KanripoX: A tagset for connecting digital texts

Christian Wittern

February 14, 2021

1 Abstract

Our cultural heritage is based to a large degree on texts. Every one of these texts is part of a network of ideas, allusions, quotations, commentaries, criticisms, improvements, adaptions and many other ways of relating. Some of these relations can be described at the level of metadata, such as in library catalogs, or by classifying them in categories. In this paper, I want to describe a way to establish relations between texts on a much lower level, to align corresponding passages as a way to support the establishment of intertextual networks.

This work originated in the practical problem of combining two existing repositories of premodern Chinese texts and is an attempt to solve this in a way that can be generalized to other projects.

2 The Kanseki Repository

The original version of the Kanseki Repository was developed since 2013 by a research group at the Center for Informatics in East-Asian Studies, Institute for Research in Humanities, Kyoto University. The website serving as one of the interfaces to the Repository was opened to the public in March 2016. Since then minor revisions and developments have been carried out, and new texts have been added regularly[1].

2.1 Overall structure

The Kanseki Repository (KR) consists of three components:

- Text repositories at GitHub: @kanripo/*
- A user facing webserver accessed through www.kanripo.org
- An extension to Emacs Orgmode called Mandoku[2] (as of January 2021, there have been 1350 downloads of the module at `https://melpa.org/#/mandoku` since October 2014)

[1]For more information on the Kanseki Repository, please see ウィッテルン・クリスティアン（編）：センター研究年報２０１５　特集　漢籍リポジトリ / *Special issue Kanseki Repository* (ed： Christian Wittern) CIEAS Research Report 2015, available at `http://hdl.handle.net/2433/210140`, as well as `http://blog.kanripo.org`.

[2]More information about Mandoku is availabe in *Special issue Kanseki Repository* p. 44-55.

Figure 1: The Kanseki Repository website

The texts in the KR are arranged in 6 main categories and 81 subcategories. The main categories are as follows:

- KR1 經部 Jing bu Confucian Classics (incl. music, dictionaries and elementary learning)
- KR2 史部 Shi bu Historiography and politics
- KR3 子部 Zi bu Masters, philosophers and treatises
- KR4 集部 Ji bu Anthologies (Poetry and Collected Writings)
- KR5 道部 Dao bu Daoist texts
- KR6 佛部 Fo bu Buddhist texts

For every text item, there might be multiple witnesses of the text, in addition, there is at least one version of the text curated by the editors of the KR. For many texts, in addition to a transcribed version of the text, there are also digital facsimiles of the source editions. Providing reliable editions of premodern Chinese texts in a coherent and simple format with a free license was the main reason for starting the project.

2.2 Website

For the benefit of readers not familiar with the Kanseki Repository web site, here is a quick overview with some screenshots.

The landing page of the website is shown in Figure 1, the main categories are available for browsing and search can be conducted for titles or in the texts.

In Figure 2, the first page of the results for a search is displayed, as can be seen, the results are displayed as KWIC (Keyword in Context) table, ordered by the characters

Figure 2: Results of the search for the term 七經 qijing

of the search string and the immediately following characters. Other sort orders and a subset of the search result based on publication time or classification within the repository can be selected from the menu on the left side.

Figure 3 finally shows a text, in this case both the transcribed version and the digital facsimile.

Users can log into to the website with their GitHub user id, this will give access to user customizations, for example in the display of search results. Users can also maintain text collections that allow them to limit search results to specific texts of interest. All user data are stored in repositories in the GitHub account of the user, none is held on the website. Users can also clone texts from the @kanripo account to their own account and edit the texts there. This is also the preferred route for reporting mistakes in the texts.

2.3 Interfaces

In addition to the website, the text of the KR can be accessed through the GitHub API. The KR website also offers an API, which provides access to the metadata and search interface.

2.4 User community

As of January 2021, there are 9351 texts available in the Kanseki Repository. These texts can be accessed through GitHub, either through the GitHub website, or through the Application Program Interface (API). The main access for users is through the website, a tiny group of users is also accessing the site through Mandoku.

- There are 195 registered users who logged into the Kanseki Repository website at least once.

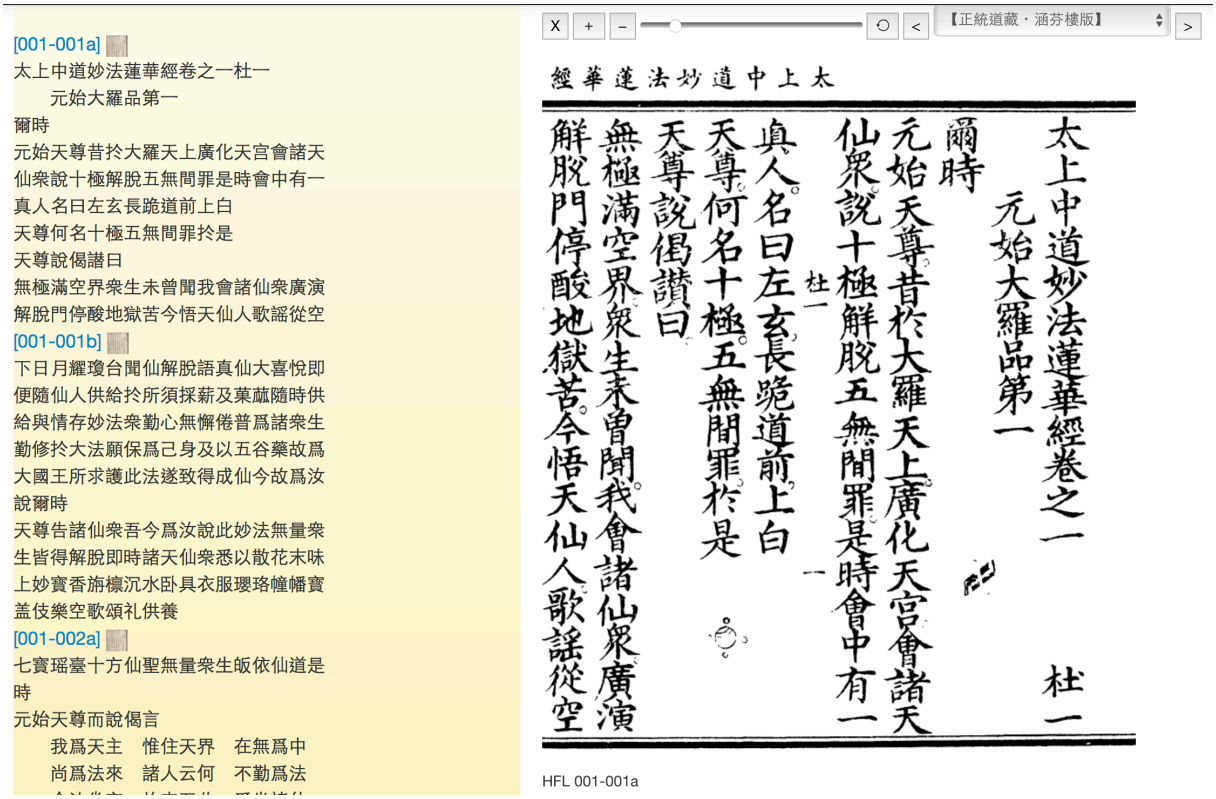

Figure 3: Transcribed text and digital facsimile of KR5h0001 太上中道妙法蓮花經

- 39 Pull requests received until 2021-01-11
- Average count of monthly page accesses to the Kanseki Repository: 2 178 647

3 Thesaurus linguae sericae

The Thesaurus linguae sericae (TLS, Chinese name 漢學文典) is an electronic resource that combines premodern Chinese texts wiht translations and a lexical database with deep linguistic description and a unique grouping of words into an organized hierarchy of concepts[3].

It has a long history, originating in a personal Filemaker database inspired by Christoph Harbsmeier and maintained by Jens Østergaard Petersen. Since 2015, efforts have been underway to provide a new interface and link it with the Kanseki Repository. The latter goal proved evasive so far, and this integration is the very topic of ongoing research, including this paper. So far, the included texts have been updated and expanded and part of the functionality is available now through the web interface at `https://hxwd.org`.

Key features of the TLS are [4]:

[3]For more information about the TLS see Christoph Harbsmeier, "Philological Reflections on Chinese Conceptual History: Introducing Thesaurus Linguae Sericae", in: *Keywords in Chinese Culture*, ed. by Wai-yee Li and Yuri Pines, The Chinese University of Hong Kong Press, 2020.

[4]The following is partly based on: `https://web.archive.org/web/20070609105645/http://www.sino.uni-heidelberg.de/database/tls/index.htm` and `http://ancientworldonline.blogspot.com/2009/11/open-access-historical-and-comparative.html` for more information.

- Integration of source text with translation and analytical tools.
- With its focus on distinctive semantic nuances, it serves as a synonym dictionary of classical Chinese.
- It systematically organises the Chinese vocabulary in taxonomic and mereonomic hierarchies, thus showing up whole conceptual schemes or cognitive systems;
- It systematically registers a range of lexical relations like antonym, converse, epithet etc.; TLS thus aims to define the conceptual space in relation to other words.
- It incorporates detailed syntactic analysis of (over 1200 distinct kinds of) syntactic usage; TLS thus enables users to make a systematic study of such basic phenomena as the natural history of abstract nouns in China.
- TLS explores the conceptual schemes of premodern Chinese on the basis of a corpus of translated texts interlinked with an analytic dictionary.
- Text and dictionary are constantly held up against one another. Our understanding of the texts and the Chinese system of meanings can be refined by through this close confrontation.
- TLS associates Chinese concepts with concepts from the European antiquity, aiming to make the classical Chinese evidence comparable to that of other cultures.
- TLS seeks to make more precise the criteria used in translation classical Chinese, through a detailed description of the semantic relations that obtain among Chinese words.

3.1 TLS in action

In order to better understand the above, here is a series of screenshots that demonstrate some of these features.

In Figure 4 the beginning of the Lunyu is shown, with one of the translations available in the TLS displayed next to the source texts. All texts in the TLS are displayed with a phrase of the Chinese texts as the basic unit, translations are aligned to it. There are two columns to display translations, commentaries or other information related to the text passage in question. Registered users with the necessary permissions can also suggest corrections to the translations or translate new passages.

Figure 5 shows the same passage with the annotations brought in view. Annotations follow the line they comment on, with details of the lemma, the concept this is assigned to, the syntactical function and a translation gloss for the lemma. The buttons to the right of the annotations are used for peer review and comments.

The TLS allows its users to inspect the concepts a given character has already been assigned to (such a combination of character, concept and syntactical function is called a syntactic word, SW in the TLS), this gives an overview of the range of registered semantic fields. In Figure 6 this view is actived for the character 學, which is done by selecting the character. The list of concepts is shown to the right sight. Registered users also use this screen to add new annotations. The numbers to the right show the number of assigned syntactic words within a given concept, for 學/STUDY for example, there are

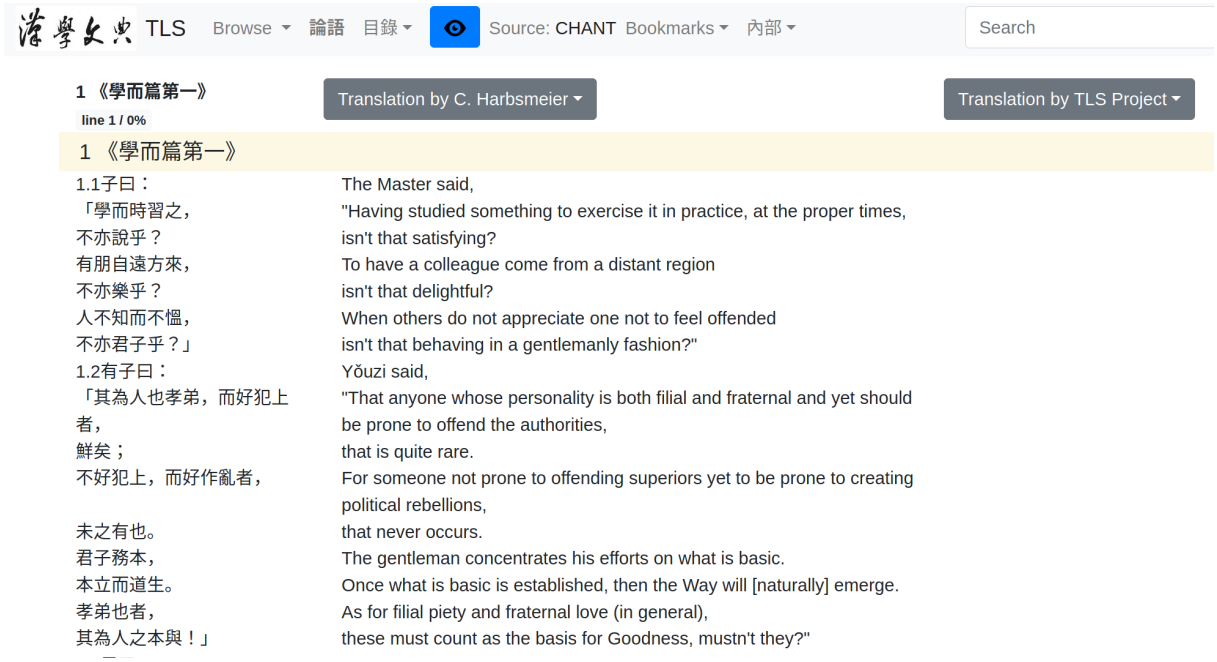

Figure 4: First chapter of the Lunyu with translation

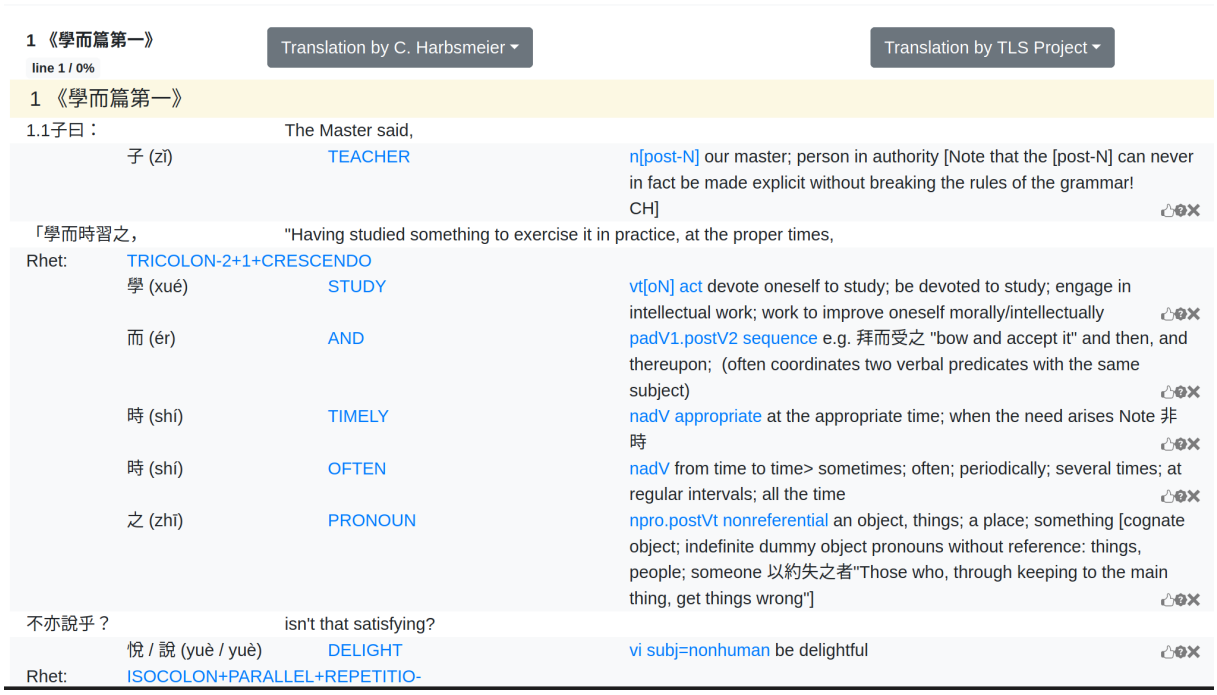

Figure 5: Lunyu, translation and annotations

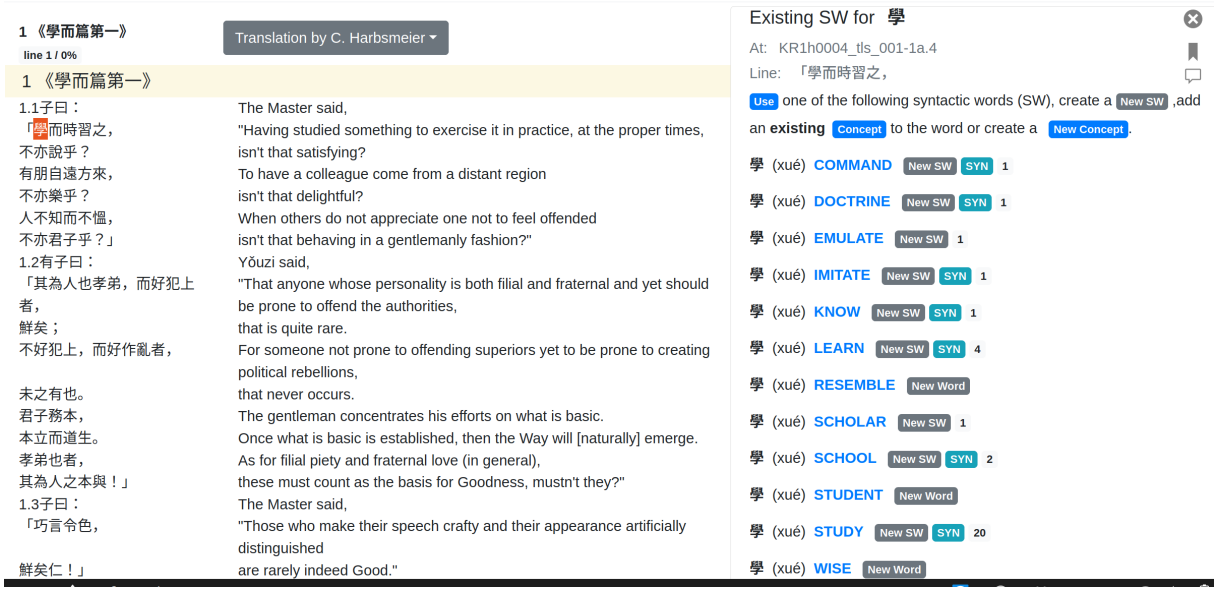

Figure 6: Concepts related to 學 xué

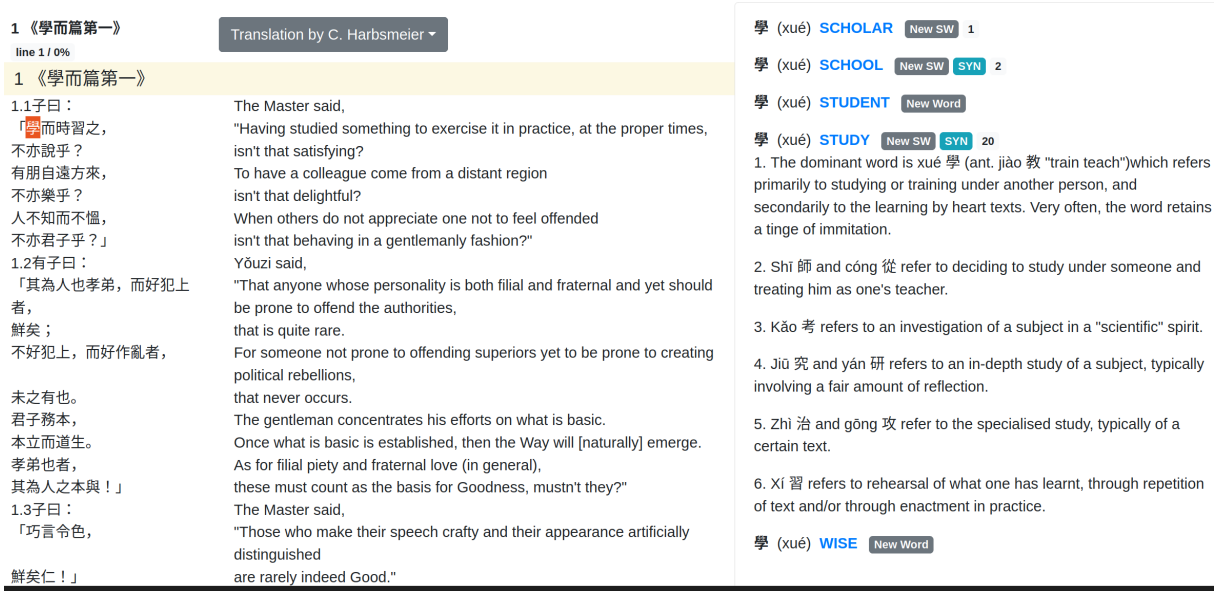

Figure 7: Contrast of words within the concept STUDY

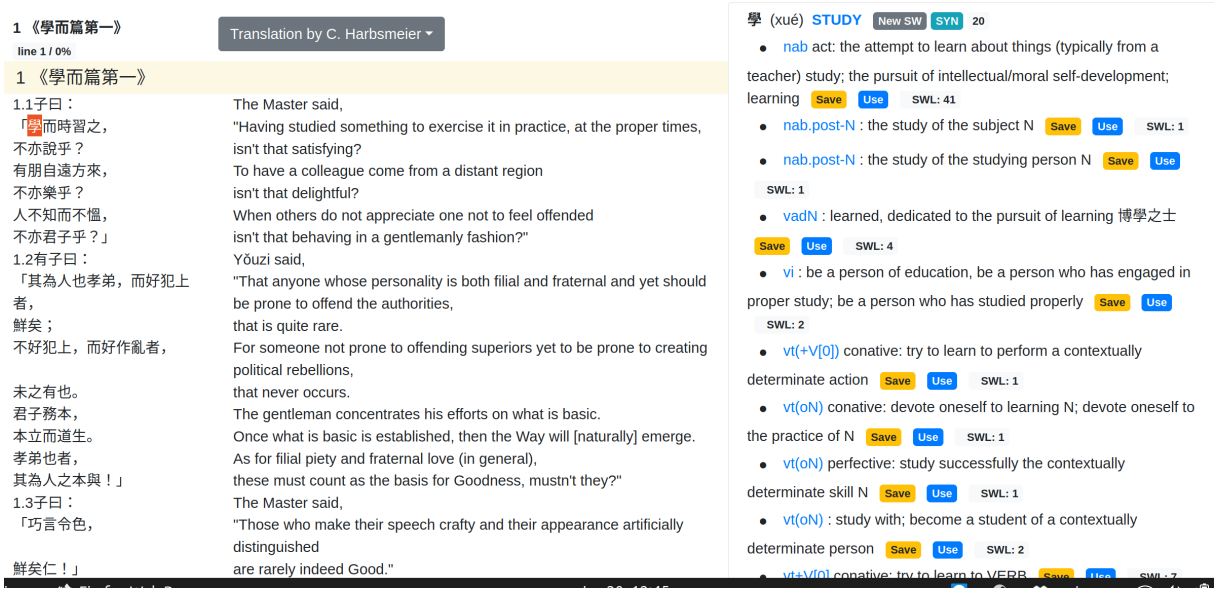

Figure 8: Syntactic words defined for STUDY/學 xué

already 20 SW registered. Clicking on these numbers will reveal the list of SW as shown in Figure 8.

The button in turquoise labelled "SYN" to the left of these numbers is contrasting the different characters that are used to cover various aspects of the concept, these contrasts for the concept STUDY are shown in Figure 7.

The list of SW in Figure 8 also gives the number of instances already registered for each of the SW, these are called syntactic word locations (SWL). In the case of 學 / STUDY / nab, that is an abstract noun, there are 41 text locations, some of which can be seen in Figure 9. Here again the text is given with the translation and there is also a link, which allows to inspect the text location in its context. Figure 10 gives the list of SWL for 學 / STUDY / vt[oN], that is as a transitive verb with a noun object.

4 Use cases for the new tagset

Before moving to a more detailed description of the new tagset, a list of use cases the system hopes to address might be useful. This list only gives desirables of functionality not currently available.

4.1 Describe multiple versions of texts

Multiple versions in different text formats have to be representable, they should be accessible simultaneously to facitilitate text comparison.

It should be easy and straightforward to add new text versions to the collection.

Special cases also need consideration, such as cases where the order of the text is different in some versions.

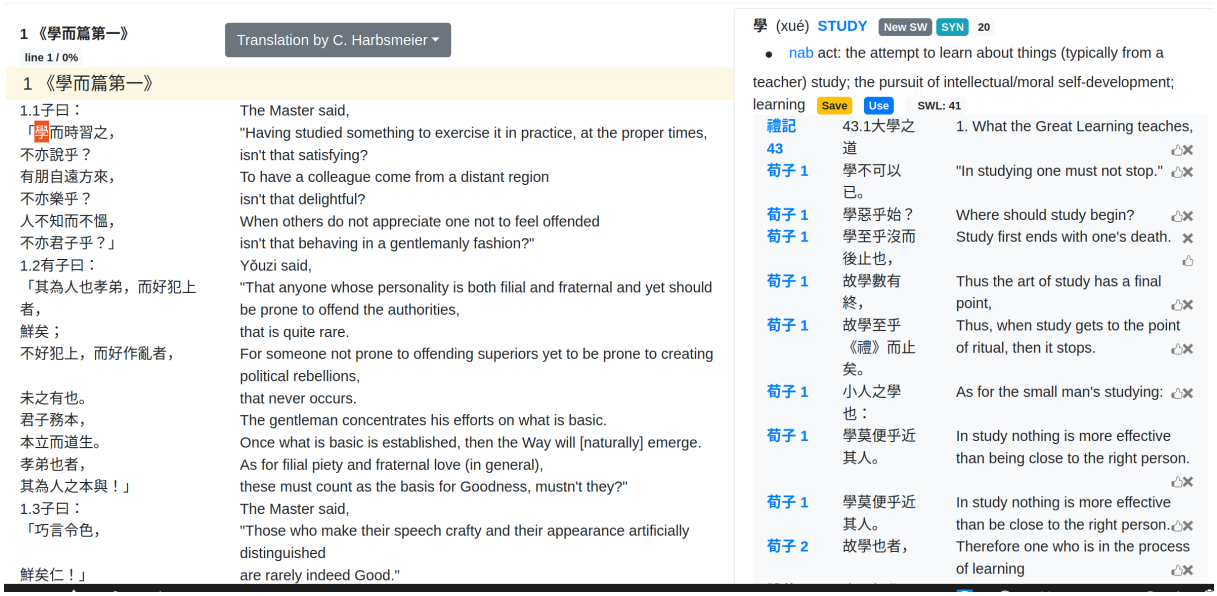

Figure 9: Text references for STUDY/學 xué nab (excerpt)

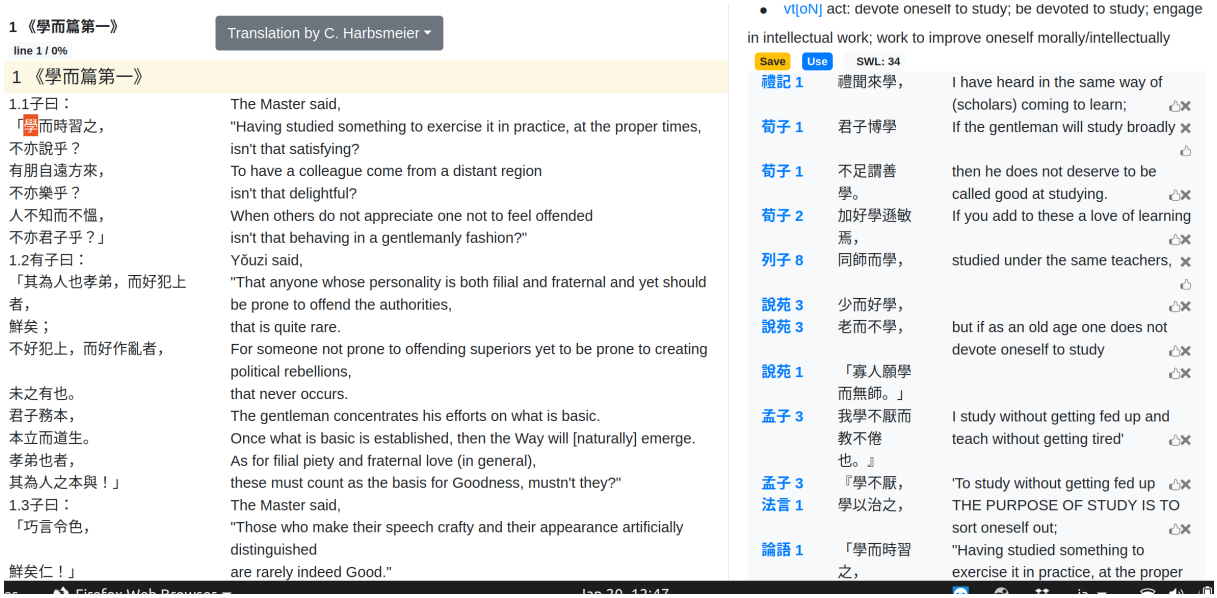

Figure 10: Text references for STUDY/學 xué vt[oN] (excerpt)

4.2 Collection of fragmentary texts / anthology

Virtual description of fragments of texts should be possible, by describing formally what text span from where has to be taken. This could also be an anthology of works in different collections.

4.3 Text and commentary

Linking of root texts and their commentaries: Looking at a text line, all relevant commentaries should be retrievable.

4.4 Annotation, translation, markup

In addition to describing and interlinking existing textual sources, it also should be possible to create and describe new content such as annotations, translations and marking of textual features.

4.5 Groups of related, but distinct texts

Some texts cover related topics, without having a strong textual dependency. To accomodate the reasoning about such texts, a mechanism for dealing with groups of related texts is necessary as a higher level of description.

4.6 Analytic tools

Making the texts available in a simple, standardized form makes it easier to develop analytical tools or to interface with existing tools.

5 XML formats for supporting the infrastructure

Rather than converting the texts in the Kanseki Repository to XML, we continue to maintain the texts using the existing format and tools and connect to the other components of the research environment using a number of predefined XML formats. This has the additional advantage of providing an open architecture with the potential of other applications connecting to the research environment or using parts of it through the same API interfaces. It will also allow to interact with texts in other formats, for example the texts in the TLS, which are currently using TEI XML[5].

Herer are some functional requirements, that need to be catered for:

- grouping and selecting of texts (`manifest`)
  - this also includes identifying and relating parts of texts in a regular way, such as a root text to commentaries or translations.
  - It will also be a convenient way to maintain metadata related to the texts.
  - The implementation should allow users to create their own manifest files, which will make it possible to make collections and metadata adjustable.

[5] An XML format defined by the Text Encoding Initiative (TEI), see `https://www.tei-c.org`.

- links between text passages (`nexus`)
- a format that can be processed to directly compare texts algorithmically (`token`)

5.1 Practical considerations

The `manifest` provides the information the different parts of the system need to interoperate. This includes for example the information where a certain version of a text is located (in the case of the KR, this will include the text ID number and the branch) and what format this version is in.

A utitlity program (or a API interface) will use this information to create a `token` file. This file contains one `<t>` element for each character in the edition, optionally grouped in a potentially nested hierarchy of `<tg>` , i.e. token group element. For examples of these usages please see the tagset documentation provided in the Appendix. If there are several editions representing one text, such a token file will be prepared for each of the editions.

Based on these token files, another utility program can be used to produce the `nexus` files, we currently maintain one such file for every edition, thus providing a view from that specific edition to all the other editions involved. In the interface of the collaborative research platform, this will provide a means to call up different versions of a phrase unter investigation, or provide quick access to the location of this phrase in commentaries. A practical example again is provided in the tagset documentation.

6 KanripoX preview

As a preview of how the information contained in the KanripoX XML files can be used to display additional versions of a text displayed, Figure 11 shows a mockup of this information for a specific line of text. This is still very preliminary and is bound to change. As can be seen, both information on alternate text version, as well as commentaries to the line in question are available.

The `token` files can also be produced from the various text formats by scripts available in the KanripoX GitHub repository. Based on these files, other scripts are used to write the `nexus` files, that list the corresponding text passages. The format is also compatible the CollateX program, which can be used to compare multiple text variants and visualize the results.

7 Conclusion

The proposed direction of development for a new collaborative research platform is still a rough sketch and will need further refinement as the platform is developed and used in practice. It is hoped that the XML format developed here can be used to support this and provide the missing link between specific editions that are maintained in different formats.

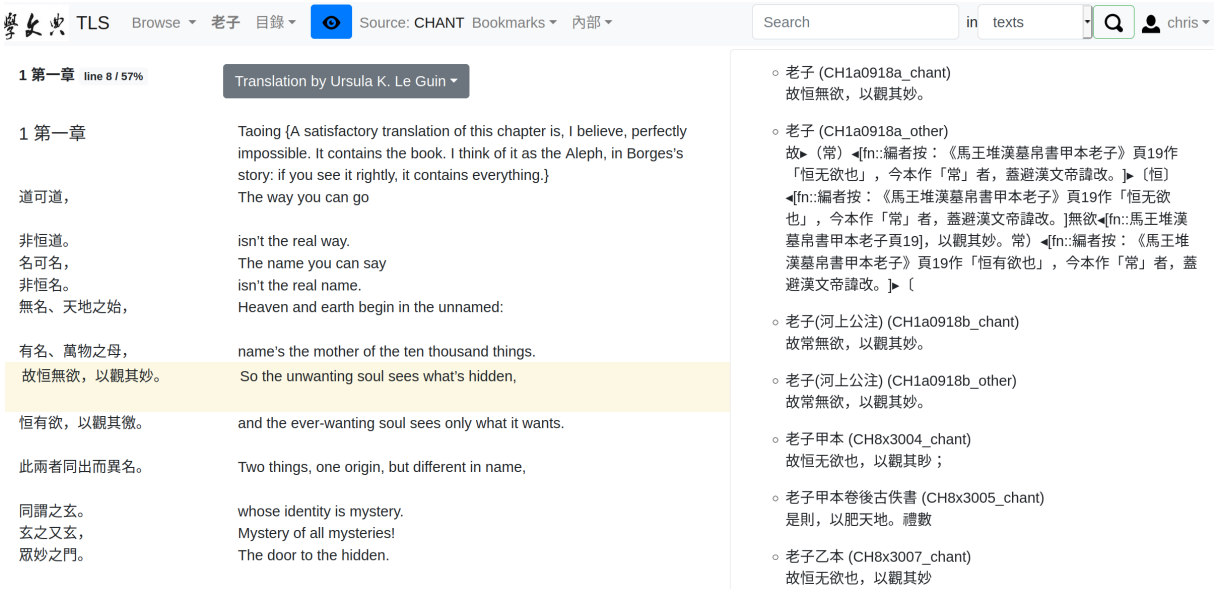

Figure 11: First chapter of Laozi with alternate editions displayed for line "故恒無欲，以觀其妙"

8 References

- Burnard, Lou, and Syd Bauman, eds. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Oxford, Providence, Charlottesville, Nancy: TEI Consortium. 2008.
- Middell, Gregor and Ronald Haentjens Dekker, *CollateX - Software for Collating Textual Sources* [accessed 2021-02-12], `https://collatex.net/`.
- Harbsmeier, Christoph, "Philological Reflections on Chinese Conceptual History: Introducing Thesaurus Linguae Sericae", in: *Keywords in Chinese Culture*, ed. by Wai-yee Li and Yuri Pines, The Chinese University of Hong Kong Press, 2020, p. 381-403.
- KanripoX, [accessed 2021-02-12], `https://github.com/kanripox/kanripox-dev`.
- Wittern, Christian (ed.), *Special issue Kanseki Repository*, CIEAS Research Report 2015, Kyoto 2016 available at `http://hdl.handle.net/2433/210140`.

Appendix: Schema XML files in the KanripoX project

1 Overview

The following sections detail the different file formats that have been defined for the extension of the Kanseki Repository. Although they constitute very different information for different purposes, for the convenience of describing the files and processing the information, they have been combined into one single schema, adressed under one single namespace. The schema allows the following entry points:

<manifests> Root for manifests that contain multiple manifest elements.

<manifest> The root of the manifest. One manifest describes one work.

<tList> Root for token that may contain one or more <tg> elements.

<nexusList> Root for Nexus that may contain one or more <nexus> elements.

The first two of these pertain to the manifest files; the <manifest> definitions can be grouped together into a list of <manifests>, thus providing two entry points for the schema for manifests, while the other two schemas define a list of <nexus> respectively <t> (token) elements, which are grouped together into lists, thus the lists provide the entry point in this case.

2 Grouping and description of texts: the manifest

The `Manifest.xml` described here contains information about a set of editions that are grouped here together, usually for the purpose of further description and processing.

There are two main elements under the root element <manifest>[1]:

<editions> The editions representing the work under consideration. Work is taken in a very broad sense here.

<divisions> The internal subdivisions of the work under consideration.

The <editions> element holds information about the editions that are collected here. It may contain either <editionGroup> elements (which hold <edition> elements), or directly contain <edition> elements, which give the details for each edition. This includes also the *type*, which will be either "documentary" or "interpretative". Documentary editions are editions that strive to reproduce an existing print edition, while interpretative editions do reflect the views of the editor and do not follow one single edition.

Other details for editions that will be collected here are the *id*, which is a unique label (or identifier) used to refer to this specific edition within the manifest and the processing systems.

The <edition> element may have the following children:

<title> Title of the work.

<creation> Information about the creation: date and responsible agent.

<description> Description of the edition or item this element is attached to.

<tokenmap> Mappings from textual features to token types.

<divisions> The internal subdivisions of the work under consideration.

All of these elements are optional. It is useful to give a <title>, which will make references to this edition easier. The <creation> element gives role, function and date of the persons involved in creating this edition. This should refer to the specific edition, more general information pertaining to all editions in this group can be given in the <editionGroup>. The <description> contains other relevant information concerning the edition. <tokenmap> is a way to map features of the text to specific token attributes, details of this are given below. And finally, the <divisions> allows reference to divisions within the edition. This will record the divisions of this specific edtion, a more abstract view of editions for the whole group of texts can be given

[1] There are in fact two possible root elements, the other being <manifests> for a grouping of `manifest` elements.

as a direct child to <manifest>. If the order of the division is not the same in all divisions, they can be re-arranged here and linked to the text location.

Here is an example for an annotated version of the Daode jing:

```
<editionGroup type="annotation">
 <edition format="xml/TEI"
  id="KX5c0065_HFL" language="lzh"
  location="doc/KX5c0065_HFL" type="documentary">
  <creation>
   <date notafter="-100" notbefore="-200">1st century CE</date>
   <resp role=" 注"> 河上公 </resp>
  </creation>
  <description> 道德真經 </description>
 </edition>
</editionGroup>
```

The <divisions> element can also occur as a child element of manifest, optionally following the <editions> element. If used here, there will be only one division element, which holds all subdivisions as possibly nesting elements. The purpose of this element is to provide an entry point to the editions, which is neither tied to one specific edition, nor to a hyperlink or similar in a technical sense. The *label* on the elements is used to provide a human readable label that can be used to point to that specific division, much the same as "Chapter 2" will (usually) refer to the same section of a work, no matter which edition is used. To serve as a link between this nesting structure of chapters, sections and so forth, each can have one or more <edRef> elements, which point to the text span in one of the editions that is covering this specific section.

```
<div label=" 第一章">
 <edRef end="61" key="KR5c0057_tls"
  start="0"/>
 <edRef end="58" key="CH1a0918a_chant"
  start="0"/>
 <edRef end="402" key="CH1a0918b_chant"
  start="2"/>
</div>
```

In this example, the *start* and *end* attributes give the number of the first and last token that is part of this section of a text, thus identifying the text span independent of the text format of the text. Other possibilities for adressing a text span are available if the edition is in TEI/XML.

3 A shadow of the text: the token file

The token files described here serve as a shadow of other digital files that more thoroughly describe the texts documented there. This relieves the token files from the burden to describe the physical appearence, structure and transmission of the text. This information is available at any time by following the links back to these other files. The purpose of the token files is to provide a minimal description, containing only the characters of the text in a form that allows easy comparison and alignment of multiple versions and a very minimal hierarchical structure if necessary. The function is similar to a concordance in that it provides access to the whole text, but without much of what a reader would expect to make reading (or editing) convenient, or even feasible. On the other hand, enough information should be retained to reconstruct a very basic version of the text.

The main elements under the root element <tList>are::

<tg> A group of tokens.

<t> A token.

The <tg> element holds the <t> elements, which have the character content of the text, one token per <t>. The purpose of the <tg> element is to group related <t> elements. <tg> can nest, and provide thus for a rudimentary structure in the token files.

Here is an example of a token file, for the beginning of the Daode jing by Laozi:

```
<tList xmlns="http://kanripo.org/ns/KRX/1.0" ed="KR5c0057_tls" n="tok-0000"
xml:id="KR5c0057_tls-tok-0000">
<tg>
 <t n="KR5c0057_tls_001-1a.3-h" pos="1"
  role="h" tp="0"> 第 </t>
 <t n="KR5c0057_tls_001-1a.3-h" pos="2"
  role="h" tp="1"> 一 </t>
 <t n="KR5c0057_tls_001-1a.3-h" pos="3"
  role="h" tp="2"> 章 </t>
</tg>
<tg xml:id="KR5c0057_tls_001-1a.3">
 <lb ed="KR5c0057_tls"
  n="KR5c0057_tls_001-1a.3"/>
 <t n="KR5c0057_tls_001-1a.3" pos="1"
  role="p" tp="3"> 道 </t>
 <t n="KR5c0057_tls_001-1a.3" pos="2"
  role="p" tp="4"> 可 </t>
 <t f="，" n="KR5c0057_tls_001-1a.3" pos="3"
  role="p" tp="5"> 道 </t>
</tg>
<tg xml:id="KR5c0057_tls_001-1a.4">
 <lb ed="KR5c0057_tls"
  n="KR5c0057_tls_001-1a.4"/>
 <t n="KR5c0057_tls_001-1a.4" pos="1"
  role="p" tp="6"> 非 </t>
 <t n="KR5c0057_tls_001-1a.4" pos="2"
  role="p" tp="7"> 恒 </t>
 <t f="。" n="KR5c0057_tls_001-1a.4" pos="3"
  role="p" tp="8"> 道 </t>
</tg>
<tg xml:id="KR5c0057_tls_001-1a.5">
 <lb ed="KR5c0057_tls"
  n="KR5c0057_tls_001-1a.5"/>
 <t n="KR5c0057_tls_001-1a.5" pos="1"
  role="p" tp="9"> 名 </t>
 <t n="KR5c0057_tls_001-1a.5" pos="2"
  role="p" tp="10"> 可 </t>
 <t f="，" n="KR5c0057_tls_001-1a.5" pos="3"
  role="p" tp="11"> 名 </t>
</tg>
<tg xml:id="KR5c0057_tls_001-1a.6">
 <lb ed="KR5c0057_tls"
  n="KR5c0057_tls_001-1a.6"/>
 <t n="KR5c0057_tls_001-1a.6" pos="1"
  role="p" tp="12"> 非 </t>
 <t n="KR5c0057_tls_001-1a.6" pos="2"
  role="p" tp="13"> 恒 </t>
 <t f="。" n="KR5c0057_tls_001-1a.6" pos="3"
  role="p" tp="14"> 名 </t>
</tg>
<tg xml:id="KR5c0057_tls_001-1a.7">
 <lb ed="KR5c0057_tls"
  n="KR5c0057_tls_001-1a.7"/>
 <t n="KR5c0057_tls_001-1a.7" pos="1"
  role="p" tp="15"> 無 </t>
 <t f="、" n="KR5c0057_tls_001-1a.7" pos="2"
```

```
   role="p" tp="16"> 名 </t>
  <t n="KR5c0057_tls_001-1a.7" pos="3"
   role="p" tp="17"> 天 </t>
  <t n="KR5c0057_tls_001-1a.7" pos="4"
   role="p" tp="18"> 地 </t>
  <t n="KR5c0057_tls_001-1a.7" pos="5"
   role="p" tp="19"> 之 </t>
  <t f="，" n="KR5c0057_tls_001-1a.7" pos="6"
   role="p" tp="20"> 始 </t>
 </tg>
 <tg xml:id="KR5c0057_tls_001-1a.8">
  <lb ed="KR5c0057_tls"
   n="KR5c0057_tls_001-1a.8"/>
  <t n="KR5c0057_tls_001-1a.8" pos="1"
   role="p" tp="21"> 有 </t>
  <t f="、" n="KR5c0057_tls_001-1a.8" pos="2"
   role="p" tp="22"> 名 </t>
  <t n="KR5c0057_tls_001-1a.8" pos="3"
   role="p" tp="23"> 萬 </t>
  <t n="KR5c0057_tls_001-1a.8" pos="4"
   role="p" tp="24"> 物 </t>
  <t n="KR5c0057_tls_001-1a.8" pos="5"
   role="p" tp="25"> 之 </t>
  <t f="，" n="KR5c0057_tls_001-1a.8" pos="6"
   role="p" tp="26"> 母 </t>
 </tg>
<!-- many more tokens ... -->
</tList>
```

4 Links between text passages: the nexus file

The nexus files described here describe links between locations in texts. The links consist of references to a span of one or more consecutive characters in a text, machine readably expressed in terms of references to the <t> elements in the token files. Related links can be grouped together to form a nexus. This can be used for example to describe corresponding passages in different versions of a text.

The main elements under the root element <nexusList>are::

<nexus> A group of <locationRef> elements.

<note> An additional note.

The <nexus> element holds the <locationRef> elements, which contain the reference information to locate the passage of the text. The reference is expressed by pointing to a sequence of one or more tokens in a token file for the edition.

This example shows a <nexus> element from a nexus file for the edition with the identifier "KR5c0057_tls" in the manuscript file. the *tcount* tells us that 6 tokens are involved in this edition, the number of tokens in the other editions is given in an attribute of the same name on the <locationRef> elements. Some have longer sequences, in these cases the commentary pertaining to this phrase has also been included in the parallel text. The *tp* gives the token position, that is the sequential number of the token in the edition, from here it can be seen that in some editions, the corresponding text is not at the beginning, either because of a different sequence (as in "CH8x3004_chant" and "CH8x3007_chant") or because a preface occurs before the text proper, as in "KX5c0065_SBCK".

```
<nexus tcount="6" tp="15"
 xml:id="KR5c0057_tls_001-1a.7">
 <locationRef ed="CH1a0918a_chant"
  target="CH1a0918_CHANT_001-1a.6" tcount="6" tp="14"/>
```

```
 <locationRef ed="CH1a0918a_other"
  target="CH1a0918_CHANT_001-1a.6" tcount="7" tp="26"/>
 <locationRef ed="CH1a0918b_chant"
  target="CH1a0918_CHANT_082-4a.3" tcount="42" tp="121"/>
 <locationRef ed="CH1a0918b_other"
  target="CH1a0918_CHANT_082-4a.3" tcount="41" tp="119"/>
 <locationRef ed="CH8x3004_chant"
  target="CH8x3004_CHANT_002-1a.15" tcount="7" tp="3063"/>
 <locationRef ed="CH8x3005_chant"
  target="CH8x3005_CHANT_002-1a.44" tcount="6" tp="5072"/>
 <locationRef ed="CH8x3006_chant"
  target="CH8x3006_CHANT_001-1a.192" tcount="6" tp="389"/>
 <locationRef ed="CH8x3007_chant"
  target="CH8x3007_CHANT_002-1a.19" tcount="7" tp="3060"/>
 <locationRef ed="KR5c0073_tls"
  target="KR5c0073_tls.seg2-1" tcount="6" tp="35"/>
 <locationRef ed="KX5c0045_HFL"
  target="KX5c0045_HFL_001-001a.03" tcount="6" tp="24"/>
 <locationRef ed="KX5c0045_ZTDZ"
  target="KX5c0045_SJB_001-110474b.03" tcount="6" tp="24"/>
 <locationRef ed="KX5c0046_HFL"
  target="KX5c0046_HFL_000-001a.03" tcount="6" tp="27"/>
 <locationRef ed="KX5c0046_ZTDZ"
  target="KX5c0046_SJB_000-110482a.03" tcount="6" tp="27"/>
 <locationRef ed="KX5c0065_SBCK"
  target="KX5c0065_SBCK_001-1a.06" tcount="38" tp="1020"/>
 <locationRef ed="KX5c0065_ZTDZ"
  target="KX5c0065_SJB_001-120001a.10" tcount="40" tp="119"/>
 <locationRef ed="KX5c0073_HFL"
  target="KX5c0073_HFL_001-001a.05" tcount="6" tp="50"/>
 <locationRef ed="KX5c0073_ZTDZ"
  target="KX5c0073_SJB_001-120272c.05" tcount="6" tp="50"/>
</nexus>
```

5 Schema for Manifest, Token and Nexus

5.1 Elements

<creation> Information about the creation: date and responsible agent.

Module KRXManifest

Contained by description edition editionGroup

May contain

KRXManifest: date resp

Content model

```
<content>
 <alternate maxOccurs="unbounded"
  minOccurs="0">
  <elementRef key="date"/>
  <elementRef key="resp"/>
 </alternate>
</content>
```

Schema Declaration

```
element creation { ( krx_date | krx_resp )* }
```

<date> Date of the work.

Module KRXManifest

Attributes

@notbefore Earliest possible date.

Status Optional

Datatype string

@notafter Latest possible date.

Status Optional

Datatype string

@cert Degree of certainty of this assertion.

Status Optional

Legal values are: **high** High degree of certainty.

middle Middle degree of certainty.

low Low degree of certainty.

Contained by creation

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration

```
element date
{
   attribute notbefore { text }?,
   attribute notafter { text }?,
   attribute cert { "high" | "middle" | "low" }?,
   text
}
```

<description> Description of the edition or item this element is attached to.

Module KRXManifest

Contained by div edition manifest

May contain

KRXManifest: creation note title

character data

Content model

```
<content>
 <alternate maxOccurs="unbounded"
  minOccurs="0">
  <textNode/>
  <elementRef key="note" minOccurs="0"/>
  <elementRef key="title" minOccurs="0"/>
  <elementRef key="creation" minOccurs="0"/>
 </alternate>
</content>
```

Schema Declaration

```
element description { ( text | krx_note? | krx_title? | krx_creation? )* }
```

One specific subdivision on any level.

Module KRXManifest

Attributes

@label A label to identify the subdivision, can be any string, but should be unique in the manifest. This can be used to access this textual division.
Status Optional
Datatype token

@edition A reference to the edition, as defined elsewhere in this manifest.
Status Optional
Datatype IDREF

@sequence Sequencial number of this division, given in such a way that ordering by this number will produce the text in the same sequence as the base edition.
Status Optional
Datatype nonNegativeInteger

@start The sequencial number of the first token of this division in the token list.
Status Optional
Datatype nonNegativeInteger

@end The sequencial number of the last token of this division in the token list.
Status Optional
Datatype nonNegativeInteger

@divid If the source file of this edition has an identifier (usually a xml:id for this subdivision), it can be recorded here.
Status Optional
Datatype token

Contained by div divisions

May contain

KRXManifest: description div edRef label

Content model

```
<content>
 <sequence maxOccurs="1" minOccurs="1">
  <elementRef key="label"
   maxOccurs="unbounded" minOccurs="0"/>
  <elementRef key="description"
   minOccurs="0"/>
  <elementRef key="edRef"
   maxOccurs="unbounded" minOccurs="0"/>
  <elementRef key="div"
   maxOccurs="unbounded" minOccurs="0"/>
 </sequence>
</content>
```

Schema Declaration

```
element div
{
   attribute label { text }?,
   attribute edition { text }?,
   attribute sequence { text }?,
```

```
   attribute start { text }?,
   attribute end { text }?,
   attribute divid { text }?,
   ( krx_label*, krx_description?, krx_edRef*, krx_div* )
}
```

<divisions> The internal subdivisions of the work under consideration.

Module KRXManifest

Attributes

@edition If necessary, the edition for which these textual divisions are valid can be given here.

Status Optional

Datatype token

Contained by edition manifest

May contain

KRXManifest: div

Content model

```
<content>
 <elementRef key="div"
  maxOccurs="unbounded" minOccurs="1"/>
</content>
```

Schema Declaration

```
element divisions { attribute edition { text }?, krx_div+ }
```

<edRef> Reference to this subdivision in one specific edition, identified by the `key`.

Module KRXManifest

Attributes

@start The sequencial number of the first token of this division in the token list.

Status Optional

Datatype nonNegativeInteger

@end The sequencial number of the last token of this division in the token list.

Status Optional

Datatype nonNegativeInteger

@key A reference to the edition, as defined elsewhere in this manifest.

Status Optional

Datatype IDREF

@timestamp The timestamp in ISO format, e.g. `2020-10-09T14:23:52+09:00`.

Status Optional

Datatype dateTime

@label A label to identify the subdivision as used in this edition. It can be any string, but should be unique in the manifest. This can be used to access this textual division.

Status Optional

Datatype token

Contained by div

May contain Empty element

Content model `<content> <empty/></content>`

Schema Declaration

```
element edRef
{
   attribute start { text }?,
   attribute end { text }?,
   attribute key { text }?,
   attribute timestamp { text }?,
   attribute label { text }?,
   empty
}
```

<edition> One edition of the work. If there are multiple <divisions>, this indicates the sequence of these divisions make up the work.

Module KRXManifest

Attributes

@xml:id The identifier of the work. This will be used to refer to this manifest from the display of this text.

Status Optional

Datatype ID

@id The identifier of the edition. This is required and has to be unique within this manifest. It will be used by the processing tools to refer to this edition.

Status Required

Datatype ID

@format The parsing tool is selected based on the format given here, there are two formats defined at the moment. Additional formats can be added, but require a plugin to parse them.

Status Required

Legal values are: **xml/TEI** TEI file encoded in XML.

txt/mandoku Mandoku format.

@location This gives either the relative path to the local folder containing the edition or a resolvable remote reference to the edition, for example on github.

Status Required

Datatype string

Note TODO: format for remote reference.

TODO: Format for identifying portion of text in file.

@base The edition marked as 'base' is the reference edition for sequential reordering.

Status Optional

Legal values are: **true** This edition is the reference edition.

false This edition is not the reference edition (default).

@type The edition has to be declared as either 'documentary' or 'interpretative' .

Status Required

Legal values are: **documentary** An edition that documents an existing print source as faithful as possible, without editorial changes.

interpretative An edition that might be based on a print source, but possibly makes editorial changes.

@role One of the editions has to be declared as the base edition, the others are reference editions.

Status Recommended

Legal values are: **base** This edition is the base edition.

reference All editions except the base edition are considered reference editions.[Default]

@language The language of the document, identified with an identifier according to RFC 1766.

Status Optional

Datatype language

@sigle A short identifier used to identify this edition.

Status Optional

Datatype string

Contained by editionGroup editions

May contain

KRXManifest: creation description divisions title tokenmap

Content model

```
<content>
 <sequence maxOccurs="1" minOccurs="1">
  <elementRef key="title" maxOccurs="1"
   minOccurs="0"/>
  <elementRef key="creation" maxOccurs="1"
   minOccurs="0"/>
  <elementRef key="description"/>
  <elementRef key="tokenmap" maxOccurs="1"
   minOccurs="0"/>
  <elementRef key="divisions"
   maxOccurs="unbounded" minOccurs="0"/>
 </sequence>
</content>
```

Schema Declaration

```
element edition
{
   attribute xml:id { text }?,
   attribute id { text },
   attribute format { "xml/TEI" | "txt/mandoku" },
   attribute location { text },
   attribute base { "true" | "false" }?,
   attribute type { "documentary" | "interpretative" },
   attribute role { "base" | "reference" }?,
   attribute language { text }?,
   attribute sigle { text }?,
   (
      krx_title?,
      krx_creation?,
      krx_description,
      krx_tokenmap?,
```

```
      krx_divisions*
   )
}
```

<editionGroup> A group of the editions representing the work under consideration.

Module KRXManifest

Attributes

@type The treatment of the editions within this group are based on the value of this attribute.

Status Required

Legal values are: **root** The root text of this work.

root+annotation The root text, interspersed with commentary.

annotation Commentary to the root text, without repeating the text.

translation Translations of the text and / or commentary.

other Texts, that are grouped with this texts for some reason other than being textually related.

@sigle A short identifier used to identify this group of editions.

Status Optional

Datatype string

Contained by editions

May contain

KRXManifest: creation edition title

Content model

```
<content>
 <sequence maxOccurs="1" minOccurs="1">
  <elementRef key="title" maxOccurs="1"
   minOccurs="0"/>
  <elementRef key="creation" maxOccurs="1"
   minOccurs="0"/>
  <elementRef key="edition"
   maxOccurs="unbounded" minOccurs="1"/>
 </sequence>
</content>
```

Schema Declaration

```
element editionGroup
{
   attribute type
   {
      "root" | "root+annotation" | "annotation" | "translation" | "other"
   },
   attribute sigle { text }?,
   ( krx_title?, krx_creation?, krx_edition+ )
}
```

<editions> The editions representing the work under consideration. Work is taken in a very broad sense here.

Module KRXManifest

Contained by manifest

May contain

KRXManifest: edition editionGroup

Content model

```
<content>
 <alternate maxOccurs="1" minOccurs="1">
  <elementRef key="editionGroup"
   maxOccurs="unbounded" minOccurs="1"/>
  <elementRef key="edition"
   maxOccurs="unbounded" minOccurs="1"/>
 </alternate>
</content>
```

Schema Declaration

```
element editions { krx_editionGroup+ | krx_edition+ }
```

<label> Additional label.

Module KRXManifest

Attributes

@language The language of the label, identified with an identifier according to RFC 1766.
Status Optional
Datatype language

Contained by div

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration

```
element label   attribute language   text  ?, text
```

<lb> This element marks the beginning of a new line or line-like section on the text-bearing surface.

Module derived-module-KRX

Attributes

@ed Identifier of the edition to which this line belongs.
Status Optional
Datatype string

@n Number or other label used to refer to this line.
Status Optional
Datatype string

@xml:id *Status* Recommended
Datatype ID

KRXToken: tg

May contain Empty element

Content model `<content> <empty/></content>`

Schema Declaration

```
element lb
{
   attribute ed { text }?,
   attribute n { text }?,
   attribute xml:id { text }?,
   empty
}
```

<locationRef>

Reference to a location in the token file. Optionally might hold a copy of the referenced text as a string of characters.

Module KRXNexus

Attributes

`@ed` Identifier of the edition (as used in the token file).
- *Status* Required
- *Datatype* string

`@tp` The sequencial number of the first token in the token file.
- *Status* Required
- *Datatype* nonNegativeInteger

`@tcount` The number of tokens that make up this text span.
- *Status* Optional
- *Datatype* nonNegativeInteger
- *Default* 1

`@target` Identifier of the first token in the text span.
- *Status* Required
- *Datatype* string

`@n` Label or identifier for this reference.
- *Status* Optional
- *Datatype* string

KRXNexus: nexus

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration

```
element locationRef
{
   attribute ed { text },
   attribute tp { text },
   attribute tcount { text }?,
   attribute target { text },
   attribute n { text }?,
   text
}
```

<manifest>

The root of the manifest. One manifest describes one work.

Module KRXManifest

Attributes

@xml:id The identifier of the work. This will be used to refer to this manifest from the display of this text.

Status Optional

Datatype ID

Contained by manifests

May contain

KRXManifest: description divisions editions title

Note Currently, only one work can be described per one manifest file. Need to think about what to do with use cases that need multiple works. Use several <manifest> in a file?

Content model

```
<content>
 <sequence maxOccurs="1" minOccurs="1">
  <elementRef key="title" minOccurs="0"/>
  <elementRef key="description"/>
  <elementRef key="editions"/>
  <elementRef key="divisions" minOccurs="0"/>
 </sequence>
</content>
```

Schema Declaration

```
element manifest
{
   attribute xml:id { text }?,
   ( krx_title?, krx_description, krx_editions, krx_divisions? )
}
```

<manifests>

Root for manifests that contain multiple manifest elements.

Module KRXManifest —

May contain

KRXManifest: manifest

Content model

```
<content>
 <elementRef key="manifest"
  maxOccurs="unbounded"/>
</content>
```

Schema Declaration

```
element manifests   krx_manifest+
```

<map>

Map of one textual feature to a specific token type.

Module derived-module-KRX

Attributes

@src Element or simple matching expression (for XML texts) or regular expressions (for plain text) that identifies the textual feature.

Status Optional

Datatype string

@tok Token type.

Status Optional

Legal values are: **h** Token is part of a heading.

p Token is part of a paragraph.

n Token is part of a note or annotation of any kind.

q Token is part of a quotation.

v Token is part of a verse line

Contained by tokenmap

May contain Empty element

Content model `<content> <empty/></content>`

Schema Declaration

```
element map
{
   attribute src { text }?,
   attribute tok { "h" | "p" | "n" | "q" | "v" }?,
   empty
}
```

<nexus> A group of <locationRef> elements.

Module KRXNexus

Attributes

@xml:id The identifier of this token group.

Status Optional

Datatype ID

@tp The sequencial number of the first token of this text span.

Status Required

Datatype nonNegativeInteger

@tcount The number of tokens that make up this text span.

Status Optional

Datatype nonNegativeInteger

Default 1

KRXNexus: nexusList

May contain

KRXManifest: note

KRXNexus: locationRef

Content model

```
<content>
 <sequence maxOccurs="1" minOccurs="1">
  <elementRef key="note"
   maxOccurs="unbounded" minOccurs="0"/>
  <elementRef key="locationRef"
   maxOccurs="unbounded" minOccurs="0"/>
 </sequence>
</content>
```

Schema Declaration

```
element nexus
{
   attribute xml:id { text }?,
   attribute tp { text },
   attribute tcount { text }?,
   ( krx_note*, krx_locationRef* )
}
```

<nexusList> Root for Nexus that may contain one or more <nexus> elements.

Module KRXNexus

Attributes

@xml:id *Status* Recommended

Datatype ID

@ed Reference to the edition defined in the manifest.

Status Required

Datatype string

@n A label

Status Optional

Datatype string

—

May contain

KRXManifest: note

KRXNexus: nexus

Content model

```
<content>
 <sequence maxOccurs="1" minOccurs="1">
  <elementRef key="note" maxOccurs="1"
   minOccurs="0"/>
  <elementRef key="nexus"
   maxOccurs="unbounded"/>
 </sequence>
</content>
```

Schema Declaration

```
element nexusList
{
   attribute xml:id { text }?,
   attribute ed { text },
   attribute n { text }?,
   ( krx_note?, krx_nexus+ )
}
```

<note> An additional note.

Module KRXManifest

Contained by description

KRXNexus: nexus nexusList

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration `element note  text`

<pb>

This element marks the beginning of a new page or page-like section on the text-bearing surface.

Module derived-module-KRX

Attributes

@ed Identifier of the edition to which this page belongs.

Status Optional

Datatype string

@n Number or other label used to refer to this page.

Status Optional

Datatype string

@xml:id *Status* Recommended

Datatype ID

KRXToken: tg

May contain Empty element

Content model `<content> <empty/></content>`

Schema Declaration

```
element pb
{
   attribute ed { text }?,
   attribute n { text }?,
   attribute xml:id { text }?,
   empty
}
```

<resp>

Person responsible for some aspect of the work.

Module KRXManifest

Attributes

@role *Status* Optional

Datatype string

Sample values include: **author** Author

compiler Compiler

translator Translator

@key A key identifying this person in some reference system.

Status Optional

Datatype string

Contained by creation

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration

```
element resp { attribute role { text }?, attribute key { text }?, text }
```

<t> A token.

Module KRXToken

Attributes

@role Token type.

Status Required

Legal values are: **h** Token is part of a heading.

p Token is part of a paragraph.

s Token is part of a seg element.

n Token is part of a note or annotation of any kind.

q Token is part of a quotation.

v Token is part of a verse line.

o Token is part of a textual feature not in this list.

@pos The sequencial number of this token within this element (or token type).

Status Optional

Datatype nonNegativeInteger

@tp The sequencial number of this token within the whole text.

Status Required

Datatype nonNegativeInteger

@f Punctuation or other non-token text items, immediately following the token.

Status Optional

Datatype string

@p Punctuation or other non-token text items, immediately preceding the token.

Status Optional

Datatype string

@n Label or identifier of the element in the text of which this token is part. If none is available, the code generating the token file should make one up on the fly.

Status Required

Datatype string

@cp Codepoint of the token character.

Status Optional

Datatype nonNegativeInteger

@position position and content of marks out of line, but related to this token. The description is similar to CSS description on HTML @style: 'left: か;' would indicate a か syllable to the left of this token.

Status Optional

Datatype string

@kundokuten Kundoku marks related to this token.

Status Optional

Datatype string

@ruby Pronounciation marks related to this token.

Status Optional

Datatype string

KRXToken: tg

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration

```
element t
{
   attribute role { "h" | "p" | "s" | "n" | "q" | "v" | "o" },
   attribute pos { text }?,
   attribute tp { text },
   attribute f { text }?,
   attribute p { text }?,
   attribute n { text },
   attribute cp { text }?,
   attribute position { text }?,
   attribute kundokuten { text }?,
   attribute ruby { text }?,
   text
}
```

<tList> Root for token that may contain one or more <tg> elements.

Module KRXToken

Attributes

@xml:id *Status* Recommended

Datatype ID

@ed Reference to the edition defined in the manifest.

Status Required

Datatype string

@n A label

Status Optional

Datatype string

@fileseq If the tokens are in several files, this gives the sequential number of the file.

Status Optional

Datatype nonNegativeInteger

—

May contain

KRXToken: tg

Content model

```
<content>
 <elementRef key="tg" maxOccurs="unbounded"/>
</content>
```

Schema Declaration

```
element tList
{
   attribute xml:id { text }?,
   attribute ed { text },
```

```
   attribute n { text }?,
   attribute fileseq { text }?,
   krx_tg+
}
```

<tg> A group of tokens.

Module KRXToken

Attributes

@xml:id The identifier of this token group.

Status Optional

Datatype ID

@n A label.

Status Optional

Datatype string

@role Token group type.

Status Optional

Legal values are: **h** Token group is a heading.

p Token group is (part of) a paragraph.

s Token group is a seg element.

n Token group is (part of) a note or annotation of any kind.

q Token group is (part of) a quotation.

v Token group is (part of) a verse line.

o Token group is (part of) a textual feature not in this list.

@position position and content of marks out of line, but related to this token. The description is similar to CSS description on HTML @style: 'left: か;' would indicate a か syllable to the left of this token.

Status Optional

Datatype string

@kundokuten Kundoku marks related to this token.

Status Optional

Datatype string

@ruby Pronounciation marks related to this token.

Status Optional

Datatype string

KRXToken: tList tg

May contain

KRXToken: t tg

derived-module-KRX: lb pb

Content model

```
<content>
 <alternate maxOccurs="unbounded"
  minOccurs="0">
  <elementRef key="tg"
   maxOccurs="unbounded" minOccurs="0"/>
  <elementRef key="t" maxOccurs="unbounded"
   minOccurs="0"/>
  <elementRef key="pb"
```

```
   maxOccurs="unbounded" minOccurs="0"/>
  <elementRef key="lb"
   maxOccurs="unbounded" minOccurs="0"/>
 </alternate>
</content>
```

Schema Declaration

```
element tg
{
   attribute xml:id { text }?,
   attribute n { text }?,
   attribute role { "h" | "p" | "s" | "n" | "q" | "v" | "o" }?,
   attribute position { text }?,
   attribute kundokuten { text }?,
   attribute ruby { text }?,
   ( krx_tg* | krx_t* | krx_pb* | krx_lb* )*
}
```

<title> Title of the work.

Module KRXManifest

Contained by description edition editionGroup manifest

May contain Character data only

Content model `<content> <textNode/></content>`

Schema Declaration `element title   text`

<tokenmap> Mappings from textual features to token types.

Module KRXManifest

Contained by edition

May contain

derived-module-KRX: map

Content model

```
<content>
 <elementRef key="map"
  maxOccurs="unbounded" minOccurs="1"/>
</content>
```

Schema Declaration `element tokenmap   krx_map+`