Surface Realization of Multimodal Output from XML representations in MURML

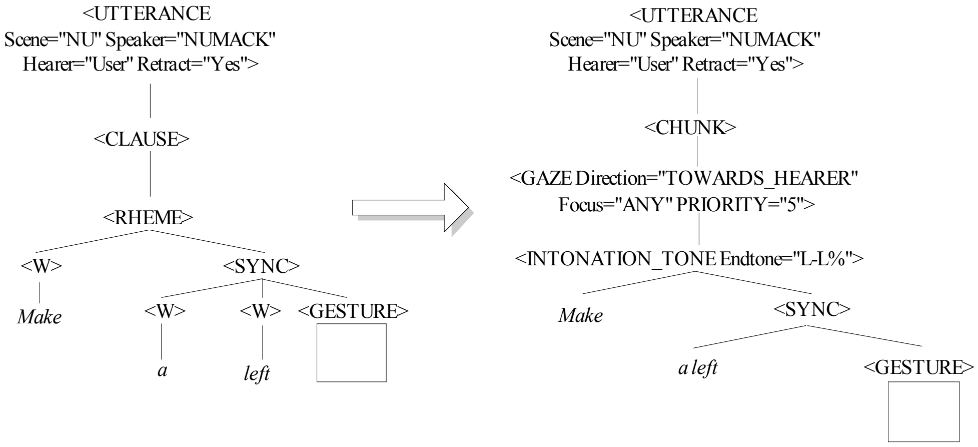
Stefan Kopp
Artificial Intelligence Group
Faculty of Technology, University of Bielefeld
D-33594 Bielefeld, Germany
skopp@TechFak.Uni-Bielefeld.DE

Introduction

Our research aims at creating a natural communication link between humans and technical systems. The way we want to achieve this is by employing anthropomorphic agents that are capable of the same interactional abilities as humans. Among other things, this involves a combined usage of several modalities in face-to-face conversations, resulting in utterances that are composed of simultaneous and synchronized verbal and nonverbal behaviors.

The automatic generation of natural multimodal output requires a time-critical production process with high flexibility. In general, we can conceive of this process in terms of the stages commonly assumed in Natural Language Generation (NLG) literature (e.g., Reiter & Dale, 2000). These architectures are usually modular, pipeline models broken down into three subtasks—*content planning* (also known as text or document planning), *microplanning*, and *surface realization* (in that order). Starting from a goal the speaker wants to achieve, in ordinary language, the work done by these three subsystems boils down to, respectively, figuring out what to say, figuring out how to say it, and finally, saying it. These stages are crucially linked and, in our case, they must operate not only on speech but include other modalities as well. Ideally, each stage provides sufficient flexibility and power as to not limit the potential outcome of the previous stage. For example, surface realization should be able to realize every verbal, gestural, phonological, etc. action the microplanner comes up with. This presupposes the ability to generate a large variety of verbal, paraverbal and gestural behaviors in real-time, from a representation that allows for specifying all decisive features of these behaviors, and the temporal relations between them.

This paper is on surface realization which I conceive of as comprising two consecutive steps: *behavior augmentation* and *behavior realization*. The purpose of behavior augmentation is to impart to an utterance apposite nonverbal and paraverbal behaviors, like intonation, eyebrow raise, head nod, or posture shifts. These behaviors cannot always been determined during microplanning for they do not encode explicit communicative goals, but, nevertheless, are essential to meaning as they underscore the central parts of the utterance. In our current agents, this task is achieved by a module that stems from the *BEAT* system (Cassell et al., 2001). It operates on a tree-based XML representation that contains the textually defined words as leaf nodes, gestures (defined as described below) in sync with certain words, and meta-information about the clausal structure or the information structure (rheme, theme) of the utterance. Based on this information, the module augments the utterance by inserting behavior nodes in the tree (see Cassell et al., 2001), before it eliminates all meta-information tags. Figure 1 shows an example from the *NUMACK* system (Kopp et al., 2004). The resulting tree is then flattened into a *MURML* description (Multimodal Utterance Representation Markup Language; Kranstedt, Kopp & Wachsmuth, 2002) and then passed on to behavior realization, which I will focus on in the remainder of this paper.mut



<UTTERANCE Scene="NU" Speaker="NUMACK" Hearer="User" Retract="Yes">
<CLAUSE>
<RHEME>
<W>
<SYNC>
Make
<W>
<W>
<GESTURE>
a
left
<UTTERANCE Scene="NU" Speaker="NUMACK" Hearer="User" Retract="Yes">
<CHUNK>
<GAZE Direction="TOWARDS_HEARER" Focus="ANY" PRIORITY="5">
<INTONATION_TONE Endtone="L-L%">
Make
<SYNC>
a left
<GESTURE>

Figure 1: Insertion of non-/paraverbal behaviors in a single-clause utterance tree during Behavior Augmentation.

Behavior Realization

Behavior realization must turn a formal representation of an utterance into multimodal output of an animated agent. Originating from microplanning or behavior augmentation, such an input representation is meant to be a straightforward description of how the utterance is supposed to look or sound like. That is, it specifies the words to be verbalized, the phonological properties, and the accompanying nonverbal behaviors. The final realization step is accomplished by the *Articulated Communicator Engine* (ACE, for short), which allows to create and visualize animated multimodal agents. In the next sections, I will first describe the underlying model of ACE and its assumptions that frame surface realization as an incremental production process, and that motivate the structure of the representations it takes as input. Then, I will explain how these input representations can be formed using the MURM specification language.

Incremental speech-gesture realization

ACE's production model aims at creating lifelike, synchronized verbal and nonverbal behaviors in a human-like flow of multimodal behavior. To this end, it tries to simulate the mutual adaptations that appear to take place between speech and gesture when humans try to achieve synchrony between co-expressive elements in both modalities. The hallmark of the ACE approach is that it rests upon an incremental process model that allows for handling the cross-modal interactions at different levels of the utterance, corresponding to decisive points in multimodal behavior generation. This production model is based upon the following hypothesis.

The segmentation hypothesis and speech-gesture synchrony

In order to organize the realization of gesture and speech over multiple sequential behaviors, we adopt an empirically suggested assumption (McNeill, 1992) as *segmentation hypothesis*: Continuous speech and gesture are co-produced in successive segments each expressing a single idea unit. The inherent segmentation of speech-gesture production in humans is reflected in the hierarchical structures of overt gesture and speech and their cross-modal correspondences. Kendon (1980) defined units of gestural movement to consist of *gesture phrases* which comprise one or more subsequent movement phases, notably *preparation*, *stroke* (the expressive phase), *retraction*, and *holds*. Similarly, the phonological structure of connected speech in intonation languages such as English and German is organized over *intonation phrases* (e.g., cf. (Levelt, 1989)). Such phrases are separated by significant pauses, they follow more the semantical (deep clause) structure than the syntactical phrase structure, and they have a meaningful pitch contour with exactly one primary pitch accent (the nucleus). We define *chunks* of speech-gesture realization to be pairs of an intonation phrase and a co-expressive gesture phrase. That is, complex utterances with multiple gestures are considered to consist of several chunks. Within each chunk, the prominent concept is concertedly conveyed by the gesture and an affiliated word or subphrase (the *affiliate*), and their co-expressivity is evidenced by a general temporal synchrony: Gestural movements are timed such that the meaning-bearing stroke phase starts before the affiliate and frequently spans it, optionally by inserting dedicated hold phases in the flow of movement. This coupling is refined if one of the affiliated words gets prosodically focused, e.g., for emphasizing or contrasting purposes, and hence carries the nucleus of the phrase. In this case, the gesture stroke starts with the nucleus at the latest and is not finished before it (deRuiter, 1998; Nobe, 2000; McNeill, 1992).

Mechanisms of cross-modal adaptation

In humans, the synchrony of co-expressive verbal and gestural elements is accomplished through cross-modal adaptations that, based on the segmentation hypothesis, can take place either within a chunk or, at a higher level, between two successive chunks. ACE's production process thus runs incrementally and simulates these mechanisms on these two levels of the utterance:

- *Producing a chunk:*
  Within a chunk, i.e. an intonation phrase and a co-expressive gesture phrase, the synchrony between the affiliate (or nucleus) and the stroke is mainly accomplished by the gesture's adapting to the timing of speech, which in turn runs mostly unaffected by gesture ("ballistically"). In producing a single chunk, the intonation phrase can therefore be synthesized in advance, potentially augmented with a strong pitch accent for narrow focus. As in previous systems, ACE thus exploits information about absolute phoneme timings retrieved from TTS to set up timing constraints for coverbal gestural or facial behaviors. The gesture stroke is thereby set either to precede the affiliate's onset by a given offset (per default one syllable's approximate duration of 0.2 sec) or to start exactly at the nucleus if a narrow focus has been applied. In any case, the stroke is set to span the whole affiliate before retraction starts. For dynamic gestures, this is achieved by inserting a post-stroke hold or additional repetitions, both strategies observable in humans (McNeill, 1992).

- *Combining successive chunks:*
  Humans appear to anticipate the synchrony of the forthcoming affiliate (or nucleus) and gesture already before the chunk starts. Indeed, one can find preparatory effects in *both* modalities at the boundary of successive chunks: The onset of the gesture phrase co-varies with the position of the nucleus, and the onset of the intonation phrase co-varies with the stroke onset (DeRuiter, 1998; Nobe, 2000; McNeill, 1992). In consequence, movement between two strokes depends on the timing of the successive strokes and may range from the adoption of intermediate rest positions to direct transitional movements (the so-called co-articulation effects). Likewise, the duration of the silent pause between two intonation phrases may vary according to the required duration of the preparation for the next gesture. Simulating these mechanisms is highly context-dependent for it has to take into account properties of the subsequent gesture stroke (form, location, timing constraints) as well as the current movement conditions when the previous chunk can be relieved, i.e., when its intonation phrase and its gesture stroke are completed. ACE extends the traditionally employed, planning-executing procedure for each chunk to additional phases

in which the production processes of subsequent chunks can relieve one another. The adaptation effects in speech and gesture are simulated during the phase when the next chunk can readily be uttered ("lurking") while the former is "subsiding", i.e., done with executing its mandatory parts (intonation phrase and gesture stroke). It is at this time, that intra-chunk synchrony is defined and reconciled with the onsets of phonation and movement, and that animations are created on-the-fly satisfying all the movement and timing constraints now determined.

Figure 2 demonstrates a case in which the affiliate in the next chunk (i+1) is located relatively early in the intonation phrase and the gesture requires—under the current movement conditions—an extensive preparation. Thus, it has to start early in order to naturally meet the mandatory timing of stroke onset (shortly before the affiliate). Since at this point the preceding gesture has not been fully retracted, a fluent gesture transition emerges after an only partial retraction, due to the placement of the affiliate within the next verbal phrase. Additionally, the time needed for the preparatory movement requires the gesture's preparation to precede speech, which in turn results in a stretched vocal pause between the intonation phrases due to the current movement context.

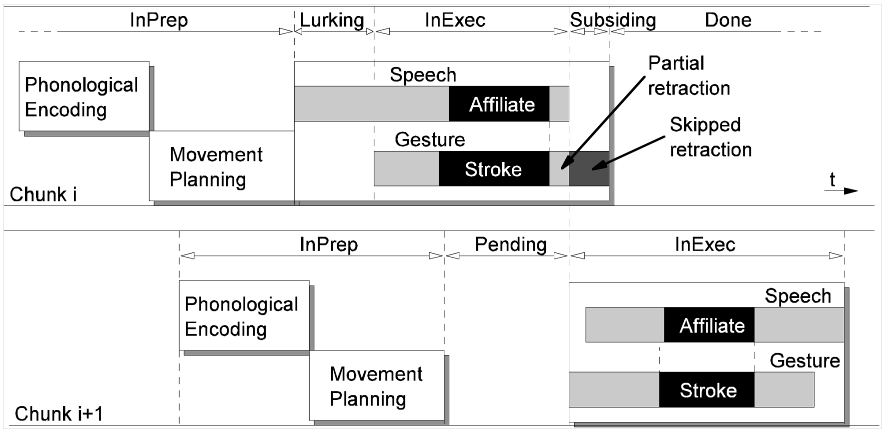


Figure 2: Context-dependent transition between two chunks.

Representing input to behavior realization in MURML

As aforementioned, the knowledge structures inputted to ACE must define all of the overt aspects of the target utterance. We employ MURML as representation language to specify this information. Following the incremental nature of behavior realization, each input specification must at least completely specify one chunk, but may also comprise a complex utterance out of multiple chunks when its inherent chunk structure is marked. As exemplified in Figure 3, such a description contains two major parts: (1) a textual definition of the verbal part of the utterance (specification), and (2) specifications of paraverbal or nonverbal behaviors such as prosodic foci (focus), gestures, or facial animations (behavioprspec). Each part can be omitted if the utterance/chunk does not contain the respective behavior.

If the chunk comprises both speech and nonverbal behaviors, the running speech is well-suited to define an intra-chunk timeline as it runs mostly ballistically (see above). The moment of appearance of a behavior can thus be defined either explicitly in terms of absolute times (start, end, duration) w.r.t. to the start of the overall chunk, or implicitly by stating its affiliation with certain linguistic elements. Both kinds of temporal placement are definable for every behavior, using either a timing or an affiliate tag. For affiliation, one must refer to the points in the course of speech when the co-expressive words start or end (onset, end). To this end, time tags (time) can be inserted in the textual specification of the verbal output and can be referred to while defining affiliation (e.g. <affiliate onset=”t4” end=“t5”/>). The same applies to prosodic focus, which is inherently coverbal and can only be assigned to parts of speech (onset, end). In addition, the time tags are used to set the boundaries of chunks in a complex utterance (e.g. <time id=“t3”' chunkborder=“true”/>).

```
<definition><utterance>
  <specification>
    And now take <time id=“t1”/> this <time id=“t2”/> bar <time id=“t3”' chunkborder=“true”/>
    and make it <time id=“t4”/> this big. <time id=“t5”/>
  </specification>

  <focus onset=”t1” end=”t2” accent=”H*”/>

  <behaviorspec id=”gesture_1”>
      <gesture id=”pointing_to”>
         <affiliate onset=”t1” end=”t3”/>
         <param name=”refloc” value=”$Loc-Bar_1”/>
      </gesture>
  </behaviorspec>

  <behaviorspec id=”gesture_2”>
      <gesture>
         <affiliate onset=”t4” end=“t5”/>
         <constraints>
            <symmetrical dominant=”right_arm” symmetry=”SymMS”>
               <parallel>
                  <static slot="HandShape" value="BSflat (FBround all o) (ThCpart o)"/>
                  <static slot="PalmOrientation" value="DirL"/>
                  <static slot="ExtFingerOrientation" value="DirA"/>
                  <dynamic slot="HandLocation">
                     <dynamicElement type="linear">
                       <value type="start" name="LocShoulder LocCenterRight LocNorm"/>
                       <value type="direction" name="DirR"/>
                       <value type="distance" name="125.0"/>
                     </dynamicElement>
                  </dynamic>
               </parallel>
            </symmetrical>
         </constraints>
      </gesture>
  </behaviorspec>
</utterance></definition>
```

Figure 3a: Sample XML specification including a verbal utterance, a prosodic focus (H* on “this”), and two gestures. The second gesture (shown in Fig 3b) is defined explicitly.

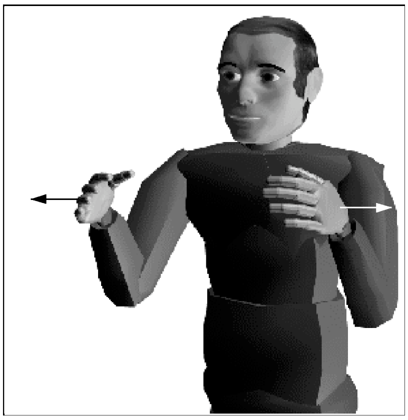


Figure 3b: Gesture generated from the behavior specification ‘**gesture_2**’ in Fig 3a.

Note that this way of specifying speech and coverbal behaviors in separate allows for having behaviors that temporally overlap. This is not possible in a single tree-structured description. The utterance tree arriving from behavior augmentation, as shown in Figure 1, is thus flattened into such a MURML description, with time tags and behavior specifications derived according to their definition and placement in the tree.

Flexibility of gesture generation means that all spatio-temporal features of a gesture can be specified in accordance to the individual context of accompanying speech. The actual form of every nonverbal behavior can, on the one hand, be defined in MURML as a parametric keyframe animation (e.g., in joint angles or face muscle values). On the other hand, communicative gestures can be represented in terms of the spatio-temporal features of their morphology (as described below). The second gesture in Fig. 3 (gesture_2) is defined in such a way. Frequently, one may want to define templates for gestures that have conventionalized features like the typical hand shape in pointing. ACE allows to equip an agent with a repository of gesture templates (a lexicon of XML descriptions) which can be drawn from in MURML specs. For example, the first gesture in Fig. 3 (a pointing gesture; gesture_1) will be drawn from this repository based on the provided identifier (pointing_to). To ensure flexibility, each template can accommodate parametrizeable feature values that must be set to actual positions, directions, or angles when being retrieved in the MURML utterance specification. In the gesture_1 example the parameter refloc, which corresponds to the location of the pointing target is bound to yet another context variable, notably, a certain object's position. Alternatively, global parameter tags can be defined for an utterance specification (either as part of the spec itself, or given to ACE in addition to the spec) to set up the individual context of this utterance in terms of slot-value pairs:

```
….
<parameter slot= “object_loc_1” value=”1500 10 100”/>
….
<static slot=”ExtFingerOrientation” value=”$object_loc_1” mode=”pointTo”/>
….
```

Feature-based gesture specification

To represent a gesture that is meant to convey a certain communicative intent, it suffices to specify the essential spatio-temporal features that make up the morphology throughout its meaningful phase. MURML exactly allows defining these features. Building on *HamNoSys* (a notation system for the German sign language of the deaf (Prillwitz et al., 1989)), a hand-arm configuration is, at first, described in terms of four components:

(1) the *location* of the wrist, symbolically specified by unique identifiers for the position in the frontal, transversal, and sagittal plane (Figure 4)

(2) the *shape* of the hand, compositionally described by the overall hand shape and additional symbols denoting the kind and degree of flexion within each finger

(3) the *extended finger orientation (EFO),* a vector originating at the wrist and running along the length of the back of the hand, specified relative to the agent’s reference frame

(4) the *palm orientation (PO*), the normal vector of the palm, specified either absolutely as direction w.r.t. the agent's frame of reference, or relative as rotation around the axis of extended finger orientation.

Note that the last two components compositionally define the orientation of the hand/wrist. Each component can be described either symbolically (using the symbols listed in the following table), or numerically in terms of vectors or angles.

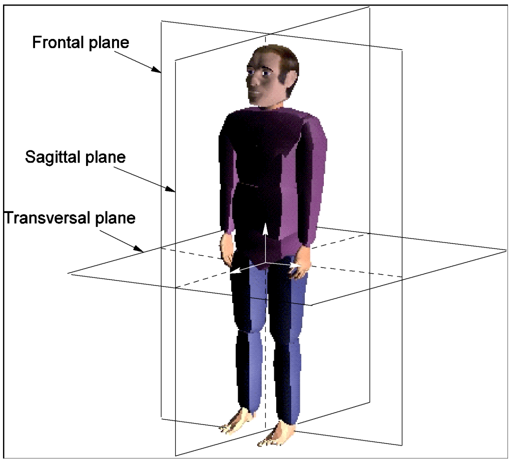


Figure 4: The three main body planes: frontal, sagittal, and transversal.

Symbol	**Value**
HandLocation = *LocFrontal LocTransversal LocSagittal*	
LocFrontal	LocAboveHead, LocHead, LocForehead, LocEyes, LocNose, LocMouth, LocChin, LocLowerChin, LocNeck, LocShoulder, LocUpperChest, LocLowerChest, LocStomach, LocBelowStomach, LocHip, LocBelowHip
LocTransversal	LocCCenter, LocBackof, LocCCenterRight, ``LocCCenterLeft, LocCenterRight. ``LocCenterLeft, LocPeripheryRight, ``LocPeripheryLeft, LocExtremePeripheryRight, LocExtremePeripheryLeft, LocLeft, LocRight
LocSagittal	LocContact, LocNear, LocNorm, LocFar, LocStreched
Handshape = *Basicsymbol [(Thumbsymbol)] (Fingersymbol)*	
Basicsymbol	BSfist, BSflat, BSffinger
Thumbsymbol	ThExt, ThAcr, ThCpart [*Degree*]
Fingersymbol	FBstr *Finger*, FBangle *Finger* [*Degree*], FBround *Finger* [*Degree*], FBroll *Finger*, FBbent *Finger*, FBstiff *Finger*
Finger	ll, p, m, r, l
Degree	c, no, o, wo
ExtendedFingerOrientation = *Direction*	
PalmOrientation = *Direction* | *RelativeRotation*	
Direction	DirL, DirLU, DirLD, DirR, DirRU, DirRD, DirU, DirD, DirA, DirAL, DirAR, DirAU, DirAD, DirT, DirTL, DirTR DirTU, DirTD
RelativeRotation	PalmL, PalmLU, PalmLD, PalmR, PalmRU, PalmRD, PalmU, PalmD, PalmA, PalmAL, PalmAR, PalmT, PalmTL, PalmTR, PalmAU, PalmAD, PalmTU, PalmTD

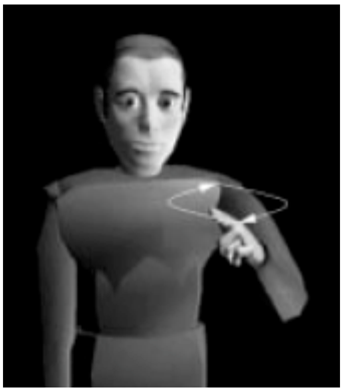
Symbolic values for the four components of hand-arm configuration descriptions.

The general idea of a MURML gesture representation is that the stroke phase can be considered as an arbitrarily complex combination of features, each of which, in turn, is a certain configuration or movement within the abovementioned components. To reproduce a gesture means to reproduce these mandatory features, and I refer to them as *movement constraints*. MURML provides two different types of movement constraints for specifying a feature over a certain period of time (the gesture definition in Fig. 3 contains examples of both of them):

(1) a *static* constraint defines a postural feature that is to be held for a certain period of time
(2) a *dynamic* constraint specifies a significant movement within a feature that is fluently connected to the adjacent movement phases (except when a hold becomes necessary)

Dynamic constraints describe (sub-)movements that are made up out of a number of segments (defined using DynamicElement tags). Depending on the component in which the movement takes place, a segment is given by start and end values for hand shape, ext. finger orientation, and palm orientation (as described above). For hand location, i.e. movement of the hand through space, the form of the trajectory becomes important and each segment is defined by its type attribute to be either linear or a curve. Each individual segment is then specified by value nodes that define, for linear segments, the start and the end position, or the start position along with direction and distance (gesture_2 in Fig. 3 comprises only one linear segment). For curved segments, value nodes must be present giving the start position, the end position, the normal vector of the movement plane, and the overall shape (C- or S-shaped to the right or left). Optionally, the form of the curvilinear segment can be concretized by giving values for the extension (the degree of curvature, from nearly straight to semicircle), the roundness (from nearly rectangular to nearly triangular), and the skewness (leveled at the beginning or the end). Using such customizable segments as building blocks, a large variety of trajectories can be assembled. For example, a horizontal circular movement (e.g. an iconic gesture for a helicopter) could be defined as follows:

```
<dynamic slot="HandLocation">
   <dynamicElement type="curve">
      <value type="start"  name="LocShoulder LocCenterLeft LocFar"/>
      <value type="end"    name="LocShoulder LocCenterLeft LocNorm"/>
      <value type="normal" name="DirU"/>
      <value type="shape"  name="LeftC"/>
      <value type="extension" name="0.6"/>
   </dynamicElement>
   <dynamicElement type="curve">
      <value type="start"  name="LocShoulder LocCenterLeft LocNorm"/>
      <value type="end"    name="LocShoulder LocCenterLeft LocFar"/>
      <value type="normal" name="DirU"/>
      <value type="shape"  name="LeftC"/>
      <value type="extension" name="0.6"/>
   </dynamicElement>
```



Note that the timing of single movement constraints can but need not be defined in the MURML spec. These values are derived at a later time from the phoneme timings of synthetic speech. While start and end times for static constraints, then, can be assigned directly, the respective timing of a dynamic constraint will lead to different start and end times of individual segments. Currently, ACE simply allots the same temporal portion to all of them.

Now that static postures as well as dynamic movements can be defined in single components, the overall structure of a gesture results from the relationships between these movement constraints, e.g., moving the hand up while keeping a fist. To compose gestures in such a way, MURML provides XML tags for expressing *simultaneity*, *posteriority*, *repetitions*, and *symmetry* of movement constraints, constituting a constraint tree for the whole gesture. The following table briefly itemizes the elements along with their possible content elements and attributes:

Element	Content elements	Attributes
parallel	symmetrical, repeat, repeat_alt, sequence, static, dynamic	start, end
sequence	symmetrical, parallel, static, dynamic, repeat, repeat_alt	start, end
symmetrical	parallel, sequence, static, dynamic	dominant, symmetry, center
repeat	symmetrical, parallel, sequence, dynamic	number

In defining symmetric two-handed gestures, MURML distinguishes between the movement of the dominant and the following hand (analogue to HamNoSys). Eight different symmetries as value to the symmetry attribute are possible, and can be defined as combinations of the possible mirror symmetries w.r.t. the main body planes of the agent (frontal, transversal, and sagittal; see Fig. 4). The following table explicates how the configuration of the following hand follows from the permutations of spatial directions of the dominant one (left-right, up-down, forward-backward) due to each symmetry.

Ident	**Symmetries**	**HandLoc**	**EFO and PO**
Sym	equal	-	-
SymMS	symm. sag.	r-l	r-l
SymMT	symm. trans.	u-d	r-l, u-d
SymMF	symm. front.	f-b	r-l, f-b
SymMST	sag., trans.	r-l, u-d	r-l, u-d
SymMSF	sag., front.	r-l, f-b	r-l, f-b
SymMTF	trans., front.	u-d, f-b	r-l, u-d, f-b
SymMSTF	sag., trans., front.	r-l, u-d, f-b	r-l, u-d, f-b

Consider ‘gesture_2’ in Fig. 3 as an example: the symmetric gesture is defined by describing the dominant right hand (<symmetrical dominant=”right_arm” symmetry=”SymMS”>). It is defined to move while keeping a static wrist orientation and a static hand shape in parallel. The left hand movement results from the mirror symmetry w.r.t. the sagittal plane. As for timing, this gesture is defined to be co-temporal to the verbal phrase “this big” being its affiliate.

Final Remarks

The representation described here is meant to interface between the microplanning and surface realization stages of a multimodal generation pipeline, where surface realization also may include the addition of non-intentional behaviors based on meta-information about the clausal and informational structure. It has been quite successfully employed in a row of systems so far. In particular the approach to specify a gesture in terms of its essential morphological features has proven useful in several applications: In a setup where a system for gesture recognition was connected to ACE, and gesture specifications in MURML were sent from the perception to the action module, real-time gesture mimicking was successfully realized for a wide range of static gestures (Kopp et al., 2003). In ongoing work, an integrated multimodal microplanner is being developed for the NUMACK agent (Kopp et al., 2004). This microplanner can come up with potentially novel iconic gestures in combination with appropriate verbal constructs. Gestures are composed and specified as typed feature structures, converted into MURML in a straightforward way, and passed on to ACE for multimodal realization. As far as we have seen from an empirical study on spontaneous gesture in direction giving, this representation promises to be powerful enough for specifying almost all of the iconic gestures used in such descriptions. Finally, in a system where we have equipped the *Max* agent with a multimodal dialogue system to serve as a presentation agent with smalltalk capabilities in a public computer museum, the combination of feature-based MURML specs with a model-based animation engine proved flexible and powerful enough to create most of the communicative gestures on the fly. However, and not surprisingly, it became evident that this approach is not always needed, nor the best way to model stereotyped behaviors. For example, symbolic gestures or facial gestures like grimaces are better defined as keyframe animations which—though entailing more modeling effort—also allows for more natural behaviors.

References

Cassell J, Vilhjalmsson H, Bickmore T. BEAT: the behavior expression animation toolkit. In Proceedings of SIGGRAPH 2001, 2001; pp 477–486.

de Ruiter JP. Gesture and speech production. PhD thesis, University of Nijmegen, 1998. MPI Series in Psycholinguistics

Kendon A. Gesticulation and speech: two aspects of the process of utterance. In The Relationship of Verbal and Nonverbal Communication, Key MR (ed.). Mouton: The Hague, 1980; 207–227.

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

S. Kopp, I. Wachsmuth. Synthesizing Multimodal Utterances for Conversational Agents. Computer Animation and Virtual Worlds 15(1): 39-52, 2004.

S. Kopp, T. Sowa, I. Wachsmuth: Imitation games with an artificial agent: From mimicking to understanding shape-related iconic gestures. In Camurri, Volpe (Eds.): Gesture-Based Communication in Human-Computer Interaction (LNAI 2915), pp. 436-447, Berlin: Springer-Verlag, 2004.

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

Levelt WJ. Speaking. MIT Press: Cambridge, MA, 1989.

McNeill D. Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press: Chicago, 1992.

Nobe S. Where do most spontaneous representational gestures actually occur with respect to speech? In Language and Gesture, McNeill D (ed.). Cambridge University Press: Cambridge, UK, 2000.

S. Prillwitz, R. Leven, H. Zienert, T. Hamke, and J. Henning. 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, Signum Press, Hamburg, Germany, 1989.

Reiter, E. & Dale, R. Building Natural Language Generation Systems. Cambridge, UK: Cambridge University Press, 2000.