Instructions for Contextual Annotation in X-MARKUS

Prof. Dr. Hilde De Weerdt

Dr. Sander Molenaar

March 2024, first draft

Table of Contents

About

X-MARKUS: Contextual Annotation (COMARKUS) is an annotation platform that facilitates the construction of ontological relations between entities. COMARKUS builds on X-MARKUS: Entity Annotation (ENTMARKUS), the annotation tool that allows users to tag and describe entities in texts. In COMARKUS, users establish and describe relations between entities through a schema that structures data annotated in ENTMARKUS. Entities are dragged from the text and dropped into data fields, and then saved as a cluster of related data. These structured data clusters allow users to interpret the context that is lost when entities are tagged in isolation and extracted from the text.

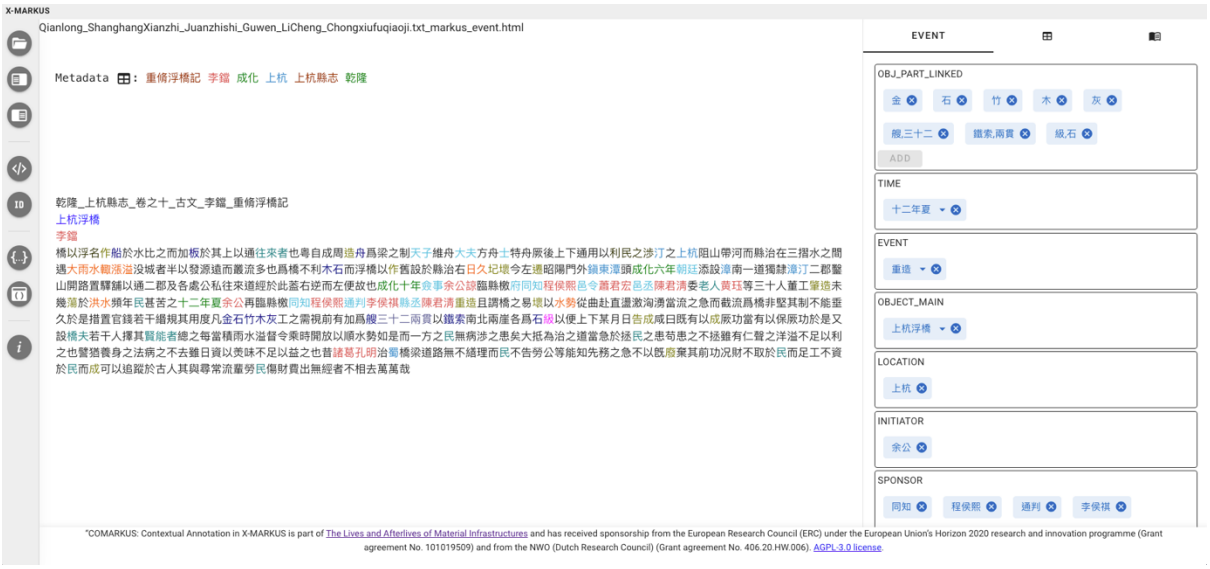

Figure 1 Main view of COMARKUS

COMARKUS opens locally stored MARKUS files in your browser without the need to store your files on a server. The contextual markup is saved locally in json files, a commonly used file format that is compatible with a wide variety of analysis and visualization platforms.

Funding Support

X-MARKUS: Contextual Annotation has been developed by Prof. Dr. Hilde De Weerdt, Dr. Hou leong (Brent) Ho, and Dr. Sander Molenaar as part of The Lives and Afterlives of Imperial Infrastructure in Southeastern China (*InfraLives*) and Regionalizing Infrastructures in Chinese History (*RegInfra*) projects.

This research is part of a project that has received funding from the NWO (Dutch Research Council) (Grant agreement No. 406.20.HW.006).

This research is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019509).

How to Cite Us

Platform:

Hilde De Weerdt, Hou Ieong (Brent) Ho, Sander Molenaar. COMARKUS: Contextual Annotation in X-MARKUS. 2024. comarkus.xmarkus.org

Code:

Hou Ieong (Brent) Ho. COMARKUS: Contextual Annotation in X-MARKUS. 2024. [TBA]

Instructions (pdf):

Hilde De Weerdt and Sander Molenaar. Instructions for Contextual Annotation in X-MARKUS. 2024. comarkus.xmarkus.org

Overview

COMARKUS allows users to work with complex ontologies and design data structures for the annotation of text corpora. It has also been designed to allow for editorial freedom during contextual annotation, including functionality for the revision of entity annotations done in ENTMARKUS.

Schema Editor

COMARKUS was originally designed for the annotation of infrastructural events (the construction, renovation, failure, and destruction of city walls, bridges, and roads) and therefore comes with a default schema that describes an ontology of infrastructural events and that regulates how entities populate event fields.

This schema serves as a model only and can be fully modified to fit the research design of your project. In the schema editor (see Figure 2), data fields can be added, duplicated, renamed, and removed; the entities that can populate those data fields can be defined and adjusted; types can be added for the classification of data fields; and, hierarchies can be introduced via embedded schemas.

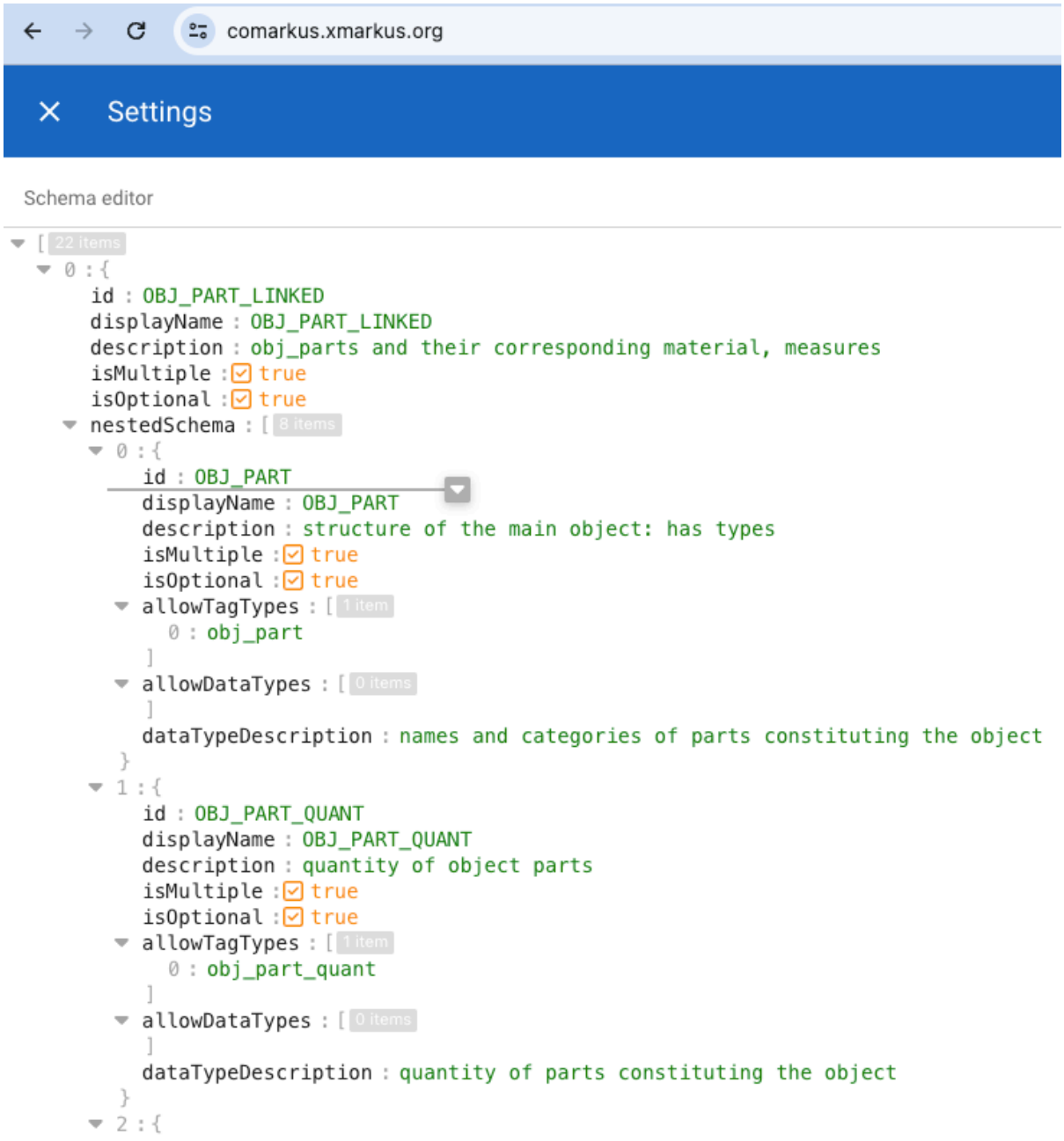

Figure 2 COMARKUS schema editor (cropped)

Jump to the section Schema Editor for further explanation.

Tag-editing

COMARKUS works with files that have already gone through entity markup. It was specifically designed to work with files annotated in ENTMARKUS. However, COMARKUS offers editorial functionality in the main viewer (see Figure 3). Tags can be added, and tag IDs can be edited and removed.

Figure 3 Tag-editing in the main viewer (cropped)

Jump to the section Tag-editing for further explanation.

Getting Started

1. Create or locate a folder with MARKUS files on your computer. Please consult ENTMARKUS for more information on how to export MARKUS files: click the camera icon in the top bar for video instructions on the export of MARKUS files.

NOTE: Your MARKUS files will be directly edited in COMARKUS. We strongly recommend that you work with copies of your files and save the original files in an archival folder to avoid the loss of originals.

2. Go to COMARKUS and open your local folder by clicking on the folder icon at the top of the left menu bar on the screen.
3. This opens a pop-up window where you can select your folder with MARKUS files. Give COMARKUS permission to edit the files in the folder (see Figure 4).

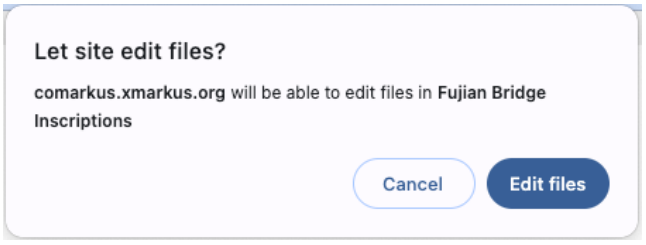

Figure 4 Prompt for permission to edit local files (cropped)

The Interface

COMARKUS consists of a main viewer, which displays one file at the time, a left side menu bar with various settings, and a panel on the right with different data views:

Figure 5 Left menu bar (blue box), main viewer (green box), and right panel (red box)

Left Menu Bar

The left menu bar contains eight icons (see Figure 6), from top to bottom:

Figure 6 Overview of left menu bar (cropped)

1. Open New Folder

Clicking this icon prompts a pop-up window where you can locate your folder with MARKUS files.

2. Pop out or Hide File List

Clicking this icon pops out or hides an overview of all the MARKUS files in the folder.

3. Pop out or Hide Data Fields

Clicking this icon pops out or hides the data fields on the right that display the schema, metadata editor, and dictionaries.

4. Show Tag

Clicking this icon shows or hides tags in the main view.

5. Show ID

Clicking this icon shows or hides IDs in the main view.

6. Open Schema Editor

Clicking this icon opens the Schema Editor.

7. Open Events/Data Clusters

Clicking this icon opens an overview of all saved events or data clusters saved in COMARKUS.

8. Info and Links

Clicking this icon opens an overview of citation information and links to other X-MARKUS services.

Right Panel

The right panel contains three tabs (see Figure 7), from left to right:

1. Event/ Data cluster fields

The first tab (see Figure 7) features all the data fields described in the schema. Required fields are visible at all times, while optional fields become visible when populated with relevant entities. The section Contextual Annotation explains in more detail how entities populate data fields.

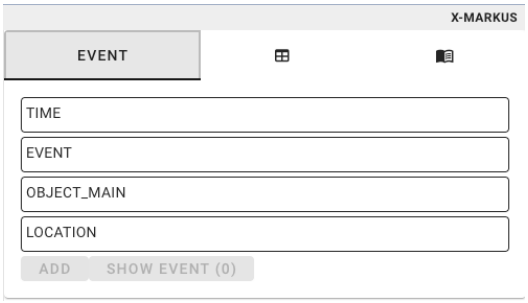

Figure 7 Right panel - Event fields (cropped)

2. Metadata editor

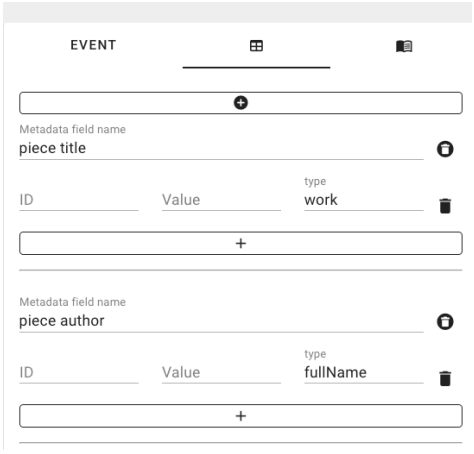

Figure 8 Right panel - Metadata editor (cropped)

The second tab (see Figure 8) contains fields for metadata associated with the file in the main viewer. You can add the following information for the text in the file: piece title, piece author, piece time, and place covered; as well as the following information for the source from which the text is derived: source title, source author, publication place, and publication time. The metadata schema can be modified. Jump to Metadata Editor for more information on modification of the metadata schema.

3. Dictionaries

The third tab (see Figure 9) features a search engine that is linked to external authorities such as the China Biographical Database (CBDB), the Temporal Gazetteer (TGAZ), and the Dharma Drum Institute of Liberal Arts Glossary (DILA Glossary), allowing the user to look up IDs and other information in external authority databases and incorporate them in contextual annotation.

Figure 9 Right side menu – Dictionaries (cropped)

Schema Editor

COMARKUS has a built-in default schema that can be modified to fit the research design of your project. Each item in the schema appears on the right panel as a data field that can be populated by entities from the text in the main viewer.

1. Click the sixth icon in the left menu bar  to open the schema editor.


```
5 : {
    id : OBJECT_MAIN
    displayName : OBJECT_MAIN
    description : main object of the action: has types
    isMultiple : false
    isOptional : false
  allowTagTypes : [ 1 item
      0 : object
    ]
  allowDataTypes : [ 3 items
      0 : bridge
      1 : wall
      2 : road
    ]
    dataTypeDescription : bridge, wall, road name; text
  }
6 : {
    id : OBJECT_ALT_NAME
    displayName : OBJECT_ALT_NAME
    description : alternative names of main object: has types
    isMultiple : true
    isOptional : true
  allowTagTypes : [ 1 item
      0 : object
    ]
  allowDataTypes : [ 0 items
    ]
    dataTypeDescription : bridge, wall, road name; text
  }
7 : {
    id : OBJECT_LENGTH
    displayName : OBJECT_LENGTH
    description : length of main object
    isMultiple : true
    isOptional : true
  allowTagTypes : [ 2 items
      0 : obj_length
      1 : obj_part_length
    ]
  allowDataTypes : [ 0 items
    ]
    dataTypeDescription : number + measure
  }
```


Figure 10 Schema editor (cropped)

Each item in the schema editor corresponds with a data field in the right panel of the main viewer. Items can be deleted, duplicated, and modified to create a fitting data structure for any project ontology. Each item has 8 lines (see Figure 10).

2. Provide an "id" and "displayName" for each item in your schema. These should correspond with the position of this item in your project ontology.

For example, in Figure 10, OBJECT_MAIN refers to the main object in this data cluster, while OBJECT_LENGTH refers to a physical property of that object, namely its length.

3. Add a "description" to provide an explanation of this data field.
4. Check "isMultiple" (true) if multiple entities can populate the data field; and uncheck (false) if the data field can only take one entity per data cluster.

For example, in this schema there can only be one main object, so OBJECT_MAIN "isMultiple" (false), while there can be many alternative names for an object, so OBJECT_ALT_NAME "isMultiple" (true).

5. Check "isOptional" (true) if the data field is not required, but optional; and uncheck (false) if the data field is required before a data cluster can be created.

For example, this schema requires a main object, so OBJECT_MAIN "isOptional" (false), but object length is not always mentioned, so OBJECT_LENGTH "isOptional" (true).

6. The line "allowTagTypes" has one or more sub-lines where tag types can be identified. Only the tag types specified here will be accepted by the corresponding data field in the right panel of the main viewer.

For example, entities tagged as object can populate the data field of OBJECT_MAIN or OBJECT_ALT_NAME, while both the entities tagged as obj_length and obj_part_length can populate the data field for OBJECT_LENGTH.

NOTE: Please be careful with spelling, tag types are case sensitive.

7. The line "allowDataTypes" enables a drop-down menu for additional classification in the data field (see Figure 11). This option can be enabled in any schema item for which a subdivision in types would add analytical value.

For example, the schema item "OBJECT_MAIN" has three sub-items under "allowDataTypes", namely "bridge", "wall", and "road". These can be used to add a type to an entity in the data field, which allows for joint, separate, and comparative analysis of bridges, walls, and roads (see Figure 11*).*

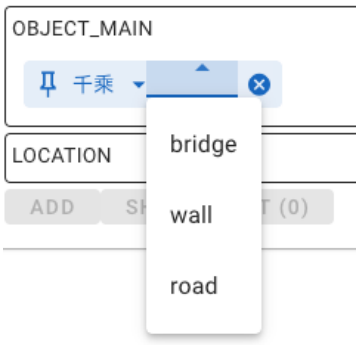

Figure 11 Drop-down list of types in right panel (cropped)

8. Write a "dataTypeDescription" to provide an explanation for "allowDataTypes".

Nested Schema

The nested schema groups entities within the larger data cluster (see Figure 12).

For example, in the default schema, an object can have physical properties such as its dimensions or the material it is made of but can also have associated object parts such as gates or towers in the case of walls. These gates and towers in turn can have physical properties. To avoid confusion, the physical properties of object parts can first be linked to an object part before the object part and its physical properties are added to the data cluster.

1. The nested schema item has an “id”, “displayName”, “description”, “isMultiple”, and “isOptional”, like regular schema items, but instead of “allowTagTypes” there are subordinate schema items, which work in the same manner as regular schema items.

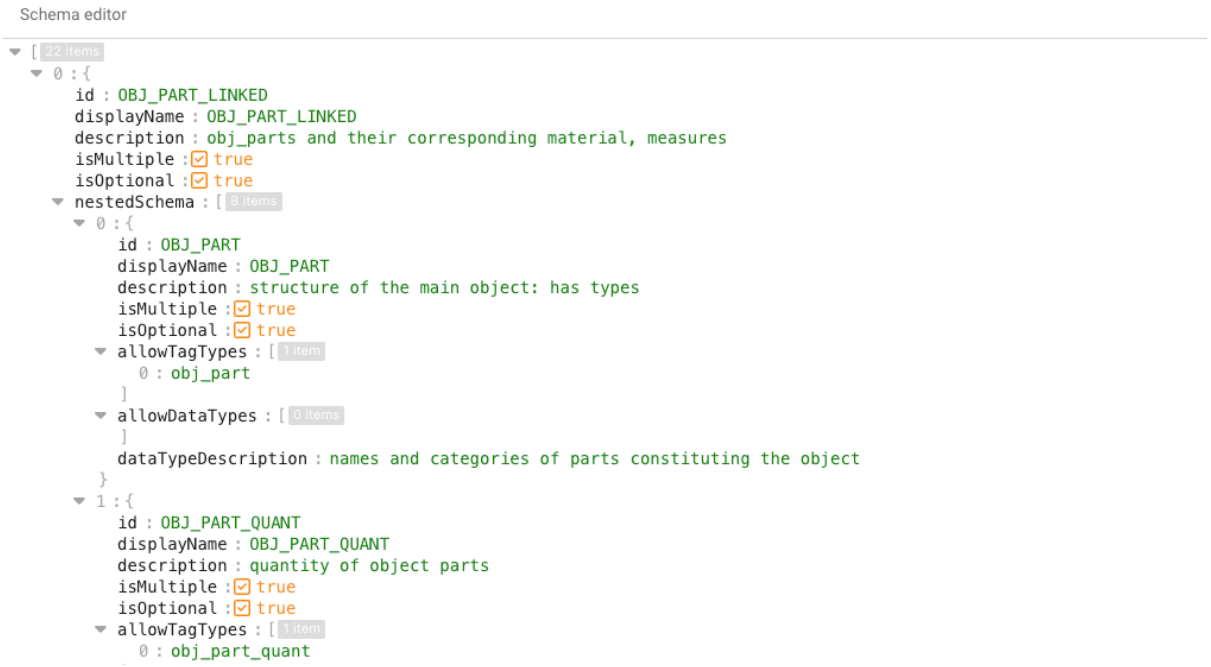

Figure 12 Nested schema in the schema editor (cropped)

Contextual Annotation

Contextual annotation takes place in the main viewer (see Figure 13).

1. Click “Open Folder” (first icon in the left menu bar), then give COMARKUS permission to edit files.
2. A list of the files in your folder will appear on the left side. This list can be hidden or shown with the second icon in the left menu bar. Select a file from this list.
3. Tagged entities in the text can be recognized by the colours they were assigned in ENTMARKUS. Alternatively, the fourth and fifth icons in the left menu bar allow you to show/hide the tag names and IDs for ease of reading.

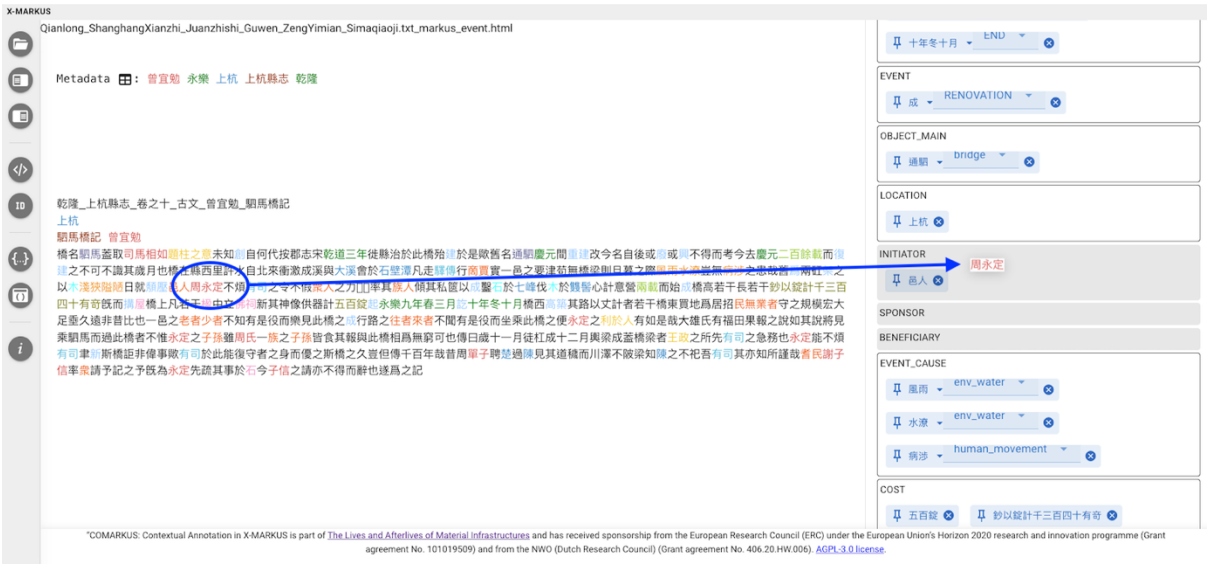

Figure 13 Main viewer with data fields in the right panel

4. Click a tagged entity and hold, then drag it towards the right panel where one or more data fields will turn grey to indicate that the selected entity can populate those fields.
5. Drag the entity into the relevant data field (see Figure 14) and release the mouse click. The entity is now ‘dropped’ in that data field. **NOTE**: Depending on the number of data fields, you may have to scroll down to see the relevant data field. If you do not see any data field turn grey, then check in the schema editor that the “allowTagTypes” is spelled correctly.

Figure 14 Drag and drop entities (zoom)

6. Drag and drop all relevant entities for a particular data cluster and click “add” at the bottom of the right panel (see Figure 15). This saves the data cluster and empties the fields for the next cluster.
7. After a data cluster has been saved, the number of saved data clusters will be indicated on the right panel as “SHOW EVENT (#)” (see Figure 15). Click on that latter menu item to bring up all saved data clusters and select one to edit or delete.
8. Click “UPDATE” to save changes, “CANCEL” to undo changes, and “DELETE” to remove the data cluster (see Figure 16).

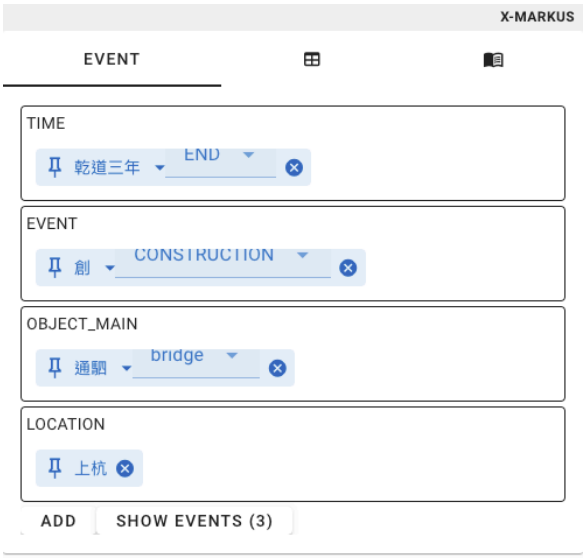

Figure 15 Right panel (cropped)

Figure 16 Edit or Delete Event (cropped)

9. Save your last data cluster by clicking “ADD” or manually remove all entities before moving on to the next file.

Metadata Editor

Metadata can be added for each individual file in the metadata editor (see Figure 17).

1. Click on the second tab of the right panel  to open the metadata editor.

Figure 17 Metadata editor (red box) in the right panel

2. Like data fields, the metadata fields can be modified. The default metadata schema prompts for information on the title, author, time, and place of the text in the file as well as the title, author, time, and place of the text's source.
3. New metadata fields can be added by clicking the + icon at the top of the editor (see red circle in Figure 18).
4. Each field consists of four items: Metadata field name, ID, Value, and type.

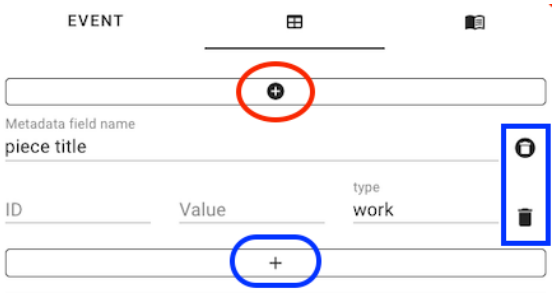

Figure 18 Metadata editor (zoom)

5. The Metadata field name can be modified to suit specific project needs.
6. Drag and drop entities into Value, and COMARKUS will automatically add ID and type.
7. Metadata items can be removed by clicking the bin icons on the right side (see blue square in Figure 18).
8. Metadata items can be added by clicking the + icon at the bottom (see blue circle in Figure 18).
9. The metadata will be displayed below the file name in the main viewer (see Figure 19).

Figure 19 Metadata in the main viewer (cropped)

Tag-editing

Tagged entities can be edited or deleted, and untagged text can be tagged in the main viewer.

Adding or removing a tag

1. Select the characters you want to tag and click "MARKUP" (see Figure 20).
2. Choose the relevant tag in the pop-up window and click "MARKUP" (see Figure 21).
3. Click Remove tag to remove a tag and click yes in the pop-up prompt (see Figure 22).

Figure 20 Adding a tag (cropped)

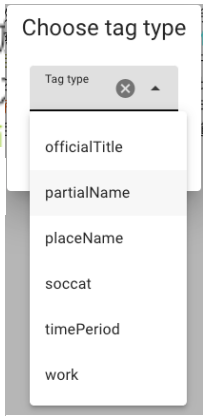

Figure 21 Add a tag prompt (cropped)

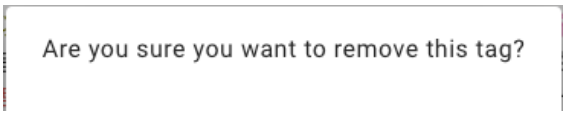

Figure 22 Remove a tag prompt (cropped)

Editing a tag ID

1. Click on a tag and four icons will appear (see Figure 23):
    a. Search dictionary for tagged text;
    b. Search dictionary for ID;
    c. Edit ID;
    d. Remove tag.

Figure 23 Tag-editing (cropped)

4. The dictionary functions search CBDB, TGAZ, and the DILA Glossary databases for either the tagged text or the ID. This allows users to access additional data and link entities to external databases.
5. Click Edit ID to change the ID of a tag, change the ID and click save (see Figure 24).

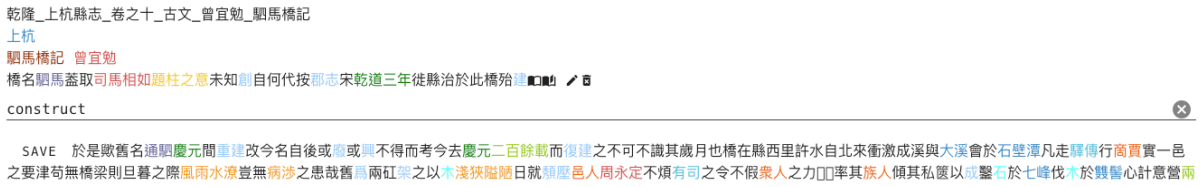

Figure 24 Edit tag ID (cropped)

Exporting Data

Saving events/data clusters

1. COMARKUS automatically saves events/data clusters in json files in the same folder as their corresponding MARKUS files (see Figure 25).
2. The json files are automatically created when you give COMARKUS permission to edit the files in your local folder.
3. The json files are updated every time you add or edit an event/data cluster.

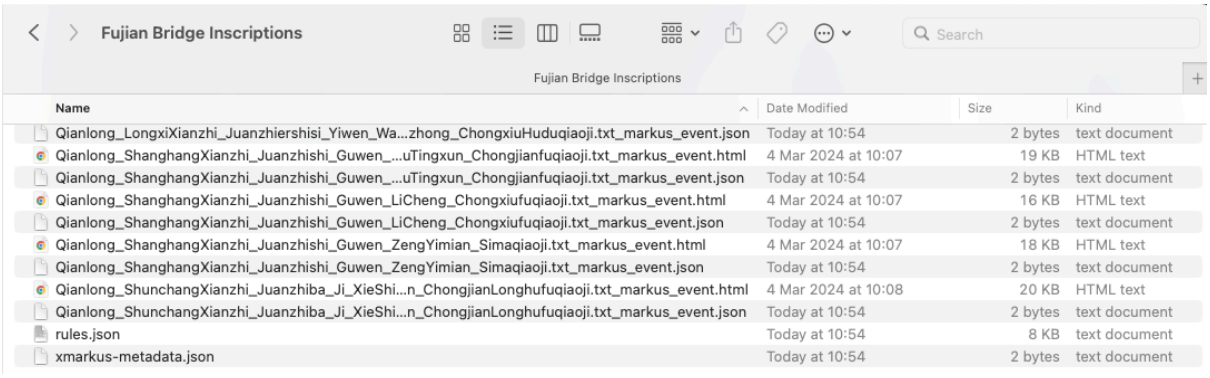

Figure 25 Local folder with json files

Saving your data schema

1. COMARKUS downloads the default schema into your local folder under the name rules.json (see Figure 25).
2. Edits made to the schema are saved in rules.json in your local folder.

3. When a rules.json file is present in your local folder, this file takes precedent over the default schema. You can copy and paste the rules.json with your adjusted schema into another folder, and when you open this folder in COMARKUS, your adjusted schema will be used instead of the default schema.

Saving your metadata schema and metadata

1. COMARKUS saves metadata in a json file called xmarkus-metadata.json (see Figure 25)

Exporting data

1. The json files can be uploaded to visualization and analysis platforms, individually or in batches, to analyze your data clusters.