WebSign: A system to make and interpret signs using 3D Avatars

Oussama El Ghoul Mohamed Jemni

Research laboratory U.T.I.C
University of Tunis
Ecole Supérieure des Sciences et Techniques de Tunis
5, Av. Taha Hussein, B.P. 56, Bab Mnara 1008, Tunis TUNISIA
oussama.elghoul@utic.rnu.tn mohamed.jemni@fst.rnu.tn

ABSTRACT

Nowadays, thanks to the technological progress in the area of information and communication, many techniques have been made up in order to allow communication in sign language. Signed contents are being published in the World Wide Web using three common formats: images, videos and 3D animations. Thanks to its advantages, 3D animation represents one of the best formats used as sign language's content support. In fact, many web developers prefer to create contents based on virtual signers.

In this context, the research laboratory UTIC of the University of Tunis [8], has initiated since 2006 a national research project entitled Websign [17] aiming to improve communication with deaf community through the use of ICT. Websign is an avatar-based system that enables the translation of written text into sign language. The system is based on a multi-community approach to be compliant with the locality feature of sign language. It allows using specific dictionary for each community in addition to a common dictionary shared by all. Ease of use is what our project is all about. In fact, adding a new sign to the dictionary and generating 3D signed scenes are done using a convivial web based human-software that we have developed without requiring any programming skill.

1. INTRODUCTION

The main target of Websign project is the implementation of a tool which facilitates the communication with deaf people and speech disabled individuals through the Web and Sign Language (SL). Our aim was the creation of an avatar-based system able to generate signs and to play them. The system is based on web technologies in order to make it accessible to the highest number of persons. Our objective is to improve the accessibility of deaf to the technologies of information and communication by the development of an application which allows the translation of written text to sign language.

SLTAT 2011, 23 October 2011, Dundee, UK.

This tool can be used by deaf persons to communicate together or to learn sign language; it would also enable people who do not know SL to communicate with deaf individuals which could be useful for deaf' parents. Therefore, it contributes in reducing the language barriers that are being faced by deaf and hearing individuals to communicate togather.

A secondary objective of this project is to make this tool available for free for all its users (i.e. instructors, teachers, students and researchers) to encourage its wide use by different communities.

This paper is organized as follows: in the next section we present the application that we have developed for the translation of written text to sign language and how to use the system presented to create or build dictionaries of signs. Section 3 is dedicated to the state of the art in which we present the most popular systems used to store signs in databases and the way they should be used to insert signs. In section 4, we describe our contribution, the approach used and the architecture of our system. In section 5, we present the new features that we have developed recently in order to improve the efficiency of the system by adding a module that includes techniques to generate facial expressions. Finally, we come up with a conclusion and some perspectives.

2. WebSign

WebSign [4,5,12,17] is a Web application based on the technology of avatars (animation in virtual world). The input of the system is a text. The output is a real-time and on-line interpretation of the output into sign language. This interpretation is rendered thanks to a dictionary of words and signs. The creation of this dictionary can be made in an incremental way by users who propose signs corresponding to words using the system described in this paper. A word and its corresponding sign are added effectively to the dictionary only after its assessment by an expert who supervises the system.

Our system is based on Client/Server architecture. To implement this design, we have realized the following steps: first, we have created the clients, the server and the database. Then, we have saved several samples of animation codes to initiate the database and finally we have implemented the interpretation algorithm via the service of messaging.

For the linguistic treatment we used, in a first step, the Xerox Web-Service [13], available for free, which ensures the segmentation of a sentence and returns the set of segmented words with their grammatical categories. The returned information is important to reduce errors and missed translations of sentences. In a second step, we started to develop a new system aiming to translate English text to American sign language.

2.1 Multi-community approach

To resolve the problem of the locality of sign language, we introduced the concept of community. A community is a group of users who can build and share a dictionary of SL. A dictionary can be totally created by a specific community or can be just a part of an existent dictionary where some specific words are interpreted differently with respect to the intrinsic specification of this community.

Technically, each community has an administrator, who has the responsibility to manage his community's dictionary. He has the possibility to add, delete, or modify words from his dictionary. He should also modify and validate, delete, or validate the submitted signs which have been created by the members of the community.

2.2 Collaborative approach to add words in the dictionary

The collaborative approach to insert words in the dictionary makes our project unique. Every user can participate in the creation of words via the web. The words added by the users will be stored as propositions to be validated by the administrator before their final insertion in the dictionary.

Using this approach, our system ensures a rapid construction of dictionary thanks to the collaboration of many volunteers and experts in sign language and a wide range of propositions of signs to represent a same word. The diversity of signs is very important to choose the sign used by the largest number of persons in one region.

It is clear that developing such a system is strongly correlated with the creation of a friendly interface for making gestures and saving the description of the gesture into a database. Any person ought to be able to create the sign without necessity of a particular knowledge or an acquaintance of the gesture description language used by the system.

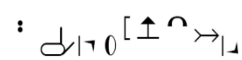
3. State of the art

In the literature many description languages are used to generate a 3D imitation of signs, employing 3D avatars. In general these languages are based on previous languages of transcription of sign language like HamNoSys [10] (Hamburg Notation System) or SignWriting [6] or a language dedicated to the animation of virtual characters like VHML [11] (Virtual Human Markup Language). The first reason why this language is actively used is the usability of the existing resources and data sources already written. The second cause is that those languages are developed and tested by linguists and it's also used by many experts of sign language and many deaf persons. The majority of methods of creating signs used are the same employed by linguist from the apparition of these languages between the 70's and 90's.

3.1 SigML

HamNoSys is a notation of sign language; it was developed as a linear phonetic transcription tool for sign languages, this transcription should be possible for all sign languages in the world, but it is not easy to use by individuals for taking notes or reading.

Using HamNoSys, SigML is developed to describe the sign, which can be played with an avatar (esign [9,2]). In fact, SigML is an XML codification of Hamburg notation system.



```
<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>
```

Fig1: Representation of sign in HamNoSys and his codification in SigML [9]

In this example (Figure 1) we represent the codification of a sign in HamNoSys and the corresponding XML codification in SigML, in which 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 [9].

The inconvenient of this codification is the difficulty to make sign, due to the linearity of this transcription and the non-linearity of sign language. Moreover the sign creator should know and understand symbols and their equivalent in sign language. In fact the user should find out the gestures formation parameters (hand shape, orientation, emplacement, movement and facial expression) from the sign and write them using this transcription. For this reason we believe that this representation is not relevant to the approach that we have chosen.

3.2 Swml

SignWriting is not a linear representation of sign such as HamNoSys; it's based on graphical, bi-dimensional representations, which uses graphical symbols. This transcription can easily be encoded in computers in a linear way, by assigning numeric codes to each symbol.

```
<signbox>
  <symb x="46" y="37" x-flop="0" y-flop="0"
color="0,0,0">
     <category>04</category>
     <group>02</group>
     <symbnum>001</symbnum>
     <variation>01</variation>
     <fill>01</fill>
     <rotation>04</rotation>
  </symb>

  …….
</signbox>
```

Fig2: Representation of sign in SignWriting and his codification in SWML [1]

In search of enhancement and development of the standard, a new format for general encoding of SignWriting was developed; using XML.This notification became more exploitable by software like Vsigns [7] which use a resources encoded on SWML to generate a 3D avatar animation of the sign.

The figure 2 represents an example of the codification of sign in SignWriting and the corresponding XML representation written in SWML format. The figure also shows that even that this representation is bi-dimensional, it remains difficult to understand and to use this transcription to build dictionaries.

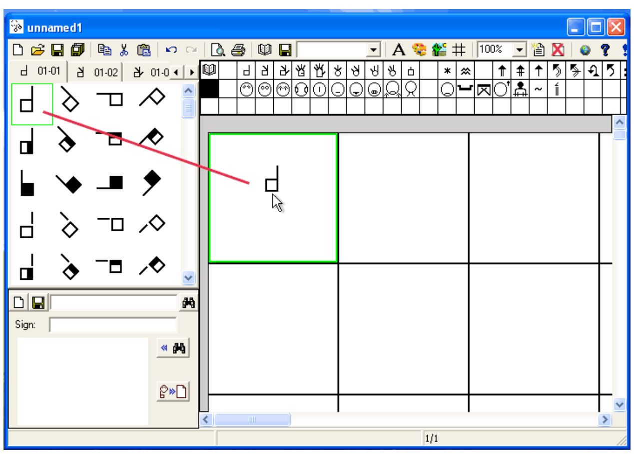


Fig3: SW Edit: an SWML editor

Although there are many tools to manipulate SignWriting, editing signs, draw an XML representation of SWML and so on, like the SW Edit (figure 3) which offers a drag and draw interface to edit signs, the creation of sign remains difficult and need background knowledge of this transcription. Furthermore this transcription contains hundreds of symbols (the equivalent of alphabet in written languages) to write or describe gestures. For this cause, we have decided to eliminate the alternative to use this technology in our approach.

4. Our contribution

The generation of gestures is essential for the creation and the development of applications dedicated to deaf persons, especially those who suffer from illitracy. In most cases, these applications need to store gestures codification/description in databases or dictionaries in order to process sign language, to translate text from or to sign language, to render signs in a 3D scene.

Realizing a dictionary of sign language consumes too much time and a lot of resources, for this reason the reduction of these two parameters improve the development of applications and the facility systems in particular those who have not a digital version of their dictionary. In this context, we propose an application based on web technologies to create dictionaries of sign language using a wiki approach. In fact, the wiki approach is adjudged as the best way to trim down the time and the cost of the creation of big information databases and to keep up-to-date systems and their contents.

In order to create a system working with a wiki approach to make dictionaries of signs, we have developed a web interface usable by deaf or experts in sign language. This interface is easy to manipulate, it uses an intuitive way to animate a virtual character. The user can use this interface to generate automatically and in a transparent way a description of the gesture written in a descriptive language based on the description of the different joins rotations of the avatar armature (skeleton).

4.1 SML

SML is a descriptive language made to support a collaborative approach to write signs. In Websign, we have used SML to store description of signs which can be generated by a virtual human, or an avatar. Unlike SigML and SWML we didn't use an existing system of transcription of sign language but we have opted to the creation of a new language in which we can animate any avatar for any purpose. This animation could be used by different systems in different contexts like chat, animation of robots, virtual characters or non-signer avatars, not necessary by systems designed for sign language.

The animation according to SML is a set of movement or rotation of groups of joins grouped according to their motion over the time axis. Each movement or rotation has a fixed time interval, during which the rotation of every join in the group is done.

The design of the armature is compliant to the H|Anim specifications [3], in which each join have a specific name and specific initial orientation.

Concerning the animation technique, we have required that at the moment *t* the avatar can play a group of rotation joins during *d* milliseconds (figure 4).

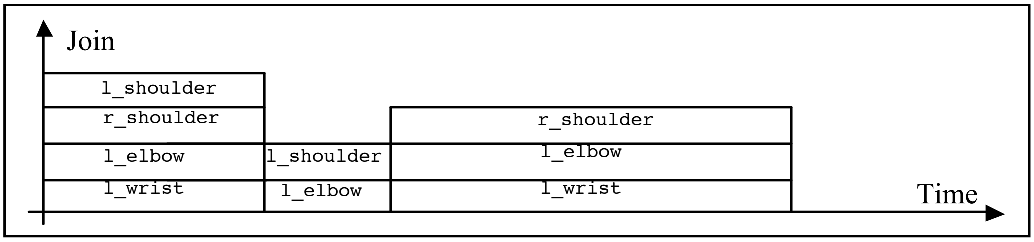


Fig4: SML(S) request

In fact, the simplified SML version (SML(S)), which is parsed and understood by the player, is declared as below:

```
Duration   join1   rotationOfJoin1   join2
rotationOfJoin2 …
```

This declaration represents the description of one group of rotations. To play all groups, we need to send them one by one to the player. The player pushes the received data to a list of movements. Once the player finished the animation of current group, it can jump to the next group while the movements list is not empty (figure 5).

SML(S) is not an extensible language, in which we can't describe easily signs, but it only represents a language to control the player. For this reason we have created an extensible version of SML (SML(X)), which is based on XML.

SML(X) is not comprehensible by the player. Therefore it should be transformed to SML(S) before sending it to the player.

The root element of SML(X) is the <sentence> which represent a sentence in sign language, each <sentence> element contains one or more elements <word>. In analogy with SML(S) this element represent a group of rotation's joins. Every <word> element contains one <duration> element and one or more <join> elements. The <join> element contains one <rotation> element. In SML(X) we can use many types of rotations.

In "Figure 6" the avatar rotates the left elbow join (l_elbow) to the rotation with heading=80, attitude=50 and bank=50 using Euler angle. The duration of movement is 0.5 seconds.

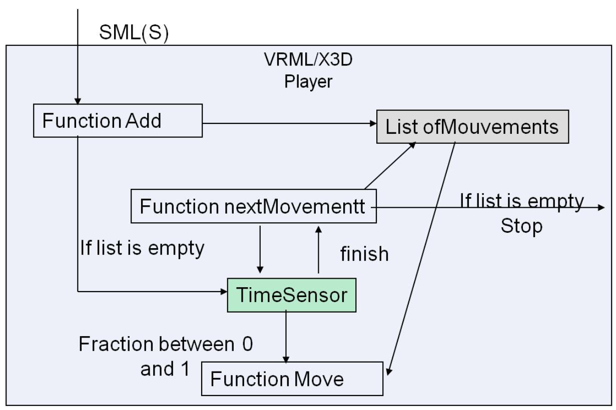
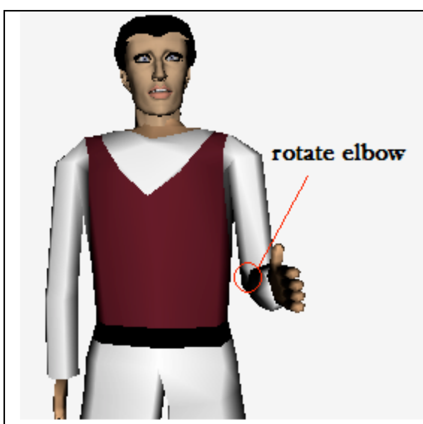


Fig5: Animating received SML(S) request




```
<Sentence>
  <word>
    <mouvement>
      <duration>0.5</duration>
      <join name="l_elbow">
        <rotation type="euler">
          <heading>80</heading>
          <attitude>50</attitude>
          <bank>50</bank>
        </rotation>
      </join>
    </mouvement>
  </word>
</Sentence>
```

Fig6: Rotation of l_elbow join using SML(X)

4.2 Creation of sign and generation of SML(X)

The interface of making signs is a web page containing two modules. The first is in charge of the animation of the virtual character coded with the X3D language and needs a plug-in to be visualized and the second is a java applet that encloses a set of buttons and sliders to allow the animation of the avatar. This interface is a drag and show interface, it offers the possibility to modify the state or the position of the avatar and shows in real time the result of the modification on the 3D character. To do this, we have adopted the key framing approach, for each important and significant position of the avatar animation the user must create a key frame in which he should specify the rotation of the joins he wants to change their states.

The procedure of creation of signs is summarized in three steps; the first is to choose the position of the key frame on the time line by the drag of the pointer over the position selected. The second step consists of the choice of the join to be rotated by the use of the join selector. And finally the user should manipulate the rotation bars to indicate the rotation value of the join and so on until the creation of the sign.

The rotation sliders integrated on the interface allow rotating joins around one of the three axis of the Euler rotation using the local landmark of the join. In fact there are three rotation sliders showing the values of the rotations around the Euler axis.

In order to optimize the time and cost of the interface manipulation we have added some buttons to allow to copy the rotation of a group of joins in a specific key frame and past them into another key frame, this manipulation is useful if the rotation of same joins is reused in many key frames, like the sign of Goodbye in which the sign contains three key frames where the first and the last are the same. Furthermore the user can copy a key frame and past it in another position.

The interface offers also the possibility to do the symmetry of some joins rotations; this manipulation is also constructive if the user has done a rotation in a key frame and needs to do the symmetric in another key frame or in the same key frame. The sign Goodbye is a good example also in this command because the second key frame is the symmetric of the first.

To make the word Goodbye we need only three steps, the first consist to specify the rotations of joins in the key frame number one, the second is to do the symmetry of the rotations joins of this key frames and to past them into the second key frame. The third step consists of copying the first and pasting it into the position of the key frame number three.

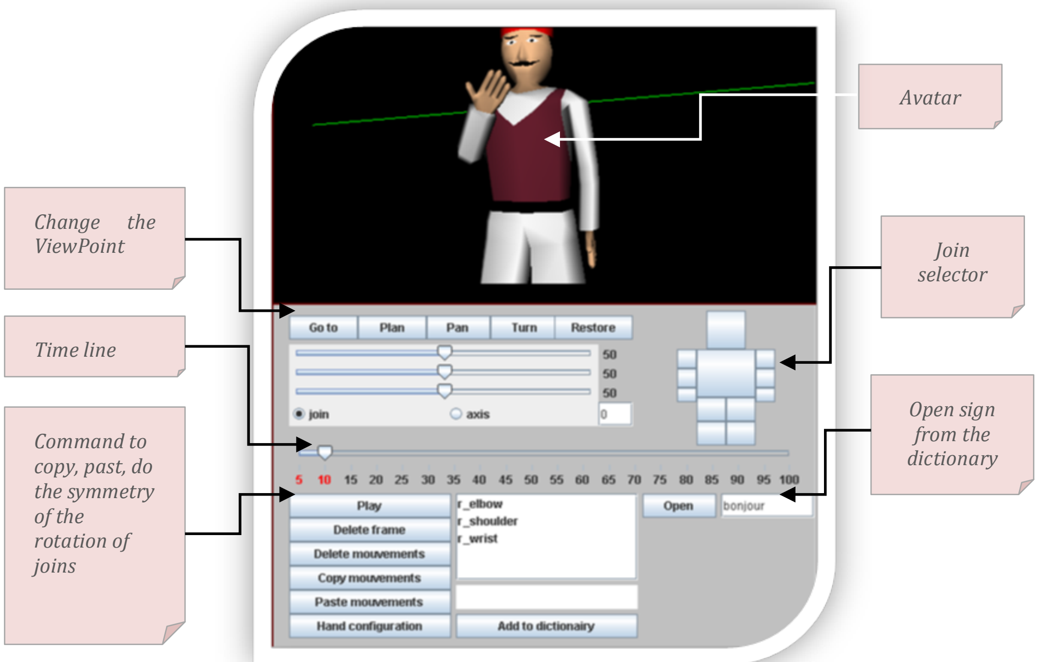


Fig7: A web-based interface of making signs

5. Facial animation

Speaking sign language by manual gestures, in sign language conversations, produces ambiguous sentences or signs which are solved by facial expressions. In fact, the position and the movements of lips, eyebrows or other face parts can be the only difference between two signs. For example, in French sign language, the signs “happy” and “sad” have the same gestures and the only difference between them resides on the face expression (happy or sad). We distinguish two types of facial expressions accompanying gestures. The first one consists of visual speech generation by lips movements and the second is based on predefined animations related to the word. Moreover, thanks to the facial expression, interlocutor can determine the conversation ton and have an idea if the signer was trying to be serious, sarcastic or telling a joke, due to human capacities to understand the emotion marks from facial expressions. Besides, facial expressions are used to specify if the signer is asking a yes/no question or if he is demanding an elaborated answer.

From technological point of view, facial animations are generated by virtual characters which respect many standards and modeled/animated by a large set of techniques. Before starting the project, we have to find out answers of questions like "how can we store facial animation description into databases?" or "how can we describe facial animation using formal description?". During our survey of literature, we have found tow formalisms, the first is developed by the psychologist Paul Ekman in 1976 and the second is specified by computer scientists on the standard Mpeg4. The first formalism Facial Action Coding System (FACS) [15] describes the movements of facial muscles derived from the face anatomy. According to FACS, facial expression is the combination of a set of action units (FACS includes 44 action units). Inspired by FACS, the standard Mpeg-4 [16] has adopted the facial animation and defined (the second formalism) the facial description parameters (FDPs) and the facial animation parameters (FAPs). FDPs are used to specify the geometry of the 3D face, and face animation is described by a set of parameters (FAPs) similar to action units of FACS. Consequently, all linguistics studies are based on FACS because it is close to their domain. However, computer scientists prefer to use Mpeg4 specification because it is related to the geometries deformation description.

5.1 Animation techniques

Since 1976, considerable research, aiming to generate realistic facial modelling and animation, has been done. The objective of the research in this area is to create a realistic facial animation in real time. Many techniques [14] have appeared in order to ameliorate the quality of rendered faces and the complexity of algorithms. The most intuitive and simplest technique is the shape interpolation: this technique is based on the computation of a linear interpolation between tow or more shapes. The second approach is the parameterization, which overcome some limitations of the interpolation techniques. This technique indicates the face expression by grouping a set of independent parameters. Contrary to the interpolation techniques, parameterization allows the control of the configuration of the face. Other approaches use deformation functions to manipulate and distort the mesh areas. Physics-based muscle modelling approach is the most recent technique. This approach is based on the face anatomy in which the face is deformed with the use of a number of virtual muscles. In our system, we propose to use the deformation based approach because it is the most adapted to the Mpeg4 specification.

5.2 Adopted technique

In our system we opted to use the skeleton-based deformation technique, witch consists of adopting the same algorithm of body animation. In fact, each vertex of the face surface can be translated by displacing one or many bones. Vertices are attached to nearby bones and skeleton movement influences the deformation of surface. We propose a skeleton hierarchy able to deform each part of the face such as eyes, lips, eyebrows and so on (figure 8).

The main advantage of this technique is its simplicity and its perfectly compliance with the X3D standard that we use to describe our virtual character. In addition, this technique is sufficient to generate the whole set of natural human expressions including those used in sign languages.

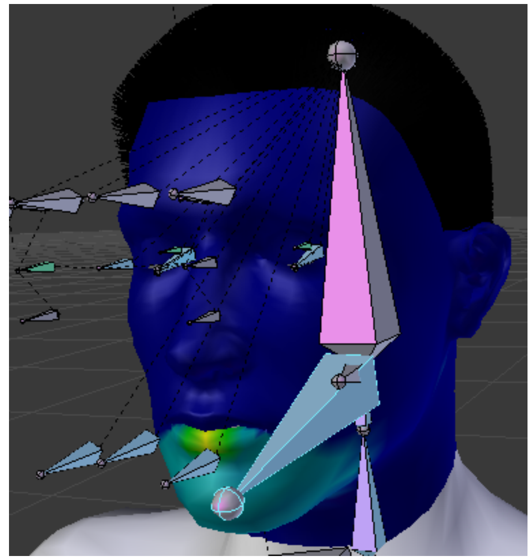


Fig8: skeleton-based deformation technique

6. Conclusion

In this paper we have presented a system based on a collaborative approach to create dictionaries of signs. This system uses a new language to describe sign and avatar gestures in addition to an ergonomic interface for automatic generation of the description of a sign written in this language with a transparent manner.

We have tested this system in our local network in our university and we have succeeded to insert more than 1200 signs in the dictionary in a short time knowing that the six persons who have created these signs are not computer scientists. The six users have appreciated the work of this system after a short tutorial of less than 30mn.

During the evaluation of our work we have seen that we need to add some new features to the interface and we plan to implement them as soon as possible. These functionalities should allow the system to be more flexible and more adapted to the user needs. In particular, the creation of circular movement should be done without need to split the movement into many key frames. We also plan to publish our work so it will be available to the community soon.

7. Acknowledgment

This project was developed by the Research Laboratory of Technologies of Information and Communication and validated by the Tunisian sign language expert Mr Abdel Halim Chalbi.

8. References

[1] Antonio Carlos da Rocha Costa, Gracaliz Pereira Dimuro,Juliano Baldez de Freitas, A sign matching technique to support searches in sign language texts, Workshop on the Representation and Processing of Sign Languages, Lisbon, 30 May 2004.

[2] E. Safar and I. Marshall, G. Angelova et al (ed), Tzigov Chark Bulgaria, The Architecture of an English-Text-to-

Sign-Languages Translation System. Sept 2001, pp223-228, ISBN 954-90906-1-2.

[3] H|Anim standard web site, http://www.h-anim.org.

[4] M. Jemni, O. Elghoul, Towards Web-Based automatic interpretation of written text to Sign Language, First International conference on ICT & Accessibility, April 12-14, 2007, Hammamet, Tunisia.

[5] Oussama El Ghoul and Mohamed Jemni, Multimedia courses generator for deaf children. The International Arab Journal for Information Technology, Vol 6, November 2009, No. 5.

[6] Official SignWriting web site, http://www.signwriting.org.

[7] Official web site of Vsigns, http://vsigns.iti.gr.

[8] Research Unit of Technologies of Information and Communication, http://www.utic.rnu.tn.

[9] Ralph Elliott, John Glauert, Vince Jennings, Richard Kennaway, An Overview of the SiGML Notation and SiGMLSigning Software System, Workshop on the Representation and Processing of Sign Languages, Lisbon, 30 May 2004.

[9] The e-sign project website, http://www.visicast.sys.uae.ac.uk.

[10] University of Hamburg, HamNoSys project, http://www.sign-lang.uni-hamburg.de/projects/HamNoSys.html.

[11] Virtual Human Markup Language web site, http://www.vhml.org.

[12] Mohamed Jemni, Oussama Elghoul, "A System to Make Signs Using Collaborative Approach", 11th International Conference, ICCHP 2008, Linz, Austria, July 9-11, 2008. Proceedings. Lecture Notes in Computer Science 5105 Springer 2008 pp. 670-677.

[13] Xerox portal, www.xrce.xerox.com.

[14] Zhigang Deng., Ulrich Neuman. “Data-Driven 3D Facial Animation”, 978-1-84628-906-4, Springer London, 2008.

[15] Ekman P., Friesen W.V. “Facial action coding system”. Consulting Psychologists Press, Palo Alto, Calif, 1978.

[16] MPEG4 (2001). MPEG Working Group on VISUAL. International standard on coding of audio-visual objects, Part 2 (Visual). ISO/IEC 14496-2 :2001.

[17] WebSign project, www.utic.rnu.tn/websign.