A Real-world Architecture for the Synthesis of Spontaneous Gesture

Patrick Olivier∗
Newcastle University
p.l.olivier@ncl.ac.uk

Daniel G. Jackson
Newcastle University
d.g.jackson@ncl.ac.uk

Chris Wiggins
Lexicle Limited
cw@lexicle.com

Abstract
Gesture synthesis in embodied conversation agents (ECAs) can only be achieved by means of a multilevel integration of the linguistic representation of a spoken utterance and the graphical model of the character. We describe a fully implemented, deployed, and configurable ECA which utilise phonetic, syntactic, dialog and semantic information pertaining to an utterance in the real-time generation of gesture. LEXICLECS is the first commercially deployed system to include such a deep integration of language and graphics in the generation of spontaneous gestures.

Keywords: embodied conversation agents, gesture synthesis, computational linguistics

1 Gesture synthesis in ECAs
Embodied conversational agents (ECAs) are synthetic characters (i.e. full body graphical or physical simulations of people) that can maintain a conversation with a user. In theory communication with a fully enabled ECA is mediated through two or more channels. Of primary importance is the speech channel that mediates spoken natural language to and from the user. However, the visual channel is also conveyed important information in the form of facial expressions and spontaneous gestures with the arms and hands. The role of gestures in an ECA is much the same as in a real conversation, both complementing and supplementing speech to achieve robust and efficient communication [1].

There has been a significant amount of research in recent years into the requirements and technology underlying multimodal interfaces, in which speech, gestures, and other modalities, can be used in the both input and output channels. We concern ourselves here with the generation problem alone. Whilst a significant amount of effort has been placed on the multimodal generation, this has mostly been concerned with the problem of coordinating language (where written or spoken) and static graphics or 2D non-character animations. The problem of automatically generating coordinated gesture in a synthetic character has received very little attention.

For research where a key issue is the grounding of the semantics of the language used by a user and an interface character, some attention has been successfully given to the problem of generating deictic references. That is, having characters generate pointing references to resolve or emphasize real-world (or more typically graphical) entities corresponding to linguistic references made in a dialog [2]. At the same time, and somewhat separately from the multimodal interfaces community, a significant research effort in the graphics community has continued into the problems of character animation, for both the face and body [3] [4]. However, most of this work has been driven by the desire to produce what amount to productivity tools for animators, which though a worthy goal in itself, falls short of the requirements of ECAs.

∗http://homepages.cs.ncl.ac.uk/p.l.olivier/
in which the actual classes of bodily motions and facial expressions themselves must be identified and not just modeled and animated. In relation to ECAs, or synthetic characters in general, the industrial state-of-the-art for generating spontaneous gesture is little more than the playing of canned biologically plausible animations with duration equal to or less than the accompanying speech.

One notable exception, is the work of Cassell et al [5] (see also [6]) in the Behavior Animation Toolkit (BEAT). BEAT was the first systematic attempt to integrate linguistic and graphics in the generation of spontaneous gestures for an ECA and uses a rule-based system to implement the initial stage of the gesture generation pipeline to mark-up a spoken utterance with respect to body and facial motion. Although there is no publicly available demonstration of BEAT, making it hard to critically evaluate, one important design goal relates to gesture mark-up and animation system modularity. That is, the separation between the component that symbolically identifies candidate gestures, and their realization by a separate animation component. To this end Kopp and Wachsmuth [7] recently presented an inverse kinematic approach for animating hand gestures which provides adaptation to temporal constraints arising from cross-modal synchrony.

LEXICLECS\footnote{Lexicle Customer Service, Lexicle Limited, http://www.lexicle.com.} is a fully deployed system (available for evaluation on the internet) in which have adopted a practically driven perspective on the process of gesture generation. Firstly we address the problem of synthesizing semantically consistent gestures through the availability of sense marked utterances, and secondly we not only consider the linguistic information as a constraint of the gestures to be generated, but properties of the underlying application and the character’s motions themselves.

2 Application context

Our account of gesture generation is in the context of a commercially deployed web enabled ECA architecture comprising LEXICLECS. The real-world nature of this solution places a number of constraints on the framework for gesture synthesis. The system architecture for LEXICLECS is shown in figure 1. The client, by which the ECA is presented to the user, runs on standard PCs, over a narrowband network connection, and the server is scalable to large numbers of concurrent users. Low-end clients restrict the character modeling framework, precluding techniques exclusively based on inverse kinematics. Conversely, a consequence of the data and processing requirements of the natural language understanding system is that it takes place on the server. The reliance of gesture generation on the linguistic analysis of user questions and system answers means that it most naturally resides as a post-processing stage of the computational linguistic pipeline on the server.

To fulfill the requirement that the ECA is usable over narrowband connection implies that the volume of data exchanged between the client and server must be minimized. This has a number of consequences, firstly the combined download for a client application and the character data has to be less than 240K to achieve an initial installation time of under 1 minute. Secondly, the prohibitive size of speech synthesis clients requires that speech synthesis must also take place on the server. Interactivity requirements mean that the data transferred from the server to the client, in response to a user query, should be minimized too. Thus the server delivers compressed plans to the client comprising: an animation schedule that references skeletal animations and morph targets included in the character package at download, and identifiers for compressed speech files that the client can request from the server. In summary, there are

![Figure 1: Client-server system architecture.](image-url)
three technological constraints that have a consequence for the gesture generation framework:

1. the requirement for the ECA to run on low-end devices (i.e. forward kinematics only);
2. the nature of the client-server architecture (i.e. NLU and speech synthesis residing on the server);
3. the restriction of narrowband connectivity (i.e. the initial character download can only contain a small and relatively fixed number of animation elements).

In the following sections we demonstrate that these constraints can be satisfied, through the formulation of gesture generation as weakly constrained search problem for transformed motions according to the linguistic properties in the character’s utterance.

3 Language processing

Language processing in LEXICLECS is cast as standard Question-Answer system, implemented using a mixture of statistical low-level analysis techniques and hand-crafted parser and matching algorithms – the processing pipeline for a user’s query is shown in figure 2. Typed user queries, in the domain of expertise of the ECA, are spell corrected, tokenized, tagged (for parts-of-speech), and parsed to a logical form capturing a shallow semantics of the question.

The character’s range of potential responses are similarly pre-processed, and a match procedure between the semantic representations of the question and possible responses is performed in the context of the preceding dialogue between the user and the ECA. As can be seen in figure 2, gesture generation takes place on the server following the completion of natural language processing pipeline (i.e. after dialogue processing). Although we do not present the full details of the natural language processing pipeline here, this simple overview helps show how the input to the gesture planner (i.e. the output of the matcher and its subsequent dialogue processing) is not the set of strings to be spoken, but for each string there is an additional set of linguistic annotations as follows:

**Phonetic properties** – system responses comprise both the answers over which semantic matching has occurred and template-based “filler” sentences generated according to dialogue progression. Thus the full set of spoken responses, corresponding to the answers, and the fully expanded set of sentences generated by the templates, can be pre-generated and the phoneme and token boundary timings passed to the gesture planner.

**Syntactic properties** – the tokenizer, tagger and parser yield the syntactic properties of the pre-generated sentences. Consequently a shallow but complete syntactic analysis of the sentences is made available to the gesture planner, and functions for retrieving relevant features from this representation are implemented in the planner itself.

**Dialogue properties** – information pertaining to the dialogue is conveyed to the gesture planner comes in two basic forms. Firstly, there are a number of different states pertaining to dialogue progression. These include recognition of a number of different dialogue states, for example, that the user is changing topic, selecting an option previously offered by the ECA, requiring clarification or a definition of a term, confused, or being rude. Secondly, the relationship between the sentences of the responses is provided, for example, whether a sentence is a filler corresponding to a topic change, a question, a statement, a repeated statement, or a reproach to the user for inappro-
prevent behavior. Both response and individual sentence properties are conveyed to the gesture planner.

**semantic properties** — as already mentioned, the set of answers that can be provided by the ECA is fixed. In addition to facilitating pre-generation of the synthesized speech (and previewing and modification of default pronunciations), fixed responses also allow the author of the content to further annotate the responses semantically by sense marking significant word deemed important by the matching algorithm (figure 3). Consequently, in addition to the surface forms of the spoken responses, disambiguated senses, based on Wordnet [8], for verbs, nouns, adjectives and adverbs are available to the gesture planner.

4 Animation framework

As has already been discussed, certain application constraints give rise to the choice of a relatively, but not unreasonably, restrictive animation framework. The requirement that the ECA should run on a low-end clients using a narrowband connection, coupled with the need for having a highly interactive client (i.e. precluding dynamic client downloads motions from a server) results in the use of a one-off character package download containing all the character animation primitives that are required, however subsequent conversations develop.

The rather restrictive technological requirements have resulted in the use of two basic, and rather conventional, animation primitives: morph target-based interpolation and skeletal animation of weighted meshes. By using conventional techniques such as these we also achieve an additional technical benefit that characters can be readily authored using standard 3D content creation packages. Figure 4 shows a character being created in 3DSstudio Max, for which the current exporters have been developed as a standard plug-in.

Facial animation, including lip synchronization and facial expressions are achieved using a base set of morph targets corresponding to (1) visemes (i.e. visual renderings of the face corresponding to the position of the lips, teeth and tongue when speaking different phonemes), and (2) facial expressions. The set of facial expression morph targets are further divided into those that are segmented with respect to the lower face, that is, morph targets that can be applied
without impacting the position of the mouth, and those that apply to the face as a whole. This is particularly important in the case of eye and eyebrow motions (e.g. eyes widening, closing, and frowning) where the facial expression must be applied concurrently with speech.

Bodily animation is achieved through a combination of standard forward kinematics, through the application of skeletal motions which in turn drive a weighted mesh corresponding to the character’s body and arms, and the overlaying of skeletal noise with the aim of achieving a further degree of biological realism. Skeletal motion fragments can further be divided into those corresponding to base motions, which themselves form an internally consistent set of motions between postures of the character. For example, there will be base motions corresponding to motion from a reclining state to a forward leaning state, from leaning forward with hands on the desk to leaning forward with hands raised, and all meaningful transitions between posture states of the character. A further set of skeletal motions can be applied through relative application, thereby overlaying them on top of base motions. Thus, through a combination of base and overlaid (i.e. relatively applied) motions, the basic limits of a character’s gestural expression is defined.

Such a rigidly sequential application of base motions, and parallel overlaying of relative motions, is clearly insufficient with regards to generating appropriately coordinated speech and gesture. Indeed such a fixed scheme is reminiscent of current game character technology in which the requirements for coordination of motion and speech are less stringent. Consequently, this basic framework has been extended to allow more flexible combination and modification of the reference motions, as follows:

- **blending** – the temporal extent of two base motions (that are sequential in the state the state chart) may be overlapped through a weighted blending of the skeletal displacements;

- **temporal scaling** – both base and overlay motions may be non-uniformly temporally scaled (i.e. speeded up and slowed down);

- **spatial scaling** – the spatial extent of overlay motion, that is, the weight of the joint angle in its combination with a base motion, may be increased or decreased.

Combinations and transformations of skeletal motions such as these cannot be applied in an ad hoc manner, and the inclusion of this level of flexibility requires the specification of a number of constraint pertaining to: the motions that may blended and the degree, and nature of the weighting function to be used, and the degree and nature of temporal and spatial scaling that may be applied to an overlay motion. The role of this configuration of a character’s motion, with respect to gesture synthesis, is discussed in the following section.

5 **Language-gesture coordination**

The availability of so many different levels of linguistic analysis of the text to be spoken, and the dialogue context in which it is delivered, presents the both opportunities and problems. On the one hand the algorithm by which spontaneous gestures are generated can be based on relatively deep linguistic principles. However, whilst the linguistic analysis at each level may be available, current theories relating how these factors interact with each other are at best nascent [9] and lack extensive empirical support. Thus any subsequent gesture generation algorithm inevitably requires us to prioritize one linguistic factor (and the associated class of gestures) over another. Whilst this might appear ad
hoc, the very formulation of the problem, that is, the post-processing of spoken language in order to apply a layer of "coordinated" gesture, is no less artificial.

Our approach to gesture generation is to postulate appropriate gestures at each level of linguistic analysis and impose a fixed prioritization between the gesture classes. We formulate our language-gesture coordination problem by considering each level of analysis to provide support for classes of gesture or motion as follows:

**Phonetic** – the phonetic analysis of an utterance, that is generated as a byproduct of speech synthesis is the data which drives lip synchronization. Though not directly relevant to the gesture generation problem, we include it here both for completeness.

**Syntactic** – syntactic properties of an utterance have a particular bearing on the generation of beat gestures. In particular, significant linguistic features such as linguistic head of the sentence, the subject and object, the syntactic categories of the tokens and their composition. We consider syntactic properties to be one of the primary drivers for beat gestures, but no other class of gesture.

**Lexical** – emblematic gestures can be thought of as directly translatable into natural language equivalents, such as the shaken head for "no" or the raised hand with palm facing forward for "goodbye". The synthesis of emblems can be treated a one of direct correspondence of gestures for particular lexical items.

**Dialog** – the posture of a character and gestures associated with asking questions, are the only aspects of the ECA that originate from the dialog function of the character’s utterances. For example, whether the function of the sentence is to set the topic, reiterate that the answer has been given before, provide a factual answer, or pose a question, can all have a bearing on the posture and base motions of the character.

**Semantics** – iconic and metaphor gestures have their origin in the meaning of the character’s utterances, or more precisely, in the conceptual model of the character that precede the actual spoken utterance itself. Since our natural language processing framework requires the ECA’s answers (though not the "fillers") to be sense marked, and the senses themselves are selected from Wordnet, icon and metaphor generation can be based on the selection of
gestures assigned to a synset or by inheritance from a gesture assigned to a synset or its hyponym.

Gesture synthesis can therefore be cast as an under-constrained search problem. On the one hand, there are the linguistic constraints identified above: the base motions corresponding to the dialog functions of the sentences comprising the answers, the tokens for which the sense marking yield iconic and metaphoric gesture within their semantic proximity in Wordnet, the assignment of lexicalized gestures for a character, and the syntactic properties of the sentence (e.g. the identification of which nouns and verbs are more likely to have associated beat gestures). Further constraints apply to the motions themselves, for example, which base motions may be blended with each other, the state chart of character postures, base and overlay motion compatibilities, and constraints on the temporal and spatial scaling that may be applied to each motion. The final constraint is timing properties of the synthesized speech itself. The search operators themselves are the selection, scaling, blending and overlaying of morph targets and motions - and there is a subsequent probabilistic filtering to control the frequency of gestures made by the ECA. Whilst this search-based formulation of the problem may be under-constrained, the order by which motions and morph targets are selected still has a significant impact on the final solution.

We impose the following priority for motion and morph target selection, based on what we contest is the natural hierarchy, that is, one in which the production of actual phonemes (through the visemes) and the ECA’s gross position take the highest priority, and beat and idle motions the lowest:

1. visemes
2. posture state (dialogue) transitions
3. metaphor/icon gesture animations
4. lexicalized gesture animations and facial motions
5. beat gesture animations and facial motions
6. idle/talking animations and facial motions

Figure 6: LEXICLECS was deployed as mortgage advisor on the first direct website in 2002 and responded to over 20,000 customer inquiries.

For an ECA with a number of different motions there is the inevitable problem of specifying all the motion-gesture (e.g. which gestures are beats), motion-motion (e.g. which base motions may be blended or overlaid with which), motion-lexicon (e.g. which motions are emblems) and motion-synset (e.g. grounding motions corresponding to metaphors and icons amongst the synsets used) constraints identified above. Parameters based on which the probabilistic filtering of gestures is performed must also be set. The assignment of these constraints corresponds to the configuration of the character and is achieved using the configuration tool shown in figure 5.

6 Closing remarks

In summary, most existing work on ECA’s suffers a conceptual shortcoming of not, with any depth, integrating the generation of character motion and syntactic, semantic, or dialogic properties of its spoken utterances. More generally, however, there is a further gap between the theoretically attractive notion of cascading levels of gesture mark-up, and the realities of resolving such a mark-up scheme in an animation plan according to the practical constraints of an ECA application and its animation framework.

The role played by the gestural behavior and facial expression of an ECA is crucial in encouraging users to accept the anthropomorphic interface, and thereby bridge the gap between a
user’s high expectations and the true functionality, of the natural language processing system they are interacting with. Indeed, although human-like natural language understanding is unlikely to be achieved in the foreseeable future, natural sounding, and moving, interface characters are our most likely means of engaging users, and motivating them to ignore the inevitable shortcomings of the language understanding systems.

LEXICLECS was deployed as mortgage advisor on the first direct website in 2002 and responded to over 20,000 customer inquiries (figure 6). The very core of our argument as to the nature of gesture generation is that it can only be achieved by means of a multilevel integration of the linguistic representation of a spoken utterances and the graphical model of the character. The gesture synthesis framework outlined here is implemented in LEXICLECS, a fully configurable ECA, and the first commercially deployed system to include such a deep integration of language and graphics in the generation of spontaneous gestures.

Acknowledgements

Thanks to Peter Stock, Iain Crampton, Suresh Manandhar and Andrew Ormsby who all also played important roles in the design and development of LEXICLECS.

References


