Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6822–6828
Marseille, 20-25 June 2022

The Multimodal Annotation Software Tool (MAST)

Bruno Cardoso and Neil Cohn
Tilburg University, Tilburg School of Humanities and Digital Sciences
Department of Communication and Cognition
P.O. Box 90153, 5000 LE Tilburg The Netherlands
bruno.cardoso@tilburguniversity.edu, neilcohn@visuallanguagelab.com

Abstract

Multimodal combinations of writing and pictures have become ubiquitous in contemporary society, and scholars have increasingly been turning to analyzing these media. Here we present a resource for annotating these complex documents: the Multimodal Annotation Software Tool (MAST). MAST is an application that allows users to analyze visual and multimodal documents by selecting and annotating visual regions, and to establish relations between annotations that create dependencies and/or constituent structures. By means of schema publications, MAST allows annotation theories to be citable, while evolving and being shared. Documents can be annotated using multiple schemas simultaneously, offering more comprehensive perspectives. As a distributed, client-server system MAST allows for collaborative annotations across teams of users, and features team management and resource access functionalities, facilitating the potential for implementing open science practices. Altogether, we aim for MAST to provide a powerful and innovative annotation tool with application across numerous fields engaging with multimodal media.

Keywords: multimodal annotation, annotation tool, multimodality, visual language, corpus linguistics

1. Introduction

Contemporary society has seen a proliferation of media that combine writing with graphics, whether it is in comics, diagrams, memes, text/emoji, or many other contexts. Complementing this, scholars of language have become increasingly sensitive to analyzing these multimodal documents and creating multimodal corpora.

Various annotation software tools have been created to carry out research on written/graphic multimodal documents, with many tied to the constraints of specific studies. However, no "standard" annotation software has yet emerged for these purposes, and none balance the principles of collaboration and flexibility we view as essential for contemporary research.

Existing visual and multimodal annotation systems provide a range of tools, but are often limited in their scope and applicability. Often, unique annotation software has been designed for specific studies, such as for the study of comics (Dunst, Hartel, & Laubrock, 2017), instruction manuals (van der Sluis, Kloppenburg, & Redeker, 2016), or diagrams (Alikhani & Stone, 2018; Hiippala et al., 2020), among others. These tools are often designed around specific annotation theories, restricting users to work with predefined, built-in sets of annotations, and thus have limited applicability beyond their specific domains.

To our knowledge, only a limited number of general, multipurpose tools have been designed for multimodal or visual annotation. For example, the UAM ImageTool (O'Donnell, 2008), allows for rectangular regions of interest to be drawn and annotated with user-defined annotation schemas. However, this application is limited to single users, requiring the sharing of saved files to facilitate collaboration. Additionally, UAM ImageTool restricts users to a single schema at a time to annotate documents, and does not let users specify relations between annotations for dependencies. Also, it is worth considering that, at the time of writing, ImageTool was last updated in 2010. Another example is the Multimodal Analysis Video & Image Software[1], a software solution for supporting the annotation of multimodal texts with language and image components (O'Halloran, Podlasov, Chua, & K.L.E, 2012). This tool, however, was designed to support single-user work and thus is not designed for remote, collaborative work. Furthermore, the annotations available for users are built-in, developed by the authors, and thus does not let users annotate documents using their own annotations.

While these systems are effective in their intended uses, and many introduce insightful and useful tools, we find them to be limited in various ways. First, in order to offer users the most value, annotation tools should neither restrict researchers in the nature of the documents they can analyze, the annotations they can use, nor the shape of the regions they can define on the documents. In addition, because multimodal documents can be complex and function across many structures, annotation tools should include the ability to annotate the same regions across multiple dimensions. Finally, we believe research tools should leverage collaboration, connections, and sharing between researchers. Other annotation tools use some of these features, but none yet integrate them all into a single platform.

With these needs in mind, here we present a novel software program for analyzing multimodal documents, the Multimodal Annotation Software Tool, or "MAST."[2] MAST was created as part of the TINTIN Project, an ERC grant-funded project aiming to examine the properties of over a thousand comics from around the world. As this project required software to facilitate multimodal annotation on a large scale, and in light of the limitations in existing systems mentioned previously, it seemed optimal to design a system that could be extended beyond the narrow uses of functionality needed for this particular project. MAST allows users to implement any type of annotation schemas they want, to apply schemas to a variety of different documents and media, to facilitate collaboration across users, and to use built-in constraints to

[1] http://multimodal-analysis.com/index.html
[2] https://www.visuallanguagelab.com/mast

allow or disallow the open sharing of documents and data. We hope MAST provides a flexible and powerful tool that can extend across researchers' needs for the annotation of multimodal documents.

Below, we further describe the architecture of MAST and its features.

2. MAST

2.1 Design & Technological Details

Aligned with the discussion so far, we designed MAST with two main goals in mind: (1) to facilitate remote collaboration; and (2) to make the annotation work as flexible as possible.

The value of allowing people to collaborate remotely is widely acknowledged and, we argue, all the more evident in light of the recent global health measures that restrict people to meet in person. We emphasize, however, that annotating a multimodal document is a labor-intensive task, often requiring parsing many layers of information, and requiring complementary approaches to analyze – a task that becomes easier when the expertise of different people is synergistically brought together. To this end, MAST was designed from scratch to foster collaboration between individual researchers and teams, regardless of their geographical location.

In order to make the annotation process flexible and fluid, we argue that annotation tools should address a number of requirements. First, annotation tools should not assume that annotation theories are static and can be built into the software. Rather, theories should be considered an evolving resource, liable to change as the field advances. Besides being truer to the dynamic nature of scientific theories in general, this approach should help stretch the platform's lifecycle and facilitate maintenance. Second, because research often occurs in collaboration between researchers, annotation tools should empower users to share their annotation schemas with other researchers, letting them use, or otherwise contribute to the development of said schemas. Finally, due to the complexity of the information in multimodal documents, annotation platforms should facilitate multi-perspective analyses to be conducted, with different theories complementing one another to allow users a more comprehensive understanding of documents.

MAST addresses these points by modelling annotation schemas as independent resources that users can manage fully, creating, updating and deleting them as needed. Furthermore, schemas can be shared with other users and teams. This allows researchers to implement the schemas of other users, thus increasing the potential reach of one's work, as well as making the best of community contributions. Finally, MAST allows users to mix and combine any number of schemas simultaneously while annotating any given document (see section 2.2.2).

MAST operates with a client-server architecture and is best described as a distributed system: users work with a client application that runs locally on their machines, while information is stored and maintained remotely in a centralized server. The client was implemented in Java and other supporting technologies (e.g., JavaScript), thereby running seamlessly in different platforms like Windows, Linux and MacOS. MAST does not require installing in user machines and has no software dependencies other than a running Java Virtual Machine (Java 11+). In turn, the remote MAST server is a PHP application that offers an interface between the client and a relational database. Both server and database are hosted by Tilburg University.

2.2 MAST Resources

MAST's work model revolves around three simple, high-level, and independent functional concepts: *documents*, *schemas* and *projects*.

2.2.1 Documents

A MAST document is an abstraction for the materials users want to annotate. New documents are created by uploading the corresponding files through the application (MAST is currently restricted to the PDF format, but we aim to offer more options in the future). Pages in the original documents are maintained, thus enabling the analysis of single- and multi-page documents, like pictures or comic books. A typical use-case scenario would be to have a MAST document per item of analysis; however, depending on the circumstances, each page of a document could be alternatively considered as an isolated case, with the whole document representing the corpus in full (such as a corpus of diagrams all compiled into the same document, with one diagram per page).

Although in some limited cases it could make sense to have full-document annotations, the complex information contained in multimodal documents requires a more granular approach. Thereby, MAST allows users to define regions on documents—arbitrarily shaped areas potentially containing interesting information. In our application, annotations are always associated to a region. Permissions can be adjusted for individual documents, granting different levels of access to the uploaded documents. *Private* documents are accessible and can be annotated only by the user that uploads them. *Shared* documents grant exclusive access to users or teams that are specified by the uploader of the document. Finally, *public* documents are accessible by all users of MAST.

2.2.2 Schemas

MAST schemas are conceptually simple, yet powerful resources that allows users to work with a considerable degree of freedom. In simple terms, schemas are dynamic collections of annotation and relation types. They are models of annotation theories and are organized as taxonomies: tree-like structures composed of classes, and annotation and relation types. In MAST, schema classes can contain other classes and they are also considered types.

Schema annotations and relations are templates that specify what resources are available to annotate a document. Schema annotations have a name and a description field that schema creators can use to document the annotation with a static HTML page, including text and images. This description field can aid users with definitions, criteria, and/or diagnostics about annotations and how to implement them in analyses. In turn, relations are associations that express the possible relationships between document annotations and/or relations (actors). A relation may have an arbitrary number of actors of any type (annotations or

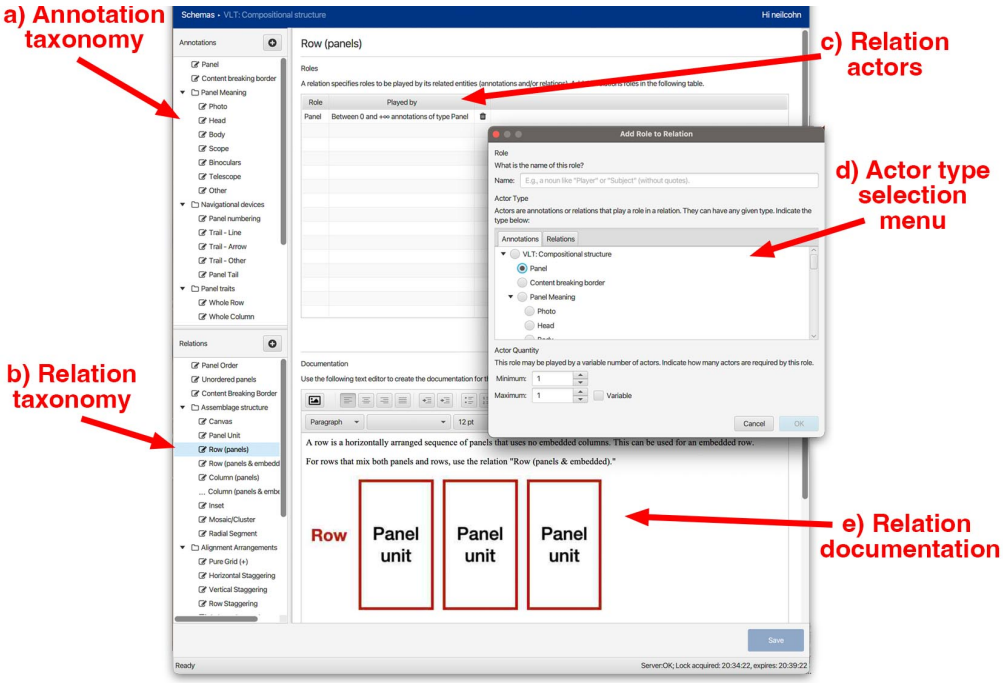

Figure 1: MAST Schema Editor

relations) and require only the creator to determine the number and said type. Because relations accept other relation instances as actors, it is possible to build relations recursively, having relations of a given type T with other instances of T as actors. This allows users to create complex constituencies and dependencies within and across modalities.

As an example, imagine we are interested in annotating the reading sequence of panels in a comic book. After drawing the regions that correspond to each panel, we would need to annotate them and define the order of reading. The simplest approach would be to create a schema with just one annotation, Panel, and one relation, Sequence, requiring a list of actors of type Panel. A sequence of three panels, P_1, P_2 and P_3 could be annotated with a single instance of Sequence, S_1, as $S_1(P_1, P_2, P_3)$. Alternatively, as MAST allows recursive relations, an alternative approach would be to make the Sequence relation recursive with two actors: a Panel annotation and an optional instance of Sequence. We could then annotate arbitrarily long sequences of panels by recursively applying the Sequence relation. To reuse the previous three-panel example, we would need three instances of Sequence, S_1, S_2 and S_3, ending up with $S_3(P_3, S_2(P_2, S_1(P_1)))$ – note that S_1 only has one actor by virtue of the second actor of the relation Sequence being optional[3]. To illustrate the value of relations in a multimodal context, they can be used to link text to the content within an image that it associates to, such as linking the name of a particular mountain in a caption with the picture of that mountain in a range.

MAST schemas may be private or shared with different users and teams, thus making schemas dynamic resources that different people can develop collaboratively. Due to the importance that schemas have (they are models of evolving annotation theories), they cannot be made publicly accessible. While this dynamism may bring advantages, it also carries risks. Imagine a hypothetical scenario in which a user is annotating a document with a given schema. If another user with access to that schema would introduce changes to it, it is easy to see the work of the first user being disrupted (like for example, if the second user deletes an annotation that the first user has been annotating with).

To prevent this situation, annotating documents is not done directly with schemas, but via *schema publications*. A schema publication is a version of a schema, a static snapshot of that schema in a given point in time. Once created, schema publications are given a unique, citable version number and cannot be changed further. If the original schema is updated, new publications must first be created before users can use the updates in their annotation work. Unlike schemas, which can only be made private or shared, schema publications may also be made public, thereby allowing them to be used by all of MAST's registered users in their own annotation work, thereby facilitating annotation standards across research groups that otherwise may not interact.

2.2.3 Projects

Projects are the main unit of work in MAST, bringing together users, documents and schema publications. Projects can be private or shared with other users or teams; for the same security reasons presented for schemas, projects can also not be made public. In order to annotate documents, users must first add them to a project and associate them with the schema publications they want to annotate with. By basing the workflow of MAST on

[3] We mention the recursive example for illustrative purposes only; the non-recursive option is more straightforward and should be preferred as it yields a simpler and easier to analyze database.

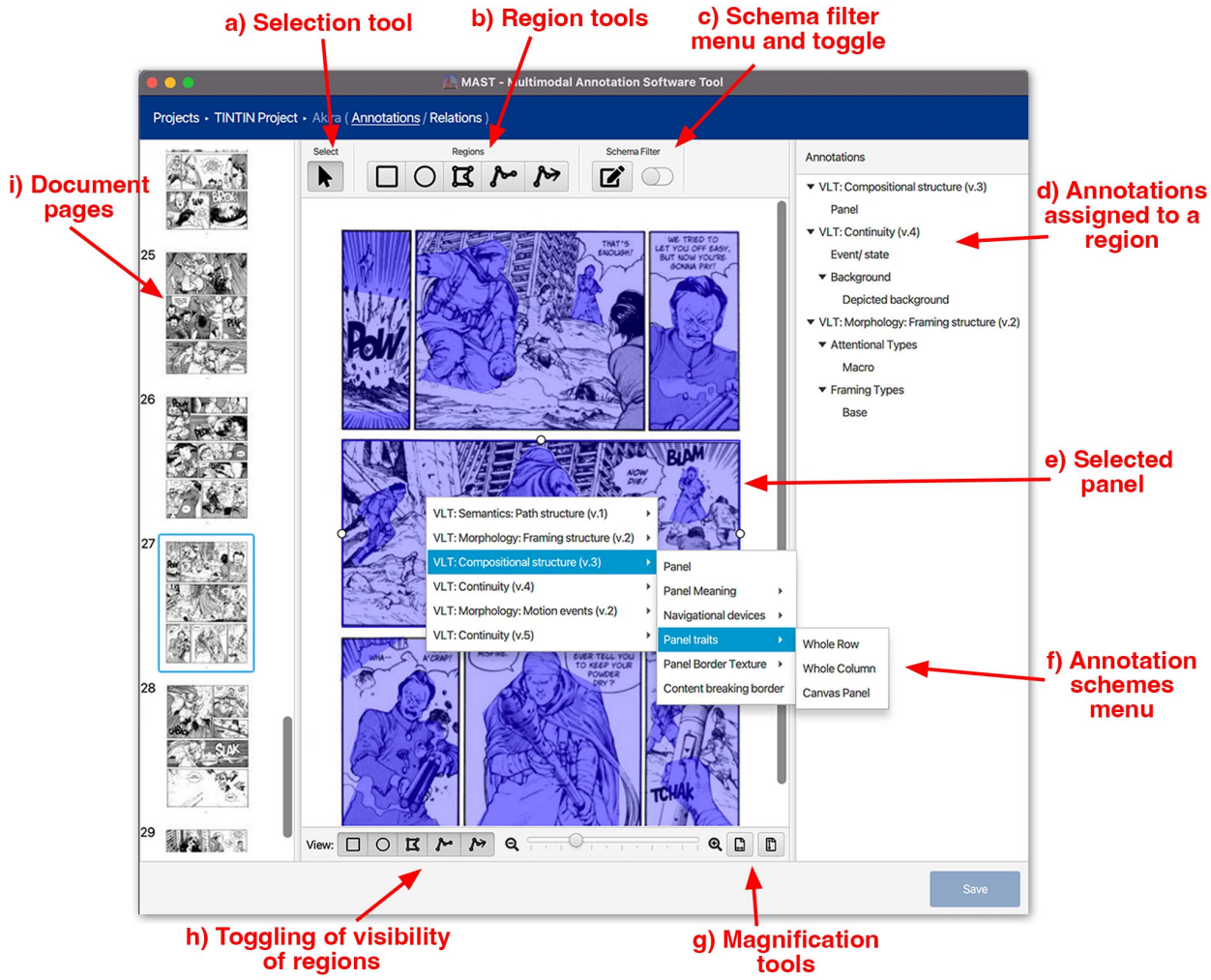

Figure 2: MAST Annotation Editor

projects, users can better organize their research and, because access can be granted to both users and teams at the project level, projects also improve data security and management.

2.3 Annotating Documents

MAST features integrated interfaces (screens) dedicated to working with specific resources, the Schema, Annotation and Relation editors, each offering designs and tools that facilitate working with their namesake resource.

2.3.1 Schema Editor

The Schema Editor, depicted in Figure 1, aims to streamline working with schemas. It allows users to build the schemas' previously mentioned taxonomies of annotations and relations (see section 2.2.2). Although both are organized in a similar tree structure, annotations are effectively different resources than relations. Thereby, the Schema Editor separates the relations and annotations taxonomies in two separate panels (Figure 1a and Figure 1b). By clicking with the secondary mouse button on either of these panels, users will be shown a context menu that will let them create new classes, subclasses and either annotations or relations (depending on the panel clicked). It is also possible to rearrange the tree structure by dragging and dropping the tree elements into their desired position.

The user may edit annotations or relations by selecting the corresponding items on the panel. MAST will then display a dedicated edition interface on the large right panel, containing the controls necessary to edit the selected annotation or relation. Annotations are simple constructs, and thus require only fields for editing their name and documentation (Figure 1e). Relations, on the other hand, are more complex as besides name and documentation, they also have a list of actors that users define with the support of a dedicated pop-up window (Figure 1c and Figure 1d).

2.3.2 Annotation Editor

The Annotation Editor is where users annotate their documents with the annotations of the schema publications currently associated to the document at the project level (see section 2.2.3). It features tools to let users define regions on the document pages and afterwards tag them with annotations. The page of a document under analysis appears in the large center space, while all pages in a document are accessible by clicking on the thumbnails of a pager in a left sidebar (Figure 2i). The perceived sizing of a document page can be toggled using a slider at the bottom of the page, where users can zoom in and out, or can snap to the page height or width (Figure 2g).

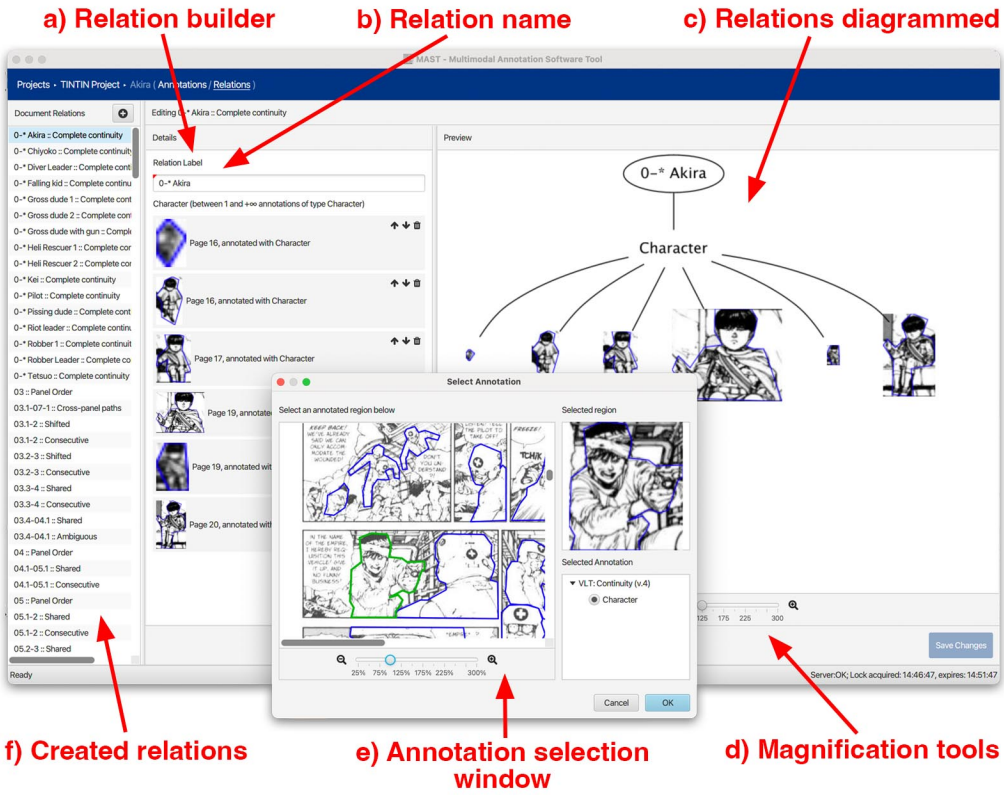

Figure 3: MAST Relations Editor

Analysis begins by users drawing regions around the areas of interest on the document using one of the region tools (Figure 2b). Regions can be defined in a variety of shapes, including rectangles, ellipses, and polygons with drawable shapes. MAST also allows to create paths: ordered sequences of line segments belonging to the same region, which may or not be directional (indicated by an arrow). Because a document may be filled with numerous regions, regions of different shapes can be toggled on or off on a page using an additional menu at the bottom of the interface (Figure 2h). The color of regions can also be toggled to adjust to the properties of the document being annotated. For example, if a page being annotated has a strong blue tint, blue regions might be hard to see, and thus a user can change them to red or yellow for contrast.

The selection tool (Figure 2a) is used to select the regions that have been created. Multiple clicks toggle between regions that might be layered on top of each other. Users can resize regions by dragging the handles that show up after selecting them, or they can move regions by selecting, clicking inside and dragging regions (Figure 2e).

While using the selection tool, annotations are applied by clicking with the secondary mouse button on a previously selected region. This brings up that region's context menu, featuring the available annotations, according to the schema publications currently being used to annotate the document (Figure 2f). The schema tree structure is maintained in this menu. Hovering the cursor over an annotation item in this menu will show a pop-up window displaying the documentation corresponding to the hovered annotation (see section 2.3.1). This pop-up aims to support users in selecting the right annotations, offering contextualized and timely information. The full documentation for the annotations of a given schema may alternatively be browsed in a dedicated window that displays a compilation of the documentations of all of the schema's individual annotations. As MAST does not restrict the number of annotations, classes and subclasses a schema may have, the taxonomy of annotations can grow to generate unwieldy region context menus. Addressing that situation, MAST provides a schema filter (Figure 2c) that lets users select the specific annotations (and classes) they want to use. After applying the filter, MAST will simplify the region context menu by hiding the schemas' unselected annotations.

Once the user selects a region, its associated annotations will appear in the rightmost area of the Annotation Editor (Figure 2d). By clicking on an annotation, a user can access its Notes field. This field can be used to write additional information about a given annotation, which can be useful for recording the justifications or the satisfied criteria or diagnostic-tests for applying an annotation, to write translations of selected text, among other uses.

2.3.3 Relations Editor

The Relations Editor provides an interface for users to create relations, associations between annotations and/or other relations. To instantiate a relation, users are first asked to select the type of the new relation among the relation types in the schemas being used to annotate the document. To streamline this task, MAST displays all available relations in a dedicated screen that also shows the relations' documentation pages (see section 2.2.2)

Once a relation type is selected, an additional workflow opens to allows users to select the annotations or relations that will be the actors in the new relation instance, as depicted in Figure 3a. To support this selection, MAST displays one of two pop-up windows, depending on whether the actor is an annotation or another relation (Figure 3e). If the actor is an annotation, the pop-up displays a black and white version of the full document, where outlines of annotated regions appear (Figure 3e). If a region contains an annotation of the right type, it is presented with a blue outline, indicating readiness for selection. It is noteworthy that relations are not constrained by pages within a document, i.e., they may be defined using annotations from different pages and can thus extend across the whole document. For example, a relation could be made for every instance of a character within a comic, as in the example in Figure 3. Alternatively, in case the relation actor is itself a relation, then the pop-up window will show a list of all previously established relations for that document with the right type, where the appropriate selection can be made.

While users define the details of the new region, MAST shows an automatically updated visual representation of the relation being built as a tree diagram (Figure 3c), complete with the relation name and the actors selected thus far. The same diagram visualization is shown when a user selects an already existing relation.

2.4 Data Export

Although MAST data is stored in a relational database it is converted and exported to CSV format, thus facilitating posterior analyses with external software. Exported data contains information about documents, document regions, annotations, and relations. What may be of particular interest to subsequent studies focusing on visual properties of regions, MAST exports not only region annotations, but also information like region type, spatial coordinates, and absolute and relative areas (in proportion to the area of the region's page) in pixels. Only coordinates are provided for paths, from which angles can be derived.

2.5 Use Case

To illustrate how the concepts presented before fit and work together, we present here a hypothetical but illustrative use case. Suppose we have three MAST users, A, B and C. A and B belong to a research team T, dedicated to the annotation of billboard advertising. A is the leader of team T, and B is a collaborator well-versed in annotation work. B is annotating the billboards using a dedicated MAST schema, S_B. In turn, user C is working on something different: a pioneering theory for annotating paintings, having formalized it as a MAST schema, S_P. After some research, A decides that it might be interesting to apply the theory of C to billboards. This collaborative work could be supported by MAST as follows:

- A creates a team T in MAST, and adds B to it;
- A uploads the billboard documents to MAST;
- A creates a project, adds the uploaded documents to it, associates a publication of S_B to the documents and shares the project with team T;
- B starts annotating the documents with the annotations of S_B;
- In the meantime, A reaches out to C, asking about using the latter's theory to annotate billboards;
- In agreement, C creates a publication of his schema S_P and shares it with team T;
- A associates the shared publication of S_P to the documents in his/her project;
- B proceeds with the annotation work, tagging the regions of the billboard documents with the annotations and relations of both S_B and S_P.

3. Future work

Thus far, MAST provides researchers with a powerful resource. We are planning a user study, however, to help us understand how we can improve further in terms of functionality and usability. Furthermore, we foresee several additional innovations that can expand MAST's potential. First, MAST could add tools that further allow for nuanced analyses of the textual components of the multimodal documents and implement other linguistic annotation formalisms. For example, a Text Annotation Editor could be added for nuanced annotating of all text in a document, and these annotations could be connected via Relations to those in the visual Annotations Editor. This could be aided by OCR technology applied to the text in documents. MAST is also currently set up for static documents, but could be altered to facilitate annotation of video documents, introducing a temporal dimension to its analyses allowing both static and dynamic analyses, beyond what is typically offered in video annotation tools (Gaur, Saxena, & Singh, 2018).

Advances in computer vision could also be implemented within the application of regions, such as automatic selection of comic panels (Nguyen, Rigaud, & Burie, 2019), human figures (Imaizumi, Yamanishi, Nishihara, & Ozawa, 2021; Nguyen, Rigaud, Revel, & Burie, 2021), or human faces (Kumar, Kaur, & Kumar, 2019; Ogawa et al., 2018).

In light of the importance that affective dimensions have in some of the document classes we are targeting (e.g., comics) and the diverse approaches to the topic, we are also considering ways to facilitate annotating this type of information. Tools that leverage discrete models of emotion may offer a simple and systematic approach – after all, emotions in these models are categorical, and thus simple to use as annotations. A promising tool to this end is the CAAT, based on Robert Plutchik's circumplex model of emotions (Cardoso, Santos, & Romão, 2015).

Additionally, to help users query MAST's complex and growing database of annotations, we are considering an adaptation of EveXL (Cardoso & Romão, 2015), a language for the expression of detectable events based on intervals of time. Although the connection between EveXL and MAST may not seem evident at first glance, a sequence of annotations may be considered a stream and queried as such—e.g., the sequence of panels in a comic book may be understood as a stream of panels and associated information. Therefore, a language such as EveXL may indeed prove to be an enhancement to MAST.

Altogether, MAST provides an advanced and flexible annotation system for visual and multimodal documents. It provides users with a resource for visual and multimodal

annotations that can be undertaken in collaborative projects and with managed preferences for facilitating open sharing of documents and data. We plan to make this resource openly available to other researchers in the near future, and to continue developing and improving its functionality to meet the needs of researchers of language, visual, and multimodal communication.

4. Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 850975).

5. Bibliographical References

Alikhani, M., & Stone, M. (2018). *Exploring coherence in visual explanations.* Paper presented at the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).

Cardoso, B., & Romão, T. (2015). *Avoiding"... too late!"-Expressing and Detecting Opportunity with EveWorks and EveXL.* Paper presented at the Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia.

Cardoso, B., Santos, O., & Romão, T. (2015). *On sounder ground: CAAT, a viable widget for affective reaction assessment.* Paper presented at the Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology.

Dunst, A., Hartel, R., & Laubrock, J. (2017, 9-15 Nov. 2017). *The Graphic Narrative Corpus (GNC): Design, Annotation, and Analysis for the Digital Humanities.* Paper presented at the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR).

Gaur, E., Saxena, V., & Singh, S. K. (2018, 12-13 Oct. 2018). *Video annotation tools: A Review.* Paper presented at the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN).

Hiippala, T., Alikhani, M., Haverinen, J., Kalliokoski, T., Logacheva, E., Orekhova, S., . . . Bateman, J. A. (2020). AI2D-RST: a multimodal corpus of 1000 primary school science diagrams. *Language Resources and Evaluation*. doi:10.1007/s10579-020-09517-1

Imaizumi, K., Yamanishi, R., Nishihara, Y., & Ozawa, T. (2021). *Estimating Groups of Featured Characters in Comics with Sequence of Characters' Appearance*. Paper presented at the Proceedings of the 2021 International Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia 2021, Taipei, Taiwan. https://doi.org/10.1145/3463946.3469242

Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques: a review. *Artificial Intelligence Review, 52*(2), 927-948. doi:10.1007/s10462-018-9650-2

Nguyen, N.-V., Rigaud, C., & Burie, J.-C. (2019, 22-25 Sept. 2019). *What do We Expect from Comic Panel Extraction?* Paper presented at the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW).

Nguyen, N.-V., Rigaud, C., Revel, A., & Burie, J.-C. (2021, 2021//). *Manga-MMTL: Multimodal Multitask Transfer Learning for Manga Character Analysis.* Paper presented at the Document Analysis and Recognition – ICDAR 2021, Cham.

O'Donnell, M. (2008). *Demonstration of the UAM CorpusTool for text and image annotation.* Paper presented at the Proceedings of the ACL-08: HLT Demo Session.

O'Halloran, K. L., Podlasov, A., Chua, A., & K.L.E, M. (2012). Interactive software for multimodal analysis. *Visual Communication, 11*(3), 363-381. doi:10.1177/1470357212446414

Ogawa, T., Otsubo, A., Narita, R., Matsui, Y., Yamasaki, T., & Aizawa, K. (2018). Object detection for comics using manga109 annotations. *arXiv preprint arXiv:1803.08670*.

van der Sluis, I., Kloppenburg, L., & Redeker, G. (2016). *PAT Workbench: Annotation and evaluation of text and pictures in multimodal instructions.* Paper presented at the Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH).