J Braz Comput Soc (2013) 19:107–125
DOI 10.1007/s13173-012-0086-2

ORIGINAL PAPER

Automatic generation of Brazilian sign language windows for digital TV systems

Tiago Maritan Ugulino de Araújo · Felipe Lacet Silva Ferreira · Danilo Assis Nobre dos Santos Silva · Felipe Hermínio Lemos · Gutenberg Pessoa Neto · Derzu Omaia · Guido Lemos de Souza Filho · Tatiana Aires Tavares

Received: 26 August 2011 / Accepted: 7 August 2012 / Published online: 11 September 2012

Abstract Deaf people have serious difficulties accessing information. The support for sign language (their primary means of communication) is rarely addressed in information and communication technologies. Furthermore, there is a lack of works related to machine translation for sign language in real-time and open-domain scenarios, such as TV. To minimize these problems, in this paper, we propose an architecture for machine translation to Brazilian sign language (LIBRAS) and its integration, implementation and evaluation for digital TV systems, a real-time and open-domain scenario. The system, called LibrasTV, allows the LIBRAS windows to be generated and displayed automatically from a closed caption input stream in Brazilian Portuguese. LibrasTV also uses some strategies, such as low time consuming, text-to-gloss machine translation and LIBRAS dictionaries to minimize the computational resources needed to generate the LIBRAS windows in real-time. As a case study, we implemented a prototype of LibrasTV for the Brazilian digital TV system and performed some tests with Brazilian deaf users to evaluate it. Our preliminary evaluation indicated that the proposal is efficient, as long as its delays and bandwidth are low. In addition, as previously mentioned in the literature, avatar-based approaches are not the first choice for the majority of deaf users, who prefer human translation. However, when human interpreters are not available, our proposal is presented as a practical and feasible alternative to fill this gap.

Keywords Brazilian sign language · Machine translation · Accessible technologies · Digital television · Deaf people

T. M. U. de Araújo (✉) · F. L. S. Ferreira · D. A. N. dos Santos Silva · F. H. Lemos · G. P. Neto · D. Omaia · G. L. de Souza Filho · T. A. Tavares
Digital Video Applications Lab (Lavid),
Federal University of Paraíba, Paraíba, Brazil
e-mail: maritan@lavid.ufpb.br

F. L. S. Ferreira
e-mail: lacet@lavid.ufpb.br

D. A. N. dos Santos Silva
e-mail: danilo@lavid.ufpb.br

F. H. Lemos
e-mail: felipel@lavid.ufpb.br

G. P. Neto
e-mail: gutenberg@lavid.ufpb.br

D. Omaia
e-mail: derzu@lavid.ufpb.br

G. L. de Souza Filho
e-mail: guido@lavid.ufpb.br

T. A. Tavares
e-mail: tatiana@lavid.ufpb.br

1 Introduction

Information and communication technologies (ICTs) are rarely developed taking into account the specific requirements and needs of deaf people [18]. Their primary means of communication are the sign languages [7], but support for them is rarely addressed in the design of these technologies. In consequence, deaf people have much difficulty communicating and accessing information.

In scientific literature, some works were developed to help deaf people communicate and access information [23,24,39].These works offer technological solutions for daily activities which enable them to watch and understand television, to interact with other people, to write a letter, among others. Examples of these solutions are the use of

emotive captioning in movies and television programs [23] and the development of games for training deaf children [24].

Other works specify formal representations [16], dictionaries [8] and recognition [39] in sign languages (SL). Fusco [16] proposes a formal representation of signs in Brazilian sign language (LIBRAS) based on XML. The author also developed a 3D virtual animated agent to represent these signs. Buttussi et al. [8] proposed a multilingual dictionary for storage, visualization and search of signs in SLs. The authors also defined an authoring tool which allows different communities to extend the dictionary for their own SL. Starner et al. [39] specify a system based on hidden Markov models (HMM) for real-time recognition of sentences in American sign language (ASL) by using a camera to monitor the user's hands. Kaneko et al. [20] defined a text-based computer language (TVML) to generate graphic animations. These animations are created by mapping human movements to a predefined graphic skeleton. Human movements are captured by using an optical sensing technology, and are mapped to a 3D model skeleton to generate graphic content for TV.

There are also other works related to machine translation for sign languages (SLs) [17,31,34,35,42,44]. Veale et al. [42], for example, described a multilingual translation system for translating English texts into Japanese sign language (JSL), ASL and Irish sign language (ISL). The translator is based on a blackboard control architecture with a set of demons that cooperate to generate the translated contents. The work explores and extends some artificial intelligence (AI) concepts to SL [31], such as, knowledge representation, metaphorical reasoning, and blackboard system architecture [32], but there is no testing or experimentation to evaluate the solution. Therefore, it is not possible to draw conclusions about the feasibility of the solution, the quality and speed of translation, among others.

Zhao et al. [44] developed an interlanguage-based approach for translating English text into ASL. Input data are analyzed and an intermediate representation (IR) is generated from their syntactic and morphological information. Then, a sign synthesizer uses the IR information to generate the signs. However, as in Veale et al.'s work [42], no test was conducted to evaluate the solution. Othman and Jemni [32] proposed a strategy for word alignment based on Jaro-distance and included it into a statistical machine translator for English to ASL. However, only the word alignment has been evaluated. No test or experiment was conducted to evaluate the quality and speed of translation, the application domain, among others.

Morrissey [31] proposed an example-based machine translation (EBMT) system for translating text into ISL. To do this task, an EBMT approach needs a bilingual corpora. However, due to the lack of a formally adopted or recognized writing system for SL, it is hard to find corpora in SL. Thus, the authors construct their bilingual corpora by using annotated video data. However, the data set was developed from a set of "children's stories", which restricts the translation for that particular domain.

San-Segundo et al. [17,34,35] proposed an architecture for translating speech into Spanish sign language (LSE) focused on helping deaf people when they want to renew their identity card or driver's license. The idea of the system is to make the dialogue between deaf people and public officials easier in this kind of service. This translation system consists of three modules: a speech recognizer, a natural language translator and an animation module. The speech recognizer is used for decoding the spoken utterance into a word sequence. The natural language translation module converts the word sequence into a sequence of signs in LSE and the animation module plays the sign sequence. However, this solution is also restricted to a particular (or specific) domain (public services) and the time needed for translating speech into LSE (speech recognition, translation and signing) is around 8 s per sentence, which makes the solution unfeasible for real time domains (e.g., television).

Thus, these works have some limitations. Some of them do not have an assessment of the feasibility and quality of the solution [32,42,44], others are only applied to specific domains [17,31,34,35] or are not efficient considering signing and translation speed [17,34,35]. These limitations reduce their application to real-time and open-domain scenarios, such as TV. In addition, there are few papers related to support this topic for Brazilian sign language (LIBRAS) in ICTs [2,38]. These works focus on the synthesis of LIBRAS signs [2] or in the presentation of signs in the SignWriting[1] language [38], but there is no proposal for investigating strategies for machine translation to LIBRAS.

Therefore, the research question of the current work is "how can we address deaf people's communication problems in real-time and open-domain scenarios (e.g., TV), especially when human interpreters are not available?" To answer this question, in this paper, we proposed an architecture for machine translation to Brazilian sign language (LIBRAS) in real-time and open-domain scenarios. This architecture, called LibrasTV, was integrated, implemented and evaluated in a digital TV (DTV) environment (real-time and open-domain) and its components allow the LIBRAS window to be generated and displayed automatically from a closed caption input stream in Brazilian Portuguese (BP).

LibrasTV also uses strategies, such as a low time consuming text-to-gloss machine translation strategy and SL dictionaries to minimize the computational resources needed to generate the sign language window in real-time.

[1] SignWriting is a writing system for sign languages, but little known by the deaf.

The text-to-gloss machine translation strategy combines the use of syntactic transfer rules, defined by human specialists (also used in other works), with new strategies of machine translation for SL, such as the use of (1) a statistical data compression method to classify the input tokens and (2) simplification strategies to reduce the complexity of the input text, before the application of translation rules. In addition, SL (LIBRAS) dictionaries store a visual representation of signs and can be stored in clients or loaded from the network channel, allowing SL regional aspects to be respected. Finally, LibrasTV also allows the LIBRAS window to be enabled, disabled, resized or repositioned in the display, allowing users to configure the display of SL windows according to their preferences.

As also mentioned by Kennaway et al. [21], it is important to point out that LibrasTV does not intend to replace human interpreters, since the quality of machine translation and virtual signing are still not close to the quality of human translation, and signing, especially considering the difficulties of machine translation approaches to explore semantic and contextual ambiguities [12], and the difficulties of virtual signing approaches to express emotions and represent movements in a more natural way [22]. Thus, the idea is to develop a complementary, practical, high speed and low cost solution that can be used, for example, to provide information for the deaf when human interpreters are not available.

In order to validate these aspects, we have implemented a prototype of LibrasTV for the Brazilian digital TV system (SBTVD) and performed a set of preliminary tests. The objective tests evaluated the speed and cost of the solution in terms of speed of translation, signing and bandwidth. The subjective tests involved Brazilian deaf users to evaluate the proposal in practice and its feasibility for them. The description of LibrasTV, the implementation of the prototype for SBTVD and the set of tests with this prototype will be described in the next sections.

This paper is organized as follows. In Sect. 2, we review the main concepts about LIBRAS. In Sect. 3, we describe LibrasTV. In Sect. 4 we integrate the components of LibrasTV in DTV systems. In Sect. 5, we describe an implementation of LibrasTV for the SBTVD. Some tests to evaluate the proposed solution are described in Sect. 6. Final remarks are given in Sect. 7.

2 LIBRAS linguistic issues

Sign languages are visual languages used by deaf people as their primary means of communication [43]. According to Brito [7], they are considered natural languages, because they came from the interactions between deaf people and they can express any descriptive, concrete, rational, literal, metaphorical, emotional or abstract concept.

Like spoken languages, sign languages have their own grammars that consist of several linguistic levels, such as morphology, syntax and semantics [7]. They also have lexical items that are called signs [40]. The main difference is the visual–spatial mode. Another difference is related to the language structure. While spoken languages have a sequential structure, i.e., phonemes are produced sequentially in time, sign languages have a parallel structure and can produce signs using several body parts simultaneously.

The signs have basic units called phonemes. According to Stokoe [40], two different signs differ by at least one phoneme. Examples of phonemes are:

- handshape: finger positions and their movements;
- location: the part of the body where the sign begins;
- hand movements and facial and/or body expressions—non-manual features (NMF);

In Brazil, the sign language used by most Brazilian deaf people that is also recognized by law is Brazilian sign language (LIBRAS). The signs in LIBRAS are composed by five phonemes: handshape, locations, hand movements, direction and NMFs. The possible values for each of these phonemes are discussed in [16]. LIBRAS also has its own vocabulary and grammar rules, different from Brazilian Portuguese (BP). Considering word order (or sign order), for example, Brazilian Portuguese usually structures the sentences in the subject–verb–object (SVO) order, while LIBRAS usually structures them in the topic–comment (TC) order [7]. For example:

- In BP: O urso (S) matou (V) o leão (O). (The bear killed the lion.)
- In LIBRAS: URSO (T), LEÃO MATAR (C). (Bear, lion to kill.)

However, there are also some similarities in the sentence structure [7]. According to Brito [7], in both languages the verb is a core element that has valence and determines the number and type of arguments or additions necessary. For example, the verb "TO send" in BP and the verb "TO SEND" in LIBRAS have the same valence, because they ask three arguments. For example:

- In BP: Paulo enviou o livro ao amigo. (Paul sent the book to a friend.)
- In LIBRAS: LIVRO AMIGO P-A-U-L-O ENVIAR. (Book friend P-A-U-L-O to send.)

As can be seen in these two examples, regardless of the word order, the sentences consist of a core (the verb "to send") and three arguments or complements ("Paulo", "a friend"

and "book"). Another feature that can also be observed is the way proper names are spelled in LIBRAS (for example, the name Paulo is represented in LIBRAS as P-A-U-L-O).

In addition, LIBRAS has some regional differences. The signs could be represented differently according to each region. For example, some signs in LIBRAS are represented differently in the northeast, southeast and south of Brazil.

Currently, the support for LIBRAS in TV is restricted to manual devices, where a window with a LIBRAS interpreter is transmitted over the video program. This solution has high operational costs (cameras, studio, staff, etc.) and needs a full-time interpreter, which restricts its use to a small portion of TV programming. In addition, according to the scientific literature, there is a lack of sign language machine translation approaches developed for real-time and open-domain scenarios, such as TV, as well as a lack of machine translation solutions developed for LIBRAS. These problems motivate the development of LibrasTV solution presented in the next section.

3 LibrasTV architecture

In this section, we describe the LibrasTV architecture. As mentioned in Sect. 1, the proposal of LibrasTV aims to address deaf people's communication problems in real-time and open-domain scenarios, such as DTV, especially when human interpreters are not available. To address this problem, the LibrasTV architecture is composed of a set of components that allow automatic generation of a LIBRAS windows from closed caption input stream in BP.

These components include a low time consuming text-to-gloss machine translation strategy and LIBRAS dictionaries to minimize the computational resources needed to generate the LIBRAS window in real-time. As mentioned above, the text-to-gloss machine translation strategy was planned for open-domain scenarios and it combines morpho-syntactic transfer rules, defined by human specialists, with new strategies, such as a statistical data compression method to classify the input tokens (words) and a simplification strategy to reduce the complexity of the input before the application of these translation rules, increasing the speed of the translation. In addition, LIBRAS dictionaries are also used to avoid the signs rendering in real-time, which is a very time consuming task. LIBRAS dictionaries store visual representations of signs in LIBRAS (pre-rendered) and each sign has a code (e.g., a textual representation) associated with its representation. Thus, it is possible to generate a video of LIBRAS from the combination of LIBRAS dictionary signs. These dictionaries could be stored in clients or loaded from the network channel. This feature allows the preservation of regional aspects of language.

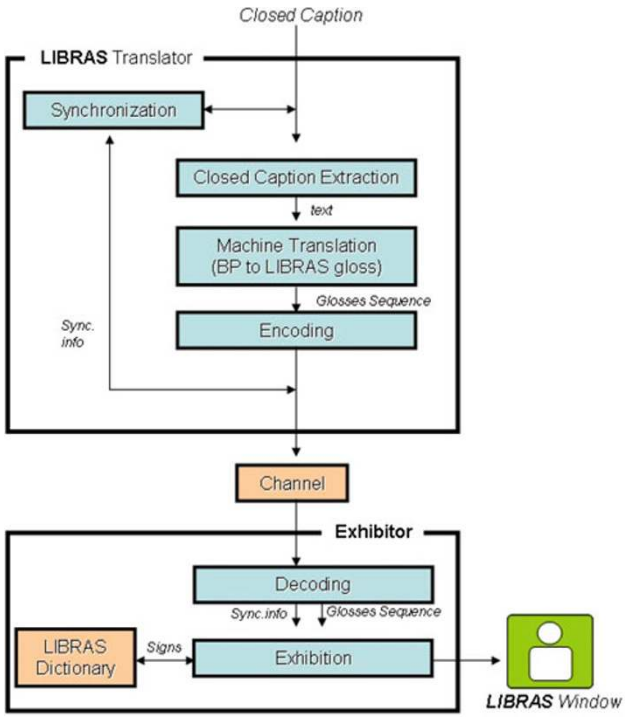

Fig. 1 LibrasTV architecture

Several meetings with deaf and LIBRAS interpreter researchers[2] were performed to discuss the key aspects of the problem, such as issues related to the translation into LIBRAS, the low acceptance of solutions based on virtual signing (i.e., avatars) [10,13,22,35], size, position and synchronization of the LIBRAS window, among others. These meetings are part of the adopted process model and occurred both at the project design and implementation stages. Thus, it was possible to specify and explore their main requirements before and during the proposal development. A summary of these meetings are given in the Appendix.

Figure 1 illustrates the LibrasTV architecture. According to this figure, initially, a LIBRAS translator component receives a closed caption input in BP. A closed caption extraction module is executed to convert the closed caption stream into a sequence of words in text format. Afterwards, the sequence of words is automatically translated (by a machine translation module) into a sequence of glosses (textual representation in LIBRAS) that is encoded (by an encoding module) along with synchronization information and is then transmitted through a communication channel. The stream generated by the encoding module is called the **encoded LIBRAS stream**. More details about these modules are given in Sect. 3.1.

[2] The deaf and interpreter researchers are members of a LIBRAS research group in the education department of the Federal University of Paraiba in Brazil.

Finally, the Exhibitor component receives the **encoded LIBRAS stream** from the channel. The Exhibitor component also decodes, synchronizes and displays the signs to generate the LIBRAS window. This component is composed of two modules: decoding and exhibition. The decoding module extracts the sequence of glosses and synchronization information from the **encoded LIBRAS stream**. The exhibition module associates each gloss with its visual representation stored in the LIBRAS dictionary. Thus, the sequence of glosses is converted to a sequence of visual representations that are synchronized to generate the LIBRAS window.

Synchronization between closed caption input in BP and the LIBRAS window output is performed by using the axis-based synchronization model [5]. This model defines synchronization points that are inserted in the stream using timestamps based on a global timer. In this case, the global timer is the reference clock of the closed caption input stream. This clock is extracted from closed caption and is used to generate the presentation timestamps for the signs in the LIBRAS window.

In the next subsections we will detail the LibrasTV components.

3.1 LIBRAS translator component

The LIBRAS translator is responsible for translating the source input stream (i.e., the closed caption stream) into a textual representation in LIBRAS (sequence of glosses) and for encoding this representation along with synchronization information to be transmitted in a communication channel. According to Fig. 1, it is composed of four basic modules: closed caption extraction, machine translation, synchronization and encoding.

The closed caption extraction modules are used to convert the closed caption input streams into a sequence of words in BP. The synchronization module is used to synchronize the closed caption input and the LIBRAS window output. As mentioned earlier, it extracts the reference clock from the input and uses it to generate the timestamps for the signs presentation. The machine translation and encoding modules will be detailed in the next subsections.

3.1.1 Machine translation module

The machine translation module is used by the LIBRAS translator component to convert a textual representation from BP to a textual representation (sequence of glosses) in LIBRAS. This module is based on the steps illustrated in Fig. 2.

In the first step (i.e., the Tokenizer step), the text in BP is split into a sequence of words (or tokens). Afterwards, the tokens are classified into morphological-syntactic categories (the morphological-syntactic classifier step). To do this task,

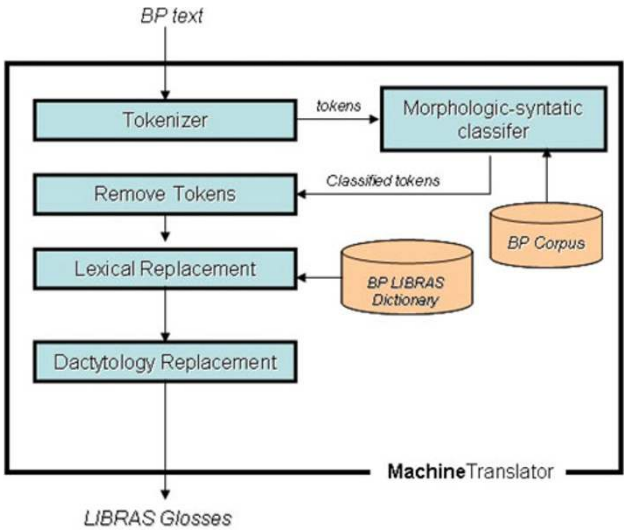

Fig. 2 Machine translation module

we use the PPM-C [29], a variant of prediction by partial matching (PPM) [9]. PPM is a statistical data compression method based on N-order Markov models. It was chosen due to its ability to build accurate statistical models [4] and to its previous use in other classification problems [6,27,28].

PPM builds a statistical model and uses it to store the frequency of different sequences of elements found. After the model is built, the next element of the sequence can be predicted according to its previous N elements. Since higher values for N increase the time needed to compute the algorithm, the order of the algorithm must be chosen so that it will have good results while maintaining acceptable completion times. The PPM-C variant is more efficient than the original implementation in terms of running time and data space exchange for marginally inferior compression.

The morphological-syntactic classifier models morphological-syntactic classes as elements in PPM-C. This model stores sequences of morphological-syntactic classes taken from a corpus of morphological-syntactic classified BP texts. Once a sentence is received for classification, the most likely morphological-syntactic class of each token is selected according to its context in PPM model.

After the tokens are classified, we apply some translation rules (defined by LIBRAS specialists) to translate these tokens (or words) for a representation in gloss notation. Initially, we simplify the text by removing some tokens (the Remove Tokens step). We chose this step because LIBRAS does not define prepositions and articles. Thus, these classes of tokens can be removed. Afterwards, some tokens (or words) are replaced (the Lexical Replacement step) in order to adapt the meaning of the sentence rewritten to LIBRAS, since the LIBRAS vocabulary is smaller than BP's [36]. For example, the words HOME, HOUSE, HABITATION in BP have the same sign (i.e., the same visual representation) in

LIBRAS, the HOME sign. Furthermore, while the BP verbs have a high degree of inflection, the LIBRAS verbs do not inflect. In this case, the BP verbs are replaced by non-inflected gloss verbs (i.e., the LIBRAS verbs). To do this replacement, we use a set of BP to LIBRAS synonyms (BP-LIBRAS dictionary). Finally, proper names and technical terms are spelled in LIBRAS [by handshapes that represent the letters of the token (word)]. Thus, we also apply a dactylology replacement to spell proper names and technical terms. The output generated is a representation in LIBRAS gloss notation.

3.1.2 Encoding module

The encoding module is responsible for encoding the sequence of glosses and the synchronization information generated by machine translation and synchronization modules, respectively. The output of this module, the **encoded LIBRAS stream**, is used by the Exhibitor component to display and synchronize the signs and, therefore, to generate the LIBRAS window. In this subsection, we describe the encoding protocol used to produce this stream.[3]

The encoding protocol has two types of message: the Sign_Control_Message (SCM), a control message, and the Sign_Data_Message (SDM), a data message. The SCM message is used to transmit periodically the initial settings (position, size, resolution) of the LIBRAS window. The SDM is used to transmit the sequence of glosses in LIBRAS. The syntax of SCM and SDM are shown in Tables 1 and 2, respectively.

According to Tables 1 and 2, the SCM and SDM messages begin with their identification and length fields (sign_control_id and sign_control_length for SCM and sign_data_id and sign_data_length for SDM). These fields are used to identify the type of message (SDM or SCM) and the message length in bytes, respectively.

The SCM is also composed of the following fields: resolution, window_line, window_column, window_width, window_height. The resolution field defines the resolution of the graphic layer used to display the window (e.g., $1{,}920 \times 1{,}080$, 720×480, etc.). The possible values of resolution field are shown in Table 3. The window_line and window_column fields define the initial window position coordinates (of top left corner) on graphic layer, while window_width and window_height define the initial window size.

On SDM, the gloss_data_bytes fields transport the glosses (used to reference signs on LIBRAS Dictionary) that are being encoded. Since this field is inside a loop, several signs (glosses) can be transmitted in the same message. The field number_of_signs specifies the number of signs encoded in each SDM.

Table 1 Syntax of SCM message

SCM{	
sign_control_id	8 bits
sign_control_length	16 bits
resolution	8 bits
window_line	16 bits
window_column	16 bits
window_width	16 bits
window_height	16 bits
}	

Table 2 Syntax of SDM message

SDM{	
sign_data_id	8 bits
sign_data_length	16 bits
number_of_signs	16 bits
for (i =0; i < N; i++){	
gloss_bytes_length	8 bits
for (j =0; j < M; j++){	8 bits
gloss_data_bytes	8 bits
}	
}	
}	

Table 3 Values of resolution field

Value	Resolution
0	$1{,}920 \times 1{,}080$
1	$1{,}280 \times 720$
2	640×480
3	960×540
4	720×480
5	320×240
6–255	Reserved for future use

3.2 Exhibitor

The Exhibitor component is responsible for extracting the data (sequence of glosses and synchronization information) from the **encoded LIBRAS stream**. It is also responsible for decoding, synchronizing and displaying the signs synchronously. According to Fig. 1, the Exhibitor is composed of two main modules: decoding and exhibition. The decoding module receives the encoded LIBRAS stream and extracts the sequence of glosses and the synchronization information from this stream. The exhibition module gets the sequence of glosses, associates each gloss (sign) with

[3] This protocol was submitted as a candidate and is being evaluated by the bodies responsible for defining the standards used in the digital television Brazilian system.

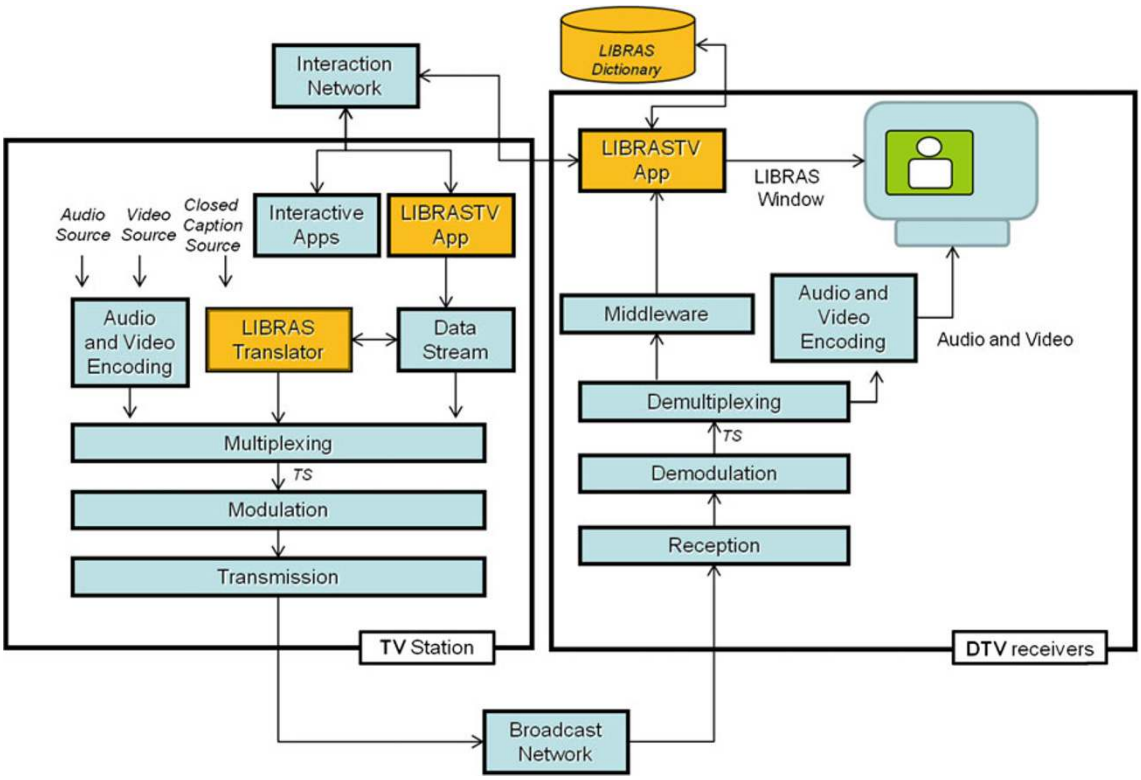

Fig. 3 LibrasTV applied in DTV systems (LibrasTV components are *highlighted*)

its visual representation in the LIBRAS dictionary and displays the visual representation of the signs. To synchronize the signs with the closed caption, the Exhibitor component uses the synchronization information.

Thus, the sequence of glosses is converted into a sequence of visual representations that are synchronized to generate the LIBRAS window. We also define a neutral configuration (position, background color, brightness) to start and finish the representation of each sign. Thus, the exhibition module can smooth the transition between consecutive signs. The exhibition module can also offer additional features, such as to enable, to disable, to resize or to reposition the LIBRAS window. This feature allows users to choose their display settings.

3.2.1 LIBRAS dictionary

The LIBRAS dictionary stores visual representations of the signs in LIBRAS and is used by the exhibition module to decode and display synchronously the sequence of glosses. In this dictionary, each sign can be represented by an animated image or a video file and has a code (i.e., the gloss) associated with its own representation.Therefore, the dictionary can be defined by a set of tuples t in the following format:

$$t = \langle g, v \rangle \quad (1)$$

where g is the gloss representation of sign and v is the visual representation of sign.

As each sign has a fixed code, the visual representation of signs can be customized. For example, the representation could be an animation with a 3D virtual animated agent (3D avatar) or a recorded video with a sign language interpreter. The 3D avatar could also be customized, for example, as a cartoon character for children. Moreover, regional specificities of LIBRAS language are respected as different dictionaries can be used to represent the same sign.

In the next section, we will describe how to integrate the components of LibrasTV into a DTV system.

4 Integration of LibrasTV into DTV systems

A DTV system is basically a client–server system where the server is the TV station (or content provider) environment and the client is the user's environment (see Fig. 3). In the TV station, the analog video and audio sources (captured from a camera or from a video server) are delivered to digital encoders, which are responsible for encoding and compressing the video and audio streams. Then, these compressed video and audio streams are multiplexed together with data streams into a single stream, called a transport stream (MPEG-2 TS—Transport Stream). The MPEG-2 TS

is then modulated and transmitted on a broadcast network (e.g., terrestrial, cable, satellite). On the receiver side, the signal is received, demodulated and delivered to the demultiplexer, which separates the audio, video and data streams. The audio and video streams are sent to the decoders, which decode and synchronize both signals for displaying, while the data streams are sent to be processed by the middleware.[4] The interactive application can also require new data that can be obtained from the interactive (or return) channel.

The integration of LibrasTV into DTV system can be done in several ways. The solution we adopted and recommend is based on the following strategy (see Fig. 3):

- The LIBRAS translator component is integrated with TV stations (or content providers). It receives the closed caption input stream, translates to a sequence of glosses in LIBRAS and encodes them with synchronization information in messages of the encoded LIBRAS stream. These messages will be multiplexed in MPEG-2 TS along with audio, video and data;
- The Exhibitor component is coded as an interactive application, called the LibrasTV application. It will run on DTV receivers. The LibrasTV application extracts the data from the encoded LIBRAS stream, decodes, synchronizes and displays the LIBRAS window with the of the LIBRAS dictionary;
- LIBRAS dictionary may be stored in an extended memory device (e.g., a USB storage device) which will be plugged into a DTV receiver. In this case, we suppose that DTV receiver supports extended memory devices. Alternatively, the LIBRAS dictionary can be loaded from the interactive channel;

The SCM and SDM messages of **encoded LIBRAS stream** (defined in Sect. 3.1.2) are transported in MPEG-2 TS. One interesting alternative is to encapsulate these messages in events defined by Digital Storage Media-Command and Control (DSM-CC) specification [19].

The DSM-CC stream events are transmitted in structures called Stream Event Descriptors allowing synchronization points that are defined at the application level. This structure has a field called eventNPT, which carries a timestamp related to the reference clock of the MPEG-2 TS stream, the Program Clock Reference (PCR). It enables applications to receive events and synchronize their actions with other media, such as video or audio streams. This structure also has a privateDataBytes field to transport private data. Thus, the timestamps for synchronization information and the (SCM and SDM) messages can be encapsulated in this structure (in eventNPT and privateDataBytes fields, respectively).

According to Fig. 3, the LIBRAS translator component (highlighted), which is located at the TV station, receives the closed caption input stream. Then, a process of closed caption extraction is executed, followed by a process of translation of the BP text to a sequence of glosses in LIBRAS. In the next step, the sequence of glosses is encoded in SDM messages and encapsulated in DSM-CC stream events. SCM messages are also generated periodically and encapsulated in DSM-CC stream events. The DSM-CC stream events are packaged, multiplexed in MPEG-2 TS and transmitted in the DTV signal along with the audio and video streams. The TV station also transmits the LibrasTV application (highlighted), which is encoded using DSM-CC Object Carousel standard [19].

At the DTV receiver side, the LibrasTV application receives the DSM-CC stream events, decodes, synchronizes and displays the signs, according to the LIBRAS dictionary (highlighted) and the initial settings defined in SCM messages. This dictionary may be stored in an extended memory device (e.g., a USB device) or loaded by the return channel (interaction network).

This solution consumes low bandwidth of the TV channel, since it only transmits the encoded LIBRAS stream. It can also adapt the presentation of the LIBRAS window according to the regional specificities (i.e., respects the regional differences), since each user can have their own LIBRAS dictionary. In addition, this solution does not require much processing at DTV receiver, since the translation and coding steps are performed at the TV station. It is also possible to integrate LibrasTV into DTV systems as follows:

1. Integrating all components with the TV station—the LIBRAS window would be generated at the TV station and transmitted in MPEG-2 TS as a secondary video.
2. Integrating all components with DTV receivers.

The first solution consumes high bandwidth, since a secondary video is transmitted in MPEG-2 TS, and does not preserve regional differences because the same LIBRAS window will be generated for all users. The second solution also has some limitations, since it requires a lot of processing by DTV receivers to translate, encode, decode, synchronize and display the LIBRAS window. These arguments justify our decision to integrate LibrasTV using the approach shown in Fig. 3 (i.e., with the LIBRAS translator component integrated with TV stations, the Exhibitor component coded as an interactive application and running on DTV receivers and the LIBRAS dictionary stored in an extended memory device).

LibrasTV could also be integrated with other platforms, such as Web and Digital Cinema. On the Web, for example, the LIBRAS translator component could run on a Web server, generating and streaming the encoded LIBRAS stream from

[4] The middleware is a software layer responsible for abstracting the specific characteristics of each receiver, allowing the same interactive application to be executed on receivers from different manufacturers [3].

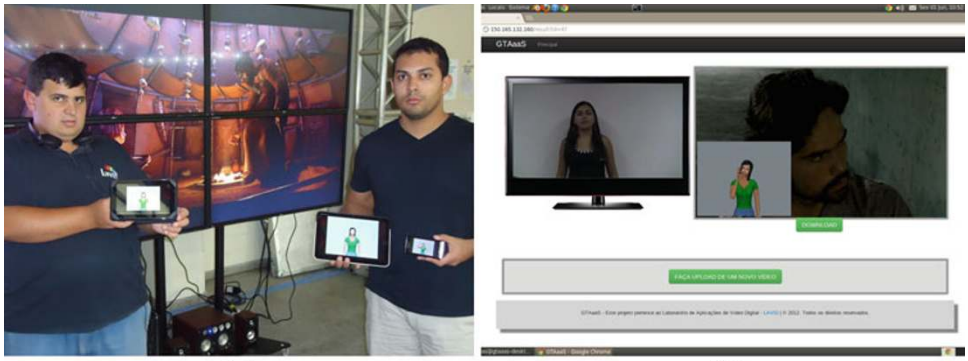

Fig. 4 Demonstration of the adaptation of LibrasTV for **a** Digital Cinema and **b** Web

an input video with closed captioning, and the Exhibitor along with the LIBRAS dictionary could run on the client side, generating and displaying the LIBRAS window in the client Web page. Another alternative would be to run all LibrasTV components on a Web server, which would receive the video input and generate the LIBRAS video to be streamed to the client Web page. It would also be possible to adapt LibrasTV input to raw text, allowing the generation of LIBRAS windows from Web texts. In the Digital Cinema, the LibrasTV input could be adapted to receive subtitles based on the Digital Cinema Package (DCP) format.[5] These subtitles could be translated into a LIBRAS video that could be streamed to mobile devices, such as a smartphone or a tablet, allowing a deaf person to see the LIBRAS translation on these devices. However, the implementation and validation of LibrasTV in these platforms (Web and Digital Cinema) is outside the scope of this work.

Figure 4 illustrates two screenshots of the adaptation of LibrasTV for Web and Digital Cinema, demonstrated in the XXX Brazilian Symposium of Network and Distributed Systems (SBRC 2012) which took place in the city of Ouro Preto.[6] In the Digital Cinema platform, the LIBRAS window was generated from DCP subtitles and transmitted to tablet devices, whereas in the Web platform, the LIBRAS window was generated in a Web server from video closed caption and was presented on the client side. These adaptations, however, are still under development, thus, implementation details and their evaluation are outside the scope of this work.

5 Brazilian digital TV system case study

The case study in this work focuses on the SBTVD. To validate the LibrasTV proposal, we implemented a prototype for SBTVD. The next subsections will detail the developed case study.

5.1 LIBRAS translator

The developed prototype of the LIBRAS translator component was implemented considering the Sect. 3.1 requirements and using C++ programming language.

The closed caption extraction module was developed based on definitions of ABNT NBR 15606-1 [1]. This module receives an MPEG-2 TS streaming and extracts BP sentences and synchronization information (i.e., timestamps) from closed captions packets. These timestamps are inserted into DSM-CC stream events.

The machine translation module receives the BP sentences and translates them into a sequence of glosses in LIBRAS. It was developed according to the class diagram illustrated in Fig. 5. The module main class is the TranslatorController. It has a receiveSentencesToTranslate() method that gets sentences and uses other methods (translate() tokenize(), removeTokens(), replaceDactylology() and replaceLexical()) to translate this sentence into a sequence of glosses in LIBRAS. It also has instances of MorphologicSyntactic Analyzer and RuleAnalyzer classes to classify the tokens and for applying the translation rules, respectively.

The morphological-syntactic classification is done based on a Portuguese language corpus, called "Bosque"[7] [15]. This corpus was developed by Floresta Sintá(c)tica (syntactic forest) project [15] and has 9,368 sentences and 186,000 words. These sentences were obtained from "Folha de São Paulo[8]", a Brazilian newspaper, and also from "Público[9]", a Portuguese newspaper. The entire corpus was morphologically and syntactically classified and fully reviewed by linguists. In our case, we use only the Brazilian Portuguese part of this corpus.

[5] DCP is a collection of digital files used to store and convey Digital Cinema (DC) audio, image, and data streams.

[6] http://sbrc2012.dcc.ufmg.br.

[7] http://www.linguateca.pt/floresta/corpus.html#bosque.

[8] http://www.folha.uol.com.br.

[9] http://www.publico.pt/.

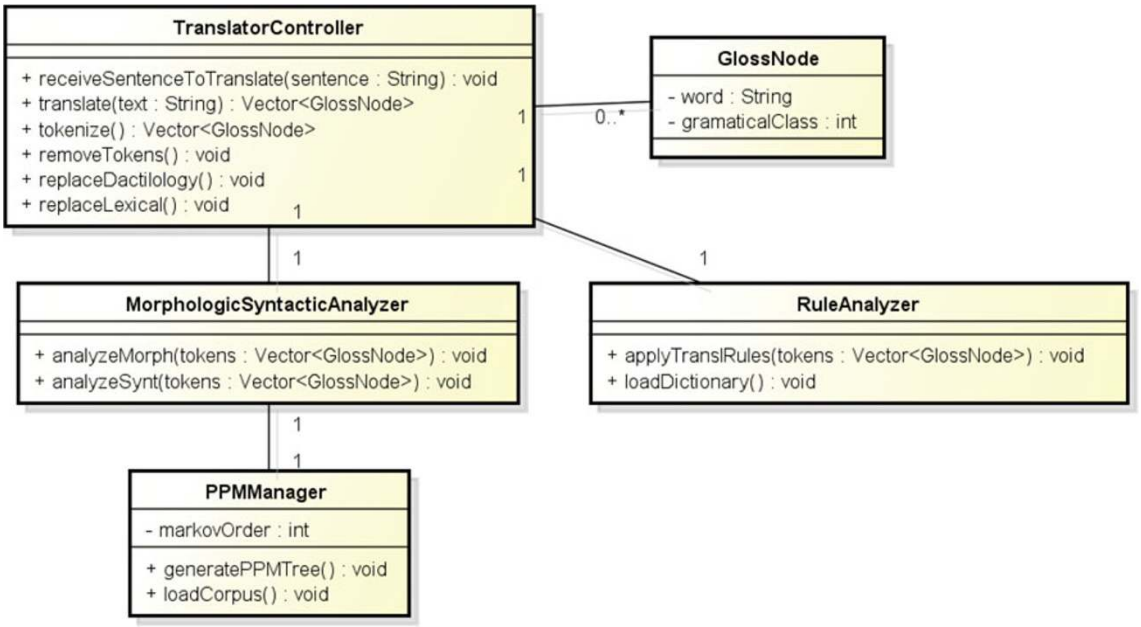

Fig. 5 Class diagram of machine translation module

The MorphologicSyntacticAnalyzer class has an instance of the PPMManager class. This class builds a statistic model based on Bosque's sentences. After that, the PPM-C algorithm [29] is applied to classify the tokens (morphologically and syntactically). The Markov order defined empirically for the PPM model was 5. This value was chosen in order to maintain a good threshold between accuracy and run time.

The RuleAnalyzer class applies the translation rules, developed by a human specialist, to the sequence of tokens and uses a BP to LIBRAS dictionary to do the lexical replacement step. By excluding the rules from the Remove Tokens, Lexical Replacement and Dactylology Replacement steps, nine high-level translation rules developed by the human specialists were taken into account. The translation rules are loaded from a file and are specified using an XML representation that allows LIBRAS specialists to easily add new rules and also modify or remove previously defined ones. Thus, it is possible to extend the set of translation rules in a simple way, just editing that file.

In this XML representation, each rule has a count property which indicates its number of linguistic elements. For each element, there is a title property to identify it morphologically or syntactically and a newpos property to indicate the new positioning of the element after the rule is applied with the value of "−1" meaning that the word must be removed. There is also an optional tag newproperty to indicate property changes in the element (e.g., every verb in LIBRAS must be in the infinitive form, so the newproperty tag is used to specify this). The elements are defined inside the rule in the order that they must appear in the original text for the rule to be applied. Each rule also has an active property and only

```
<rule>
    <active>true</active>
    <count>3</count>
    <class>
        <title>ver</title>
        <newpos>2</newpos>
        <newproperty>inf</newproperty>
    </class>
    <class>
        <title>sub</title>
        <newpos>1</newpos>
    </class>
    <class>
        <title>pre</title>
        <newpos>0</newpos>
    </class>
</rule>
```

Fig. 6 Example of the representation of a translation rule

rules that are active will be applied by the system, making it easier to test different sets of rules without having to necessarily remove a rule.

Figure 6 illustrates an example of a morphological rule representation. This rule indicates that whenever a sequence of a verb followed by a noun and preposition is found, the words in the translated text should be rearranged so that the preposition would come first, followed by the substantive and the verb.

The BP to LIBRAS dictionary was developed in two parts. The first part was extracted from the "LIBRAS Illustrated Dictionary of Sao Paulo" [11], a LIBRAS dictionary which has 43,606 entries, 3,340 images and 3,585 videos, where an interpreter represents the LIBRAS signs. The other one was generated by a human specialist from the verbal inflection variation, where each inflected verb has its translation to its infinitive form. The full dictionary consists of 295,451 entries.

Finally, the list (sequence) of glosses generated by the machine translation module is used by the encoding module to generate the encoded messages, i.e., the SCM and SDM messages, according to module described in Sect. 3.1.2. The SCM are transmitted at periodic intervals of 100 ms. Those messages are then encapsulated in DSM-CC stream events, packaged and sent via User Datagram Protocol (UDP) for multiplexing.

5.2 LibrasTV application

The LibrasTV application was implemented as a Ginga-J[10] application, and OpenGinga[11] was used to run and validate it. The decoding module was developed using a set of "Broadcast streams and file handling" classes, available in the com.sun.broadcast package of Ginga-J. By using these classes, the application can decode DSM-CC stream events and extract the sequence of glosses and the synchronization information from them. The exhibition module was developed using "Java Media Framework (JMF) 1.0", available in javax.media packages of Ginga-J. Similar APIs and packages are also available in others DTV middlewares, such as the Americans Advanced Common Application Platform (ACAP) and OpenCable Application Platform (OCAP) and European Multimedia Home Platform (MHP) [30]. The class diagram of LibrasTV application is illustrated in Fig. 7.

According to Fig. 7, the main class is the LIBRASController. It has instances of LIBRASProcessor and LIBRASPlayer classes which implement the functionalities of the Decoding and Exhibition modules, respectively. It also has instances of other classes of Ginga (e.g., javax.tv.xlet.Xlet, com.sun.dtv.ui. event.UserInputEventListener) to control the user input, to manage the application life cycle, among others.

We had several discussions about the LIBRAS dictionary in the meetings with deaf and LIBRAS interpreter researchers. The main issue discussed was the low acceptance of avatar-based solutions[12] by deaf users, also mentioned in other works [10,13,22,35]. According to these authors, a solution based on videos recorded by LIBRAS interpreters, if technically feasible, would be more appropriate, because it would be probably more natural. Thus, we have initially used a LIBRAS dictionary available on the Internet [25]. In this dictionary, each sign of LIBRAS is a video represented by a human interpreter. However, we observed some problems for generating a LIBRAS window in the preliminary tests. For example, the transitions between two consecutive signs were not smooth. The final position of a sign was usually different from the initial position of the next sign. Other problems were the position and configuration of hands, the distance to the camera and the differences in lighting, which disturbed the clarity of the generated signs.

Another problem is related to the dictionary's update. As LIBRAS is a living language and new signs may arise, it would be always necessary to record new videos for new signs with the same interpreter and under the same conditions of the other signs.

Thus, the deaf and LIBRAS researchers considered that use of avatar is a valid alternative, in the sense that the use of avatars would allow access to a TV program's audible information, especially when a human interpreter is not available. In Sect. 6, we will discuss some results obtained from tests with users to verify this fact.

Therefore, we have modeled and implemented a 3D virtual animated agent (a 3D avatar) to represent the signs of the LIBRAS dictionary. The 3D avatar was developed and modeled using Blender software[13] with an armor composed of 82 bones: 15 bones in each hand to set up handshape, 23 bones to set up facial elements, 22 bones to set up arm and body movements and seven auxiliary bones (i.e., bones that do not deform the mesh directly). Thus, to configure, for example, the movements of the fingers, it is necessary to define the parameters of location and rotation of each of these 15 bones. The same should be done to the bones of the face of the avatar. The arm movement is performed by moving only two bones. The first one is located on the pulse of the avatar and the second one is an auxiliary bone which controls the deformation of the elbow and forearm. We have used inverse kinematics to combine the deformation between related bones. Thus, if there is, for example, a movement in the wrist bone, it will spread to the bones of the arm and forearm. The 3D avatar model is illustrated in Fig. 8. Figure 8b–d illustrates the 3D avatar bones of the face, hand and body, respectively.

The problem with the initial and final positions of signs was solved by designing the 3D avatar animations to start

[10] Ginga-J is the procedural part of the Ginga middleware, the middleware defined by SBTVD. The Ginga-J APIs are based on the Java programming language [37].

[11] OpenGinga is an open-source reference implementation of Ginga middleware available on http://gingacdn.lavid.ufpb.br/projects/openginga.

[12] I.e., solutions that use virtual animated agents to represent the signs.

[13] http://www.blender.org/.

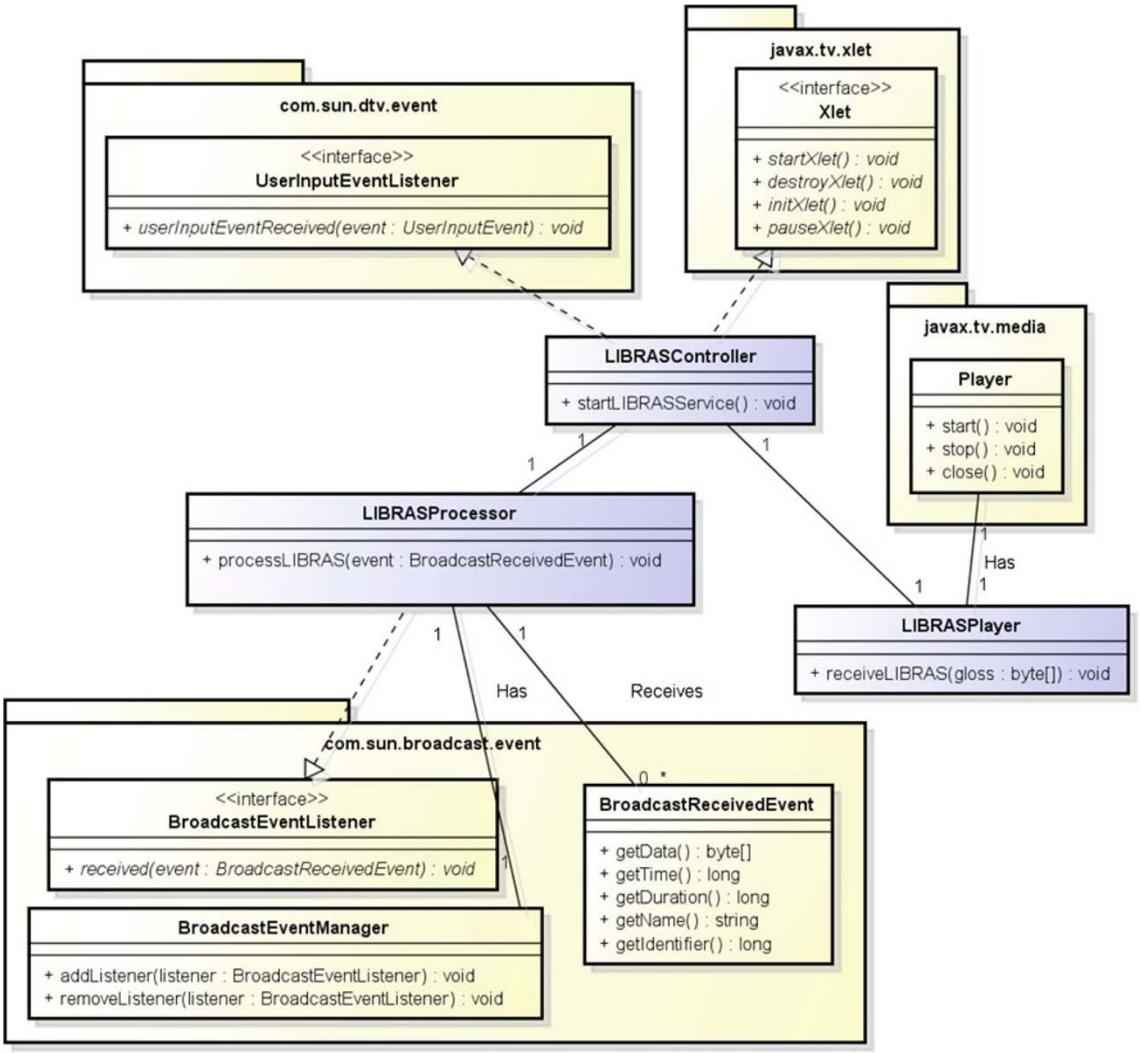

Fig. 7 Class diagram of LibrasTV application

and finish the signs in a neutral position (as mentioned in Sect. 3.2). The neutral position was defined according to the suggestion of LIBRAS interpreters, placing the hands and arms extended in a straight line down and with a neutral facial expression (i.e., without applying movement in the facial bones). Figure 9 shows two screenshots of LibrasTV application execution with the 3D avatar LIBRAS dictionary over OpenGinga.

6 Evaluation

After implementing the prototype, some tests with the prototype were performed to evaluate the proposed solution. These tests include quantitative measures and qualitative evaluation with deaf users. Section 6.1 describes the test environment. Sections 6.2 and 6.3 describe the tests and discuss the results obtained.

6.1 Test environment

To perform the tests with the prototype, we used two minicomputers with an Intel Dual Core T3200 2 GHz processor and 4 GB of RAM. One of these computers was used to run the LIBRAS translator prototype (described in Sect. 5.1) and the other to run the OpenGinga with the LibrasTV application prototype (described in Sect. 5.2). The operating system used in both was the Linux Ubuntu 10.0.4 kernel 2.6.32.

The computer that ran the LIBRAS translator prototype was integrated with some DTV station equipment, including a video streamer, a carousel generator (data stream) and a multiplexer.[14] In this scenario:

[14] The carousel generator and multiplexer equipment used in these tests are compliant with SBTVD and manufactured by Linear Equipamentos Eletronicos (http://www.linear.com.br).

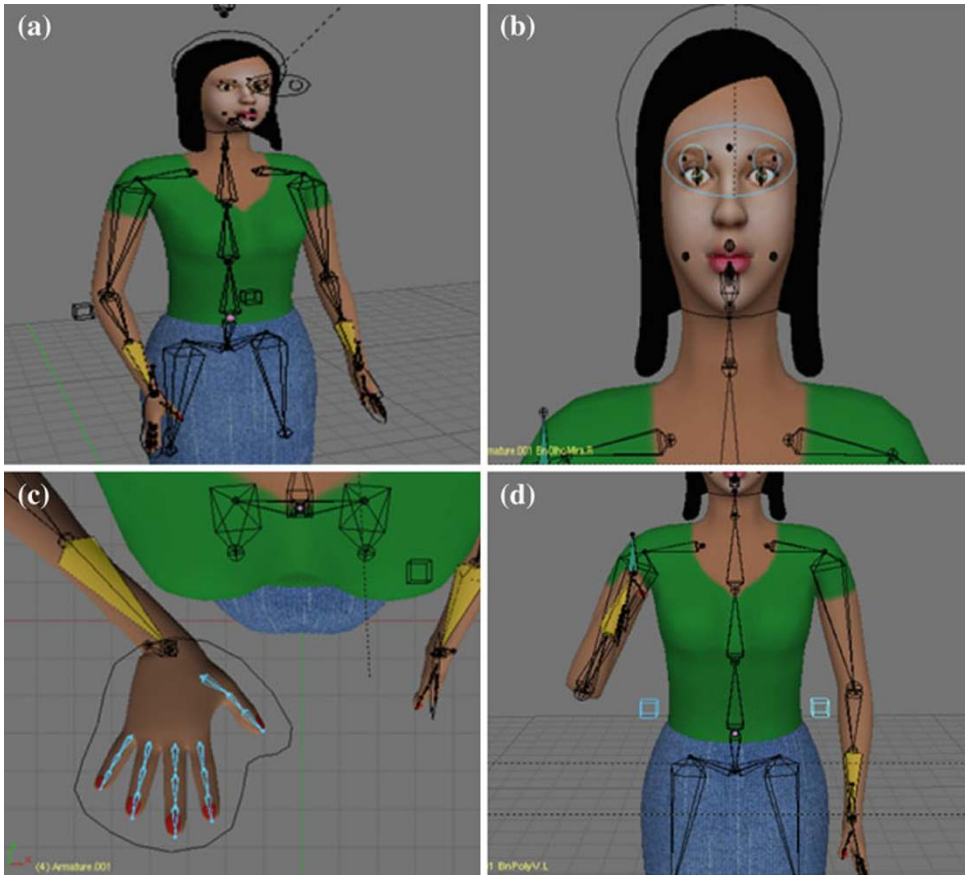

Fig. 8 **a** 3D avatar model. Emphasis on the bones of the **b** face, **c** hand and **d** body

Fig. 9 Screenshots of LibrasTV application execution over OpenGinga

- The carousel generator generates a data stream with the LibrasTV application;
- The video streamer transmits via streaming some test (MPEG-2 TS) videos with BP closed captioning to the LIBRAS translator and multiplexer;
- The LIBRAS translator receives the video stream and generates the encoded LIBRAS stream;
- The multiplexer receives the video, the encoded LIBRAS stream and the data stream. Afterwards, it multiplexes them into a single stream (MPEG-2 TS) and transmits it via streaming to the other computer that runs OpenGinga;
- OpenGinga receives this stream, plays the video, decodes and runs the LibrasTV application that will display the LIBRAS window synchronously.

The videos used in the tests have closed caption and are presented in Table 4.

6.2 Objective measures

In this section, we describe some objective tests performed to evaluate quantitative measures as: the machine translation

Table 4 Videos with closed captioning used in the evaluation

Videos	Duration (s)	Description
Video1	26	This video is a part of a news program presented on 14 October 2008 on TV Globo, a Brazilian TV station
Video2	79	This video is a part of a news program presented on 31 January 2007 on TV Globo, a Brazilian TV station
Video3	65	This video is a part of a movie produced by UFPB TV (the TV of Federal University of Paraiba-UFPB), developed with academic purposes, and consists of a dialogue between two characters

error, the delay of the LibrasTV, the bandwidth used by the **encoded LIBRAS stream**, among others.

6.2.1 Machine translation

Initially, we applied the multiple cross-validation technique to evaluate the performance of the morphological-syntactic classifier. In the cross-validation technique, the data set is partitioned into K equal subsets. Then the model is built (or trained) with all the subsets except the one that is used to compute the validation error. The procedure is repeated K times and each cycle uses a different subset for validation.

We applied this technique by partitioning our data set, the Bosque corpus, into ten equal parts. The procedure was then repeated ten times and, in each execution, nine parts were included in the training set (used to build the PPM-C model) and the remaining part (a different part in each execution) was used to evaluate the performance of the classifier. The percentage of the correct classification for each execution is illustrated in Table 5.

As can be seen, the classifier had an average accuracy of 81.88 % in the classification of the validation sets, i.e, the average rate of misclassification was <20 %. Since prepositions and articles, for example, are later removed by the machine translation module (see Sect. 3.1.1), the prepositions and articles misclassifications probably have no effect on the quality of translation. In practice, this implies that the impact of this misclassification rate may be even lower. A further analysis to evaluate the impact of this misclassification rate in the quality of translation is a proposal of future work.

Table 5 Measures of the percentage of correct morphological-syntactic classifications over the Bosque corpus

Execution	Correct classification (%)
1	82.81
2	83.50
3	82.85
4	83.07
5	81.90
6	79.72
7	81.15
8	81.44
9	81.34
10	81.01
Avg.	81.88

Table 6 BLEU and WER for LibrasTV and a SBP solution [2]

	LibrasTV (%)	SBP solution [2] (%)
BLEU		
1-gram	48.5	40.7
2-gram	30.1	22.2
3-gram	18.9	11.4
4-gram	12.0	5.5
WER	75.3	87.7

Afterwards, we calculated the word error rate (WER) and Bilingual Evaluation Understudy (BLEU) [33] to evaluate the machine translation output. We chose these measures because they were also used in other related works (although in different domains) [34,41]. To carry out this task, initially, we asked two LIBRAS interpreters[15] to translate all sentences of the Bosque corpus into a sequence of glosses in LIBRAS, generating a reference translation for the entire corpus. Then, we translated all the sentences of Bosque using the prototype system and calculated the values of WER and BLEU based on the reference translation. We also calculated the values of BLEU and WER for a Signed Brazilian Portuguese (SBP) solution, i.e., a solution based on direct translation from BP to LIBRAS (without considering grammar differences), such as the solution proposed by Amorim et al. [2]). The idea was to analyze the LibrasTV and SBP results and compare them. Table 6 illustrates the percentage values of BLEU (with different n-gram precisions) and WER for both solutions.

According to Table 6, in these tests, LibrasTV measurements were better than SBP measures for all n-grams precisions. The values of the 4-gram for BLEU was 12 % and

[15] One LIBRAS interpreter was responsible for translating and the other for reviewing.

Table 7 Measure of the average delay of each module of LibrasTV

	Avg. (ms)	SD (ms)	Max. (ms)	Min. (ms)
CC extraction	0.024	0.022	0.554	0.017
Machine translation	0.975	2.957	80.126	0.220
Coding	0.215	0.089	1.061	0.072
Decoding	0.170	0.143	0.519	0.020
Exhibition	42.445	8.747	59.998	20.000
Total	43.805	–	142.21	20.509

for WER was 75.3 %, which helps to evaluate how difficult this task is in an open scenario such as DTV. However, this result is not sufficient to conclude if the proposed translation is good or not. According to Su and Wu [41], objective evaluation based on objective measures is insufficient to evaluate the quality of translation for SLs, since SLs are visualized and gestural languages. Thus, it is also necessary to perform subjective tests with users. In Sect. 6.3, we will describe some tests performed with Brazilian deaf users to evaluate it.

6.2.2 Delay and bandwidth

We also performed some tests to evaluate the delay of each LibrasTV module and the bandwidth used by the encoded LIBRAS stream.

The test to calculate the average delay of LibrasTV modules was performed using a real DTV signal as input during a whole day (24 h). During this time, the MPEG-2 TS of "TV Record" Brazilian DTV channel[16] was tuned in real-time and streamed to the LIBRAS translator and multiplexer.[17] The whole time MPEG-2 TS packets with closed caption data were received by the LIBRAS translator. LIBRAS window was generated by the prototype from these closed caption data and the delay of each LibrasTV module was measured and stored. The average, SD, maximum and minimum values of these measures are shown in Table 7.

According to Table 7, the average delay to run all LibrasTV modules (i.e., the sum of CC Extraction, Machine Translation, Coding, Decoding and Exhibition delays) was <43 ms. The maximum delay obtained (considering the maximum delay of each module) was 142.26 ms, whereas the minimum delay was 20.509 ms. Considering that the test was conducted with an open and representative vocabulary[18] and in a real scenario, the LIBRAS windows can be generated in real-time and with an average time probably much smaller than the time taken for a human translation (although with a lower quality of translation too). Furthermore, this average delay time is smaller than the time used in other related works, such as the solution proposed by San-Segundo et al. [17,34,35] which reported an average delay for translating speech into LSE of around 8 s per sentence.

Finally, we evaluated the bandwidth used by the encoded LIBRAS stream. For Videos1 and Video2, the LibrasTV was run in loop for 4 min and the bandwidth (in Kbps) used by encoded LIBRAS stream was calculated. These values are shown in Fig. 10.

According to Fig. 10, the absolute bit rate used to transmit the encoded LIBRAS stream was always <50 Kbps. The average bit rate was 5.37 Kbps for Video1 and 5.57 Kbps for Video2. As seen in Fig. 10, the bandwidth used by the encoded LIBRAS stream was very low. Thus, it may also be possible to transmit this stream in other network platforms, such as the Web. Furthermore, this bandwidth was significantly lower than the bandwidth used if we choose to transmit LIBRAS window as a video instead of encoded LIBRAS stream.

6.3 Evaluation with users

The subjective measures were collected from questionnaires answered by deaf users. The purpose of these tests was to provide qualitative measures about some aspects of the solution, such as the translation quality, the ease of understanding of the LIBRAS window generated by the solution and its naturalness, among others.

These tests were performed with five Brazilian deaf users,[19] in João Pessoa, a northeastern Brazilian city. The group of users consisted of three men and two women ranging in age from 24 to 36 years (with an average value of 29.2 years). We also observed their education level and their knowledge of LIBRAS and BP. Table 8 shows the profiles of our users considering these aspects.

Users were invited to watch the Video3 (Table 4) with the LIBRAS window automatically generated by the LibrasTV and to complete a questionnaire about some aspects of the solution. To perform the test, we generated a SL Dictionary with 119 signs in LIBRAS.[20]

[16] http://rederecord.r7.com/.

[17] To support the implementation of this test, a "MPEG-2 TS IP retransmitter" equipment was used to tune the DTV channel and stream it MPEG-2 TS to LIBRAS translator and multiplexer

[18] According to a survey conducted by Fundação Getúlio Vargas (FGV) and Brazilian Association of Radio and Television (ABERT) [14], Brazilian DTV channels have, in general, diverse programming, which consists of movies, series and soap operas (35.3 %), news programs (20.3 %), children's programs (14.1 %), variety shows (12.5 %), sports programs (5.0 %), educational programs (2.6 %), comedy shows (2.5 %), religious programs (2.0 %), reality shows (1.4 %), among others.

[19] We tried to find a larger set of users, but, unfortunately, we could not find more volunteers.

[20] We know that one sample video that lasts 65 s is a small sample, but, unfortunately, we do not have enough 3D designers in our laboratory

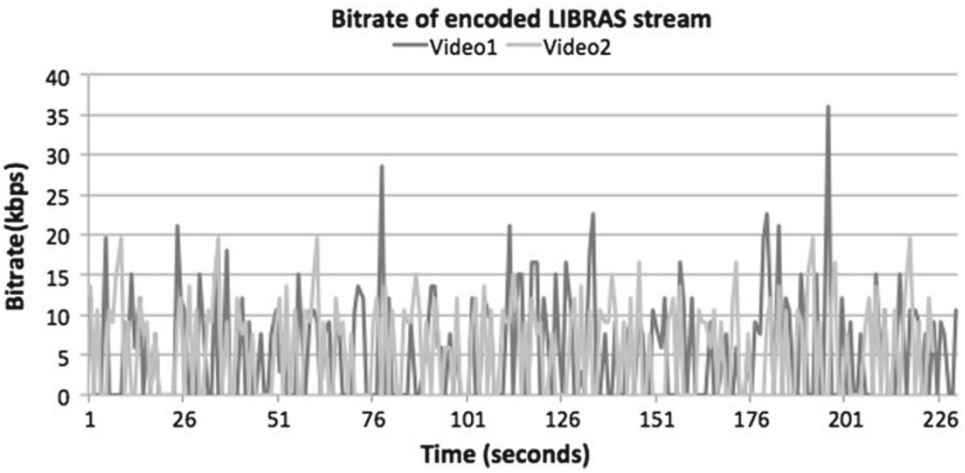

Fig. 10 Bandwidth (in kbps) used by the encoded LIBRAS stream

Table 8 User's profile in terms of level of education and knowledge of LIBRAS and BP

Users	Level of education	Knowledge in LIBRAS (1-to-5)	Knowledge in BP (1-to-5)
User1	Complete elementary school	5	4
User2	Incomplete high school	5	2
User3	Complete high school	5	2
User4	Undergraduate	4	4
User5	Master's degree	5	4

Table 9 Sample of questionnaire

Question	Options
(1) Easy to understand? (5 clear, 1 confusing)	5 4 3 2 1
(2) Good LIBRAS grammar? (5 perfect, 1 bad)	5 4 3 2 1
(3) The signing is natural? (5 move likes a person, 1 like a robot)	5 4 3 2 1
(4) Hand and arm movements are correct? (5 perfect, 1 bad)	5 4 3 2 1
(5) Facial expressions are correct? (5 perfect, 1 bad)	5 4 3 2 1
(6) Which choice on the right matches with the content?	A: jealousy
(What did the couple discuss?)	B: financial problems
	C: problems with children

The applied questionnaire had six questions. The first five questions rated these contents on a 1-to-5 scale[21] for LIBRAS grammatical correctness, clarity, naturalness, quality of presentation, among others. We also used an additional question to check whether users really understood the content that was being transmitted. This question asked the user to select which of three choices (choice A, B or C) was used to match with the content that was being transmitted. Among the three choices presented just one was correct. For example, in the video, a couple has a quarrel due to financial problems. So, the question asked the users why the couple quarrels: (A) jealousy, (B) financial problems, or (C) problems with children. Tables 9 and 10 show a sample of this questionnaire and the average results of the users' evaluation, respectively.

According to Table 10, the clarity had the highest score (3.6). This result is probably compatible with the

Footnote 20 continued
to generate signs to a larger set of samples. We are working on some alternatives, such as the development of a collaborative Web solution, that will allow users to generate signs in a semi-automatic way. It will make the generation of signs faster.

[21] A 1-to-5 scale was chosen because it was also used in the subjective evaluation of other sign language systems [22,41] and it is widely used in other subjective tests, such as tests of audio and video quality.

Table 10 Average scores for the questions

Aspects	Mean score	SD
Clarity	3.6	0.89
Grammatically correct	2.1	1.23
Naturalness	2.8	1.48
Quality of movements	3.4	1.51
Quality of facial expressions	3.6	1.67
Match-success	80 %	–

match-success test, since 80 % of users chose the correct answer to the question. However, this measure would be statistically more significant if more questions (with more contents and more users) were used to assess it. Further analysis with a greater number of questions, contents and users is a proposal of future work.

The quality of movements and facial expressions also had a moderate score (3.4 and 3.6, respectively). However, these measures had the highest SD (1.51 and 1.67, respectively), which shows that the opinions of users in these aspects were more divergent. Grammatical correctness and the naturalness of signs, on the other hand, had the lowest scores (2.1 and 2.8, respectively). As in [35], we observed some probable causes for this outcome during the tests. For example, during the tests, there were discrepancies between users about the structure of the same sentences in LIBRAS. Like other sign languages (e.g., LSE [35])), LIBRAS has an important level of flexibility in the structure of sentences. This flexibility is sometimes not well understood and some of the possibilities were considered as wrong sentences.

Another probable cause observed during the tests was that avatar signing naturalness is not comparable to a human signing. As mentioned in previous works [10,13,22,35], avatar-based approaches are not the first choice for the majority of deaf users, who prefer human translation and signing. One of the reasons for this preference, according to Kipp et al. [22], is the difficulty of virtual signing approaches to represent emotions and movements with less rigidity. Thus, we believe that it is necessary to keep investing more effort to increase flexibility and naturalness of avatar-based solutions.

Finally, there were also discrepancies between users about the correct signing of some signs. For example, users disagreed about the correct signing of the CAFÉ (coffee) and MERCADO (market) signs. One alternative to reduce these discrepancies would be to use custom LIBRAS dictionaries in the users' DTV receivers. However, the development of custom LIBRAS dictionaries is a very time consuming task. Another alternative would be to invest more effort to standardize LIBRAS. In this case, a wider dissemination of LIBRAS in ICTs (e.g., in TV and the Web), would help to standardize it as has also happened in other languages (e.g., in other minority languages in Spain [35]).

7 Conclusions and future works

In this paper, we proposed an architecture for automatic generation of LIBRAS windows and we also implemented and evaluated it for DTV systems. The idea is to improve the access of deaf users to digital contents, especially DTV contents, when a human interpreter is not available or the cost of using human interpreters is not feasible.

Some alternatives to integrate LibrasTV in a DTV were discussed and a case study was developed for the SBTVD. This case study includes the implementation of a prototype of the proposed solution and some tests with deaf users to measure some aspects of the proposal for a real DTV system. This initial evaluation indicates that the proposal is efficient in the sense that its delay and bandwidth are low. In addition, as shown in previous works [10,13,22,35], avatar-based approaches are not the first choice for the majority of deaf users, who prefer human translation. However, when human interpreters are not available, our proposal is presented as a practical, complementary and viable alternative to fill this gap.

Despite the focus of this work being the translation for LIBRAS and its evaluation for DTV, we believe that the proposal would be adaptable to other platforms (e.g., Web, Mobile, Digital Cinema) with minor modifications (although we have not evaluated it yet). For example, we believe that we could adapt the solution for the Web platform just by running all LibrasTV components on a Web server. It is also interesting to investigate the use of audio inputs for machine translation to LIBRAS from BP speech. Thus, a proposal of future work is to adapt the proposed solution for audio inputs and other platforms.

Another proposal for future work is the use motion capture tools (e.g., Microsoft Kinect[22]) to build the signs of the SL dictionary, making it more natural, as well as the development of collaborative strategies to allow deaf and LIBRAS experts to improve the quality of translation and presentation of the LIBRAS window (e.g., by editing or adding translation rules or the translation) over time. The language developed for describing rules (presented in Sect. 5.1) is a first step to perform this task.

Acknowledgments We would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) of Brazil and Linear Equipamentos Eletrônicos for the financial support.

[22] http://www.xbox.com/kinect.

Appendix A: Meetings with deaf and LIBRAS interpreters

Meeting 1 (requirements)

Participants: four deaf researchers, four LIBRAS interpreters and four project team members Location: Digital Video Applications Lab (Lavid) at UFPB

Agenda:

- Direct Translation
- Human videos and avatars

Conclusions:

- Direct translation is not appropriated. LIBRAS has its own grammar.
- Videos with human interpreters are the first choice for deaf users.
- Avatars movements are rigid and inflexible, and have difficulty representing facial expressions that are semantically important. However, they are an alternative when interpreters' videos are not available or are not viable, especially since it would allow improving their access to TV.

Meeting 2 (requirements)

Participants: three deaf researchers, four LIBRAS interpreters and three project team members Location: Digital Video Applications Lab (Lavid) at UFPB
Agenda:

- Synchronization of LIBRAS window
- Size and position of LIBRAS windows
- Regional aspects

Conclusions:

- Fine synchronization is not necessary. Deaf people are already accustomed to small delays that occur in real environments (e.g., in translation of a talk by a human interpreter.)
- Current LIBRAS windows are very small and users cannot resize.
- It would also be interesting if users could reposition the LIBRAS window.

Meeting 3 (design)

Participants: three deaf researchers, five LIBRAS interpreters and four project team members Location: Digital Video Applications Lab (Lavid) at UFPB Agenda:

- LIBRAS dictionary
- Design of avatar

Conclusions:

- Dictionary with human videos are not appropriate due to the difficulty of developing smooth transitions between consecutive signs.
- 2D-avatars are not appropriated. Occlusion in the fingers may cause misinterpretation of the signs.

Meeting 4 (implementation)

Participants: four deaf researchers, two LIBRAS interpreters and three project team members Location: Digital Video Applications Lab (Lavid) at UFPB Agenda:

- Evaluation of 3D avatar
- Definition of a neutral position

Conclusions:

- Placing the hands and arms extended in a straight line down and with a neutral facial expression.
- Neutral position would be to place the hands and arms of the 3D avatar extended in a straight line down and with a neutral facial expression.

Meetings 5, 6 and 7 (implementation)

Participants: three LIBRAS interpreters and two project team members Location: Digital Video Applications Lab (Lavid) at UFPB Agenda:

- Definition of translation rules

Conclusions:

- Definition of a set of initial translation rules.
- One interpreter and one deaf users continue working on the development of translation rules with project team members.

References

1. ABNT NBR 15606-1 Specification (2007) Televisão digital terrestre—codificação de dados e especificações de transmissão para radiodifusão digital—parte 1: codificação de dados (Terrestrial digital television—data coding and specifications of transmission for digital broadcasting. Part 1: data coding)
2. Amorim MLC, Assad R, Lóscio BF, Ferraz FS, Meira S (2010) RybenáTV: solução para acessibilidade de surdos para TV Digital (RybenáTV: solution to accessibility for deaf in digital TV). In: Proceedings of the XVI Brazilian symposium on multimedia and the web, vol 1, pp 243–248

3. Batista CECF, Araujo TMU et al (2007) TVGrid: a grid architecture to use the idle resources on a digital TV network. In: Proceedings of the 7th IEEE international symposium on cluster computing and the grid, vol 1, pp. 823–828
4. Batista LV, Meira MM (2004) Texture classification using the Lempel–Ziv–Welch algorithm. Lect Notes Comput Sci 3171: 444–453
5. Blakowiski G, Steinmetz R (1996) A media synchronization survey: reference model, specification and case studies. IEEE J Sel Areas Commun 14:5–35
6. Bratko A et al (2006) Spam filtering using statistical data compression models. J Mach Learn Res 7:2673–2698
7. Brito LF (1995) Por uma gramática de língua de sinais (For a sign language grammar). Editora Tempo Bras, Rio de Janeiro
8. Buttussi F, Chittaro L, Coppo M (2007) Using Web3D technologies for visualization and search of signs in an international sign language dictionary. In: Proceedings of the 12th international conference on 3D web technology, vol 1, pp 61–70
9. Cleary JG, Witten IH (1984) Data compression using adaptive coding and partial string matching. IEEE Trans Commun 32: 396–402
10. Cox S et al (2002) Tessa, a system to aid communication with deaf people. In: Proceedings of the 5th international ACM conference on assistive technologies, vol 1, pp 205–212
11. Dicionário Ilustrado de Libras do Estado de São Paulo (LIBRAS Illustrated Dictionary of São Paulo) (2006) http://www.acessasp.sp.gov.br/. Accessed 3 Nov 2011
12. Dorr BJ, Jordan PW, Benoit JW (2008) A survey of current paradigms in machine translation. Adv Comput 49:1–68
13. Ferreira FLS, Lemos FH, Araujo TMU (2011) Providing support for sign languages in middlewares compliant with ITU J.202. In: Proceedings of the international symposium on multimedia, vol 1, pp 1–8
14. FGV/ABERT (2012) Pesquisa sobre TV digital no Brasil (Research on Brazilian digital TV). http://www.abert.org.br/site/images/stories/pdf/TV_Programacao.pdf. Accessed 15 May 2012
15. Freitas C, Rocha P, Bick E (2008) Floresta sintá(c)tica: bigger, thicker and easier. In: Texeira A, de Lima VLS, Oliveira LCO, Quaresma P (eds) Conf comput process port lang. Lect notes comput soc. Springer, Aveiro, pp 216–219
16. Fusco E (2004) X-LIBRAS: Um ambiente virtual para a língua Brasileira de sinais (A virtual environment for Brazilian sign language). Dissertation, University of Eurípedes de Marília
17. Gallo B et al (2009) Speech into sign language statistical translation system for deaf people. IEEE Latin Am Trans 7:400–404
18. Haddon L, Paul G (2001) Design in the ICT industry: the role of users. In: Coombs R, Green K, Richards A, Walsh V (eds) Technology and the market: demand, users and innovation. Edward Elgar Publishing, London, pp 201–215
19. ISO/IEC 13818-6 TR (1996) Information technology—generic coding of moving pictures and associated information: part 6: extension for digital storage media command and control
20. Kaneko H, Hamaguchi N, Doke M, Inoue S (2010) Sign language animation using TVML. In: Proceedings of the 9th ACM SIGGRAPH conference on virtual-reality continuum and its applications in industry (VRCAI '10), vol 1, pp 289–292
21. Kennaway JR, Glauert RW, Zwitserlood I (2007) Providing signed content on the Internet by synthesized animation. ACM Trans Comput Hum Interact 14:15–29
22. Kipp M et al (2011) Assessing the deaf user perspective on sign language avatars. In: Proceedings of the 13th international ACM conference on assistive technologies, vol 1, pp 1–8
23. Lee DG, Fels DI, Udo JP (2007) Emotive captioning. Comput Entertain 5:3–15
24. Lee S, Henderson V et al (2005) A gesture based American sign language game for deaf children. In: Proceedings of the conference on human factors in computing systems (CHI 2005), vol 1, pp 1589–1592
25. Libras (2008) Libras—Dicionário da língua brasileira de sinais (Libras—Brazilian Sign Language Dictionary). http://www.acessobrasil.org.br/libras. Accessed 10 Jan 2010
26. Lopez A (2008) Statistical machine translation. ACM Comput Surv 40:1–49
27. Mahoui M, Teahan WJ, Sekhar WJT, Chilukuri S (2008) Identification of gene function using prediction by partial matching (PPM) language models. In: Proceedings of the 17th ACM conference on information and knowledge management, vol 1, pp 779–786
28. Medeiros TFL, Cavalvanti AB et al (2011) Heart arrhythmia classification using the PPM algorithm. Proc Biosignals Biorobotics Conf 1:1–5
29. Moffat A (1990) Implementing the PPM data compression scheme. IEEE Trans Commun 38:1917–1921
30. Morris S, Smith-Chaigneau A (2005) Interactive TV standards: a guide to MHP, OCAP and Java TV. Elsevier/Focal Press, Amsterdam
31. Morrissey S (2008) Data-driven machine translation for sign languages. PhD thesis, Dublin City University
32. Othman A, Jemni M (2011) Statistical sign language machine translation: from English written text to American sign language gloss. Int J Comput Sci Issues 8:65–73
33. Papineni K, Roukos S, Ward T, Zhu W (2001) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics, vol 1, pp 311–318
34. San-Segundo R et al (2008) Speech to sign language translation system for spanish. Speech Commun 50:1009–1020
35. San-Segundo R et al (2011) Development and field evaluation of a Spanish into sign language translation system. Pattern Anal Appl. doi:10.1007/s10044-011-0243-9
36. Santos GS, Silveira MS, Aluisio SM (2009) Produção de textos paralelos em língua portuguesa e uma interlíngua de LIBRAS (Production of parallel texts in Portuguese and a LIBRAS interlanguage). In: Proc XXXVI semin integr softw hardw (SEMISH), vol 1, pp 371–385
37. Sousa Filho GL, Leite LEC, Batista CECF (2007) Ginga-J: the procedural middleware for the Brazilian digital TV system. J Braz Comput Soc 12:47–56
38. Souza VC, Vieira R (2006) Uma proposta para tradução automática entre Libras e português no sign webmessage (An proposal for automatic translation from Libras to portuguese in Sign Webmessage). In: Proceedings of the 19th Brazilian symposium on artificial intelligence, vol 1, pp 1–10
39. Starner T, Pentland A, Weaver J (1998) Real-time American sign language recognition using desk and wearable computer based video. IEEE Trans Pattern Anal Mach Intell 20:1371–1375
40. Stokoe WC (1980) Sign language structure. Annu Rev Anthropol 9:365–390
41. Su HY, Wu CH (2009) Improving structural statistical machine translation for sign language with small corpus using thematic role templates as translation memory. IEEE Trans Audio Speech Lang Process 17:1305–1315
42. Veale T, Conway A, Collins B (1998) The challenges of cross-modal translation: English to sign language translation in the Zardoz system. Mach Transl 13:81–106
43. Wang Q, Chen X, Zhang LG, Wang C, Gao W (2007) Viewpoint invariant sign language recognition. J Comput Vis Image Underst 108:87–97
44. Zhao L et al (2000) Machine translation system from English to American sign language. In: Proceedings of the fourth conference of the association for machine translation, vol 1, pp 293–300