

# Formational Parameters and Adaptive Prototype Instantiation for MPEG-4 Compliant Gesture Synthesis

Björn Hartmann  
Center for Human Modeling and Simulation  
University of Pennsylvania  
bjoern@graphics.cis.upenn.edu

Maurizio Mancini, Catherine Pelachaud  
Department of Computer and System Science  
University of Rome “La Sapienza”  
cath@dis.uniroma1.it

## Abstract

*This paper introduces Gesture Engine, an animation system that synthesizes human gesturing behaviors from augmented conversation transcripts using a database of high-level gesture definitions. An abstract scripting language to specify hand-arm gestures is introduced that incorporates knowledge from sign language research, psycholinguistics, and traditional keyframe animation. A new planning algorithm instantiates and adjusts gestures according to communicative context and temporal constraints obtained from a speech synthesizer. The system animates an MPEG-4 compliant skeleton using Body Animation Parameters.*

## 1 Introduction

The main objective of our work is the creation of an embodied conversational agent [4] and in particular the creation of an agent that is able to exhibit communicative gestures while talking. To this end we are applying the metaphor of human-human communication to drive the verbal and nonverbal behaviors of our agent. Human communication is very rich and complex – we routinely and effortlessly combine verbal and nonverbal channels to express ourselves: we may express our emotion with our voice [32] or our face [12]; our gaze direction may indicate when we want to take our turn to speak or when we want to pass [10, 7]; gestures in turn may indicate the shape of an object or accentuate a particular word; they may replace a word or indicate a point in space [15, 22, 27]. Several researchers [22, 14, 5] have shown the subtle link between the production of a gesture and the accompanying speech. Gesture is not just a mere translation of the verbal discourse. Among other functions, gesture may complement information (in a noisy bar we indicate to the barkeeper with our fingers how many drinks we desire); it may help us to process our thought; it may show how certain we are of what we are

saying – or how uncertain (when we raise our palms with open hands).

The work described in this paper is part of a larger project, MagiCster, a 3-year long venture supported by the European Union<sup>1</sup>, which aims to create a believable embodied agent. The first prototype focuses on an information delivery application. The agent, named Greta, is able to converse with a user. The types of dialogue we are focusing on currently are of query-answer form; the user asks for information on a given domain and the agent replies to the request. This gives rise to simple sub-dialogue exchanges. In previous work we have been concentrating on the creation of a facial model compliant to MPEG-4 [25] as well as on the communicative aspects of facial expressions and gaze [29]. In this paper we are turning our attention to the specification and the animation of gestures. We have developed a body model compliant to MPEG-4 specification and we have elaborated a language to describe gesture. Indeed, gestures may exhibit very complex motions and hand movements. Describing a gesture through a set of joint angles is very cumbersome, tedious, and non-intuitive. We have decided to use the formational approach developed by Stokoe in his pioneering work on the structure of sign language [35]. He describes gestures using a set of so-called formational parameters [28, 34]; a gesture is made up of a combination of several elements, namely the hand shape, the wrist orientation, but also the arm movement and the place of articulation [34]. We base our gesture definition on such an approach.

In the following section, we provide an overview of the literature, specifying how our method differs from previous ones. We then present our body model. Section 4 presents our language specification for gesture and an editor that has been developed to interactively create new gestures for the

---

<sup>1</sup>IST project IST-1999-29078, partners: University of Edinburgh, Division of Informatics; DFKI, Intelligent User Interfaces Department; Swedish Institute of Computer Science; University of Bari, Dipartimento di Informatica; University of Rome, Dipartimento di Informatica e Sistemistica; AvatarME

system. The paper continues by describing the motion planning algorithm. Finally we present future work, specifically looking at how such an engine may be used to create communicative gesture in a conversational setting.

## 2 Related Work

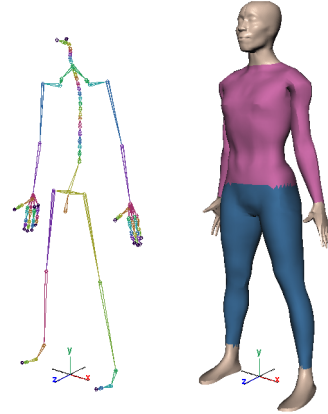
Gesture Engine is an integrated gesture specification and animation system with several distinguishing features: its database is easily extensible by lay users in an intuitive way; it instantiates gestures according to communicative context and it synchronizes its output to speech and possibly other non-verbal modalities (such as facial expression and gaze pattern). Furthermore, the employed gesture notation language allows for the representation of complex articulated joint structures with a small number of abstract parameters. While some authors have previously addressed the individual issues presented here and others have proposed complete multimodal communication architectures of considerable complexity, we still believe Gesture Engine to be a valuable contribution to the field since it represents a unique synthesis of, and extension to, modern gesture animation concepts. We now review some of the principal works undertaken in this area thus far.

Cassell et al. [6] animate conversational agents' dialogues by automatically generating facial expression and hand gestures synchronized with speech. We follow their approach insofar as we also separate gesture generation into independent arm, wrist and hand specifiers and we also store gestures in a database of predefined prototypes. However, we extend their work with an improved scheduling and instantiation algorithm that integrates beats with other gestures more closely and introduces flexible transition states.

Lebourque and Gibet [20, 21], use a sign-language derived coding scheme with an inverse-kinematics-based dynamics model for gestures but they do not attempt to integrate synthesized gestures into a multi-modal communication context. Our results also show that a less complex key-frame and joint-angle based animation procedure can work very well in the restricted domain of gesturing.

Kopp and Wachsmuth [17, 18] present a hierarchical planning architecture based on experimentally derived knowledge. While our architecture is partly modeled after their system, our planning algorithm focuses on the linking and context-specific adaptation of gestures and not on trajectory formation. None of the above articles addresses the issue of user-friendly maintenance and extension of the database of known and possible gestures.

Recently, the association between discourse structure and gesture generation was discussed by Cassell et al. in the BEAT model [8]. In addition, the authors here do address the importance of user extensibility; however, the BEAT system represents only an intermediate building block that



**Figure 1. Gesture Engine's H-Anim skeleton and the associated body mesh**

needs to communicate with other elements to form a complete agent control and animation system. Another comprehensive system of abstract representation of agent actions is presented in PAR [2]. While its generality is very powerful, it also depends on an external gesture and animation model to execute actions.

Most existing agent systems use proprietary visualization systems custom-tailored to their specific research environments. We believe such practice to be counter-productive to the proliferation of conversational agents in their function as communicators interacting with humans. Device-independent, distributable animation files promise to reach a much larger end-consumer audience. The MPEG-4 standard for facial and corporeal animation is a good current working platform to accomplish this goal. MPEG-4 compliant facial animations using Facial Animation Parameters (FAPs) have been demonstrated in our own previous efforts [25] as well as in [24]. Less work is extant on animating the body: a Body Animation Parameter (BAP) player has been implemented [1] and some animations have been generated [30], but utilization beyond proofs-of-concept appears to be sparse.

## 3 Body Model

Before we discuss details of the Gesture Engine architecture, it is of utility to introduce the employed body model, since we will refer to it in the following sections to illustrate underlying ideas and technical concepts. A fully articulated humanoid skeleton (see Figure 1), as defined in the H-Anim 1.1 Specification [40], forms the basis of our agent. The current polygonal body mesh model is taken from Alias|Wavefront's Maya™ 3.0 suite, but other mod-

els fitting the underlying skeletal proportions can be substituted. Note that at the current time our animation system is somewhat model-specific. It is possible but not trivial to transfer animations to agents of different anthropometric proportions, as the pose library defining goal positions and finger configurations has to be adapted (see Section 5.2). Techniques such as motion mapping[3] or motion retargeting [13] could be used to overcome this problem. However, the gesture definitions introduced in Section 4.2 can remain unaltered since they only refer to abstract reach space and hand configurations.

Gesture Engine updates the skeleton's joint angles for each animation frame using standardized MPEG-4 BAPs [36]. The skeleton in turn deforms a unified polygonal mesh. We have developed two animation playback solutions to demonstrate the portability of our chosen format - an internal visualization system within Gesture Engine implemented in OpenGL and an external BAP player for Alias|Wavefront's Maya<sup>TM</sup>, for which we can export ASCII-format BAP files for one conversational turn at a time<sup>2</sup>. In Maya<sup>TM</sup>, the agent's body mesh is bound to the skeleton using the built-in Rigid Skinning feature set. In our own OpenGL implementation, mesh vertices are partitioned into skin clusters that are transformed as children of individual joints in the skeletal hierarchy.

## 4 Gesture Specification

### 4.1 Communicative Acts

As mentioned in the introduction, gestures may have several roles in a conversation. In previous work we have developed a taxonomy of communicative functions for facial expressions and gaze [29]. It is our view that this taxonomy can be extended to gestures as well [28]. Indeed, gestures may provide information on the agent's beliefs, intentions, and her affective state. They may provide information on the world or they may convey meta-information on the agent's mental state. Five classes are present in this taxonomy that provide information on:

- location and properties of concrete or abstract objects or events
- the agent's beliefs
- her intentions
- her affective state
- metacognitive information on her mental actions

The first class includes deixis, which is the indication of the relative spatial location of referents ('this box over there'),

---

<sup>2</sup>A conversational turn is defined to be the interval from the time the agent takes over discourse control from its dyadic partner to the time she relinquishes it again.

and information on physical or metaphorical properties of referents ('this box is round'). The second class is representative of the degree of certainty with which the agent believes what she is saying ('palm up open hand' may indicate certainty [23] as cited in [28]). The third cluster gathers gestures that are intended to express a goal of the agent: the performative of her sentence (e.g., 'I order you' may be indicated with an angry finger pointed toward the addressee vs. 'I apologize', which may be marked with pulling back a flat hand raised near the shoulder [28]), the topic-comment distinction (often occurring with beats), and the turn allocation in conversation (as the agent starts to talk she often starts to gesticulate). The fourth cluster represents the expression of emotions (the expressiveness of the movement, i.e., how sharp or tense it is, or for how long it lasts, is dependent on the agent's emotion [9]). Finally, the last set gathers expressions concerning the kind of thinking activity in which the agent is currently engaged in: when 'remembering a fact' or 'trying to make inferences', we often avoid gaze input by looking up with our finger on our cheek, as we are concentrating on our thought and want to avoid an information overload.

### 4.2 Gesture Specification Language

To specify gesture prototypes that fit the above categorization, a proprietary, abstract, high level scripting language for hand-arm gestures, based on a functional separation of arm position, wrist orientation, and hand shape, was developed. We have chosen a key frame-based approach to record the dynamics of gestures. Gestures are defined by a sequence of timed key poses which we call gesture frames. The key frame method allows for a compact and easily interpretable storage of definitions; it also permits us to leverage practices of traditional figure animation to increase the realism of the generated animations [19], as we will describe in Section 5.1.

Following McNeill [22], gestures are broken down temporally into distinct phases: preparation, stroke, hold, and retraction. Only the stroke is mandatory for all gestures, since it carries the semantic load of the gesture and also represents the critical element in synchronizing gesture to speech.

Gesture space is parametrized using McNeill's [22] system of concentric squares centered on the actor. In this scheme, the space in front of the gesturer is divided into seven horizontal, seven vertical and three distal sectors, each of which can be the target for the arm position during any gesture frame. McNeill empirically determined that different varieties of gestures are predominantly executed in certain reach-space sectors. While such knowledge is not built into the Gesture Engine system per se, it can be easily expressed in the specification process.

For wrist orientation and hand shape determination, we rely on a subset of HamNoSys [31], a language-independent corpus of formal hand sign classifications. While gesturing behavior is culture-specific, it must not be confused with a language-dependent sign-system. Thus the choice of a set of internationally applicable notation conventions not bound to any particular sign-language appeared most suitable for the design of a general gesture synthesizer. In our implementation of HamNoSys, the basic shape of the hand can be chosen from twelve fundamental forms or symbols. Additionally, the configuration of the thumb can be changed and the opening/closing of each of the remaining four fingers can be set independently. Wrist orientation is specified as a combination between two orthogonal vectors: the Finger Base vector extending from the wrist to the first joint of the index finger, and the Palm Normal vector, which extends out of the inner palm plane at a right angle. The wrist orientation is thus defined globally in relation to the body orientation, which corresponds to the kind of goal directed motion planning found in people. A short example will illustrate this point: we want our Finger Base to be perpendicular to the ground plane and our Palm Normal to face away from us when we signal “Stop!” – regardless of the particular position of our arm at that moment.

Each gesture frame can define any or all separate gesture components. Thus different joint chains are uncoupled and allowed to follow their own respective trajectories for maximum flexibility and control. Complete gesture definitions store a sequence of gesture frames along with global identifiers and constraints that apply to the gesture as a whole. Definitions are stored in a human-readable ASCII file format. The initial set of test gestures was transcribed directly from video analysis of the performance of a trained linguist acting as the signer. We shall demonstrate the introduced format with a specific example from that set of transcribed gestures. The following is the definition of the gesture denoting the adjective “small” - a frame of the execution of this gesture by our agent is shown in Figure 2:

```
GESTURECLASS adjectival
GESTUREINSTANCE small
DURATION 1.5
STARTFRAME 0.0
  FRAMETYPE stroke_start
  ARM XC YUpperP ZNear
  HAND symbol_1_open
  FINGER index bend_curved
  WRIST FBUp PalmInwards
ENDFRAME
STARTFRAME 0.4
  FRAMETYPE stroke_end
  ARM XC YUpperP ZMiddle
  WRIST FBDefault PalmDefault
  ADDNOISE
```

```
ENDFRAME
STARTFRAME 1.0
  FRAMETYPE hold
  ARM XC YUpperP ZMiddle
  HAND symbol_1_open
  FINGER index bend_curved
  WRIST FBUp PalmInwards
ENDFRAME
```

The gesture definition file delineates the specific gesture “small”, which belongs to the larger group of iconic gestures accompanying adjectives. The default duration of this gesture is defined to be 1.5 seconds. This duration is a goal value that the Gesture Planner will try to satisfy; however, the exact timing of the gesture is dependent on temporal constraints imposed by the previous as well as following gestures in the conversation plan. Optionally, a default arm assignment could have been made. This feature is mainly used for gestures which animate both arms simultaneously (as in Figure 5). Without this option, the Gesture Planner (Section 5.1) will handle arm assignments to create consistent performances. Subsequent to the header information, a list of gesture frame blocks fully define the gesture animation. In the present example, the gesture consists of three gesture frames.

The first frame has a relative timing of 0.0 and will thus be scheduled at the very beginning of the time slot allocated to the gesture by our system. This frame represents the hand-arm configuration that the actor will assume at the start of the stroke phase. No preparation phase was specified, causing the system to transition directly to the stroke position from the assigned arm’s previous location during a conversation performance. In the sixth line, the arm position is defined to be in the outer center (XC for center) in height of the actor’s shoulder (YUpperP for upper periphery), close to the body (ZNear). The basic hand shape is of type symbol\_1\_open, which is a pointing gesture with an extended thumb parallel to the index finger. The thumb/index distance is too great to signify “small”, so the index finger is bent down slightly by the addition of a FINGER command.

In the second gesture frame, the end of the stroke is indicated. The arm has moved away from the body (ZMiddle) and the wrist is aligned with the forearm without twist or abduction. Note that no hand shape was specified. This allows the animation system to freely interpolate positions to create smooth animation transitions. The ADDNOISE keyword subtly alters the position of this frame randomly in a manner similar to Perlin’s [26], to avoid mechanical-looking execution, which becomes especially important during stroke repetition, which will be discussed below in Section 5.1. ADDNOISE can be added to any frame. The third frame finally holds the arm at its semi-extended position until the end of the allocated gesture time.



Figure 2. Sample gesture : “small”

To summarize, a gesture has the following global specifiers:

- classification of gesture
- typical duration of the gesture in seconds
- assignment of a single arm or both arms (optional)

In addition, each gesture frame can contain a combination of the following specifiers:

- timing of the key relative to the duration of the entire gesture (mandatory)
- gesture phase of the current key (e.g., stroke-end or retraction; mandatory)
- arm position (optional)
- wrist orientation (optional)
- hand shape (optional)
- additional positioning for each finger (optional)

We acknowledge the existence of gesture positions that are difficult or cumbersome, if not impossible, to define within the presented framework. Consequently we have added an override option that allows the user to define a position by explicitly fixing joint angles. We have provided scripted tools that allow to load and modify any pose in Alias|Wavefront’s Maya™ and to subsequently export a newly created pose back to Gesture Engine. However, this mechanism circumvents one of the main features of a universally applicable gesture specification language, namely abstraction, and should thus be employed with restraint.

### 4.3 Gesture Editor

We have built a visual interface to allow system users to interactively compose gestures in a graphical environment

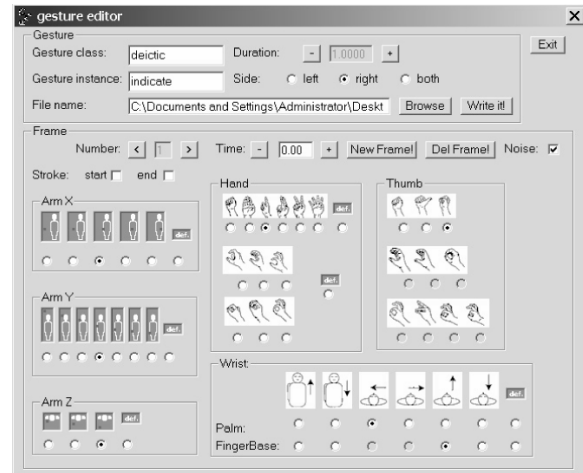


Figure 3. The interactive Gesture Editor

and export the created gestures to the Gesture Engine library (see Figure 3). The user can compose any gesture using any number of parameters and their corresponding values. Gesture Editor saves the created gesture in a text file using the same language specification outlined above. Beyond allowing natural and interactive gesture specification, the editor also provides the advantage of checking input for plausibility and correctness. The editor does not allow the selection of non-sensical parameter combinations, thus assuming the function of an error controller for the user. For instance if the user has selected the palm has being oriented upward, the user can subsequently only choose from those remaining finger base values that satisfy the orthogonality requirement between the two vectors: finger base oriented away, inwards, outwards or towards. Additionally, the editor prevents keyframe timing errors (frame times have to increase and adhere to upper and lower limits), and verifies arm shape (the spatial position has to be well-defined) and hand–thumb shape associations.

## 5 Gesture Animation

In previous work [4], we have developed a dialog manager that not only plans discourse moves but also provides information on the co-occurring nonverbal behaviors. To ensure synchronism between the verbal and nonverbal streams, we are utilizing the XML framework. We have extended XML to include nonverbal communicative acts [4]. The output of the discourse planner is an utterance tagged with nonverbal information. Gesture Engine parses the XML file for one conversational turn and passes the utterances to Festival [37], a speech synthesizer that returns a sound file as well as a list of phonemes and their durations (see Figure 4). A high-level Turn Planner then matches

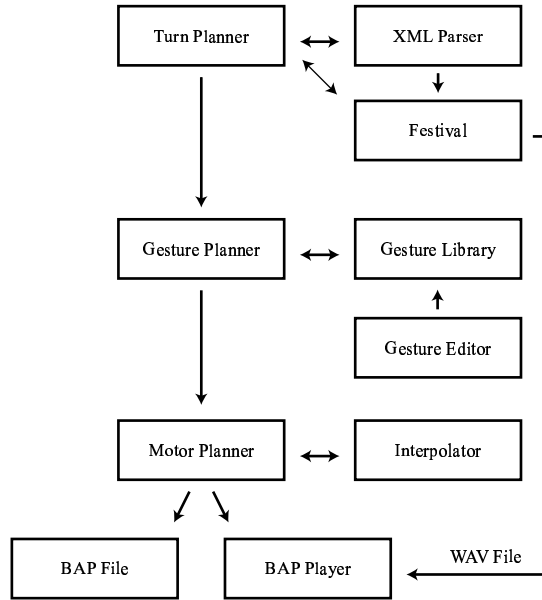


Figure 4. System outline

meta-tag information against gesture prototypes stored in the Gesture Library. Currently, matches are found using a straightforward lexical lookup. For gesture requests that do not carry sufficient contextual information, or for specific gesture requests that cannot be matched with suitable prototypes in the Gesture Library, one of multiple generic beat gestures is chosen to be executed. A more sophisticated inference engine sensitive to communicative context is to be implemented in the future; the Turn Planner will therefore not be further discussed here. At the subsequent level, a Gesture Planner<sup>3</sup> also calculates rest positions for arms during inter-gestural pauses. The gesture plan thus constructed is finally passed to a Motor Planner that computes smooth joint-angle trajectories using quaternions [33] and tension-continuity-bias (TCB) spline curves [16]. The Motor Planner then stores the joint angles as MPEG-4 BAP frames, which can be written to file or sent directly to the internal viewer (some generated gesture frames are reproduced in Figure 5).

## 5.1 Gesture Planner

The task of the Gesture Planner is to instantiate the prototype gestures defined in the database according to temporal and contextual constraints. First and foremost, generated gestures have to be matched with the semantic structure of the accompanying utterances. The meta-informational

<sup>3</sup>We adopt Kopp and Wachsmuth's [17] instantiates gesture prototypes and schedules gestures, adjusting their timing and positioning. The Gesture Planner fitting terminology for our architecture here.

XML mark-up in the discourse transcript supplies us with this term-level mapping. In addition, tight phonological synchronization to the synthesized speech is necessary if the gestures are to contribute to, not detract from, effective communication. McNeill [22] provides us with an empirically backed axiom – the emphasis in gesture, namely the stroke, has to coincide or slightly precede the emphasis in speech. It may never follow. Thus the Gesture Planner regards synchrony of the *end* of a gesture's stroke with the utterance of the emphasized word as its absolute timing constraint around which other gesture frames are then scheduled.

The duration of preparation and retraction phases of gesture prototypes can either be compressed or expanded during the instantiation process. The limits of such duration modifications should be further investigated - it is clear that variability in execution exists in human gesturing; however, too large an alteration will impair the readability of the gesture. The Gesture Planner aims to guarantee a minimum transition period between gestures. If this transition time cannot be allocated without exceeding a temporal scaling threshold, the problematic gesture is classified as unexecutable and is erased from the gesture plan. If on the other hand scheduled gestures are separated by extended periods of inactivity, the Gesture Planner inserts rest positions into the inter-gestural periods to return gesturing arms to neutral positions.

To further increase the realism of the gestural performance, we introduce the concept of *stroke expansion*. Observation of human gesturing suggests that beat gestures can co-articulate with other kinds of gestures [6] to express additional rhythmical emphases in a sentence. In these cases, the first execution of the stroke of a given gesture carries its usual semantic function; afterwards, however, the hand remains in its assumed shape and the arm partially repeats the gesture's stroke movement to further accentuate the rhythm of the associated speech. We identify this behavioral pattern by analysis of the XML meta mark-up. If multiple intonational emphases are present within a rheme clause for which an appropriate gesture has been found in the Gesture Library, that gesture's stroke will be repeated with diminishing amplitudes such that each stroke end coincides with an emphasized word in speech.

While it is sensible for the sake of clarity and conciseness to specify an entire pose in a single gesture frame, the limbs of a real person rarely if ever move simultaneously and in synchrony. Instead, motions originate in one part of the body and propagate to neighboring joints with a non-negligible delay. In the case of keyframed action, this results in small but significant timing differences between successive joint keys. Traditional animators have known and exploited this fact for a long time and have assigned to it the name *follow-through* [19]. We implement follow-

through according to Lasseter’s [19] observation that arm movements start at the shoulder and propagate down towards the fingers. For a given key pose that defines arm position as well as hand shape, the Gesture Planner shifts the shoulder joint key backwards in time while finger joint timings will be moved forwards successively. The magnitude of follow-through is determined by a linear scaling of the time interval to the following keyframe, clamped to a maximum of 9 frames.

## 5.2 Motor Planner

Once a complete timed plan of key poses has been assembled, the motor planner computes the actual corresponding key joint angles. To this end, arm position and finger bend angles are loaded from a list of pose definitions which contain appropriate joint angle values for a given anthropometry. After the arm angles have been determined, the wrist orientation is dynamically calculated through multiplication of rigid transformation matrices. The Motor Planner then sets up individual interpolating splines through the key positions of each formational unit for each arm. We chose to implement interpolation using Tension-Continuity-Bias (TCB) splines, also known as Kochanek-Bartels splines in reference to the authors [16]. TCB splines allow for convenient high-level control of tension, continuity and bias parameters of the curve using three scalar values per control point. Joint angle rotations are mostly animated with quaternions [33], using an extension of TCB splines to quaternion interpolation described by Eberly [11]. Otherwise, Euler angle values are animated directly. Interpolating Euler angles is known to yield non-linear mappings and to show degeneracies [38] due to the surjection of Euler angles to unique rotation matrices. However, because of the limited range of possible angle values in the human arm–hand chain, degeneracies can be successfully avoided given careful initial orientation – our results were convincing and stable. Nevertheless we recognize the need for a more robust and mathematically sound interpolation procedure and are thus planning to replace all references to Euler angles in the interpolation stage.

## 6 Future Work

We have outlined a compact, yet flexible system for the specification and generation of humanoid gesturing behavior. Gesture Engine is an important step towards attaining the goal of the MagiCster project - the creation of a believable embodied conversational agent. And yet, it is but *one* step. On a macroscopic level, we now need to integrate Gesture Engine with the existing Greta system for facial animation to produce a truly multi-modal communicator. The problem of conveying emotion in addition to fac-



**Figure 5. Sample gestures: A deictic gesture (pointing to the agent’s own chest) and three different beat gestures**

tual information has also come to the attention of numerous researchers recently. In this context we wish to investigate if we can adapt the EMOTE [9] model for effort and shape to Gesture Engine, which may require a switch from orientation-based interpolation to position-based interpolation along with the use of an inverse kinematics module. Finally, we regard it as essential to include larger postural considerations in the agent architecture. The field of torso animation is still wide open - a definitive study is sorely needed. The motivation for a holistic animation system was delivered by Watzlawick more than 25 years ago, when he postulated the first principle of the Interactional view of communication: “One cannot not communicate.”[39] Adapted to the context of embodied conversational agents, the axiom asserts that the “non-action” of an agent in some specific channel of communication is never overlooked or ignored by the agent’s human counterpart. In the best case, it will be interpreted as a somewhat odd quirk; more likely though, it will seriously detract from the agent’s real communicative intent and thus limit its efficacy.

## 7 Acknowledgments

We would like to thank Isabella Poggi and Massimo Bilvi for their indispensable input during the development of Gesture Engine. We would also like to thank Alias|Wavefront for providing us with their Maya™ animation software and a body model. This research is partially supported by IST project MagiCster, IST-1999-29078.

## References

- [1] C. Babski. MPEG-4 player on the web for H-Anim bodies. <http://ligwww.epfl.ch/~babski/StandardBody/mpeg4/>.
- [2] N. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao, and M. Palmer. Parameterized action representation for virtual human agents. In S. P. J. Cassell, J. Sullivan and E. Churchill, editors, *Embodied Conversational Characters*. MITpress, Cambridge, MA, 2000.
- [3] R. Bindiganavale and N. Badler. Motion abstraction and mapping with spatial constraints. In *International Workshop on Modelling and Motion Capture Techniques for Virtual Environments, CAPTECH'98*, pages 70–82, November 1998.
- [4] N. D. Carolis, C. Pelachaud, I. Poggi, and F. de Rosi. Behavior planning for a reflexive agent. In *IJCAI'01*, Seattle, USA, August 2001.
- [5] J. Cassell. Embodied conversation: Integrating face and gesture into automatic spoken dialogue systems. In Luperfroy, editor, *Spoken Dialogue Systems*. MIT Press, Cambridge, MA, 2001.
- [6] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Computer Graphics Proceedings, Annual Conference Series*, pages 413–420. ACM SIGGRAPH, 1994.
- [7] J. Cassell, O. Torres, and S. Prevost. Turn taking vs. discourse structure: How best to model multimodal conversation. In Y. Wilks, editor, *Machine Conversations*. Kluwer, The Hague, 1999.
- [8] J. Cassell, H. Vilhjálmsón, and T. Bickmore. BEAT : the Behavior Expression Animation Toolkit. In *Computer Graphics Proceedings, Annual Conference Series*. ACM SIGGRAPH, 2001.
- [9] D. M. Chi, M. Costa, L. Zhao, and N. I. Badler. The EMOTE model for effort and shape. In *Computer Graphics Proceedings, Annual Conference Series*, pages 173–182. ACM SIGGRAPH, 2000.
- [10] S. Duncan and D. Fiske. *Interaction Structure and Strategy*. Cambridge University Press, 1985.
- [11] D. Eberly. Key frame interpolation via splines and quaternions. <http://www.magic-software.com>.
- [12] P. Ekman. The argument and evidence about universals in facial expressions of emotion. In H. Wagner and A. Manstead, editors, *Handbook of Social Psychophysiology*, pages 143–164. Wiley, Chichester; New-York, 1989.
- [13] M. Gleicher. Retargetting motion to new characters. *Computer Graphics*, 32(Annual Conference Series):33–42, 1998.
- [14] A. Kendon. Movement coordination in social interaction: Some examples described. In S. Weitz, editor, *Nonverbal Communication*. Oxford University Press, 1974.
- [15] A. Kendon. Gestures as illocutionary and discourse structure markers in southern Italian conversation. *Journal of Pragmatics*, 23:247–279, 1995.
- [16] D. H. U. Kochanek and R. H. Bartels. Interpolating splines with local tension, continuity, and bias control. In H. Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 33–41, 1984.
- [17] S. Kopp and I. Wachsmuth. A knowledge-based approach for lifelike gesture animation. In W. Horn, editor, *ECAI 2000 Proceedings of the 14th European Conference on Artificial Intelligence*. IOS Press, 2000.
- [18] S. Kopp and I. Wachsmuth. Planning and motion control in lifelike gesture: A refined approach. In *Proceedings of Computer Animation*, pages 92–97, 2000.
- [19] J. Lasseter. Principles of traditional animation applied to 3D computer animation. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 21, pages 35–44, 1987.
- [20] T. Lebourque and S. Gibet. Synthesis of hand-arm gestures. In P. A. Harling and A. D. Edwards, editors, *Proceedings of Gesture Workshop '96*, London, 1997. Springer-Verlag.
- [21] T. Lebourque and S. Gibet. High level specification and control of communication gesture: the GESSYCA system. In *Proceedings of Computer Animation*, pages 24–35. IEEE Computer Society, 1999.
- [22] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago, 1992.
- [23] C. Mueller. Conventional gestures in speech pauses. In C. Mueller and R. Posner, editors, *The Semantics and Pragmatics of Everyday Gestures*. Berlin Verlag Arno Spitz, Berlin, 2001.
- [24] J. Ostermann. Animation of synthetic faces in MPEG-4. In *Proceedings of Computer Animation*, pages 49–55, 1998.
- [25] C. Pelachaud, E. Magno-Caldognetto, C. Zmarich, and P. Cosi. An approach to an italian talking head. In *Eurospeech'01*, Aalborg, Denmark, September 3-7 2001.
- [26] K. Perlin. Noise, Hypertexture, Antialiasing and Gesture. In D. Ebert, editor, *Texture and Modeling, A Procedural Approach*. AP Professional, Cambridge, MA, 1994.
- [27] I. Poggi. Mind markers. In *5th International Pragmatics Conference*, Mexico City, July 5-9 1996.
- [28] I. Poggi. Gesture, gaze and touch: Literal and indirect meaning. In *virtual symposium on "Multimodality of human communication: Theories, problems and applications"*, Toronto, 2001.
- [29] I. Poggi, C. Pelachaud, and F. de Rosi. Eye communication in a conversational 3D synthetic agent. *Special Issue on Behavior Planning for Life-Like Characters and Avatars of AI Communications*, 13(3):169–181, 2000.
- [30] M. Preda, T. Zaharia, and F. Prêteux. 3D body animation and coding within a MPEG-4 compliant framework. In *International Workshop SNHC*, 1999.

- [31] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, and J. Henning. Hamburg notation system for sign languages: An introductory guide. In *International Studies on Sign Language and Communication of the Deaf*, volume 5. Signum Press, Hamburg, Germany, 1989.
- [32] K. Scherer. Vocal affect expression: A review and a model for future research. *Psychological Bulletin*, 99(2):143–165, 1986.
- [33] K. Shoemake. Animating rotation with quaternion curves. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 245–254, 1985.
- [34] W. Stokoe. *Sign language structure: An outline of the communicative systems of the American deaf*. Linstock Press, Silver Spring, 1978.
- [35] W. Stokoe, D. Casterline, and C. Croneberg. *A dictionary of American Sign Language on linguistic principles*. Gallaudet College Press, Washington, D.C., 1965.
- [36] G. Taubin. SNHC verification model 7.0. Technical report, MPEG-4, 1998.
- [37] P. Taylor, A. Black, and R. Caley. The architecture of the the Festival speech synthesis system. In *Third International Workshop on Speech Synthesis*, Sydney, Australia, November 1998.
- [38] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques*. Addison Wesley, 1992.
- [39] P. Watzlawick, J. Beavin, and D. Jackson. *Pragmatics of human communication; a study of interactional patterns, pathologies, and paradoxes*. Norton, New York, 1967.
- [40] Web3D Consortium Humanoid Animation Working Group. H-Anim 1.1 specification. <http://h-anim.org/spec1.1/>, 1999.