

KAZOO: A Sign Language Generation Platform Based on Production Rules

Annelies Braffort, Michael Filhol, Maxime Delorme, Laurence Bolot, Annick Choisier, Cyril Verrecchia

▶ **To cite this version:**

Annelies Braffort, Michael Filhol, Maxime Delorme, Laurence Bolot, Annick Choisier, et al.. KAZOO: A Sign Language Generation Platform Based on Production Rules. Universal Access in the Information Society, 2016, 15, pp.541-550. 10.1007/s10209-015-0415-2 . hal-01633619

HAL Id: hal-01633619
https://hal.science/hal-01633619

Submitted on 6 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Universal Access in the Information Society 2016, vol 15, pp. 541-550 - Authors' version manuscript No.
(will be inserted by the editor)

KAZOO: A Sign Language Generation Platform Based on Production Rules

Annelies Braffort · Michael Filhol · Maxime Delorme · Laurence Bolot · Annick Choisier · Cyril Verrecchia

Received: date / Accepted: date

Abstract This paper describes KAZOO, a web application for sign language (SL) generation using a virtual signer. Firstly, it explains the motivation to this project, which is grounded on an approach designed solely from SL corpus analysis and modelling. Then, various projects conducted in the past few years on linguistic modelling and 3D animation are presented. The platform's architecture integrates parts of this work and new pieces of software allowing control and linking of all these components. This is an on going project, though the current version offers the possibility to animate a virtual signer automatically and synthesize the contents using an abstract representation, the authors' own linguistic model AZee, providing a means of validating this model.

Keywords Sign Language Generation · Virtual Signer · Signing Avatar

1 Introduction

Recent advances in computer graphics and animation have allowed a significant leap forward in the world of accessibility for the deaf, by offering the possibility to create and display 3D content in sign language (SL). Research in SL synthesis has proven that virtual signer approaches represent useful solutions for broadcasting information in public areas and for translating portions of text. Various platforms have been developed with this aim in the past years [17, 18, 26, 15].

This paper aims to describe KAZOO, a web-based platform for French sign language (LSF) content synthesis and animation using a virtual signer. The design of this platform was initiated one year ago, with two initial aims: first, to gather the results of the various studies carried out at LIMSI in recent years, on linguistics and computing aspects, and second, to allow evaluation and offer proofs of concepts. This is an on going project that will still evolve significantly during the following years; however, the first version is presented here.

This paper explains the motivation and specifications for a new platform, then presents the main blocks it is built on, and describes the current implementation.

A. Braffort
LIMSI-CNRS
Campus d'Orsay, F-91403 Orsay, France
annelies.braffort@limsi.fr

2 KAZOO: motivations, constraints and specifications

The specifications of the platform consist of three aspects: its architecture, the type of animation generated and displayed, and the underlying linguistic modelling and associated processing. The motivations for the chosen solutions on these three aspects are presented in the following.

2.1 KAZOO architecture: a web-based platform using WebGL

For the software design aspects, it was decided to use a web-based infrastructure. The benefits of such architectures are the following:

- web applications only require a compatible web browser to be used;
- they provide cross-platform compatibility in most cases because they operate within a web browser window;

- they allow for easier introduction of new user devices (e.g. smartphones, tablets) because they have built-in browsers;
- they require no upgrade procedure since all new features are implemented on the server's side and automatically visible to the users.

These benefits come with some drawbacks:

- the application is no longer usable when the connection is interrupted;
- the application availability depends on the availability of the server delivering the application.

Despite these drawbacks, web applications become increasingly popular and participate to the growing tendency of ubiquitous computing. Moreover, in the present case, the targeted applications for the time being are research prototypes. Web-based architecture allows to easily design proofs of concepts accessible to the community and to evaluate them. Additionally, this choice of architecture will allow designing collaborative applications, such as the SignWiki prototype[1] designed during the Dicta-Sign project [6].

Very few virtual signer platforms are web-based at this moment. JASigning[2] is one of them, and is based on the Java programming, and language thus requires a plugin to be installed on the client side. On the contrary, KAZOO is designed in a way that the clients do not need any specific plugin or applet to be installed. All computations are performed on the server side. The client only has to deal with the 3D display, based on the WebGL[3] technology, which is now integrated in most web browsers.

WebGL is a programming interface specification of dynamic 3D for HTML5 pages and applications. It is based on OpenGL and uses the JavaScript language. WebGL is designed to run on the client side, with the computationally intensive rendering operations carried out if possible by the client's graphics processing unit (GPU). Of course, a consequence is that the efficiency and quality of the graphics rely on the client's capabilities. WebGL scenes can be created without programming using a content creation tool such as Blender. The scenes are then exported to WebGL.

Within KAZOO, the WebGL element will be in charge of displaying signing animation.

[1] http://www.dictasign.eu/
[2] http://vh.cmp.uea.ac.uk/index.php/JASigning
[3] http://www.khronos.org/webgl/

2.2 KAZOO animations: combining synthesis and ready-made signing

Several SL animation systems have been designed in the past years, with various approaches, which can be categorized in two main types:

- ready-made animations, using motion capture data or created using rotoscoping or authoring tools (e.g. the Sign Smith Studio platform[4] is a stand-alone application for authoring content using ready-made animations);
- synthetic animations, built from symbolic representations and associated processing techniques (e.g. the JASigning platform is a web-based system that produces animated SL in an automatic way).

As stated in [13], the adequacy between the technology (ready-made or synthetic animation) and the targeted content (from fixed to completely open) is variable. While these technologies could be used for any kind of content, they are not equally appropriate for all types of content. Typically, ready-made animations are more appropriate for fixed or parameterized content, where the content elements are known in advance, while synthetic ones are better for the cases where the content is subject to frequent change or not known in advance.

Moreover, the variability of the sign production due to its grammatical context and the frequent use of productive structures such as depicting units make completely ready-made animations not sufficient even for very small content, given that these productive constructions are built on the fly and cannot be listed in advance. On the other side, symbolic representations that are used to produce synthetic animations are not yet able to represent all the linguistic rules of SLs, those languages still being less described linguistically. For example, not enough is known on non-manual phonetics and their linguistic role to be able to generate these events with the correct form for every SL utterance.

For all these reasons, the aim is for KAZOO to offer the possibility to use both synthetized and ready-made animations, and to combine them where needed.

2.3 KAZOO language processing: SL-specific approach

The classical approaches in natural language processing (NLP) are the rule-based and data-based approaches, or a mix of them. These approaches, designed at the beginning for written or speech processing, have been applied with some adaptations in the field of SL processing [21, 22, 23, 27].

[4] http://www.vcom3d.com/

The common point of these approaches is that they consider SL utterances as describable in a sequence of items, from a sequence of glosses to a sequence of phonetic descriptions. In these approaches, non-manual events are sometimes considered, but the utterance remains modelled as a sequence of gestural units whose components, whether manual or non-manual, all share the same temporal boundaries, with possibilities to shift boundaries in a limited way and always related to the manual units. However, linguistic studies based on corpus analysis have described much more complex temporal arrangements. At this moment, these approaches do not allow representing the multilinearity and the complex synchronization patterns involving all articulators of SL production that can be observed when annotating SL corpora at a sufficiently fine level.

The authors have carried out several linguistic studies, mainly dedicated to phonetic aspects, at a very fine level, such as the study on eyebrow movements and blinking [1] or on coarticulation [25]. More recent studies, initiated during the Dicta-Sign project [6] are related to specific linguistic functions and the way they are performed [9]. Figure 1 shows an example of annotation of the LSF part of the Dicta-Sign corpus using iLex [14]. All these studies have emphasized the importance of the multilinear property of SL and of non-manual components in discourses and dialogues, with many complex synchronization patterns [12]. Often discarded as prosody, or considered less important than what is conveyed by the hands, non-manual components actually bear various kinds of linguistic functions that must be accounted for when designing SL grammar models for SL processing. This point is verified when virtual signer production is assessed by deaf signers [19].

Existing "linear-based" platforms are not flexible enough to deal with descriptions of fine parallel gestural events. Some researchers advocate the need to adopt other points of view, specific to SLs, to overcome the limitations inherent to the classical approaches, especially on the multilinearity of gestural activity. Some models account for parallel gestural events, carried by different articulators over different time spans [16, 20, 10]. This was the reason for designing a platform, grounded on these SL-specific linguistic models.

In summary, KAZOO's specifications are that it must be a web-based application using WebGL technology to display 3d animations of ready-made animations, together with synthetic ones based on an SL-specific linguistic model.

3 LIMSI's existing building blocks

In addition to these specifications, the aim was to gather several models and applications carried out in previous projects, related to linguistic modelling and signing animation. A detailed description of these elements is out of the scope of this paper. This section provides the information needed to understand the purpose and capabilities of each part. More details can be found in the given references.

3.1 Linguistic modelling: the AZee representation

Linguistic modelling is the core of the authors' research. They have worked on two representations these last years: Zebedee and Azalee, now merged to form the AZee model.

3.1.1 Zebedee

Zebedee [7] represents signs in a temporal and geometric way, defining constraints for the body at specific times and how to move between these postures. The syntax and operators, not detailed here, allow the signs to adjust to semantic context by using abstract dependencies.

For instance, a sign such as BALL in LSF (see Fig. 2a) can be performed in different ways, according to the contextual size of the ball and its location in the signing space.

Size and location are therefore parameters of the sign, called contextual dependencies. The temporal structure of this sign is composed of three timing units: two key-postures separated by a transition, each with a duration. Each of these timing units is composed of a set of geometrical constraints, some coming from the context, others internal.

For example, the first key posture constraints for the sign BALL are the following (Fig. 2b):

- the handshapes are parameterized by the size of the ball: more or less rounded, fingers more or less outspread (contextual dependencies);
- the hands orientation is parameterized by the location of the centre of the circle that represents the ball in the signing space (contextual dependencies);
- the strong hand location is parameterized by the same location, and a given distance to the centre of the circle (contextual dependencies);
- the weak hand orientation and location are parameterized by a type of symetry to the strong hand (internal dependencies).

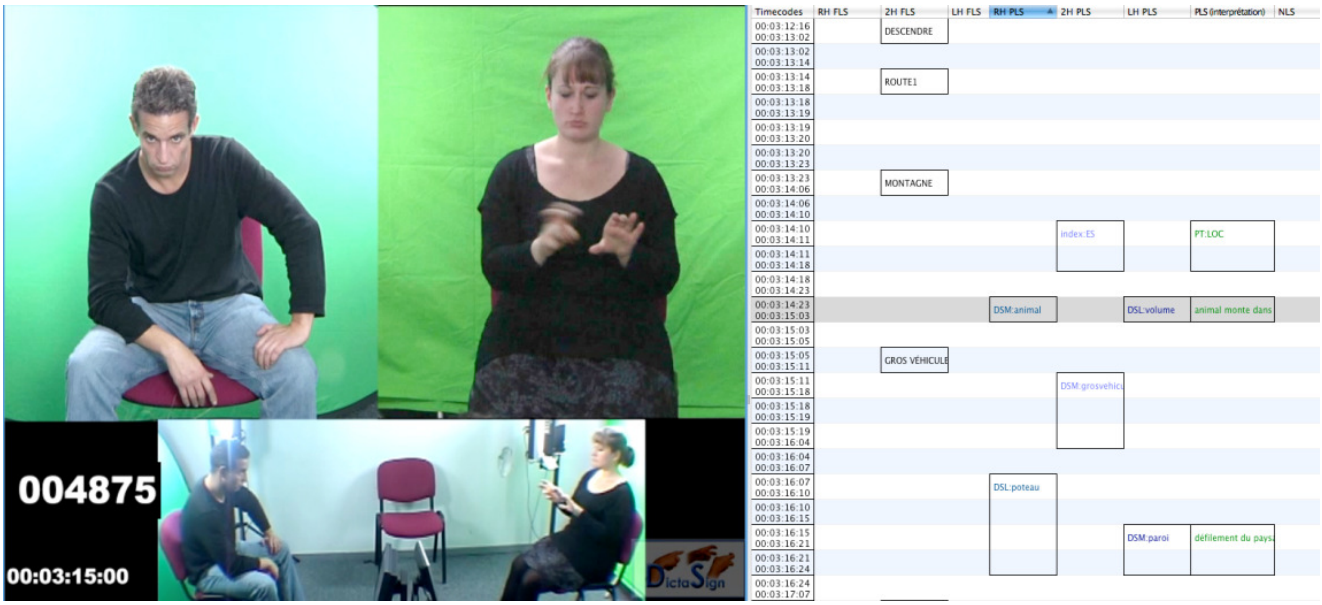


Fig. 1 Annotation of the Dicta-Sign corpus.

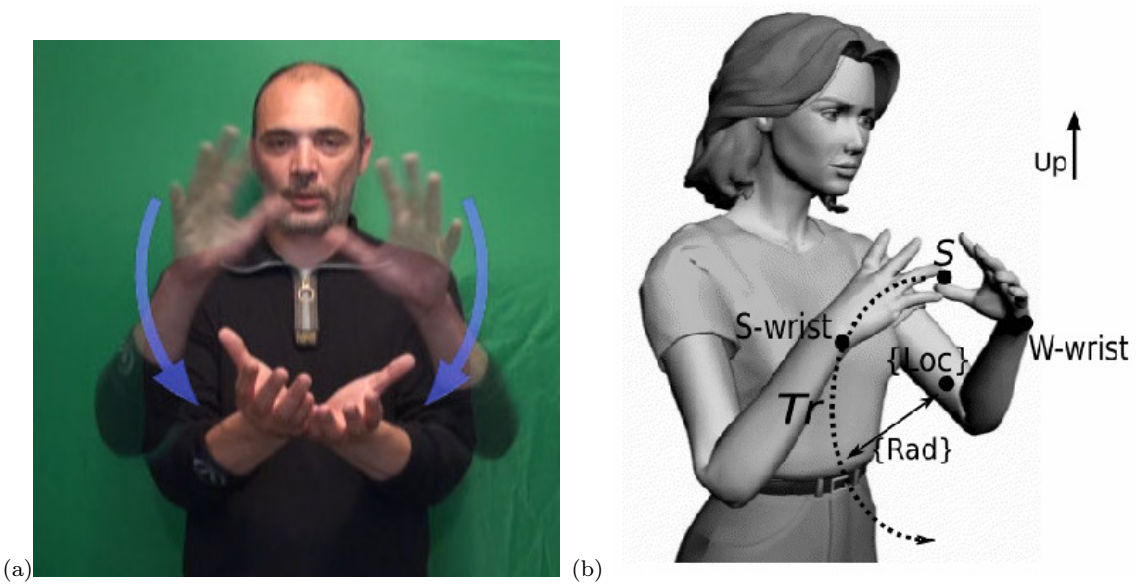


Fig. 2 The BALL sign in its citation form (a), and the parameters used in the description (b).

3.1.2 Azalee

Azalee [8] is a language designed to specify synchronization patterns for sets of identified time intervals (TIs) on a single timeline. It describes the synchronization constraints between the articulators. The current syntax and available operators, not detailed in this paper, allow specifying:

- separate articulators groups, such as head activity or hand stroke;
- synchronize their boundaries in time.

For example, Azalee can describe the synchronization constraints in depicting units such as those showing the displacement of an entity towards another entity (e.g. a person towards a car). The articulator groups involved are eyegaze, weak hand and strong hand (see Fig. 3). The weak hand has been anchored to its current location in a previous signing unit. The eyegaze just precede the strong hand movement.

3.1.3 AZee

AZee [8, 10] is a model for production grammars, usable for SL synthesis, composed of set of nestable production



Fig. 3 Displacement of a person towards a car: Beginning of the strong hand movement.

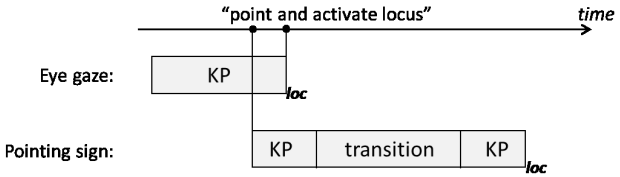


Fig. 4 AZee timeline illustration of "pointing and activate locus" production rule.

rules and able to derive any LSF utterance, integrating Zebedee and Azalee.

For example, a pointing gesture coordinated with an eye-gaze that activates the pointed locus can be described using AZee (Fig. 4). In this example, the precedence of eye-gaze on the manual activity is represented, and for each articulator, the needed constraints: for the eye, the target point 'loc' and for the strong hand, the temporal structure 'a little movement, i.e. a transition surrounded by two key-postures (KP)' and the pointed direction, which is the same 'loc'.

Details on the syntax and operators together with examples can be found in other papers [12, 2].

This model allows to describe the various constraints in a very flexible way and, most importantly, all the linguistically motivated constraints, and only them. As a result, the animation synthesis module must be able to precess description of flexible synchronization between the various articulators.

3.2 Virtual signer animation and posture synthesis: Octopus and GeneALS

The authors have worked on two animation frameworks in recent years: Octopus and GeneALS.

3.2.1 Octopus

Octopus [3] allows ready-made animations to be displayed in a standalone application. It handles the generation of a SL utterance as follows:

1. A database contains ready-made animations of signs or chunks, and morphs for facial animation. These sets of animations and morphs are designed using rotoscoping by a specialized animator, providing highly natural animations (Fig. 5).
2. The SL utterance is built by coarticulating animations, combined with facial morphs. Moreover, morphs can be weighted and combined (e.g. one for the eyebrows, one for the cheeks).

Octopus integrates results from studies such as eyebrow movements and coarticulation [3], allowing to adjust:

- transition lengths between postures within a sign;
- duration of gestural phases for both manual and non-manual events;
- weight of facial events, such as eyebrow raising or lowering.

Octopus has been used for evaluation of some of the models, such as coarticulation [24].

Moreover, it has been used in the first version of a product developed by the company WebSourd and used by the SNCF French railway company. This product provides announcements about train incidents, modifications or delays in LSF. The messages in audio, written and LSF forms are generated at the same time (Fig. 6).

The sentences are composed of concatenated predefined pieces of information stored in a database. This is a typical parameterized type of content. For example in sentence (1) below, there are obviously variable parts, such as the platform and train numbers, the type of train, the name of the stations and the arrival time. There also are bigger chunks that can be reused in other sentences, such as "your attention please".

Sentence 1: "Your attention please, platform 10, due to a technical problem, TGV trains number 2456 and 2457, coming from Strasbourg and Reims, will arrive at 1.30pm."

3.2.2 GeneALS

GeneALS [4, 5] is a generation module that produces synthetic body postures. The initial aim was to evaluate the capability of the Zebedee linguistic representation.

GeneALS takes a sign description in Zebedee as input and produces a list of the posture descriptions



Fig. 5 Illustration of the use of rotoscoping for animation generation.



Fig. 6 Jade, the virtual signer used in SNCF French railway stations (here: Gare de l'Est in Paris).

needed to animate this sign as output. The output describes the joint angles for each bone composing the skeleton, excluding the legs and the face (Fig. 7).

Given the description of a sign, GeneALS extracts constraints on the skeleton for each key posture, which defines the problems to be solved by the inverse kinematics (IK) process. This process works in three steps:

1. Explore: the system builds a high number of random solutions within the limits of the skeleton. The N best solutions are kept for the next step. The best solutions are selected regarding their overall score (how close they are from the target posture), and their distance to the others solutions (the further, the better).
2. Optimize: the system processes gradient descent on each of the N solutions, normally converging to the nearest optimum.
3. Select: each of these solutions is given a comfort score defining which posture is the most comfortable for the skeleton. The solution with the best comfort score is considered to be the most realistic one. The comfort score is computed using a statistical comfort model which has been built thanks to a prior study based on motion capture corpora.

The skeleton has also been designed with special focus on the shoulder and thumb structures for more human-like postures [5].

In summary, KAZOO uses the AZee linguistic model and partly reuses Octopus and GeneALS applications, with some adaptations. AZee and GeneALS have been put together to provide an animation from an AZee description. Parts of Octopus have been used for computation and final display of the 3D animation.

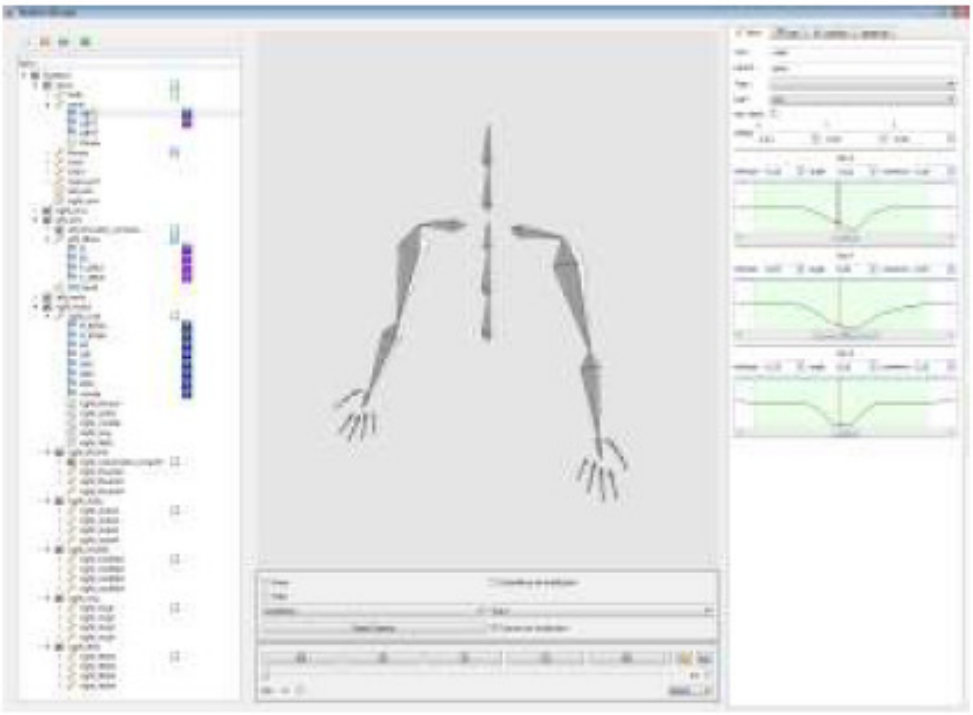


Fig. 7 GeneALS manager window for manipulation of the anatomic model and posture resolution.

4 KAZOO: a web application for virtual signer animation

In addition to these building blocks, KAZOO integrates a database and new pieces allowing to control and link all the components.

The database is used to store LSF representations in AZee. The GeneALS's IK module computes postures. The coarticulation and combination capabilities of Octopus are used to render the final animation, including interpolations between postures, and addition of morphs and the visualization of the animation.

In addition to these modules, a web server has been added that allows the user to input his/her request and receive the animation description. The overall architecture is illustrated in Figure 8. The features of the various modules and the current state of their implementation are described hereafter.

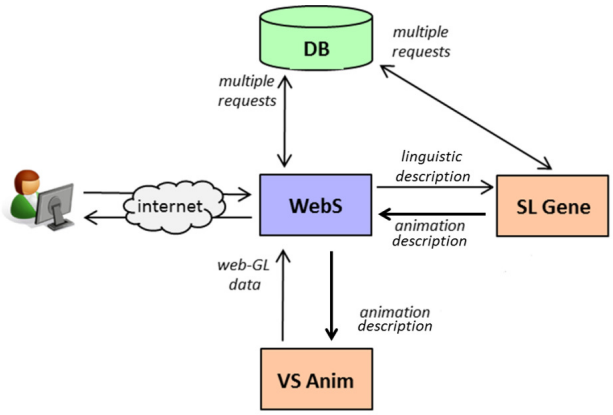


Fig. 8 Kazoo: module organization.

4.1 Database module (DB)

The database is a PostgreSQL server. It is planned to integrate the AZee production rules, together with representations of various formats, e.g. ready-made animation using rotoscoping or motion capture techniques.

At this moment, the database contains a representative set of signs and production rules, described using AZee descriptions. Four categories of input are distinguished regarding the way the interaction with the user interface must be implemented to complete a description (see Sec. 4.2):

- frozen descriptions, with no contextual dependencies;
- modifiable descriptions, where one or more elements can optionally depend on the context (such as the sign for tree, whose location can be modified depending on the context);
- descriptions where at least one contextual element has no default value (such as a pointing sign);
- components that can be part of a description but not processed alone, i.e. that do not result on a

signed performance (such as the list of sites on the body, or locations in the signing space).

Figure 9 shows an example of description for the second category, the sign for tree, which can be signed in a neutral location in the signing space or in a different location, driven by the context. The description starts with this contextual dependency (line 2: "loc") and default value at a location in the signing space (lines 3 and 4). Then, the information on the temporal structure is given (lines 5 and 6: a key-posture with duration). Finally, five articulatory constraints are listed for this key posture:

- a "place" for the weak-side palm to be moved to the strong-side elbow (lines 8-14);
- a "place" for the strong-side elbow to be moved to the 'loc' contextual dependency (lines 15-19);
- an "apply" using the "cfg5" shortcut for the description of the handshape of the strong hand (lines 20-23);
- an "orient" for the strong-side fore-arm (lines 24-28);
- an "orient" for the strong-side palm (lines 29-36).

The database is currently being populated with a set of more than 1000 lexical sign descriptions and a dozen of production rules currently being prepared. Its size will grow with the results of future studies.

```
1  azop
2      'loc'
3      site
4          'ABST'
5      key
6          10
7          cstrblock
8              place
9                  site
10                     'PA'
11                     w
12                 site
13                     'EL'
14                     s
15             place
16                 site
17                     'EL'
18                     s
19                 @loc
20             apply
21                 @cfg5
22                 'side'
23                 s
24             orient
25                 bone
26                     'f_arm'
27                     s
28                 up
29             orient
30                 bone
31                     'palm'
32                     s
33                 dir
34                     bone
35                         'f_arm'
36                         s
```

Fig. 9 Linguistic description for the lexical sign ARBRE (tree).

4.2 Web server module (WebS)

The web server is in charge of collecting the user's request and of displaying the virtual signer animation. It is built on a client-server architecture, using PHP and JavaScript languages, Ajax techniques for the interaction side, and WebGL for the 3D animation display.

The planned functionalities include:

- An interface that allows listing the available elements in the database, and collecting the user's request; this interface must depend on the category of the chosen element: that is, if a complex structure is chosen, the interface must analyse which parts of the description must be populated and by what type of data (e.g. another element from the database, a direction, a location in the signing space to be chosen for the TREE description for example);
- A WebGL window for displaying the virtual signer animation, which is updated dynamically depending on the user's requests.

At this moment, the interface is designed with up-to-date technology (HTML5, CSS3, etc.), though a basic graphical user interface. For example, only a subset of possible locations is proposed for a given relocatable lexical sign. This will be improved in the future versions, by adding descriptions for more possible locations or sites.

This module is operational and provides a complete AZee description of the selected input, which is then transferred to the SL generation module.

4.3 SL generation module (SL Gene)

This module is in charge of the automatic computation of the animation description from the linguistic description. It proceeds in four steps:

1. collects the user's request;
2. builds a complete and consistent AZee representation combining all the linguistic constraints on all the articulators;

3. computes the animation description using GeneALS as described in Sec. 3.2.2, together with specifications for the facial components;
4. outputs a low-level XML description of the animation.

This module interacts with the database a number of times in order to get all the AZee descriptions needed, because structures or sign descriptions can embed AZee descriptions.

After collecting all the necessary descriptions, the module solves the constraint system and generates the key frames and articulations needed for the animation. The result of the computation is stored in an XML structure containing all the key frames needed for the animation, together with the skeleton posture, facial morphs, durations and profiles for the transitions (Fig. 10).

At present, most of the parts have been implemented. However, the automatic synthesis of sign postures derived from GeneALS is really costly and can take a few minutes to process. Therefore, optimization of the generation system is a necessary improvement.

4.4 Virtual signer animation module (VS Anim)

This module is in charge of applying the animation to a virtual signer. It uses part of Octopus, implemented in C# language. It also uses the JavaScript language and the WebGL API. The specifications of this module are the following:

- it takes the output of the SL Gene module as input (the XML description);
- it outputs an animation in a WebGL window using the Octopus capabilities (see Sec. 3.2.1).

At this moment, the skeleton animation is operational, and the face animation using morphs is being integrated. Figure 11 shows a screenshot of the current version of the KAZOO web page.

5 Conclusion

KAZOO, a web application for SL generation using a virtual signer, and the theoretical and technical principles it is built on, has been presented. While this is an on-going project for which significant changes are expected in the near future, a first operational version is already available.

AZee linguistic model allows more flexibility, precision and completeness in the descriptions of SL productions, paying more attention to all articulators (not centered on the manual ones) and the way they are combined (not time-aligned). The synthesis and animation modules allow for the creation of 3D SL content that follows this linguistic approach.

This first version now offers to animate a virtual signer from this linguistic model, which is a first step towards its validation.

Once KAZOO has reached a sufficient level of completeness and robustness, the plan is to offer public web access to it. It will be used to assess both the linguistic models and the animation modules. Other possible uses of this platform are web-based collaborative applications, such as SL wikis or e-learning platforms, and computer-assisted translation system from text to SL [11].

References

1. Braffort A, Chételat-Pelé E (2012) Analysis and description of blinking in french sign language for automatic generation. In: Efthimiou G E Kouroupetroglou, S-E F (eds) Gesture and Sign Language in Human-Computer Interaction and Embodied Communication, LNCS/LNIA, vol 7206, Springer, pp 173–182
2. Braffort A, Filhol M (2014) Constraint-based sign language processing, Cambridge Scholar Publishing, pp 189–215
3. Braffort A, Bolot L, , Segouat J (2011) Virtual signer coarticulation in octopus, a sign language generation platform. In: Efthimiou E, Kouroupetroglou G (eds) 9th International Gesture Workshop (GW'11): Gesture in Embodied Communication and Human-Computer Interaction, ILSP/ATHENA RC and National and Kapodistrian University of Athens, pp 29–32
4. Delorme M (2011) ModÃ©lisation du squelette pour la gÃ©nÃ©ration rÃ©aliste de postures de la langue des signes franÃ§aise. PhD thesis, Paris-Sud University
5. Delorme M, Filhol M, Braffort A (2012) Thumb modelling for the generation of sign language. In: Efthimiou G E Kouroupetroglou, S-E F (eds) Gesture and Sign Language in Human-Computer Interaction and Embodied Communication, LNCS/LNIA, vol 7206, Springer, pp 151–160
6. Efthimiou E, Fotinea T Stavroula-Evita Hanke, Glauert J, Bowden R, Braffort A, Collet C, Maragos P, Lefebvre-Albaret F (2012) Sign language technologies and resources of the dicta-sign project. In: 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon. Satellite Workshop to the 8th

```
<Score>
  <KeyFrames>
    <KeyFrame id="1000" time="0"/>
    <KeyFrame id="1001" time="4"/>
    <KeyFrame id="1002" time="4.5"/>
    <KeyFrame id="1003" time="0.00014999500016666114"/>
    <KeyFrame id="1004" time="1.5000999666667778"/>
  </KeyFrames>
  <ScoreSpec>
    <Hold start="1000" end="1001">
      <Rots>
        <Rot joint="left_elbow" x="-0.0262369" y="0.0800809" z="0
        <Rot joint="left_clavicle" x="-0.0679108" y="-0.995625" z
        <Rot joint="left_shoulder" x="-0.153357" y="-0.852298" z=
      </Rots>
    </Hold>
    <Hold start="1004" end="1002">
      <Rots>
        <Rot joint="right_elbow" x="-0.65249" y="0.527171" z="0.5
        <Rot joint="right_clavicle" x="0.0173203" y="0.999718" z=
        <Rot joint="right_shoulder" x="-0.212689" y="0.324449" z=
      </Rots>
    </Hold>
```

Fig. 10 XML description of the animation.

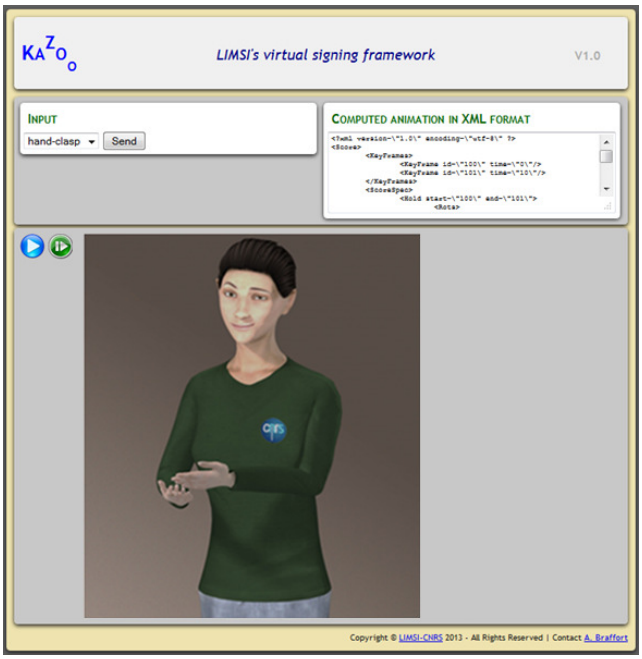


Fig. 11 Kazoo demo page, version 1.0.

International Conference on Language Resources and Evaluation (LREC 2012), ELRA, pp 37–44

7. Filhol M (2008) ModÃšle descriptif des signes pour un traitement automatique des langues des signes. PhD thesis, UniversitÃ© Paris-Sud
8. Filhol M (2012) Combining two synchronisation methods in a linguistic model to describe sign language. In: Efthimiou E, Kouroupetroglou G, Fotinea SE (eds) Gesture and Sign Language in Human-Computer Interaction and Embodied Communication, LNCS/LNAI, vol 7206, Springer Berlin Heidelberg
9. Filhol M, Braffort A (2012) A study on qualification/naming structures in sign languages. In: 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon. Satellite Workshop to the 8th International Conference on Language Resources and Evaluation (LREC 2012), ELRA, pp 63–66
10. Filhol M, Braffort A (2012) What constraints for representing multilinearity in sign language? In: Duchier D, Parmentier Y (eds) 7th International Workshop on Constraint Solving and Language Processing (CSLP 2012), LIFO, Orléans University
11. Filhol M, Hadjadj M, Testu B (2013) A rule triggering system for automatic text-to-sign translation. In: Wolfe R, McDonald J (eds) 3rd International Symposium on Sign Language Translation and Avatar Technology (SLTAT 2013), De Paul University
12. Filhol M, Hadjadj M, Choisier A (2014) Non-manual features: the right to indifference. In: 6th

Workshop on the Representation and Processing of Sign Languages: Beyond the manual channel. Satellite Workshop to the 9th International Conference on Language Resources and Evaluation (LREC 2014), pp 49–54
13. Hanke T (2013) Will there ever be a market for signing avatars? some observations on the past and future of our field. In: Wolfe R, McDonald J (eds) 3rd International Symposium on Sign Language Translation and Avatar Technology (SLTAT 2013), De Paul University
14. Hanke T, Storz J (2008) ilex - a database tool integrating sign language corpus linguistics and sign language lexicography. In: 3th Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora. Satellite Workshop to the 6th International Conference on Language Resources and Evaluation (LREC 2008), ELRA, pp 64–67
15. Huenerfauth M, Lu P (2012) Effect of spatial reference and verb inflection on the usability of sign language animations. Universal Access in the Information Society 11(2):169–184, DOI 10.1007/s10209-011-0247-7, URL http://dx.doi.org/10.1007/s10209-011-0247-7
16. Huenerfauth MP (2006) Generating american sign language classifier predicates for english-to-asl machine translation. PhD thesis, University of Pennsylvania
17. Jennings V, Efthimiou E, Kennaway R, Glauert J (2010) Requirements for a signing avatar. In: 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Languages Technologies. Satellite Workshop to the 7th International Conference on Language Resources and Evaluation (LREC 2010), Paris: ELRA, pp 133–136
18. Kipp M, Heloir A, Nguyen Q (2011) Sign language avatars: Animation and comprehensibility. In: Proceedings of the 10th International Conference on Intelligent Virtual Agents, Springer-Verlag, Berlin, Heidelberg, IVA'11, pp 113–126, URL http://dl.acm.org.gate6.inist.fr/citation.cfm?id=2041666.2041682
19. Kipp M, Nguyen Q, Heloir A, Matthes S (2011) Assessing the deaf user perspective on sign language avatars. In: 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS'11), ACM, pp 107–114
20. López-Colino F, Colás J (2012) Hybrid paradigm for spanish sign language synthesis. Journal of Universal Access in the Information Society (UAIS) 11(2):151–168
21. Marshall I, Sáfár E (2005) Grammar development for sign language avatar-based synthesis. In: 3rd International Conference on Universal Access in Human-Computer Interaction (UAHCI2005), Lawrence Erlbaum Associates
22. Morrissey S (2011) Assessing three representation methods for sign language machine translation and evaluation. In: Mikel L Forcada VV Heidi Depraetere (ed) 15th Annual Meeting of the European Association for Machine Translation (EAMT 2011), EAMT, pp 137–144
23. San-Segundo R, Barra R, Córdoba R, Fernando D'Haro L, Fernández F, Ferreiros J, Manuel Lucas J, Macías-Guarasa J, Manuel Montero J, Manuel Pardo J (2008) Speech to sign language translation system for spanish. Speech Communication 50(11-12):1009–1020
24. Segouat J (2010) ModÃ©lisation de la coarticulation en langue des signes franÃ§aise dans le contexte de la diffusion automatique d'informations en gare ferroviaire Ã l'aide d'un signeur virtuel. PhD thesis, UniversitÃ© Paris-Sud
25. Segouat J, Braffort A (2010) Toward modeling sign language coarticulation. In: Kopp S, Wachsmuth I (eds) Gesture in embodied communication and human-computer interaction, LNCS/LNAI, vol 5934, Springer, pp 325–336
26. Wolfe, Rosalee Cook P, McDonald JC, Schnepp J (2011) Linguistics as structure in computer animation: Toward a more effective synthesis of brow motion in american sign language. Sign Language and Linguistics 14(1):179–199
27. Wu CH, Su HY, Chiu YH, Lin CH (2007) Transfer-based statistical translation of taiwanese sign language using pcfg. ACM Transactions on Asian Language Information Processing (TALIP) 6(1):1–18