XML: EXTENSIBLE MARKUP LANGUAGE
AN INTRODUCTION TO THE USE OF THE XML METALANGUAGE WITH THE TEI AND CEI SCHEMAS

Daniel Jeller
ICARUS – International Centre for Archival Research, Vienna, Austria

Abstract
This paper is an overview on the whys and hows of the metalanguage XML as well as an introduction to its possible uses and benefits in (digital) humanities. To provide the latter, the well-established Text Encoding Initiative (TEI) language and its offspring Charters Encoding Initiative (CEI) will serve as practical examples for real-life use-cases.

Keywords
XML, Text Encoding Initiative (TEI), Charters Encoding Initiative (CEI), Digital Humanities, text markup

Introduction

In its most basic aspect XML, the eXtensible Markup Language, is a way to add markup, that is, contextual information, directly to a digital text. An example for this is the use of quotation marks and a number with attached footnote to indicate a quotation inserted into a text but authored by a different person. In this way it is possible to add further data to a text and to enable a reader to distinguish between different kinds of information in it. While it is beneficial to add information such as the quotation marks to an actual quote to ease the identification of its different origin, the emergence of computers and digital media made markup not only convenient but a necessity. This is best shown with an example. Consider these two paragraphs:

Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae, aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli apellantur.

This sentence makes up the beginning of the first section of the book of "De bello Gallico" by Caius Iulius Caesar.[1] *It tells of the division of Gallia, a precursor of the modern country France, in three parts and states the names of the people that lived there at the time of writing.*

The first paragraph is a sentence in Latin and the second one is an explanation of its contents in English. While they do not look very similar, they have one thing in common: both paragraphs are made up of characters that are reproduced using a computer. While it is certainly possible to display these characters in various environments, for instance on a computer screen, or in a book produced with a digital printer, the digital system used to create the textual representation does not understand any of its meaning apart from a (possibly already quite sophisticated) spell-checker and the typographic information connected to the visual display and layout of the letters, sentences and paragraphs that make up the actual text. The system does not even understand, at least not in a semiotic sense, what a letter is, apart from how it has to look on screen or when printed. While this is fine for the production of a text only intended to be digested by human readers, for a computer to process this or other data in a meaningful way we have to provide additional information, or *metainformation*. As already stated above, electronic markup like XML is just this, a way to attach metainformation to the text-like data in order to enable a computer to process its contents. The example paragraph encoded in an arbitrary XML markup could look like this:

```
<sentence author="Caius Iulius Caesar" lang="Latin" book="1" paragraph="1">
    <place modern="France">Gallia</place> est omnis divisa in partes tres, quarum unam incolunt <name>Belgae</name>, aliam <name>Aquitani</name>, tertiam qui ipsorum lingua <name>Celtae</name>, nostra <name>Galli</name> apellantur.
</sentence>
```

1 Caesar, Gaius Iulius. De bello Gallico. Der Gallische Krieg: Lateinisch/Deutsch. Stuttgart: Reclam, 1991. P. 4.

Using so-called XML *elements*, denoted by markup tags contained within angle brackets that describe or encode aspects of the original text, the natural language explanation given at the beginning of this section can be embedded in a way that enriches the original text. This makes it easy for a human reader to understand as well as simple to process for any system that knows how to handle XML code.[2] The structured syntax enables a computer program to present the text in a typographic layout that takes the structure of the books and paragraphs of the original into consideration, and (as an example) to prepare an index for all places and names. Additionally, if further texts by Caesar and other authors were to be combined in a database, a user would be able to get a list of all texts by the same author or, for instance, get a list of all texts in the database that mention the ancient *Belgae* tribe.

Considering the usefulness of this approach, a number of different methods have historically been developed to reach the goal of a simple yet powerful way to encode the text in an electronic way. One of the first and most adaptable has been the development of SGML, the *Standard Generalized Markup Language* in 1986.[3] Contrary to what the "L" in its abbreviation suggests (and XML's for that matter), SGML is not a markup language by itself, but rather a way to create a markup language that is tailored to the individual needs of its creators yet based on standardised structural elements.[4] With HTML, the *Hypertext Markup Language*,[5] being one of the building blocks of the Internet and TEI bring the markup language of the *Text Encoding Initiative*,[6] these two SGML-based languages are still widely used today. Unfortunately, although SGML is very powerful and was created by two influential partners, the *American National Standards Institute* (ANSI) and the *International Organization for Standardization* (ISO),[7] for various reasons it failed to get widely accepted "outside of selected industries and large organizations".[8] XML was developed after it became clear that

2 Actually, the computer still doesn't *understand* (in a human sense) what the information means, but it can be told how to process it in a meaningful way.

3 Renear, Allen H. Text Encoding. // A Companion to Digital Humanities / edited by Susan Schreibman, Ray Siemens and John Unsworth. Oxford: Blackwell Publishing, 2004. P. History of SGML and XML: Part I.

4 Markup Systems and the Future of Scholarly Text Processing / ed. by James H. Coombs, Allen H. Renear and Steven J. DeRose, 2005 [cited: 2013-09-27]. Available at: http://xml.coverpages.org/coombs.html, p. Document Portability.

5 Renear, A. H. Op. cit., History of SGML and XML: Part II.

6 Ibid., Origins.

7 Ibid., History of SGML and XML: Part I.

8 Ibid., History of SGML and XML: Part II.

SGML was too complex to be used in a broad range of contexts[9] and that its offspring HTML lacked the structural elements necessary for "[w]eb publishing [...] to achieve its promise".[10]

eXtensible Markup Language

The eXtensible Markup Language (XML) is a "subset of SGML [...]. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML."[11]

Because XML is based on SGML it also is not a markup language, but rather it is a metalanguage that can be used to create different, possibly connected markup languages with a fixed set of simple rules. It consists primarily of two parts:

1. The basic entities from which any XML file is built. The most important ones are: elements, element attributes as well as character data, the actual text marked up with the elements and attributes. XML elements can be nested into each other to create documents with a hierarchical structure similar to books with chapters, sections, paragraphs, sentences and so on.
2. An underlying schema that can (but does not have to) exist in the form of one or more files that work both as a guide to create instances of this schema and as a means to control whether or not the instances conform to the schema. It can be compared to the construction plan of a car with its various parts that may look different in different brands of cars but that share the same functional features as all other cars.

These two elements are used to create the actual XML files, which are called *instances* of their abstract schema. The resulting files can be processed by a category of software called an XML parser, for example *Expat*, the parser used in the web browser *Mozilla Firefox*.[12]

9 Ibid.

10 Ibid.

11 Extensible Markup Language (XML) 1.0 (Fifth Edition): W3C Recommendation 26 November 2008 / ed. by Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler and François Yegeau, 2013 [cited: 2013-09-28]. Available at: http://www.w3.org/TR/REC-xml. P. Abstract.

12 Mozilla Firefox. https://developer.mozilla.org/en-US/docs/XML_in_Mozilla / ed. by Mozilla Developer Network, 2013 [cited: 2013-09-30]. Available at: https://developer.mozilla.org/en-US/docs/XML_in_Mozilla. P. Supported Core XML W3C Recommendations.

The following snippet is another example of valid but arbitrary[13] XML code.

```
<author>
        <name type=”first”>Daniel</name>
        <name type=”last”>Jeller</name>
</author>
```

It consists of a *root element* called <*author*> with two nested *child elements*. The individual elements start and end locations are marked by so called *tags* that are built from angular brackets, their literal name, and, in the case of the closing tag, a forward slash. The two child elements in the above example are both called *“name”* but differ from each other in the literal value of an attribute called *“type”* and specified by an equals sign, and the value in double quotation marks. Both *“name”* elements only have simple character data as contents although they would be allowed to contain any number of additional nested child elements.

This short example, which by no means uses all features of XML, shows one of the advantages of this kind of metalanguage: the markup is not encrypted in any way, so, if chosen sensibly, it enables a human reader to grasp some of the basic meanings of the marked up text. At the same time the structure of the markup is strictly fixed by the basic rules of XML so a computer is able to work with the text because it can, for instance, look for tags with a specific name or attribute for further processing. This enables a wide range of applications, some of which will be outlined later in this paper.

One of the main advantages from a user’s point of view is clearly visible in the above example: even if the text is encoded in a way that enables a computer to process the marked up data, the markup itself stays legible for everybody. There remains a problem, though. Without a schema that not only formulates the technical structure of the dialect but also explains what this structure is supposed to mean, the XML can remain technically valid but ambiguous or even wrong content-wise. Consider the following three elements.

```
<title>De bello Gallico</title>
<title>Doctor</title>
<title>Casablanca</title>
```

13 This means it does not have an attached schema file.

All three elements are called "title" but their content refers to possibly different kinds of entities. "De bello Gallico" is both the title of a book and the name of the war the book is about. The second element probably refers to the academic title "Doctor", but could also mean the medical profession or, for instance, a character in a book or movie called the "Doctor". "Casablanca" is the name of a city in Morocco and the title of a movie set in this city. In addition, it is also the name of a brand of cigarettes in Austria. Without further context it is not clear which kind of information an element is supposed to contain or whether content fits another element of the same type (Casablanca, the city, would better fit into a "name" element than a "title"). The declaration of the use of an XML schema can provide this kind of semantic interconnection while at the same time specifying the hierarchical structure that fits the individual nature of the marked up text. It would be possible, for instance, to create a schema that is specifically tailored to storing information about movies. This hypothetical schema could contain a range of possible elements, title being one of them, probably accompanied by other elements for the director, year of production, film studio, producer, actors and so on. Additionally, the schema would fix the order and number of individual elements. A movie has only one title after all[14] but multiple actors. In Figure 1 you can find a schematic display of a simple movies schema.

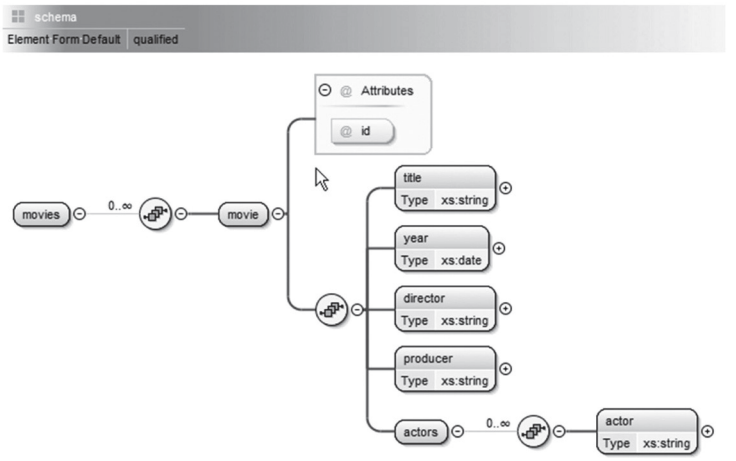

Figure 1.
Graphical representation of an XML schema for movies

14 At least if we are not considering for example translations of the title for different countries/languages.

The following sections of this paper will use focus on the use of these features applied to the creation of specialised XML vocabularies (as schemas are also called) for different fields of the digital humanities.

Text Encoding Initiative (TEI)

"The practice of creating machine-readable texts to support humanities research began early and grew rapidly."[15] It started soon after the development of the first machines that were able to process text and at the time the SGML was created there already was a significant amount of "needlessly diverse and often poorly designed encoding systems [that] threatened to block the development of the full potential of computers to support humanities research".[16] In an effort to improve this situation, sponsored by three American scientific institutions, 1987 saw several expert committees starting to work on what would be published in 1993, after a period of drafts and revisions, as the first official version of the TEI Guidelines.[17] As of 2013, P5, the latest version, released in 2007,[18] is "widely used by libraries, museums, publishers, and individual scholars to present texts for online research, teaching, and preservation".[19]

With its schema, the TEI consortium thus created a tool to take in a broad range of texts and encode them using a solid set of elements describing most general aspects of their internal and external structure. Our example from Caesar's "De bello Gallico", encoded in TEI could look like this:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader type="text">
    <fileDesc>
      <titleStmt>
        <title type="work" n="Gal." ref="http://dbpedia.org/page/
Commentarii_de_Bello_Gallico"
        xml:lang="lat">De bello Gallico</title>
      </titleStmt>
       <publicationStmt>
```

15 Renear, A. H. Op. cit., Background.
16 Ibid., Origins.
17 Ibid.
18 TEI: P5 Guidelines / ed. by Lou Burnard and Syd Bauman, 2013 [cited: 2013-10-01]. Available at: http://www.tei-c.org/Guidelines/P5/
19 TEI: Text Encoding Initiative / ed. by Text Encoding Initiative Consortium, 2013 [cited: 2013-10-01]. Available at: http://www.tei-c.org/index.xml

```
        <date>Around 50 C.E.</date>
      </publicationStmt>
      <sourceDesc>
        <bibl>
          <author xml:lang="lat" ref="http://dbpedia.org/page/Julius_Caesar">C. Julius
          Caesar</author>
          <title xml:lang="lat">C. Iuli Commentarii Rerum in Gallia Gestarum Gestarum VII A.
          Hirti Commentarius VII</title>
          <lang>Latin</lang>
        </bibl>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
 <text xml:lang="lat">
  <body>
    <div1 type="Book" n="1" org="uniform" sample="complete">
      <head>COMMENTARIUS PRIMUS</head>
        <p>
          <milestone n="1" unit="chapter"/><milestone n="1" unit="section"/><region
          ref="http://dbpedia.org/page/Gaul">Gallia</region> est omnis divisa in partes
          tres, quarum unam incolunt <orgName ref="http://dbpedia.org/page/Belgae">
          Belgae</orgName>, aliam <orgName ref="http://dbpedia.org/page/Aquitani">
          Aquitani</orgName>, tertiam qui ipsorum lingua <orgName
          ref="http://dbpedia.org/page/Celts" xml:lang="cel">Celtae</orgName>, nostra
          <orgName ref="http://dbpedia.org/page/Celts">Galli</orgName> appellantur. [...]
        </p>
      </div1>
    </body>
  </text>
</TEI>
```

As you can see, the encoded text is made up of two distinct parts:

- A first section called "teiHeader" with information about the encoded text.
- A second part called "text": It contains the actual words written by Caesar as well as some structural information about books, chapters and sections, distinguishing between headline and a paragraph. The names of people and regions are encoded using the appropriate TEI elements and additionally enriched with references to the linked open data version of the Wikipedia, DBpedia.[20] It is especially noteworthy that both "Celtae" and "Galli" refer to the same entry about the ancient Celts while only "Celtae" is marked as in the language "cel" which denotes the Celtic language group.

This data enables storage and processing by computer software. The document could, for instance, be presented in an online database where users could search for specific terms, follow the links to the encoded documents that are prepared for online presentation using a so-called style-sheet that enables the transformation from one XML format into another. For example, using a simple transformation from the TEI schema into a HTML document to be presented online, the first paragraph by Caesar looks like the following screenshot in Figure 2:

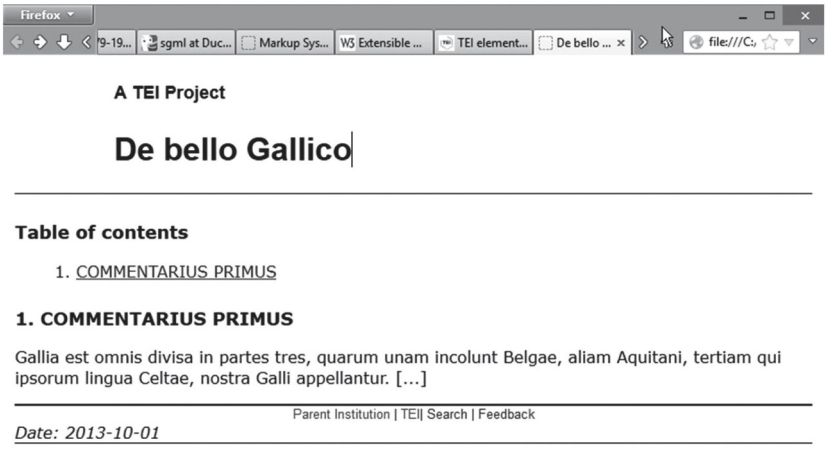

Figure 2.
Screenshot of an XHTML Representation of a TEI document

20 wiki.dbpedia.org: About / ed. by Christopher Sahnwaldt, 2013 [cited: 2014-03-28]. Available at: http://dbpedia.org/About

In addition, the text could be transformed into a PDF document to be downloaded and printed out by users, or it could be transformed into another XML schema and combined with a different document to create a completely new output in a new environment. Also, using the embedded links, the document can be put into context with other documents containing information about the same entities (like the DBPedia Link for Caesar above).

Similar results can easily be achieved for plays, poems, history texts, novels, law texts and so on under the condition that the editors find a way to fit their text into the possibilities of the TEI schema. If the original data fits poorly, however, for instance if very specific information needs to be encoded that has no equivalent in the TEI, the encoder is free to draw upon XML to extend the schema to account for the specific necessities of the source material at hand. An example for this is the CEI.

Charters Encoding Initiative (CEI)

Medieval charters,[21] historical documents that were used to record legal acts, are among the oldest and most complex items in European historical archives. As the central object of the historical science of diplomatics, scholarly interest in them reaches as far back as to the seventeenth century.[22] Because of the highly evolved scientific method in their field, diplomatics scholars were tempted right from the start of the information age to use the new technologies at hand to further improve their toolset to work with and on their source material.[23] Nonetheless it took until after the turn of the millennium for the first XML-based charter projects to appear. Georg Vogeler, one of the contributors to the CEI: The Charters Encoding Initiative, describes three possible methods[24] to encode charters:

21 For an image of one, see Figure 3 below.

22 Hartmann, Josef. Urkunden. // Die archivalischen Quellen: Mit einer Einführung in die Historischen Hilfswissenschaften / edited by Friedrich Beck and Eckart Henning. 5th ed. Köln, Wien [u.a.]: Böhlau, 2012. Pp. 25-54, p. 35.

23 For example Borsa, Iván. A Magyar Országos Levéltár Diplomatikai Levéltára egyes adatainak gépi segítséggel történő feldolgozása. // Levéltári Közlemények 42 (1971), 3-32.

24 Vogeler, Georg. Charters Encoding Initiative (CEI): Zu Möglichkeiten der Integration mit Hilfe eines Standards für Urkundendigitalisierung. // Alte Archive - Neue Technologien / Old Archives - New Technologies / edited by Thomas Aigner and Karin Winter. St. Pölten: DASP - Bischöfliches Ordinariat St. Pölten Diözesanarchiv, 2006. Pp. 182-198.

- Use of the EAD (Encoded Archival Description)[25] schema that is focussed on the administrative facets of charters and whose eadHeader was modeled on the teiHeader element. However, its feature-set is too narrow to be of general use for the work with charters.[26]
- Application of TEI. According to the features of this standard, TEI is best suited for the treatment of a charter as a text. Unfortunately, difficulties in describing external features of the charters seems to limit its use for this type of historical source material.[27]
- A schema specifically created for a project. This approach adds to the already described fragmentation of the field and its resources.[28]

The CEI, launched in 2004[29], focussed on the same approach as the TEI before it, and aimed at creating a unified way to encode medieval charters. It was felt that this new standard should combine some of the features developed by independent projects while at the same time retain the tried and tested structures of the TEI to describe textual content, so that a system already familiar to many encoders could be used.

The following code snippet[30] is an extract from a charter encoded in CEI:

```
<witnessOrig>
  <traditioForm>orig.</traditioForm>
    <archIdentifier>
      <arch>Archive of the Monastery of Schlägl (www.stift-schlaegl.
at)</arch>
  </archIdentifier>
  <physicalDesc>
    <material>Parchment</material>
    <dimensions>90/95 x 200 mm, no plica.</dimensions>
  </physicalDesc>
  <auth>
```

25 EAD: Encoding Archival Description, http://www.loc.gov/ead/
26 Vogeler, Georg. Charters Encoding Initiative (CEI): Zu Möglichkeiten der Integration mit Hilfe eines Standards für Urkundendigitalisierung. // Alte Archive - Neue Technologien / Old Archives - New Technologies / edited by Thomas Aigner and Karin Winter. St. Pölten: DASP - Bischöfliches Ordinariat St. Pölten Diözesanarchiv, 2006. Pp. 182-198, p. 185.
27 Ibid., pp. 187-188.
28 Ibid., pp. 189-190.
29 Ibid., p. 182.
30 The text is originally in German.

```
    <sealDesc>
      Seal hanging on parchment leaflet: Rüdiger of Haichenbach.
      Brown, circular seal 33/35 mm diam; Coat of arms with tricorn.
      Corners crumbled. Circumscription: "... DEGERI ...
HAICHENPACH" (barely readable)
    </sealDesc>
  </auth>
  <nota>
    Back: 1274 / Super mülholtz cum / suis attinentiis / N.I. / A: Clas.
III Jura Rub. c / Sig: 2.
  </nota>
</witnessOrig>
```

This example represents the description of the object itself. This is probably the area with the biggest differences from the TEI, which is, as already stated, mainly concerned with the textual content of a document. As is shown by this example, the complexity of the marked up

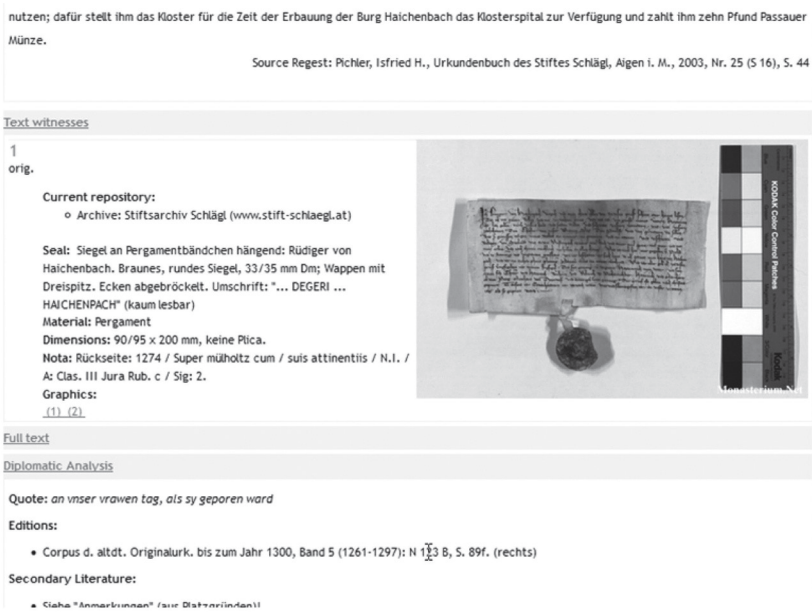

Figure 3.
Screenshot of an XHTML Version of a CEI document (from the Monasterium Database)

content can become very high, and the resulting XML document might look complicated, but it enables a broad range of technical and scholarly applications. Applying an appropriate style-sheet, an online representation can get much easier to digest. The following screenshot from the Monasterium.net,[31] one of the first projects to use CEI, illustrates this. It is created from the above code snippets with an added image of the charter (Figure 3).

By using XML as a basis for the data stored in the Monasterium database and by combining it with modern web technologies, a highly usable online representation for the complex matter of medieval charters can be achieved.

Conclusion

While developed out of dissatisfaction with the existing technologies in the field for marking up texts electronically, XML fared better than its predecessors and is widely used today. It is not only applied in its most basic form--as arbitrarily developed code--but via shared vocabularies that enable researchers to exchange data and work collaboratively in an effective way. This makes XML, be it in the form of TEI, CEI or any other schema, as long as it fits the source material, a very flexible and powerful tool in any field of the digital humanities.

References

Borsa, Iván. A Magyar Országos Levéltár Diplomatikai Levéltára egyes adatainak gépi segítséggel történő feldolgozása. // Levéltári Közlemények 42(1971), 3-32.

Caesar, Gaius Iulius. De bello Gallico. Der Gallische Krieg: Lateinisch/Deutsch. Stuttgart: Reclam, 1991.

Extensible Markup Language (XML) 1.0 (Fifth Edition): W3C Recommendation 26 November 2008 / ed. by Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler and François Yegeau, 2013 [cited: 2013-09-28]. Available at: http://www.w3.org/TR/REC-xml

Hartmann, Josef. Urkunden // Die archivalischen Quellen: Mit einer Einführung in die Historischen Hilfswissenschaften. 5th ed. / edited by Friedrich Beck and Eckart Henning. Köln, Wien [u.a.]: Böhlau, 2012. Pp. 25-54.

Markup Systems and the Future of Scholarly Text Processing / ed. by James H. Coombs, Allen H. Renear and Steven J. DeRose, 2005 [cited: 2013-09-27]. Available at: http://xml.coverpages.org/coombs.html

31 Monasterium.net [cited: 2014-03-28]. Available at: http://icar-us.eu/cooperation/online-portals/monasterium-net/

Monasterium.net [cited: 2014-03-28]. Available at: http://icar-us.eu/cooperation/online-portals/monasterium-net/

Mozilla Firefox. https://developer.mozilla.org/en-US/docs/XML_in_Mozilla / ed. by Mozilla Developer Network, 2013 [cited: 2013-09-30]. Available at: https://developer.mozilla.org/en-US/docs/XML_in_Mozilla

Renear, Allen H. Text Encoding // A Companion to Digital Humanities / edited by Susan Schreibman, Ray Siemens and John Unsworth. Oxford: Blackwell Publishing, 2004.

TEI: P5 Guidelines / ed. by Lou Burnard and Syd Bauman, 2013 [cited: 2013-10-01]. Available at: http://www.tei-c.org/Guidelines/P5/

TEI: Text Encoding Initiative / ed. by Text Encoding Initiative Consortium, 2013 [cited: 2013-10-01]. Available at: http://www.tei-c.org/index.xml

Vogeler, Georg. Charters Encoding Initiative (CEI): Zu Möglichkeiten der Integration mit Hilfe eines Standards für Urkundendigitalisierung. // Alte Archive - Neue Technologien / Old Archives - New Technologies / edited by Thomas Aigner and Karin Winter. St. Pölten: DASP - Bischöfliches Ordinariat St. Pölten Diözesanarchiv, 2006. Pp. 182-198.

wiki.dbpedia.org : About / ed. by Christopher Sahnwaldt, 2013 [cited: 2014-03-28]. Available at: http://dbpedia.org/About

XML: EXTENSIBLE MARKUP LANGUAGE
UVOD U UPORABU METAJEZIKA XML SA SHEMAMA TEI (TEXT ENCODING INITIATIVE) I CEI (CHARTERS ENCODING INITIATIVE)

Sažetak

Rad predstavlja prikaz razloga i načina uporabe metajezika XML, ali i uvod u moguću uporabu i korist koju (digitalna) humanistika može imati od tog metajezika. Na primjeru stvarnih situacija korištenja shema TEI (Text Encoding Initiative) i CEI (Charters Encoding Initiative) prikazuje se uporaba metajezika XML.

Ključne riječi: XML, Text Encoding Initiative (TEI), Charters Encoding Initiative (CEI), digitalna humanistika, označivanje teksta