



Design and Evaluation of a User-Interface for Authoring Sentences of American Sign Language Animation

Abhishek Kannekanti¹, Sedeeq Al-khazraji^{1,2},
and Matt Huenerfauth¹(✉)

¹ Rochester Institute of Technology, Rochester, NY 14623, USA
{ak9492, sha6709, matt.huenerfauth}@rit.edu
² University of Mosul, Mosul, Iraq

Abstract. Many individuals who are Deaf or Hard of Hearing (DHH) in the U.S. have lower English language literacy levels than their hearing peers, which creates a barrier to access web content for these users. In the present study we determine a usable interface experience for authoring sentences (or multi-sentence messages) in ASL (American Sign Language) using the EMBR (Embodied Agents Behavior Realizer) animation platform. Three rounds of iterative designs were produced through participatory design techniques and usability testing, to refine the design, based on feedback from 8 participants familiar with creating ASL animations. Later, a usability testing session was conducted with four participants on the final iteration of the designs. We found that participants expressed a preference for a “timeline” layout for arranging words to create a sentence, with a dual view of the word-level and the sub-word “pose” level. This paper presents the details of the design stages of the new GUI, the results, and directions for future work.

Keywords: Human computer interaction · American Sign Language · Authoring tools · Timeline

1 Introduction

American Sign Language (ASL) is the primary means of communication for over one and a half million people in the United States of America [10], and it is ranked as the third most popular “foreign language” studied in U.S. universities [3]. The grammar of ASL is different than English; it has a unique word-order and vocabulary. Standardized literacy testing has revealed that many deaf secondary-school graduates in the U.S. have lower levels of written and reading English skills [14]. Hence, the use of English text captions on dynamic multimedia content (TV, radio, movies, and computer programs) may also be difficult to understand for such users. Several sign language writing systems have been proposed [12, 13] but have not gained popularity among the deaf community. Therefore, if website or media designers wish to provide information in the form of ASL for these users, they must provide a video or an animation of the ASL information content [6].

What is required is a practical way for adding sign language to media or websites. While providing video recordings of a human is possible, it is expensive to remake a video of a human producing sign language, for content that is frequently changing. However, with animated characters, the content could be dynamically created from a “script” of the message, which could be more easily updated. Such an approach could provide high-quality ASL output that is easier to update for websites or media. While some researchers have investigated the problem of automatically translating from written English into a script of an ASL message [11], currently a human who is knowledgeable of ASL is needed to produce such a script accurately. Therefore, there is a need for software to help this person produce ASL sentences or longer messages that use ASL signs in proper word-order, with other details of the animation correct [9], such as the timing of words, the facial expression of the animated human, etc.

1.1 Problem Statement

Beyond this broader future use of ASL animation technology to produce content for websites, there is a more immediate need for software tools: Our laboratory uses the EMBR animation platform [8] as a basis for its research on ASL animation technology, and as part of this work, the lab is often producing new animations of sentences (or longer messages) in ASL. For instance, the lab may need to produce sentences that will be shown during an experiment, in order to evaluate some of the technical details of the animation technology. Thus, as an important form of research infrastructure, this tool should enable someone to build sentences as well as to identify and add items from a repository of individual ASL signs (words) that members of the laboratory had previously authored and saved for future use. Being able to make use of these pre-built words from the repository reduces the effort needed for creating a new sentence, by allowing someone to use pre-built components (the individual ASL sign animations that have already been created).

This paper reports on our design and evaluation of a software user-interface to enable someone to author sentences (or multi-sentence messages) in ASL. The primary users of this system would be researchers who are investigating ASL animation technologies, but we anticipate that understanding how to produce a useful research system for authoring such sentences may also shed light on how to best design an ASL animation authoring system that could be used in the future outside of a laboratory context. To investigate the design of this software, we utilized participatory design techniques, interviews, and iterative prototyping methodology. The users who participated in this design and evaluation were “expert users” – i.e. researchers at the Linguistic and Assistive Technology (LATLab) at the Rochester Institute of Technology.

While the laboratory had an existing software system for allowing someone to reuse individual signs to build entire sentences (or longer messages), as described in [7], current users of that software reported that it is too complicated to use. For instance, users had reported that, to create a new ASL sentence, it took about 3 h for an experienced user to import pre-built ASL signs from the lab’s existing collection, adjust the timing properties of these signs, and to make small modifications to the individual signs to enable them to smoothly flow from one to the next. Hence a more intuitive and efficient user experience was necessary for generating ASL animations.

The laboratory's current animation platform is based on EMBR [9], and the existing user-interface for assembling words into sentences is a Java application called Behavior Builder, which is distributed with the open-source EMBR software. The Behavior Builder software can be used to perform two different functions: (1) creating an individual ASL sign and (2) assembling signs into a sentence or longer message (and customizing aspects of the signs for this purpose, e.g. adding additional details to the animation, such as facial expressions). Therefore, the system allows users to modify different aspects of the virtual human animation (e.g., the hand shapes, facial expressions, torso positions, etc.) to produce an animation of a message.

In work reported in this paper, our focus is on this second functionality (2): enabling the user to assemble signs into a sentence or a longer message. While the existing system had been a Java application, the lab is currently re-implementing the authoring tool as an HTML5 application, based on the proposed final design from the design and usability testing process described in this paper.

2 Prior Work

Some researchers provided an overview of sign language generation and translation [8], technologies that would plan ASL sentences in a fully automatic manner, when given an English input sentence. While there is continuing research on machine translation for sign languages, progress in this field is still limited by the challenging linguistic divergences encountered during text-to-sign translation and the relatively small size of training data (collections of linguistically labeled ASL recordings with parallel English transcription) [7]. Furthermore, members of the Deaf community have expressed concerns about the deployment of such technologies before they are proven to be of high quality [16]. For this reason, as discussed in [1] our laboratory investigates "human in the loop" workflows for generating ASL animation, in which a human who is knowledgeable of ASL produces a script for an intended message (specifying the words in the sentences, their order, and other details), software automatically generates animations of a virtual human performing ASL (more efficiently than if the human author had been required to animate the movements of the virtual character manually). In this context, it is necessary to provide "authoring tools" for users to develop scripts of ASL animations.

Authoring software would allow users to build sentences of sign language by ordering single signs from a database onto a timeline. Later the authoring software produces an animation of a virtual character based on this user-built timeline. There have been several prior commercial and research efforts to create such tools: For instance, Sign Smith Studio [15] was a commercially available software product that allowed users to make ASL animations on a timeline, using a relatively small dictionary of several hundred words. However, at this time, the product is no longer available commercially.

While the Sign Smith Studio system enabled users to build sentences, there was a complimentary product called "Vcom3D Gesture Builder" [15], which could be used to create entirely new signs, to expand the dictionary of words available for use when building sentences in the Sign Smith Studio system. The Gesture Builder software had

several intuitive controls for adjusting the orientation and location of the hand through direct manipulation (click and drag) interaction. The software also featured a timeline of the individual “poses” that comprise a sign, to allow the user to flexibly in changing the hand shape and orientation, by adding time segments for both the hands. The software also contained a repository of pre-built hand shapes, categorized based on the number of fingers, which allowed users to select a hand shape to build ASL sign. However, at this time, the product is also no longer available commercially.

Another example of an authoring system for (European) sign languages was the eSIGN project [2], which created a plugin for a web browser that enabled users to see animations of a sentence on a web page. The technology behind the plugin allowed users to build a sign database using a symbolic script notation called “HamNoSys” in which users typed symbols to represent aspects of the handshape, movement, and orientation of the hands. However, that input notation system was designed for use by expert linguists who are annotating the phonetic performance of a sign language word, and it has a relatively steep learning curve for new users [4].

A recent research project [17] has investigated the construction of a prototype sign language authoring system, with design aspects inspired by word-processing software. While building the prototype, the researchers identified two major problems that their users encountered during the authoring task: (1) retrieving the sign language ‘words’ (from the collection of available words in their system’s dictionary) and (2) specifying the transition movements between different words.

Our lab’s EMBR-based animation system uses a constraint-based formalism to specify kinematic goals for a virtual human [5, 7, 8]. While this platform enables keyframe-based animation planning for ASL animation, as discussed in Sect. 1.1, the GUI provided with EMBR was not optimized for selecting ASL signs from a pre-existing word-list nor efficiently adjusting timing characteristics specific to ASL. Thus, we investigate a new user-interface for planning ASL sentences.

3 Methodology

3.1 Iterative Prototyping and User Interviews

To design and evaluate a new user-interface for authoring animations using our EMBR web application, we opted to use Lean UX methodologies, which include rapid sketching, prototyping, creating design mockups, and collecting user feedback. Observation of current users of the existing authoring software as they created animations was conducted to understand current limitations of the system. Next, we presented iterative prototypes to users during three rounds of user studies with a total of 8 to 12 participants in each round (1 to 3 female and 7 to 9 males, in each round). The participants consisted of research students working at the Linguistic and Assistive Technologies Laboratory at Rochester Institute of Technology who had some experience in ASL and familiarity with the need for the laboratory to produce animations of ASL periodically in support of ongoing research projects, e.g. to produce animation stimuli for experimental studies.

The designs presented in the study consisted of high-fidelity static prototype images illustrating various steps in the use of the proposed software. In each round, participants provided subjective feedback about the proposed design iteration. At the end of each presentation round, the feedback and suggestions in the form of qualitative data were synthesized (via affinity diagramming), and changes were applied to the designs. User suggestions in each round (e.g. feature request or UI changes) became test hypotheses for the subsequent round.

Participants in each round of feedback were asked to consider how they would create an ASL animation for the English sentence “Yesterday, my sister brought the car.” In addition to creating the sentence, participants were asked to add some additional pause time after one word (additional “Hold” time), and they were asked to adjust the overall speed of one of the words in the sentence (the “Time Factor” for the word).

The final design which resulted from this iterative design process is presented in Fig. 1, with detailed images available in subsequent figures. As shown in this figure, there is a “Left Portion” of the GUI that consists of a list (top to bottom) of the sequence of words in the sentence being constructed (beginning to end). Each word that is listed in the Left Portion of the GUI represents an ASL word from our system’s available word list. The numbers shown to the right of each word consist of the start-time and end-time for each word on the sentence, in milliseconds. When words are added to a sentence from the dictionary, their duration is initially set to the duration of the word as stored in the dictionary, and an automatic gap of 30 ms is placed between words. The word duration and time between words are both adjustable (as discussed below, when individual elements of the GUI are described in greater detail).



Fig. 1. The final design for authoring ASL sentences, resulting from three rounds of iterative design and testing.

The Right Portion of the GUI contains another view of the information shown in the Left Region. It displays a corresponding representation on the left-to-right timeline for each word of the sentence. As changes are made in one region, the changes are

reflected in the other. The rationale for this dual representation of the sentence was that a right-to-left timeline is a more traditional method of displaying a sequence of words in a sentence, yet because an individual ASL sign may consist of multiple “poses” (multiple key frames that occur during time that represent individual landmark positions of the hands in space during time), it is useful to have a method of viewing the detailed numerical information for a sign or a pose in the vertical list-like arrangement in the Left Region of the GUI.

This sentence-authoring user-interface shown in Fig. 1 is only a portion of an overall word-authoring and sentence-authoring tool. In other work at our laboratory, we are investigating methods for controlling the pose of a virtual human character in order to create an individual ASL word (sign), and therefore the upper portion of the screenshot shown in Fig. 2 displays the user-interface elements for that word-authoring task. In this paper, we are specifically focused on our efforts to create a user-interface for assembling signs into a sequence (for a sentence or a longer message), which consists of the lower portion of the screenshot shown in Fig. 2.

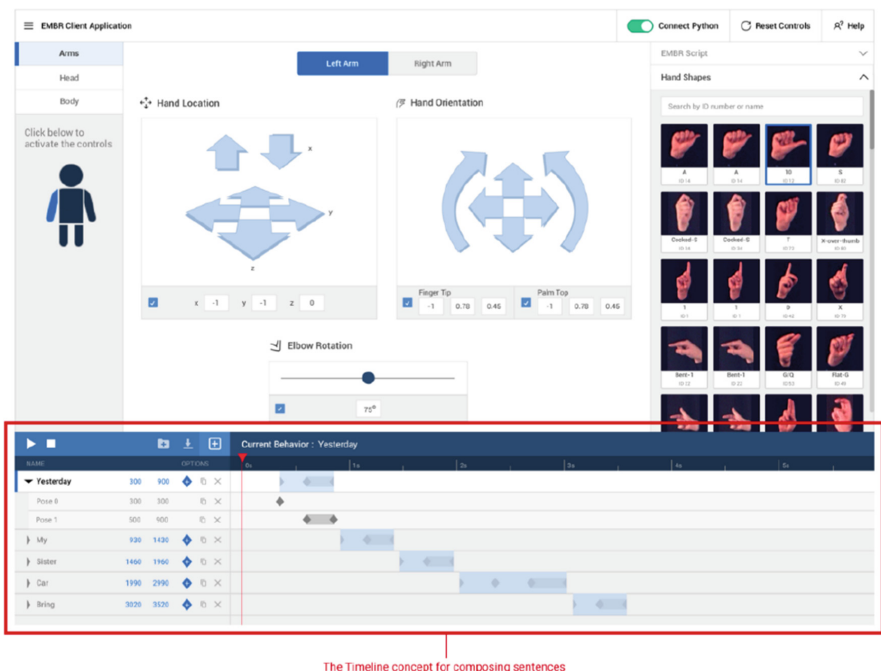


Fig. 2. The GUI of our overall user-interface for both word-authoring (moving the arms and hands to create a sign) and sentence-authoring (assembling words into sentences); the highlighted region shows the sentence-authoring timeline, which this the focus of this paper.

3.2 Detailed Discussion of Elements of the GUI’s Left and Right Portions

The Left Portion of the GUI (shown in Fig. 3), with its vertical list-like arrangement of information about the words in the sentence contains multiple controls, which include:

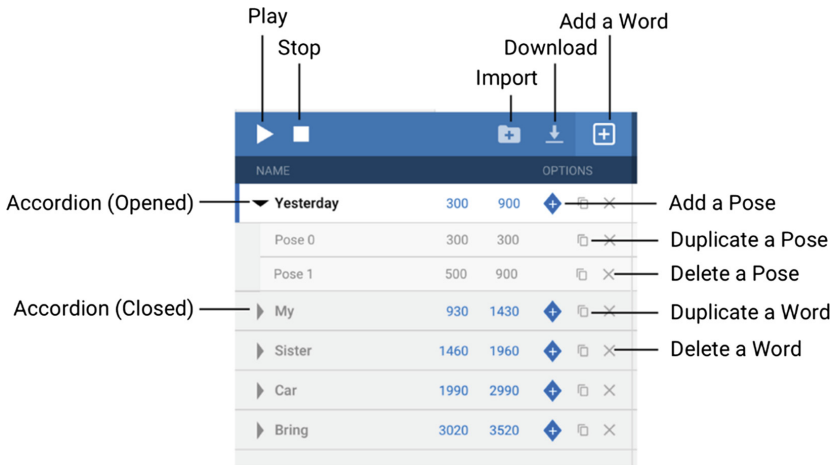


Fig. 3. Detailed of the Left Portion of the GUI, listing words in the sentence.

- Buttons for playing or stopping the display of animation of the virtual human character who performs the sentence.
- Buttons for importing a previously saved sentence sequence or downloading a local copy of the sentence sequence to the computer.
- An “Add a Word” button for inserting an additional word into the sentence.
- In addition, the time values shown for the start-time and end-time of a word (and each sub-pose of a word) may be edited by clicking and typing new number values.

For each individual word in the sentence, the Left Portion of the GUI also presents controls for:

- An “accordion” control for expanding the view of a single word so that the user can see detailed information about each “pose” that constitutes the individual key frames of movement of this ASL sign.
- Adding another pose to a word. In this case, the user would make use of the hand and arm movement controls on the top portion of Fig. 2 to manipulate the position of the virtual human for this newly inserted pose.
- Duplicating a word in the sentence.
- Deleting a word in the sentence.
- In addition, individual words can be dragged and dropped to rearrange their position.

When viewing detailed information about individual poses, there are analogous controls for editing time values, duplicating individual poses, or deleting individual poses.

As discussed above, the Right Portion of the GUI is simply an alternative left-to-right visualization of the sequence of words, which corresponds to the vertical top-to-bottom arrangement of words in the Left Portion of the interface. As shown in Fig. 4, as the animation is played, there is a playhead indicator that moves left-to-right to indicate the current time during the sentence animation. Each word on the timeline is presented as a translucent blue rectangle region, and each individual “pose” of the hands/arms that constitute a word is indicated as a grey diamond. If a “pose” has a non-instantaneous duration in time (i.e. if the virtual human holds its hands and arms motion-less of a brief period of time), then there will be a pair of grey diamond that are linked with a line, to indicate this duration information on the timeline. Notably, if the user uses the “accordion” control on the Left Portion of the interface to expand the view of a word (to see its component poses), then these individual poses are also displayed on the Right Portion timeline – appearing as gray diamonds that are displayed below their parent word.

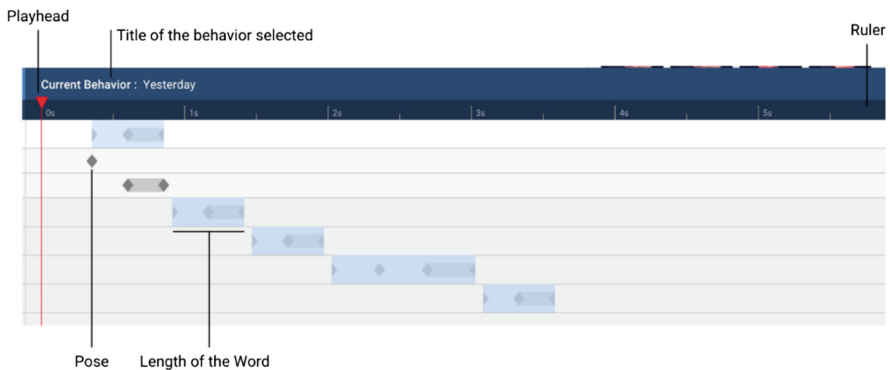


Fig. 4. Detailed view of Right Portion of GUI, showing words and poses.

3.3 Major Recommendations from All Rounds of Testing

While the previous section has described the elements of the final design, this section documents some of the feedback provided by participants during the iterative testing of the preliminary designs. Our first iteration of the interface had a simple design that used the left-to-right timeline metaphor only – without the vertical top-to-bottom list. Overall the left-to-right timeline layout resonated well with the participants, but they also wanted to have a way to control the timing of any selected pose or a word. In the second iteration, we added the Left Portion of the GUI, which revealed numerical values for the “start-time” of each word, as well as a number that represented the “duration” of the word in milliseconds. Participant reacted positively to having this specific numerical information revealed, but some expressed confusion about seeing a “start-time” and a “duration.” Several participants mistakenly interpreted the second

number as an “end time” for the word, which they believed would be more natural. Thus, our third and field iteration (as shown in Figs. 1, 2, 3 and 4) included both the start-time and the end-time of each word as text input fields (listed to the right of each word in the Left Portion of the interface).

After the final iteration, an additional usability study was conducted via interactive paper-prototyping (utilizing the high-fidelity static images from the final iteration) with four participants who were asked to step through the process of authoring an ASL sentence. No new issues were revealed in this final paper-prototyping session.

3.4 Limitations of This Study

A major limitation of the studies used during the design of this system is that high fidelity static images were displayed to users, rather than an interactive system. Thus, all of the step-by-step screens for the interface were presented as static screens, which limited users’ ability to try different ways of arranging the words. While the researcher had several paper prototype images available during these sessions, there were times that participants proposed “playing around with” the sequence of words in a sentence in an exploratory manner, which was too difficult to support through paper prototyping. Another limitation of this work was that the users were asked to consider a single ASL sentence; each round of user testing used the same sentence as an example (“Yesterday, my sister brought the car”). In addition, because we were targeting the design of research software for expert users, we were limited to recruiting users who had some experience in authoring ASL sentences for research studies. This restricted us to researchers from a specific laboratory, and furthermore, there was a disproportionate number of male participants.

4 Conclusion and Future Work

This paper has presented our design and formative evaluation of a user-interface for authoring an animation of an ASL sentence (or multi-sentence messages). Our user studies found that the interface was intuitive for expert users. Three rounds of iterative designs were conducted to understand the factors that influenced the efficiency of composing sentences to create animations. After each round of presentation, the interface design was updated with the feedback given by the participants – and also through validation with subject matter experts and developers, to determine whether the final design was implementable, given our existing software platform.

This work has only presented the design process for prototyping a new user-interface design for this sentence authoring task. The implementation of the GUI is ongoing work at our laboratory. Hence, after the implementation of a working prototype that is compatible with our animation platform, we anticipate that in future work, we will investigate the usability and efficacy of the resulting system. For instance, we anticipate measuring the time needed to create a sentence on a working prototype – and to conduct an A/B testing with the existing EMBR application to examine the efficiency.

Acknowledgements. This material is based upon work supported by the National Science Foundation under award number 1400802 and 1462280.

References

1. Al-khazraji S., Berke, L., Kafle, S., Yeung P., Huenerfauth, M.: Modeling the speed and timing of American Sign Language to generate realistic animations. In: Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2018), pp. 259–270. ACM, New York (2018)
2. Elliott, R., Glauert, J., Kennaway, J., Marshall, I., Safar, E.: Linguistic modelling and language-processing technologies for Avatar-based sign language presentation. *Univ. Access Inf. Soc.* **6**(4), 375–391 (2006)
3. Goldberg, D., Looney, D., Lusin, N.: Enrollments in languages other than English in United States Institutions of Higher Education, Fall 2013. In Modern Language Association. Modern Language Association, New York (2015)
4. Hanke, T.: HamNoSys-representing sign language data in language resources and language processing contexts. *LREC* **4**, 1–6 (2004)
5. Heloir, A., Nguyen, Q., Kipp, M.: Signing avatars: a feasibility study. In: 2nd International Workshop on Sign Language Translation and Avatar Technology (SLTAT) (2011)
6. Huenerfauth, M., Hanson, V.: Sign language in the interface: access for deaf signers. In: *Universal Access Handbook*. Erlbaum, NJ, p. 38 (2009)
7. Huenerfauth, M., Kacorri, H.: Augmenting EMBR virtual human animation system with MPEG-4 controls for producing ASL facial expressions. In: The Fifth International Workshop on Sign Language Translation and Avatar Technology (SLTAT), Paris, France (2015)
8. Huenerfauth, M., Lu, P., Rosenberg, A.: Evaluating importance of facial expression in American Sign Language and Pidgin Signed English animations. In: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 99–106. ACM, New York (2011)
9. Kacorri, H., Lu, P., Huenerfauth, M.: Evaluating facial expressions in American Sign Language animations for accessible online information. In: Stephanidis, C., Antona, M. (eds.) *UAHCI 2013*. LNCS, vol. 8009, pp. 510–519. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39188-0_55
10. Mitchell, R.: How Many Deaf People are There in the United States. Gallaudet Research Institute, Graduate School and Professional Programs, Gallaudet University, Washington, DC (2004)
11. Morrissey, S., Way, A.: Manual labour: tackling machine translation for sign languages. *Mach. Transl.* **27**(1), 25–64 (2013)
12. Newkirk, D.: *SignFont Handbook*. Emerson and Associates, San Diego (1987)
13. Sutton, V.: The SignWriting Literacy Project. Presented at the Impact of Deafness on Cognition AERA Conference, San Diego, CA (1998)
14. Traxler, C.B.: The Stanford Achievement Test: national norming and performance standards for deaf and hard-of-hearing students. *J. Deaf Stud. Deaf Educ.* **5**(4), 337–348 (2000)
15. VCom3D. Sign Smith Studio. <http://www.vcom3d.com/language/sign-smith-studio/>. Accessed 10 July 2018
16. World Federation of the Deaf. WFD and WASLI Issue Statement on Signing Avatars. <https://wfdeaf.org/news/wfd-wasli-issue-statement-signing-avatars/>. Accessed 15 Jan 2019
17. Yi, B., Wang, X., Harris, F.C., Dascalu, S.M.: sEditor: a prototype for a sign language interfacing system. *IEEE Trans. Hum. Mach. Syst.* **44**(4), 499–510 (2014)