Agree to disagree: Modelling co-existing scholarly perspectives on literary text

Elli Bleeker, Bram Buitendijk, and Ronald Haentjens Dekker
Research and Development Group, KNAW Humanities Cluster, Amsterdam, the Netherlands

Abstract

This essay addresses two open challenge in the domain of digital scholarly editing: (1) formally defining the meaning of markup, and (2) allowing the reuse and exchange of textual data through a distributed editorial workflow that allows the editing of texts from multiple, diverging yet co-existing perspectives. We argue that successfully addressing these issues would promote the distribution and exchange of scholarly knowledge, on a technical as well as a theoretical level. The essay introduces ongoing work on a new data model for text called 'TAG' (Text-as-Graph) and its reference implementation 'Alexandria'. The essay outlines how TAG, based on a hypergraph for text, can improve the modeling of complex literary texts, and how Alexandria supports the exchange of markup files in a way that sustains scholarly discourse. We discuss three components of TAG: first, the markup technology stack allows for the formal definition of the meaning of markup ('markup semantics'); secondly, users can add multiple layers of markup that each represent an alternative perspective on text; and finally the editorial workflow is set up in a git-like distributed version management system. As a result, the TAG model provides for the synthesis of dispersed scholarly practices and the advancement of academic discourse.

Correspondence:
Elli Bleeker, Research and Development Group, KNAW Humanities Cluster, Amsterdam, the Netherlands.
E-mail:
elli.bleeker@gmail.com

1 Introduction

'Scholars disagree on everything', a satirical piece in the New Yorker claims.[1] Ironically, most scholars would agree with that statement. Opposing views, dialectic discourse and debate are driving forces in research. Humanities research, in particular, requires room for diverging interpretations: cultural and historical artefacts can often be approached from multiple coexisting perspectives. This is one of the reasons why the academic world welcomed the digital medium with great enthusiasm: its infrastructure and theoretically infinite storage space would be able to accommodate various views on an artefact without privileging one over the other. What is more, it would facilitate the exchange of findings and promote the reuse of one another's work. In short, digital humanities (DH) was said to promote the organization and dissemination of scholarly knowledge in a way that leaves ample room for multiple co-existing perspectives on data; data that are both reusable and understandable for our scholarly peers. So far, however, that particular academic ideal has not been entirely realized. On the contrary, some have criticized the application of the digital medium to humanities research for reducing research to 'problem solving' and for promoting a neo-positivist take on research

Digital Scholarship in the Humanities, Vol. 0, No. 0, 2019.
doi:10.1093/llc/fqz061

(Eyers, 2013, 2017; Ramsay, 2011; Marche, 2012). They pose that DH methodologies leave little leeway for the existence of incomplete information, ambiguous interpretations, or varying perspectives.

Rather disconcertingly, our desire to capture the messy complexity of the natural world and express it algorithmically has been described as 'the holy grail of computer science' (Davidson, 2006, p. 66). Scholarly editions are quite idiosyncratic as a result of their material, their creators, and their intended audience. This is reflected in the Guidelines of the Text Encoding Initiative (TEI, Burnard and Bauman, 2015) that are renowned—or notorious—for their flexibility and customizable nature. While humanities scholarship is interested in the smallest, original details, computers work best with formal rules and standards. It seems, then, that the text encoding community is currently suspended in limbo, balancing between, on the one hand, the desire to customize the TEI Guidelines or develop local markup systems to perfectly harmonize every edition project and, on the other hand, the need for interoperability and the development of tools for computational text analysis. Is it possible to bring together these seemingly disparate worlds? Ideally, scholarly editors convey their interpretation of natural text objects to a computer while making sure that their interpretations can be understood, evaluated, and used by others. This situation forms the starting point of the present essay, which focuses on two long-standing issues within the field of digital scholarly editing: defining the meaning of the markup elements ('markup semantics'), and allowing the exchange and reuse of data that represent multiple, diverging perspectives on text.

Currently, the Research and Development group of the KNAW Humanities Cluster is working on a new data model for text called text-as-graph (TAG)[2] that includes the markup language TAGML[3] and the decentralized workflow Alexandria.[4] This essay describes the first results of this ongoing research and outlines how the TAG data model can address the two challenges sketched above. The first part of the essay explores the requirements of a markup system that brings together different sets of annotations on the same text in order for the source text as well as the annotations to be studied, (partly) reused, and queried by others. A typical edition project begins with the transcription of a textual source, resulting in a machine-readable transcript which is in turn encoded using some form of markup (cf. figure 1 in Rehbein and Fritze, 2012, p. 52). While 'making a transcription' sounds straightforward, on a technical level, the first step of the editorial workflow already necessitates several decisions about the data model, the markup language, the schema, and the encoding guidelines. Such decisions are closely linked to the reasons for editing the text, the research objective, and—sometimes subconsciously—by the scholar's textual orientation (cf. Van Hulle and Shillingsburg, 2015). TAG proposes a multi-layered approach to markup that brings together different scholarly perspectives on the same text, each perspective expressed by one or more layers of markup nodes. We also briefly describe how a markup technology stack can be organized to include an ontology describing the semantics of the markup elements.

The second part of the essay discusses how the editorial workflow can be organized to support collaborative work on the same text. (Re-)using or repurposing someone else's encoded texts is not a straightforward endeavour. Indeed there are few examples of TEI/XML files being effectively reused by other projects (Schmidt, 2014; Kalvesmaki and Oaks, 2014), primarily due to the idiosyncratic text encoding that stands in the way of readability and hinders interoperability. Over the years many collaborative editing platforms have been developed within the context of a project, but these are rarely picked up by non-project members. The main obstacles are that many edition projects start from scratch and that editors rarely collaborate with or join existing edition projects. In short, the risk to reinvent the wheel appears to outweigh the complexities of the exertion of finding common ground and setting up a collaboration that transcends project boundaries. We propose to set up the editorial work environment as a git-like system that facilitates version management, dissemination, and reuse of the multi-layered textual data. To be clear, these are both substantial topics, the scope of the essay is necessarily limited, and our work is under development. Nevertheless, we consider our experiences with designing a multi-layered markup

language and an editorial work environment as relevant for a productive discussion on text modelling and, by extension, textual scholarship in the digital age.

2 Markup, Meaning, and Layers

One of the greatest challenges for digital editing today is not the translation of one's interpretations of a source text to a machine-readable data format. After all, a set of markup elements, such as described by the TEI Guidelines, provides users with an extensive vocabulary to express their understanding of a text in a machine-readable format. The challenge is to ensure that these data can be read, understood, and used by others, even though they may not share your particular research perspective on the text. By way of illustration, let us take a short verse written on a manuscript folio. The text and the manuscript can be studied from different perspectives, such as the material aspects of the folio page, the meter of the poem, or the linguistic information of the text. After transcribing the text into a machine-readable data format, one can use markup elements to make additional information explicit. Accordingly, adding markup to a transcription can be understood as adding one or more layers of additional information (structural, documentary, poetic, etc.) to the information expressed by the sequence of textual characters. The markup can be either embedded in the text transcription or stored in a separate file ('stand-off').

A functioning markup system typically consists of at least four components: (1) a data model, (2) a syntax, (3) a schema, and (4) a query language.[5] In theory, each component has a different responsibility: e.g. the data model determines how the data are stored, the schema describes the structure of the data and the markup, the syntax provides a markup vocabulary to represent the data. Using a schema, the computer can validate the structural information and process the markup file. Once validated, the query language can easily find and extract elements from the markup file. How we can best deploy markup systems to express information about complex literary texts has been an ongoing research question since the late 1970s, and took solid form with the foundation of the TEI in the late 1980s. XML-based TEI can be now considered the de facto standard for text encoding (Andrews, 2013; Pierazzo, 2015, p. 130); the TEI's active and inventive user community compensating for the data model's limited capacity to express complex texts in an idiomatic way.

Over the years, several alternative markup technologies have been developed, such as GODDAG/TexMECS, LMNL, EARMARK, or stand-off approaches,[6] that aimed to address the limitations of the XML data model. These alternatives focused primarily on expressing overlapping and other non-hierarchical text structures that cannot be natively expressed in XML's mono-hierarchical model. Having interoperable and reusable markup files, however, also requires that other users (and other processors) understand the meaning of each markup element in the file. As XML elements are usually meaningful items in a natural language, some users are inclined to assume that the meaning of markup is as clear to the processor as it is to them. But, as Fabio Ciotti and Francesca Tomasi point out (Ciotti and Tomasi, 2016), the XML processor has no access to the meaning behind XML elements: it only validates the grammatical well-formedness of the XML document and does not understand that the XML element <pers> signifies 'person'.[7] What is more, one scholar's particular use of tags is not necessarily shared among their scholarly peers: editors and text encoders may disagree on the use of particular elements to tag a certain textual feature.[8] A formal mechanism to convey the meaning of a markup element to the computer, then, would not only significantly increase the processing and query potential but also facilitate the exchange and reuse of markup files. Describing the meaning of markup elements is generally classified as 'markup semantics'.

In a discussion about meaning of markup, it is important to distinguish between markup semantics and Semantic Markup (Peroni *et al.*, 2011). The former refers to the meaning of markup elements 'so that the facts and relationships represented by the occurrence of XML constructs can be explicitly, comprehensively, and mechanically identified' (Renear *et al.*, 2002, p. 119). In contrast, the term Semantic Markup concerns the description of the

content of web resources, for instance, with an RDF model, and is used within the framework of the Semantic Web. Indeed, RDF technology partly addresses the need for semantic tagging and markup ontologies, but the primary goal of RDF is to link data into existing, external networks, conform the principles of Linked Open Data (LOD) (Bizer *et al.*, 2011). Using an RDF model to transcribe and model a literary text creates a network of linked 'entities' in the form of triples (subject–predicate–object). The RDF model thus works particularly well for describing the conceptual information about a source text, but actually transcribing that source text in RDF results in a highly verbose and barely human-readable transcription that would always require some visualization tool.

As for markup semantics, there exists a considerable amount of literature on the topic, notably in the domains of Document Engineering and Semantic Publishing. One of the first approaches to adding semantic information to markup of literary texts, in particular, was the Bechamel project[9] which intended to 'give an intellectually satisfying ... account of the meaning of markup' (Sperberg-McQueen *et al.*, 2000). Although development on this project appears to have ceased for financial reasons, its cause remains laudable: to create a formalized tag set description in addition to a schema, a kind of meta-markup, and store it outside of the XML document it relates to. Over the last 10 years, an increasing amount of studies emphasized the need for a markup semantics for literary text encoding in particular.[10] The RDFTef proposal (Tummarello *et al.*, 2005) envisioned both the ontological description of literary texts in RDF/OWL and export to TEI/XML or RDF/XML formats. A similar approach is presented by EARMARK (Peroni *et al.*, 2011), which is at the same time a markup metalanguage that can express both the syntax and the semantics of markup as OWL assertions, and an ontology of markup. EARMARK also expresses textual information in RDF statements that are embedded in an XML document. More recent work on markup semantics is presented in Ciotti and Tomasi (2016) and Ciotti (2018) who propose an extensive proof-of-concept using LA-EARMARK (an extension of EARMARK using the Linguistic Annotation ontology) to formally describe a core set of TEI elements in order to create a TEI Ontology: 'a subset of shared assumptions, a common ground of notions about the role and meaning of TEI markup' (Ciotti, 2018, p. 143). This approach, too, takes into account the need for specific user ontologies: subsets of RDF statements that formalize the specific understanding of an individual user, which in turn would facilitate the exchange of textual data. In general, however, the field of text encoding knows but few practical applications for formal markup semantics, largely due to the complexity of the topic and the lack of interest from the community (Ciotti and Tomasi, 2016, pp. §8–9).

In agreement with the approaches described above, the TAGML markup stack will provide a means to express markup semantics using ontologies. The meaning of markup elements will be formally described as a set of meta-information. The TEI Ontology proposal (Ciotti, 2018) distinguishes a shared ontology, based on TEI simple, which can be complemented with module-specific ontologies. In contrast, we propose to distinguish between a TAG base ontology which defines the markup semantics of the markup elements used. This base ontology can be complemented with domain-specific user ontology. Every tag and every attribute used is mapped to an ontology, which contains both a human-readable description of the tags and a formalized description of the relationships. Together, these ontology files hold domain-specific knowledge: in this case, information that is informed by a certain scholarly perspective on a text. Figure 1 shows how the TAGML parser will access both the information in the associated ontologies and the text in the TAGML markup file. For each markup element, a number of RDF triples will be created, so the TAGML [line>Hello world<line] could produce the following statement: 'there is a document d01 with an m01 of the type ''markup element'' with the tag ''line'' that contains a t01 of the type text node with the string ''hello world''. This information will then be stored in the TAG hypergraph for text in the form of RDF-like statements: each markup element gets a node in the graph.[11] The associated ontology would explain the semantic meaning of the markup. Using

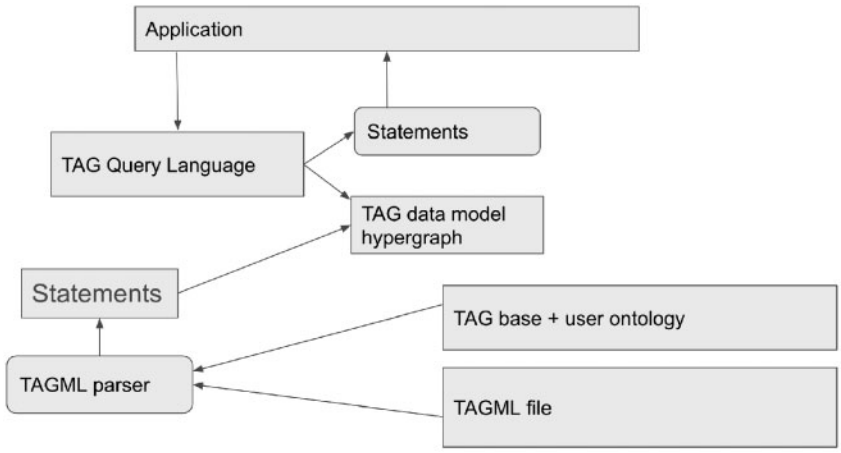

Fig. 1 Schematic representation of the TAGML markup stack

the query language TAG Query Language (TAGQL), applications will be able to query the data in the hypergraph.

By making use of the detailed scholarly knowledge that is stored in the markup and the ontologies (the domain-specific knowledge), applications will be able to carry out more complicated and powerful queries. Furthermore, as others have pointed out, having a formal description of the markup elements used would also make the file's content more accessible. Others may be interested in your transcription and may not share your interpretation of a markup element, but whether you have encoded an added signature with 'fw', 'handNote', or 'add', your interpretation is now described formally and thus readable for humans and machines alike. In short, by complementing a TAGML transcription with a formal description of the markup elements used in that file and a mapping to one or more ontologies, it can be more easily understood, shared, and reused by others.

Another important point regards the human-readability of the markup files. Embedding RDF statements into a transcription results in a verbose and significantly less legible text: a downside that cannot be underestimated in view of sharing and reuse. For that reason, encoding systems based on RDF usually resort to a standoff approach: the RDF statements are stored in an external file and linked to the source text transcription by means of pointers with identifiers.[12] The source text transcription consists primarily of text, perhaps with some minimal markup, making it easy to read. However, such standoff systems depend on a stable base text: the transcription of the source text is the reference text and cannot change as that will mess up the pointer system. Furthermore, having the markup in a separate file would hinder indexing or searching using the information in the markup. So, whereas standoff approaches have traditionally been commended for their potential to enhance interoperability and interchange (see Schmidt, 2014), we pose that to successfully query, share, and reuse markup files, a better option would be an embedded markup system with (1) the markup directly associated with the source text transcription and (2) the possibility to make changes to the markup as well as the source text transcription. The human-readability of a transcription with multiple layers of embedded markup that most probably overlap remains a challenge, which TAG addresses with its 'layer' functionality.

In TAG, the term 'layer' is used to identify a set of markup nodes that are grouped together for some reason: a layer may express a certain scholarly perspective on text (e.g. a linguistic perspective or a documentary perspective), but a layer can also identify the markup added by a particular user. Layer identifiers can thus be used for both query and display purposes. Furthermore, layers can be used to

deal with overlapping text structures (see Haentjens Dekker *et al.*, 2018 for a more extensive description of TAG layers). The layers are kept apart by means of identifiers and each layer may have its own schema that expresses the structural aspect of the markup. For example, a layer of markup nodes created from a historical perspective receives an identifier (e.g. 'hist' or 'H'); a layer of markup elements created from a linguistic perspective another identifier (e.g. 'ling' or 'L'). The same principle can be applied to distinguish between the markup elements of different users (e.g. the identifier 'BB' for markup added by Bram Buitendijk and the identifier 'EB' for markup added by Elli Bleeker).

As mentioned in the closing of the previous section, layers also function to increase the readability of a markup file. The human-readability of a source text transcription with embedded markup decreases as the number of markup increases, but we pose that few—if any—users would want to see every aspect of the information contained by the file at once. It is, therefore, safe to assume that the option to filter out information by means of the layer identifiers will be a valuable feature in view of (re)use of the TAGML transcriptions. A user can choose, for example, to view only the markup added by a colleague, or only the markup containing linguistic information, and so on. Querying a file with several layers of markup works according to the same principle. In general, it holds that the more generous the (meta)data, the more detailed the queries: queries on a plain text will return only character matches, whereas we can ask a TEI/XML file to return all speakers in a play. Now let us consider a TAGML file containing several layers of markup with an associated schema and one or more ontologies that describe the semantics behind the markup elements for each layer. Using a query language that is both ontology-aware and schema-aware, a user will be able to search for all markup elements and their associated text nodes that express information about added signatures in the margin, whether these are classified as 'fw', 'add', 'handNote', or something else. Finally, the combined information in the schema (which concerns the structure of the document), the markup, and the ontology (which describes the meaning of the markup) can be used to generate human-readable documentation.

3 Editorial Workflow

As implied in the previous section, TAG would enable a collaborative editing workflow that includes scholars from outside the edition project. This requires the accessibility of the entire data set for anyone who is interested in making use of or participating in the editorial work, conform a distributed system. Here, the term 'distributed' describes a system with decentralized access to resources all over the world, but not the way it is understood in the LOD framework. Our understanding of a distributed system is a git-like framework in which there is an open repository containing TAGML files, schemata, and ontologies, which users can 'clone' (i.e. download), edit and, if they like, upload again. The concept of a distributed architecture fits quite well within textual scholarship—or at least within the ideals of many textual scholars (Van Zundert, 2018). Without question, implementing such a system worldwide would require changes to both scholarly practices and international copyright laws which make it easier said than done. Nevertheless, we can take two important principles of a distributed system and use them to shape TAG's editorial workflow, namely a (1) platform independent system that uses (2) data in open source repositories. By way of illustration, we imagine a diverse group of scholars working simultaneously on the same source text. Note that we have used the term 'simultaneously' here instead of 'collaboratively'. The latter implies that a group of users, perhaps part of the same project, work towards one goal, say a scholarly edition of the manuscript's text. As co-workers, they would probably straighten out any discord over whether an inscription signifies an 'A' or an 'O' before finalizing the transcription. Although they still need to establish a workflow with some kind of system to manage the consecutive versions of the transcription, it gets real interesting once we think about users who simultaneously transcribe the manuscript but from different perspectives, from different (geographical)

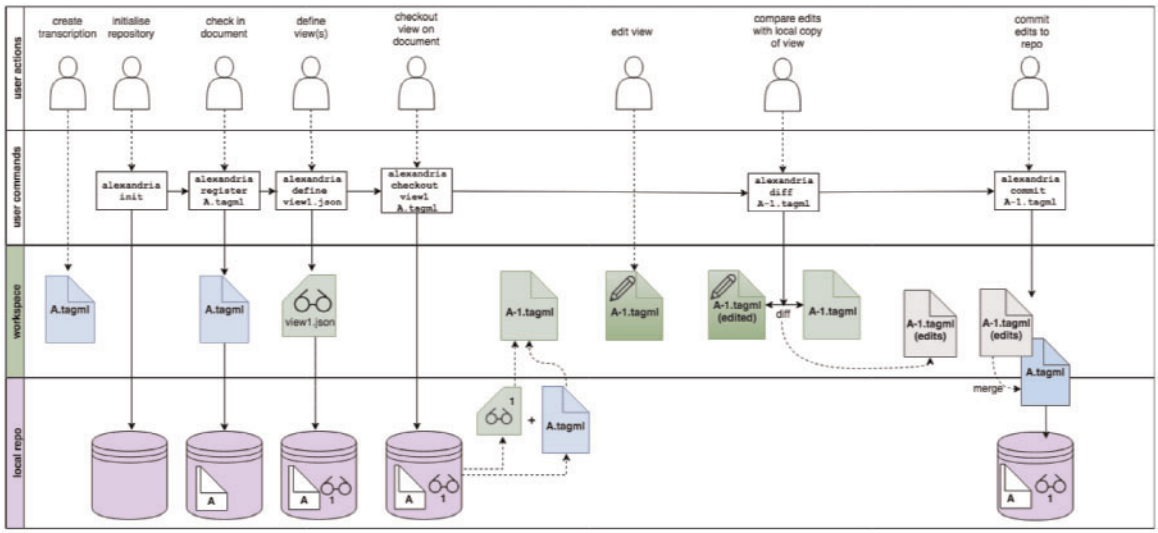

Fig. 2 Detailed schematic representation of a workflow with users editing document A

locations and/or from different edition projects. Above we already described how markup layers with unique identifiers are able to support the annotation of this document; let us now turn to the digital work environment of each scholar.

Over the years, numerous digital text edition platforms have been developed, all with the best intentions to support every aspect of digital editorial work but usually resulting in sealed-off data silos. As a result, there now is a 'graveyard' of integrated transcriptions environments (Van Zundert, 2018). Instead of reinventing the wheel by developing, once more, a new work environment or platform that promises to fulfil all editorial needs ('this time for sure'), we will have scholars work with the editor of their preference.[13] Furthermore, we take inspiration from the principles of a distributed architecture and propose a git-like[14] workflow in which scholars can take a version of the data set that they want to work on at any time and make any change they want. The reference implementation of TAG, Alexandria, is set up according to this principle. In Alexandria, users can add TAGML files and/or download existing files and make changes to them on the level of text as well as markup. After editing a TAGML file, users can upload their changes into the repository. Because it is likely that the original data set has changed in the meantime, Alexandria allows for the syncing up of changes made locally with changes made remotely (in git terms: 'merging').[15] Figure 2 illustrates this workflow.

Users of Alexandria have a lot of freedom. First, they are free to develop their own editing environment around this workflow. This implies that the version management system of Alexandria, which organizes the set of documents and layers, can be integrated into an editor of choice. Furthermore, while the Alexandria system is controlled via the command line, users can of course develop a graphical interface.

A git-like approach may offer a technical approach to a distributed version management system, it also prompts a number of philosophical and conceptual questions. First, the word 'distributed' implies that there is no longer a single authority who decides what the 'correct' text is. Different people can edit the text concurrently and in different ways: they do not have to agree with each other. As a consequence, the decision about the 'definitive' version/edition/text/document becomes a social one rather than a technical one. The social process on how to best create an edition can vary from edition to edition and from organization to organization.

Furthermore, a workflow and version management system in which editors can check in and check out a file requires a way to compare one version to another and establish the differences between them. Versions of a text are, traditionally, defined based on the concept of variance. In the context of digital editing, the comparison of two or more text versions is described as 'collation'. Current collation tools are quite competent in comparing strings of textual characters, but few—if any—take into account markup elements (see Bleeker *et al.*, 2018).[16] The workflow described in the section above necessitates a collation that gives equal treatment to text characters and markup elements. Structural changes, though, differ in some aspects from textual changes. For starters, they take place on a different level. Let us consider File A, containing one sentence that is tagged with the markup 's'. In version A-2, a user has split up this markup element and thus created two 's' elements. In a TAGML serialization, that may look as follows[17]:

File A
[s> This is a sentence. <s]

File A-2
[s> This is one sentence. <s] [s> And this is a second sentence. <s]

This split means that we go from one 's' element to two 's' elements. Arguably, the 's' elements in version A-2 are not the same as the 's' element in file A. That means we cannot simply consider the changes as the addition of an end tag <s] and an open tag [s>. We, therefore, propose to introduce a new set of edit operations: 'split' of markup and (in the opposite case) 'join' to deal with such cases. A few studies have examined how we can keep track of changes in markup of data-centric and text-centric XML (Barabucci *et al.*, 2016; Ciancarini *et al.*, 2016), but comparing transcriptions of literary texts on the level of the text as well as the markup presents a number of compelling challenges. A simple example is the presentation of a user's edit operations. This seems a superficial point, but we are dealing with a significant amount of information and arguably it makes most sense to have the user decide what edit operations they are interested in and how they prefer to see them (e.g., as a list, as a table, as a variant graph, and so on). Another conceptual question concerns our definition of 'change'. When has a file changed so significantly that it has become a new object; after one altered text character or only after a set number of edits? Do we consider new edits to overwrite older versions, or do we store a history of all edits somewhere? If it is technologically possible to compare markup files on several levels, it means that scholars are given freedom of choice which compels them to review (and communicate!) their position on long-established notions.

4 Conclusion

The first aim of this essay was to examine how we can address a vital yet open challenge of computational research in the humanities: supporting the distribution and exchange of scholarly knowledge, on a technical as well as a theoretical level. To this extent, we identified two components: formally defining markup semantics, and allowing the exchange and reuse of the textual data that are edited from multiple, diverging perspectives. Subsequently, we outlined how the TAG model can address these components and thus promote the exchange of textual data and advance scholarly discourse. Future and ongoing work on TAG focuses on the further development of (1) the distributed architecture system Alexandria; (2) the TAG hypergraph model for text that makes use of markup layers representing multiple coexisting perspectives; and (3) a markup stack TAGML that offers a formal way to describe an ontology with semantic information about the markup elements used. With our work in these areas, we intend to contribute to uniting dispersed scholarly practices and advance academic discourse.

With this essay and with the presentation of a new data model for text, we also aimed to illustrate the advantages of scholars reflecting on the ways in which they can work towards a formalization of their knowledge. This reflection is necessary if they want to ensure that textual scholars can easily understand, evaluate, and/or reuse both the textual data and the markup elements. At the same time, it

compels scholars to take another step back and reflect on the prevalent notions behind existing models. After all, the conceptual limitations of a data model may unconsciously encourage us into thinking along its structure; subtly encouraging us to ignore elements that it does not account for.

Along the same lines, it is important to continuously reflect on the needs and requirements of the text encoding community. We have to avoid remaining in an echo-chamber in which we repeat what everyone says that scholars want, without actually considering whether they (still) want it. A prominent example of that is the discrepancy between, on the one hand, an apparent desire for 'complete freedom' in text encoding and, on the other hand, the fact that we need standardization and formalization before we can really think about sharing our data. Could it be that perhaps the scholarly community does not really want to work with someone else's transcriptions? On the topic of (the lack of) distributed scholarly editions, Van Zundert remarks that '. . . colleague scholars seldom need the edition as a whole, but more often a part thereof' (2018). Future research could, therefore, explore the potential of a compromise between flexible encoding guidelines and formal standards: an API in combination with documentation that is generated from the document's schema and its ontology.

This essay set out to address two persistent challenges in digital scholarly editing: markup and an open source, distributed editorial workflow in which multiple coexisting perspectives on text are synthesized. In lieu of invalidating intrinsic humanities values by forcing complex artefacts into rigorous structures, we propose the hypergraph data model TAG that respects the intricate character of the products of the natural world, that allows room for different interpretations to exist side-by-side, and that makes use of established resources and infrastructure in a distributed manner. In short, TAG appreciates the productivity of scholarly disagreement and, instead of striving for objective descriptions, works towards formalizing our interpretations. The essay thus emphasizes the value of having various co-existing perspectives instead of one leading perspective or hierarchical structure. To put it another way—and without satire this time—to productively model scholarly disagreement.

References

Andrews, T. (2013). The third way: philology and critical edition in the digital age. *Variants*, **10**: 61–76.

Barabucci, G., Ciancarini, P., Di Iorio, A., and Vitali, F. (2016). Measuring the quality of diff algorithms: a formalization. *Computer Standards & Interfaces*, **46**: 52–65.

Bizer, C., Heath, T., and Berners-Lee, T. (2011). Linked data: the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. Hershey, PA: IGI Global, pp. 205–27.

Bleeker, E., Buitendijk, B., Dekker, R. H., and Kulsdom, A. (2018). Including XML markup in the automated collation of literary texts. Presented at XML Prague 2018, Prague, Czech Republic, 8–10 February 2018. *XML Prague 2018 - Conference Proceedings*, pp. 77–95. http://archive.xmlprague.cz/2018/files/xmlprague-2018-proceedings.pdf

Burnard, L. and Bauman, S. (eds) (2015). *TEI P5: Guidelines for Electronic Text Encoding and Interchange, Version 3.1.0*. Charlottesville, VA: Text Encoding Initiative Consortium. http://www.tei-c.org/Guidelines/P5/ (accessed 28 June 2019).

Ciancarini, P., Di Iorio, A., Marchetti, C., Schirinzi M., and Vitali, F. (2016). Bridging the gap between tracking and detecting changes in XML. *Software: Practice and Experience* **46**(2): 227–50.

Ciotti, F. and Tomasi, F. (2016). Formal ontologies, linked data, and TEI semantics. *Journal of the Text Encoding Initiative*, vol. **9**. https://journals.openedition.org/jtei/1480 (accessed 28 June 2019).

Ciotti, F. (2018). A formal ontology for the text encoding initiative. *Umanistica Digitale* **2**(3). doi:10.6092/issn.2532-8816/8174.

Davidson, J. (2006). *Measure for Measure: Exploring the Mysteries of Conducting*. The New Yorker, pp. 60–9. https://www.newyorker.com/magazine/2006/08/21/measure-for-measure-6 (accessed 28 June 2019).

Eyers, T. (2013). *The Peril of the Digital Humanities: New Positivisms and the Fate of Literary Theory*. Youtube 2013. https://www.youtube.com/watch?v=VoGc8GZ_tQs (accessed 28 June 2019).

Eyers, T. (2017). *Speculative Formalism: Literature, Theory, and the Critical Present*. Evanston, IL: Northwestern University Press.

Haentjens Dekker, R., Bleeker, E., Buitendijk, B., Kulsdom, A., and Birnbaum, D. J. (2018). TAGML: a markup language of many dimensions. Presented at Balisage: The Markup Conference 2018, Washington, DC, 31 July–3 August 2018. *Proceedings of Balisage: The Markup Conference* 2018. Balisage Series on Markup Technologies, vol. **21**. https://doi.org/10.4242/BalisageVol21.HaentjensDekker01 (accessed 28 June 2019).

Kalvesmaki, J. and Oaks, D. (2014). Canonical references in electronic texts: rationale and best practices. *Digital Humanities Quarterly* **8**(2). http://digitalhumanities.org/dhq/vol/8/2/000181/000181.html (accessed 28 June 2019).

Marche, S. (2012). *Literature is not Data: Against Digital Humanities*. Los Angeles Review of Books. https://lareviewofbooks.org/article/literature-is-not-data-against-digital-humanities/# (accessed 28 June 2019).

McCall, B. (2018). *The Oxford Book of Footnotes*. New Yorker, 1 October 2018. https://www.newyorker.com/magazine/2018/10/01/the-oxford-book-of-footnotes (accessed 28 June 2019).

Peroni, S., Gangemi, A., and Vitali, F. (2011). Dealing with markup semantics. *Proceedings of the 7th International Conference on Semantic System*. New York, NY: ACM, pp. 111–18.

Peroni, S., Poggi, F., and Vitali, F. (2013). Tracking changes through EARMARK: a theoretical perspective and an implementation. Presented at the *DChanges Conference 2013*, 10 September 2013, Florence, Italy, 10 September 2013. http://ceur-ws.org/Vol-1008/paper6.pdf (accessed 28 June 2019).

Pierazzo, E. (2015). *Digital Scholarly Editing: Theories, Models and Methods*. Surrey: Ashgate Publishing, Ltd.

Renear, A., Dubin, D., and Michael Sperberg-McQueen, C. (2002). Towards a semantics for XML markup. *Proceedings of the 2002 ACM Symposium on Document Engineering*. New York, NY: ACM, pp. 119–26.

Ramsay, S. (2011). *Reading Machines. Towards an Algorithmic Criticism*. Urbana, IL: University of Illinois Press.

Rehbein, M. and Fritze, C. (2012). Hands-on teaching digital humanities: a didactic analysis of a summer school course on digital editing. In Brett D. Hirsch (ed.), *Digital Humanities Pedagogy*. Cambridge: OpenBook Publishers, pp. 47–79.

Schmidt, D. (2014). Towards an interoperable digital scholarly edition. *Journal of the Text Encoding Initiative*, (7). doi: 10.4000/jtei.979.

Sperberg-McQueen, M., Huitfeldt, C., and Renear, A. (2000). Meaning and Interpretation of Markup not as simple as you think. Presented at the *Extreme Markup Languages Conference*, Montréal, Québec. https://www.w3.org/2000/07/07-sperberg-mcqueen-mim.html (accessed 28 June 2019).

Tummarello, G., Morbidoni, C., and Pierazzo, E. (2005). Toward textual encoding based on RDF. *Proceedings ELPUB2005 Conference on Electronic Publishing*, Catholic University of Leuven, June 2005.

Van Hulle, D., and Shillingsburg, P. (2015). Orientations to text, revisited. *Studies in Bibliography*, **59**: 27–44.

Van Zundert, J. (2018). On not writing a review about Mirador: Mirador, IIIF, and the epistemological gains of distributed digital scholarly resources. *Digital Medievalist* **11**: 1. http://doi.org/10.16995/dm.78 (accessed 28 June 2019).

Notes

1 McCall (2018) in the 1 October 2018 issue.
2 https://huygensing.github.io/TAG/.
3 https://github.com/HuygensING/TAG/tree/master/TAGML.
4 https://huygensing.github.io/alexandria/.
5 For example, the TEI/XML markup technology stack would be, respectively (1) the OHCO model, (2) TEI syntax, (3) DTD, XML Schema, RelaxNG, or Schematron, (4) XQuery.
6 An extensive overview of the strengths and limitations of markup systems for literary texts is presented elsewhere and therefore not within the scope of the article; we will discuss them where relevant.
7 To be clear, an ODD document gives users the opportunity to describe the meaning of the elements in natural human-readable language, and the ODD thus provides useful documentation for customized XML schemata. However, the information about the meaning of the markup elements is not available to the processor.
8 A recent discussion on the TEI mailing list illustrates this point: opinions differed greatly whether to tag the addition of an archival number on a letter with <add>, <note>, <metamark>, or <fw> (see the TEI List under 'encoding problem', 1–3 April 2019).
9 Primary investigator David Dubin at the Illinois School of Information Science, see https://ischool.illinois.edu/research/projects/bechamel-markup-semantics-project.

10 For a concise overview, see Ciotti and Tomasi (2016, p. §6); see furthermore Peroni *et al.* (2011); Peroni *et al.* (2013).
11 In his contribution for the workshop *Scholarly Digital Editions, Graph Data-Models and Semantic Web Technologies* (2019), Ronald Haentjens Dekker explains this workflow in more detail. See also http://wp.unil.ch/graphsde/.
12 Other encoding systems embed RDF statements in the TEI/XML transcription to identify entities (e.g. persons or places), but this has little to do with markup semantics.
13 Well-known environments for text editing are oXygen XML Editor or the open source environments jEdit or Sublime Text.
14 https://git-scm.com/.
15 In Haentjens Dekker *et al.* (2018), the authors provide a detailed description of a workflow with a distributed version management system.
16 Some advanced text collation methods let their users 'pass along' markup but the elements do not influence the comparison; their presence in the outcome is mainly for reasons of visualization.
17 For reasons of clarity and legibility, this example does not contain layer information. For more detailed information about TAGML and the representation of layers, we refer to Haentjens Dekker *et al.* (2018) and https://github.com/HuygensING/TAG/tree/master/TAGML.