Synthesizing Sign Language by Connecting Linguistically Structured Descriptions to a Multi-track Animation System

Michael Filhol, John McDonald, Rosalee Wolfe

DePaul University, Chicago, IL, USA
michael.filhol@limsi.fr,{jmcdonald, wolfe}@cs.depaul.edu

Abstract. Animating sign language requires both a model of the structure of the target language and a computer animation system capable of producing the resulting avatar motion. On the language modelling side, AZee proposes a methodology and formal description mechanism to build grammars of Sign languages. It has mostly assumed the existence of an avatar capable of rendering its low-level articulation specifications. On the computer animation side, the Paula animator system proposes a multi-track SL generation platform designed for realism of movements, programmed from its birth to be driven by linguistic input.

This paper presents a system architecture making use of the advantages of the two research efforts that have matured in recent years to the point where a connection is now possible. This paper describes the essence of both systems and describes the foundations of a connected system, resulting in a full process from abstract linguistic input straight to animated video. The main contribution is in addressing the trade-off between coarser natural-looking segments and composition of linguistically relevant atoms.

Keywords: Sign language, computational linguistics, computer animation, avatar

1   Introduction

Being a comparatively young field of research, Sign Language (SL) linguistics is still one of lively debate more than a set of tested and established scientific theories. Linguistic features and properties, for which a wide theoretical consensus exists, are not numerous. Most propositions to formalize SLs remain in question, including the most basic premises of language studies such as the nature or even existence of lexical units and a phonological level. [1] [2] [3]

Computer generation is therefore a goal located between two potentially unfortunate choices. One would be to choose a set of description formalisms arbitrarily. The other would be to keep playing back recorded (e.g. with motion capture) or hand-crafted (drawn) data with no model of it at all, which would never allow or foster content editing tasks. This paper presents our first exploration of a way forward between those two

problematic paths. It combines a language description model set on making as few assumptions as possible about language structure, and an animation system built on the principle that the language description should drive the animation while making as few assumptions and limitations on how language processes can combine to affect avatar motion.

2 The AZee Language Description

The initial goal of the AZee framework was to enable formal description, without assuming any of the still debated hypotheses, especially when they would lay the foundations of the entire system.

For example, all formal approaches to NLP assume not only a syntactic level of description but also a clear-cut and finite list of categories (e.g. N, V, Adj…), and a fixed mapping from a set of known lexical units to those categories. Whereas, the nature, relevance or even existence of such categories and a syntactic level are not agreed on when considering Sign Languages.

To enable SL description and synthesis without unwisely categorizing language objects, the AZee model proposed to fall back on three very weak linguistic assumptions before formalizing its visible features [4], which are explained in turn below.

Language productions create observable forms carrying intended meaning. Observable *forms* are visible states and movements of any of the language's articulators, and synchronisation features between them. For example, A, B and C below are all visible form descriptors.

  A: "eyelids closed"

  B: "left fingertip in contact with left-hand corner of mouth"

  C: "orient palm downwards"

When two or more articulations are involved on a shared time line, synchronisation constraints between them such as "B starts before A ends" or "C is fully contained in A" are also form descriptors.

AZee calls a *function* any meaning that could be associated with such forms, whether intended (when the forms are produced) or interpreted (when the forms are perceived). For example, D, E and F are functions in this sense.

  D: "house" (as a concept carried by this word in English)

  E: "pejorative judgement on [person/object]"

  F: "[date & time] form the context of [event]"

Systematic links between the two are what specifies the language. The idea behind this assumption is that language is a system, i.e. is governed by rules, shared between the users of the language, and that an experimental approach can identify. So we define the modelling problem as one of finding systematic associations between forms and functions. In other words, AZee proposes to capture every:

- invariant combination of forms observed for the occurrences of an identified function, which yields a *production rule* that can be animated by SL synthesis software;
- function consistently interpreted from the occurrences of a given form, which yields an *interpretation rule* that can support recognition software.

With our objective of synthesis, the former rule type is the one we will be referring to in this paper.

Languages allow for compositional structures. Compositionality in language is an essential premise of all language studies, admittedly already assumed in both sections above since they refer to "combinations" of forms and "pieces" of the meaning. This is why AZee allows for rule parametrisation, whereby a structure is defined as a process applied to its arguments. Comparably to common formal grammars, it also allows recursion.

These assumptions have two noteworthy corollaries:

- **Multi-linearity**: allowing to describe any sort of form combination, enrolling multiple articulators and defining potentially complex synchronisation patterns calls for the ability to partition and layer articulation tracks on top of one another, like a music score arranges multiple instrument productions on a single time line.
- **No functional partitioning of the articulator set**: any form feature, articulation or synchronisation constraint can be associated with a function if systematically observed, and a single articulator can take part in many different functions.

These are novel properties for a formal model describing SL, which we have argued were an advantage over those generally used to this day [5]. For example, the widespread standard HamNoSys [6] relies primarily on manual activity as the vector for the lexical function, and builds SL utterances as a strings of juxtaposed lexical units, which makes it mostly linear.

The AZee framework proposes a methodology to identify, and a language to formalise, production rules turning interpretable functions into the forms to articulate, and parametrise them to account for rule composition. Every rule is a <H, P, f(P)> triplet:

- H is the rule **header**, usually named after the function it carries, e.g. "house";
- P is a list of **parameters** on which both the interpretation and the form might depend, e.g. the location of the house;
- f(P) is a description of the **form** to produce if the rule is applied and given its necessary parameters, including all necessary and sufficient articulations and syn-

chronisations, e.g. finger extension, duration of contact, etc. (by analogy with production rules in formal grammars, we call this part the *right-hand side* of the rule, RHS henceforth).

For example in LSF, expression of non-subjectivity in a measure, estimate or judgement X (in the sense that the speaker reckons X is generally not disputed), requires that lips be moved forward over the time used to sign X, see [1]. This yields a rule:

- header: "non-subjectivity";
- parameters: X = the estimated value or judgement;
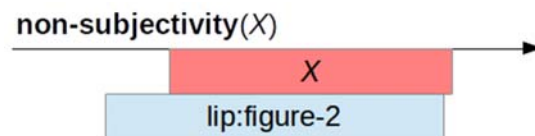- RHS: see box diagram in **Fig.** 1, and the lip form to produce in **Fig.** 2.

Fig. 1. RHS box diagram for "non-subjectivity"

**Fig. 2.** Lip form used in RHS of "non-subjectivity"

Fig. 3. Manual form for "good"

Rules being combinable, any rule's production can be used in place of a parameter for another rule. For example, consider the LSF rule "good", named after the English word for its meaning, with no parameters and whose RHS results in the fixed manual form in **Fig. 3**. One can use the product of this rule as parameter X for "non-subjectivity", nesting the former function inside the latter in an AZee expression:
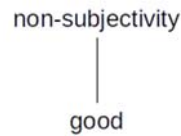
(E1) non-subjectivity(good())

This *composes* the meaning of "generally deemed good", and directly *produces* the resulting form (**Fig. 4**) by combining the RHS of the invoked rules, which results in a multi-track time-line specification of the full arrangement. In the case of (E1), this means layering the lip pout of **Fig. 2** over the manual form of **Fig. 3**. We call the resulting arrangement a sign score.
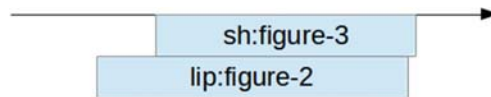

Fig. 4. Resulting score for (E1)

An AZee expression is generally represented by a tree structure, called a functional tree, whose nodes are the contained rules' headers used in the expression, and in which node A has child nodes B_i, if B_i are the arguments of A in the expression. The tree for our simple expression (E1) is:

We give another example, of a more complex tree below, for expression (E2):
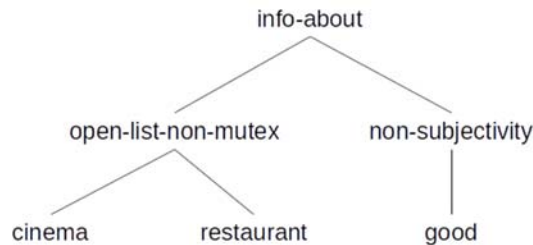
(E2) info-about(open-list-non-mutex(cinema(), restaurant()), non-subjectivity(good()))

where:

- "info-about" has parameters *A* and *B*, carries function "*B* is the information given about *A*" [4] and has the RHS of **Fig. 5**;
- "open-list-non-mutex" has any number of parameters *Ei*, carries the function of itemising every *Ei* to form a non-exhaustive list of non-mutually exclusive items, and whose RHS is the succession in time of a **Fig. 6** for each *Ei*;
- "cinema" and "restaurant" have an optional point parameter each for location, carry the respective concepts they are named after in English, and have the manual RHSs illustrated in pictures **Fig. 7** and **Fig. 8**.
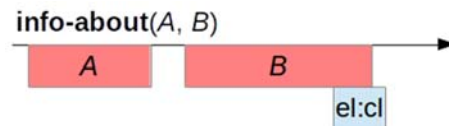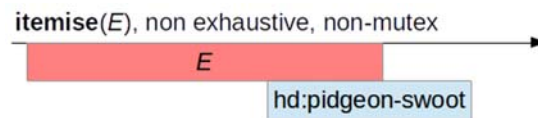

Fig. 5. RHS of "info-about"

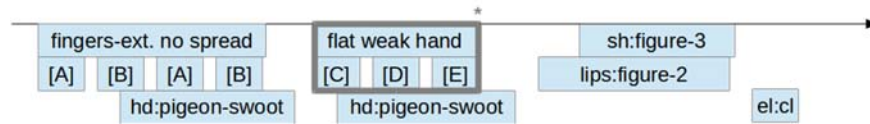**Fig. 6.** RHS of every item for "open-list-non-mutex"

Fig. 7. RHS of "cinema" in LSF, image from [7]

Fig. 8. RHS of "restaurant" in LSF, image from [7]

Layering all the RHS forms contained in the rules invoked in the expression, (E2) results in the score in **Fig. 9**, while composing the meaning: "cinema[s], restaurant[s], etc. are generally deemed good".

Fig. 9. Resulting score for (E2)

The hope from there is that there exists a set of rules able to cover all language productions by combination, and that it remains small enough to be acceptable as a grammar model.

3 The Paula Animation System

Animating signed languages with an avatar is an ongoing challenge with difficulties arising both from the nature of human motion and the requirements of the input linguistic model. The flexibility and multi-track nature of the AZee description requires a commensurate level of flexibility in an animation system. An animation system such as [8], which is based on the HamNoSys and SiGML representations, scripts animation in a sequential manner and would present many challenges in connecting to AZee. The Paula sign synthesis system [9] provides a multi-track animation system that resembles AZee's representation. This section explores the nature of computer animation as it relates to sign synthesis, and provides an overview of the Paula system.

Synthesizing sign directly from linguistic descriptions that specify key poses is well known to create robotic motion, but does have the advantage of being able to convey any utterance that can be encoded. Several other options for human character animation are also available, including traditional keyframe, procedural and motion-capture based animation. Each of these has distinct advantages and challenges when applied to the animation of signed languages.

Traditional hand-crafted key frame techniques rely on an animator for the composition of the movements of the avatar. The movements are controlled by setting key postures and interpolation data [10]. Depending on the skill of the artist, the resulting animations can be extremely natural and appealing, with the advantage that the sparse nature of the keyframe data can be relatively easy to edit, modify and connect together to build animations from individual segments [11]. The main disadvantage of hand-crafted keyframe animation is the high cost of animator time and the fact that new animations cannot be produced at will based on linguistic descriptions such as AZee.

Motion capture can produce some of the most realistic animation on an avatar since it is recording all of the subtleties of human motion. It can also be an invaluable tool for studying the motion of signers. Mocap recordings have provided insight into many aspects of sign including prosody and ambient non-linguistic motion [12] [13].

Mocap does come with challenges that make it difficult to use in a flexible sign synthesis system. First, motion capture can be nearly as costly to record and clean as hand-crafted animation is to produce [14]. It relies heavily on clean-up artists to both remove ambient noise in the system [13] and also to fill in segments that are missing due to occluded sensors [15]. Second, the simple fact that mocap records all of the small details of human motion makes it challenging to stitch mocap recordings together in a natural way.

Another challenge with motion capture from a synthesis perspective is that the sheer detail and data density can often record more than we actually want when trying to layer linguistic forms. A motion capture clip will have all of the linguistic relevant forms (the ones meant to be described by the AZee framework) the signer is expressing baked in with all other observable forms, and swapping out only part of the effects on a set of joints for a different form will be akin to attempting to remove the sugar from a baked cake to try to replace it with honey. In contrast, a skilled animator with knowledge of sign language structure can produce animations which are very clean in their expression of the linguistic parameters of the utterance. This facilitates layering linguistic processes in synthesized utterances.

Finally, procedural techniques in animation model the movements of the body mathematically and are useful for synthesis in several situations including:

- Providing time saving utilities for keyframe animation [16]. Such tools can shorten the time that animators need to spend in fine tuning details of the animation and reduce the cost of this form of animation.
- Modelling the motion for linguistic forms that affect specific parts of the body [17]. Such processes can be provided with a set of parameters to tune the motion to the linguistic form.

In addition, procedural techniques can be used to produce non-linguistic forms that nevertheless increase legibility of sign synthesis by adding carefully tuned procedural noise to an avatar's joints to simulate the ambient motion that all human signers exhibit even when still or holding a pose [13].

In short, different animation techniques are useful or appropriate for different situations, and the Paula animation system is designed to leverage multiple animation techniques and capitalize on each technique's strengths. It strives to use coarser hand-animated blocks to increase the naturalness of the motion while leveraging procedural blocks to both increase the naturalness of the motion and layer additional linguistic functions over the base movement. The Paula animation system is thus a multi-level hybrid system that strives to strike a balance between flexibility and naturalness in the avatar's motion through two components.

The first component is a set of hand-crafted animations built in a sign transcriber that resembles a traditional animation package, but with controls tied to the linguistic components of sign language, see **Fig. 10**. This component allows the animation of basic lexical signs and other animation components that rely on artist-driven key-frame techniques [9].
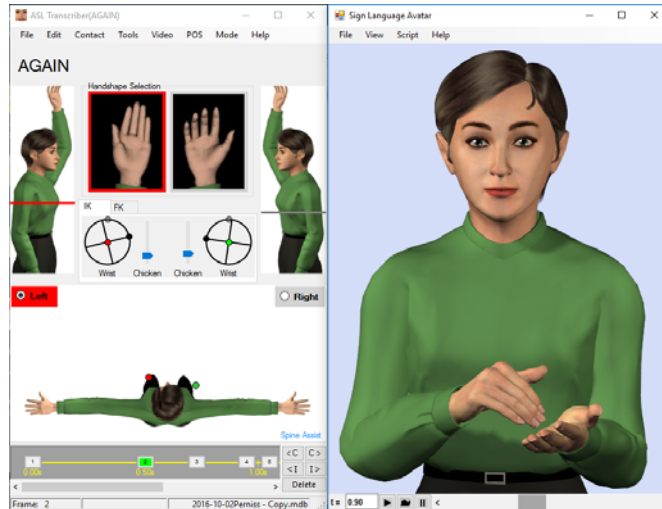
Fig. 10. Paula's animation interface

The artist's work is augmented by procedural techniques such as a spine assist computation that augments the forward and inverse kinematics controls on the model's arms to move the torso in a natural way, thus reducing work for the artist [11]. Another important aspect of these controls and procedural techniques is that they help assure that animations encode as many linguistic parameters as possible and are as sparse, i.e. contain as few "key postures", as possible to enable easier editing and combining of animations.

The second component of the Paula system composes complete signed utterances. The system structures signed language animation as a collection of parallel tracks. Each track contains a sequence of animation blocks which can be timed and synchronized arbitrarily, much like a sound recording.

Unlike traditional animation systems, each track can affect entire sections of the avatar's anatomy, and multiple tracks can affect the same anatomy without disruption or conflict. For example, in the example in **Fig. 11**, the system animates several overlapping processes present in (E2). The base lexical items previously animated are combined with a collection effects that influence the face, torso and head. For more details on the techniques used by this system see [9].
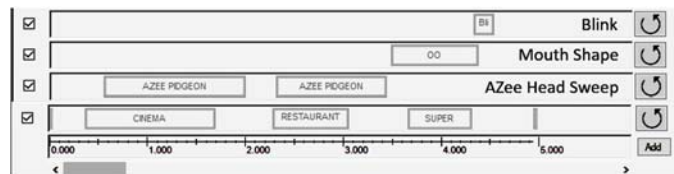

Fig. 11. Paula's Sentence Generation Component

In summary, the Paula animation system provides a flexible framework in which

1. Animations can be built from a sequence of tracks, each of which contains a sequence of animation blocks.
2. Tracks can manipulate any part of the avatar's anatomy, and multiple tracks can manipulate the same anatomy.
3. Track blocks can use a variety of animation techniques including key-frame and procedural techniques. In the future we plan to support other techniques such as motion capture and synthesis directly from linguistic input parameters.

4 Appropriateness of the Two Systems for Connection

As we have explained, Paula implements a time line of parallel tracks, on which timed animation blocks are placed so that performing the tracks simultaneously produces a full Sign Language utterance. The animation blocks on the time line have so far mainly been filled with:

- hand-crafted animations, more or less computer-assisted at the time of creation but ultimately fixed to be played back;
- procedural specifications, i.e. parametrised descriptions directly producing articulations, e.g. extent and duration of mouth aperture or position of a hand in space.

In the Paula system, virtually any contents can fill an animation block, as long as it specifies an articulation.

On the AZee side, by construction, a node in a functional tree can only exist if a production rule was identified, i.e. together with a parametrised description of a form to produce (the rule's RHS). This RHS mixes *invariant descriptors* (fixed articulations like the lip pout in "non-subjectivity") with *parameter dependencies* like X in the same rule, usually containing more *function calls*. In any case, it always results in a score of synchronised parallel blocks, containing the layered fixed articulations contained in the RHS terms of the input expression.

In short, AZee provides with a mechanism that turns any functional tree into a resulting score that is similar to the internal animation format of the Paula animation system. We propose to take advantage of this by mapping every fixed form descriptor to a Paula animation block. This way, any resulting score from AZee would be trivially translated to a Paula time line. For example, one could map:

- the lip pout descriptor in the RHS of "non-subjectivity" over to a Paula mouth shape animation block [OO] performing the lip movement, fitting on the lip articulator track;
- and the fixed form in **Fig. 3** of "good" to, say, a hand-crafted animation block [GOOD] on the manual track.

Expression (E1) generates a Paula timeline with [OO] and [GOOD] directly in place. And similarly for (E2) if all blocks of the resulting score in **Fig. 9** is mapped to a Paula block.

This proposition allows to consider full synthesis of Sign Language animations straight from semantically-loaded linguistic expressions. We emphasise that it is possible mostly thanks to the fact that the two corollaries of AZee's axioms were assumed in the design of Paula from the start. That is, both systems use parallel tracks and free block synchronisation, with no need to project unit composition a 1D time axis or split the tracks early in the process.

However, linguistically meaningful (i.e. systematically interpretable) compositions in SL are meant to be accounted for in AZee, even within an RHS. For example, the form for "restaurant" in LSF involves an iconic weak hand shape reference to a flat surface, in this case of a waiter's notepad (see **Fig. 8**). This flat palm and finger arrangement being interpretable in other contexts the same way, good AZee practice would be to incorporate a function call "flat surface" in the RHS of "restaurant".

This basically captures the morphemic function of the hand shape, and is a token of AZee's flexibility as it can capture any sort of linguistic function in any sort of context. This has the power of accounting for as much compositionality of the language as possible. But in our circumstance of generating realistic animations, we understand that this proposition leads us to generate and combine many low-level animation blocks, which we know leads to robotic gestures, and is generally unacceptable for final rendering [18]. Thus, the connection between the two systems is guided by the following principle:

Guiding principle: The coarser the basic animation blocks, the more natural the final animation.

To generate coarser elements and honour this principle, we have to avoid processing the deeply nested AZee parts, and recognise subtrees for which we could have Paula blocks available, whether as pre-recorded natural-looking animations or as procedures defined on the Paula side.

For example, function "resto" applied with no arguments, represented by the simple one-node subtree, always generates the same form, namely that in the picture in **Fig. 8** and enclosed in the thick box (*) in **Fig. 9**. If we have that sign recorded, we can shortcut the AZee production rule and avoid reading and developing its RHS (incl. its flat weak hand description). We may play back the recorded form directly from the recognised subtree, which allows for a natural looking play back of the whole rule's RHS. Generalising this, an alternative solution is to match Paula animation blocks from full AZee expressions instead of just the basic AZee form descriptors. Applied to our first example (E1), this causes the system to search for a Paula block ready for play-back and meaning "generally deemed good".

Clearly, pushing the principle to the extreme rapidly goes out of scale. There is no formal limit against wishing for direct Paula matches for higher-level subtrees, or indeed for entire inputs like the full expression (E2), which would require an infinite number of recorded forms. So, we need some counter-balance.

We thus propose to prefer coarse building blocks when they are available, but allow the system to fall-back on finer pieces otherwise. Therefore, we propose to create mappings to Paula animation blocks from both of the following sets:

- a subset of all possible AZee expressions;
- the complete set of basic AZee form descriptors.

From any AZee input, we only then need to read the functional tree from the root down, try matching the full subtree against the inventory of available Paula blocks at every step (giving preference to the taller tree matches when they exist), and developing the RHS if no match is found (which takes the process one level down inside the score building expression).

5   Validation and Moving Forward

The expression (E2) is a first example that validates the direction of this mapping and makes a first test of the two systems as potential components of a complete synthesis solution. Initially, the two systems were each built as separate research efforts, yet:

1. The design of the Paula animation system has always had as a goal the eventual connection to a structured linguistic system such as AZee. To this point, however, the linguistic features expressed by the avatar have been incorporated in an ad-hoc manner, driven partially by the physical capabilities of the avatar. The result has been a system that displays increasingly fluid, legible sign, but which needs a more complete linguistic structure to drive its animation.
2. AZee also has been designed with the intention of connecting to an animation system, but to this point has had to itself perform all the work of translating its forms into articulatory specifications for connection to an avatar that would simply interpolate those articulator specifications.

So, while both recognized the need and intention to connect to a system much like the other, they have each pursued their own goal and separate evaluations. However, each have now reached a level of maturity that the bridge between them described in the last section is feasible.

Expression (E2) applied the AZee to Paula mapping for a set of test utterances in French Sign language (LSF). In each case, AZee built the required LSF structure and produced a high level score that coordinated linguistic forms as shown in **Fig. 9**. This score was translated in to a preliminary version of the XML encoding of the utterance. An excerpt of this XML encoding is displayed in **Fig. 12**. Notice that in the XML output, AZee explicitly produces a series of hooks for the Paula system to use to shortcut if it has the capability.

The XML encoding was then used to populate the tracks of the Paula sentence generator, as seen in **Fig. 11**. It is up to Paula to translate and interpret the timing hints in such a way as to ensure smooth transitions between animation blocks on a given track and to preserve an overall flow in the animation. The head sweeps, mouth shapes and blinks are layered as procedures on top of the hand-crafted animations that provide a shortcut for the three hand-animated blocks CINEMA, RESTAURANT and GOOD. The resulting animation can be seen at the following URL:

http://asl.cs.depaul.edu/Paula-Azee/Paula-Azee-Example.mp4

The mapping between the two systems provides a first validation of each system's approach from a technical standpoint, gives evidence of feasibility, and lays the groundwork for further development of both systems.

```
<block_score>
        <block id="gl1">
                <paulablock name="CINEMA"/>
                <sync/>
        </block>
        <block id="pn1">
                <paulablock name="AZEE-PIDGEON"/>
                <sync>
                        <param name="dur" value=".8"/>
                        <param name="end" value="gl1:-1:.25"/>
                </sync>
        </block>
```

Fig. 12. AZee XML output of the utterance

6   Conclusion

The results presented in this paper demonstrate the feasibility of the mapping between AZee and Paula and also an initial validation of the proposition in section 4. The power of this mapping, and thus the method for animating structured Sign Language linguistics lies in the fact that higher levels of animation can be used to animate entire sections of the AZee tree thus leveraging hand-animated or captured motion. In addition, the structured nature of the linguistic model allows us to exploit the layered nature of the AZee score to structure the corresponding animation. These lessons are important for any system that wishes to synthesize Sign Language animation based on a structured linguistic description.

It is, however, important to note that the details of some of the elements presented here, such as the XML format in **Fig. 12** are preliminary and will need further study to expand into a full system. Moving forward, we plan to expand the mapping between these two systems to more fully encompass the structures encoded by AZee and the range of expression available in Paula.

References

[1] C. Cuxac, La langue des signes française, les voies de l'iconicité, Editions Ophrys, 2000.

[2] E. Van der Kooij, Phonological categories in Sign Language of the Neth-erlands: the role of phonetic implementation of and iconicity, Utrecht: PhD Thesis, 2002.

[3] W. C. Stokoe, "Semantic phonology," *Sign Language Studies,* vol. 71, no. 1, pp. 107-114, 1991.

[4] M. Filhol and M. N. Hadjadj, "Juxtaposition as a form feature; syntax captured and explained rather than assumed and modelled," in *Language resources and evaluation conference (LREC), Representation and processing of Sign Languages*, Portorož, Slovenia, 2016.

[5] M. Filhol, M. N. Hadjadj and A. Choisier, "Non-manual features: the right to indifference," in *Language resource and evaluation conference (LREC), 6th workshop on the representation and processing of Sign Language: beyond the manual channel*, Reykjavik, Islande, 2014.

[6] S. Prillwitz, R. Leven, H. Zienert, T. Hanke and J. Henning, HamNoSys version 2.0, Hamburg notation system for sign languages, an, Hamburg: Signum Press, 1989.

[7] B. Moody, Dictionnaire bilingue LSF/français, IVT editions, 1998.

[8] R. Elliott, J. Glauert, J. Kennaway, I. Marshall and E. Safar, "Linguistic modelling and language-processing technologies for Avatar-based sign language presentation," *Universal Access in the Information Society,* vol. 6, no. 4, pp. 375-391, 2007.

[9] R. Wolfe, J. McDonald and J. Schnepp, "An Avatar to Depict Sign Language: Building from Reusable Hand Animation," Berlin, Germany,, January 10-11, 2011.

[10] R. Parent, Computer animation: algorithms and techniques, Boston: Newnes, 2012.

[11] J. McDonald, R. Wolfe, J. Schnepp, J. Hochgesang, D. Jamrozik, M. Stumbo, L. Berke, M. Bialek and F. Thomas, *An automated technique for real-time production of lifelike animations of American Sign Language.,* 2015, pp. 1-16.

[12] R. B. Wilbur and E. Malaia, "A new technique for analyzing narrative prosodic effects in sign languages using motion capture technology," in *Linguistic foundations of narration in spoken and sign languages*, A. H. &. M. S. (eds.), Ed., Amsterdam, John Benjamins, In Press.

[13] J. McDonald, R. Wolfe, R. Wilbur, R. Moncrief, E. Malaia, S. Fujimoto, S. Baowidan and J. Stec, *A new tool to facilitate prosodic analysis of motion capture data and a data-driven technique for the improvement of avatar motion,* Portoroz, 2016.

[14] S. Dent, "What you need to know about 3D motion capture," Engadget, 14 August 2014. [Online]. Available: https://www.engadget.com/2014/07/14/motion-capture-explainer/. [Accessed 21 January 2017].

[15] M. Gleicher, "Animation from observation: Motion capture and motion editing," *ACM SIGGRAPH Computer Graphics,* vol. 33, no. 4, pp. 51-54, 1999.

[16] R. Wolfe, J. McDonald, R. Moncrief, S. Baowidan and M. Stumbo, "Inferring biomechanical kinematics from linguistic data: A case study for role shift," in *Symposium on Sign Language Translation and Avatar Technology (SLTAT)*, Paris, France, 2015.

[17] J. McDonald, R. Wolfe, R. Moncrief and S. Baowidan, "Analysis for synthesis: Investigating corpora for supporting the automatic generation of role shift," in *Proceedings of the 6th Workshop on the Representation and Processing of Sign Languages: Beyond the Manual Channel. Language Resources and Evaluation Conference (LREC)*, Reykjavik, Iceland, 2014.

[18] M. Kipp, Q. Nguyen, A. Heloir and S. Matthes, "Assessing the deaf user perspective on sign language avatars," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility. ACM*, 2011.