An Overview of the *SiGML* Notation and *SiGMLSigning* Software System

Ralph Elliott, John Glauert, Vince Jennings, Richard Kennaway

School of Computing Sciences
University of East Anglia
Norwich NR4 7TJ, UK
{re,jrwg,vjj,jrk}@cmp.uea.ac.uk

Abstract

We present an overview of the *SiGML* notation, an XML application developed to support the definition of Sign Language sequences for performance by a computer-generated virtual human, or avatar. We also describe *SiGMLSigning*, a software framework which uses synthetic animation techniques to provide real-time animation of sign language sequences expressed in SiGML.

1. Introduction

We have developed the SiGML notation (Elliott et al., 2001) to support our work in the ViSiCAST and eSIGN projects (Glauert, 2002; Glauert et al., 2004). These projects have been concerned with the development of techniques for the generation of sign language performances by a computer-generated virtual human, or avatar.

The name SiGML is an abbreviation for "Signing Gesture Markup Language". SiGML is an XML application (Bray et al., 2004). Thus, SiGML data is represented as plain text in computer systems. SiGML encompasses several data formats used at different stages in the generation of virtual human animations, but its most prominent rôle is as the interface notation used in a prototype system supporting the generation of signed animation from natural language text. This system was a major part of the ViSiCAST project; as outlined in (Elliott et al., 2000), it contains two major subsystems:

- A "front-end" which uses natural language processing techniques to translate (English) text into an equivalent Sign Language form, for which a phonetic-level description is generated.
- A "back-end" which uses 3-D animation technology (together with artificial language processing) to generate a virtual human animation from the given phonetic-level description.

The natural language subsystem is designed to support output for several different national sign languages. Thus, it divides into a common initial stage, producing a language-neutral semantic representation (using DRT), followed by a stage specific to the target sign language. The most fully developed of the latter is that for British Sign Language (BSL) (BDA, 1992), which uses HPSG as the supporting grammatical formalism. More details on this work by our colleagues, Marshall and Safar, can be found in (Safar and Marshall, 2001; Safar and Marshall, 2002b; Safar and Marshall, 2002a; Safar and Marshall, 2002c).

The interface between the two subsystems is the SiGML notation, specifically the SiGML module we refer to as "gestural" SiGML. In the following section we describe gestural SiGML in more detail, concentrating on its relation to HamNoSys, the long established notation system for sign language transcription developed by our partners at the University of Hamburg. We then give a brief overview of SiGMLSigning, the back-end software subsystem identified above. We conclude with a simple example.

2. Gestural SiGML and HamNoSys

As we have indicated, gestural SiGML is based on HamNoSys (Prillwitz et al., 1989), that is, the Hamburg Notation System. This notation has been developed to support phonetic-level transcription of sign language performance by (real) human signers, and is intended to provide a model of sign language phonetics that is independent of any particular sign language. We have developed gestural SiGML with the explicit intention of formulating a model of signing gesture production which respects HamNoSys's model of sign language phonetics. At the start of the ViSiCAST project, HamNoSys stood at version 3. In preparation for the development of gestural SiGML, an initial phase of the ViSiCAST project saw the development of HamNoSys version 4 (Hanke et al., 2000; Hanke and Schmaling, 2002). As far as the manual aspects of signing are concerned, HamNoSys 4 does not radically alter the already well-established features of HamNoSys 3, but generalises and regularises several of those features. The more prominent changes in HamNoSys 4 occur in connection with the non-manual aspects of signing, for which a far more comprehensive framework is provided than was previously available. Following HamNoSys, gestural SiGML includes both a manual component, concerned with the configuration and actions of the hands, and a non-manual component, concerned with other linguistically significant features of signing such as head movement, eye movement, eye gaze, and mouthing. In the rest of this section we outline some general features of the SiGML notation before briefly describing the two components in turn.

2.1. General Features of Gestural SiGML

Considered as XML, a valid SiGML document is a pure element hierarchy: every element is constrained by the DTD (Kennaway et al., 2002) either to have element content or to be empty, that is, no SiGML element contains any embedded text, although of course it can, and in most cases does, contain attribute definitions. A SiGML document defines a sequence of "signing units". Typically, a signing unit is an explicit gestural definition for a single sign, but it may also be a direct definition of avatar animation parameters, or an indirect reference to another SiGML document. A gestural sign definition is represented by a <hamgestural_sign> element. Since it is intended that any HamNoSys sign definition can be represented in SiGML, we also allow a tokenised form of a HamNoSys sign, represented by a <hns_sign> element. For convenience of reference each of these sign elements has a `gloss` attribute, giving a (spoken language) gloss of the sign's meaning.

2.2. Manual SiGML

The manual component of a SiGML sign is represented by a <sign_manual> element. SiGML ascribes the same general structure to the manual component of a sign as does HamNoSys: an initial configuration followed by a sequence of actions or motions, which may well themselves be composite. Each of these components may involve both hands or just one hand, usually the signer's "dominant" hand (i.e. right hand for a right-handed signer). The initial configuration is a hand configuration, together with a location for that configuration. The configuration for each hand defines its hand shape, and its orientation in 3-D space. This orientation is specified as two components: extended finger direction (the direction of the metacarpal of the index finger) and palm orientation (the rotation of the palm about the axis defined by the other component). There is a basic set of a dozen standard handshapes, such as a fist, a flat hand, and a "cee" formed by the thumb and index finger. Many variations of these can be defined by specifying adjustments to the position of the thumb, various forms of bending of some or all fingers, and specific forms of contact or crossing between pairs of fingers. Hand shapes exemplify of HamNoSys's rather "operational" approach to the structure of feature definition: a simple instance of the given feature can be specified with no more than one or two symbols, while a more complex instance is obtained by appending additional modifier symbols defining how the required instance can be obtained from a simpler one.

In general terms, the location of a hand is defined with reference to a site on the signer's body, head, arm or (other) hand, and a rough measure of the proximity of the hand to that site. With some misgivings, we have retained in SiGML the HamNoSys concept of a "hand constellation", a special form of location which allows the definition of a potentially quite elaborate configuration of the hands as a pair, with (optionally) a location of this configuration relative to the body.

SiGML structures motions in a broadly similar fashion to HamNoSys, although SiGML tends to relegate to the level of informal semantics physical constraints to which HamNoSys gives direct syntactic embodiment. There is a repertoire of primitive motions, which may be combined in temporal sequence or in parallel, that is, concurrently, to any extent that makes physical sense. In SiGML, there are two other forms of structured motion (both inspired by comparable features in HamNoSys)

- Targeted motion: a motion for which an explicit target location (possibly a hand constellation) is specified.
- Repeated motion: various forms of single or multiple repetition of a given motion.

The simplest form of motion is a straight line motion in a given direction (any of the 26 directions defined by a non-zero position vector each of whose individual 3-D coordinates is either zero or one, or half-way between two adjacent directions of this kind). A straight line motion may be modified in a wide range of ways, including changing the distance moved, and tracing a curved, wavy or zig-zag path to the given end point. Other forms of simple motion include circular and elliptical motions (again with a wide range of variants), fluttering of the fingers, and several forms of wrist motion.

2.3. Non-Manual SiGML

The non-manual component of a SiGML sign is represented by a <sign_nonmanual> element. As described in (Elliott et al., 2004), the internal structure of this element closely follows non-manual feature definitions in HamNoSys 4. Thus, non-manual actions are partitioned into a hierarchy of tiers, corresponding to distinct articulators, as follows:

- Shoulder movements
- Body movements
- Head movements
- Eye gaze
- Facial expression: Eye-Brows, Eye-Lids, and Nose
- Mouthing: Mouth Pictures and Mouth Gestures.

Here, "facial expression" refers solely to those expressive uses of the face which are phonetically significant; by contrast those uses which express the signer's attitude or emotions about what is being articulated, important though they may be, cannot at present be expressed in SiGML (nor in HamNoSys). The two forms of mouthing reflect the distinction between motion of lips and tongue caused by spoken accompaniment to signing (mouth pictures), and other phonetically significant motions of lips, tongue, jaw and cheeks (mouth gestures). A mouth gesture often has a relatively elaborate internal structure which SiGML does not attempt to reflect, instead just identifying the unanalysed whole by a single label.

3. SiGMLSigning Animation Software System

SiGMLSigning is the software system we have developed, with support from partners in the ViSiCAST and eSIGN projects, to generate virtual-human signing animations on-screen from a sign sequence specified in SiGML. Architecturally, this system can be viewed as a pipeline of three processing stages, together with a control module which coordinates and schedules the transfer of data between these stages, stores the data they generate, and provides a programmable control interface. In its current form, the software is packaged as a set of Active X controls, which allow it to be deployed relatively easily in applications and

HTML pages on Microsoft Windows systems. The three processing stages are:

- SiGML Input and Pre-processing
- Animation Generation
- Virtual Human Animation

The interface between the first two stages is a sequence of gestural SiGML sign definitions; the interface between the second and third stages is a sequence of animation parameter sets, one set for each frame in the final animation. We outline each of these stages in turn, taking them in reverse order, in order to highlight the context each stage defines for its predecessor.

The final stage uses conventional 3-D animation technology. An avatar is represented by a virtual skeleton – a connected hierarchy of virtual bones – and a surface mesh – a connected tissue consisting of thousands of small, coloured, textured polygons. The configuration of these polygons determines the appearance of the avatar. The position and orientation of every polygon is determined (as part of the avatar's definition) by the position and orientation of one or more of the avatar's virtual bones. Hence a static posture of the avatar's surface appearance is completely determined by a static posture of its virtual skeleton: standard 3-D rendering techniques, using a combination of software and special-purpose graphics hardware, can be relied on to produce the one from the other. So, an animation of the avatar is defined simply by the appropriate sequence of static skeleton configurations, one for each animation frame (typically at the rate of 25 fps). A refinement of this system allows the avatar's appearance (in each frame) to be further modified by applying predefined distortions, known as morph targets or morphs, directly to the surface mesh. This technique is especially useful to us in defining facial non-manual gestures. The supplier of an avatar must therefore provide, as a minimum, a description of the physical structure of the avatar's skeleton and a list of its available morphs, together with a simple rendering interface which (i) allows a skeleton configuration to be specified (together with morph weights, if required), and (ii) accepts a request to render the corresponding posture.

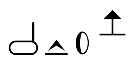
The preceding stage, at the heart of the SiGMLSigning system, is the animation generation stage, performed by a module called AnimGen. This maps a given sequence of gestural SiGML sign descriptions to the corresponding stream of avatar animation parameters. This stream is avatar-specific, since it depends crucially on the definition of the avatar's physical characteristics provided by the avatar supplier. Indeed, we have found that avatar-independent sign synthesis depends crucially on the specification by the avatar supplier of of the locations (relative to the skeleton) of quite a large number of sites on the avatar's surface mesh, in addition to the basic physical characteristics already mentioned. The task of this stage, therefore, is to derive precise numerical animation parameters from the physically relatively imprecise SiGML sign definitions. The manner in which this is done currently, and some of the issues that arise, have been described more fully elsewhere (Kennaway, 2001; Kennaway, 2003; Elliott et al., 2004).

The first processing stage performs relatively straightforward pre-processing of the SiGML input. Its most basic function is to decompose this input into individual sign definitions, so that each can be handled in the appropriate manner: the <hamgestural_sign>s can be fed directly to the AnimGen stage, the <hns_sign>s are first passed through a HamNoSys-to-(gestural-)SiGML translator, while those containing pre-generated animation data are just converted directly to the internal stored format output by the AnimGen stage, which is by-passed in this case. The HamNoSys-to-SiGML translation takes the form of an additional processing pipeline: conventional context-free parsing techniques (augmented with backtracking to account for HamNoSys's many syntactic ambiguities) are used to generate a syntax tree, which is then transcribed into an intermediate XML form, called HamNoSysML or HML; gestural SiGML is then generated from this using an XSLT transform (Clark, 1999; Kay, 2000).

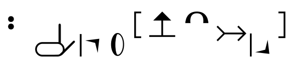
The SiGMLSigning software system is thus a "scriptable", virtual human signing animation system, accepting as input arbitrary signing sequences expressed in SiGML, and providing the corresponding animation on any avatar which supports the simple rendering interface described above. Finally, it is noteworthy that the core animation module, AnimGen, generates frames at a sufficiently high rate that the animation appears almost instantaneously in response to the SiGML input.

4. A Simple Example

The following is the HamNoSys sequence for a very simple gesture (which does not represent any actual sign):



Here, the first symbol specifies the hand shape, a fist with the index finger extended, the second and third symbols specify the orientation of the hand: the index finger points outwards from the signer's body, with the palm facing to the left; no initial location is explicitly specified for the hand, so a default, neutral, position in front of the signer's body is assumed; the final symbol specifies a straight movement from this initial position in an outwards direction, that is, away from the signer's body. The insertion of a few more symbols into this example results in a genuine sign, namely the DGS (German Sign Language) sign "going-to":



Here, the hand shape has a modifier specifying that the thumb is extended, the initial finger direction is now upwards-and-outwards, the outward motion has an upward arc modifier attached to it, and this motion is composed in parallel with a change of finger direction to downwards-and-outwards. The whole is prefixed with a symbol specifying motion of both hands in parallel, with the initial configuration of the non-dominant hand mirroring that of the explicitly specified dominant hand. The HNS-SiGML form of this is:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE sigml SYSTEM .../sigml.dtd>
<sigml>
<hns_sign gloss="DGS_going-to">
  <hamnosys_manual>
    <hamsymmpar/>
    <hamfinger2/>
    <hamthumboutmod/>
    <hamextfingeruo/>
    <hampalml/>
    <hamparbegin/>
    <hammoveo/>
    <hamarcu/>
    <hamreplace/>
    <hamextfingerdo/>
    <hamparend/>
  </hamnosys_manual>
</hns_sign>
</sigml>
```

This is parsed during the input/pre-processing stage into the intermediate HML form shown (at the end of the paper) in Figure 2. In this easily generated but rather verbose format, an element typically corresponds to a HamNoSys syntactic category, while an attribute typically corresponds to an individual HamNosys symbol, although the HamNoSys parallel composition brackets and the HML `<paraction1>` elements provide a counter-example to this general rule of thumb.

The XSLT translation which is applied to the HML form shown in Figure 2 produces the much flatter Gestural SiGML form shown immediately below:

```
<sigml>
<hamgestural_sign gloss="DGS_going-to">
<sign_manual both_hands="true">
  <handconfig handshape="finger2"
    thumbpos="out"/>
  <handconfig extfidir="uo"/>
  <handconfig palmor="l"/>
  <par_motion>
    <directedmotion direction="o"
      curve="u"/>
    <tgt_motion>
      <changeposture/>
      <handconfig extfidir="do"/>
    </tgt_motion>
  </par_motion>
</sign_manual>
</hamgestural_sign>
</sigml>
```

The synthetic animation module, AnimGen, pre-processes this Gestural SiGML into a more explicit form of SiGML in which the hand-shape information is reduced to numerical measures of joint angles (on a scale of 1 to 4), and the rôle of both hands is made explicit. This explicit form is shown (at the end of the paper) in Figure 3.

The stream of animation data output by AnimGen is extremely voluminous, and is usually passed directly from the computer system's internal memory to the avatar rendering module. However, if desired, this data stream may be recorded for future reference in a file, in which case it is stored in SiGML's CAS (Character Animation Stream) format. A few lines of the output for our "going-to" example on the VGuido avatar, developed by our eSIGN project partner Televirtual, is shown in Figure 4 below.

The animation generated for this sign in isolation has a duration of about 320ms (preceded by another 320ms while the avatar's hands move from the rest position to the initial position of the sign itself. In Figure 4. below we show the animation frames for the start and finish of this sign.

Acknowledgements

We acknowledge with thanks financial support from the European Union, and assistance from our partners in the ViSiCAST and eSIGN projects.

5. References

BDA, 1992. *Dictionary of British Sign Language*. Faber and Faber.

Bray, T., J. Paoli, C.M. Sperberg, E. Mahler, and F. Yergeau (eds.), 2004. Extensible Markup Language (XML) 1.0. http://www.w3.org/TR/REC-xml/.

Clark, J., 1999. XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt.

Elliott, R., J.R.W. Glauert, and J.R. Kennaway, 2004. A framework for non-manual gestures in a synthetic signing system. In *Proc. Cambridge Workshop Series on Universal Access and Assistive Technology (CWUAAT)*.

Elliott, R., J.R.W. Glauert, J.R. Kennaway, and I. Marshall, 2000. Development of language processing support for the ViSiCAST project. In *ASSETS 2000 4th International ACM SIGCAPH Conference on Assistive Technologies*. Washington DC, USA.

Elliott, R., J.R.W. Glauert, J.R. Kennaway, and K.J. Parsons, 2001. D5-2: SiGML definition. Working document, ViSiCAST Project.

Glauert, J.R.W., 2002. ViSiCAST: Sign language using virtual humans. In *International Conference on Assistive Technology ICAT 2002*. Derby: BCS.

Glauert, J.R.W., J.R. Kennaway, R. Elliott, and B-J. Theobald, 2004. Virtual human signing as expressive animation. In *Proc. AISB-2004: Symposium on Language, Speech and Gesture for Expressive Characters*.

Hanke, T., G. Langer, C. Metzger, and C. Schmaling, 2000. D5-1: Interface definitions. working document, ViSiCAST Project.

Hanke, T. and C. Schmaling, 2002. HamNoSys 4 (Course Notes). http://www.sign-lang.uni-hamburg.de/Projekte/HamNoSys/HNS4.0/HNS4.0eng/Contents.html.

Kay, M., 2000. *XSLT – Programmer's Reference*. Wrox Press Ltd.

Kennaway, J.R., 2001. Synthetic animation of deaf signing gestures. In *4th International Workshop on Gesture and Sign Language Based Human-Computer Interaction*, LNAI. Springer-Verlag.

Kennaway, J.R., 2003. Experience with and requirements for a gesture description language for synthetic animation. In *5th International Workshop on Gesture and Sign Language Based Human-Computer Interaction*, LNAI, to appear. Springer-Verlag.

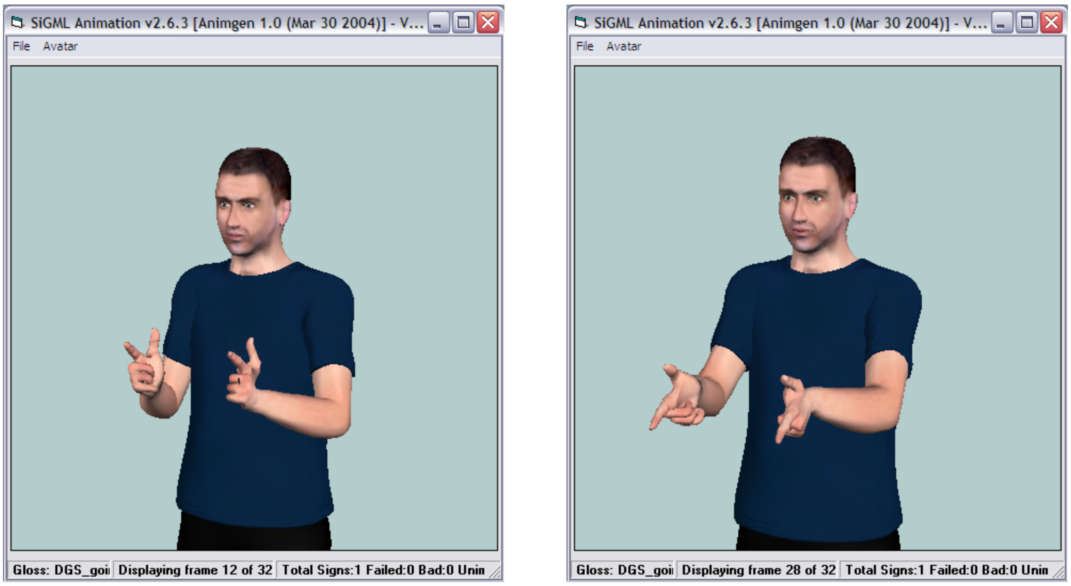


Figure 1: Animation frames for the “Going-To” Example.

Kennaway, J.R., R. Elliott, J.R.W. Glauert, and KJ Parsons, 2002. SiGML Document Type Definition (DTD). http://www.visicast.cmp.uea.ac.uk/sigml/sigml.dtd.

Prillwitz, S., R. Leven, H. Zienert, T. Hanke, J. Henning, et al., 1989. *HamNoSys Version 2: Hamburg Notation System for Sign Languages — An Introductory Guide*, volume 5 of *International Studies on Sign Language and the Communication of the Deaf*. SIGNUM Press, Hamburg.

Safar, E. and I. Marshall, 2001. Translation of English Text to a DRS-based, Sign Language Oriented Semantic Representation. In *Conference sur le Traitment Automatique des Langues Naturelles (TALN)*, volume 2.

Safar, E. and I. Marshall, 2002a. An Intermediate Semantic Representation Extracted from English Text For Sign Language Generation. In *7th Symposium on Logic and Language*.

Safar, E. and I. Marshall, 2002b. Sign Language Synthesis using HPSG. In *9th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

Safar, E. and I. Marshall, 2002c. Sign language translation via DRT and HPSG. In *3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.

```
<hamnosysml>
<sign gloss="DGS_going-to">
<hamnosys_sign>
<sign2>
  <symmoperator att_par_or_lr="hamsymmpar"/>
  <minitialconfig2>
    <handconfig2>
      <handshape2>
        <handshape1 handshapeclass="ham_finger2" thumbpos="ham_thumb_out"/>
      </handshape2>
      <extfidir2>
        <extfidir1 extfidir="direction_uo"/>
      </extfidir2>
      <palmor2>
        <palmor1 palmor="ham_palm_l"/>
      </palmor2>
    </handconfig2>
  </minitialconfig2>
  <action2t>
    <action1t>
      <action1>
        <par_action1>
          <action1>
            <simplemovement>
              <straightmovement
                arc="ham_arc_u" movement="ham_move_o"/>
              </simplemovement>
          </action1>
          <action1>
            <simplemovement>
              <replacement>
                <extfidir1
                  extfidir="direction_do"/>
              </replacement>
            </simplemovement>
          </action1>
        </par_action1>
      </action1>
    </action1t>
  </action2t>
</sign2>
</hamnosys_sign>
</sign>
</hamnosysml>
```

Figure 2: Intermediate HML form for the “Going-To” Example.

```
<sigml/>
<hamgestural_sign gloss="dgs_going-to">
  <sign_manual both_hands="true">
    <handconfig handshape="finger2" thumbpos="out"
        bend2="0.00 0.00 0.00 0.00"
        bend3="4.00 4.00 4.00 0.00"
        bend4="4.00 4.00 4.00 0.00"
        bend5="4.00 4.00 4.00 0.00"
        bend1="-0.30 2.20 2.20 0.30 0.00"   />
    <split_handconfig>
      <handconfig extfidir="uo" palmor="l"/>
      <handconfig extfidir="uo" palmor="r"/>
    </split_handconfig>
    <handconstellation contact="medium">
      <location location="palm" bodyside="nondom" contact="touch"/>
      <location location="palm" bodyside="dom"    contact="touch"/>
      <location location="chest" contact="medium"/>
    </handconstellation>
    <par_motion manner="targetted">
      <directedmotion manner="targetted" direction="o" size="medium"
            curve="u" curve_size="medium" ellipse_direction="l"/>
      <tgt_motion manner="targetted">
        <split_handconfig>
          <handconfig extfidir="do"/>
          <handconfig extfidir="do"/>
        </split_handconfig>
        <handconstellation contact="medium">
          <location location="palm" bodyside="nondom" contact="touch"/>
          <location location="palm" bodyside="dom"    contact="touch"/>
        </handconstellation>
      </tgt_motion>
    </par_motion>
  </sign_manual>
</hamgestural_sign>
</sigml>
```

Figure 3: Explicit low-level SiGML for the “Going-To” Example.

```
<CAS Version="CAS2.0" Avatar="VGuido">
  <Frames Count="32">
    <Frame Duration="20.0000" BoneCount="67" MorphCount="42">
      <Morph Name="eee" Value="0.0000"/>
      ....
      <Bone Name="ROOT">
        <Position x="-0.0007" y="-0.0501" z="-0.0496"/>
        <QRotation x="-0.0286" y="-0.7137" z="0.0276" w="0.6993"/>
      </Bone>
      ....
    </Frame>
    ....
  </Frames>
</CAS>
```

Figure 4: Character Animation Stream (CAS) Data for the “Going-To” Example.