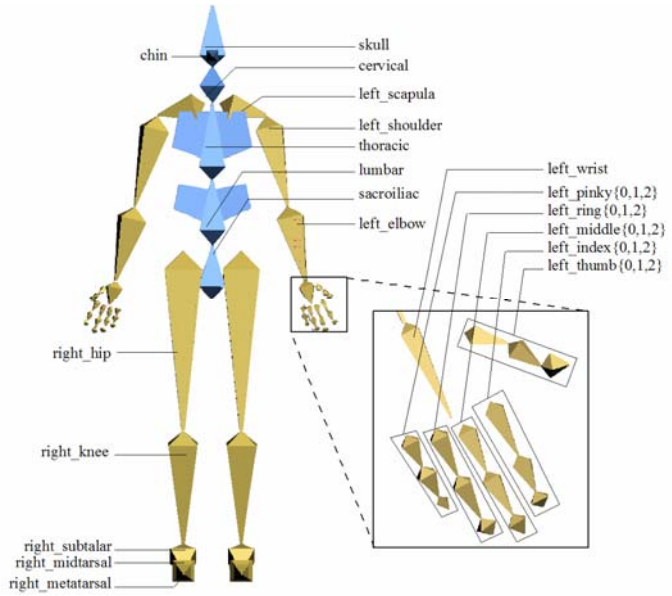


Fig. 3.2 The virtual human skeleton.

Skin deformation is based on the *vertex blending* technique [89]. In this technique, each polygonal mesh vertex is associated with a subset of the skeleton's bones which influence its position. For instance, elbow articulation vertices are influenced by the upper and lower parts of the arm. An advantage of this technique is that it avoids discontinuities at such junctions.

3.1.2 Idle Motion

To enhance virtual human believability, primitives for the generation of idle motion were added to the behavior layer. Presently, blinking and Perlin noise [90] are supported.

3.1.3 Expression Modalities Integration

Each expression modality is conceptually related to a subset of the layers. Deterministic expression relies on an underlying skeleton to implement keyframe animation and thus, relates to the geometry and animation layers. Non-deterministic expression extends the skeleton with robotic manipulators and control is defined in the animation layer. Vocal expression, which integrates with a text-to-speech system, is implemented in the behavior layer. As gesticulation expression relies on all previous modalities, it is also implemented in the behavior layer. Finally, serving as an interface for a mind, the multimodal expression markup language parser is placed at the top behavior layer.

3.2 Deterministic Body Expression

Deterministic body expression is about animation defined as predefined sequences of virtual human poses – *keyframes* – usually obtained from human artists. The model generates in-between poses, supports several animation combination mechanisms but, ultimately, is not independent from the human creator and, thus, is not very flexible. Still, deterministic expression is useful for gesticulation expression to: (a) animate complex gesticulation which is too hard to model through features, (b) animate gesticulation which involves other body parts than arms and hands; (c) support the hand shape static feature.

Deterministic expression revolves around animation players. An *animation player* is responsible for animating a subset of the skeleton's bones according to a specific *animation mechanism*. Several animation mechanisms are supported: weighted combined animation, body group animation and pose animation. These will be described in the following subsections. Furthermore, several animation players can be active at the same time. As animation players may compete for the same bones, an arbitration mechanism based on priorities is supported.

3.2.1 Weighted Combined Animation

Weighted combined animation averages several simpler animations. Consider the hand salutation gesture. This gesture can be performed in several contexts: while walking, sitting or driving. Consider how to model such gesture. A first solution would be to create three different animations. However, what if it were necessary to model the hat salutation gesture? Three more animations would be required. Naturally, it is possible to do better. Effectively, the hand salutation gesture involves a different portion of the body than does walking, seating or driving. Thus, different animations involving different body portions may be created and, afterwards, combined. This is the intuition behind weighted combined animation. Here, animations are

placed on *layers*. Each layer defines a *weight* parameter. The total weight sum must equal the unit. Final animation corresponds, thus, to the weighted average among all layers' animations.

3.2.2 Body Group Animation

Body group animation corresponds to combination of several simpler animations assigned to disjoint bone sets. Consider again the hand salutation gesture. Weighted combined animation models it through two different animations: one for the salutation itself; the other for the context. However, even though each animation aims to affect only a portion of the body, in effect, it keeps the rest on the neutral position. Thus, final animation does not have intended motion amplitude. The salutation gesture animation is subtler than intended. Body group animation solves this problem. Here, layers are associated with disjoint subsets of the skeleton's bones – the *body groups*, Fig. 3.3. Animations in each layer affect only the respective bones.

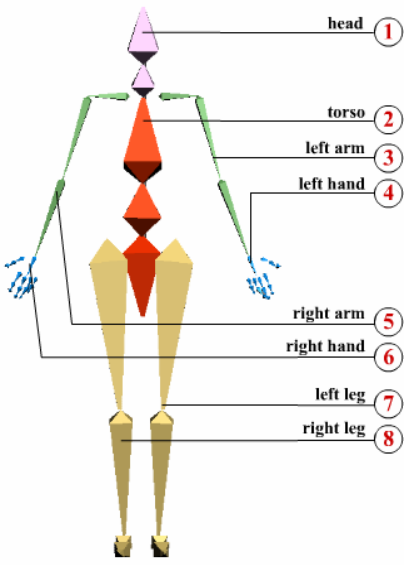


Fig. 3.3 Virtual human default body groups.

3.2.3 Pose Animation

Pose animation applies static stances to bones. Consider, yet again, the hand salutation gesture. Even though the wrist is moving, the hand shape remains the same. Thus, this animation could be modeled as a combination of one animation for the wrist, using one of the previous mechanisms, and a stance for the hand, using pose animation. Pose animation also supports combination of two stances and a parameter which controls interpolation between them.

3.2.4 Applications

To develop, evaluate and debug deterministic expression two applications were built: the *virtual human viewer*; and the *step lesson*. All applications were developed using the *C#* programming language and *Managed DirectX 9* graphics API.

The virtual human viewer (Fig. 3.4) is an interactive application which supports: (a) loading of virtual human models; (b) deterministic animation loading and combination according to weighted combined, body group and pose animation mechanisms.

In the *step lesson* application, a virtual human performs a cardio-fitness step lesson. The lesson takes about four minutes and is animated resorting to deterministic expression EML scripts. In concrete, 28 animations were defined and combined through weighted combined and body group animation.

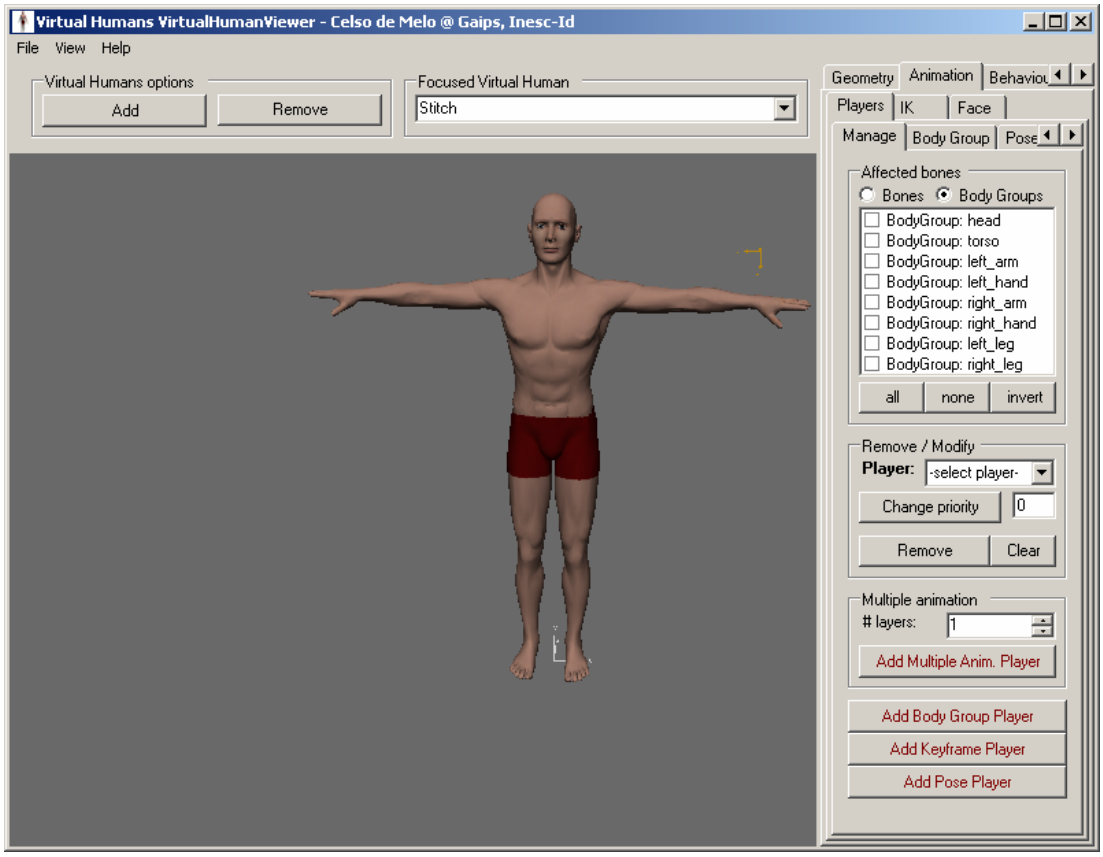


Fig. 3.4 The 'virtual human viewer' application.

3.3 Non-Deterministic Body Expression

Deterministic expression relies on human artists to animate the virtual human however, power comes from removing humans from the loop. Non-deterministic body expression applies robotics to virtual humans thus, laying the foundations for human-free procedural animation. In

concrete, this work applies robotics to the three-layer architecture. In the geometry layer, robotic manipulators are integrated with the skeleton to control limbs and head and, in the animation layer, three inverse kinematics and one inverse velocity primitives are defined.

Non-deterministic expression is at the core of flexible gesticulation expression as it provides the means to position and orient the hands arbitrarily in space. Furthermore, dynamic features, such as motion profiles, are implemented through robotics-based control primitives.

3.3.1 Background

This work focuses on control and mechanics of an important subset of industrial robots – *mechanical manipulators*. A mechanical manipulator is composed of rigid *links* connected by *joints* which support relative motion between adjacent links. In case of rotation, joints are called *revolute*. In case of translation, joints are called *prismatic*. Prismatic joints are not explored in this work. The number of *degrees-of-freedom* a manipulator has usually corresponds to joint number. At the free end of the kinematic chain lies the *target* or *end-effector*. Most manipulator applications are interested in the target's position and orientation at each time instant.

3.3.1.1 Denavit-Hartenberg representation

To support manipulator kinematic analysis, it is common to associate *frames* to rigid links according to the *Denavit-Hartenberg* representation [91]. This representation handles well many degrees-of-freedom and supports systematic kinematic analysis [92]. A frame defines position and orientation and can be described by a homogeneous transformation [93].

3.3.1.2 Kinematic Problems

This work explores forward and inverse kinematics. The *direct kinematics* problem can be stated as follows: given values for all joint variables, determine the target's position and orientation. The *inverse kinematics* problem can be stated as follows: given the target's position and orientation, determine all joint configurations which place the target accordingly. This problem results in 6 non-linear non-trivial equations with n variables – 3 for orientation and 3 for position. In general, these are not easy to solve. First, the problem might not have a solution, second, it might have multiple solutions, third, even if it does have a solution, it might be too difficult to obtain. [93]

Regarding inverse kinematics algorithms, *geometric closed-form solutions* are the most adequate for real-time applications [93]. However, these only apply to special manipulators. In concrete, if the manipulator has 6 revolute joints and the last 3 intersect at a point, then it is possible to apply the *kinematic decoupling* technique to solve the inverse kinematics problem [92]. This technique divides the problem into two sub-problems: the *inverse position problem*,

which calculates the values for the first 3 joints, responsible for the target's position; the *inverse orientation problem*, which calculates the values for the last 3 joints, responsible for the target's orientation. This work explores only manipulators to which this solution applies.

Inverse velocity is also explored in this work. The *inverse velocity* problem relates the target's Cartesian and angular velocity to joint angular velocity. Solutions to these problems are based on the manipulator's *Jacobian*. [92]

3.3.1.3 Application to virtual humans

Robotics application to virtual humans has been the target of much attention. However, direct application to computer graphics is not suitable and specialized solutions are required [94]. Thus, several limb manipulators have been proposed, sharing the following characteristics: (a) both upper and lower limbs are modeled by the same manipulator relating, respectively, shoulder, elbow and wrist to thigh, knee and ankles; (b) they usually have 7 degrees-of-freedom distributed as follows, 3 for the shoulder (thigh), 1 for the elbow (knee) and 3 for the wrist (ankle). This work considers, however, simpler 6 degrees-of-freedom manipulators.

3.3.2 Robotic Manipulators

3.3.2.1 Limb manipulator

The limb manipulator has 6 revolute joints, namely: θ_1, θ_2 and θ_3 responsible for the target's position; θ_4, θ_5 and θ_6 responsible for the target's orientation. Furthermore, joint axis 4, 5 and 6 intersect at a single point. Fig. 3.5 presents the manipulator and Table 3.1 its Denavit-Hartenberg parameters.

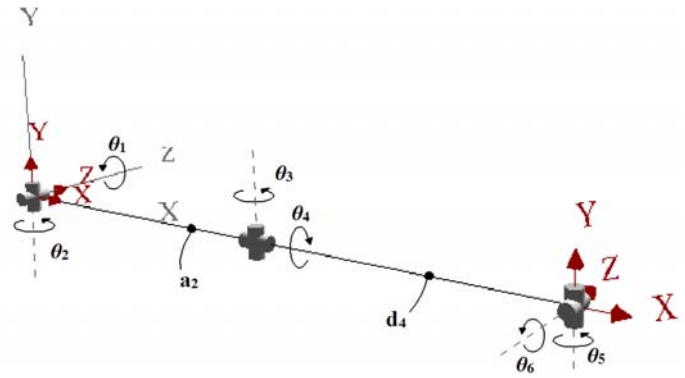


Fig. 3.5 The limb manipulator.

This manipulator structure was chosen for two reasons: (a) arbitrary target's position and orientation within the *workspace* can be set for a 6 degrees-of-freedom manipulator [93]; (b) six revolute joints, with the last three intersecting at a point, allow resolution of the inverse

kinematics problem by kinematic decoupling. Following current approaches (subsection 3.3.1.3), both upper and lower members are modeled with the same manipulator.

Link	α_i	a_i	θ_i	d_i
1	$\pi/2$	0	*	0
2	0	a_2	*	0
3	$\pi/2$	0	*($\pi/2$)	0
4	$-\pi/2$	0	*	d_4
5	$-\pi/2$	0	*($-\pi/2$)	0
6	0	0	*	0

Table 3.1 The limb manipulator's Denavit-Hartenberg parameters. Note that variables θ_3 e θ_5 have initial *offsets*. The offset is added to the joint angle to confer a natural neutral pose.

The inverse kinematics algorithm is shown in Fig. 3.6. In line 1, the *target frame* is transformed into manipulator base coordinates. In lines 2-3, if the target frame is outside the manipulator's workspace, i.e., outside the sphere centered on the manipulator base and with radius a_2+d_4, then it is projected onto the sphere's surface. In line 4, the kinematic decoupling technique is applied to obtain joint configurations which position and orient accordingly the target. Finally, line 5 filters one solution from potentially multiple solutions, as described below.

```
InverseKinematics
input:      target frame
output:     joint configuration if solution exists; null, otherwise

1: target = target * invert(manipulatorBase);
2: if(!withinWorkspace(target))
3:      target = projectTarget(target);
4: solutions = kinematicDecoupling(target);
5: return selectFromMultipleSolutions(solutions);
```

Fig. 3.6 Pseudocode for the limb manipulator inverse kinematics algorithm.

As, in humans, articulations constrain motion, joint limits are simulated. Joint limits are based on anthropometry data in [95] (in [96]). In concrete, the 95th percentile of the population in the study is used. Appendix B describes these limits in detail.

Solution selection from possibly multiple solutions to the inverse kinematics problem is based on *filter* application. In concrete, three filters are defined: (1) *matrix error*, which selects the solution which minimizes the sum of the squares of the difference between the elements of the solution and intended target frame matrices; (2) *joint value variation*, which selects the solution which minimizes the difference between current joint configuration and solution joint configuration; (3) *user preference*, which selects a solution based on user-defined preferences.

To integrate the limb manipulator with the skeleton, manipulator bases where associated to appropriate bones and joint values were mapped to bone transformations. Having defined this mapping, it is possible to control the skeleton through kinematic algorithms.

3.3.2.2 Head manipulator

The head manipulator controls neck and skull. This manipulator has 5 revolute joints, namely: θ_1 and θ_2 responsible for the target's position; θ_3, θ_4 and θ_5 responsible for the target's orientation. Kinematic algorithms for this manipulator are similar to the ones for the limbs.

3.3.3 Control Primitives

Several non-deterministic animation primitives were added to the animation layer. *Non-deterministic animation* corresponds to sequences of skeleton poses dynamically generated in runtime by computational algorithms. In concrete, four primitives are defined: (1) *joint interpolation*, which animates the target through interpolation in the joint space; (2) *function based interpolation*, which animates the target according to a transformation defined, at each instant, by a mathematical function; (3) *frame interpolation*, which animates the target according to interpolation between the current target frame and the intended frame; (4) *Jacobian based animation*, which resorts to inverse velocity algorithms to animate the target according to intended Cartesian and angular velocities.

3.3.4 Applications

To develop, evaluate and debug non-deterministic expression, first, the virtual human viewer application (subsection 3.2.4) was extended to support non-deterministic animation control according to the four robotics-based primitives; second, the *mathematical dance* application was developed where a virtual human dances to the sound of a Portuguese song. The dance takes about four minutes and is fully animated resorting to non-deterministic expression EML scripts.

3.4 Vocal Expression

As synchronization with speech is at the core of gesticulation expression, this work integrates a voice synthesis system. Furthermore, as gesticulation phases synchronize with the speech intonation contour, a high-level voice characterization language – SABLE – is integrated.

3.4.1 Background

3.4.1.1 Text-to-speech synthesis

This subsection is based on an overview of the speech synthesis process by Dutoit [97].

A *text-to-speech synthesizer* is a computational system capable of reading out loud any text. As in humans, it is composed of a *natural language processing* module, which produces a text phonetic transcription with prosody, and by the *digital signal processing* module, which transforms symbolic information it receives into speech. Fig. 3.7 summarizes such a system.

The natural language processing module is divided into three blocks: (1) *text analyzer*; (2) *automatic phonetization*; (3) and, *prosody generator*. Text passes successively through each block before reaching the signal processing module. The text analyzer implements the following procedure: (a) *pre-processing*, which organizes phrases into word lists; (b) *morphological analysis*, which associates words with possible linguistic categories; (c) *contextual analysis*, which, on the basis of context, reduces the list of linguistic categories; (d) *syntactic-prosodic parsing*, which defines the text structure. Next, the phonetization block creates a phonetic transcription from text. This block is based on either a dictionary or a set of rules. Finally, the prosody generator defines precisely each phoneme duration, including silences, and intonation.

The signal processing module is the computational analogue to articulatory muscles control and vocal folds vibratory frequency in humans. It is known that phonetic transitions are more important than stable states for understanding speech. This led to two approaches for this module: (1) *rule-based synthesizers*, which produce sound based on explicit generative rules; (2) *concatenative synthesizers*, which produce sound from a database with phonetic transitions and co-articulations samples.

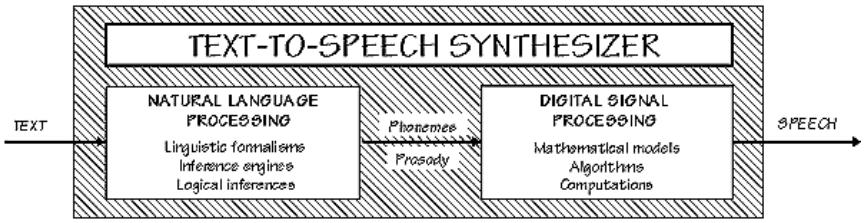


Fig. 3.7 Text-to-speech synthesizer high-level view. [97]

3.4.1.2 SABLE

SABLE [68] is a standard for marking input text to a text-to-speech system. SABLE follows these objectives: enable markup of speech synthesis text input; be system-independent; be easy to use and learn; and be extensible. SABLE supports elements for text emphasis, prosodic breaks definition, velocity, pitch and text volume configuration, among others.

3.4.1.3 Festival speech synthesis system

This work uses the *Festival Speech Synthesis System* [98], developed at the University of Edinburgh. This is a free speech synthesis platform which follows a pipeline as described in subsection 3.4.1.1. Festival supports several languages including English (and dialects) and,

with less quality, Spanish. Furthermore, a Portuguese voice, developed at the research group L^2F at Inesc-ID, is used. However, currently, this voice doesn't integrate with SABLE.

Festival features include: (a) simple *Scheme* programming interface; (b) server/client interaction through sockets thus, supporting clients in other programming languages; (c) access to synthesized utterance structure (words, phonemes, times, etc.) including the ability to save this data in files; (d) incremental real-time synthesis; (e) limited support to SABLE.

3.4.2 Integration of the Text-to-Speech System

Vocal expression integrates the Festival text-to-speech system. This integration involves four aspects: (1) the notion of speech; (2) Festival's voice synthesis pipeline extension; (3) a communication protocol between Festival and virtual humans; (4) a new behavior layer API for voice control. Fig. 3.8 summarizes this integration.

A *speech* is simply a set of files representing the synthesized text. The files have the following information: (a) *utterance structure* including information about phonemes, words and times; (b) *utterance waveforms*; (c) a *configuration* file with information about all speech files. Implementation resorts to Festival's utterance structure manipulation primitives.

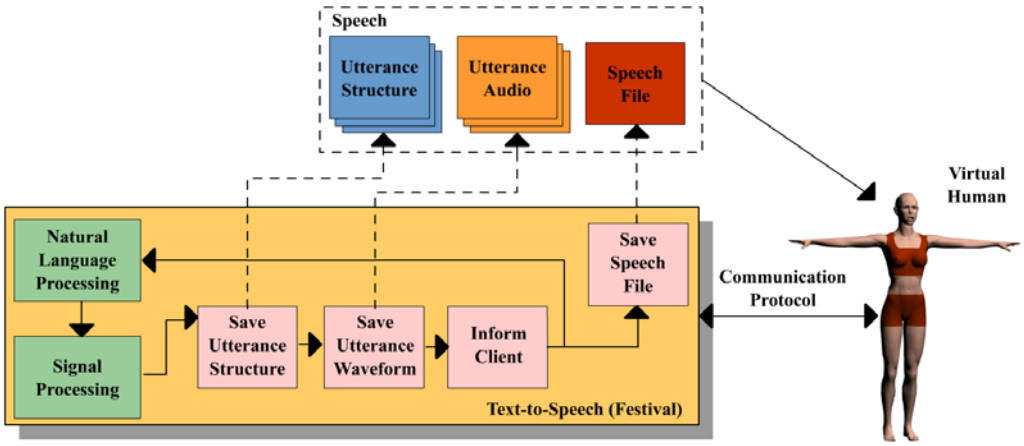


Fig. 3.8 Integration of Festival with virtual humans.

Festival's voice synthesis pipeline extension consists of adding four steps after natural language and signal processing: (a) *save utterance structure*, which saves utterance structure in a file after synthesis; (b) *save utterance waveform*, which saves the utterance waveform in a file after synthesis; (c) *inform client*, which communicates that an utterance is ready to be played after synthesis; (d) *save speech file*, which saves the speech file and communicates to the virtual human about speech completion after all utterances have bee synthesized. Implementation relies on Festival's Scheme interface.

The integration builds on Festival's ability to function according to the server/client model. Thus, a *communication protocol* was developed which is characterized as follows: (a) supports

voice synthesis primitives; (b) supports utterance conclusion communication throughout synthesis; (c) supports communication of speech synthesis conclusion.

At the virtual human side, the behavior layer was extended to support three kinds of voice primitives: (a) *synchronous text-to-speech*, which initiates voice synthesis with real time feedback as utterances are synthesized; (b) *preprocess text*, which synthesizes voice and saves the speech into a persistent format for posterior playback; (c) *asynchronous text-to-speech*, which initiates voice synthesis disregarding feedback from the server causing, thus, synthesis to occur solely on the server side.

3.4.3 Applications

To develop, evaluate and debug vocal expression, first, the virtual human viewer application (subsection 3.2.4) was extended to support real-time speech synthesis and SABLE voice characterization; second, the *virtual poets* application was developed where virtual humans recite poetry. Two versions of the English poem *The Road Not Taken* by Robert Frost are recited: one with SABLE-based voice characterization; the other without voice characterization. The first version sounds more natural than the second. Finally, an excerpt of the Portuguese *Lusíadas*, by Luís de Camões, is recited without SABLE characterization.

3.5 Gesticulation Expression

The gesticulation expression model controls arms and hands and is built on top of the aforementioned expression modalities. In concrete, limb manipulators control the arms, hands' position and orientation while pose animation players control the hands' shape. The model is feature-based, i.e., gesticulation form is modeled as a sequence in time of constraints on static and dynamic features. Features are described on subsection 3.5.1. The model supports multimodal synchronization, in particular, between speech and gesture. Synchronization is described on subsection 3.5.2. The model supports automatic reproduction of annotated gesticulation according to GestuRA, a gesture transcription algorithm. GestuRA and its integration with the model are described on subsection 3.5.3. Finally, the model supports expression of emotion through three environment channels – camera, illumination and music. Environment expression is described on subsection 3.5.4.

3.5.1 Features

Gesticulation is modeled as a sequence in time of constraints on static and dynamic features. Static features are represented in *gesticulation keyframes* and include: hand shape, position, orientation palm axis, orientation angle, and handedness. Dynamic features define keyframe interpolation motion profiles.

Regarding static features, the *hand shape* feature can assume any Portuguese Sign Language hand shape ([99] or see C.Fig. 4 in Appendix C). Furthermore, any two shapes can be combined and a parameter is provided to define how much each contributes. Implementation relies on pose player ability to combine stances (subsection 3.2.3) and on a library of stances for Portuguese Sign Language shapes. The *position* feature is defined in Cartesian coordinates in three-dimensional space. Both world and speaker references can be used. Hand shape orientation is defined by two features: *orientation palm axis*, which defines the palm's normal; and *orientation angle* which defines a left handed angle about the normal. Implementation relies on inverse kinematics primitives (subsection 3.3.3). The *handedness* feature defines whether the gesticulation keyframe applies to the left, right or both hands. In the last case, remaining features apply to the speaker's dominant hand and *symmetrical* values apply to the non-dominant hand. Symmetry is intuitively understood as the gesticulation which would result if a mirror stood on the sagittal plane.

Regarding dynamic features, the model supports several kinds of (keyframe) interpolators, namely: *linear*, which defines linear interpolation; *cosine*, which defines cosine interpolation; and *parametric cubic curves*, which can represent any kind of velocity profile, such as deceleration near the target position and overshooting effects. Currently, the model supports Bézier and Hermite cubic curves, as well as piecewise combinations thereof. Furthermore, interpolators can be structured into hierarchies thus, leading to sophisticated motion profiles. Furthermore, either Cartesian or joint angle velocity can be used. Implementation of interpolation in Cartesian and joint angle space relies, respectively, on the frame interpolation and joint interpolation non-deterministic control primitives (subsection 3.3.3).

3.5.2 Synchronization

Sub-second synchronization of gesture phases with speech relies on a control markup language – Expression Markup Language (EML) – which supports phoneme-level synchronization. The language integrates with SABLE [68] and thus, supports synchronization with speech properties such as intonation contour. Similarly to SMIL [82], modality execution time can be set to absolute or modality relative values. Furthermore, *named timestamps* can be associated with text to be synthesized. The following events can be associated with named timestamps: (a) start of a word; (b) end of a word; (c) start of a phoneme. EML is detailed on section 3.6.

As synchronization between speech and gesture is conveniently described at the gesture phase level, the model supports explicit *gesticulation phase keyframes*. The phase keyframe extends regular keyframes as follows: (a) a *duration* feature is added which defines total phase time; (b) sequences of constraints can now be associated to shape, position and orientation features; (c) constraints within a sequence can be set to start at absolute time offsets relative to

phase start time or at percentages of the total phase duration. However, phase keyframes do not add expressiveness to the model in the sense that gesticulation described with phase keyframes could be converted into an equivalent sequence of regular keyframes.

3.5.3 Automatic Reproduction of Gesticulation Annotations

The gesticulation model supports automatic reproduction of *Gesture Recording Algorithm (GestuRA)* annotations. GestuRA, based on [2] and [100], is a linguistically motivated iterative algorithm for gesticulation form and meaning transcription. It is structured in seven passes. First, speech is transcribed from the video-speech record. Second, text is organized into utterances. Third, utterances are classified according to discourse levels – narrative, metanarrative and paranarrative [1]. Fourth, gesticulation is filtered ignoring remaining gestures. Fifth, gesticulation phases are annotated. Sixth, gesticulation form is formally annotated. Finally, seventh, gesticulation is classified according to its dimensions and its meaning analyzed. GestuRA is fully described in Appendix C.

GestuRA integration with the gesticulation model is achieved through *Anvil* [101], a generic multimodal annotation tool. In concrete, implementing GestuRA in Anvil benefits from its capability of exporting annotations to a XML format. This format can, then, be converted into EML (section 3.6) for immediate execution in virtual humans - Fig. 3.9.

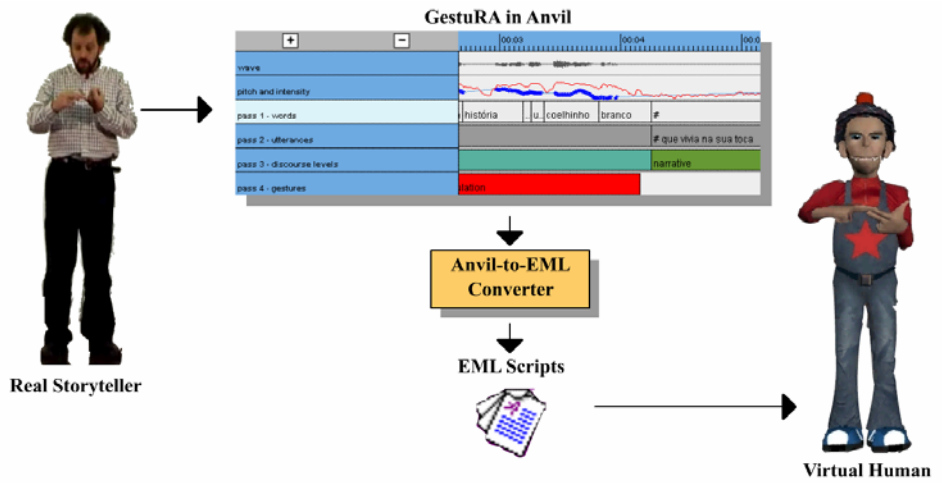


Fig. 3.9 Gesticulation model integration with GestuRA.

Automatic reproduction from GestuRA is valuable for various reasons. First, it is useful to test transcription accuracy. Second, automatic reproduction of transcribed annotations constitutes an important evaluation tool for the gesticulation expression model. As speech and gesture production from communicative intent is not simulated, an alternative to evaluating the

model is to compare it to real life situations. Third, as discussed in subsection 2.2.3, automatic reproduction from transcribed annotations is more flexible than from recognition algorithms.

3.5.4 Expression of Emotions through the Environment

The gesticulation model supports expression of emotions through three channels of the environment – camera, lights and music. The idea is to confer emotion qualities to gesticulation without altering its semantic meaning. Presently, this work synthesizes emotion through the Ortony, Clore and Collins (OCC) emotion theory [102]. However, any emotion theory should apply to the idea of environment expression. Furthermore, though this work focuses only on gesticulation expression, the applications of environment expression are broader [103].

3.5.4.1 The camera channel

A *shot* is a camera configuration, with a certain duration, which is not broken up by cuts [104]. Shots can vary, among others, with *distance* and *angle*. Regarding distance, the closer to the camera, the higher the audience's emotional attachment to the character [104][105]. Five distance shots are usually defined: (1) *extreme close up*, focuses a particular detail, like the character's eyes; (2) *close up*, focuses the face; (3) *medium shot*, focuses from the waist up; (4) *full shot*, focuses the whole character; (5) *long shot*, focuses the character and surrounding environment. Regarding angle, Hornung [105] mentions three representative shots: (1) *eye angle* – the camera is placed at the point of interest's height, representing a neutral view; (2) *high angle* – the camera films the character from above thus, creating the impression of isolation and smallness; (3) *low angle* – the camera films the character from below thus, creating the impression of a powerful character.

The camera channel expresses the character's strongest emotion as follows:

(1) *If it is anger or pride*, a low angle shot is chosen;
(2) *If it is fear*, a high angle shot is chosen;
(3) *If its potential is on the interval [0; 1.5[*, the full shot is chosen;
(4) *If its potential is in the interval [1.5; 2.5[*, the medium shot is chosen;
(5) *If its potential is in the interval [2.5; 4.5[*, a close up is chosen;
(6) *Otherwise*, an extreme close-up is chosen.

3.5.4.2 The illumination channel

The *three-point-lighting* technique is regularly used in movies to illuminate the characters [106]. This technique corresponds to a configuration composed of the following lights: (1) *key light*, which corresponds to the character's main illumination source; (2) *secondary light*, which is low intensity light illuminating the rest of the scenery which would, otherwise, be in the dark; (3)

backlight, which is used to distinguish the character from the background. Regarding illumination color, there is a large body of evidence linking color to emotion (see [107] and references). For instance, red is normally associated to something exciting or aggressive; yellow to something cheerful; green to nature and, therefore, something relaxing; blue to calmness; green-yellow to vomit and thus, to something unpleasant; gray is neutral; among others. Regarding brightness, it is known that well illuminated scenes suggest happiness, while dark scenes suggest mystery or sadness [106].

The illumination channel expresses emotions through lights using a variation of the *three-point-lighting* technique. The key light is placed in-between the character and the camera and emotion expression is achieved through appropriate parameterization. Color is chosen according to strongest emotion as shown in Table 3.2. Finally, brightness varies according to the strongest emotion intensity. Variation is implemented through the attenuation parameter according to equation (3.1) if the emotion is positive and to equation (3.2) if the emotion is negative.

$$\text{Attenuation}_{\text{positive}} = \min(0.5,\ 1 - \text{emotionIntensity} / \text{maxEmotionIntensity}) \quad (3.1)$$

$$\text{Attenuation}_{\text{negative}} = \max(0.25,\ \text{emotionIntensity} / \text{maxEmotionIntensity}) \quad (3.2)$$

Emotion type	Color (RGB)
anger, reproach	red (255, 0, 0)
disappointment, fears-confirmed	grey (200, 200, 200)
disliking	green-yellow (220, 255, 0)
distress	dark grey (153, 153, 153)
fear, relief, neutral	white (255, 255, 255)
hope, liking, satisfaction	bright yellow (255, 255, 200)
joy	yellow (255, 255, 0)

Table 3.2 Emotion type to key light color mapping.

3.5.4.3 The music channel

The relationship between music and emotion can be explored on four dimensions [108]: (1) *structural features* – which relates the music's structure with emotions; (2) *performance features* – which refer to the influence of the artist's interpretation of the music; (3) *listener features* – which refer to the influence of the listener's attitudes and cultural influences; (4) *contextual features* – which refer to aspects of the performance and/or listening situation. Regarding structural features, tempo is one of the most influencing factors affecting emotional

expression in music. Fast tempo may be associated with happy/exciting emotions and slow tempo with sad/calmness emotions. There are many others parameters which lie beyond the scope of this work. Regarding performance features, [108] says that the expressive intention of the performer is converted into various cues during the performance.

The music channel expresses the character's *mood valence* – positive, neutral and negative. To convey mood valence, music, with the same valence, is randomly selected from a library. To fill in the library, music was selected according to the following simple criteria: (1) positive songs have fast tempo and, if they have lyrics it should be positively valenced; (2) neutral songs have medium tempo; (3) sad songs have slow tempo and, if they have lyrics, it should be negatively valenced. Regarding the association of lyrics emotional valence to the music's valence, if the performer tries to convey the music's mood through cues, then it is reasonable to expect that the lyric's mood propagates to the performance's structural features.

3.5.5 Applications

To develop, evaluate and debug gesticulation expression, the virtual human viewer application (subsection 3.2.4) was extended to support incremental animation of feature-based gesticulation and several specialized applications were developed as described next.

Two applications focus on particular aspects of the gesticulation expression model. The *gesticulation viewer* application focuses on feature-based animation of gesticulation, supporting incremental definition of static and dynamic constraints. The *digital expression viewer* application focuses on environment expression, supporting thorough exploration of each expression channel separately.

The *Papous, the virtual storyteller* (Fig. 3.10) application tests gesticulation expression in a storytelling context. Here, a virtual storyteller narrates the traditional Portuguese story "The White Rabbit". The storyteller's voice consisted of synthesized speech audio records. Facial expression consisted of proper lip-synch and emotion expression [109]. Body expression was based on a GestuRA transcription of the human storyteller video, lasting 7 minutes and 30 seconds. In total, 286 gestures were transcribed of which 95% were automatically reproduced through feature-based gesticulation expression and 5% through keyframe animation. This application was used in the first two studies described in chapter 4.

The *dancing solids* application (Fig. 3.11), which is a cartoon-like storytelling application, tests emotion expression through the environment. Stories are not predefined as the outcome varies according to the characters' personalities. The underlying story is a simple one: "Once upon a time, there were a bunch of geometric dancing solids. There were pyramids, cylinder and ellipsoids. There were boys and girls. The girls allured the boys. If the boy liked the girl, he'd court her with a dance. If they both liked each other they simply married. The end". The

rationale behind the application is: (a) *the plot should have lots of emotion eliciting situations*, thus providing opportunities for characters to synthesize emotions and for environment expression to express emotions; (b) *the characters' bodies should have limited expression capabilities* so as to force the audience to rely on environment expression for interpretation. Thus, bodies consist of simple geometric solids with eyes. This application was used in the third study described in chapter 4.

Fig. 3.10 The 'Papous, the virtual storyteller' application.

Fig. 3.11 The "dancing solids" application.

3.6 Multimodal Expression Control

Thus far, a virtual human is endowed with an intelligent body capable of deterministic, non-deterministic, vocal and gesticulation expression but, is it ready to be controlled by a mind? To

answer this question it is necessary to reflect on the body-mind *interface*. This is a collection of independent local interfaces which control each modality. This is not enough. What is necessary is an *integrated*, *synchronized* and *abstract* interface. Integrated because control must rely on the same kind of symbols for each modality. Synchronized because it must support precise multimodal synchronization. Finally, abstract because the interface must be resilient to change.

Thus, to control multimodal expression, this work proposes an integrated, synchronized and abstract language - the *Expression Markup Language (EML)*. The language can be used in two ways: (1) as an *interface for a mind* which needs to express synchronously, in real-time and multimodaly through the body; (2) as a *script* which describes a story, written by a human or digital author, in real-time or not, where the virtual human expresses multimodaly. In the first case, the mind communicates to the body in real-time, through a socket or API, a set of EML clauses which are immediately executed. In the second case, the script defines a sequence of clauses, temporally ordered, which defines a story which can be played later by different virtual humans. Fig. 3.12 summarizes EML integration with virtual humans.

Regarding specification, EML is a markup language which is structured into modules. The *core* module defines the main elements. The *time and synchronization* module defines synchronization mechanisms between modalities and is characterized as follows: (a) supports execution time definition relative to other clauses; (b) supports execution time definition relative to a word or phoneme time in a vocal expression clause; (c) supports loops; (d) supports parallel and sequential execution. This module is based on W3C's SMIL 2.0 specification [82]. The *body* module controls both deterministic and non-deterministic expression. The *voice* module controls vocal expression. Finally, the *gesticulation* module controls gesticulation expression. Appendix D presents the full language specification.

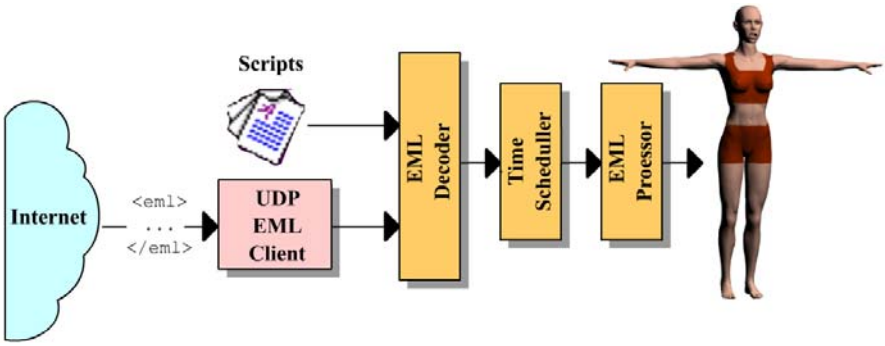


Fig. 3.12 EML integration with virtual humans.

3.6.1 The "Hello World" Example

Let us suppose the virtual human wishes to express "Hello world!". This distributes across modalities as follows: (1) vocal expression synthesizes "Hello world!" emphasizing "world";

(2) gesticulation expression performs a beat gesture, synchronizing with “world”, superimposed on a hand salutation conventionalized gesture; (e) environment expression reflects the emotion of joy. One possible EML codification is as follows:

```
1: <eml name='helloWorld'>
2:  <body>
3:   <gesture-is-on time='0' isOn='true' />
4:    <voice-text time='0' timeId='t1'>
5:     <div>Hello <emph><tm name='t2' event='onStart'/>world!
6:     </emph></div></voice-text>
7:   <gesture-key time='t1-0.1s' duration='0.2' handedness='right'>
8:    <hand-shapes><key time='0' id='7'/></hand-shapes>
9:    <palms><key time='0' a='200' x='0.0' y='0.0' z='-1.0'/></palms>
10:   <motion coords='speaker'>
11:     <key time='0' x='-7.0' y='45.0' z='3.0'/>
12:   </motion>
13:  </gesture-key>
14:  <gesture-key time='t2' duration='0.2' handedness='right'>
15:   <hand-shapes><key time='0' id='7'/></hand-shapes>
16:   <palms><key time='0' a='120' x='0.0' y='0.0' z='-1.0'/></palms>
17:   <motion coords='speaker' >
18:    <key time='0.1' x='-15.0' y='40.0' z='3.0'/>
19:   </motion>
20:  </gesture-key>
21: </body>
22:</eml>
```

Fig. 3.13 EML codification of the “Hello World!” example.

Lines 4-6 define vocal expression. The SABLE element “emph” is used to emphasize “world” and a time marker is associated with the start of the word. Gesticulation expression is divided into two gesticulation keyframes. The first, represented in lines 7-13, raises the right hand above the head with an open-like claw Portuguese Sign Language hand shape. Notice the gesture starts 0.1s before “Hello”. The second, which is represented in lines 14-20, represents the beat-like gesticulation where the right hand abruptly moves to the right while still maintaining overall salutation form. Synchronization occurs with “world”. Finally, regarding environment expression, even though not explicitly represented, the idea is that the joy emotion would be elicited at the start of “Hello” leading to a closer camera shot, an increase in illumination brightness, light’s color change to yellow and, finally, fade in of a positively valenced music.

3.6.2 Applications

To develop, evaluate and debug EML, first, the virtual human viewer application (subsection 3.2.4) was extended to support EML scripts playback from file or socket; second, most of the applications described in previous sections – step lesson (subsection 3.2.4), mathematical dance (subsection 3.3.4), virtual poets (subsection 3.4.3) and Papous, the virtual storyteller (subsection 3.5.5) – ultimately resort to EML scripts to drive participant virtual humans.

4 Evaluation

This chapter reports three studies conducted to evaluate the model in storytelling contexts:

- The first study evaluates the contribution of gesticulation expression for story comprehension, emotion expression, credibility and subject satisfaction;
- The second study builds on the previous, clarifying some issues and exploring further aspects of gesticulation expression;
- The third study evaluates emotion expression through the environment.

Three studies were conducted to assess the model's expressiveness. The first two compare the expression of a human storyteller with that of a virtual storyteller while narrating the Portuguese traditional story "The White Rabbit". The third study evaluates this work's approach for the expression of emotion through the environment.

4.1 First Study

The first study was conducted in the scope of the Papous project at Inesc-ID [3] and aimed at comparing a human storyteller with a virtual storyteller with respect to story comprehension, emotion expression, credibility and subject satisfaction for each of gesticulation, facial and vocal expression. This document will focus only on gesticulation expression results. The human storyteller was a non-professional actor which was simply asked to tell the story in an expressive way without imposing any requirements on gesticulation expression. Regarding the virtual storyteller, the voice consisted of modulated synthesized speech audio records. Facial expression, including proper lip-synch and emotion expression, was generated from a muscular model described in [109]. Gesticulation expression was based on a GestuRA transcription of the human storyteller video, lasting 7 minutes and 30 seconds. In total, 286 gestures were transcribed of which 95% were automatically reproduced through feature-based gesticulation expression (subsection 3.5.1) and 5% through keyframe animation (section 3.2).

Regarding structure, first the subject visualized the story video and, then, answered to a questionnaire. Each subject was presented with one of four video versions: (1) CRVR – Human narrator with real voice; (2) CRVS – Human narrator with synthetic voice; (3) CSVR – Virtual narrator with real voice; (4) CSVS – Virtual narrator with synthetic voice. The questionnaire consisted of twelve classification questions where the subject was asked to classify, from 1 (totally disagree) to 7 (totally agree), whether each modality help understand the story, express emotions properly, is believable and is to his liking.

The study was presented to 108 students at the Technical University of Lisbon. Average age was 21 years and 89% of which were males. Most students related to computer science courses. Each video version was presented to 27 students.

Gesticulation expression results are summarized in Table 4.1. As can be seen, in general synthetic gestures are classified lower than real gestures. However, classification differs only in about 0.45 points. Finally, notice that real gesture classification (about 5) was well below 7.

	CRVR	CSVR	CRVS	CSVS
Gestures helped to understand the story	5.19	4.91	5.04	4.82
Gestures expressed the story's emotions	5.15	4.76	5.30	4.82
Gestures were believable	5.07	4.30	5.30	4.61
I liked the gestures	4.89	4.49	5.22	4.82

Table 4.1 Body expression questions average classifications.

From these results it is possible to conclude that synthetic gestures fared well when compared to its real counterpart. Furthermore, in absolute terms, a classification of about 4.6 is reasonably good. However, this study had some limitations. Firstly, subjects were asked to evaluate gestures explicitly when it is known that gesture interpretation is essentially unconscious (see subsection 2.1). Secondly, subject to multiple interpretations, the notion of "believability" is hard to define thus, results related to the question "Gestures were believable" must be interpreted with caution.

4.2 Second Study

To further assess the model's expressiveness and to correct some of the flaws in the previous study, a second study was conducted. In this study, first, subjects are told that the evaluation is about virtual storytelling and "gesticulation expression" is never mentioned throughout. Second, synthetic gestures are indirectly evaluated through story interpretation questions. Third, each subject sees the story alternatively narrated by the human or virtual storyteller thus, allowing for direct storyteller comparison. Finally, as the study focused on gesticulation expression, the real voice was used for both storytellers and three variations of the virtual storyteller are defined: (1) ST – where both feature-based and keyframe gesticulation are expressed; (2) SF – where only feature-based gesticulation is expressed; (3) SN – where no gesticulation is expressed.

The evaluation is structured into three parts. In part 1 – *profile* – the subject profile is assessed. In part 2 – *story interpretation* – the whole story is presented to the subject. To facilitate remembering, the story is divided into 8 segments of 30 seconds each. Segments are

narrated by either the human storyteller or one of the three kinds of virtual storytellers randomly selected at the start. In concrete, the third and sixth segments are narrated by a subject selected storyteller, while the rest is arbitrarily narrated either by the human or virtual storyteller provided that in the end each narrates an equal number of segments. After each segment, multiple choice interpretation questions are posed. In total 32 questions were formulated. Importantly, a subset, named the *highly bodily expressive (HBE)* questions, focused on information specially marked in gestures, i.e., information which was either redundantly or non-redundantly conveyed through complex gestures like iconics or metaphorics. Finally, in part 3 – *story appreciation* – the subject is asked to choose the preferred storyteller and to describe which is the best and worst feature of each storyteller.

The study was presented to 39 subjects, 90% of which were male, with average age of 23 years and mostly having higher education. The study was fully automated in software and average evaluation time was about 20 minutes. Distribution of virtual storyteller kinds across subjects was: 46% for ST; 31% for SF; 23% for SN. Subject recruitment included personal contact mainly at both campuses of Technical University of Lisbon and distribution of the software through the Web.

Regarding story interpretation results, if we define *diff* to be the difference between the percentage of correct answers following the human storyteller and the percentage of correct answers following the virtual storyteller, then *diff* was: for ST, 4.69%; for SF, -0.68%; for SN, -1.62%. However, if we consider only HBE questions, than distribution is as follows: for ST, 4.75%; for SF, 0.00%; for SN, 9.19%. Regarding subject storyteller selection on the third and sixth segments, the human storyteller was selected about 75% of the time (for ST, 75.00%; for SF, 83.30%; for SN, 72.22%). Regarding subject storyteller preference, the human storyteller was preferred about 90% of the time (for ST, 88.89%; for SF, 83.33%; for SN, 100.00%). Finally, some of the worst aspects mentioned for the virtual storyteller were “body expression limited to arms”, “static/rigid”, “artificial” and “low expressivity”. These relate to the best aspects mentioned for the human storyteller, namely “varied postures”, “energetic/enthusiastic”, “natural” and “high expressivity”.

As can be seen by the results, the human storyteller is better than the virtual storyteller. Interpretation with the human storyteller is better, but not that much (*diff* of 4.69% for ST). Furthermore, when given a choice, subjects almost always chose the human storyteller. Analyzing the best and worst aspects selected for each storyteller might give insight into this issue. Surprisingly, if all questions are considered, *diff* actually reduces for SN when compared to ST (-1.63% over 4.69%). The fact that, for the human storyteller, the voice and face were highly expressive and gestures were mostly redundant might help explain this. However, if only

HBE questions are considered, *diff* considerably increases for the SN case (from 4.75% to 9.19%). Furthermore, for the SN case, the human storyteller was preferred 100% of the times. This confirms that gesticulation affects interpretation. Finally, comparing ST with SF, *diff* for all questions reduces for the latter case (from 4.69% to -0.68%). This suggests that the lack of feature-based gesticulation support for the small fraction of highly complex gestures does not impede effective interpretation.

4.3 Third Study

A study was conducted to evaluate the adequacy of this work's approach for the expression of emotions through camera, illumination and music.

The study was based on the dancing solids (see subsection 3.5.5) which is a cartoon-like application were solids dance for each other while expressing emotions. The study was organized into four parts: (1) *subject profile*, where the subject's profile was assessed; (2) *emotion perception*, where the subject was presented with one of seven emotions – anger, disliking, distress, fear, joy, liking, reproach – or neutral emotion expression with varying configurations of two of the expression channels – camera and illumination. The subject was then asked to guess the expressed emotions from a set of options which were provided; (3) *music emotional valence*, where the subject was asked to classify 12 music compositions according to one of the following mood valences: positive/happy; neutral; negative/sad. The study was fully automated and presented to 50 students, average age of 23 years, at the Technical University in Lisbon.

Regarding emotion perception results, data revealed that: perception of distress, joy, liking, neutral was highly accurate (above 75%) even without environment expression; illumination color expression increased accuracy particularly for anger (from 13% to 43%), disliking (from 13% to 20%) and reproach (from 46% to 60%); the camera channel emotion to camera shot mapping, in general, did not influence accuracy. Finally, regarding music emotion valence, average subject classification matched predictions for 92% of the music.

5 Conclusions and Future Work

This work proposes a virtual human gesticulation expression model which supports: (a) real-time gesticulation animation described as sequences of constraints on static (Portuguese Sign Language hand shapes, orientation palm axis, orientation angle and handedness) and dynamic (motion profiles) features; (b) multimodal synchronization between gesticulation and speech; (c) automatic reproduction of GestuRA gesticulation annotations; (d) expression of emotions through three environment channels – camera, illumination and music; (e) expression control through the abstract integrated synchronized Expression Markup Language. The model builds on top of a layered virtual human architecture and several other expression modalities, namely: deterministic expression, which defines keyframe animation and supports animation of complex gesticulation; non-deterministic expression, which defines robotics-based procedural animation and supports arbitrary placement and orientation of the hands as well as motion profiles; vocal expression, which supports voice synthesis and speech-gesticulation synchronization.

Three studies were conducted to evaluate the model. The first two evaluate the model in storytelling contexts and compare the expression of a human with a virtual storyteller. Results indicate that synthetic gestures fare well when compared to real gestures and story interpretation does not differ significantly between storytellers however, the human storyteller was still preferred. The third study focus on expression of emotions through the environment. Results indicate that the camera channel needs further tuning, the illumination channel is effective in influencing emotion interpretation and tempo and lyrics reasonably predict music valence.

Still, further improvements can be made to the model, the underlying virtual human architecture and supporting expression modalities.

Regarding the virtual human architecture, the vertebral column model, which considers only 4 of the 33 human vertebras, and the shoulder complex model, which considers only 1 of the 3 human articulations, can be improved. Regarding deterministic expression, the standard body group configuration can be improved and animation transition constraints can be defined as in Perlin's work [87]. Regarding non-deterministic expression, first, instead of six, seven revolute joint manipulators should be considered for the limbs. Seven joints introduce redundancy, i.e., the same position and orientation can be described by more than one joint configuration. This materializes into better elbow and knee control which leads to more natural animation. However, underlying mathematics is more complicated. Second, dynamics could be explored to generate physically realistic animation. Third, specialized hand manipulators could be explored. This would lead to control primitives which support dynamic generation of

arbitrary hand shapes. Regarding vocal expression, improvements include, among others, high-level emotional parameters, which SABLE doesn't support, and SABLE integration with the Portuguese voice.

Regarding gesticulation expression, first, gesticulation needs to go beyond arms and hands and explore other body parts. Posture shifts, which relate to discourse structure [110], could be explored. Second, some features' implementation restrict expressiveness. Nothing guarantees that Portuguese Sign Language hand shapes and non-spline parametric curves (such as Bézier and Hermite) and combinations thereof suffice to express, respectively, all shapes and motion profiles. Furthermore, lack of elbow control in the upper limb manipulator limits naturalness. Third, preparation and retraction motion, as well as co-articulation effects, could be automatically generated. Fourth, formal annotation of meaning in GestuRA should be attempted, perhaps resorting to some kind of logic, which would lead to a more flexible automatic reproduction algorithm. Fifth, regarding expression of emotions through the environment: in the camera channel, the mapping between emotion types and shots could be refined; in the illumination channel, shadows should be explored as these can be very expressive [106]; in the music channel, music selection should reflect more emotion parameters (e.g., arousal) and music parameters (e.g., mode, rhythm, loudness). Finally, a more anatomically correct hand model with appropriate constraints (subsection 2.2.1) would lead to more realistic gesticulation simulation.

At a more global level, the next step is to tackle the gesticulation production problem. Altogether, the model seems ready to support speech and gesticulation production models (subsection 2.1.4). Regarding de Ruiter's model, the gestuary can mostly be implemented through feature-based and keyframe gesticulation; signal passing synchronization is straightforwardly supported. Krauss' model which is feature-based is also compatible with the model but, cross-modal priming is not supported. The language effect on gesture in Kita and Özyürek's and theme/rheme distinctions on Cassell and Prevost's models occur early in the production process and, ultimately, materialize into specific features which this model supports. McNeill's growth point model lacks details on morphology generation however, if the dialectic ultimately materializes into features and synchronization can be described with respect to a finite number of specific synchronization points, then this model may support it.

References

[1] McNeill, D.; *Hand and Mind: What gestures reveal about thought*; University of Chicago Press; 1992

[2] McNeill, D.; *Gesture and Thought*; University of Chicago Press; 2005

[3] Papous; Papous project at Inesc-ID; Ref.: POSI / SRI / 41071 / 2001

[4] Nonverbal Behavior Nonverbal Communication Links; www3.usal.es/~nonverbal/introduction.htm

[5] Kendon, A.; *How gestures can become like words* in F. Poyatos (ed.), *Cross-cultural perspectives in nonverbal communication*, pp.131-141, Hogrefe; 1988

[6] McNeill, D.; *Language and Gesture*; Cambridge University Press; 2000

[7] Kendon, A.; *Some relationships between body motion and speech* in A. Siegman and B. Pope (eds.), *Studies in dyadic communication*, pp.177-210, Pergamon Press; 1972

[8] Kendon, A.; *Gesticulation and speech: Two aspects of the process of utterance* in M. Key (ed.), *The Relationship of Verbal and Nonverbal Communication*, pp.207-227, Mouton and Co.; 1980

[9] Nobe, S.; *Where do* most *spontaneous representational gestures actually occur with respect to speech?* in D. McNeill (ed.), *Language and Gesture*, pp.186-198, Cambridge University Press; 2000

[10] Kendon, A.; *Sign languages of Aboriginal Australia: Cultural, semiotic and communicative perspectives*; Cambridge University Press; 1988

[11] Kita, S.; *The temporal relationship between gesture and speech: A study of Japanese-English bilingual*; MhD thesis, Department of Psychology, University of Chicago; 1990

[12] Duncan, S.; McNeill, D.; McCullough, K.; *How to transcribe the invisible – and what we see* in D. O'Connell, S. Kowal, R. Posner (eds.), *Zeichen für Zeit: Zur Notation und Transkription von Bewegungsabläufen* (special issue of KODIKAS/CODE) 18:75-94; 1995

[13] de Ruiter, J.; *The production of gesture and speech* in D. McNeill (ed.), *Language and gesture*, pp.284-311, Cambridge University Press; 2000

[14] Krauss, M.; Chen, Y.; Gottesman, R.; *Lexical gestures and lexical access: A process model* in D. McNeill (ed.), *Language and gesture*, pp.261-283, Cambridge University Press; 2000

[15] Kita, S.; Özyürek, A.; *What does cross-linguistic variation in semantic coordination of speech and gesture reveal? Evidence for an interface representation of spatial thinking and speaking* in *Journal of Memory and Language 48*, pp.16-32; 2003

[16] Cassell, J.; Prevost, S.; *Distribution of semantic features across speech and gestures by humans and machines* in Lynne S. Messing (ed.), *Proceedings of WIGLS*, pp.253-269, Applied Science and Engineering Laboratories; 1996

[17] Levelt, W.; *Speaking*; MIT Press; 1989

[18] Schmidt, R.; *A schema theory of discrete motor skill learning* in *Psychological Review* 82:225-260; 1975

[19] Wilkins, D; *What's 'the point'? The significance of gestures of orientation in Arrernte* presented at the *Institute for Aboriginal Developments*, Alice Springs, July; 1995

[20] Kita, S.; *How representational gestures help speaking* in D. McNeill (ed.), *Language and gesture*, pp.162-185, Cambridge University Press; 2000

[21] Prevost, S.; *An information structural approach to monologue generation* in *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz; 1996

[22] McNeill, D.; *Catchments and contexts: non-modular factors in speech and gesture production* in D. McNeill (ed.), *Language and gesture*, pp.312-328, Cambridge University Press; 2000

[23] Thompson, D.; Buford, W.; Myers, L.; Giurintano, D.; Brewer III, J.; *A Hand Biomechanics Workstation* in *Computer Graphics*, vol.22, no.4, pp.335-343; 1988

[24] Albrecht, I.; Haber, Jörg, H.; Siedel, H.; *Construction and Animation of Anatomically Based Human Hand Models* in *SIGGRAPH 2003*, pp.98-109; 2003

[25] Wagner, C.; *The pianist's hand: Anthropometry and biomechanics* in *Ergonomics*, 31(1):97-131; 1988

[26] Magnenat-Thalmann, N.; Laperrière, R.; Thalmann, D.; *Joint-Dependent Local Deformations for Hand Animation and Object Grasping* in *Proceedings of Graphics Interface '88*, pp.26-33; 1988

[27] Kunni, T.; Tsuchida, Y.; Matsuda, H.; Shirahama, M.; Miura, S.; *A model of the hands and arms based on manifold mappings* in *Proceedings CGI'93*, pp.381-393; 1993

[28] Huang, Z.; Boulic, R.; Thalmann, N.; Thalmann, D.; *A Multi-sensor Approach for Grasping and 3D Interaction* in *Proceedings CGI'95*, pp.235-254; 1995

[29] Ip, H.; Chan, C.; *Dynamic Simulation of Human Hand Motion Using an Anatomical Correct Hierarchical Approach* in *Proceedings 1997 IEEE International Conference Systems, Man, and Cybernetics*, Orlando, Florida, pp.1307-1312; 1997

[30] Ip, H.; Chan, S.; Lam, M.; *HACS: Hand Action Coding System for Anatomy-Based Synthesis of Hand Gestures* in *Proceedings International Conference Systems, Man, and Cybernetics 1998*, pp.1307-1312; 1997

[31] Kim, J.; Cordier, F.; Magnenat-Thalmann, N.; *Neural Network-based Violinist's Hand Animation* in *Proceedings Computer Graphics International (CGI 2000)*, pp.37-41; 2000

[32] Mulero, J.; Batlle, J.; Coronado, J.; *Parametric Neurocontroller for Positioning of an Anthropomorphic Finger Based on an Opponent-Driven Tendon Transmission System* in *Proceedings IWANN*, pp.47-54; 2001

[33] Moccozet, L.; Magnenat-Thalmann; *Dirichlet Free-Form Deformations and their Application to Hand Simulation* in *Proceedings Computer Animation'97*, pp.93-102; 1997

[34] Wan, H.; Luo, Y.; Gao, S.; Peng, Q.; *Realistic Virtual Hand Modeling with Applications for Virtual Grasping* in *Proceedings of SIGGRAPH 2004*, pp.81-87; 2004

[35] Sibille, L.; Teschner, M.; Srivastava, S.; Latombe, J.; *Interactive Simulation of the Human Hand* in *CARS'02*, pp.7-12; 2002

[36] Tsang.W.; Singh, K.; Fiume, E.; *Helping Hand: An Anatomically Accurate Inverse Dynamics Solution for Unconstrained Hand Motion* in *Proceedings of SIGGRAPH 2005*, pp. 320-328; 2005

[37] Rohrer, T.; *The Body in Space: Dimensions of Embodiment* in J. Zlatev, T. Ziemke, R. Frank, R. Dirven (eds.), *Body, Language and Mind*, vol.2, Berlin: Mouton de Gruyter; 2005

[38] Rijpkema, H.; Girard, M.; *Computer Animation of Knowledge-Based Human Grasping* in *Computer Graphics*, vol.25, no.4, pp.339-348; 1991

[39] Sanso, R.; Thalmann, D.; *A Hand Control and Automatic Grasping System for Synthetic Actors* in *Computer Graphics Forum*, 13(3):167-177; 1994

[40] Pollard, N.; Zordan, V.; *Physically based grasping control from example* in *Proceedings of SIGGRAPH 2005*, pp.311-318; 2005

[41] Shapiro, A.; Pighin, F.; *Hybrid Control for Interactive Character Animation* in *Pacific Graphics* 2003, pp.455-461; 2003

[42] Gao, Y.; *Automatic extraction of spatial location for gesture generation*; MhD thesis, Electrical Engineering and Computer Science, Cambridge, MA, MIT; 2002

[43] Elkoura, G.; Singh, K.; *Handrix: Animating the Human Hand* in *Proceedings of SIGGRAPH 2003*, pp.110-119; 2004

[44] Jaimes, A.; Sebe, N.; *Multimodal Human Computer Interaction: A Survey* in *IEEE International Workshop on Human Computer Interaction* in conjunction with *ICCV 2005*, Beijing, China; 2005

[45] Oviatt, S.; Cohen, P.; Wu, L.; Vergo, J.; Duncan, L.; Suhm, B.; Bers, J.; Holzman, T.; Winograd, T.; Landay, J.; Larson, J.; Ferro, D.; *Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions* in *Human-Computer Interface*, 15:263-322, 2000.

[46] Pavlovic, V.; Sharma, R.; Huang, T.; *Visual Interpretation of hand gestures for human computer interaction: A review* in *IEEE Trans. Pattern Analysis Machine Intelligence*, vol.19, pp.677-695, July; 1997

[47] Gavrila, D.; *The visual analysis of human movement: A survey* in *Computer Vision and Image Understanding*, vol.73, pp.82-98, Jan.;1999

[48] Wu, Y.; Huang, T.; *Hand Modeling, Analysis, and Recognition* in *IEEE Signal Processing Magazine*, 18(3):51-60; 2001

[49] Wu, Y.; Huang, T.; *Vision-based gesture recognition: A review* in *3rd Gesture Workshop*; 1999

[50] Lin, J.; Wu, Y.; Huang, T.; *Modeling the Constraints of Human Hand Motion* in *Proceedings Workshop on Human Motion*, pp.121-126; 2000

[51] Bicchi, A.; *Hands for Dexterous Manipulation and Robust Grasping: A Difficult Road Towards Simplicity* in *IEEE Transactions on Robotics and Automation* 16, 6:652-662; 2000

[52] Cassell, J.; Pelachaud, C.; Badler, N.; Steedman, M.; Achorn, B.; Becket, T.; Douville, B.; Prevost, S.; Stone, M.; *Animated Conversation: Rule-based Generation of Facial Expression, Gesture & Spoken Intonation for Multiple Conversational Agent* in *Proceedings of SIGGRAPH 1994*, pp. 413-420; 1994

[53] Cassell, J.; *A Framework For Gesture Generation And Interpretation* in R. Cipolla, A. Pentland (eds.), *Computer Vision in Human-Machine Interaction*, pp.191-215, New York: Cambridge University Press; 1998

[54] Cassell, J.; Bickmore, T.; Bilinghurst, M.; Campbell, L.; Chang, K; Vilhjámsson, H.; Yan, H.; *An Architecture for Embodied Conversational Characters* in *Proceedings of the First Workshop on Embodied Conversational Characters*, Tahoe City, California; 1998

[55] Cassell, J.; Bickmore, T.; Billinghurst, M.; Campbell, L.; Chang, K.; Vilhjálmsson, H.; Yan, H.; *Embodiment in Conversational Interfaces: Rea* in *Proceedings of the CHI'99 Conference*, Pittsburgh, PA, pp.520-527; 1999

[56] Cassell, J.; Stone, M.; *Living Hand to Mouth: Psychological Theories about Speech and Gesture in Interactive Dialogue Systems* in *Proceedings of the AAAI 1999 Fall Symposium on Psychological Models of Communication in Collaborative Systems*, pp.34-42, North Falmouth, MA; 1999

[57] Cassell, J.; Vilhjálmsson, H.; Chang, K.; Bickmore, T.; Campbell, L.; Yan, H.; *Requirements for an Architecture for Embodied Conversational Characters* in D.Thalmann, N. Thalmann (eds.), *Computer Animation and Simulation '99*, Eurographics Series, pp.109-120, Viena, Austria: Springer Verlag; 1999

[58] Cassell, J.; Bickmore, T.; Vilhjálmsson, H.; Yan, H.; *More than Just a Pretty Face: Affordances of Embodiment* in *Proceedings of the Conference on Intelligent User Interfaces*, pp.52-59; 2000

[59] Cassell, J.; Stone, M.; *Coordination and Context-Dependence in the Generation of Embodied Conversation* in *Proceedings of the International Natural Language Generation Conference*, pp.171-178. Mitzpe Ramon, Israel; 2000

[60] Stone, M.; Doran, C.; *Paying heed to collocations* in *International Natural Language Generation Workshop*, pp.91-101; 1996

[61] Cassell, J.; Vilhjálmsson, H.; Bickmore, T.; *BEAT: the Behavior Expression Animation Toolkit* in *Proceedings of SIGGRAPH 2001*, pp.477-486; 2001

[62] Cassell, J. Stocky, T.; Bickmore, T.; Gao, Y.; Nakano, Y.; Ryokay, K.; Tversky, D.; Vaucelle, C.; Vilhjálmsson, H; *MACK: Media lab Autonomous Conversational Kiosk* in *Proceedings of Imagina02*, Monte Carlo; 2002

[63] Stocky, T.; *Conveying Routes: Multimodal Generation and Spatial Intelligence in Embodied Conversational Agents*; MhD Thesis, Electrical Engineering and Computer Science, MIT, Cambridge, MA; 2002

[64] Kopp, S.; Wachsmuth, I.; *Planning and motion control in lifelike gesture: A refined approach* in *Proceedings of Computer Animation 2000*, Philadelphia, USA, pp.104-111; 2000

[65] Kopp, S.; Wachsmuth, I.; *A knowledge-based approach for lifelike gesture animation* in *Proceedings of the 14th European Conference on Artificial Intelligence*, Amsterdam, IOS Press; 2000

[66] Wachsmuth, I.; Kopp, S.; *Lifelike Gesture Synthesis and Timing for Conversational Agents* in Wachsmusth, Sowa (eds.), *Gesture and Sign Language in Human-Computer Interaction, International Gesture Workshop (GW 2001)*, pp.120-133, Springer-Verlag; 2002

[67] Prillwitz, S.; Leven, R.; Zienert, H.; Hamke, T.; Henning, J.; *HamNoSys Version 2.0: Hamburg Notation System for Sign Languages: An Introductory Guide*, volume 5 of *International Studies on Sign Language and Communication of the Deaf*; Hamburg, Germany: Signum Press; 1989

[68] SABLE; *SABLE: A Synthesis Markup Language (version 1.0)*; www.bell-labs.com/project/tts/sable.html

[69] Kopp, S.; Tepper, P.; Cassell, J.; *Towards Integrated Microplanning of Language and Iconic Gesture for Multimodal Output* in *Proceedings of the International Conference on Multimodal Interfaces (ICMI'04)*, pp.97-104, ACM Press; 2004

[70] Tepper, P.; Kopp, S.; Cassell, J.; *Content in Context: Generating Language and Iconic Gestures without a Gestionary* in *Working Notes AAMAS'04* Workshop on *Balanced Perception and Action for Embodied Conversational Agents*, New York, pp.79-86; 2004

[71] Kopp, S.; Tepper, P.; Ferriman, K.; Cassell, J.; *Trading Spaces: How Humans and Humanoids use Speech and Gesture to Give Directions* in *Spatial Cognition and Computation*; Lawrence Erlbaum Associates, Inc.; in press

[72] Unuma, M.; Anjou, K.; Takeuchi, R.; *Fourier principles for emotion-based human figure animation* in *Proceedings of SIGGRAPH 1995*, Los Angeles, CA, pp.91-96; 1995

[73] Brudelin, A.; Williams, L.; *Motion signal processing* in *Proceedings of SIGGRAPH 1995*, Los Angeles, CA, pp.97-104; 1995

[74] Amaya, K.; Bruderlin, A.; Calvert, T.; *Emotion from motion* in *Proceedings Graphics Interface'96*, pp. 222-229; 1996

[75] Rose, C.; Bodenheimer, B.; Cohen, M.; *Verbs and Adverbs: Multidimensional Motion Interpolation* in *IEEE Computer Graphics and Applications*, vol. 18 (5):32-40; 1998

[76] Chi, D.; Costa, M.; Zhao, L.; Badler, N.; *The EMOTE model for effort and shape* in *Proceedings of SIGGRAPH 2000*, New Orleans LA, pp. 173-182; 2000

[77] Hartmann, B.; Mancini, M.; Pelachaud, C.; *Implementing Expressive Gesture Synthesis for Embodied Conversational Agents* in *Gesture Workshop*, LNAI; Springer; 2005

[78] Solmer, A.; *Manual de Teatro*; Temas e Debates, Lda.; 2003

[79] Arafa, Y.; Kamyab, K.; Mamdani, E; *Character Animation Scripting Languages: A Comparison* in *Proceedings of the International Conference on Autonomous Agents 2003*, Melbourne, Australia; 2003

[80] Kopp, S.; Krenn, B.; Marsella, S.; Marshall, A.; Pelachaud, C.; Pirker, H.; Thórisson, K.; Vilhjálmsson, H.; *Towards a Common Framework for Multimodal Generation: The Behavior Markup Language* in *Proceedings of Intelligent Virtual Agents (IVA) 2006*, pp.205-217; 2006

[81] VHML; *VHML – Virtual Human Markup Language*; www.vhml.org/

[82] SMIL; *SMIL: Synchronized Multimedia*; www.w3.org/AudioVideo/

[83] Kranstedt, A.; Kopp, S.; Wachsmuth, I.; *MURML: A Multimodal Utterance Representation Markup Language for Conversational Agents* in *AAMAS'02 Workshop Embodied conversational agents-let's specify and evaluate them!*, Bologna, Italy; 2002

[84] Ruttkay, Z.; Noot, H.; *Variations in Gesturing and Speech by GESTYLE* in *International Journal of Human-Computer Studies*, Special Issue on 'Subtle Expressivity for Characters and Robots', 62(2), 211-229; 2005

[85] de Carolis, B.; Pelachaud, C.; Poggi, I.; Steedman, M.; *APML, a Mark-up Language for Believable Behavior Generation* in H. Prendinger (ed.), Life-like Characters. Tools, Affective Functions and Applications, Springer; 2004

[86] Blumberg, B.; Galyean, T.; *Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments* in *Proceedings of SIGGRAPH 1995*, 30(3):47-54; 1995

[87] Perlin, K.; Goldberg, A.; *Improv: A System for Scripting Interactive Actors in Virtual Worlds* in *Proceedings of SIGGRAPH 1996*, pp.205-216; 1996

[88] Watt, A.; Policarpo, F.; *The Computer Image*; Addison-Wesley; 1999

[89] Akenine-Möller, T.; Haines, E.; *Real-time Rendering*, second edition; A K Peters; 2002

[90] Ebert, D.; Musgrave, F.; Peachey, D.; Perlin, K.; Worley, S.; *Texturing and Modeling. A Procedural Approach*; AP Professional; 1998

[91] Denavit, J.; Hartenberg, R.; *A kinematic notation for lower-pair mechanisms based on matrices* in *Journal of Applied Mechanics*, Vol.77, pp.215-221; 1955

[92] Spong, M.; Vidyasagar, M.; *Robot Dynamics and Control*; John Wiley & Sons; 1989

[93] Craig, J.; *Introduction to Robotics – Mechanics and Control*, third edition; Pearson Education, Inc.; 2005

[94] Tolani, D.; Goswani, A.; Badler, N.; *Real-time inverse kinematics techniques for anthropomorphic limbs* in *Graphics Models* 62, pp.353-338; 2000

[95] NASA; *NASA Man-Systems Integration Manual*; (NASA-STD-3000)

[96] Grosso, M.; Quach, R.; Otani, E.; Zhao, J.; Wei, S.; Ho, P.; Lu, J.; Badler, N.; *Anthropometry for Computer Graphics Human Figures*, Technical Report; University of Pennsylvania, Dept. of Computer and Information Science; 1987

[97] Dutoit, T.; *A Short Introduction to Text-to-Speech Synthesis*; tcts.fpms.ac.be/synthesis/introtts_old.html

[98] Festival; The Festival Speech Synthesis Systems; www.cstr.ed.ac.uk/projects/festival/

[99] Secretariado Nacional para a Reabilitação e Integração das Pessoas com Deficiência; *Gestuário – Língua Gestual Portuguesa*, fifth edition.

[100] Gut, U.; Looks, K.; Thies, A.; Trippel, T.; Gibbon, D.; *CoGest – Conversational Gesture Transcription System*, Technical Report; University of Bielefeld; 1993

[101] Kipp, M.; *ANVIL – A Generic Annotation Tool for Multimodal Dialogue* in *Proceedings of the 7th European Conference on Speech Communication and Technology*, Aalborg, pp.1367-1370; 2001

[102] Ortony, A.; Clore, G.; Collins, A.; *The Cognitive Structure of Emotions*; Cambridge University Press; 1988

[103] de Melo, C.; Paiva, A.; *Environment Expression: Expressing Emotions through Cameras, Lights and Music* in *Proceedings of Affective Computing Intelligent Interaction (ACII05)*, Beijing, China, pp.715-722; 2005

[104] Arijon, D.; *Grammar of the Film Language*; Silman-James Press; 1976

[105] Hornung, A.; *Autonomous Real-Time Camera Agents in Interactive Narratives and Games*; MhD thesis, Laboratory for Mixed Realities; 2003

[106] Birn, J.; *[digital] Lighting & Rendering*; New Riders; 2000

[107] Kaya, N.; *Relationship between color and emotion: a study of college students* in College Student Journal, September; 2004

[108] Juslin, P.; Sloboda, J.; *Music and Emotion: theory and research*; Oxford University Press; 2001

[109] Raimundo, G.; *Contador de histórias – A diferença entre o real e o virtual*, technical report; 2006

[110] Cassell, J.; Nakano, Y.; Bickmore, T.; Sidner, C.; Rich, C.; *Annotating and Generating Posture from Discourse Structure in Embodied Conversational Agents* in *Workshop on Representing, Annotating, and Evaluating Non-Verbal and Verbal Communicative Acts to Achieve Contextual Embodied Agents*, *Autonomous Agents 2001 Conference*, Montreal, Quebec; 2001

Appendix A – Virtual Human Skeleton

This appendix presents the virtual human skeleton formal definition. Section A.1 presents the human skeleton. Section A.2 defines the virtual human bone hierarchy. Finally, section A.3 defines the bone's frames of reference.

A.1 The Human Skeleton

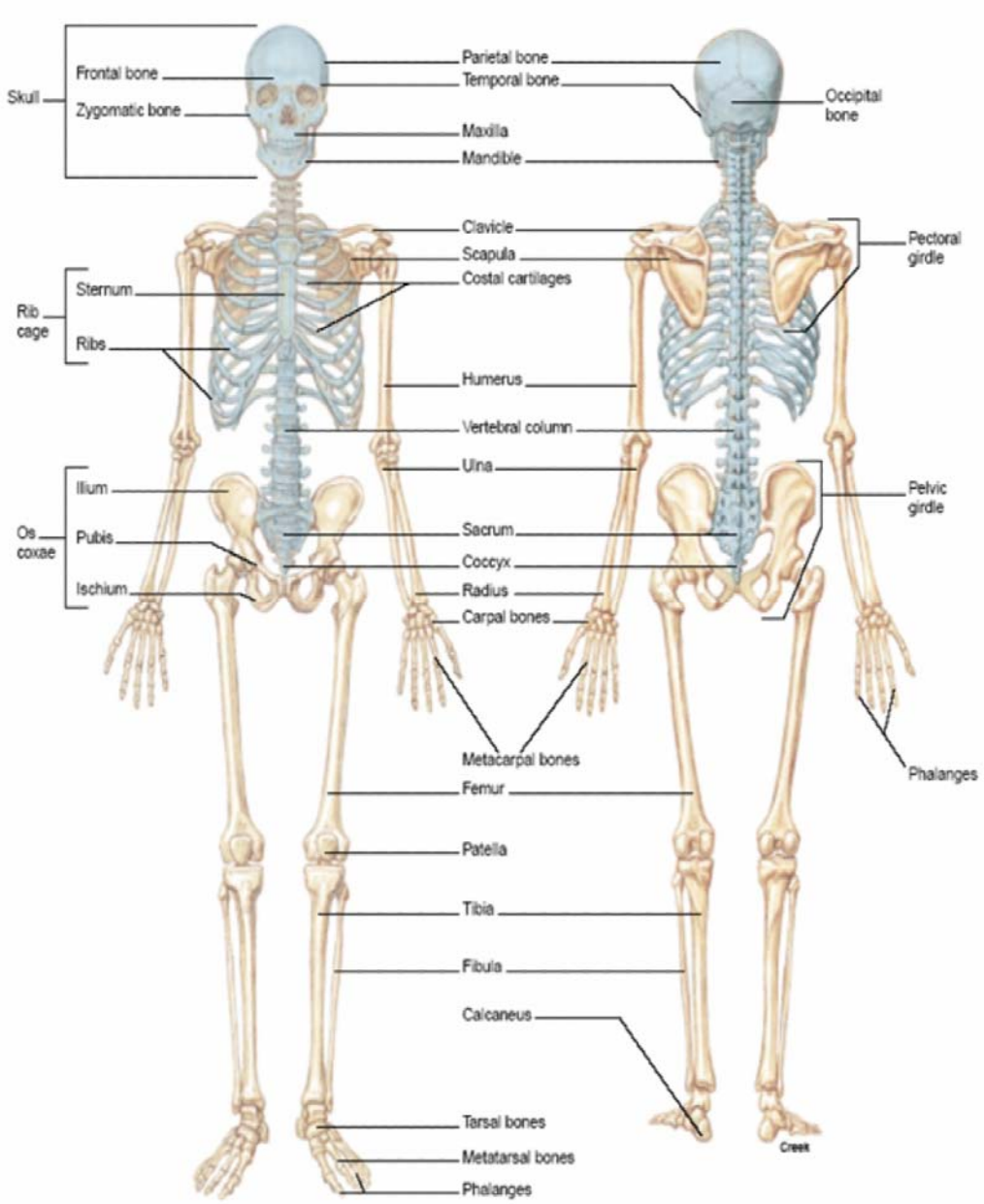


Fig.A.1 The human skeleton. On the left, the anterior view. On the right, the posterior view. Blue bones correspond to the axial system. Beige bones correspond to the appendicular system. [1]

A.2 Hierarchy

```
bone_sacroiliac
--bone_r_hip
    bone_r_knee
      bone_r_subtalar
        bone_r_midtarsal
          bone_r_metatarsal
--bone_l_hip
    bone_l_knee
      bone_l_subtalar
        bone_l_midtarsal
          bone_l_metatarsal
--bone_lumbar
----bone_thoracic
------bone_r_scapula
        bone_r_shoulder
          bone_r_elbow
            bone_r_wrist
              bone_r_index0
                bone_r_index1
                  bone_r_index2
              bone_r_middle0
                bone_r_middle1
                  bone_r_middle2
              bone_r_ring0
                bone_r_ring1
                  bone_r_ring2
              bone_r_pinky0
                bone_r_pinky1
                  bone_r_pinky2
              bone_r_thumb0
                bone_r_thumb1
                  bone_r_thumb2
------bone_l_scapula
        bone_l_shoulder
          bone_l_elbow
            bone_l_wrist
              bone_l_index0
                bone_l_index1
                  bone_l_index2
              bone_l_middle0
                bone_l_middle1
                  bone_l_middle2
              bone_l_ring0
                bone_l_ring1
                  bone_l_ring2
              bone_l_pinky0
                bone_l_pinky1
                  bone_l_pinky2
              bone_l_thumb0
                bone_l_thumb1
                  bone_l_thumb2
------bone_cervical
        bone_skull
          bone_chin
```

A.3 Frames of Reference

Name	Hand	Euler Z-Y-Z	Position – male (example)
bone_sacroiliac	L	(178.7079; 90.0000; -90.0000)	(-0.13; 37.82; 0.00)
bone_r_hip	R	(-83.8065; 178.1118; -173.7980)	(-3.89; 44.43; 0.68)
bone_r_knee	R	(88.2029; 173.5071; -1.8040)	(-3.89; 20.11; -0.12)
bone_r_subtalar	R	(-89.758; 143.5556; -179.8095)	(-3.89; 4.89; 1.61)
bone_r_midtarsal	R	(-89.9240; 91.1118; 179.9998)	(-3.89; 1.22; -1.10)
bone_r_metatarsal	R	(-89.9240; 96.0126; 179.9996)	(-3.89; 1.16; -4.47)
bone_l_hip	L	(-96.1374; 1.8407; 6.1522)	(3.89; 44.43; 0.68)
bone_l_knee	L	(91.7311; 6.5409; 178.2690)	(3.89; 20.11; -0.10)
bone_l_subtalar	L	(-90.2381; 36.4449; 0.1221)	(3.89; 4.89; 1.64)
bone_l_midtarsal	L	(-90.1413; 88.8054; 0.0031)	(3.88; 1.22; -1.06)
bone_l_metatarsal	L	(-90.1416; 83.9045; 0.0049)	(3.88; 1.15; -4.44)
bone_lumbar	L	(-179.6485; 89.9992; -90.0000)	(0.04; 45.07; 0.00)
bone_thoracic	L	(179.8697; 90.0000; -90.0000)	(-0.02; 53.96; 0.00)
bone_r_scapula	R	(104.5500; 90.0000; 90.0000)	(-0.91; 64.17; 1.46)
bone_r_shoulder	R	(89.3726; 90.0000; 89.9990)	(-7.24; 62.52; 1.46)
bone_r_elbow	R	(89.9871; 90.0001; 89.9990)	(-19.76; 62.66; 1.46)
bone_r_wrist	R	(93.8431; 90.0002; 89.9990)	(-30.10; 62.66; 1.46)
bone_r_index0	R	(179.9871; 163.3189; -180.000)	(-33.28; 62.81; -0.05)
bone_r_index1	R	(179.9871; 163.1309; -180.000)	(-34.98; 62.81; -0.56)
bone_r_index2	R	(179.9873; 163.131; -179.9998)	(-36.33; 62.81; -0.96)
bone_r_middle0	R	(179.9875; 172.9396; -180.000)	(-33.59; 62.81; 0.86)
bone_r_middle1	R	(179.9875; 172.7565; -180.000)	(-35.39; 62.81; 0.64)
bone_r_middle2	R	(179.988172.7565; -179.9995)	(-36.97; 62.81; 0.44)
bone_r_ring0	R	(179.9871; 173.7635; -180.000)	(-33.60; 62.61; 1.85)
bone_r_ring1	R	(179.9871; 173.5668; -180.000)	(-35.28; 62.61; 1.66)
bone_r_ring2	R	(179.9871; 173.5668; -180.000)	(-36.61; 62.61; 1.51)
bone_r_pinky0	R	(179.987; 177.7818; -179.9999)	(-33.59; 62.21; 2.65)

bone_r_pinky1	R	(179.987; 177.4742; -179.9999)	(-34.67; 62.21; 2.61)
bone_r_pinky2	R	(179.987; 177.4742; -179.9999)	(-35.67; 62.21; 2.56)
bone_r_thumb0	R	(179.9870; 139.5749; 154.3371)	(-31.37; 61.41; 0.18)
bone_r_thumb1	R	(175.4640; 144.9105; 150.7566)	(-32.36; 60.79; -0.66)
bone_r_thumb2	R	(175.4640; 144.9105; 150.7566)	(-33.02; 60.41; -1.11)
bone_l_scapula	L	(75.4502; 90.0000; -90.0000)	(0.91; 64.17; 1.46)
bone_l_shoulder	L	(90.6230; 90.0000; -89.9994)	(7.23; 62.52; 1.46)
bone_l_elbow	L	(90.0086; 90.0000; -89.9994)	(19.76; 62.66; 1.46)
bone_l_wrist	L	(86.1526; 89.9999; -89.9994)	(30.10; 62.66; 1.46)
bone_l_index0	L	(0.0086; 16.6826; 0.0000)	(33.28; 62.81; -0.05)
bone_l_index1	L	(0.0086; 16.8705; 0.0000)	(34.98; 62.81; -0.56)
bone_l_index2	L	(0.0084; 16.8705; 0.0001)	(36.32; 62.81; -0.97)
bone_l_middle0	L	(0.0084; 7.0620; -0.0001)	(33.59; 62.81; 0.86)
bone_l_middle1	L	(0.0084; 7.2452; -0.0001)	(35.39; 62.81; 0.63)
bone_l_middle2	L	(0.0079; 7.2452; 0.0003)	(36.97; 62.81; 0.43)
bone_l_ring0	L	(0.0088; 6.2382; -0.0001)	(33.59; 62.61; 1.85)
bone_l_ring1	L	(0.0087; 6.4349; -0.0001)	(35.28; 62.61; 1.66)
bone_l_ring2	L	(0.0087; 6.4349; -0.0001)	(36.61; 62.61; 1.51)
bone_l_pinky0	L	(0.0090; 2.2198; -0.0004)	(33.59; 62.21; 2.65)
bone_l_pinky1	L	(0.0089; 2.5288; -0.0003)	(34.67; 62.21; 2.61)
bone_l_pinky2	L	(0.0089; 2.5288; -0.0003)	(35.67; 62.21; 2.56)
bone_l_thumb0	L	(0.0086; 40.4267; -25.6629)	(31.37; 61.41; 0.17)
bone_l_thumb1	L	(4.5315; 35.0912; -29.2432)	(32.35; 60.79; -0.67)
bone_l_thumb2	L	(4.5315; 35.0912; -29.2432)	(33.02; 60.41; -1.11)
bone_cervical	L	(179.9640; 90.0019; -90.0369)	(0.00; 64.11; 0.00)
bone_skull	L	(179.9038; 90.0039; -90.0330)	(0.01; 68.99; 0.00)

Table A.1 Virtual human bone frames of reference.

References

[1] V. de Graaf; *Human Anatomy* – sixth edition; McGraw Hill; 2002

Appendix B – Robotic Manipulators Anthropometry

This appendix lists virtual human joint limits based on the 95th percentile of the population in [1] cited in [2]. Only relevant values are presented. All measures are relative to the anatomical neutral position. Frames are defined as follows: *axis X* – normal to the coronal plane (*YoX*); *axis Y* – normal to the sagittal plane (*YoZ*); *axis Z* – normal to the transversal plane (*XoZ*).

B.1 Male Population 95th Percentile

	Axis X		Axis Y		Axis Z	
Joint	min	max	min	max	min	max
neck	-63.5	63.5	-71.0	70.0	-99.1	99.6
cervical	-45.0	45.0	-25.0	40.0	-10.0	10.0
shoulder l	-188.7	63.0	-83.3	210.9	-	-
elbow l	-	-	0.0	159.0	-	-
wrist l	-47.9	36.7	-78.0	94.8	-125.8	26.1
shoulder r	-188.7	63.0	-83.3	210.9	-	-
elbow r	-	-	0.0	159.0	-	-
wrist r	-47.9	36.7	-78.0	94.8	-125.8	26.1
thigh l	-53.5	53.5	-148.0	148.0	-	-
knee l	-	-	0.0	145.6	-	-
ankle l	-39.0	35.0	-19.9	79.6	-55.0	63.0
thigh r	-53.0	53.5	-148.0	148.0	-	-
knee r	-	-	0.0	145.6	-	-
ankle r	-39.0	35.0	-19.9	79.6	-55.0	63.0

Table B.1 Male population 95th Percentile. ([1] in [2])

B.2 Female Population 95th Percentile

	Axis X		Axis Y		Axis Z	
Joint	min	max	min	max	min	max
neck	-77.2	63.5	-84.4	70.0	-109.0	108.8
cervical	-45.0	45.0	-25.0	40.0	-10.0	10.0

shoulder l	-192.9	63.0	-87.9	217.9	-	-
elbow l	-	-	0.0	165.9	-	-
wrist l	-43.0	36.1	-74.7	98.1	-135.9	28.9
shoulder r	-192.9	63.0	-87.9	217.9	-	-
elbow r	-	-	0.0	165.9	-	-
wrist r	-43.0	36.1	-74.7	98.1	-135.9	28.9
thigh l	-53.0	40.15	-148.0	148.0	-	-
knee l	-	-	0.0	145.2	-	-
ankle l	-39.0	35.0	-19.9	79.6	-55.0	63.0
thigh r	-53.0	40.15	-148.0	148.0	-	-
knee r	-	-	0.0	145.2	-	-
ankle r	-39.0	35.0	-19.9	79.6	-55.0	63.0

Table B.2 Female population 95th Percentile. ([1] in [2])

References

[1] NASA; *NASA Man-Systems Integration Manual*; (NASA-STD-3000)

[2] Grosso, M.; Quach et al.; Otani, E.; Zhao, J.; Wei, S.; Ho, Pei-Hwa; Lu, Jiahe; Badler, N.; *Anthropometry for Computer Graphics Human Figures*; 1987

Appendix C – GestuRA – Gesture Recording Algorithm

Abstract

In order to understand Human gesticulation and, in particular, to simulate it in computational systems it is, first, necessary to *transcribe* it. This document presents a gesture transcription algorithm – *Gesture Recording Algorithm (GestuRA)* – which focuses on gesticulation *communication* and *kinesics*. On the one hand, linguistically motivated, the algorithm follows McNeill's theory of gestures. Thus, gestures are not categorized based on physical aspects, but on their communicative function. On the other hand, kinesic aspects are recorded so as to support reproducibility. Furthermore, a formal language for form transcription is proposed – *Gesture Scripting Language*.

GestuRA is an iterative algorithm structured into seven passes. In the first, words are transcribed from the video-speech record. In the second, text is organized into utterances. On the third, utterances are classified according to discourse level. On the fourth, gestures are classified and gesticulation is filtered. On the fifth, gesticulation phases are annotated. On the sixth, gesticulation form is annotated. Finally, on the seventh, gesticulation is classified according to its dimensions and its meaning analyzed.

GestuRA Process (GestuRAP), the broader process under which GestuRA can be applied, is presented. GestuRAP involves goal definition, data gathering, data modification, GestuRA application, statistical analysis and conclusion.

Keywords:

Gesture Transcription Algorithm, Gesticulation, McNeill's Theory

Contents

Introduction

In order to develop gesticulation models it is necessary to study and methodically *transcribe* human gesticulation. In this sense, the present work has the following goals:

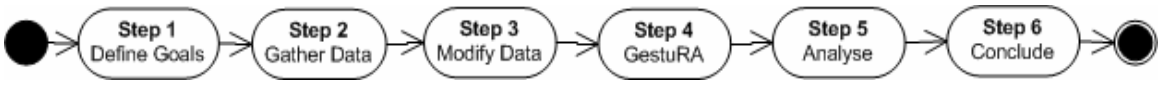
- Research the state-of-the-art in gesture transcription algorithms;
- Research technology which supports gesture transcription algorithms;
- Develop and document GestuRA, a gesture transcription algorithm.

The document is organized as follows. Section C.1 introduces GestuRAP, a process which provides the broader context – from goal definition, to data gathering, statistical analysis and conclusion – under which a transcription algorithm applies. Section C.2 presents the state-of-the-art in gesture transcription algorithms and relevant technology. Section C.3 presents GestuRA, the proposed gesture transcription algorithm. Section C.4 proposes an Anvil implementation of GestuRA. Finally, section C.5 draws conclusions and discusses further work.

C.1 The GestuRA Process

This section introduces a methodology which supports the broader context under which a gesture transcription algorithm is applied. This methodology is called *GestuRA Process (GestuRAP)* and is composed of six steps – C.Fig. 1:

- *Step 1, define goals* – in which the analyst defines the experiment's goals;
- *Step 2, gather data* – Upon goal definition, data should be gathered. However, first, the following should be considered: Which *stimulus* should be used? Should it be a video? Or text? How long should it be? How should data be gathered? What should be said to the subjects before presenting the stimulus? How should the experiment be recorded? Should the camera be concealed?
- *Step 3, modify data* – Upon data gathering but before applying the gesture transcription algorithm the analyst may need to modify the collected data. Data modification can involve the following operations: (1) *video-speech record digitalization*, i.e., convert the record from analog to digital format; (2) *video-speech record segmentation*, i.e., divide the record into smaller chunks for the purpose of parallel analysis; (3) *audio record separation*, i.e., separate the audio component from the video-speech record. Furthermore, the audio's intensity and pitch sub-components could also be obtained and background noise removed; (4) *text transcription*, i.e., from the audio record obtain time aligned text transcription;
- *Step 4, GestuRA* – In this step, GestuRA, the proposed gesture transcription algorithm is applied to each segment (or whole record). Any other algorithm could, however, be applied;
- *Step 5, analyze* – Once information has been gathered, statistical analysis ensues;
- *Step 6, conclude* – Having gathered and modified the data, transcribed gestures, performed statistical analysis, the analyst should be able to draw appropriate conclusions.



C.Fig. 1 - GestuRAP overview.

C.2 Background

In this work only linguistically motivated gesture transcription algorithms are relevant. In this sense, algorithms which are based solely on gesture kinesic aspects are ignored. In this context, two transcription algorithms – *McNeill Lab's* and *CoGesT* – are described.

C.2.1 McNeill Lab's Transcription Algorithm

This algorithm is based entirely on McNeill's theory [1]. The algorithm's gesture categories – iconic, metaphoric, deictic and beat – distinguish references to concrete events, abstract concepts, relations and orientations. Thus, the concern is in *what* the gesture mean. The algorithm provides a guided, systematic, and disciplined method for inferring these meanings and functions and for transcribing speech-gesture synchronization.

The algorithm consists of eight passes through the speech-video record: [2][3]

- Pass 1 – Consists of watching the complete speech-video record all the way through;
- Pass 2 – Consists of transcribing all words spoken in the discourse, from beginning to end, as one big paragraph. Grammatical structuring is disregarded as well as narration pauses;
- Pass 3 – Consists of organizing the speech into short utterances, reflecting the grammatical structuring of the speech sequences. Here, each clause is time stamped, the narrator pauses are timed and annotated and the listener interventions ("mm-mm", laughter, etc.) annotated;

The following steps are executed recursively on chunks rather than on the whole discourse.

- Pass 4 – Consists of annotating, on the transcript obtained from passes 2 and 3, points of primary peak prosodic emphasis;
- Pass 5 – Consists of annotating gesture phrases;
- Pass 6 – Consists of annotating the within-phrase gesture phases. Each phase is time stamped. Additionally, nested gestures, which are discernible in this pass, are annotated;
- Pass 7 – Consists of reorganizing the short utterances transcript in accord with what gesture phraseology reveals about the organization of communicative dynamism peaks. This pass involves analyst's judgment on the discourse structure of the example under consideration;
- Pass 8 – Consists of a revision pass. This pass, reflects the backward-adjusting nature of gesture analysis and annotation. As the analyst gains insight into the narrator's idiosyncratic gesture style, previous annotations may be clarified.

C.2.2 Conversational Gesture Transcription System

Conversational Gesture Transcription System (CoGesT) [4] is a twofold system for gesture transcription in conversational contexts. First, it provides a system of linguistically motivated categories for gestures. Second, it is a machine and human readable transcription scheme.

CoGesT distinguishes two gesture aspects: *form* and *function*. Regarding form, three spatiotemporal-oriented phases are considered: *source*, corresponding to the initial position; *target*, corresponding to the final position; and, *trajectory*, which describes the path between source and target. These phases contrast with the function-oriented phases in McNeill's theory. Additionally, the gesture's hand shape, symmetry and modifiers (speed of execution, number of

repetitions, and size of the gesture) also characterize the gesture's form. Regarding gesture's function, CoGesT acknowledges that meaning is conveyed in a multimodal way and, in particular, using both speech and gestures. In this sense, while communicating, three situations may arise: (1) neither speech, nor gesture occurs – CoGesT simply ignores this situation since it carries no communicative significance; (2) only speech occurs – CoGesT classifies these gestures as held postures which do not convey meaning beyond emotive body language; (3) only gesture or both occurs – these gestures are classified as either *gestural idioms* or *non-conventionalized gestures*. Gestural idioms have a particular meaning understood within a particular cultural group. These include the emblematic and deictic gestures as defined by McNeill. Non-conventionalized gestures, on the other hand, do not have an immediately apparent meaning by themselves. These gestures subsume McNeill's metaphorics and beats.

CoGesT defines an annotation scheme for transcribing gestures according to form and function. Transcribed gesture parameters include: (1) source and target; (2) location; (3) hand shape; (4) directionality; (5) trajectory shape; (6) modifiers; (7) symmetry; (8) function.

C.2.3 Technology

The GestuRA Process (see section C.1) can benefit throughout from various software tools:

- *Video Editing* - In GestuRAP's step 3 the recorded video-speech narration can be edited for many purposes. Of particular interest, is video segmentation. Analyzing smaller segments improves performance for many of the tools used in the next steps. Additionally, it can be an efficient strategy for team work. Movie editing can also include: audio retrieval; movie crop and/or compression; zoom to focus on a particular feature of the movie; etc. Useful tools are: *VirtualDub* [5], developed by Avery Lee, which is a free video capture/processing utility streamlined for fast linear operations; *Adobe Premiere [6]*, developed by Adobe, which is a commercial tool for real-time editing for professional video production;
- *Audio Editing* - In GestuRAP's step 3 audio from the recorded video-speech narration can be edited, for instance, to remove background noise. However, greatest benefit comes from audio analysis. As gestures and speech synchronize in many ways [1], it is useful to transcribe the text and synchronize it with the video. This time-consuming operation can be manually realized with the help of pitch, intensity and prosody analysis of the audio file. Besides text synchronization, gesture phase annotation can also benefit from pitch and intensity analysis of the audio file. A good tool for analysis and reconstruction of speech signals is PRAAT [7];
- *Multimodal annotation* - McNeill Lab's gesture transcription algorithm (see subsection C.2.1) barely considers technology. It proposes a method for annotating gestures directly

into the text transcription and for realizing text synchronization by ear. Today's technology can significantly improve annotation efficiency and accuracy. Some useful tools were already described in the previous sections. However, greatest benefit comes from so-called *multimodal annotation tools*. These are generic multi-layered tools which allow playback of audiovisual material at various speeds and allow insertion of time anchored general-content tags in the layers. These tools are, thus, ideal for supporting GestuRAP's step 4. Anvil [8] is one of the most referenced. It is XML based, platform-independent and supports all the common operations in multimodal annotation. Additionally, supports easy integration with PRAAT and with statistical analysis tools;

- *Data analysis* - GestuRAP's step 5 is about statistical analysis of gathered data. Thus, use of data analysis and statistic tools can be helpful. Currently, there are many such commercial tools, for instance: *StatSoft's STATISTICA* [9]; *SPSS* [10]; and *Microsoft Excel*.

C.3 GestuRA - Gesture Recording Algorithm

Gesture Recording Algorithm (GestuRA) is a linguistically motivated gesture transcription algorithm. GestuRA has two main goals:

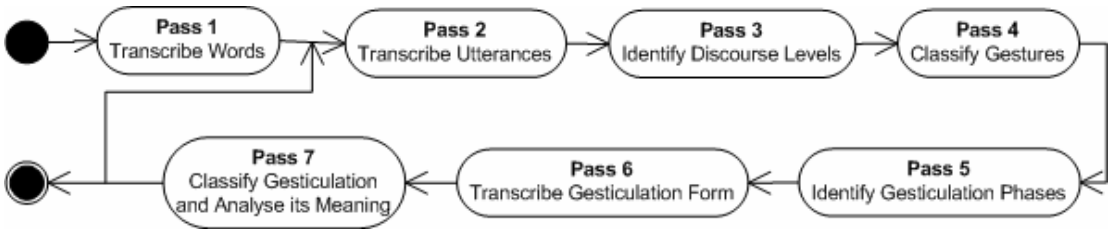
- Understand gesture contribution to overall meaning of the speaker's communication;
- Annotate enough gesture information so as to be able to exactly reproduce it.

The first goal states the algorithm's *linguistic* motivation. Meaning in one's communication is conveyed through context, the verbal and non-verbal channel. The purpose is to understand *what* the gesture means and how does it contribute to overall communication.

The second goal focuses on gesture *form*. The purpose is to understand *how* and *when* is the gesture executed. The idea is that a machine, which would receive a gesture transcription as input, would be able to reproduce it all.

The algorithm is mainly based on McNeill Lab's Transcription Algorithm (see subsection C.2.1). However, gesture *form* transcription is closer to CoGesT (see subsection C.2.2).

The algorithm's main structure has seven passes – C.Fig. 2. Notice, reflecting its experimental nature, it is iterative. The following subsections describe each pass in detail.



C.Fig. 2 - GestuRA overview.

C.3.1 Pass 1 – Transcribe Words

Concept

In the first pass all the words, including pauses, in the spoken discourse are transcribed into textual format. Grammatical structure, such as punctuation, is not relevant. Furthermore, in order to save gesture-speech synchronization information, transcribed words should be time stamped. Precision should at least be to the $1/100^{th}$ of a second. Finally, prosody, pitch and audio intensity constitute valuable information and should also be transcribed and synchronized with the video-speech record.

Rationale

Gesticulation and speech are manifestations of the same underlying mental process [1]. Therefore, in order to understand gesture meaning it is important to understand the correlation with speech. Thus, a transcription algorithm should logically begin by distinguishing verbal from non-verbal information. Now, it could be argued that speech in audio format would be enough. However, transcribing speech into textual format will, first, force the analyst to clarify the speaker's speech and, second, allow for more flexible manipulation of speech information. As gesture synchronizes with speech at least at the syllable level [1], word synchronization should be very precise. Finally, prosody, pitch and intensity information will help understand correlation between gesture and discourse structure.

C.3.2 Pass 2 – Transcribe Utterances

Concept

In the second pass, speech is organized into short utterances, reflecting speech sequences' grammatical structure. Each utterance is associated with the communication of a single "idea".

Rationale

McNeill [1] argues that an idea of thought, at its primitive stage, exists as a combination of imagistic (idiosyncratic) and linguistic (cultural) information – this is what he calls the *growth point*. The process of communication transforms the growth point into an utterance plus gestural manifestation. Thus, in order to understand the idea of thought, the analyst needs to look at the utterance as whole as well as to gestures. Thus, the second pass should logically be to group transcribed words into utterances.

However, one could reasonably argue that if an idea manifests itself through the verbal and non-verbal channels, then how would it be possible to perceive it just by listening to speech? In fact, it is not. Gestural information would be required. However, in order to understand gestural information one needs to listen to the utterances. It is a circularity problem. So, as a first cut to transcribing the sequence of ideas which constitute the narration, text is organized into

utterances just by listening and *informally* looking at gestures. On the next *iteration*, utterance organization can be reviewed. This is one reason why GestuRA needs to be an iterative process.

C.3.3 Pass 3 – Identify Discourse Levels

Concept

In the third pass, speech is organized into discourse levels – narrative, metanarrative and paranarrative. *Narrative level* references correspond to events related to the main story line. *Metanarrative level* references correspond to the event of observing the stimulus itself. *Paranarrative level* references refer to the event of storytelling.

Rationale

Each discourse level is characterized by different types of gesticulation [1]. Additionally, typical gesticulation occurs when changing level. Therefore, it is important to annotate discourse level in order to better understand the meaning of gestures. Discourse levels should, logically, only be annotated after clause identification and transcription. Thus, the third pass is correctly placed. Finally, a circularity problem similar to the one discussed in subsection C.3.2 exists. The solution is also similar and is based on the recursive nature of the algorithm.

C.3.4 Pass 4 - Classify Gestures

Concept

In the fourth pass every gesture in the video-speech record is classified. Here, gestures are classified according to the following categories: (1) *sign*, if it is a symbol of some sign language; (2) *emblem*, if it is culturally defined such as the “ok” sign; (3) *gesticulation*, if it is a spontaneous non-verbal manifestation of the speaker’s communicative intent; (4) *adaptor*, if it satisfies personal needs and/or helps adapt to the environment. Regarding timing information, whenever possible, gestures should be aligned to phonemes or syllables instead of words in the speech. Finally, gesture classification should be classified according to a confidence scale, where one (1) corresponds to “marginally confident” and four (4) to “totally certain”.

Overlapping and Nested Gestures

Two gestures *overlap* if one begins before the other but, ends within the time span of the second. A gesture is *nested* if its time span resides totally inside the time span of a second gesture. Gesture overlapping and nesting can occur in storytelling. Regarding overlap, the analyst should follow these guidelines: if overlapping gestures *appear* to be both gesticulations *concatenate* their time spans, i.e., annotate the first gesture’s starting time and the last gesture’s finishing time; otherwise, annotate separately each of the gestures’ time boundaries. Regarding

nesting, the analyst should follow these guidelines: if gestures are of the same type, ignore the nested gesture; otherwise, annotate separately each phase time boundaries.

Rationale

In this pass, gesture analysis begins. The purpose is to filter gestures which relate to the ongoing speech, i.e., *gesticulation* gestures. From this pass on, only these will be relevant to understand the meaning of the speaker's communication. Following this work's linguistic orientation, gesture classification is based on McNeill's taxonomy[5]. Regarding the confidence scale, this reflects the experimental nature of the process. As more iterations are made, confidence should increase. Regarding gesture overlap and nesting, whenever these phenomena involves two or more gesticulation gestures the guidelines suggest ignoring, for the moment, the nested or overlapping gesture. This is because it can be difficult to define with precision these boundaries without exploring the gesticulation *phases* which are the focus of the next pass.

C.3.5 Pass 5 – Identify Gesticulation Phases

Concept

In the fifth pass, the analyst identifies the gesticulation phases. Namely: (1) *preparation* (optional) – in which the limb moves away from its rest position to a position where the stroke begins; (2) *pre-stroke hold* (optional) – which is the position and hand posture reached at the end of preparation itself; this may be held more or less briefly; (3) *stroke* – which is the peak of the effort in the gesture; (4) *post-stroke hold* (optional) – which is the final position and posture of the hand reached at the end of the stroke; (5) *retraction* (optional) – which is the return of the hand and limb to a rest position; (6) *partial-retraction* (optional) – is a partial return of the hand and limb to the rest position; normally occurs in gesticulation sequences. Timing information should be annotated to the phoneme or syllable level. Finally, regarding gesticulation overlap and nesting, the analyst identifies the exact boundaries of nested and overlapping gesticulation.

Rationale

Having classified gestures in the previous pass, this pass begins gesticulation analysis. This analysis spans phase, form, meaning and classification. On the one hand, form refers to kinesic aspects at the various phases (see section C.3.6). On the other hand, annotating meaning and classification can benefit from phase annotation (see section C.3.7). Thus, this analysis should begin with phase annotation and proceed, in the next passes, to analyze form, classification and meaning. Further, having defined phases, *phrase* definition is straightforward. Regarding phase categories, they follow GestuRA's linguistic motivation and are, naturally, based on McNeill's

[5] Pantomimes, however, will *not* be considered.

theory. Regarding timing information, it should be as accurate as possible, because phases are important to understand both gesticulation meaning and the correlation with discourse structure. Regarding gesticulation overlap and nesting, by identifying the inner structure of gesticulation the analyst can discern gesticulation time boundaries. For instance, each gesticulation should have only one stroke phase.

C.3.6 Pass 6 – Transcribe Gesticulation Form

Concept

In the sixth pass, gesticulation *form* is annotated. Form is annotated through the following features: (1) *phases' source*; (2) *phases' target*; (3) *phases' motion*; (4) *handedness*; (5) *symmetry*; (6) *hand shapes*; (7) *palm orientations*; (8) *body shape*. Before discussing these in detail, however, it is important to define relevant *frames of reference*.

Frames of reference

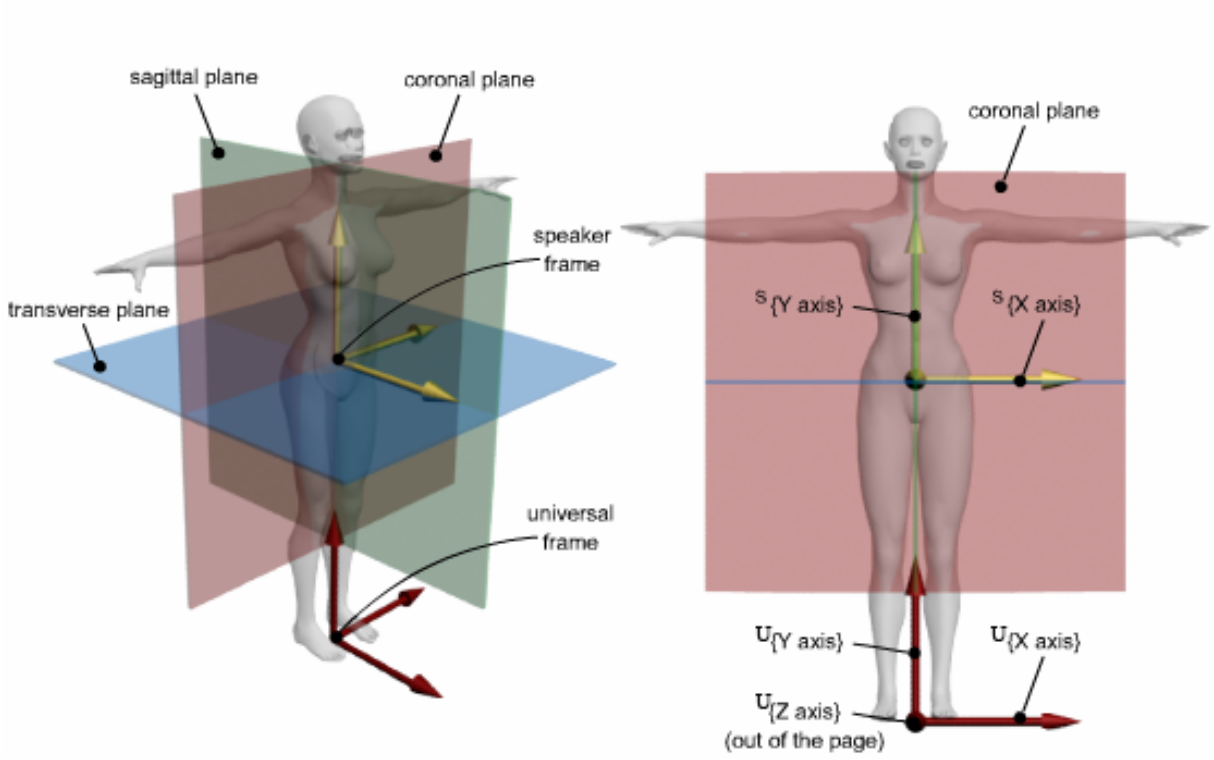
In order to transcribe gesticulation from a kinesic perspective, the analyst needs to position body parts in three-dimensional space. For this purpose, two frames of reference shall be defined: the global *universal frame of reference*; and the local *speaker frame of reference*. To define a frame, two parameters need to be defined: the *position*; and, the *orientation*. Regarding the universal frame, the position is defined relative to the first video frame of the video-speech record. Imagining the speaker in the open anatomical position[6] on the aforementioned video frame, the position is defined as the intersection of the floor with the normal to the floor plane that passes through the speaker's centre of mass[7]. Regarding orientation, the *y axis* is perpendicular to the floor plane and points upward. The *x axis* is parallel to the arms and points towards the left. Finally, the *z axis* completes the orthonormal frame using the right hand rule.

The universal frame is defined once and is fixed throughout the video-speech record however, it would be useful to have a frame which "moved" with the speaker. This is the purpose of the speaker frame, which is positioned at the speaker's centre of mass. Its orientation is *fixed* and the same as the universal frame of reference. This frame defines three planes which divide space into eight subspaces: the *coronal plane* which corresponds to all points with null *z* coordinate; the *sagittal plane* which corresponds to all points with null *x* coordinate; and, the *transverse plane* which corresponds to all points with null *y* coordinate. The eight subspaces

[6] In the open anatomical position, the body is erect, the feet parallel to each other and flat on the floor, the eyes are directed forward and the arms are stretched alongside the body with the palms of the hands facing down

[7] Calculating a human's centre of mass can be a daunting task. Normally, the centre of mass is located in the pelvic area. Thus, the analyst can assume it to be in the centre of mass of the pelvic girdle bone.

are: *ULF* – Up, Left, Front – which corresponds to all points with x>0, y>0 and z>0; *DRB* – Down, Right, Back; *ULB*; *URB*; *URF*; *DRF*; *DLB*; and *DLF*. C.Fig. 3 summarizes all reference frames as well as planes definitions.



C.Fig. 3 - The universal and speaker frames of reference. The figure also displays the sagittal, coronal and transverse planes defined by the speaker frame.

Gesticulation Features

For each phase, as defined in the previous pass, the following features should be annotated:

- *Source, target and motion* - the analyst should annotate the initial – *source* – and final – *target* – phase positions in the speaker frame. Notice that some of these points should be coincident – for instance, the preparation's target and stroke's source. *Motion* should also be annotated. Motion can be mathematically defined using a function in $\Re$(time) into $\Re^3$(position) or can just be annotated informally. The speaker frame's position in universal coordinates should also be annotated. Notice it is not always the case that these parameters need to be annotated for both hands. If the gesture is symmetric, for instance, only the dominant hand's parameters need to be annotated;
- *Handedness* – this feature can assume one of the following values: *RH* – only right hand involved; *LH* – only left hand involved; *BH*- both hands involved;
- *Symmetry* – this is a boolean which is true if the gesticulation is symmetric on both hands;

- *Hand shapes* – hand shapes are annotated on a per-phase basis. Every different hand shape, from a kinesic perspective, should be annotated, as well as the time of occurrence. Each hand shape is annotated as the *most similar* in the static gestures of the Portuguese Signal Language - C.Fig. 4. A modifier may be added to the shape if necessary. This feature should be annotated for both hands, except if the gesticulation only involves one or is symmetric in which case only the dominant hand is annotated;
- *Palm orientations* – palm orientation is defined by a vector which is normal to the palm plane and points outward. This feature should be annotated once per annotated hand shape;
- *Body shape* – this feature spans every kinesic aspect of the speaker's body which is not related to the hands. Sometimes non-verbal communication is not conveyed through the hands but, through other body parts or the whole body. The analyst should annotate body posture, informally, only if necessary to understand the speaker's communication.

Rationale

This pass continues gesticulation analysis. In particular, focus lies on gesticulation kinesics. This pass simply supports the second GestuRA's goal stated at the start of the appendix. Mathematically precise annotation supports machine reproducibility. Additionally, because analyst judgment is required to understand gesticulation communication, accurate form transcription will allow posterior interpretation by other analysts. Regarding hand shapes, standard hand shapes are based on a sign language so as to allow for some kind of normalization. Additionally, GestuRA only has to benefit from years of research and evolution that usually lead to a sign language. Hand shapes are non-coincident and varied. There is a high probability that every non-neutral non-ambiguous static hand shape has already been considered by the language. The reason to choose the *Portuguese* sign language was not scientific.

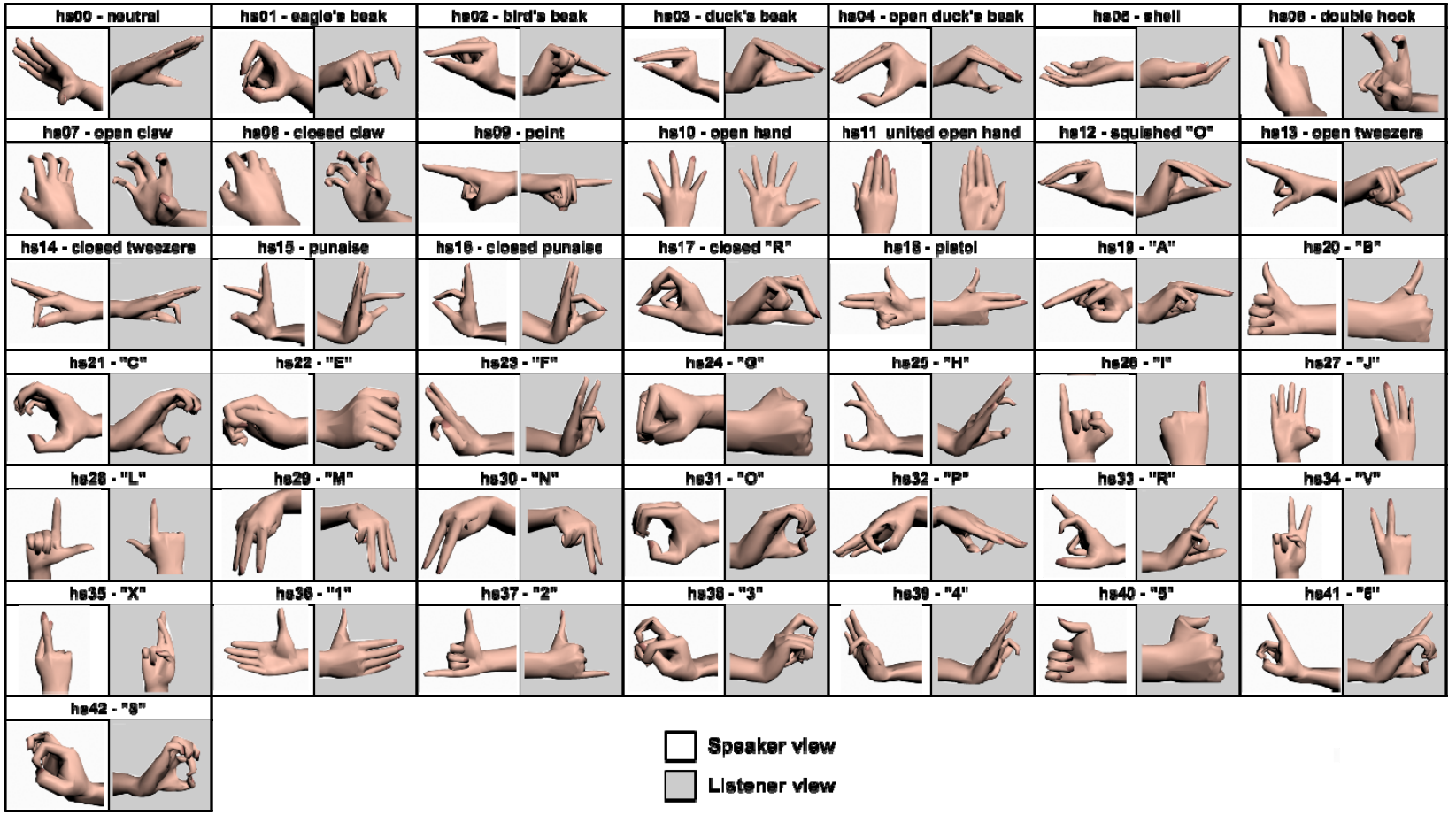
Formal Gesture Form Annotation

Gesture Scripting Language (GSL) is a formal language for gesture form annotation[8]. Built on top of GestuRA, GSL introduces a mathematical notation to describe gesture phases as sequences of hand shapes, palm orientations and positions.

Describing hand shapes

Hand shapes are simply described by the Portuguese Sign Language numeric code.

[8] In a related work, not described here, GSL annotations were automatically reproduced by virtual humans in digital environments.



C.Fig. 4 - Portuguese Sign Language Hand Shapes. For each hand shape the following is described: (1) *code*; (2) *name*; (3) *picture from the speaker's perspective*; (4) *picture from the listener's perspective*. All hand shapes use the speaker's dominant hand. Some shapes are distinguished based only on orientation. Even though for the present work this distinction is not relevant, hand orientation is described below just to maintain coherence relative to the language definition. However, no two gestures will be listed which differ only on orientation. Finally, regarding dynamism, only static shapes are listed.

Describing palm orientations

Palm orientations annotation reduces to the problem of representing an arbitrary orientation in three-dimensional space. GSL uses the rotation axis representation [12]. The axis is the palm orientation vector as defined above. The angle is defined according to the left-hand rule. C.Fig. 5 summarizes typical palm orientations for quick reference.

Example:

```
[-90;[0;0;1]] //Palm facing the left and -90 degrees rotation
```

Describing positions

Positions are simply defined by a three-dimensional vector describing the position relative to the speaker's frame of reference.

Example:

```
[0;10;0] //10 units above the speaker's frame of reference origin
```

Describing sequences

A single gesture phase can span several hand shapes, palm orientations and positions thus, GSL supports sequence definition through arrays.

Example:

```
[[0;0;0];[0;10;0]]      //Gesture phase starts at the speaker's frame
                        //and then raises 10 units
```

Handling non-symmetry

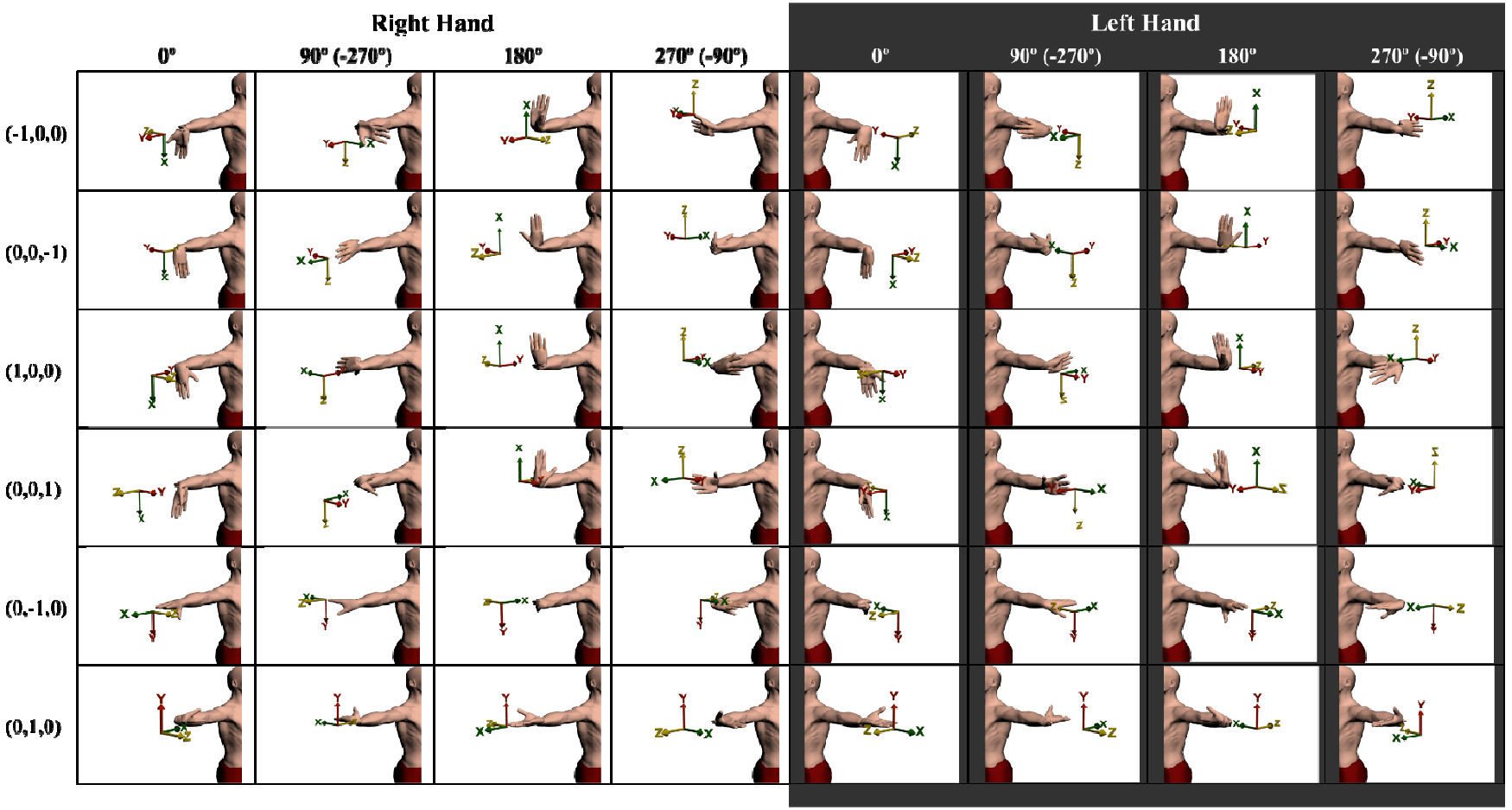
If a gesture is non-symmetric it is necessary to define different feature sequences for each hand. This is achieved in GSL through two labels: `RH`, for the right hand; `LH`, for the left hand.

Example:

```
RH=[-10;0;0];LH=[10;0;0]       /* Right hand is 10 units to the right
of the speakers' frame of reference and the left hand 10 units to the
left */
```

Describing time

By default, sequence items represent features evenly spaced in time. However, it might be convenient to represent features at arbitrary time offsets. Thus, GSL supports two ways of representing explicit time offsets: (1) absolute time offsets – which define an explicit amount of time after gesture phase start; (2) relative percentage time offsets – which defines a fraction of the total gesture phase duration. Furthermore, when time offsets are represented, features are represented as a two element array, where the first element is the time offset and the second is the feature as described above.



C.Fig. 5 - Typical palm orientations. Angle and axis values, according to GSL, are described for each orientation.

Example:

```
[[0.12;[-10;0;0]];[50%;[-10;10;0]]]  /* Hand starts moving 10 units to
the right 0.12 seconds after gesture phase start and 10 units above
after 50% of the total gesture phase time has elapsed */
```

C.3.7 Pass 7 – Classify Gesticulation and Analyze its Meaning

Concept

In the seventh and last pass, gesticulation is classified and its meaning annotated. Classification and meaning annotation are intertwined. Only by interpreting the meaning of a gesticulation can the analyst classify it. Therefore, for each gesticulation, the following procedure is suggested: (1) informally understand its meaning without annotating anything; (2) determine its most salient dimension as: *iconics; metaphorics; deictics* or *beats*. Apply the *beat filter*, described below, to identify beats. If it is not a beat, consider the informal meaning analysis from the previous step to determine the type; (3) formally annotate its meaning according to the parameters described below.

Beat filter

The beat filter [1] is a formal method, based solely on kinesic aspects, for differentiating imagistic (iconic and metaphoric) from beat gesticulation. The filter consists of a series of questions, and a score of one is added for each positive answer:

- Does the gesticulation have other than two phases (i.e., either one or three or more phases)?
- How many times does wrist or finger movement or tensed stasis appear in any movement phase not ending in a rest position? (add this number to the score)
- If the gesticulation begins in a non-centre part of space, is any subsequent phase executed in the centre space?
- If there are exactly two phases, is the space from the first phase different from the second's?

A score of zero means no imagery on formal grounds, and the gesture is probably a beat. Otherwise, the higher the score, the higher the imagery on formal grounds.

Iconics

To annotate the meaning of iconics, the analyst should annotate the following parameters: (1) *body meaning* – first, recall that iconic gesticulation is not necessarily executed with the hands but, may consist of body postures. This parameter registers what the iconic *form* represents. Which entity? Which character? Which object? (2) *motion meaning* – this parameter registers what the iconic *motion* represents; (3) *viewpoint* – this parameter registers the voice the speaker is using. There are three possibilities: (a) *character*, in which case the speaker should be using its whole body to gesticulate; (b) *observer*, in which case the gesticulation is concentrated on the

hands; (c) *both*, this viewpoint is uncommon but, logically possible; (4) *contribution* – this is the most important parameter as it registers the gesticulation contribution to the communication. What information is added to the one conveyed verbally? Why did the speaker gesticulate? This parameter shall be registered for all gesticulation types; (5) *confidence* – classification and meaning analysis should be evaluated according to a confidence scale where one corresponds to "marginally confident" and four to "totally certain".

Metaphorics

To annotate the meaning of metaphorics, the analyst should annotate the following parameters: (1) *base* – in a metaphoric an *abstract* concept is represented through a *concrete* entity. The base refers to the concrete entity; (2) *referent* – contrasting to the base, this parameter refers to the abstract concept [1]; (3) *motion meaning* – this parameter registers what the metaphoric motion represents; (4) *contribution* – this parameter registers the contribution to the speaker's communication; (5) *confidence* – classification and meaning analysis confidence.

Deictics

To annotate the meaning of deictics, the analyst should annotate the following parameters: (1) *geographic target* – this parameter registers where in the space surrounding the speaker is the deictic pointing at; (2) *narration target* – sometimes the space surrounding the speaker is associated with narration entities or abstract concepts [1]. This parameter registers the real meaning of the pointed space; (3) *contribution* – this parameter registers the contribution to the speaker's communication; (4) *confidence* – classification and meaning analysis confidence.

Beat

To annotate the meaning of beats, the analyst should annotate the following parameters: (1) *contribution* – this parameter registers the contribution to the speaker's communication; (2) *confidence* – classification and meaning analysis confidence.

Rationale

This pass concludes gesticulation analysis. Having annotated the phrase, form and phases the analyst should be ready to annotate the gesticulation meaning and classification. This pass has a component of subjectivity since the analyst must rely on its own experience to try to understand how is the *unconventionalized* gesticulation related to the *conventionalized* verbal utterance and, in the end, how does it contribute to the speaker's communication.

Having reached this point, the analyst has concluded a full algorithm's iteration. Thus, it should have a deeper understanding of whole speaker's communication and discourse structure. Thus, on the next iteration, the analyst can review utterance transcription (pass 2) and discourse

level definition (pass 3). Having reviewed these it can proceed to review gesticulation analysis and try to improve the confidence level associated with each gesture annotation.

C.4 GestuRA Implementation

This document proposes an implementation of GestuRA based on Anvil. Anvil is organized around the concept of *tracks*. Within a track, time stamped *elements* can be annotated. Each track is associated with a single *type* of element. Each element is associated with a set of *attributes*. An attribute can be a string, a number, a list, etc. Furthermore, Anvil supports two types of tracks[9]: *primary* and *secondary*. Primary tracks define elements with start and end times relating to the video-speech record. Secondary tracks define elements which relate to other track's elements. The latter is called the *referenced track*. Thus, secondary track elements' time intervals span the corresponding referenced track elements' time intervals. Finally, inheritance between tracks can be defined through the concept of a *group*. Tracks can be grouped together by a group node. Attributes defined at group level propagate to the tracks within the group. [8]

GestuRA's passes can be modeled as Anvil tracks:

- *Audio Tracks* – As was mentioned in section C.3, speech waveform, intensity and pitch data can be useful to better understand and annotate speech synchronization with gestures. Thus, the first two tracks maintain this information: (1) *waveform track* – contains the speech waveform synchronized with the video-speech record. Anvil supports waveform retrieval directly; (2) *pitch and intensity track* – contains pitch and intensity data. This data can be obtained through PRAAT;
- *Words Track* – GestuRA's pass 1 (see subsection C.3.1) – transcribe words – is implemented through the *words track*. Here, elements correspond to transcribed text words;
- *Utterances Track* – GestuRA's pass 2 (see subsection C.3.2) – transcribe utterances – is implemented through the *utterances track*. This is a secondary track which references the words track. The track's elements maintain utterance transcriptions;
- *Discourse Levels Track* – GestuRA's pass 3 (see subsection C.3.3) – identify discourse levels – is implemented through the *discourse levels track*. This is a secondary track which references the utterances track. The track's elements maintain discourse level annotation;
- *Gestures Track* – GestuRA's pass 4 (see subsection C.3.4) – classify gestures – is implemented through the *gestures track*. This track's elements maintain two attributes: (1) *gesture type*; (2) the classification *confidence-level*;

[9] A third track type – *set* – is not relevant to the present work.

- *Gesticulation Analysis Tracks* – GestuRA's passes 5, 6 and 7 are implemented through the *gesticulation group*. This group is further subdivided into two sub-groups: (1) *phases sub-group*; (2) *phrases sub-group*. Each of these sub-groups is composed of multiple *layer tracks*, where gesticulation is annotated. Multiple layers support gesticulation overlapping and nesting. GestuRA's pass 5 (see subsection C.3.5) – identify gesticulation phases – is implemented in the phases sub-group through the *phase attribute*. GestuRA's pass 6 (see subsection C.3.6) – transcribe gesticulation form – is implemented in both sub-groups. In the phases sub-group, per-phase form attributes are defined: (1) *source*; (2) *target*; (3) *motion*; (4) *hand-shapes*; (5) *palm-orientations*; (6) *body-shape*. In the phrases sub-group, per-gesticulation form attributes are defined: (1) *handedness*; (2) *symmetry*; (3) *speaker-frame-position*. GestuRA's pass 7 (see subsection C.3.7) – classify gesticulation and analyze its meaning – is implemented in the phrases sub-group. As was mentioned before, meaning is annotated through different parameters according to gesticulation dimensions. The parameters are represented through the following track attributes: (1) *phrase*; (2) *confidence-level*; (3) *contribution*; (4) *iconic-body-meaning*; (5) *iconic-motion-meaning*; (6) *iconic-viewpoint*; (7) *metaphoric-base*; (8) *metaphoric-referent*; (9) metaphoric-*motion-meaning*; (10) *deictic-geographic-target*; (11) *deictic-narration-target*.

C.5 Conclusions and Future Work

This document presented GestuRA which is an iterative algorithm structured into seven passes. In the first, words are transcribed from the video-speech record. In the second, text is organized into utterances. On the third, utterances are classified according to discourse level. On the fourth, gestures are classified and gesticulation is filtered. On the fifth, gesticulation phases are annotated. On the sixth, gesticulation form is annotated. Finally, on the seventh, gesticulation is classified according to its dimensions and its meaning analyzed.

GestuRA application on two Portuguese stories narration, which was not described here, presented some preliminary results. Regarding gesture information, collected data enabled a better understanding of the narrator's communicative intent. Regarding gesture form, gesture transcriptions were automatically reproduced by virtual humans with reasonable accuracy.

However, further GestuRA testing is needed. In particular, GestuRA determinism needs to be thoroughly evaluated. In this sense, different analysts should apply GestuRA to the same video-speech record and the respective transcriptions be compared. A good gesture transcription algorithm should produce similar transcriptions. Exactly equal transcriptions should *not* be expected since the algorithm involves a component of subjective interpretation of gesticulation's meaning.

C.6 References

[1] D. McNeill; *Hand and Mind: What gestures reveal about thought*; The University of Chicago Press; 1992

[2] Duncan, S.; *McNeill Lab Coding Methods*

[3] McNeill, D.; *Gesture and Thought*; University of Chicago Press; 2005

[4] Gut, U.; Looks, K.; Thies, A.; Trippel, T.; Gibbon, D.; *CoGesT – Conversational Gesture Transcription System*, Technical Report; University of Bielefeld; 2003

[5] *VirtualDub Homepage*; www.virtualdub.org/

[6] *Adobe Premiere Pro Homepage*; www.adobe.com/products/premiere/main.html

[7] *PRAAT Homepage*; www.fon.hum.uva.nl/praat/

van Lieshout, P.; *PRAAT Short Tutorial*; 2004

[8] Kipp, M.; *ANVIL – A Generic Annotation Tool for Multimodal Dialogue* in *Proceedings of the 7th European Conference on Speech Communication and Technology, Aalborg, pp.1367-1370*; 2001

[9] *StatSoft's STATISTICA Homepage*; www.statsoftinc.com/index.htm

[10] *SPSS Homepage*; www.spss.com/

[11] Secretariado Nacional para a Reabilitação e Integração das Pessoas com Deficiência; *Gestuário - Língua Gestual Portuguesa* – 5th edition.

[12] Akenine-Möller, T.; Haines, E.; *Real-time Rendering*, second edition; A K Peters; 2002

Appendix D – Expression Markup Language

Abstract

This document specifies *Expressive Markup Language (EML)*, an integrated synchronized high-level language which supports virtual human multimodal expression control. This language can be used, essentially, in two ways: (1) as an *interface for a mind* which requires real-time synchronous and multimodal expression through a body; (2) as a *script* defining a story, written by a human or digital author, in real time or not, in which virtual humans express multimodally. In the first case, the mind communicates with the body in real time through a set of EML clauses which are to be executed immediately. In the second case, a story is defined as sequence of clauses, ordered in time, which can be played repeatedly at different times and through different virtual humans.

EML is a markup language which can be structured into five modules: (1) *Core* – which defines the main elements; (2) *Time and Synchronization* – which defines synchronization mechanisms between modalities; (3) *Body Expression Markup Language (BEML)* – which controls both deterministic and non-deterministic body expression; (4) *Vocal Expression Markup Language (VEML)* – which controls vocal expression; (5) *Gesture Expression Markup Language (GEML)* – which controls gesticulation expression.

Contents

D.1 Core

This module defines the language's main elements including, among others, the following: eml, the root element; head, which defines configuration clauses; body, which defines sequences of virtual human expression clauses.

D.1.1 eml

specification	
Root element.	
attribute	**value**
`name`	The document's name.
example	

```
<eml name='myProject'>
...
</eml>
```

D.1.2 head

specification
Expression configuration parameters. Descends from `eml`.
example

```
<head>
      ...
</head>
```

D.1.3 body

specification
The project body. Descends from `eml`.
example

```
<body>
      ...
</body>
```

D.1.4 event

specification	
Raises an event. Events are categorised according to *types*. This supports selective listening by event receivers. Character data is passed on with the event. Descends from `body`, `seq` or `par`.	
attribute	**value**

type	Event type is an arbitrary string. Optional.
example	
<event key=’alarm’><![CDATA[<shout>wake up!</shout>]]></event>	

D.1.5 null

specification
Does nothing. Can be used to force a script to run for a certain duration, for instance. Descends from body, seq or par.
example
<null time=’5’/>

D.1.6 vh-move

specification	
Changes the position and/or orientation of the virtual human, according to a function. Descends from body, seq or par.	
attribute	**value**
functionName	The function name.
isLooping	Whether to consider periodic time. Optional. Default value is false.

example

```
<vh-move time='5'
 functionName='segment' isLooping='false'>
 <function-args>
  <arg name='timePeriod' value='4.0' />
  <arg name='startX' value='32.5' />
  <arg name='startY' value='82.5' />
  <arg name='startZ' value='12.5' />
  <arg name='endX' value='32.5' />
  <arg name='endY' value='82.5' />
  <arg name='endZ' value='-32.5' />

  <arg name='startYaw' value='0' />
  <arg name='startPitch' value='180' />
  <arg name='startRoll' value='-45' />
  <arg name='endYaw' value='90' />
  <arg name='endPitch' value='180' />
  <arg name='endRoll' value='0' />
 </function-args>
</vh-move>
```

D.1.7 function-args

Function arguments. Descends from vh-move or body-non-deterministic-function.

D.1.8 arg

Function argument. Descends from function-args.

D.2 Time and Synchronization

This module defines synchronization mechanisms between modalities. This module is characterized as follows: (1) supports execution time definition relative to other clauses; (2) supports execution time definition relative to a word or phoneme time in a vocal expression clause; (3) supports loops; (4) supports parallel and sequential execution. This module is based on W3C's SMIL 2.0 specification [1].

D.2.1 time (attribute)

specification
Defines the clause execution time.
example
`<some-clause time=’5.0’ />`

D.2.2 timeId (attribute)

specification
Defines a named timestamp which can be referred to by other clauses. This attribute is optional.
example
`<some-clause time=’5.0’ timeId=’t1’ />` `<other-clause time=’t1+1.0’ />`

D.2.3 Time expression examples

Complex time expressions can be defined using numeric time and named timestamps. The following examples illustrate the rules for expression construction:

```
<!-- 5 seconds, seconds is default time unit -->
<c1 time='5.0' timeId='t1' />
<!-- 6 seconds, time prefixes: s-seconds; m-minutes; h-hours -->
<c2 time='t1+1s' />
<!-- 8 seconds, supported math operations: sum, aubtraction -->
<c3 time='t1+4s-1s' />
<!-- 10 seconds, arbitrary combination order -->
<c4 time='5+t1' />
```

D.2.4 seq

specification
Defines a set of clauses to be executed in sequence. Sequence clauses have relative times.
example

`<seq time=’1.0’>`
`<some-clause time=’0’ />`
`<other-clause time=’1’ />`
`</seq>`

D.2.5 par

specification
Defines a set of clauses to be executed in parallel. Set clauses have relative times.
example

```
<par time=’1.0’>
 <some-clause time=’0’ />
 <other-clause time=’1’ />
</par>
```

D.2.6 repetitionCount & duration (attributes)

specification
Repeats a clause a certain number of times. Valid only for seq and par clauses. Duration defines the iteration duration. These attributes are optional.

example

```
<seq time=’1.0’ repetitionCount=’2’ duration=’2’>
 <some-clause time=’0’ />
 <other-clause time=’1’ />
</seq>
```

D.3 BEML – Body Expression Markup Language

D.3.1 body-expression-configuration

Body expression configuration parameters. Descends from `head`.

D.3.2 sets-configuration

This element encapsulates *set* definitions. Descends from `body-expression-configuration`.

D.3.3 set, it, ref

specification
A *set* is either a collection of layers, of body group or other layers. This element defines a named set of tracks. Descends from `tracks-configuration`.

attribute	value
`name`	The set’s name.
`type`	The set type. Possible values: (1) *multiple*; (2) *bodyGroup*.

example

```
<body-expression-configuration>
   <sets-configuration>
       <set type='bodyGroup' name='upperBody'>
            <it name='head' />
            <it name='torso' />
            <it name='left_arm' />
            <it name='right_arm' />
            <it name='left_hand' />
            <it name='right_hand' />
       </set>
       <set type='bodyGroup' name='fullBody'>
            <ref name='upperBody' />
            <ref name='lowerBody' />
       </set>
   </sets-configuration>
```

```
 ...
</body-expression-configuration>
```

D.3.4 body-configure-geometry

specification	
Sets geometry properties. Descends from body, seq or par.	
attribute (*)	**value**
fillMode	Rendering fill mode: Possible values: (1) *solid*; (2) *wireframe*; (3) *point*
areFramesVisible	Whether the skeleton's frames are visible.
areHierarchyCentersVisible	Whether the hierarchy centers are visible
visibleMeshes	The visible meshes. Possible values: (1) *body*; (2) *bones*; (3) *both*
areMaterialsActive	Whether the materials are to be rendered.
areTexturesActive	Whether the textures are to be rendered.
areManipulatorsVisible	Whether the manipulators are to be rendered.

example

```
<body-configure-geometry time='0'
fillMode='solid' visibleMeshes='bones'/>
```

(*) All attributes are optional. If not set, the property is not assigned.

D.3.5 body-configure-deterministic-animation

specification	
Sets deterministic animation global properties. Descends from body, seq or par.	
attribute	**value**
smooth	Whether animation transition should be smooth.
swapTime	Animation transition duration in seconds.

example

```
<body-configure-deterministic-animation time='0'
 smooth='false' swapTime='0.5' />
```

D.3.6 body-configure-deterministic-track-property

specification	
Sets a deterministic animation's layer or body group property. Descends from body, seq or par.	
attribute	**value**
property	The track property. Possible values: (1) *speed* (transitable); (2) *weight* (transitable); (3) *enable*
newValue	The property's new value.
player	The animation player. Defaults to the first player.

duration	Transition duration. Only applies to transitable properties.
transition	Transition type. Possible values: (1) *linear*

example

```
<body-configure-deterministic-track-property time='2.5'
      property='speed' newValue='0.5'
      duration='0' transition='linear'>
      <tracks><ref name='fullBody' ></ref></tracks>
</body-configure-deterministic-track-property>
```

D.3.7 tracks

A list of tracks. Descends from `deterministic-track-property` or `body-deterministic-primitive`.

D.3.8 ref

Reference to named set. Descends from `tracks`.

D.3.9 bg

Refers to a body group. Descends from `tracks`.

D.3.10 ly

Refers to a layer. Descends from `tracks`.

D.3.11 body-player-add

specification

Adds an animation player. Descends from `body`, `seq` or `par`.

attribute	value
name	The animation player. Defaults to the first player.
type	The player type. Currently only supports *multiple* and *bodyGroup*.
numberOfLayers	Desired number of layers. Only applies if player type is *multiple*.

example

```
<body-player-add
 time='0' name='mPlayer' type='multipleAnimation' numberOfLayers='2'>
 <bones><ref name='fullBody' /></bones>
</body-player-add>
```

D.3.12 body-player-set-exclusive

specification

Sets an exclusive animation player. Descends from `body`, `seq` or `par`.

attribute	value
name	The animation player. Defaults to the first player.
type	The player type. Currently only supports *multiple* and *bodyGroup*.
numberOfLayers	Desired number of layers. Only applies if player type is *multiple*.

example
`<body-player-set-exclusive time='0' name='mPlayer' type='multipleAnimation' numberOfLayers='2'> <bones><ref name='fullBody' /></bones> </body-player-set-exclusive>`

D.3.13 bones, ref, bg, bone

Defines the bones affected by an animation player. Uses the following elements: *ref* which refers to a named set; *bg* which refers to a body group; *bone* which refers to a particular bone. Descends from `body-player-add` or `body-player-set-exclusive`.

D.3.14 body-player-change-priority

specification	
Changes an animation player's priority. Descends from `body`, `seq` or `par`.	
attribute	**value**
`name`	The animation player. Defaults to the first player.
`priority`	The new priority value.
example	
`<body-player-change-priority time='5.0' name='mPlayer' priority='10' />`	

D.3.15 body-player-remove

specification	
Removes a particular animation player. Descends from `body`, `seq` or `par`.	
attribute	**value**
`name`	The animation player. Defaults to the first player.
example	
`<body-player-remove time='7.0' name='mPlayer' />`	

D.3.16 body-player-clear

specification
Clears all animation players. Descends from `body`, `seq` or `par`.
example
`<body-player-clear time='15.0' />`

D.3.17 body-deterministic-primitive

specification	
Invokes a deterministic animation operation. Descends from `body`, `seq` or `par`.	
attribute	**value**
`primitive`	The deterministic animation primitive. Possible values: (1) *push;* (2) *exchange*; (3) *pop*; (4) *clearAnimationStack*

player	The animation player. Defaults to the first player.
animationSetName	The deterministic animation.

example

```
<body-deterministic-primitive time='0.1'
operation='push' animationSetName='stepStance'>
      <tracks><ref name='fullBody' ></ref></tracks>
      <animation-style playbackStyle='loop'></animation-style>
</body-deterministic-primitive>
```

D.3.18 animation-style

specification

Defines a deterministic animation style. Descends from `body-deterministic-primitive`.

attribute	value
playbackStyle	The playback style. Possible values: (1) *loop*; (2) *loopReverse*; (3) *pingPong*; (4) *iterations*; (5) *iterationsReverse*; (6) *temporaryIterations*; (7) *temporaryIterationsReverse*

D.3.19 body-configure-manipulator

specification

Sets manipulator's properties. Descends from `body`, `seq` or `par`.

attribute	value
manipulator	The manipulator.
isOn	Whether the manipulator is active.
jointLimitsActive	Whether the manipulator's joint limits are active.

example

```
<body-configure-manipulator time='0'
 manipulator='leftArm' isOn='true' jointLimitsActive='false' />
```

D.3.20 body-non-deterministic-fk

specification

Forward kinematics primitive. Descends from `body`, `seq` or `par`.

attribute	value
manipulator	The manipulator.
nJoints	The manipulator's number of joints.
j1...jn	One attribute per joint. The attribute represents the intended value, in degrees, for the joint.

example

```
<body-non-deterministic-fk time='0'
 manipulator='leftArm' nJoint='6'
 j1='90.0' j2='45.0' j3='0.0' j4='0.0' j5='0.0' j6='0.0' />
```

D.3.21 body-non-deterministic-joint-interpolation

specification	
Joint interpolation inverse kinematics primitive. Descends from `body`, `seq` or `par`.	
attribute	**value**
`manipulator`	The manipulator.
`staticTime`	Interpolation time, in seconds. Also, defines the interpolation type to *static time*.
`staticVelocity`	Interpolation velocity, in rad/sec. Also, defines the interpolation type to *static velocity*.
`posX`	Target position X coordinate.
`posY`	Target position Y coordinate.
`posZ`	Target position Z coordinate.
`yaw`	Target orientation yaw angle, in degrees.
`pitch`	Target orientation pitch angle, in degrees.
`roll`	Target orientation roll angle, in degrees.

example

```
<body-non-deterministic-joint-interpolation time='8'
 manipulator='leftLeg' staticTime='2.0'
 posX='7.5' posY='20' posZ='-10'
 yaw='0' pitch='0' roll='-90' />
```

D.3.22 body-non-deterministic-frame-interpolation

specification	
Frame interpolation inverse kinematics primitive. Descends from `body`, `seq` or `par`.	
attribute	**value**
`manipulator`	The manipulator.
`timePeriod`	Interpolation time, in seconds.
`posX`	Target position X coordinate.
`posY`	Target position Y coordinate.
`posZ`	Target position Z coordinate.
`yaw`	Target orientation yaw angle, in degrees.
`pitch`	Target orientation pitch angle, in degrees.
`roll`	Target orientation roll angle, in degrees.

example

```
<body-non-deterministic-frame-interpolation time='8'
 manipulator='leftLeg' timePeriod='2.0'
 posX='7.5' posY='20' posZ='-10'
 yaw='0' pitch='0' roll='-90' />
```

D.3.23 body-non-deterministic-function

specification	
Function based interpolation inverse kinematics primitive. Descends from `body`, `seq` or `par`.	
attribute	**value**
`manipulator`	The manipulator.
`functionName`	The function name.
`isLooping`	Whether to consider periodic time.

example

```
<body-non-deterministic-function time='44'
 manipulator='leftArm' functionName='segment' isLooping='false'>
 <function-args>
  <arg name='timePeriod' value='4.0' />
  <arg name='startX' value='32.5' />
  <arg name='startY' value='82.5' />
  <arg name='startZ' value='12.5' />
  <arg name='endX' value='32.5' />
  <arg name='endY' value='82.5' />
  <arg name='endZ' value='-32.5' />
  <arg name='startYaw' value='0' />
  <arg name='startPitch' value='180' />
  <arg name='startRoll' value='-45' />
  <arg name='endYaw' value='90' />
  <arg name='endPitch' value='180' />
  <arg name='endRoll' value='0' />
 </function-args>
</body-non-deterministic-function>
```

D.3.24 body-non-deterministic-jacobian

specification	
Jacobian based inverse velocity primitive. Descends from `body`, `seq` or `par`.	
attribute	**value**
`manipulator`	The manipulator.
`isStaticTarget`	True, if based on a static target. False, if based on a cartesian and angular velocities vector.
`posX`	Static target position X coordinate.
`posY`	Static target position Y coordinate.
`posZ`	Static target position Z coordinate.
`yaw`	Static target orientation yaw angle, in degrees.
`pitch`	Static target orientation pitch angle, in degrees.
`roll`	Static target orientation roll angle, in degrees.
`cartesianX`	Cartesian velocity X component.
`cartesianY`	Cartesian velocity Y component.
`cartesianZ`	Cartesian velocity Z component.
`angularX`	Angular velocity X component, in degrees/sec.

angularY	Angular velocity Y component, in degrees/sec.
angularZ	Angular velocity Z component, in degrees/sec.

example

```
<body-non-deterministic-jacobian time='8'
 manipulator='leftLeg' isStaticTarget='true'
 posX='7.5' posY='20' posZ='-10'
 yaw='0' pitch='0' roll='-90' />

<body-non-deterministic-jacobian time='8'
 manipulator='leftLeg' isStaticTarget='false'
 cartesianX='1.0' cartesianY='0' cartesianZ='0'
 angularX='45.0' angularY='45.0' angularZ='45.0' />
```

D.3.25 body-non-deterministic-stop

specification

Stops a manipulator. Descends from `body`, `seq` or `par`.

attribute	value
manipulator	The manipulator.

example

```
<body-non-deterministic-stop time='0' manipulator='leftArm' />
```

D.3.26 body-non-deterministic-lock

specification

Locks/unlocks a manipulator to/from the current target. Descends from `body`, `seq` or `par`.

attribute	value
manipulator	The manipulator.
isLock	Whether to lock(true) or unlock(false) the manipulator.

example

```
<body-non-deterministic-lock time='0'
 manipulator='leftArm' isLock='true' />
```

D.3.27 body-set-control-parameter

specification

Sets a control parameter's value. Descends from `body`, `seq` or `par`.

attribute	value
parameterId	The parameter identification string.
value	The parameter value.

example

```
<body-set-control-parameter time='0.0'
      parameterId='Disgust' value='1.0'/>
```

D.3.28 body-animate-ADSR-control-parameter

specification	
Plays a four-phase (Attack, Delay, Sustain, Release) control parameter animation. Descends from `body`, `seq` or `par`.	
attribute	**value**
`parameterId`	The parameter identification string.
`attackDurationTime`	Attack phase duration time in seconds.
`attackValue`	Parameter value during the attack phase.
`decayDurationTime`	Decay phase duration time in seconds.
`decayValue`	Parameter value during the decay phase.
`sustainDurationTime`	Sustain phase duration time in seconds.
`sustainValue`	Parameter value during the sustain phase.
`releaseDurationTime`	Release phase duration time in seconds.
`releaseValue`	Parameter value during the release phase.
example	
`<body-set-control-parameter time='0.0' parameterId='Disgust' value='1.0'/>`	

D.3.29 body-load-control-parameter-animation

specification	
Loads a control parameter animation into the library. Descends from `body`, `seq` or `par`.	
attribute	**value**
`file`	The file path.
`isResourcesRelative`	Whether the file path is relative to the ‘resources’ directory.
example	
`<body-load-control-parameter-animation time='5.0' file=’visemes-1.xml' isResourcesRelative='true'/>`	

D.3.30 body-animate-control-parameter

specification	
Plays a control parameter animation from the library. Descends from `body`, `seq` or `par`.	
attribute	**value**
`animationId`	The animation identification string.
`style`	The playback style. Possible values are: (1) *forward*; (2) *reverse*; (3) *pingPong*; (4) *reversePingPong*
`value`	The parameter value.
example	
`<body-set-control-parameter time='0.0' parameterId='Disgust' value='1.0'/>`	

D.4 VEML – Vocal Expression Markup Language

D.4.1 voice-text

specification
Root tag for text to speech. Represents SABLE's `sable` tag. Supports time markers. Descends from `body`, `seq` or `par`.
example
`<voice-text time='1.0'>Hello world!</voice-text>`

D.4.2 voice-say

specification	
Synthesizes speech from plain text or a file. Does not support time markers. Descends from `body`, `seq` or `par`.	
attribute	**value**
`voiceOperation`	The voice operation to perform. Possible values: (1) *text*; (2) *file*; (3) *preprocessText*; (4) *preprocessFile*; (5) *asynchronousText*; (6) *asynchronousFile*.
`type`	Utterance type. Possible values: (1) *text*; (2) *sable*.
example	
`<voice-say time='1.0' voiceOperation='preprocessFile' type='text'> miscellaneous/poetry/osLusiadas.txt</voice-say>`	

D.4.3 tm

specification	
Sets a named timestamp within the text. The timestamp is associated to an event related to the next word. Descends from any SABLE container tag.	
attribute	**value**
`name`	The timestamp name
`event`	Possible values: (1) *onStart* – associate timestamp to start time of next word; (2) *onEnd* – associate timestamp to end time of next word; (3) *onPhone* – associate timestamp to start time of particular phone in word Default is *onStart*.
`phone`	The ascii representation of the phone to associate the word with. Should only be defined if event is *onPhone*.
example	
`<voice-text time = '1.0'> <tm name='t1'/>Hello <tm name='t2' event='onPhone' phone='hh'/>world! </voice-text>`	

D.4.4 audio

specification

Load and play an audio URL. Represents SABLE's `audio` tag. Descends from any SABLE container tag.	
attribute	**value**
`src`	URL of a document with an appropriate mime-type
`mode`	Possible values: (1) *background;* (2) *insertion*
`level`	A positive number: 1.0 is same as the original audio, 0.0 is silent
example	

```
<voice-text time = '1.0'>
 <audio src="http://www.music.com/trackX.au"/>
</voice-text>
```

D.4.5 break

specification	
Sets an intrasentential, prosodic break at current position. Represents SABLE's `break` tag. Descends from any SABLE container tag.	
attribute	**value**
`level`	Defines the break level. Can be numeric or discrete. Possible discrete values: *large, medium, small, none*
`msec`	A number greater than zero defining the length of the pause associated with this break
`type`	A punctuation symbol that represents the kind of intonation contour. Possible values: *"?", for question; "!", for exclamation; ".", for a statement; ",", for the impression that more is coming*
example	

```
<voice-text time = '1.0'>
Two roads diverged in a yellow wood,<break level="small"/>
And sorry I could not travel both
</voice-text>
```

D.4.6 div

specification	
Classifies the contained region as a division of a certain type. Represents SABLE's `div` tag. Descends from any SABLE container tag.	
attribute	**value**
`type`	Type of division. Possible values: (1) *sentence;* (2) *paragraph*
example	

```
<voice-text time = '1.0'>
<div>
Two roads diverged in a yellow wood,
And sorry I could not travel both
</div>
</voice-text>
```

D.4.7 emph

specification	
Sets an emphasis. Represents SABLE's `emph` tag. Descends from any SABLE container tag.	
attribute	**value**
`level`	A number greater than 0.0 or one of the following descriptive values: *strong, moderate, none, reduced*
example	

```
<voice-text time = '1.0'>
<emph>Two roads</emph> diverged in a yellow wood,
</voice-text>
```

D.4.8 engine

specification	
Engine specific content. Represents SABLE's `engine` tag. Descends from any SABLE container tag.	
attribute	**value**
`id`	Identifier for the specific TTS engine
`data`	Any string to be substituted for the contained context
example	

```
<voice-text time = '1.0'>
An example is <engine id="festival" data="our own festival speech
synthesizer"> the festival speech synthesizer</engine> or
the Bell Labs speech synthesizer.
</voice-text>
```

D.4.9 pitch

specification	
Sets pitch properties. Represents SABLE's `pitch` tag. Descends from any SABLE container tag.	
attribute	**value**
`base`	Sets the base line of the intonation as a percentage relative to the current value, a number, or one of the following values: *highest, high, medium, low, lowest, default*
`middle`	Sets the middle line of the intonation as a percentage relative to the current value, a number, or one of the following values: *highest, high, medium, low, lowest, default*
`range`	Sets the range of the intonation as a percentage relative to the current value, a number, or one of the following values: *highest, high, medium, low, lowest, default*
example	

```
<voice-text time = '1.0'>
Without his penguin, <pitch base="-30%"> which he left at home,
</pitch>he could not enter the restaurant.</voice-text>
```

D.4.10pron

specification	
Defines a specialized pronunciation. Represents SABLE's `pron` tag. Descends from any SABLE container tag.	
attribute	**value**
`ipa`	String in Unicode IPA describing the pronunciation
`sub`	String representing an attemp at "phonetic" spelling
`origin`	Identifier for the language of origin of the enclosed text
example	

```
<voice-text time = '1.0'>
Homographs are words that are written the same but have different
pronunciations, such as <pron sub="lyves">lives</pron> and
<pron sub="lives">lives</pron>.</voice-text>
```

D.4.11rate

specification	
Sets the speech rate. Represents SABLE's `rate` tag. Descends from any SABLE container tag.	
attribute	**value**
`speed`	Sets a words-per-minute speed, as a percentage relative to the current value, or a descriptive speed. Possible values: *fastest; fast; medium; slow; slowest*
example	

```
<voice-text time = '1.0'>
Two roads <rate speed="-40%"> diverged in a yellow wood,
</rate>.</voice-text>
```

D.4.12sayas

specification	
Defines a way in which the contained region is to be said. Represents SABLE's `rate` tag. Descends from any SABLE container tag.	
attribute	**value**
`mode`	Possible values: *literal, date, time, phone, net, postal, currency, math, fraction, measure, ordinal, cardinal, name*
`modetype`	Possible values: *DMY, MDY, YMD, YM, MY, MD, HM, HMS, EMAIL, URL*
example	

```
<voice-text time = '1.0'>
As a test of marked-up numbers. Here we have
a year <sayas mode="date">1998</sayas>,
or an ordinal <sayas mode="ordinal">1998</sayas></voice-text>
```

D.4.13volume

specification
Set the volume for the contained text. Represents SABLE's `rate` tag. Descends from any

SABLE container tag.	
attribute	**value**
`level`	Defines a numeric amplitude level, or the amplitude level as a percentage relative to the current value, or as one of the following descriptive values: *loudest; loud; medium; quiet*
example	

`<voice-text time = '1.0'>`
`Two roads <volume discreteLevel="quiet"> diverged in a yellow wood </volume>.`
`</voice-text>`	

D.4.14voice-say

specification	
Invokes the `voice-say` primitive. Descends from `body`, `seq` or `par`.	
attribute	**value**
`voiceOperation`	The voice operation. Possible values: *text, file, preprocessText, preprocessFile, AsynchronousText, AsynchronousFile*
`type`	Utterance type. Possible values: *text, sable*
`isResourcesRelativeFilePath`	Whether the file path is relative to the `{HOME/resources}` directory
example	

`<voice-say time='7'`
` voiceOperation='preprocessFile' type='text'>`
`miscellaneous/poetry/osLusiadas.txt</voice-say>`	

D.4.15voice-change

specification	
Changes the virtual human voice. Descends from `body`, `seq` or `par`.	
attribute	**value**
`voice`	The voice name
example	
`<voice-change time='1.0' voice='don_diphone' />`	

D.5 GEML – Gestures Expression Markup Language

This module defines a set of gesture expression control clauses. Gesture expression can be interpreted as specialized body expression through the arms and hands.

D.5.1 gesture-is-on

specification
Sets the gesture model activation status. Descends from `body`, `seq` or `par`.

attribute	value
isOn	Whether to active the gesture model.
example	
`<gesture-is-on time='0' isOn='true' />`	

D.5.2 gesture-key

specification
Executes a *gesture key*. A gesture key is defined according to a sequence of hand shapes, a sequence of palm orientations and a sequence of positions. Descends from `body`, `seq` or `par`.

attribute	value
duration	The gesture key duration in seconds.
handedness	Defines which hands are used. Possible values: (1) both; (2) left; (3) right

D.5.3 hand-shapes, key

specification	
Defines a sequence of portuguese sign language hand shapes. `hand-shapes` descends from `gesture-key` and key descends from `hand-shapes`.	
attribute (hand-shapes)	**value**
hand	Affected hand. Possible values: (1) *left*; (2) *right*. If ommited, both hands are affected.
attribute (key)	**value**
time	Time offset to assume the shape. Optional.
id	Portuguese Sign Language hand shape id.

example

```
<gesture-key time='0.0' duration='3' handedness='right'>
      <hand-shapes>
            <key id='24'/>
      </hand-shapes>
</gesture-key>
```

D.5.4 palms, key

specification	
Defines a sequence of palm orientations. palms descends from `gesture-key` and `key` descends from `palms`.	
attribute (hand-shapes)	**value**
hand	Affected hand. Possible values: (1) *left*; (2) *right*. If ommited: (a) both hands are affected; (b) values should be specified for dominant hand; (c) symmetry is automatically applied for non-dominant hand
attribute (key)	**value**
time	Time offset to assume the shape. Optional.

id	Portuguese Sign Language hand shape id.

example

```
<gesture-key time='3.0' duration='3' handedness='right'>
      <palms>
            <key a='90' x='1.0' y='0.0' z='0.0'/>
      </palms>
</gesture-key>
```

D.5.5 motion, key

specification

Defines a sequence of positions in space. motion descends from `gesture-key` and `key` descends from `motion`.

attribute (hand-shapes)	value
hand	Affected hand. Possible values: (1) *left*; (2) *right*. If ommited: (a) both hands are affected; (b) values should be specified for dominant hand; (c) symmetry is automatically applied for non-dominant hand
coords	Whether the coordinates are relative to the world or speaker frames. Possible values: (1) *world*; (2) *speaker*
attribute (key)	**value**
time	Time offset to assume the shape. Optional.
id	Portuguese Sign Language hand shape id.

example

```
<gesture-key time='6.0' duration='3' handedness='right'>
      <motion coords='world' >
            <key x='-30.0' y='70.0' z='10.0'/>
      </motion>
</gesture-key>
```

References

[1] SMIL; *SMIL: Synchronized Multimedia*; http://www.w3.org/AudioVideo/