An Anthropomorphic Assistant for Virtual Assembly: Max

Stefan Kopp    Bernhard Jung

AG Wissensbasierte Systeme
Technische Fakultät, Universität Bielefeld
Postfach 100 131, D-33501 Bielefeld, Germany
e-mail: {skopp, jung}@TechFak.Uni-Bielefeld.DE

1 The Virtual Agent Max and his Environment

The inclusion of nonverbal modalities into the communicative behavior of virtual anthropomorphic agents has moved more and more into focus of human-computer interface researchers. Besides the increased robustness of communication, humans are more likely to consider computer-generated figures lifelike when appropriate nonverbal behaviors are displayed in addition to speech. This in turn, enables the evocation of social communicative attributions to the artificial interlocutor, e.g. social responses, behaviors and internal states [Cas99], which increases efficiency and smoothness of human-human communication. Hand-arm gestures and facial expressions, which are an integral part of human-human dialogues and therefore ingredients of practiced communicative skill, are first candidates for extending the communicative capabilities of such virtual agents. In earlier work we discussed the coordination of spoken and gestural utterances in humans and how related research results from the cognitive sciences might be transferred to animation of virtual humans [KW99].

![Figure 1](image1.png)

Figure 1: (a) Max, the Multimodal Assembly eXpert (b) The agent’s underlying kinematic skeleton, exposed during a pointing gesture.

In this paper, we present Max, the Multimodal Assembly eXpert (Fig.1(a)), a virtual anthropomorphic agent which combines facial and upper limb gestures with...
spoken utterances yielding natural multimodal behaviors. The task of our agent Max is to demonstrate to the user the assembly of complex aggregates using gesture and speech. Max is situated in a virtual environment for simulated assembly tasks, the Virtual Constructor. The Virtual Constructor, as a stand-alone system, is a knowledge-based simulation environment for the interactive assembly of complex aggregates from CAD-based mechanical objects. It supports the simulation of various assembly-related operations such as assembly and disassembly of parts and rotation of subassemblies w.r.t. other components of a larger aggregate. The user may instruct the system in natural language or may use the mouse or similar input devices for direct, snap-to-fit enhanced manipulation of the 3D scene. See [JHW98] for a more detailed description of the Virtual Constructor’s assembly simulation capabilities. By placing Max in this environment, the user meets a communication partner with expert knowledge about the construction of certain predefined assembly groups. For example, Max might be instructed “Show me how to build a propeller”. If a propeller can be built from the currently available parts in the virtual environment, Max will demonstrate the necessary assembly steps by producing verbal utterances like “insert this bar into that screw” which will be accompanied by gesturing, e.g. pointing (Fig.1(b)) or two-handed gestures for indicating rotation of parts. Max will also send messages to the assembly simulation system such that the verbalized steps are actually performed in the virtual environment.

![Propeller Diagram](image)

**Figure 2:** The propeller of a toy airplane (left) and its representation in the Frame-based language COAR (right).

## 2 Planning the Demonstration of Assembly Tasks

As the task of our agent Max is to demonstrate to the user the assembly of complex aggregates, we also had to provide Max with some planning capabilities. Max’s assembly planner exploits a further interesting feature of the Virtual Constructor, namely its ability to maintain a dynamic conceptual model of the assemblies built in the virtual environment. For example, Figure 2 shows the propeller of a toy airplane built from the parts of a children’s construction kit and a conceptual description of the propeller in the Frame-based representation language COAR. Inferences in the COAR-framework include aggregate conceptualization, by which aggregates built in the virtual environment are recognized as instances of predefined assembly groups such as the propeller; and role assignment which reclassifies representations of aggregate components according to their use in larger assemblies, for example a three-hole-bar would be reclassified as propellerblade when used as part of a propeller [Jun98]. These inferences enable the user to refer in verbal instructions not only to spatial and visual properties of objects but also to currently assembled aggregates and functional roles of objects.

Max’s plans for demonstration of assembly tasks consist of *utterance steps* which
specify Max's gesture and language production when explaining an assembly step and action steps which specify assembly instructions to the simulation system. The planner is based on the same COAR-representations used so far for recognition of assemblies in the virtual environment. In the first planning phase, an assembly plan containing only action steps is built by matching the required parts in the COAR definition of an assembly group against the currently available parts in the virtual environment. If the COAR definition requires two of these parts to be connected an action step requiring the connection of these two parts is inserted into the plan. If the COAR definition requires two parts to be oriented in a certain spatial relation, e.g. parallel, an action step achieving the rotation is generated. In the second planning phase, a multimodal plan is generated which not only includes action steps but also utterance steps that describe verbal and gestural utterances. The multimodal utterance steps are specified in two alternative formats: a text field contains a string description of the verbal output including XML-style annotations for gestural expressions; this format is intended for use when a text-to-speech system is available. Alternatively, the playback field names an utterance specification which in turn contains, among other information, a pointer to a prerecorded sound file for playback; the additional arguments in the playback field specify the target objects for pointing and other gestures (see Sec. 3). Figure 3 shows the multimodal plan generated for Max's task to demonstrate the assembly of a propeller. Note that it is very well possible that a certain assembly cannot be built from the parts currently available in the virtual environment. Then, no assembly plan can be generated. A multimodal plan however is always generated. If an assembly cannot be built in the current state of the virtual environment, a plan is generated that will cause Max to shrug while verbally reporting this failure to the user.

3 Plan Execution and Figure Animation

Max's execution of multimodal plans is tightly integrated with the animation and the rendering of the agent (Fig. 4). To this end, Max executes the single steps of the
plan subsequently, distinguishing between action steps, which define formally the object manipulations to be done by the Virtual Constructor, and utterance steps, which specify the according multimodal utterance to be performed. While action steps are executed instantaneously by sending messages to the Virtual Constructor, the execution of utterance steps requires the production of appropriate utterances in each modality, as well as cross-modal coordination in order to maintain the life-likeness and believability of the virtual agent [KW99]. For example, the onset of the stroke, which is the most meaningful and mandatory part of a gesture, is known to covary in time with the most contrastively stressed syllable in speech [deR98]. Therefore, temporal coordination requires additional features of the gesture like duration and completion times of strokes. These have been determined in advance from prerecorded spoken utterances and are made available to Max in a database of utterance specifications.

Max, hence, employs a hierarchical motion control (see Fig.4): A higher level motion control component, first, retrieves the detailed utterance specifications as specified by the multimodal plan. Then, separate lower control modules for generating movements of the upper limbs, the hands, and the face are employed in a coordinated fashion, yielding a set of dedicated timed animations which are fed into a central animation queue.

Figure 4: Plan execution and figure animation are integrated into a main control loop.

**Body Control:** Max is based on a simplified articulated model of the human skeleton. The underlying kinematic structure, as shown in Fig.1(b), comprises 43 DOF in 29 joints of the main body and 17 DOF in 15 joints of each hand. The agent's body and hand movements are constrained by a set of key postures, each defined by a set of joint angle values. Inbetween postures are calculated by means of parametric interpolation satisfying temporal constraints (as defined in the utterance specifications) as well as continuity constraints. Natural movement kinematics is simulated by creating "slow in/out" effects during interpolation. Key posture generation aims at reproducing the definite spatiotemporal properties of gestures, which enable humans to distinguish them from subsidiary movements.
and to recognize them as meaningful. Those mandatory gesture features can be retrieved either from the actual situation, e.g. the location of the referent for deictic gesture, or further representations of gesture knowledge, e.g. the conventionalized hand shape during pointing or the preferred ways of pointing [deR98]. Therefore, our agent is equipped with both, knowledge about its actual environment, in particular about present objects and their locations, and a gesture lexicon which contains qualitative descriptions of spatial gesture features using a symbolic gesture notation system, HamNoSys [PLZ+89] (Fig.4). It is important to note, that during gesture generation, information of both kinds has to be taken into account, e.g. hand shape during pointing and target of the pointing gesture. Calculation of key postures of the upper limbs exploits available the knowledge about current environmental conditions and is based on real-time inverse kinematics techniques. Our algorithm takes the desired end-effector position and orientation of arbitrary kinematic chains as inputs and converges in a series of iterations steps against a stable state corresponding to a geometrically correct solution. Posture selection in the inverse kinematics algorithm also takes into account limitations of the joint space, like those due to restrictions of the human body, as well as biomechanical heuristics in form of cost functions yielding the adoption of natural body postures.

**Hand Control:** Hand movements are parametric transitions between hand postures which in turn are generated from the symbolic HamNoSys descriptions. Note that hand location and orientation are controlled solely by the arm control component.

**Face Control:** The face model simulates muscle activity by moving affected vertices of the head geometry. Hence, facial expressions and movements can be defined, on the higher level, by alterations of tension and relaxation values for a set of face muscles. In addition, this vertex displacement mechanism allows for lip shapes and jaw movements during spoken utterances, which are controlled by a phoneme sequence of the contemporary speech. The utterance specifications include the necessary phoneme descriptions as well as optional specifications of additional facial expressions.

All generated movements of the participating body parts and the face are timed animations which, once fed into the central animation queue, affect the face resp. different joints of Max’s body, in synchrony with the playback of the prerecorded verbal output. To this end, animations are (de-)activated in the main control loop depending on their predefined start and end time with respect to Max’s internal wall-clock time. In each iteration of the control loop, joint angle alterations are defined by active animations for the next frame time, extrapolated from current frame time and actual visualization frame rate, and are imposed on Max’s kinematic body model. Similarly, face muscle actions are controlled by face animations in a strictly timed manner.

Finally, Max’s shape, which is modelled using seamless polygonal meshes for the agent’s body, its hands, and the head (Fig.1(a)), is modified before the frame is virtually rendered. While face muscle actions are directly mapped onto skin deformations by the face model, the polygon meshes of the body and the hands are adjusted according to the movements of the underlying kinematic structure. The geometry deformation algorithm calculates relative vertex displacements as weighted sums of the altered joint transformations.

The execution of subsequent steps of the multimodal plan is deferred for the current utterance’s duration (as defined in the plan step’ utterance specification file).
4 Conclusions

We have presented Max, an anthropomorphic virtual agent which combines body and facial gestures with prerecorded verbal utterances, in order to provide multimodal explanations of the assembly of complex aggregates in a virtual environment. Exploiting background knowledge about such aggregates, Max constructs multimodal plans on the fly which, besides specifying a assembly procedure for the inquired aggregate, include descriptions of appropriate verbal and gestural utterances. Taking such a plan as input, Max generates movements of the body, the hands, and the face, by reproducing mandatory spatiotemporal features for each unimodal utterance and achieves a natural cross-modal integration by taking into account temporal synchrony constraints. While spatial features of facial and hand gestures are retrieved from explicit representations, e.g. in symbolic gesture notations, temporal coordination is based on annotations to the currently prerecorded verbal utterance. Our mid-range goals include the integration of text-to-speech and speech-synthesis techniques as well as run-time extraction of temporal constraints for control and coordination of gesture and speech. At the time of this writing, work on the skin deformation algorithms is still underway but we expect to show first results at the workshop.

Acknowledgements: We would like to thank Timo Alteseuer, Christian Becker, Björn Knaf, Alfred Kranstedt, Thies Pfeiffer, Jörg Wulstmathe, Matthias Weber, Tim Scheele, and Jure Zakotnik.

References


