



Building and Maintaining the TEI LingSIG Bibliography

Using Open Source Tools for an Open Content Initiative

**Piotr Bański, Stefan Majewski, Maik Stührenberg and Antonina
Werthmann**



Electronic version

URL: <http://journals.openedition.org/jtei/486>

DOI: 10.4000/jtei.486

ISSN: 2162-5603

Publisher

TEI Consortium

Electronic reference

Piotr Bański, Stefan Majewski, Maik Stührenberg and Antonina Werthmann, « Building and Maintaining the TEI LingSIG Bibliography », *Journal of the Text Encoding Initiative* [Online], Issue 3 | November 2012, Online since 15 October 2012, connection on 19 April 2019. URL : <http://journals.openedition.org/jtei/486> ; DOI : 10.4000/jtei.486

This text was automatically generated on 19 April 2019.

TEI Consortium 2012 (Creative Commons Attribution-NoDerivs 3.0 Unported License)

Building and Maintaining the TEI LingSIG Bibliography

Using Open Source Tools for an Open Content Initiative

Piotr Bański, Stefan Majewski, Maik Stührenberg and Antonina Werthmann

1. Introduction

- 1 While the TEI has been successful in becoming a de facto standard for numerous applications in Digital Humanities, its status in the area of linguistic annotation is not as clear. After the initial success of the TEI-encoded British National Corpus (Dunlop 1995), the TEI has given way to simpler and more specialized formats for corpus annotation, such as (X)CES (Ide et al. 1996; Ide 2000), TigerXML (Mengel and Lezius 2000; Lezius 2002), and, more recently, PAULA (Dipper and Götze 2005; Dipper et al. 2007). Currently, the ISO TC37 SC4 committee is working on the so-called LAF (Linguistic Annotation Framework) family of standards: see (Stührenberg 2012) for more details.
- 2 The LingSIG (the “TEI for Linguists” special interest group of the TEI)¹ has been created to examine the actual and potential relationship between TEI markup and the needs and requirements of linguists. This goal may require adapting (or re-adapting) TEI markup to the common tasks faced in everyday linguistic practice. In order to achieve that, a serious review of existing resources is needed, as well as access to people who are experts in the relevant areas. Both these infrastructural subtasks can be supported by creating a comprehensive bibliography of works dealing with linguistic markup that is TEI-inspired or that may inspire new TEI solutions. This bibliography can serve both as a repository of knowledge and as a resource that can attract non-TEI markup specialists by providing them with a useful service.
- 3 This paper addresses an infrastructural issue of universal relevance—the collective creation of a shared bibliography—congenial with the TEI’s overall aims and methodology and presented here in the context of the LingSIG. Below, we describe a combination of open-source general tools and an open-access approach to creating knowledge

repositories. We believe that, for an initiative such as the TEI, it is important to choose non-proprietary, freely available solutions. If these solutions have the advantage of attracting new users and promoting the initiative itself, so much the better, especially if it is done in a non-committal way: no one using the LingSIG bibliographic repository has to be a user of the TEI. On the other hand, the solution described here may enhance the culture of sharing that the TEI has grown within.

- 4 In what follows, we first mention the roots of the idea to establish a repository of bibliographic references in the context of the TEI LingSIG, then briefly describe Zotero—the tool that has been chosen to create, store and access the repository—and finally present the TEI-Zotero Translator—initially a separate Firefox add-on and now part of the Zotero package that further connects the communities involved by creating a bridge between the bibliographic recommendations of the TEI Guidelines and the activities of the LingSIG.

2. LingSIG Reference Library

- 5 The reference library discussed here is the product of activities connected with the “TEI for Linguists” special interest group of the TEI (LingSIG). The LingSIG’s roots reach back to the Digital Humanities conference in London in 2010, where its future conveners met and decided to prepare a formal application to the TEI Council outlining the SIG’s aims. What soon followed was the informal “LLiZ” (*Linguistic Lunch in Zadar*), organized by Piotr Bański, and the first official SIG meeting during the 2010 Annual Meeting of the TEI Consortium in Zadar. During that meeting, the participants agreed that one of the aims that the SIG should address is the creation of a common repository of references to works that should be taken into account in the process of building a consistent set of TEI encoding proposals targeting the needs of linguists.
- 6 The first version of the reference library was created as a TEI Wiki resource and announced on the SIG mailing list, but, despite an initially positive reaction, the low number of responses indicated that the barrier to active contribution was too high. It became obvious that, although using a wiki opened the resource for *collective building*, it was only a partially successful move: the results could only be pasted straight from the wiki page and each time had to be reformatted to conform to a given style sheet. Furthermore, only a simple web-page search was available to locate references and a lot of work would have to be devoted to maintaining the entries in a uniform shape. A more flexible resource was needed that combined the Web 2.0 idea of collective building and maintenance with greater flexibility of the result format, easier access to bibliographic data and better search facilities. At this point, the decision was made to transfer the development to the Zotero platform.²
- 7 These days, a researcher’s life is punctuated with deadlines. With the date of the next TEI meeting approaching fast, Zotero-based development manifested one more advantage over wiki-based creation: it was rapid. It took only a moment to import the BibTeX of Maik Stührenberg’s extensive linguistic-markup-oriented bibliography and only several days of Antonina Werthmann’s post-editing to create a sizeable and usable resource.

3. Zotero

- 8 Zotero is an open-source citation manager.³ Citation management software is nowadays a standard component in the preparation workflow for scientific texts; most of the available tools offer a standard set of features, including adding and editing bibliographic references, exporting citations formatted according to most standard academic citation styles, working with citations directly from a word processor using a plug-in, and creating searchable catalogues of references. While Zotero offers all these functionalities, it is unique in that it was specifically designed to be used within the context of a web browser.⁴

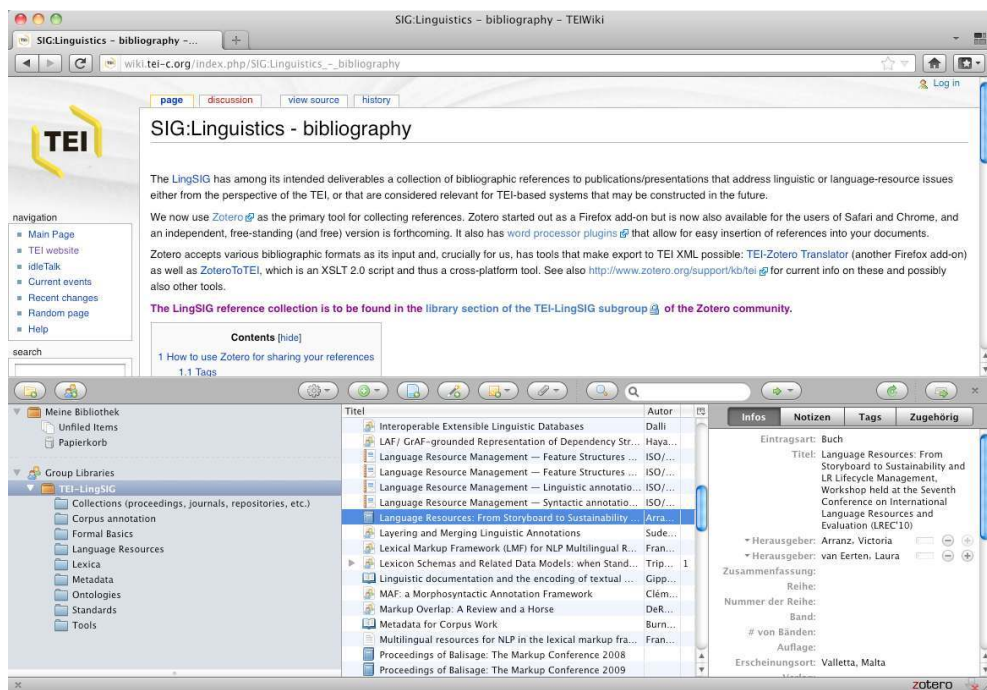


Figure 1. Zotero user-interface, complementing web-oriented research

- 9 Zotero's functionality is designed mainly for web-based research activities. Given the extensive repositories of publicly accessible library catalogues, proprietary services such as Google Scholar, pre-print archives such as arXiv.org, and countless online archives of journals, this functionality can be expected to cover a great part of the bibliographic work for scientific writing in many disciplines. Zotero includes *import translators* which allow the direct import of bibliographic data for items discovered while browsing the Web, reducing time otherwise spent on creating citations manually.
- 10 Apart from having all the advantages of standard web-oriented tools, Zotero offers cloud-based synchronisation features that allow any item edited or changed in one Zotero instance by one collaborator to be updated automatically in all the other instances. Zotero's rigid data model and import translators help to reduce the number of errors that can be introduced by collaborative editing.⁵
- 11 Zotero comes in two flavours: as a plugin for current web browsers or as a stand-alone tool. The first option was built as a plugin for Mozilla Firefox, but since the release of version 3.0.2, the Zotero Connector is also available as a plugin for Google Chrome and Apple Safari. The Zotero stand-alone version, which runs under Windows, Mac OS X and

Linux, has been available since early 2012. Both versions feature connectors to web browsers and plugins for popular word processors, such as Microsoft Word or OpenOffice/LibreOffice/NeoOffice.

3.1. Creating Bibliographies

- 12 New bibliographic items can be edited manually or created automatically from the content of a particular site that the user is visiting (using an import translator). In the first case, the information is entered into a form with predefined fields corresponding to particular types of items (book, book section, journal article, etc.; see the lower right part of fig. 1). In the case of automatic generation of bibliographic items, the required metadata is copied automatically from web pages, though accuracy and completeness depends on whether an import translator is available for the cited content. This includes homepages of publishers, library catalogues, databases of journals and books, but also sites such as scholar.google.com, amazon.com or popular blogging platforms. The availability and quality of the assisted automatic creation of bibliographic items within the Zotero database is dependent on whether the site provides such information and on whether Zotero provides a suitable import plugin, whose presence is indicated by an icon in the browser's address bar. This icon generally corresponds to the available item types and supplies a one-click-solution, that is, by clicking the icon, the user saves all the corresponding metadata in the Zotero database. If a PDF file is available as well, it will be automatically attached to the newly created item. After creating a Zotero item, one may modify it by correcting or adding metadata entries. Finally, the item can be tagged with categories, keywords and additional information.
- 13 In addition to importing data from individual Web pages, Zotero also supports import of bibliographic metadata in the following bibliographic file formats: MODS (Metadata Object Description Schema),⁶ BibTeX, RIS (Research Information System Format), Refer/BibIX,⁷ and Unqualified Dublin Core RDF. Recent discussions on TEI-L and between developers indicated that there is some interest in creating import facilities for TEI bibliographies as well. The LingSIG plans to implement an import feature via a student project or when a particular project that uses the exporter could immediately benefit from reversing the flow of information.

3.2. Working with Reference Libraries

- 14 Once a Zotero library has been created, it is not only possible to use the information stored in the metadata of the respective bibliographic items but also to add notes and attachments (such as electronic versions of articles). In addition, the ability to define tags allows for a very flexible categorization scheme (in addition to the use of folders to organize library items). For the LingSIG library, we have chosen tags such as "XCES", "TEI", "EXMARaLDA", and "BNC"; since these tags can be used for both searching and organizing items, they constitute a facility that is powerful and easy to use.
- 15 Libraries created with Zotero can then be shared among the members of the respective Zotero groups. By joining the LingSIG group,⁸ new members are allowed to use the collection and to add to it in a manner much more straightforward than that offered by wiki-based solutions. All members of the group are allowed to modify the library.⁹ Changes made by group members can be synchronized with the online library either on

demand or automatically. Apart from accessing the library via Zotero front-ends, one can also use APIs for read- and write-access to the library using other tools. File attachments can be synchronized via Zotero File Storage or WebDAV.

3.3. Exporting Bibliographies

- 16 Storing bibliographic items in a Zotero database opens up several export possibilities. Citations and reference lists can be generated by Zotero in a great variety of bibliographic styles as defined by the Citation Style Language (CSL).¹⁰ Some styles, including Chicago, MLA, APA, and Vancouver, are already predefined in Zotero. Others can be installed via the Zotero Style Repository.¹¹
- 17 Apart from exporting single or multiple library items, Zotero can create reports, interactive timelines, and reference lists (the last in a variety of formats, such as HTML or RTF, and according to different styles). It thus promises to be a nearly universal writing aid for the members of the LingSIG, and by extension, the entire TEI community. This is made even more obvious by the fact that, thanks to work by Stefan Majewski and feedback from the TEI community, Zotero is now able to export TEI XML `<biblStruct>` elements directly. This is the topic of the following section.

4. TEI and Zotero

- 18 As we have shown above, there are numerous reasons for choosing Zotero for citation management. While Zotero's integration with major word processors is sufficient for many purposes, text-encoding scholars often have more advanced needs. For this reason, some members of the TEI community have begun developing tools capable of transforming bibliographic items from Zotero to structures that may be used with TEI-encoded documents. The resulting prototypes addressed particular requirements of specific tasks and were not meant to be general-purpose tools, but the creation of the TEI Zotero translator—once a separate Firefox plugin but now integrated into the Zotero code itself—opens the way towards potential standardization in this area.

4.1. Possible Translation Workflows

- 19 Two approaches have been used for exporting bibliographic items from Zotero to TEI. Firstly, it is possible to take one of the standardized output formats that are supported by default (such as MODS¹² and Zotero RDF¹³) and translate that into TEI XML by means of an XSL transformation. Another option is to extend Zotero to provide facilities to directly export its library to TEI XML. From the conceptual perspective, both approaches are similar: the main challenge is to find the appropriate mapping between Zotero fields and their closest matches in the TEI. Nevertheless, they differ in the workflow required to generate the TEI encoding. The first approach requires an additional transformational step after the initial export into an intermediate format.¹⁴ The other approach implements the transformation as a built-in Zotero feature that might be selected as an option on export. Clearly, the latter requires one fewer step by the user, offers greater stability (due to its lesser dependence on an intermediate format controlled by a third party), and makes the task of maintenance simpler: only the initial and the target data structures have to be considered, not how these map to the intermediate format. The

downside of this approach is that it requires the export translator to be written in non-XML technology (in the case at hand, ECMAScript). In what follows, we concentrate on the built-in exporter and, hence, on the direct mapping from Zotero fields to TEI XML structures.

4.2. Data-mapping Decisions

- 20 Given an object that represents the items that should be exported, the translator has to construct the most appropriate output representation. It is therefore essential to know all possible data structures in the source format and their equivalents in the target format. The documentation for Zotero plug-in developers is not explicit about the available data fields in the source database. Nevertheless, as an open source project, Zotero offers information on the data structures in its source code and in the ample selection of available export translators, especially the translators to Zotero RDF and to MODS, which provide good guidance on the availability and handling of the data fields.
- 21 In TEI encoding, it is often possible to represent information in multiple ways. That is because the TEI offers a toolkit which has to be customized, with the particular modeling decisions dependent on the particular use cases. While numerous out-of-the-box TEI customizations exist, in the area addressed here no ready-made solutions are available and each project tends to make its own choices. For the TEI Zotero export translator, encoding decisions have been made at three levels, discussed in the sections that follow: base encoding (section 4.2.1), item-type-specific encoding (section 4.2.2), and item-specific encoding (4.2.3). By fleshing those decisions out for scrutiny, and by offering the translator as a solution employed by the LingSIG bibliography, we hope to take a step toward standardizing the resulting format.

4.2.1. Base Encoding

- 22 The fundamental modeling decision concerning the translator was made at the level of what we call the “base encoding”: the choice among the three possible top-level elements for bibliographic references (`<bibl>`, `<biblStruct>`, and `<biblFull>`). For the purpose of Zotero’s export to TEI, the top-level element `<biblStruct>` is used. In what follows, we justify this choice.
- 23 The element `<bibl>` is a container for any kind of bibliographic reference that features a mixed content model: it may contain a mixture of plain text and elements in any order. Therefore, `<bibl>` is specifically suited for the representation of existing bibliographies (that is, the transcription of physical source documents), but it is not the optimal choice for born-digital bibliographies designed for further processing. For the latter, it is crucial to have unified, predictable encoding. For this purpose, the element `<biblStruct>` was devised. It requires a specific structure and ensures that particular types of information—especially the core information about the author, the place of publication, and the title—are stored at the same location in the structure. The core set of information is structured by bibliographic level: using the element `<monogr>` for the monographic level, `<analytic>` for the analytic level, and `<series>` for the series level. This distinction is particularly useful when it comes to making formatting decisions in XSLT.
- 24 `<biblFull>` is similar to `<biblStruct>` in that it is highly structured, but it follows a different approach: it uses the same content model as `<fileDesc>`, and is thus less

rigid with respect to ordering the relevant information. The more predictable structure of `<biblStruct>` and its advantages for processing were the factors that determined the choice for the base target encoding for the export from Zotero to TEI.

- 25 Bibliographic items are typically arranged in a list-like structure. Consequently, some kind of a structuring device or a container has to be used to hold the individual items. As suggested by the Guidelines, the `<listBibl>` element is used for this purpose in the output of the translator. The base encoding for the Zotero export is therefore a `<listBibl>` containing multiple `<biblStruct>`s.

4.2.2. Item-type-specific Encoding

- 26 The second level concerns the item-type-specific encoding—that is, the way in which the item type for a Zotero item (“journal article”, “book section”, etc.) affects the mapping to the corresponding elements within the `<biblStruct>`. While every item type within the Zotero database features a unique set of properties, many of these properties are shared and the mapping to TEI is the same irrespective of the type. For example, the place of publication will always be mapped to the element `<pubPlace>` within the `<imprint>` part of `<biblStruct>`. Nevertheless, some mappings are affected by the item type: for example, the property `item.title`¹⁵ maps to `<title>` within `<analytic>` for analytic item types such as ‘journal article’ or ‘book section’, and to `<title>` within `<monogr>` for types that do not have an analytic level.
- 27 The first fundamental question at this level of encoding is whether the given item features an analytic level. The TEI Zotero translator defines the item types journal article, book section, magazine article, newspaper article, and conference paper as analytic. While Zotero has a schema that determines which fields may be used for a bibliographic item of a specific type, it does not require the user to enter a minimal amount of data for any item type. In practice, this can lead to situations where it is not possible to meet the minimal requirements for `<biblStruct>`. For the rare cases where no title is given for a bibliographic resource, an empty `<title>` element is generated in `<monogr>` or respectively in `<analytic>`—in other words, the translator remains neutral with respect to apparent omissions in the content of Zotero items and translates them into corresponding empty elements in the TEI markup, thus making them easier to spot in the process of validation.

4.2.3. Item-specific Encoding

- 28 Decisions made at the level of the individual bibliographic items are determined by the values of the Zotero fields for these items. Firstly, as has been mentioned, the TEI Zotero translator depends on which of the available fields are actually filled in by the user. Secondly, for fields that may hold an arbitrary number of individual values, the exporter will handle items differently depending on how many values they have. In particular, the area where Zotero provides great flexibility is the assignment of responsibilities for the creation of the work referenced, and these need to be carefully mapped to TEI.
- 29 In Zotero, any bibliographic item can have an arbitrary number of creators of a particular type. The available creator types are determined by the item type (for example, in Zotero books may have editors while websites do not have editors but rather contributors). Many of the Zotero creator types have direct equivalents in the TEI (for example,

`creator.type` with the value “editor” or the value “seriesEditor” can both be mapped to the element `<editor>`). Nevertheless, this does not apply to all available types (for example, `creator.type` with the value “contributor”). For those creator types that do not map directly to TEI elements, a `<respStmt>` is used with an element `<resp>` that contains the name of the Zotero creator type. Consider the following example:

```
<respStmt>
  <resp>contributor</resp>
  <persName>
    <forename>Kevin</forename>
    <surname>Hawkins</surname>
  </persName>
</respStmt>
```

- 30 The above fragment is the typical choice for the encoding of information about a contributor to a wiki, while the following fragment would be the encoding of information concerning the authorship of the present paper:

```
<author>
  <forename>Piotr</forename>
  <surname>Bański</surname>
</author>
<author>
  <forename>Stefan</forename>
  <surname>Majewski</surname>
</author>
<author>
  <forename>Maik</forename>
  <surname>Stührenberg</surname>
</author>
<author>
  <forename>Antonina</forename>
  <surname>Werthmann</surname>
</author>
```

- 31 This is an example of how the structure of the exported item is determined by the content available within the given data field.

4.3. Output Options

- 32 Apart from the direct representation of the item data, the TEI Zotero translator offers a set of output options. First of all, it optionally generates `@xml:id` attributes for each exported `<biblStruct>`. These IDs are generated from the name of the author, the

year of publication, and if necessary a character for the disambiguation of publications if there is more than one reference per author per year (e.g. “Dipper2005b”). Secondly, the translator can optionally put a simple minimal TEI document around the `<listBibl>` for use cases where a complete TEI file is needed for processing or validation. Finally, since Zotero organizes bibliographic items in collections, it is possible to represent Zotero’s collection structure within the generated TEI. Collections in Zotero can, first of all, nest. Secondly, individual bibliographic items may be put into multiple collections. As `<listBibl>` can nest as well, it is ideally suited to representing Zotero collections. The title of the collection is put in a `<head>` element at the beginning of the `<listBibl>` corresponding to the exported collection.

```
<listBibl>
  <head>Recent Papers</head>
  <listBibl>
    <head>to be read</head>
    <biblStruct>
      ...
    </listBibl>
  ...
</listBibl>
```

- 33 While the TEI Zotero translator is now a mature piece of software, as evidenced by its recent inclusion into the mainstream Zotero distribution, some important functionality, such as import facilities for existing TEI-encoded bibliographies, is still missing. It should be stressed, however, that the translator has been released under an open-source license and is thus open to contributions in the form of code patches, feedback, and general discussion.¹⁶

5. Summary and Conclusions

- 34 The present paper highlights the needs relevant for modern collaborative research practice and, using the example of the TEI LingSIG, shows how Zotero answers many of the demands that such practice creates. The existence of Zotero-to-TEI translation tools further confirms that this is not a random choice, and the fact that the tool described here, the TEI Zotero translator, has been integrated into Zotero testifies to the reception of the ideas presented here by a broader community of developers and users.
- 35 The findings reported here go beyond the confines of the LingSIG for two reasons: its Zotero repository is meant to be usable beyond the SIG and even the TEI community, and the co-operative resource-building strategy recommended here constitutes a feasible blueprint for other open-content and open-source initiatives. Also, the mapping solution used by the translator follows a set of choices that are subject to community acceptance as the potential *de facto* way of creating bibliographies.
- 36 Apart from the matter of acceptance of the Zotero-to-TEI mapping choices, which is an issue to be decided by the TEI community, we have identified some features that Zotero

users would benefit from. One is the need to ensure preservation of Zotero databases via automatic backups, versioning, or the like. It would also be beneficial in some contexts to be able to require a value for some fields, such as the “title” field, possibly by having incomplete citations appear in a shared “waiting room” before they are added to the store as complete references. Being able to restrict and directly manipulate the inventory of tags defined for a particular bibliography store would also help ensure the overall consistency of the database.

- 37 The final issue concerns the definition and implementation of a TEI-to-Zotero mapping (in the other direction). At first glance, it seems reasonable to expect to be able to import `<biblStruct>` objects into Zotero, but more concrete solutions will require needs analysis and further funding.

BIBLIOGRAPHY

Dipper, S. and M. Götze. 2005. “Accessing Heterogeneous Linguistic Data – Generic XML-based Representation and Flexible Visualization”. In *Human Language Technologies as a Challenge for Computer Science and Linguistics: 2nd Language & Technology Conference, April 21–23, 2005: Proceedings*, 206–210, Poznań, Poland: Wydawnictwo Poznańskie.

Dipper, S., M. Götze, U. Küssner, and M. Stede. 2007. “Representing and Querying Standoff XML”. In *Datenstrukturen für linguistische Ressourcen und ihre Anwendungen. Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007*, edited by G. Rehm, A. Witt, and L. Lemnitzer, 337–346, Tübingen, Germany: Gunter Narr Verlag.

Dunlop, D. 1995. “Practical considerations in the use of TEI headers in a large corpus.” *Computers and the Humanities* 29: 85–98.

Ide, N., G. Priest-Dorman, and J. Véronis. 1996. *Corpus Encoding Standard (CES)*. Technical report, Expert Advisory Group on Language Engineering Standards (EAGLES).

Ide, N., P. Bonhomme, and L. Romary. 2000. “XCES: An XML-based Encoding Standard for Linguistic Corpora”. In *Proceedings of the Second International Language Resources and Evaluation (LREC 2000)*, 825–830. Athens: European Language Resources Association (ELRA).

Lezius, W. 2002. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. Ph.D. Thesis, Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart.

Mengel, A. and W. Lezius. 2000. “An XML-based encoding format for syntactically annotated corpora”. In *Proceedings of the Second International Conference on Language Resources and Engineering (LREC 2000)*, 21–126. Athens: European Language Resources Association (ELRA).

Stührenberg, M. 2012. “The TEI and Current Standards for Structuring Linguistic Data: An Overview.” *Journal of the Text Encoding Initiative*, 3.

NOTES

1. http://wiki.tei-c.org/index.php/SIG:TEI_for_Linguists
2. We are grateful to Stuart Yeates for the initial suggestion to use Zotero, made on the LingSIG mailing list. We also wish to acknowledge the pioneer role that the SIG on Education has played by setting up a Zotero repository of TEI-related works at <http://www.zotero.org/groups/tei>. At the time when the LingSIG repository was created, the general TEI repository had barely started, and the two were developed in parallel. Our repository differs in scope, as its primary focus is *linguistic* markup, be it TEI or not. Thus, the two repositories merely overlap to some extent. However, it is worth noting that users who belong to both groups have all the resources at their disposal and can combine them (and automatically detect and merge duplicates) in the user's private Zotero space. It is also worth noting that, unlike the Education SIG's library, which is a unitary resource that can only be searched by string-matching, the LingSIG library features catalog-based and tag-based categories.
3. For a comparison of citation managers, see http://en.wikipedia.org/wiki/Comparison_of_reference_management_software. What played a decisive role in our case is that Zotero is open-source, cross-platform, web-oriented, and extremely flexible.
4. In the presence of a running stand-alone instance, browser add-ons become merely interfaces, or "connectors", between the web content accessed by the browser and the database controlled by the stand-alone Zotero.
5. One shortcoming of Zotero's features for collaboration is the lack of version history and the ease of propagation of errors introduced into the content. That is, if a major maintenance error occurs, as, for example, when one participant accidentally deletes a set of bibliographic items, there is no version history available that could be used to revert the changes. Therefore, frequent manual backups by the project participants are advisable pending an enhancement that targets this issue. On the other hand, Zotero provides the functionality for duplicate detection and merging that is not present in wiki-like resources.
6. MODS is developed by the Library of Congress. See <http://www.loc.gov/standards/mods/mods-schemas.html> for schema files.
7. The import format of the EndNote citation manager is based on the Refer/BibIX format.
8. The LingSIG group at Zotero is accessible at <https://www.zotero.org/groups/tei-lingSIG>.
9. This is not the only possible administrative choice in Zotero groups, but any attempt to limit the write access would run counter to the aims of the entire project, which is to involve as many contributors as we can.
10. See <http://citationstyles.org/citation-style-language/schema/> for the current version of the CSL schema in the RELAX NG notation. Since CSL 1.0, the schema is not only supported by Zotero but by the Mendeley reference manager as well.
11. The Zotero Style Repository is located at <http://www.zotero.org/styles/>. The styles can be used with any client software that supports CSL 1.0.

12. For more information on the *Metadata Object Description Schema*, see <http://www.loc.gov/standards/mods/>.
 13. Zotero RDF is the custom export format of Zotero that can also export attached files and notes.
 14. Laura Mandell's XSL Transformation from Zotero RDF to TEI follows this approach (see <http://wiki.tei-c.org/index.php/ZoteroToTEI>).
 15. Properties of the items as provided by Zotero are used in dot-notation (i.e. `item.property`).
 16. Contributions are welcome via E-Mail to Stefan Majewski or via <https://github.com/smjwsk/translators> or <http://code.google.com/p/tei-zotero-translator/>. The author follows discussions on TEI-L.
-

ABSTRACTS

The present contribution addresses an infrastructural issue of universal relevance, addressed in the specific context of the TEI. We describe a combination of open-source tools and an open-access approach to creating knowledge repositories that have been employed in building a bibliographic reference library for the “TEI for Linguists” special interest group (LingSIG). The authors argue that, for an initiative such as the TEI, it is important to choose open, freely available solutions. If these solutions have the advantage of attracting new users and promoting the initiative itself, so much the better, especially if it is done in a non-committal way: no one using the LingSIG bibliographic repository has to be a member of the LingSIG or a “TEI-er” in general.

INDEX

Keywords: LingSIG, Zotero, structured bibliography, reference management, collaborative workflow

AUTHORS

PIOTR BAŃSKI

Piotr Bański is an assistant professor of Linguistics at the Institute of English Studies of the University of Warsaw, and a senior researcher at the Institut für Deutsche Sprache in Mannheim, where he is the coordinator of the project “Corpus Analysis Platform of the Next Generation”. He is also an elected member of the TEI Technical Council for term 2011–2012 and an expert of the ISO TC37 SC4 committee for Language Resources Management. His current interests focus mostly on text encoding as well as the creation and use of robust language resources.

STEFAN MAJEWSKI

Stefan Majewski studied English Language and Literature as well as Sociology at the University of Vienna, in addition to Electronics at the Vienna University of Technology. He graduated in English Linguistics with a focus on research infrastructures for corpus linguistics. Currently, he is working at the Austrian Academy of Sciences, where he is coordinating and working for the “Data Service Infrastructure for the Social Sciences and Humanities” (DASISH) project. He is also employed by the Göttingen State and University Library, where he works for the “TextGrid” project in research and development. His current interests focus on research infrastructures and annotation systems.

MAIK STÜHRENBERG

Maik Stührenberg received his Ph. D. in Computational Linguistics and Text Technology from Bielefeld University in 2012. After graduating in 2001, he worked on various projects at the Justus-Liebig-Universität Gießen and Bielefeld University. He is currently employed as a research assistant at the Institut für Deutsche Sprache (IDS, Institute for the German Language) in Mannheim as a member of the CLARIN-D project and is involved in the NA 105-00-06 AA, the German mirror committee of ISO TC37 SC4. His main research interests include specifications for structuring multiply annotated data (especially linguistic corpora), query languages, and query processing.

ANTONINA WERTHMANN

Antonina Werthmann studied computational linguistics at Heidelberg University and has been working since 2011 as a research assistant at the Institut für Deutsche Sprache in Mannheim (IDS, Institute for the German Language) as a member of the CLARIN-D project. Her main research tasks consist in the collection, description and extension of the information systems on standards in the field annotation of linguistics resources.