Towards the Integration of Synthetic SL Animation with Avatars into Corpus Annotation Tools

Ralph Elliott[†], Javier Bueno[‡], Richard Kennaway[†], John Glauert[†]

[†]School of Computing Sciences, UEA Norwich, Norwich NR4 7TJ, UK
[‡]Departamento de Ciencias de la Computación, Universidad de Alcalá, Alcalá de Henares, Spain
R.Elliott@uea.ac.uk, fjavier.bueno@uah.es, R.Kennaway@uea.ac.uk, J.Glauert@uea.ac.uk

Abstract

We outline the main features of our synthetic virtual human sign language system, JASigning. We describe how we have extended its input notation, SiGML, to allow explicit control of performance time, and we describe our initial steps on the path to integrating virtual human sign language performance into annotation tools, where it may be compared with video depicting the corresponding real human performance.

1. Introduction[1]

JASigning is the current incarnation of our earlier synthetic virtual human signing system, SiGMLSigning. Like its predecessor, the system uses SiGML as its input notation. In this paper we start with a brief overview of the system before going on to describe our recent work in comparing virtual human sign language performance with real human signing as recorded in video sequences. We describe the introduction of explicit timing features into SiGML, and the way this can be exploited when making the comparison between real and virtual human signing. Finally we describe our initial moves towards the integration of virtual human signing into sign language annotation tools, and consider briefly the benefits of this integration.

2. Background

2.1. The JASigning System

JASigning (Java Avatar Signing) is a synthetic sign language animation system. In terms of its capabilities, JASigning is very similar to the SiGMLSigning system that we developed a few years ago in the ViSiCAST and eSIGN projects (Elliott et al., 2004; Elliott et al., 2007).

Thus JASigning supports both desktop and Web applications (Figure 1) that allow the user to have a virtual human, or avatar, perform a sign language sequence described in the SiGML (Signing Gesture Markup Language) notation. The system operates in real-time, so the SiGML sequence performed by the avatar at any point in time may be selected, or even generated dynamically, in response to user interaction. The most prominent difference between JASigning and SiGMLSigning is that the earlier system could run only on Windows computer systems, whereas JASigning, whose avatar software is implemented in Java, can be deployed on multiple platforms. It is currently available on both Windows and Mac OS X systems.

2.2. The SiGML Notation

As we have said, the input notation for any JASigning application is SiGML (Elliott et al., 2004; Elliott et al., 2007),

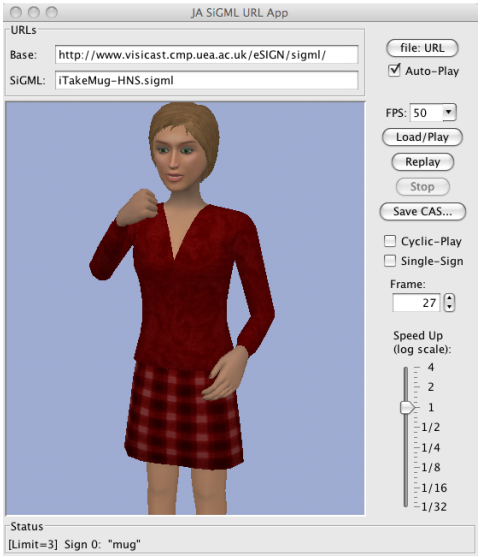

Figure 1: SiGML URL Player Application

an XML application which is based closely on HamNoSys (Hamburg Notation System) (Prillwitz et al., 1989; Hanke, 2004), and which is thus a vehicle for sign language description at the phonetic level.

The basic notions in the HamNoSys/SiGML model are those of posture and movement (transition). The manual component of a posture is characterised by its handshape, its spatial orientation, and its location in signing space — these features being specified for the dominant hand only in a single-handed sign, or for both hands in a two-handed sign. A basic movement consists of a change in some aspect of posture. These changes may be combined either concurrently, where that makes physical sense, or in sequence.

Historically, HamNoSys focused predominantly on the definition of the manual features of sign language performance, but its current version, HamNoSys 4 defines a comparatively rich repertoire of nonmanual features on different tiers, corresponding to distinct articulators such as body, eyes and mouth. SiGML follows HamNoSys 4 in including this repertoire of nonmanual features.

[1]We acknowledge with gratitude that the work described here has been partially funded under the European Union's 7th Framework Programme, through the Dicta-Sign project (grant 231135).

In many ways the HamNoSys SiGML model as just described resembles the phonetic model for sign languages of Liddell and Johnson (Liddell and Johnson, 1989), although there are some significant points of difference.
A SiGML document is structured as a sequence of individual signs. The notation allows sign language sequences to be represented in several distinct forms, of which the two most important are:

- HNS-SiGML In essence this is simply HamNoSys dressed in XML form, one element per symbol.
- Gestural SiGML This contains the same information as an HNS-SiGML or HamNoSys definition (in fact, potentially a slightly generalised version of this information), but in a more explicitly structured form, comparable to that of an abstract syntax tree for the corresponding HamNoSys or HNS-SiGML definition.

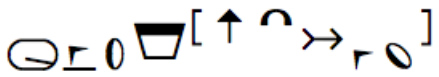

Figure 2: Manual HamNoSys Sign – "mug" in BSL

```
<hamgestural_sign gloss="mug">
  <sign_nonmanual>
    <mouthing_tier>
      <mouth_picture picture="mVg"/>
    </mouthing_tier>
  </sign_nonmanual>
  <sign_manual>
    <handconfig handshape="fist"
        thumbpos="across"
        extfidir="ol" palmor="l"/>
    <location_bodyarm
        location="shoulders"/>
    <par_motion>
      <directedmotion
          direction="u" curve="u"/>
      <tgt_motion>
        <changeposture/>
        <handconfig
            extfidir="ul" palmor="dl"/>
      </tgt_motion>
    </par_motion>
  </sign_manual>
</hamgestural_sign>
```

Figure 3: Gestural-SiGML Sign – "mug" in BSL

In Figure 2 we show the HamNoSys for the manual component of the BSL sign "mug", a snapshot of which is shown in Figure 1. The first three symbols describe the handshape and orientation, and the fourth the location (shoulder-level), for the initial posture; the remaining symbols specify a composite movement from this posture. Figure 3 shows the Gestural SiGML form of this sign. The motion from the initial posture, once attained, is a composite of two basic motions performed in parallel, that is, concurrently: an upwards curved motion of the dominant hand, and a change of hand orientation. Together these motions function iconically, tilting the hand (whose shape itself functions iconically to represent a mug) towards the signer's mouth. In the HamNoSys (and HNS-SiGML) forms the fact that these motions are performed concurrently with one another is indicated by the pair of square bracket symbols, whereas in the Gestural SiGML form the motion structure is directly reflected in the XML element structure, in which a `par_motion` element has a child element for each of the two component motions — the `directedmotion` and the `tgt_motion` (targetted motion).
SiGML can effectively be regarded as a kind of programming notation for the avatar: in principle any sign language utterance can be described in SiGML, as in HamNoSys; hence it can be performed by an avatar in the JASigning system.

2.3. Organisation of the JASigning Software

A signing avatar in the JASigning system is based on conventional 3D computer animation techniques. These techniques are augmented with additional data files defining those characteristics of the avatar that are needed for sign language performance — described in a companion paper (Jennings et al., 2010) — and with a software module, Animgen (Kennaway et al., 2007), whose function is to generate a sequence of animation frames, each defining an instantaneous posture for a specific avatar. Animgen does this given two inputs: the (avatar independent) SiGML description of the required sign language sequence, and the dataset describing the avatar for which the animation is required.

3. Working towards Integration with Annotation Tools

The synthetic sign language animation system is certainly still capable of further refinement and improvement, but it has reached a stage of maturity at which it is feasible to consider how it might be integrated into sign language annotation tools such as ELAN (Hellwig et al., 2009) and ILex (Hanke, 2004), and what the benefits of doing this might be. We outline here our recent activities in this area, the first of which involves an extension to the SiGML notation and its implementation.

3.1. Introduction of Explicit Timing into SiGML

The timing model for sign language performance used by JASigning's animation generation module, Animgen, can be described as follows. Each basic movement is assigned a supposedly "natural" duration. This is done by means of one of the avatar-specific configuration data files described in the companion paper (Jennings et al., 2010). Hence, for a given avatar it is possible to vary these individual duration values relative to one another, and also to vary some or all of these configuration parameter values from one avatar to another. In addition, a configuration parameter determines the "natural" value for the movement to the initial posture of a sign. Once fixed, these duration values for basic movements determine those for composite movements. In the case of a sequence of movements, the duration of the sequence is simply the sum of the individual component du-

ration values. For a parallel combination of movements the overall duration is the longest of the individual component durations, the other component durations being extended to that maximum value.
We have recently extended the SiGML notation, and its implementation in Animgen, to allow explicit timing characteristics to be attached both to any individual motion, whether basic or composite, and also to an entire sign.
This is done by means of an additional pair of attributes, each with a floating point value, either or both of which may be attached to any relevant Gestural SiGML component:

- `duration`, measured in seconds, whose default value is the "natural" duration value, as described above.
- `timescale`, a slow-down factor, whose default value is 1.0.

(There is a third attribute, `speed`, whose effects are identical to those given by using the `timescale` attribute with the reciprocal value, so we omit it from the following discussion.) For any motion, if its (explicit or default) `duration` and `timescale` values are, respectively, d and t, then the duration value assigned to it, a, is given by the formula:

$$a = d * t$$

(according to which, the default duration value is indeed the "natural" one).
Whenever a composite motion, including an entire sign, is explicitly given a non-standard duration value in this way, that value is propagated down the motion structure as follows. Any increase or decrease in the duration of a composite motion is propagated to each of its components in proportion to the relative durations assigned to them prior to this adjustment. Any increase or decrease in the duration of a parallel motion is applied to each of its constituent motions (which in some cases may simply be a matter of undoing, to some degree, a previously applied extension). If any constituent motion is itself composite, its new duration value is propagated recursively to its components.

3.2. Comparing Virtual and Real Sign Language Performance

Our first activity in this area consisted of an investigation of the fidelity with which the signing avatar system could reproduce some Spanish Sign Language (LSE) sequences for which video material was already available. This was partly a matter of considering the basic quality of the animation produced from a HamNoSys or SiGML transcript, and partly a matter of determining the extent to which it is possible to improve the fidelity of the animation by adjusting the SiGML transcript, usually by adding more explicit detail relating to certain aspects of the original human performance.
To compare the results with the original it is useful to have video of the real and the virtual human performance side by side. This can be achieved by converting the animation system output to a video file, which can then be imported into an annotation tool. We have done this using ELAN 4.
An important issue for the comparison is that of synchronization, or the lack of it, between the real and the virtual animation. Using the SiGML enhancements for explicit timing control just described, it is relatively simple to align the two performances temporally, as is shown in Figure 4. So far we have pursued this only to the point of aligning sign boundaries, but in principle it is possible also to align individual movement phases within signs.
More recently, we have done some work with the ILex sign language corpus annotation tool (Hanke, 2004). ILex is able to export an annotation transcript, which includes segmentation and timing data, as well as a HamNoSys transcription of each sign. From this transcipt we have been able to derive (almost) automatically a SiGML description of that sequence. When played by a signing avatar, the avatar performance exhibits some variations from that of the human signer in the video accompanying the transcript. In particular, as in the case of the LSE sequences described above, there are significant variations in the timing of the two performances.
Using the timing data from the ILex transcript, together with the new explicit timing attributes in SiGML, we have also been able to generate automatically a modified SiGML description of the sequence in which each sign is temporally aligned with its counterpart the original human performance. Thus from the exported ILex transcript we are able to produce a synthetic avatar performance – either in our avatar player, or exported from it as a video clip – which is temporally aligned, sign by sign, with the human performance.
A cursory comparison of the two performances gives rise to a couple of observations:

- The avatar makes some rather violent elbow movements, indicating scope for possible improvement of the generated animation.
- There are some variations in handshape and/or orientation, suggesting in some cases that the HamNoSys annotation may not be entirely accurate.

4. Conclusion

We have described the basic features of the JASigning synthetic signing system and the SiGML notation which is used to drive it. We have also described the introduction of explicit timing into SiGML and its implementation, and our moves towards the incorporation of virtual human signing into annotation tools, where it can be compared in detail with real human signing.
As yet, within an annotation tool (ELAN) we have augmented the original annotated video with the corresponding synthetic performance only in video form, but there is clearly no obstacle in principle to quite tight and iteractive integration into an annotation tool of the process of generating and displaying synthetic sign language performance.
On the basis of our experience to date, we can envisage several uses for such a scheme. As we have already seen, it can be used evaluate and to improve the quality of our synthetic sign language generation techniques.
Conversely, the capacity to get immediate feedback in the form of a synthetic animation provides a means of verifying

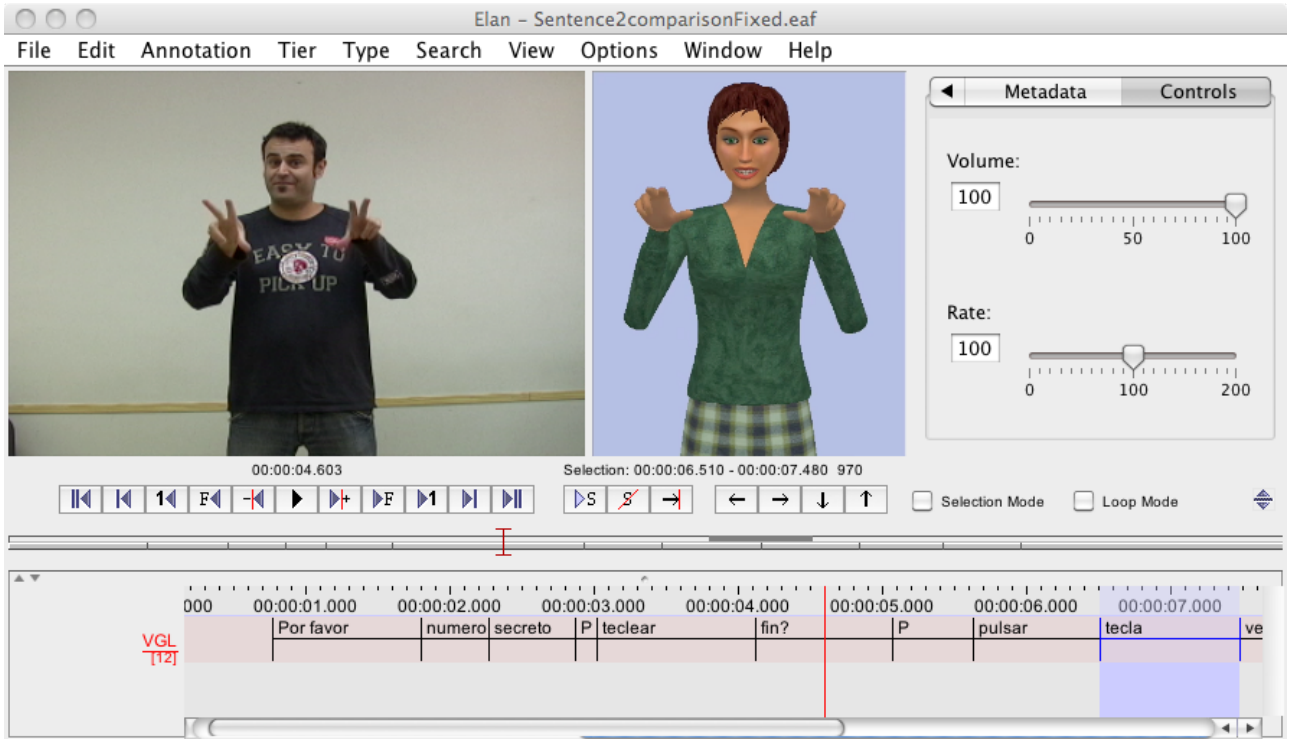

Figure 4: ELAN window with video of real and virtual human signers

the accuracy and quality of a HamNoSys transcription as soon as it has been generated. This can be useful both in the context of corpus collection and transcription, as well as in the context of signed content creation using a virtual human.

From the point of view of the kind of sign language study that annotation tools are intended to facilitate and support, the ability to compare and contrast virtual and real human sign language performance in great detail has the potential to assist in exploring more substantial questions in sign language modelling. For example, when confronted by variations between different performances of the same sign language sequence it is possible to ask whether these variations are linguistic in character, or whether they are matters of individual style or mood, whether they are peculiar to the particular utterance or part of a more persistent pattern. By taking our work further and fully integrating a synthetically signing avatar into an annotation tool, we can envisage a situation where it would be possible dynamically to modify some of the avatar's configuration parameters, for example those characterising its signing space, and exploring the way such variations cause the synthetic performance to align with or deviate from the original human performance. Experiments of this kind could help in leading to a richer characterisation — and hence annotation — of the original human sign language performance.

5. References

R. Elliott, J.R.W. Glauert, V. Jennings, and J.R. Kennaway. 2004. An overview of the sigml notation and sigmlsigning software system. In O. Streiter and C. Vettori, editors, *Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 98–104, Lisbon, Portugal.

R. Elliott, J.R.W. Glauert, R. Kennaway, I. Marshall, and E. Safar. 2007. Linguistic modelling and lanuage-processing technologies for avatar-based sign language presentation. *Universal Access in the Information Society*, 6(4):375–391.

T. Hanke. 2004. Hamnosys—representing sign language data in language resources and language processing contexts. In O. Streiter and C. Vettori, editors, *Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 1–6, Lisbon, Portugal.

Birgit Hellwig, Dieter Van Uytvanck, and Micha Hulsbosch. 2009. Elan - linguistic annotator, version 3.8.

V. Jennings, J.R. Kennaway, J.R.W. Glauert, and R. Elliott. 2010. Requirements for a signing avatar. In T. Hanke, editor, *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Valletta, Malta.

J.R. Kennaway, J.R.W. Glauert, and I. Zwitserlood. 2007. Providing signed content in the internet by synthesized animation. *ACM Transactions on Computer Human Interaction*, 14, 3(15):1–29.

S.K. Liddell and R.E. Johnson. 1989. *American Sign Language : The Phonological Base*. Linstok Press.

S. Prillwitz, R. Leven, H. Zienert, T. Hanke, and J. Henning. 1989. *Hamburg Notation System for Sign Languages—An Introductory Guide*. International Studies on Sign Language and the Communication of the Deaf. IDGS, University of Hamburg.