



D5.4: FINAL INTEGRATION ALLOWING POST-EDITING WITH DISCOURSE REPRESENTATION DIAGRAMS

Revision: v.1.0

Work package	WP5
Task	T5.3
Due date	31/12/2023
Submission date	31/01/2024
Deliverable lead	CNRS
Version	1.0
Authors	Michael Filhol, Thomas von Ascheberg, John McDonald (CNRS)
Reviewers	Giacomo Inches (Martel), Claire Bonrepos (INT)

Abstract	This is a cover document for the code delivered as D5.4. It presents a feature recap of the implemented prototype with screenshots, and a link to its source code.
Keywords	Graphical editor, discourse representation diagrams, AZVD, AZee.



Grant Agreement No.: 101016982
Call: H2020-ICT-2020-2
Topic: ICT-57-2020
Type of action: RIA

Document revision history

Version	Date	Description of change	List of contributor(s)
v0.1	18/12/2023	Initial version	Michael Filhol
V1.0	31/01/2024	Submission	

DISCLAIMER

The information, documentation and figures available in this deliverable are written by the "Intelligent Automatic Sign Language Translation" (EASIER) project's consortium under EC grant agreement 101016982 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

COPYRIGHT NOTICE

© 2021 - 2023 EASIER Consortium

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:		OTHER
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to EASIER project and Commission Services	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

1 INTRODUCTION

Task T5.3, of which this deliverable presents the final result, focused first on continuing the specification of AZVD, a system that allows to graphically represent Sign Language content with diagrams. Secondly, it supported development of a software prototype capable of editing such diagrams, and subsequently sending instructions to an avatar to animate them in Sign Language.

This is a cover document presenting deliverable D5.4, which is the source code available on the “EASIER-Paula” branch of the GitHub development repository [1].

Prior deliverable D5.3 contained the first version of the AZVD editor, whose corresponding document explained its major features. A summary is given in section 2 here. That prototype was extended afterwards to integrate animation from the composed diagrams. This constitutes the present deliverable D5.4, whose section 3 explains that new aspect. A last section opens a few prospects for the reported work.

2 AZVD AND THE EDITING SOFTWARE PROTOTYPE

The graphical system AZVD (for “AZee verbalising diagrams”) follows the suggestions published in research initiated just prior to the project [2, 3]. AZVD diagrams are designed to maximise human-readability and script adoptability, by relying on a background study of regularities in spontaneous graphical productions by language users. Figure 1 gives an example of AZVD representing the sentence: “[a/the] French person comes back from Iraq”, with a grammatical use of the right-hand signing space to localise Iraq first, then used to produce the sign meaning “return”.

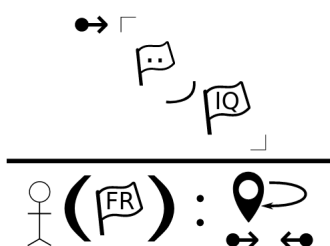


Figure 1: Example of AZVD

AZVD is also designed to determine signed forms entirely, i.e. every composed diagram has a specific signed production that one can read out from it, or indeed that an avatar can render. This is done by specifying an AZee template counterpart for every AZVD graphical symbol or combination, their compositions into complex diagrams making the output AZee expressions grow accordingly.

AZee is a formal representation of Sign Language, whose entries take the form of text resembling computer code. Although interpretable on a semantic level, it is not considered human-readable. Its main purpose is to enable animation by avatar-rendering software. The discourse expression matching the example sentence above is given below.

```

:in-context
  'context
  :about-ref
    'pt
    ^Rssp
    'info
    :instance-of
      'type
      :country
      'elt
      :Iraq
  'process
  :info-about
    'topic
    :side-info
      'focus
      :person
      'info
      :France
    'info
  :return
    'source
    ^Rssp
    'dest
    ^Lssp

```

The point is that AZee is deterministic of a signed output by construction. Any AZee discourse expression can be evaluated to a sign “SCORE” result, i.e. a description of every necessary sign articulation and the way they synchronise on a timeline. Such a SCORE output can finally be given to an animation system, whose task it is to render the content. We come back to this aspect in the next section.

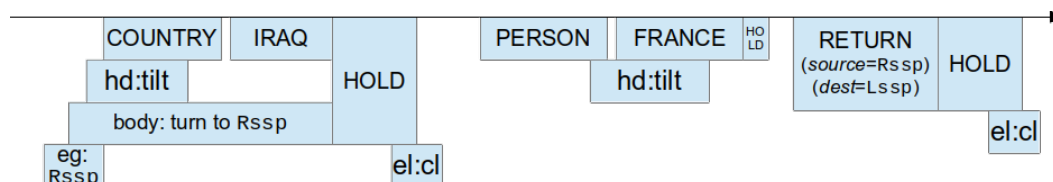


Figure 2: SCORE arrangement of the signed production for the example sentence above

The editor in D5.3 stage (tagged “EASIER-D5.3” in the code repository) displayed three main panels (fig. 3): the main **canvas** (middle) where the graphical editing takes place, the **graphical layout menu** (left), and the **AZee panel** (right). By dragging layouts from the menu and dropping them onto the canvas or inside empty “drop zones”, the user composes diagrams of growing size and complexity. The corresponding AZee code for the diagrams is displayed synchronously in the right-hand panel, updated with every diagram modification in the central canvas.

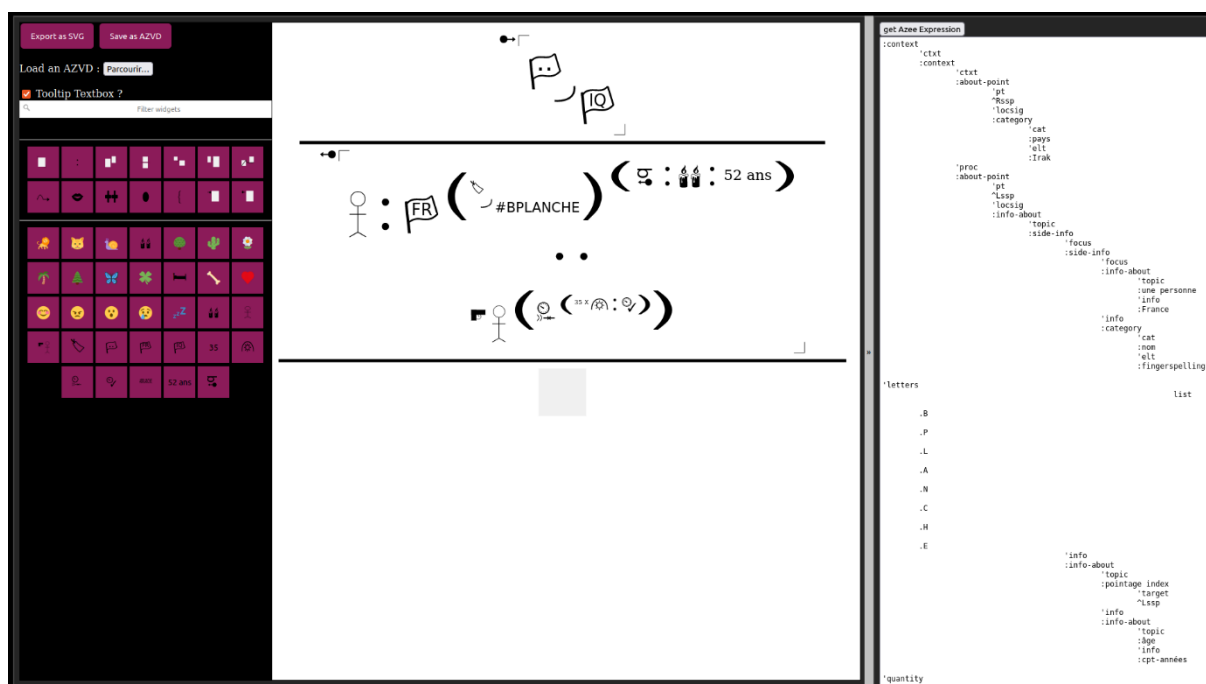


Figure 3: AZVD editor interface before new extension

Note on customising the layout menu entries

The set of layouts populating the left-hand menu shown here is sufficient to demonstrate the capabilities of the system, both for diagram creation and for AZee generation. But for a better coverage and to adjust to different AZee production sets (each set being language-specific), it is necessary to extend it.

The menu is dynamically built at start (loading page) from a set of editable JSON files on the server side, describing the contained layouts and their respective AZee templates. The way to do this is fully described in the “doc/JSONGuide/readme.md” markdown file on the repository.

3 INTEGRATING SIGN SYNTHESIS

In summary, AZVD can be viewed as a graphical abstraction of AZee discourse expressions. For every AZVD composed in the editor canvas, the AZee counterpart is dynamically presented to the user in the right-hand AZee panel. These formal expressions can be rendered by an avatar if evaluated with the AZee interpreter, which yields a SCORE specification of the signed form to synthesise.

The Paula avatar used in the EASIER project has been developing the capability of reading the AZee SCORE format in the recent years, as demonstrated outside of the project [4, 5]. The natural idea to finish T5.3 was therefore to integrate access to this facility in the editor.

This was done by extending the interface to the Paula service, and integrating a new query and display in the AZVD interface.

Paula service extension

Paula already had a querying system through POST requests containing a “style” field that allowed to choose between types of requested animation. One could select a “stock phrase” to render, or query for synthesis of gloss sequences... In this work we have added the “AZee” option for the field. The input AZee expression then fills the “signedPhrase” field in piped style (i.e. expression on a single line, pipe characters substituting new lines). The other fields remain identical.

The query to use for the example sentence is:

```
{
  "style": "AZee",
  "signedLanguage": "fsl",
  "spokenLanguage": "eng",
  "spokenPhrase": "A French person comes back from Iraq.",
  "signedPhrase": ":in-context| 'context| :about-ref| 'pt| ^Rssp| 'info|
:instance-of| 'type| :country| 'elt| :Iraq| 'process| :info-about| 'topic|
:side-info| 'focus| :person| 'info| :France| 'info| :return| 'source| ^Rssp|
'dest| ^Lssp",
  "colorScheme": "4",
  "pad": "string"
}
```

AZVD editor extension

A new panel was added under the AZee panel, fitting a Paula display and a button to send the currently shown AZee expression to the server in the fashion explained above. A screenshot is given in figure 4.

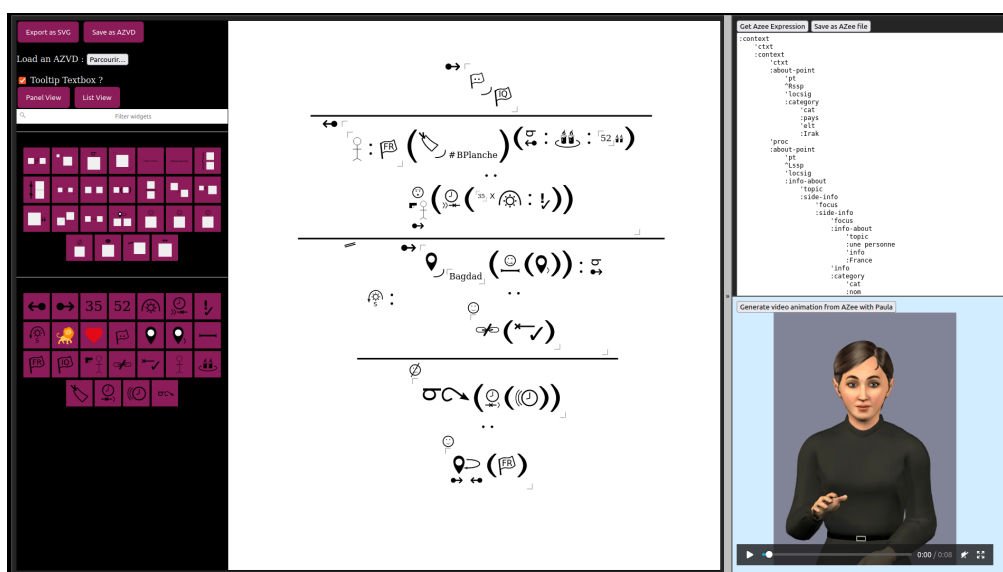


Figure 4: AZVD editor interface before new extension

As per the “README”, provided the required dependencies (Java>8 and Apache Maven) are installed, the editor can be run as a web service (command given) and connected to through <http://localhost:9000> using a regular browser. By clicking on the “get Paula video” button in the new panel—and waiting for the server-side rendering of the resulting video—the client can now display the signed result.

4 CONCLUSION AND PROSPECTS

From AZVD to AZee code, then to SCORE specifications and finally to an avatar able to render them, this work provides a full path from graphical, human-readable diagrams to Sign Language display. There are now however two main steps we wish to take in the immediate wake of this project.

The first one is to further improve the AZVD coverage of AZee. Some features like references to signing space points have been intergrated since last D5.3, but a lot more will be needed to cover things like descriptions of whole scenes, with relative topological positions and shape deployments.

The second is to collect 3D data from the server instead of retrieving a video, which is the main reason for the client’s waiting time with the current implementation. We are developing the possibility of plugging a different avatar in a different branch of the repo, which will allow both types of tests, and give evidence of a certain modularity. In the future, this can allow different avatar systems to be used from the same source of AZVD.

REFERENCES

- 1 <https://github.com/michael-filhol/AZVD> (contact michael.filhol@cncrs.fr for access)
- 2 Michael Filhol (2020). “Elicitation and corpus of spontaneous Sign Language discourse representation diagrams”, in *Proceedings of the 9th workshop on the Representation and Processing of Sign Languages*, Marseille, France.
- 3 Michael Filhol (2020). “A human-editable Sign Language representation inspired by spontaneous productions... and a writing system?”, in *Sign Language Studies*, Gallaudet University Press.
- 4 M. Filhol, J. Mcdonald, R. Wolfe (2017). “Synthesizing Sign Language by connecting linguistically structured descriptions to a multi-track animation system”. *11th International Conference on Universal Access in Human-Computer Interaction (UAHCI) held as Part of HCI International 2017*, Vancouver, Canada.
- 5 M. Filhol, J. Mcdonald (2020). “The Synthesis of Complex Shape Deployments in Sign Language”, *Proceedings of the 9th workshop on the Representation and Processing of Sign Languages*, Marseille, France.

