Originally published in: Flanders, Julia/Jannidis, Fotis (Eds.): The shape of data in digital humanities. Modeling texts and text-based resources. - London [et al.]: Routledge, 2019. Pp. 217-235. (Digital Research in the Arts and Humanities)

Piotr Bański
Andreas Witt

Modeling and annotating complex data structures

1 Introduction

Although it is possible to associate an unlimited number of arbitrary, complex layers of annotations with a text, an image, or an audio/video file, the most common applications almost always follow the classical approach: additional information associated with primary data is expressed in an ordered hierarchy, using a tree structure as its underlying data model.

The present contribution offers a brief review of the more popular ways of data structuring and highlights some of the problems that each of them is meant to handle. The first part of the present chapter focuses on the most relevant issues of data modeling for researchers in the humanities and reviews the basic kinds of the relevant data models. The second part addresses ways to capture these abstract models in concrete encoding formats available to digital humanists. We focus here on approaches that use XML, but the models can also be applied more generally.

Information and communication are tightly related: communication relies on the exchange of information, but just as the individual information containers are determined by many kinds of variables, organizing these containers into higher level structures is vital for ensuring success in transmitting complete and compact messages. Finding the appropriate level of complexity for the structuring of information is one of the key problems in the field of digital humanities. Simple information packages are quick to set up, process and visualize, but as the individual fields of study develop, more and more information needs to be accommodated within a vertically tight space of electronic documents.[1] Packaging of complex information raises new theoretical questions and demands new, more efficient, technological solutions.

For the purpose of an introductory example, let us assume that the "information containers" are words, subject to the choice of the natural language but also, on the technological plane, to, for example, the selection of the character encoding, such as ISO 8859-1 (known as "Latin-1") or Unicode. These words are grouped into larger units: phrases, sentences, or utterances. The structure of these larger units, on the one hand, is dictated by the internal syntactic rules of the given language but, on the other, it is also modeled technologically by the selection of

the given XML document grammar (schema). Words may also have semantic or presentational properties of interest: they can be highlighted in various ways (italicized, struck out, capitalized, etc.), or they can be linguistically distinctive—for example, Latin intrusions in an English text—and we may even want to encode the information on where they can be split if they would otherwise exceed the page margins. Longer sequences of words can be formatted as section headers or may be split into enumerated lists, grouped into paragraphs, sections, and so on.

Digital humanists are information architects nearly by definition. Apart from the language- or text-internal features mentioned above, they might decide to accommodate more information within their markup. In the example at hand, this may mean further grammatical or prosodic features of individual words or word groups, or it may be information that fleshes out the relationships between words or their roles in the discourse or in verse. Finally, the encoder may wish not to privilege a single theoretical model but to record instead the potentially *conflicting* information provided by very different theoretical approaches—after all, examination of the discrepancies among the predictions made by different theories may also be of value, both in research and in teaching.

Nowadays, the reasons for adding information can be manifold: scholarly, didactic, practical, with the eventual results typically performing more than one function. However, this has not always been the case. In the very early days of markup standards in the 1970s and 1980s, there was mainly only one reason for enriching text with additional information: to provide technical means to enhance the typesetting process. These origins have strongly influenced the structure of all formalisms used to add information to primary texts and have led to an essentially linear and sequential approach to information structuring. This linear focus arose even though it was noted early that books contained many techniques (such as cross-references) for escaping the linear structure of a text, and even though scholars had long been aware that a focus on a linear structure is neither intellectually nor technologically adequate. The literature on the so-called "OHCO thesis," which initially stated that text is an "Ordered Hierarchy of Content Objects," illustrates the gradually developing acceptance of the fact that tree-based structures can only offer handy approximations for modeling *selected views* of the text.[2]

The additional data items containing analytical information about text are typically referred to as *annotations*. In the context of the technology assumed here and centering on XML and its ilk, annotations are typically realized as *markup*—that is, special markers (tags) directly or indirectly added to the text stream.

In the remainder of the present chapter, we provide an overview of modeling approaches to annotation (section 2) and then, in section 3, we review selected ways of implementing the abstract conceptual structures, focusing mostly on techniques that make use of XML, which is still the most popular markup formalism used in digital humanities. Our examples come mostly from the linguistic domain, but they straightforwardly translate into other domains and applications.

2 Modeling complex information

Various definitions of complex documents can be offered depending on the individual bias and the research angle. Our classification of complexity is based on a range of initiatives focusing mostly on linguistic aspects of document modeling and interpretation, but not confined to these aspects alone. This is because there exists no clear division between linguistic markup and "digital humanities markup," not only because linguistic considerations at large are part of the digital humanities landscape, but also because of wide areas of overlap among the various subdisciplines of digital humanities and broadly understood linguistics, involving, for example, discourse structure or meter.

We distinguish among three major types of complexity that commonly occur in annotated data. We shall refer to these arrangements of data as "linear," "complex," and "concurrent." An additional factor that complicates the picture is the nature of what precisely gets annotated—annotations can be constructed over a single data stream or over multiple parallel streams of data—we shall refer to such construals as "single-stream" and "multi-stream," respectively. Below, we provide a brief overview of the above-mentioned data arrangements. Sections 2.1 through 2.3 assume a single-stream arrangement, while section 2.4 adds one more dimension, introducing "horizontal" alignments between data streams and/or annotations built over them.

2.1 Linear arrangement

Linear arrangement is the simplest case out of those that are of interest to us here. It concerns a single layer of objects that exhaustively or partially cover the data stream.

Such an arrangement of objects is practically unproblematic for any approach. However, the complexity that we wish to recognize in this case occurs when more than one information package is attached to a single textual object. While various formal classifications can be provided for such an arrangement, for the purpose of our discussion, we shall continue to use the term "linear" when referencing this kind of complexity.

Figure 2 shows the same text annotated with two kinds of information, added, for example, by two different annotation tools.[3] That this is not exotic at

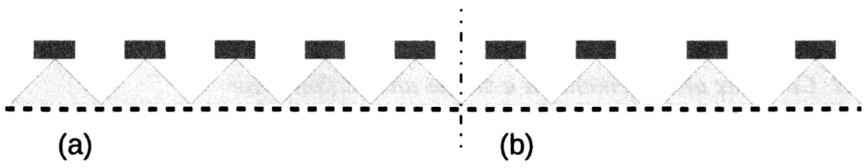

Figure 1 Exhaustive (a) and non-exhaustive (b) coverage of a data stream by a linear arrangement of annotation objects. The dashed line symbolizes a sequence of characters.

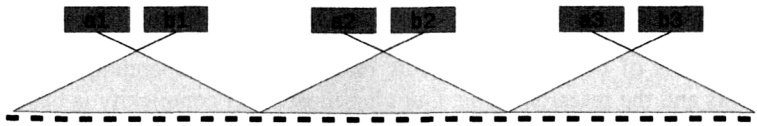

Figure 2 Multiple information packages attached to single elements of the document: series "a" is provided by one tool (or one human annotator), and series "b" by another.

Partei (IKP) wurde im März 1934 **gegründet** und ist heute eine der größten

Layer	-- Foundry	Die	Irakische	Kommunistische	Partei	IKP	wurde	im	März	1934	gegründet	und
d	xip										VMAIN	
l	cnx		irakisch	Kommunistische	partei	IKP	werden		märz	1934	gegründet	und
	mate		irakisch	kommunistisch	partei	ikp	werden	in	märz	1934	gründen	und
	tt		Irakische irakisch	Kommunistische kommunistisch	Partei	ikp	werden	im	März	1934	gründen	und
	xip		irakisch	kommunistisch	Partei	IKP	werden	in	März	1934	gründen	und
m	cnx			Prop			IND PAST					
	mate		case:nom degree:pos gender:fem number:sg	case:nom degree:pos gender:fem number:sg	case:nom gender:fem number:sg	case:nom gender:fem number:sg	mood:ind number:sg person:3 tense:past	case:dat gender:masc number:sg	case:dat gender:masc number:sg			
ne_dewac_175m_600	corenlp		I-ORG	I-ORG	I-ORG	I-ORG						
ne_hgc_175m_600	corenlp		I-ORG	I-ORG	I-ORG	I-ORG						
p	cnx		A	N	N	N	V		N	NUM	A	CC
	mate		ADJA	ADJA	NN	NE	VAFIN	APPRART	NN	CARD	VVPP	KON
	opennlp		ADJA	ADJA	NN	NE	VAFIN	APPRART	NN	CARD	VVPP	KON
	tt		ADJA NN	ADJA NN	NN	NE NN	VAFIN	APPRART	NN	CARD	VVPP	KON
	xip		ADJ	ADJ	NOUN	NOUN	VERB	PREP	NOUN	NUM	VERB	CONJ
syn	cnx		@PREMOD	@PREMOD	@NH	@NH	@MAIN		@NH	@PREMOD	@NH	@CC

Irakische Kommunistische Partei by Jorgen, a; published on 2005-03-28 as III.02914 (WPD)

Figure 3 A fragment of KorAP annotations, visualized in the form of a table. Among other pieces of information, the table shows various part-of-speech tags (visible in row "p") supplied by different tools. "cnx," "mate," "opennlp," "tt," and "xip" are names of the particular foundries (that is, sets of annotation layers, cf. section 3 below).

all can be demonstrated with an example from the KorAP system (Bański et al., 2013), which organizes annotations into so-called foundries, to which we return in section 3.

At the level of conceptual modeling, we do not establish *how* the information packages are added to the text. They may end up enclosing each relevant text fragment or they may merely point at it. In the latter case, the identity of each span in a simple linear arrangement can be established by comparing the character offsets of the beginning and/or end of the span. However, it is often useful to provide a level of indirection between annotations and text, by including a unique identifier (ID) as the basic part of each annotation package.

2.2 Complex arrangement on a single annotation layer

By a complex arrangement, we understand one that has to do with hierarchical structures, or with what is commonly known as *relational* or *dependency graphs*. Consequently, in order to provide a gross taxonomy for the purpose of discussion in the present chapter, we divide complex arrangements into hierarchical (see section 2.2.1), relation-based (s. 2.2.2), and mixed (s. 2.2.3).

2.2.1 Hierarchical structures

Hierarchical structures arrange abstract nodes into trees (technically, a restricted type of directed acyclic graphs) by means of relations of dominance and precedence. In the figure below, the node labeled "S" is the root—it dominates all other nodes; the node labeled "NP" precedes the node labeled "VP" (as well as, indirectly, the terminal nodes V and N that the VP dominates). In trees, each node other than the root node must be dominated by at most a single node (in other words, while a node can have multiple children nodes, it can only have one mother node).

In Figure 4, the information package in each case has to minimally contain the grammatical label ("NP" for noun phrase, etc.), and the ordered reference(s) to the items that it dominates. Note that in an ID-based system, the references target IDs, and therefore an ID must be included in the information package. An alternative is to use the offsets of the beginning and the end of the fragment of base text to which the given annotation refers. In such a case, an additional mechanism is needed to properly arrange the last NP and N, because they address the same spans.

We use a syntactic tree diagram in order to illustrate this kind of arrangement, but in fact arrangements of this type are ubiquitous: HTML, TEI, and more generally, all SGML- and XML-based formats are based on this kind of tree-based hierarchy, even if in actual practice the hierarchy is enriched with additional non-hierarchical devices, to which we now turn.

2.2.2 Relational arrangements

Relational arrangements of the simplest kind typically focus on relationships among the objects identified in linear arrangements (cf. section 2.1). As an example of simple structures of that sort, we use so-called grammatical dependency relationships. Information packages in such cases typically contain the name of the relation (e.g., "subject of," etc.) and an ordered sequence of pointers at the elements of that relation.

In Figure 5, the middle element (most commonly a verb) points to its grammatical subject on the left, and the grammatical object on the right.

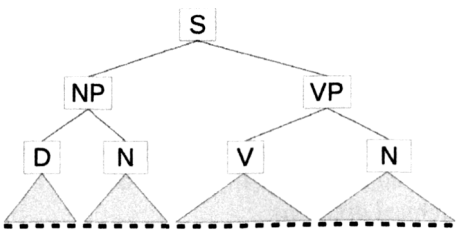

Figure 4 An example of a hierarchical annotation (syntactic tree).

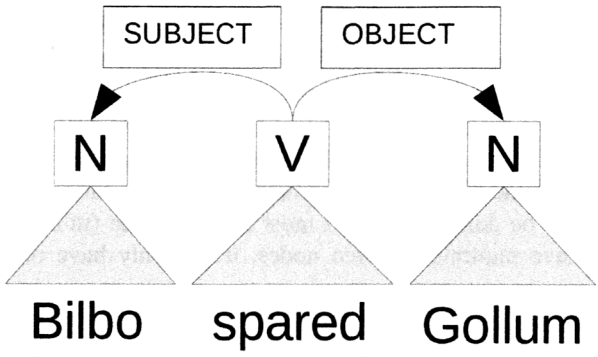

Figure 5 Dependency relationships among elements of the sentence (as an example, assume a three-word sentence such as "Bilbo spared Gollum").

The information encoded here identifies the source and the goal of the grammatical relation, together with its name.

Note that at the conceptual modeling stage, we do not determine how exactly the relevant information is encoded. In more concrete terms, it may be placed either together with the individual text fragments, or it may point at them remotely—this depends both on the choice of the concrete realization (e.g., XML or a set of RDF triples), and on the type of approach (in XML, it can be represented as local inline markup, or as remote standoff markup).

2.2.3 Mixed complex arrangements

It is not uncommon in syntactic trees to find long-distance relations linking terminal nodes with more or less remote non-terminal nodes. In order to build on the examples used above, we present a case where dependency information is mapped onto hierarchical structure. This example features also a more complex case: an anaphoric relationship among parts of the tree structure: a word-sized element (the reflexive pronoun "himself") referencing a constituent (the noun phrase "the boy") *across* the tree structure.[4]

In Figure 6, first, the two grammatical relationships from Figure 5 have been mapped onto the corresponding branches of the tree (other kinds of information, identifying various kinds of modifiers and heads of phrases, can be added by analogy). Second, the long-distance anaphoric relationship is marked by a so-called "secondary edge" (dotted arrow) that indicates additional information that relates constituents across the tree structure, in this case expressing the information that the reference of the reflexive "himself" depends on the reference of the noun phrase "the boy"; such relationships are naturally free to occur across sentence boundaries or any other hierarchical divisions.

The mixed arrangement is notorious in digital representations, which are typically based on hierarchical data models of HTML or XML, and use *links*

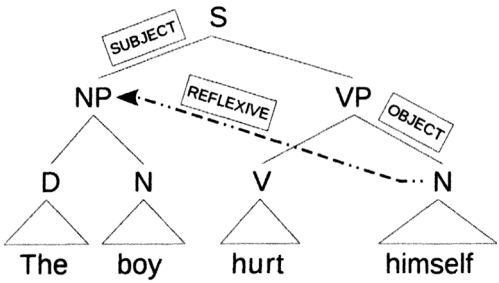

Figure 6 An example of a hierarchical relationship mixed with relational information of two kinds.

(or more generally, *references*), in order to cope with the constraints imposed by the tree structure. Naturally, such references are not restricted to pointing at fragments of the current document and can also point outside of it (in which cases we talk about *external links*).

2.3 Concurrent arrangement (multiple annotation layers)

Concurrent arrangement is found where two or more annotation layers are built on a single stream of data. The layers may then differ in how they order the data described (i.e., in the structures assigned to the data) or the structures may be identical but differ in the content. The former case is exemplified in Figure 7, where two different tokenization (segmentation) structures are assigned to the same sequence of characters.

In Figure 7, lexicalized expressions such as "mother-in-law" may well be tokenized as a single element. However, many linguistic tools will treat them as sequences of three (or even five) tokens that may be relinked at another stage of annotation. Multi-word idioms, numbers or names fall under this pattern as well.

The latter case is exemplified in Figure 8, which shows identical tree structures with different labels attached to them, corresponding to two possible and equally likely analyses.

While for ease of illustration, the example in Figure 8 invokes a linguistic structure of an obvious structural ambiguity ("They are making planes fly" vs. "These are planes that fly"), the principle goes beyond ambiguities: we may be

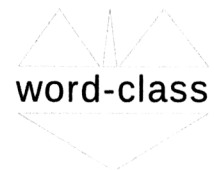

Figure 7 Concurrent tokenizations of a single compound noun.

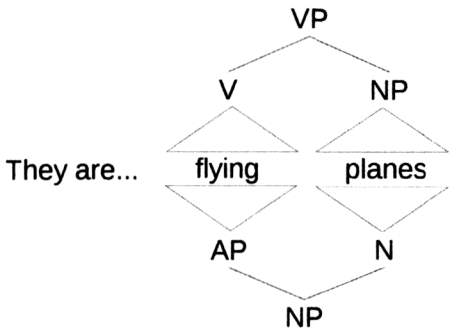

Figure 8 Examples of concurrent analyses of a single tokenized stream.

looking at competing syntactic analyses or even competing structural divisions of a single text.

Similarly, we can build hierarchical and dependency annotations on the same base text, thus effectively putting together structures such as those shown in Figures 5 and 6.

2.4 Multi-stream arrangement

In the data configurations reviewed above, we have been assuming that documents contain a single base data stream (for convenience, taken to be a stream of characters in a text). Multi-stream arrangements are found where the document contains more than a single base data stream. This is attested in parallel corpora (with two or more translations arranged in parallel and aligned on the sentence and word levels), transcribed speech corpora (with the transcription in a phonetic alphabet running along the orthographic transcription), but also textual variation in cultural heritage texts, where it is very often the unaligned gaps that scholars find interesting.

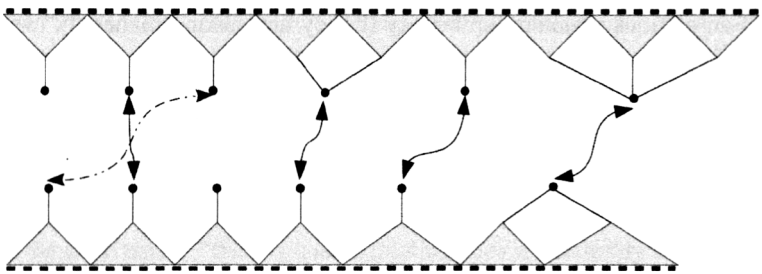

Figure 9 Multi-stream arrangement: this is nearly the simplest example imaginable, though with a level of indirection that is usually needed in such cases.

In Figure 9, we present nearly the simplest case, enhanced with a level of indirection: we may assume the dashed line to indicate characters that are grouped into tokens, which in turn are grouped into larger entities ("word forms" or phrases, cf. ISO MAF, ISO 24611:2012), and those entities are aligned with the corresponding groups built upon the other data stream. It can be seen that such a simple model already allows for expressing one-to-many and many-to-many alignments, and to identify mismatches. The above model may be used for a parallel corpus of two languages/dialects, but also for multiple versions/editions of a manuscript, in which case we can identify relationships of transposition (marked above with a dashed arrow), deletion, and insertion.

Naturally, this model can become even more complex if we decide not only to align segments (words/phrases), but also to include their annotations of various degrees of complexity. A simple example is provided below, where one stream consists of tokens annotated with glosses (literal translations), and grouped into a syntactic structure, while the other stream consists of phonetic segments (note that mismatches at this point are possible).

Finally, it is also possible to have more than two parallel data streams aligned in such a fashion—for multilingual corpora, but also for video, audio, and subtitles (in the last example, we would naturally not use characters to measure the granularity of the data stream, but rather time stamps of more or less arbitrary sampling intervals).

Note that this is a borderline category: on many approaches, multi-stream arrangement is not considered to be a representation for single documents, but rather for structures involving multiple documents (such an interpretation comes naturally when looking at parallel corpora or manuscript variation encoding).

3 Representing complex information

There are theoretically no limits to how much annotation information could be added to the text. However, from the practical point of view, things look different. The main causes of *practical* difficulties concerning the addition of multiple annotations to a single text are the various ways in which the data is "packaged" or meant to be "unpackaged."

Token	de	la	crème	glacèe
Gloss	some	the	cream	iced
Phonemics	dla		krEm	glase
Syntax	P	NP		

Figure 10 Multi-stream annotation structure: the original token stream (in French) is accompanied by a parallel stream of glosses (in English), and a stream of phonetic transcription. Copied from Wörner et al., 2006.

The digital switch in the humanities was much more than merely a transfer from a paper-based, largely linear medium to an electronic and potentially multidirectional one. Apart from speeding up and making more precise the calculations and operations that could otherwise take a very long time (with regard to calculating word/phrase frequencies and degrees of co-occurrence, or to querying individual words), and apart from opening the same text to potentially multiple different visualizations, the electronic medium has gradually made it possible to equip individual words with extra "vertical" space (in the sense discussed earlier), in which additional data could be placed, and then to use those extra data both as simple enrichment of the content, but also as the basis for various correspondence mechanisms.

These mechanisms (generally referred to as *linking*, as will be seen below) have made it possible to flesh out the dependencies that, in the general printed medium, were mostly (practically only with the exception of footnotes or bibliographical references) realized only by associations made and maintained in the mind of the reader. In contrast to such implicit and often subjective mechanisms, in the electronic medium, it is possible to explicitly link speeches by the same character in a drama, passages of uniform narration in works where narration or the narrator change, or to make use of anaphoric mechanisms or finally to encode grammatical dependencies among words in a single sentence.

On the packaging side, the problem lies in how to express the fact that a single information container (i.e., our example word) may be annotated with more than one information package, or in how relationships among such containers (e.g., the subject and the main verb in a sentence; two or more rhyming verses or the alliterating word fragments in a poem) can be encoded, or finally in how these individual elements can be grouped together (into paragraphs, sentences or stanzas).

In this section, we offer a bird's-eye view of the landscape of approaches both to encoding and to the general architecture of text resources that have been used in the e-humanities in order to enrich the primary text with annotations of various degrees of complexity. It has to be borne in mind that these technologies are often conditioned by extremely non-theoretical factors, such as funding schemes, their primary processing/visualization purpose (which may change over years), or even the habits and fashions of the local IT departments. Because of that, what follows cannot be taken as a definitive review or recommendation—the decision to adopt one kind of technology over another is influenced by too many variables.[5] Finally, we briefly present the advantages of a solution that we have adopted in the KorAP project, as an example of how some complex annotations are built and used in our everyday research.

For ease of description, and making use of the terminology introduced by Goecke et al. (2010), we divide text-annotation representations into *single-layer*, whereby the data is enriched along a single dimension (modulo cross-element references), and *multi-layered*, where it is a priori assumed that a single view of the data is either not achievable or not advisable.

3.1 Single-layer representations

Single-layer representations allow for simple linear and hierarchical arrangements that may be enhanced with cross-references that enrich the representations by providing means of relating elements that are not adjacent or that are not parts of the same hierarchy. Examples of such approaches may be simple TEI XML as exemplified below, or HTML.

HTML has a very impoverished representation model, so we mention it only for the sake of its popularity as a means to visualize the results, in the form that would in most cases not be suitable for further processing, because in most cases the transformation from the underlying data format into HTML is monodirectional—that is, since HTML is not able to express complex annotations, information is lost during the conversion from a more expressive format into HTML. Another outstanding problem that HTML faces is that, without some cumbersome workarounds, it cannot handle linear arrangements that have more than a single piece of additional information of the same kind attached to a single document element (see Figure 2 above).

Both HTML and simple TEI XML are mono-hierarchical: added information of any kind intervenes within the original text stream, which goes counter the principle of keeping the base text pristine for the sake of its sustainability and for the purpose of offering multiple and potentially conflicting views of the underlying data.

The example that follows shows two ways to accomplish the simple task of annotating line divisions and basic speaker information in the text of Euripides's *Medea*, taken from the Oxford University Text Archive (OTA, cf. http://ota.ox.ac.uk/desc/2414). What changes from one example to the next is what we refer to as the packaging of information.

```
<       THE MEDEA OF EURIPIDES       >

<P MHD.>

<S TR>

<V 0001>

EIQ' WFEL' $ARGOUV MH DIAPTASQAI SKAFOV

<V 0002>

$KOLCWN EV AIAN KUANEAV $SUMPLHGADAV,

<V 0003>

MHD' EN NAPAISI $PHLIOU PESEIN POTE
```

Figure 11 Euripides's *Medea*: fragment of an Ancient Greek text transliterated to English in an unknown system.[6] COCOA format, part of the OTA.[7]

```
<text>
  <body>
    <head type="author">MHD.</head>
      <speaker>TR</speaker>
      <ab>EIQ' WFEL' $ARGOUV MH DIAPTASQAI SKAFOV</ab>
      <ab>$KOLCWN EV AIAN KUANEAV $SUMPLHGADAV,</ab>
      <ab>MHD' EN NAPAISI $PHLIOU PESEIN POTE</ab>
```

Figure 12 The same fragment of *Medea* converted to TEI P5.

Figure 11 presents the text encoded in a legacy COCOA format (Russell, 1965), and Figure 12 shows the same fragment encoded in TEI P5 XML.

The dramatic difference between the two markup techniques is in how precise XML can be with respect to delimiting the content of the particular element or defining the hierarchical structure visible already in such a short fragment, and in how much space is provided for the various XML attributes for annotations inside element tags.

```
<sp>
    <speaker>Prospero</speaker>
    <l part="Y">I'll deliver all,</l>
    <l>And promise you calm seas, auspicious gales,</l>
    <l>Be free and fare thou well. <stage type="exit">Exit
      Ariel</stage> Please you, draw near. <stage
          type="exit">Exeunt all but Prospero</stage>
        <note place="margin">Epilogue</note>
    </l>
    <l>Now my charms are all o'erthrown,</l>
    ...
</sp>
<stage type="mix">He awaits applause, then exit.</stage>
```

Figure 13 Prospero's speech from Shakespeare's *The Tempest*.[8]

While a simple XML format exemplified above can help tackle the basic annotation tasks, the encoder will run into problems when facing the need to encode conflicting annotations, both at the basic level (e.g., tagging the pronoun *they* as "PPHS2" in the CLAWS7 tagset or as "PNP" in the BNC tagset[9]) and at the more complex level (e.g., the division into verses as well as the division into sentences, or where one grammatical theory constructs phrases in a different manner from another). An example of more complex markup, where a line of a play had to be split in order to make the result obey certain formal conditions, is presented below.

Apart from the aim of enriching basic data with more and more information, anyone dealing with digital data formats nowadays is aware of the principle of data sustainability (see Schmidt et al., 2006; Rehm et al., 2009). Despite the decreasing prices of storage and the increasing processing power of CPUs, there is a sure way to dramatically increase the cost of data production and curation: by locking the data within an unsustainable format and risking that in five or ten years, it will be extremely difficult and costly not only to retrieve and process the annotations added to the data, but even to retrieve the data itself, as we hinted in the discussion of Figure 11 and Figure 12. As has been argued by many by now, data should best be stored in open, well-documented and well-supported formats, and preferably kept separate from annotations. The latter guideline points at technical solutions commonly labeled as *standoff*—featuring markup of whatever sort that is not interspersed among the fragments of base data, but rather points at the data remotely. As we shall see below, this kind of approach makes it much more feasible to store multiple annotations for a single data fragment.

3.2 Multi-layered representations

The term "multi-layered representations" refers to data arrangements which it would be impossible, or highly impractical, to model on a single layer. Theoretically, such a task opens a hierarchy of choices, the first of which is whether to abandon the XML medium altogether and experiment with other data models and syntaxes, or whether to keep to the well-established technology and stretch it wide enough to allow for multiple representations to coexist, while at the same time being able to efficiently manipulate, curate, and query the data. The former path is represented by, among others, LMNL (Layered Markup and Annotation Language; e.g., Tennison and Piez, 2002; Piez, 2013), while the latter either relies on branching a single XML tree into subparts that encode the separate (and potentially conflicting) representations, linking them by internal XML devices, or on spreading the multiple layers across multiple documents, which can then be associated by various means.[10]

Much of this has been studied under the heading of "markup overlap" and resulted in research reported among others in the *Extreme Markup Languages* and later, the *Balisage* conference series (cf. Hilbert et al., 2005, for an overview). Part of these approaches involve annotating a single text stream with multiple series of tags, resulting in potentially major issues concerning the curation of

such resources. A partial solution to this was offered by an approach known as *multiply annotated text* (cf. Goecke et al., 2010), whereby text with multiple and potentially conflicting annotations was exploded into more than one physical document (with in-line markup) under the crucial condition that the underlying text stream provided the common baseline to the various annotation trees. This is shown in Figure 14, where the three annotation layers describe three identical copies of the very same text.

Figure 14 shows three instances of the same text annotated in-line (ideally), one layer per document. The character stream is the sole pivot that makes it possible to combine or compare the annotations. The three example character offset positions (0, 3, 6) are provided as a means of visualizing the mismatch among the annotation spans and the resulting impossibility of building a single tree structure over them.

A theoretically descendant approach (although it dates at least as far as Durand et al., 1995 and Thompson and McKelvie, 1997) is one known as *remote* or *standoff markup*, whereby the base text stream is either only lightly annotated in-line (mostly to introduce the basic XML skeleton) or not annotated in-line at all, with all the annotations in separate documents and addressing the base text "remotely," by a mixture of structural information and string offsets (see Bański, 2010, for more terminology and illustrations). See Figure 15.

Goecke et al.'s multiply annotated text was proposed (originally in Witt, 2005) as a way to overcome the technical difficulties in reassociating standoff annotations with the text, and entailed a shift of focus from maintaining and curating the annotation-text relationship to synchronizing the multiple copies of text, each with different kinds of markup.

When it is necessary to maintain a rich variety of concurrent annotation layers, and consequently to cope with multiple sources of annotation data and potentially numerous files in which they are stored, an approach based on "foundries" (or distinct sets of informational components) becomes a virtual necessity. This approach, introduced in the KorAP project (Bański et al., 2013), implements a

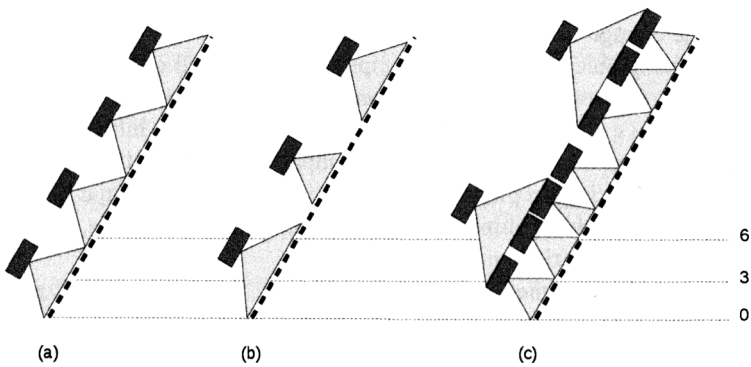

Figure 14 An example of multiply annotated text.

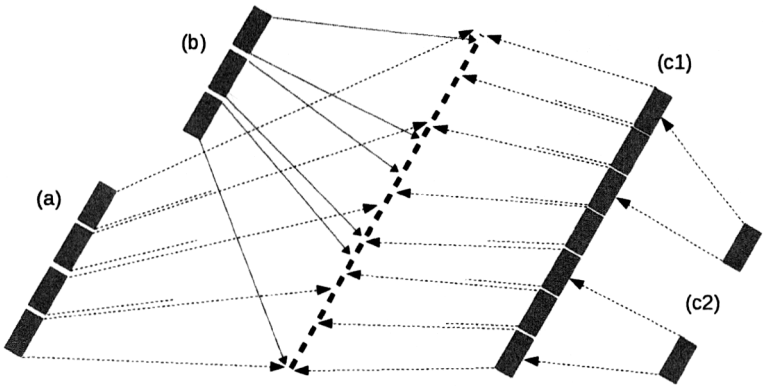

Figure 15 A sample of standoff annotation. Numerous annotation documents (a–c1 correspond to the annotations in the preceding figure) point at the same base text document, or at another document containing annotations (cf. c2). Some association lines have been shortened for the sake of clarity.

model in which each well-defined set of annotations forms a separate component of a corpus document, with its own metadata section describing the origin and the various properties of the individual annotation layers. This is diagrammed in Figure 16.

In Figure 16, each foundry contains a well-defined set of annotations that provide separate interpretations for the data described. Cross-foundry dependencies are possible—for example, foundries B and C may rely on segmentation

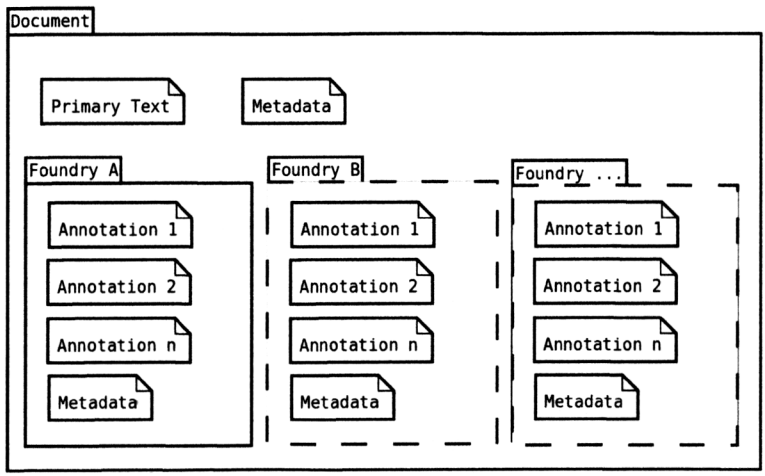

Figure 16 A document in the KorAP data model consists of the primary text (base data) that is kept separate from foundries.

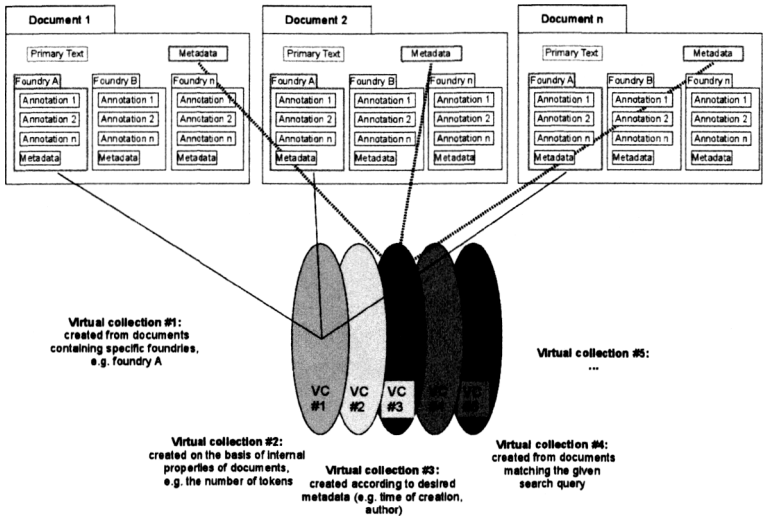

Figure 17 Illustration of the versatility of the KorAP document model. Virtual collections can be created on the basis of practically any subpart of the document, depending on the research needs.

information provided by foundry A, but offer their own annotation sets, based on different tagsets and according to different grammatical theories or modules.

Each document consists of the base text, document metadata (expressible, e.g., in the form of a TEI Header), and one or more foundries, which contain the annotation information. This kind of arrangement allows to view the text as if it was annotated by a specific foundry alone (in a single-layered fashion), or to compare the contents of foundries in the way offered by multi-layer approaches. As shown in Figure 17, *virtual collections* composed of various slices of the source repository can be created on the basis of the selected sets of criteria.

4 Summary

We have presented a representative fragment of the landscape of current practices of structuring information in complex documents. The presentation is slightly geared toward linguistic uses, but that has also served as a way to restrict the discussion, so that it could serve as a compact presentation of some of the principles that are easy to extrapolate into many of the areas in which digital humanities have recently so successfully expanded.

The examples adduced above scratch the surface of the theoretical and practical issues concerning the packaging of data, and in particular the enrichment of primary textual data with annotations. One must not forget, however, that in most cases, the packager has a further goal than merely arranging the information in a neat manner: that information should then be put to work. It should be interpretable

and processable by computers—for example, it should be easy to query, and to visualize in various ways. Exploring these issues would take us beyond the scope of this chapter and into the land of standardization, data exchangeability and tool interoperability.

Notes

1 By "vertical space," we refer to the growth of information in paradigmatic terms (to be exemplified below); the "horizontal space" of linear growth can be more readily accommodated year by year, with the development of storage technologies always staying ahead of digitisation efforts.

2 A good deal of the debate took place within the lively platform of *Extreme Markup* and its descendant *Balisage* conferences. As an entry point, consider DeRose, 2004.

3 In the process of the creation of linguistic resources, tools are used to add various kinds of grammatical information automatically, often at the cost of accuracy. Because different tools introduce different errors or are based on different theoretical assumptions, it makes sense to process the same text with more than one tool.

4 Nodes in trees can only have a single mother (that is, can only be dominated by a single node). Therefore, the relationship between the antecedent noun phrase (*the boy*) and the reflexive (*herself*) cannot be encoded as part of tree structure because the reflexive is dominated by the VP node. Note that a tree structure is not a necessary condition for the presence of references of this kind; a similar effect could be created in the linear arrangement in Figure 1, by having, for example, the last token reference the first one, in this way circumventing the linear ordering.

5 It is worth mentioning at this point that initiatives aiming at standardization of many aspects of linguistic annotation have found their home within the ISO Technical Committee 37, Subcommittee 4, "Language resource management." While some of the solutions proposed there are very specific, many can be re-used by scholars in e-humanities in their projects, especially given that the ISO efforts have been increasingly tied to aspects of the TEI Guidelines. See Burnard (this volume) for further remarks on the role of standardization in the e-humanities.

6 The corresponding text in Ancient Greek is as follows:

Εἴθ᾽ ὤφελ᾽ Ἀργοῦς μὴ διαπτάσθαι σκάφος
Κόλχων ἐς αἶαν κυανέας Συμπληγάδας,
μηδ᾽ ἐν νάπαισι Πηλίου πεσεῖν ποτε

(cf. http://data.perseus.org/citations/urn:cts:greekLit:tlg0006.tlg003.perseus-grc1:1-48)

Note, among others, that the decision to use capital letters has forced *ad hoc* markup by means of the "$" character (as in $ARGOUV vs. Ἀργοῦς), and that most of the accentual information is missing.

7 We are grateful to Sebastian Rahtz for pointing us to this example, which also demonstrates that sustainability of digital texts is a very real issue. The conversion was facilitated by the cocoa-to-tei.xsl script by James Cummings and Sebastian Rahtz, available from https://github.com/TEIC/Stylesheets.

8 XML adapted from TEI Consortium (Eds.) "Prologues and Epilogues," Guidelines for Electronic Text Encoding and Interchange. Version 2.8.0. Available at: www.tei-c.org/release/doc/tei-p5-doc/en/html/DR.html#DRPRO.

9 Both tagsets can be found at http://ucrel.lancs.ac.uk/claws/. "Tagging" refers to the process of labeling words with, for example, part-of-speech tags, whereby the individual labels come from a closed inventory of symbols called a tagset. Various tagsets have been proposed, depending on which grammatical features of words needed to be distinguished.

10 The existence of supplementary specifications such as Xinclude blurs the distinction to some extent, by allowing parts of a single XML hierarchy to reside in separate documents that can be processed either as free-standing XML instances, or as part of the entire original tree.

References

Bański, P., 2010. Why TEI Stand-off Annotation Doesn't Quite Work: And Why You Might Want to Use It Nevertheless. In: *Balisage: The Markup Conference 2010.* Montreal, Canada, August 3–6, 2010. Rockville, MD: Mulberry Technologies.

Bański, P., Bingel, J., Diewald, N., Frick, E., Hanl, M., Kupietz, M., Pęzik, P., Schnober, C., and Witt, A., 2013. KorAP: The New Corpus Analysis Platform at IDS Mannheim. In: Z. Vetulani and H. Uszkoreit (Eds.) 2013. *Human Language Technologies as a Challenge for Computer Science and Linguistics: Proceedings of the 6th Language and Technology Conference.* Poznan: Fundacja Uniwersytetu im. A. Mickiewicza.

Burnard, L. How Modeling Standards Evolve: The Case of the TEI. In: J. Flanders and F. Jannidis (Eds.) 2018. *The Shape of Data in the Digital Humanities.* London: Routledge.

DeRose, S., 2004. *Markup Overlap: A Review and a Horse.* Proceedings of Extreme Markup Languages. Available at: http://conferences.idealliance.org/extreme/html/2004/DeRose01/EML2004DeRose01.html.

Durand, D., Ide, N., LeMaitre, J., and Véronis, J., 1995. Internal Standard Formats. MULTEXT Deliverable 1.3.1B. Available at: www.cs.vassar.edu/~ide/papers/MultextD1.3.1B.ps (accessed February 23, 2015).

Goecke, D., Metzing, D., Lüngen, H., Stührenberg, M., and Witt, A., 2010. Different Views on Markup. Distinguishing Levels and Layers. In: A. Witt and D. Metzing (Eds.) *Linguistic Modeling of Information and Markup Languages: Contributions to Language Technology.* Dordrecht: Springer, pp. 1–21.

Hilbert, M., Schonefeld, O., and Witt, A., 2005. *Making CONCUR Work.* In: Proceedings of Extreme Markup Languages. Available at: http://conferences.idealliance.org/extreme/html/2005/Witt01/EML2005Witt01.xml.

International Standards Office (ISO), 2012. *ISO/FDIS 24611:2012(E) Language Resource Management—Morpho-Syntactic Annotation Framework (MAF).* Geneva: ISO.

Piez, W., 2012. Luminescent: Parsing LMNL by XSLT Upconversion. In: *Balisage: The Markup Conference 2012.* Montreal, Canada, August 7–10. Rockville, MD: Mulberry Technologies.

Rehm, G., Schonefeld, O., Witt, A., Hinrichs, E., and Reis, M., 2009. Sustainability of Annotated Resources in Linguistics: A Web-Platform for Exploring, Querying, and Distributing Linguistic Corpora and Other Resources. *Literary & Linguistic Computing,* 24 (2009/2), pp. 193–210.

Russell, D.B., 1965. COCOA—A Word Count and Concordance Generator (computer program). Associates Technology Literature Applications Society.

Schmidt, T., Chiarcos, C., Lehmberg, T., Rehm, G., Witt, A., and Hinrichs, E., 2006. *Avoiding Data Graveyards: From Heterogeneous Data Collected in Multiple Research Projects to Sustainable Linguistic Resources.* EMELD (Electronic Metastructure for Endangered Language Data). Available at: http://emeld.org/workshop/2006/proceedings.html.

Shakespeare, W., 1623. *Mr. William Shakespeares Comedies, Histories, & Tragedies.* London: Jaggard and Blount.

TEI Consortium, 2015. TEI P5: Guidelines for Electronic Text Encoding and Interchange. TEI Consortium. Available at: www.tei-c.org/Guidelines/P5/.

Tennison, J. and Wendell, P., 2002. *The Layered Markup and Annotation Language (LMNL)*. In: Proceedings of Extreme Markup Languages. Available at: http://conferences.idealliance.org/extreme/html/2002/Tennison02/EML2002Tennison02.html.

Thompson, H.S. and McKelvie, D., 1997. *Hyperlink semantics for standoff markup of read-only documents*. Proceedings of SGML Europe. Available at: www.ltg.ed.ac.uk/~ht/sgmleu97.html.

Witt, A., 2005. Multiple Hierarchies: New Aspects of an Old Solution. *Interdisciplinary Studies on Information Structure (ISIS)*, 2, pp. 55–85.

Wörner, K., Witt, A., Rehm, G., and Dipper, S., 2006. *Modelling Linguistic Data Structures*. In: Proceedings of Extreme Markup Languages. Available at: http://conferences.idealliance.org/extreme/html/2006/Witt01/EML2006Witt01.html.