This paper presents a novel approach for generating VRML animation sequences from Sign Language notation, based on MPEG-4 Face and Body Animation. Sign Language notation, in the well-known SignWriting system, is provided as input and is initially converted to SWML (Sign Writing Markup Language), an XML-based format that has recently been developed for the storage, indexing and processing of Sign Writing notation. Each basic sign, namely signbox, is then converted to a sequence of Body Animation Parameters (BAPs) of the MPEG-4 standard, corresponding to the represented gesture. Inverse Kinematics are also employed for synthesizing complex animation sequences (e.g. contacts). In addition, if a sign contains facial expressions, these are converted to a sequence of MPEG-4 Facial Animation Parameters (FAPs), while exact synchronization between facial and body movements is guaranteed. These sequences, which can also be coded and/or reproduced by MPEG-4 BAP and FAP players, are then used to animate H-anim compliant VRML avatars, reproducing the exact gestures represented in the sign language notation. Envisaged applications include interactive information systems for the persons with hearing disabilities (Web, E-mail, info–kiosks) and automatic translation of written texts to sign language (e.g. for TV newscasts).

Keywords: Sign synthesis, SWML, MPEG-4 Face and Body Animation

1 INTRODUCTION

The SignWriting system is a writing system for deaf sign languages developed by Valerie Sutton for the Center of Sutton Movement Writing, in 1974 (Official SignWriting site.1). A basic design concept for this system was to represent movements as they are visually perceived, and not for the eventual meaning that these movements convey. In contrast, most of the other systems that have been proposed for writing deaf sign languages, such as HamNoSys (the Hamburg Notation System) or the Stokoe system employ alphanumeric characters, which represent the linguistic aspects of signs. Almost all international sign languages, including the American Sign Language (ASL) and the Brazilian Sign Language (LIBRAS), can be represented in the SignWriting system. Each sign-box (basic sign) consists of a set of graphical and schematic symbols that are highly intuitive (e.g. denoting specific head, hand or body postures, movements or even facial expressions). The rules for combining symbols are also simple, thus this system provides a simple and effective way for common people with hearing disabilities that have no special training in sign language linguistics, to write in sign languages. Examples of SignWriting symbols are illustrated in Figure 1.

![Figure 1: Three examples of representations of American Sign Language in SignWriting system.](image-url)
such a visualization of sign-boxes, available in SWML. The proposed technique first converts all individual symbols found in each signbox to sequences of MPEG-4 Face and Body Animation Parameters. The resulting sequences can be used to animate any H-anim-compliant VRML avatar using MPEG-4 SNHC BAP and FAP players, provided by EPFL. The system is able to convert all hand symbols as well as the associated movement, contact and movement dynamics symbols contained in any ASL sign-box. Manual (hand) gestures and facial animations are currently supported, while we plan to implement other body movements (e.g. torso) in the near future. Inverse kinematics techniques were used to generate BAP key-frames for specific movements (e.g. contacts), that are difficult to precisely define otherwise. Inverse kinematics (IK) are common techniques used to control the movement of a rigid multibody which is an assemblage of rigid links connected by joints. The proposed technique has significant advantages:

- **Web- (and Internet-) friendly visualization of signs.** No special software has to be installed except a VRML plug-in to a Web browser,
- **Allows almost real-time visualization of sign language notation, thus enabling interactive applications,**
- **Avatars can easily be included in any virtual environment created using VRML, which is useful for a number of envisaged applications,** such as TV newscasts, automatic translation systems for the deaf, etc.
- **Efficient storage and communication of animation sequences, using MPEG-4 coding techniques for BAP/FAP sequences.**

Significant similar work for producing VRML animations from signs represented in the HamNoSys transcription system to VRML has been carried out by the EC IST ViSiCAST project (6), and its follow-up E-Sign project (7). Current extensions of HamNoSys are able to transcribe all possible body postures, movements and facial expressions and significant work towards supporting MPEG-4 BAPs has been made. The main contribution of the proposed approach in this paper is the attempt to work towards the same direction for the most common and popular representation of Sign Languages, which is the SignWriting notation system. The paper is organized as follows: In Section 2, the proposed technique for converting SWML to MPEG-4 Face and Body Animation Parameters is described. The synthesis of animations for H-anim avatars and sample results are presented in Section 3, while the design of the experimental “Vsigns” Web page to evaluate the sign synthesis results is outlined in Section 4. Finally, conclusions are drawn and future work is presented in Section 5.

## 2 Conversion of SWML to MPEG-4 Face and Body Animation Parameters

SWML (Costa et al., 2) is an XML-based format that specifies two types of SWML documents: sw_text (sign language text generated e.g. an SWML editor or converter) and sw_table (sign language database or dictionary generated by an SWML aware application). This work focuses on the conversion of sw_text documents which contain SWML **sign_boxes**. Each signbox consists of a set of symbols, which is specified using the following fields:

- a) A shape number (integer) specifying the shape of the symbol,
- b) A variation parameter (0 or 1 for hand symbols / 1,2 or 3 for movement and punctuation symbols) specifying possible variations (complementary transformations) of the symbol,
- c) A fill parameter (0,1,2 or 3 for hand and punctuation symbols / 0,1 or 2 for movement symbols) specifying the way the shape is filled, generally indicating its facing to the signer,
- d) A rotation parameter (0-7) specifying a counter-clockwise rotation applied to symbol, in steps of 45 degrees,
- e) A transformation flip parameter (0 or 1) indicating whether the symbol is vertically mirrored or not, relatively to the basic symbol and, finally,
- f) The x and y coordinates of the symbol within the signbox.

Currently, symbols from the 1995 version of the Sign Symbol Sequence (SSS-1995) are supported. This sequence comprises an “alphabet” of the SignWriting notation system, while true images (in gif format) of each symbol contained in this sequence are available in (SWML site, 2004). The proposed system is able to convert:

- All 106 hand symbols,
- All 95 (hand) movement symbols
- Two punctuation symbols (180,181), which contain synchronization information.
- 27 Facial expression/animation symbols
- Other punctuation symbols as well as symbols that represent torso and shoulder movements (12 symbols) are currently not implemented (decoded) by the system.

For sign synthesis, the input for the sign synthesis system consists of the SWML entries of the signboxes to be visualized. For each signbox, the associated information corresponding to its symbols is parsed. Information related to symbols that are supported by the sign synthesis application, i.e. hand symbols as well as corresponding movement, contact and movement dynamics symbols, is
then used to calculate the MPEG-4 Body Animation Parameters. In the following, we describe the procedure to convert SignWriting notation in SWML format to MPEG-4 Face and Body Animation Parameters.

The issue of body modeling and animation has been addressed by the Synthetic/Natural Hybrid Coding (SNHC) subgroup of the MPEG-4 standardization group,(8). More specifically, 168 Body Animation Parameters (BAPs) are defined by MPEG-4 SNHC to describe almost any possible body posture. In addition, 68 Face Animation Parameters (FAPs) are used to describe almost any possible facial expression. Most BAPs denote angles of rotation around body joints, while FAPs usually denote movements of specific facial features (Facial Definition Points, FDPs) along a pre-determined axis in 3-D space.

The conversion of the symbols contained in a SWML signbox to BAP sequences starts by first examining the symbols contained within the input signbox. If no symbols describing dynamic information such as hand movements, contact or synchronization exist, the resulting BAP sequence corresponds to just one frame (i.e. a static gesture is reproduced). Information provided by the fields of the (one or two) hand symbols, contained in the signbox, is used to specify the BAPs of the shoulder, arm, wrist and finger joints. On the other hand, if symbols describing dynamic information exist, the resulting BAP sequence contains multiple frames, describing animation key-frames (i.e. a dynamic gesture is reproduced). The first key-frame is generated by decoding the existing hand symbols, as in the case of static gestures. Since the frame rate is constant and explicitly specified within a BAP file, the number of resulting frames may vary, depending on the complexity of the described movement and its dynamics. Synchronization and contact symbols also affect the represented movement and in some cases require special treatment.

When a signbox contains facial expression or animation symbols, the corresponding FAP frame(s) are determined by predefined lookup tables, which provide the FAP values defining one or more FAP frames per facial animation symbol. When two or more facial expression symbols co-exist within the same sign-box, these may either define an animation sequence or have to be combined all together (if each symbol activates different FAPs). The latter case, which is more common, is currently supported by the proposed system.

Smooth and natural-looking transitions between the Face and Body Animation parameters corresponding to each signbox is achieved by generating additional intermediate frames using a FAP/BAP interpolation procedure. A linear interpolation function is used to generate additional FAP/BAP frames to implement:

a) The transition between the neutral face/body position and the first frame of the first sign-box
b) The transition between the end frame of one signbox and the start frame of the next signbox
c) The transition between the end frame of the last sign-box and the neutral body position.

Furthermore, in order to achieve Face/Body synchronization:

a) The framerates defined for the FAP and BAP sequences should be equal
b) The number of generated FAP frames generated for each sign-box should be always equal to the corresponding number of BAP frames. In order to achieve this goal, the BAP frame sequence is first generated and then specific linear interpolation procedures are used to generate the FAP frame sequence.

Additional details about the generation of BAPs for static and dynamic gestures as well as the generation of FAPs for gestures containing facial expressions/animations are provided in the following Subsections.

2.1 Static gestures

The Sign Writing system allows various transformations to be applied to a basic symbol. A hand symbol for example can exist in many different postures with bent fingers etc, represented with different shape numbers. Also the signer may either see his/her palm, the back of his/her palm or the side of his/her palm (Figure 2).

![Figure 2: The signer sees a) his/her palm, b) the back of his/her palm c) the side of his/her palm.](image-url)
As seen in Figure 3, the hand may either be parallel with the wall (wall plane) or with the floor (floor plane).

![Figure 3: a) Hand is parallel with the wall plane b) Hand is parallel with the floor plane](image)

The position of the palm may also change due to a rotation around the wrist joint. Furthermore, a “flipped” symbol represents a symbol that is “mirrored” around the vertical axis. This means that it actually describes a posture of the other hand. A hand symbol and its flipped version are illustrated in Figure 4.

![Figure 4: A basic hand symbol and its flipped version.](image)

In the following, the procedure to extract useful information from the SWML representation of a hand symbol is summarized:

Initially, the binary “transformation flip” parameter is used to identify whether the symbol corresponds to the left or right hand. Then the fill and variation parameters of each symbol are used to determine the animation parameters of the shoulder and elbow joints:

- If \((\text{variation, fill}) = (0,0), (0,1) \text{ or } (1,3)\) then the axis of the arm is parallel to the floor (floor plane).
- If \((\text{variation, fill}) = (1,0), (1,1) \text{ or } (1,2)\) then the axis of the arm is parallel to the human body (wall plane).
- If \((\text{variation, fill}) = (1,0) \text{ or } (1,3)\) then the signer sees his/her palm.
- If \((\text{variation, fill}) = (1,1) \text{ or } (0,0)\) then the signer sees the side of his/her palm.
- If \((\text{variation, fill}) = (1,2) \text{ or } (0,1)\) then the signer sees the back of his/her palm.

In addition, the rotation parameter is used to determine the animation parameters of the wrist joint:

- If the signer sees the side of his/her palm, the rotation value (multiplied by 45 degrees) is used to define the R_WRIST_FLEXION BAP (for the right hand) or the L_WRIST_FLEXION BAP (for the left hand).
- In the other two cases (signer sees his/her palm or the back of his/her palm), the rotation value (multiplied by 45 degrees) is used to define the R_WRIST_PIVOT BAP (for the right hand) or the L_WRIST_PIVOT BAP (for the left hand).

Finally, the symbol shape number is used to specify the animation parameters corresponding to finger joints, using look-up tables of BAP values corresponding to each symbol.

If the signbox contains a second hand symbol, concerning a left palm, similar procedures are used to extract the body animation parameters of the other hand. After the processing of the first right and left existing hand symbols, all body animation parameters corresponding to shoulder, elbow, wrist and finger joints are determined and stored.

2.2 Dynamic gestures

MPEG-4 standard allows the description of human body movement using a specific set of Body Animation Parameters corresponding to each time instant. Systems like SignWriting that use a high level animation description define movement by specifying a starting and an ending position, in case of simple motion with constant velocity, or the full trajectory, in case of more complex motion. However, the description of complex motion is also possible by specifying a number of intermediate key-frames.

In SignWriting, there are three categories of dynamic symbols, namely:

- Movement symbols
Contact symbols and
Synchronization symbols
which are briefly described in Section 2.2.1. The procedures for generating these BAP key-frames are described in Section 2.2.2, and are based in the design of complex functions that have many parameters (symbol fields, etc). Since the design of these functions is usually difficult and perfect results are difficult to achieve, inverse kinematics techniques have been employed to generate BAP key-frames for specific movements (e.g. contacts), as described in Section 2.2.3. Finally, Section 2.2.4 describes the BAP interpolation techniques used to generate the final BAP sequence from the generated key-frames.

2.2.1 SignWriting dynamic symbols
Movement symbols
A movement symbol may exist in many forms describing either simple or complex movements. Movement can be either parallel to the wall plane or to the floor plane. Furthermore, as can be seen in Figure 5a, movement symbols for the left and right hand have different representations. When the movement is associated with the right (left) hand, the arrow representing its direction has a dark (light) arrowhead. When both hands are simultaneously moving to the same direction as a group, the representation of the movement is done using a neutral arrowhead, which is neither dark nor light. In some cases, the size of a movement symbol is used to specify the duration (i.e. the speed) of the hand.

For example, the arrow symbol in Figure 5b is illustrated in three different sizes: the first represents a fast movement forward, the second represents a movement forward with normal speed and the last represents a slow movement forward.

Contact symbols
A signbox, which represents a dynamic gesture in addition to movement symbols may also contain one or more contact symbols. Of course movement symbols may not exist in a dynamic gesture but just contact symbols. The existing contact symbols in SignWriting are illustrated in Figure 6.


Figure 5: Three versions of a symbol specifying: a) movements of different hands, b) movements with different time durations.

2.2.2 Generation of BAP key-frames
The shape number field of movement description symbols, which indicates the symbol shape, indicates the type of movement. First, the total number of key-frames to be produced is determined, based on the number and nature of the available movement, movement dynamics, contact, and synchronization symbols. More specifically,
a look-up table is used to define an initial number \( k \) of key frames for each movement symbol. Furthermore, the fill parameter specifies whether the motion is slow, normal or fast. In addition, some symbols explicitly specify the movement duration. For this reason, a classification of such symbols into three categories has been defined and a different duration value \( D \) is defined for each category:

- Slow motion (\( D=3 \))
- Normal motion (\( D=2 \))
- Fast motion (\( D=1 \))

The total number of frames to be generated when only one motion symbol exists is \( N=kDP \), where \( P \) is a fixed multiplier (e.g. \( P=10 \)). If the number more than one such symbols, the total number of key-frames is the maximum between the numbers of key-frames, corresponding to each symbol. Finally, if the signbox contains a contact symbol, the total number of frames is increased by two (in case of simple contact) or four (in case of double contact).

The initial key-frame is generated by decoding the available hand symbols, exactly as in the case of static gestures. The rotation and transformation flip fields specify the exact direction of movement. Also, the variation field specifies whether the right or the left hand performs the movement. Using information from all available dynamic symbols, the remaining BAP key-frames of the specific dynamic gesture are then generated from a specific function for each key-frame, which depends on the values of the symbol fields (variation number etc) for all available dynamic symbols.

However, the proper design of these functions is often difficult and the results are in some cases imperfect. In the case of palm contacts, Therefore, we have decided to employ inverse kinematics techniques for calculating specific key-frames (e.g. a key-frame corresponding to a palm contact). These procedures are described in detail in Section 2.2.3.

Synchronization (Movement Dynamics) symbols (180,181 and 182) are handled in a similar way as movement symbols. An exception is the “Un-even alternating” symbol, where first one hand moves, while the other is still and then the opposite. Thus, in this case, the total number of key frames is doubled (\( N=2kDP \)), since \( kDP \) frames are generated for the first hand, while the second hand remains still and vice versa.

After the generation of the key-frames related to the available hand, contact, movement and synchronization symbols, it is checked whether more palm postures for the right or both of the hands exist. If there are more than one palm symbols for one or both hands, additional key-frame(s) are generated containing the values of the BAPs which represent the final palm position(s).

2.2.3 Use of Inverse Kinematics (Selectively Damped Least Squares Method) for complex gestures

Inverse kinematics techniques were used to generate BAP key-frames for specific movements (e.g. contacts), that are difficult to precisely define else ways. Inverse kinematics (IK) are common techniques used to control the movement of a rigid multibody which is, an assemblage of rigid links connected by joints (Buss, 10). The IK problem is defined as follows: joint values are determined so that specific points of the multibody, called “end effectors”, are placed into desired target positions.

There are several methods for solving IK problems, stemming originally from robotics applications. These include cyclic coordinate descent methods, pseudoinverse methods, Jacobian transpose methods, Levenberg-Marquardt, quasi-Newton and conjugate gradient methods, neural net and artificial intelligence methods (Buss, 10). The most common applications are animating humans or creatures by specifying the positions, and possibly the orientations, of their hands, feet and head. In this work, we use a refinement of the Damped Least squares Method, called Selectively Damped Least Squares (SDLS) (Buss,11), which can be viewed as an extension of the numeric filtering of Maciejewski and Klein, (12).

Our technique is applied only to rotational joints, whose configuration is specified by one or more scalar values, describing the angle values (degrees of freedom) of a rotational joint. The complete configuration of the multibody is specified by \( n \) unknown scalars \( \theta_1,\ldots,\theta_n \) (joint angles) describing the configuration of all joints (Buss, 10). The positions of the \( k \) end effectors are denoted by a vector \( s=(s_1,\ldots,s_k) \). The (desired) target positions are also expressed by a vector \( t=(t_1,\ldots,t_k)^T \), where \( t_i \) is the target position for the \( i \)th end effector, and \( e_i = t_i - s_i \) is the corresponding error. If \( \theta = (\theta_1,\ldots,\theta_n)^T \) is the column vector of the unknown joint angles, the position of the \( j \)th end effector is given by a function \( s_j(\theta) \), \( 1 \leq j \leq k \) of the joint angles. In vector notation, this can be expressed as \( s=s(\theta) \), where \( s_j = s_j(\theta) \). According to the IK problem we seek values for the \( \theta_j \)’s such that \( t_i = s_i(\theta) \), for all \( i \). These equations can be solved by iterative local search based on the \( n \times n \) Jacobian matrix \( J \) whose elements are defined by:

\[
J_{i,j} = \frac{\partial s_i}{\partial \theta_j}.
\]

According to that iterative method, the current values of \( \theta \), \( s \) and \( t \) are used for the computation of a value \( \Delta \theta \) and the incrementing of the joint angles \( \theta \) by \( \Delta \theta \). Since \( \Delta s = J(\theta) \Delta \theta \), the resulting change in end effector positions can be estimated as \( \Delta s \approx JA\theta \). The angle update may be performed either once per frame so that the end effectors only approximately follow the target positions, or iteratively until the end effectors are sufficiently close to the targets.

The entries in the Jacobian matrix are usually easy to calculate. If \( p_j \) is the position of the joint, \( v_j \) is a unit
vector pointing along the current axis of rotation for the joint, and the ith end effector is affected by the joint, then the corresponding entry in the Jacobian \( \frac{\partial \mathbf{s}_i}{\partial \theta_j} = \mathbf{v}_j \times (\mathbf{s}_i - \mathbf{p}_j) \), where angles are measured in radians with the direction of rotation given by the right rule. If the ith end effector is not affected by the jth joint, then \( \frac{\partial \mathbf{s}_i}{\partial \theta_j} = \mathbf{0} \).

The update value \( \Delta \theta \) is computed using the Selectively Damped Least Squares (SDLS) method (Buss,11), where the damping constants depend not only on the current configuration of the articulated multibody, but also on the relative positions of the end effector and the target position as well as on the difficulty of reaching the target rather than just the distance to the target.

To apply the SDLS method for the generation of a specific gesture, two IK trees are initially defined, for the right and the left hand respectively, each containing nodes, corresponding to the degrees of freedom of the shoulder, elbow, and wrist joints. More specifically, each node of a tree represents a degree of freedom of an avatar’s joint, as follows:

- shoulder_twisting \( \rightarrow \) rotation axis: y
- shoulder_flexion \( \rightarrow \) rotation axis: x
- shoulder_abduct \( \rightarrow \) rotation axis: z
- elbow_flexion \( \rightarrow \) rotation axis: x
- wrist_flexion \( \rightarrow \) rotation axis: z (wall plane), rotation axis: y (floor plane)

The inputs in the inverse kinematics technique are the 3D coordinates of each joint, calculated from the BAPs values of the joints at the previous key frame. One end effector is determined for each tree based on the gesture to be synthesized (e.g. the tip of each index finger, in the case of a contact gesture between these two fingers). The corresponding target positions are also defined based on the desired (contact) point for the two effectors.

The output of the inverse kinematics technique is the rotation angle about the rotation axis defined for each joint node, which can be used to compute the BAPs values of the above input joints. Following the above, key-frames corresponding to complex gestures (e.g. single contacts or double contacts) can be accurately synthesized after the convergence of the SDLS method.

### 2.2.4 BAP Interpolation

Finally, when the BAPs for all key-frames have been computed, BAP interpolation is used to increase the frame rate of the resulting BAP sequence. This interpolation procedure results to smoother transitions between key frames.

Interpolation is generally achieved by approximating the motion equation using a mathematical function and then re-sampling this function to obtain the desired intermediate positions at intermediate time instants. Various interpolation functions can be selected in order to improve results. Since Body Animation Parameters represent rotations around specific joints, quaternion interpolation was seen to provide good results (Preda et al, 8), but the complexity of the method is increased. For this reason, a linear interpolation technique was applied, which was seen to be very efficient for most signs, since key-frames have been selected so as to simplify the movement description between consecutive key-frames.

### 2.2.5 Gestures containing facial expressions–animations

The generation of the FAP frame sequence is performed after the generation of the BAP frame sequence, so that the total number of generated FAP frames is exactly the same as the total number of BAP frames. For each sign-box, the FAP key-frames are determined, based on the existing facial expression/animation symbols, from predefined lookup tables for each symbol. The number of FAP key-frames, \( N_{\text{FAP\_keyframes}} \), is generally much smaller than the total number of BAP frames \( N_{\text{BAP}} \) that have been already generated using the procedures described in the previous Subsections. Therefore, if \( FAP(k), k = 0, \ldots, N_{\text{BAP}} - 1 \) denotes the vector of FAPs corresponding to frame \( k \), the FAP key-frames are first positioned every \( s = N_{\text{BAP}} / (N_{\text{FAP\_keyframes}} - 1) \) frames:

\[
FAP(i \cdot s) = FAP\_keyframe(i), i = 0, \ldots, N_{\text{FAP\_keyframes}} - 1
\]

Then, each of the remaining FAP frames is determined using linear interpolation between the two closest available FAP key-frames.

### 3 SYNTHESIS OF ANIMATIONS USING H-ANIM AVATARS

The "EPFLBody" BAP player (Vergnenegre et al, 9), developed by the École Polytechnique Fédérale Lausanne (EPFL) for the Synthetic and Natural Hybrid Coding (SNHC) subgroup of MPEG-4 was used to animate H-anim-compliant avatars using the generated BAP sequences. Since most BAPs represent rotations of body parts around specific body joints, this software calculates and outputs these rotation parameters as animation key-frames to produce a VRML ("animation description") file that can be used for animating any H-anim-compliant VRML avatar. The “Miraface” FAP player, also...
developed for MPEG-4 SNHC, by MIRALab, University of Geneva and LIG, EPFL was used for Facial Animation. This software had to be modified so that:

a) VRML animation output is produced using one CoordinateInterpolator node per face model vertex. A problem with the chosen implementation is that the computational demands for the hardware that is reproducing these animations are increased. A possible solution for this problem that should be investigated in the future is to add CoordinateInterpolator nodes only for the points that have actually been moved.

b) The face model to be animated using the FAP frame sequence was attached to the body to be animated using the BAP frame sequence. Some slight modifications of the VRML face model were also required (e.g. addition of teeth).

In the following, two example synthesis results of dynamic gestures are illustrated in Figure 8 and Figure 9.

In Figure 10 and Figure 11, two gestures that include contact action between palms are synthesized using the Inverse Kinematics described in Section 2.2.3. In both cases, precise contact is achieved, which proves the ability of the IK method to achieve accurate synthesis of complex gestures.

Figure 10: The contact frame of the sign “above”

Figure 11: The contact frame of the sign “with”

By including a VRML TouchSensor Node within the VRML file describing the H-anim avatar, the viewer can interactively start and/or replay the animation sequence, by clicking on the avatar. The viewer can also interact by zooming in and out to any specific body region and/or by rotating and translating the model within the 3-D space, in order to fully understand the represented sign.

4 THE VSIGNS WEB APPLICATION

Furthermore, further evaluation of the proposed sign synthesis system was possible by developing an online system (Vsigns, http://vsigns.iti.gr) for converting text to Sign Language notation and corresponding VRML animation sequences for H-anim compliant avatars. The application, whose interface is illustrated in Figure 12, is currently based on a 3200-word SWML dictionary file, which has been parsed and inserted into a relational database. The user is allowed to enter one or more words, which are looked up in this dictionary. If more than one entry is found, all possible interpretations are presented to the user, so that he can choose the desired one. On the
other hand, if no entries are found for a specific word, the word is decomposed using its letters (finger-spelling). In any case, the user may choose whether to include a particular term to the selected terms to be used for sign synthesis or not. The user then selects a column corresponding to an H-anim compliant avatar, which is used for sign synthesis of the selected term or terms. A fourth column (“Baxter FBA”) allows the user to observe facial animation in addition to body animation, using the modified “Baxter avatar”. Furthermore, the user may produce and display the corresponding sign(s) in SignWriting format (in PNG format) and SWML for a specific term or the selected terms.

Further evaluation is planned for the future, using Greek and International SignWriting users, and attempts will be made to solve possible problems in the reproduction of specific signs. Although these problems indicate that more work is needed for correct synthesis of all signs, we believe that with this Web tool, a very important step towards automatic Text to Sign synthesis has been made.

5 DISCUSSION AND FUTURE WORK

A novel approach for generating VRML animation sequences from Sign Language notation, based on MPEG-4 Body Animation has been presented. The system is able to convert almost all hand symbols as well as the associated movement and dynamics symbols contained in any ASL sign-box. Furthermore, most facial expression and animation symbols are also supported while some very complex gestures will be able to be generated soon. Improved reproductions of palm touching movements have been implemented using inverse kinematics techniques. In the future, more cases of gestures containing contact symbols, for example gestures containing contact between palm and head or palm and elbow, will be actuated with the utilization of those techniques. Torso and head movements will be also supported in the near future. Some facial expressions, e.g. cheek wrinkles, have not been implemented, since no FAPs exist to produce such movements. Results are satisfactory and are currently being evaluated by SignWriting users and experts, so that problems associated with specific SignWriting symbols are identified and solved.

Acknowledgement: This work was supported by FP6 IST Network of Excellence “SIMILAR” -“The European taskforce creating human-machine interfaces SIMILAR to human-human communication”.

6 REFERENCES

7. Official site of E-sign (Essential Sign Language Information on Government Networks) project. http://www.visicast.sys.uea.ac.uk/eSIGN/


