Collaborative Construction of Updatable Digital Critical Editions: A Generic Approach

Vincent Barrellon

Collaborative Construction of Updatable Digital Critical Editions: A Generic Approach

Vincent Barrellon
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
vincent.barrellon@liris.cnrs.fr

ABSTRACT

In the frame of Digital Humanities, many collaborative scholarly publishing projects are arising. Editors often give shape to those projects by designing a data structure that validates the annotated content of the edition. In practise, in the course of annotation, data structures have to be updated. Besides, they determine the expressivity of the critical apparatus. The challenge is to design a data structure that will: be updatable; guarantee the consistency of the collective editorial project; reflect the different editors' needs in terms of expressivity. In this paper, we present the basis to build an edition tool dedicated to collaborative data structuring. To do so, we introduce a composite structure, constituted of a core structure (CS) and of ephemeral, peripheral ones (PS). PS will be created by individual editors to amend the core structure. They will then be discussed by the community, and eventually adopted or rejected. Means will be provided to "translate" the structured data instantiating one structure into a shape validated by the others. This way, if a PS is accepted, the CS will be updated and the instances of the previous CS will be transformed so as to match the updated CS.

Categories and Subject Descriptors

[**Human-centered computing**]: Collaborative and social computing

General Terms

Design, Human Factors, Theory

Keywords

Digital Humanities, Scholarly Publishing, Annotation, Collaborative work

Digital Libraries 2014 Doctoral Consortium
September 8, 2014, London, UK.

1. MOTIVATION

1.1 Context

Digital Humanities can be defined by their vocation to become a digital research infrastructure for humanists, in an analogous way to the infrastructure that libraries, universities, and so on, constitute in the physical world [7]. In this frame, taking advantage of the vast digitalization campaigns of cultural resources that have been led in libraries and museums, many ambitious scholarly digital publishing projects have been undertaken recently.

This work takes place at the crossroads between four such scholarly publishing projects: the edition of the documentation Gustave Flaubert gathered for his unfinished novel *Bouvard et Pécuchet*[1], the exploratory analysis of philosopher Jean-Toussaint Desanti's archive[2], the double publication (printed and online) of Stendhal's *Journaux et papiers*[3] and the critical edition of the Diderot and D'Alembert's *Encyclopédie*[4]. The four corpora are huge (e.g. more than 74.000 articles in the *Encyclopédie*), composite (e.g. J.-T. Desanti's archive contains manuscripts, administrative documents, audio files, etc.) and want extensive critical enlightening.

Over the four teams, more than sixty editors are involved. One of the teams is widely international and multicultural; all of them are multidisciplinary. Each of the four projects needs a human-computer interface (HCI) dedicated to the collaborative annotation of their respective corpus. Such a tool will be referred to as an *edition tool* hereafter.

Thus, in this work, we consider a multidisciplinary, distributed and collaborative team of scholars (the editors) gathered to produce a digital critical edition of some complex documentary corpora.

We decided to begin by working on a specific task derived from the whole editorial process, namely: *corpus construction*. In the editors' terms, it means: properly ordering the resources at hand, identifying relevant items in the corpus they represent, characterizing those items with some well-defined classification scheme, establishing correspondences between such qualified elements across the corpus, annotating the resulting contents, etc. It basically means *defining* a data structure and *structuring* the available data accordingly.

[1] http://www.dossiers-faubert.fr/
[2] http://institutdesanti.ens-lyon.fr/
[3] http ://manuscrits-de-stendhal.org/
[4] http://enccre.academie-sciences.fr/

1.2 Problem statement

Indeed, in practice, the data structures that model an edition are defined explicitly by the publishing team. In the context of scholarly digital editions, data structures formalize the informal publishing policy that makes a scholarly edition “an argument about a text” [12].

Data structures define the *types* that will be instantiated through annotation and the *links* that can be reified between instances of these types[5]. In other words, data structures define the vocabulary and the grammar of annotation.

It is clear from the history of the four publishing projects that, however well-thought-out the initial schema was, reasons occurred that led the editors to fine-tune, update, or even dramatically change the data structure, *while the corpus is in daily use*. Here are a few examples :

- the Stendhal project’s output is XML files, validated against a home-made DTD. The first DTD, in use during a few months, proved not to match the editorial policy, that was to make a semi-diplomatic transcription of the folios – while the DTD did not allow to encode tabulars, indentation, special characters, etc. A brand new DTD was designed and the whole annotation work had to be restarted from scratch. Since then, about 30 versions of the DTD have been made.

- the Desanti project started as a classification project; the current objective is to extract a dictionary of concepts from the corpus. The two enterprises involve two different but “overlapping” data structures – the latest being not entirely designed yet. Additionally, unexpected audio sources have just joined the archive. The editors want to be able to annotate those resources and to link them to the rest of the archive.

One question arises at this point : if data structures can change over time, what (who) drives their evolution? Our proposition is based upon the following assumption: since they impose a grammar and a vocabulary of annotation, data structures determine the expressivity of the critical apparatus. Thus, they define the means of expression of the editors themselves, as individuals in charge of that critical apparatus.

However, because the archive to edit grows, or the editorial policy changes, or eventually because unexpected items (e.g. tabulars; a specialization of any existing type; etc.) are uncovered during the annotation process, *editors* sometimes face resources that cannot be modeled adequately with the current data structures. Therefore, it seems interesting to design an edition tool in which editors *themselves*, as individuals working collaboratively, initiate the evolution of the data structure – in order to be able to describe those resources properly.

The problem we want to solve can be phrased as follows: a data structure must reflect the evolutive, different and even conflicting editors’ needs in terms of expressivity; at the same time it must support a single, consistent collaborative product: the edition itself. In other terms, how to allow the collaborative definition and amendment of the data structure, and how to ease the update of the structured data in case of a shift in the data structure?

[5]Hence data structures can be represented as graphs.

2. STATE OF THE ART

In traditional publishing, the notions of *critical edition* and *annotation* are inseparable. Analogously, annotation plays a fundamental role in digital scholarly editions.
Just as in the physical world, in the digital setting, “annotation” refers to two connected things : *annotation* as the process of annotating, and *an annotation* as the information (or piece of data) resulting from this process. We will see that it is crucial to consider annotation from those two points of view.

Indeed, over the last years, many tools have been designed to allow editors to annotate a given corpus. We will draft a panorama of those tools and will show that those in which the definition of the data structure is explicit (i.e. annotation lies upon a schema) model the *process* of annotation as a linear one, beginning by the definition of a data structure and then consisting in its instantiation over the corpus. On the contrary, our problem statement suggests that corpus construction must be considered a process lying upon an evolutionary data structure – that is, a cyclic process.
It is unclear so far what a collaborative and cyclic editorial process may look like. To get more insight into that question, we will seek into models of collaborative work. We will also study the notion of bidirectional transformations, that are mechanisms meant to maintain the consistency between two sources of structured information that share items, just like the respective instances of a former and an updated data structures.

Eventually, to go further, a model of annotation (here regarded as the information resulting from the process) will be required. We will mention some such models and dwell into more details about one of them, called Annotation Graphs.

2.1 Edition tools for scholars

Not all tools dedicated to sholarly edition are made to support corpus construction. For instance, it is worth mentioning Virtual Research Environments (VREs) and Creativity Support Environment (CSEs) (e.g. [2], [1]) dedicated to corpus exploration and self-targeted annotation. Obviously, those promising investigation tools are not meant to support any data structure instantiation over the corpus, let alone the data structure definition itself.

What we call an edition tool hereafter is a tool specifically made to support the more advanced stages of the editorial work that we refer to as “corpus construction” in the introduction, for which the consistency of the annotations made within a team of editors matters greatly. To be more specific, even though formalization can be regarded as an obstacle for scholars who are not used to abstraction [15], we believe that resorting to implicit structuring ([2], [1]) is not a solution to our problem, since it appears to be incompatible with a collaborative work driven by a shared editing-policy.

Indeed, many collaborative projects start by defining an ontology or a schema, on which the annotation work will be based later on ([26], [27]). This method was adopted by two of the four projects associated with this work in their early phases (namely the *Journaux et Papiers* and *Bouvard et Pécuchet* projects). The resulting schema would then be instantiated thanks to a simple XML-editor, even though such a tool is not meant to support collaborative work.

User-friendly and collaboration-oriented edition tools do exist nonetheless. A good example of that is the Glozz Platform [16].

The Glozz Platform is dedicated to linguistic annotation. It is designed in a holostic way. It provides the editorial team with means to define several data structures, in order to allow to enrich the corpus according to several linguistic paradigms (syntax, semantics, etc.) at the same time. The annotation process takes place through an ergonomic HCI. Several views of the documents are available.

However, we can mention here two limitations of this system that prevent it from meeting the requirements set by our problem statement. Firstly, in the Glozz Platform, a separation is made between "campaign managers", who basically define the text to be annotated and the data structures to be instanciated, and "editors", whose role is to instanciate the pre-defined data structures. This hierarchical division of the editorial work is more about cooperation than real collaboration, according to [18], and reduces editors to the role of operators in the editorial process, who can not personnally amend the data structure they use to give shape to the critical apparatus. More fundamentally, the underlying editorial process is *linear*. Secondly, it appears that no intrinsic way is given for the editorial team to collaboratively define the data structures; in case of a shift in one of the data structures in the course of annotation, only ad-hoc update of its instances (by means of scripts) is possible.

2.2 Collaborative Editorial Routines

Those limitations are not particular to the Glozz Platform. Indeed, they are rarely considered problematic : most tools dedicated to explicit structuring rely on a two-phased routine in which the annotators are not collaboratively engaged in the definition of the data structure. A few counter-examples prove that this linear routine *is* problematic in some sophisticated editorial settings.

The DINAH system [20] provides the user with the possibility to define a vocabulary of annotation types in the very course of annotation, and then to exploit this vocabulary to generate multidimensional views of the corpus in which documents are displayed along several dimensions that each correspond to a type of annotation. Compared to implicit structuring, this process allows the editor to weave connections more tightly across the corpus; it also gives the annotator back the definition of her own means of expression. However, because each editor is free to define her own vocabulary of annotation, the tool does not seem to fit collaborative work driven by a common editorial policy.

Another promising example is the @-note system [17]. It is an online application dedicated to the collaborative annotation of literary texts. This system can be classified as en explicit structuring one, along with the Glozz Platform, but two characteristics set it apart :

- the system is based upon the notion of "annotation activities", which basically refers to the association of a corpus and a simple annotation schema (a hierarchy of annotation categories and types). Importantly, the system supports the *collaborative* definition of those "annotation activities", by a restricted set of editors – so it is a first example of collaborative schema definition platform. However, it is unclear what the collaborative rules are from the availible literature on @-note.

- in the course of annotation, any editor has the possibility to modify schemas, under certain conditions, by defining new types and categories.

While those examples illustrate the need to question annotation (as a process), and to propose new routines more adapted to digital edition, some other studies target a better understanding of the sociological or psychological aspects of collaborative work, in order to design well-fitted collaborative processes, interfaces or benchmarks. The widely adopted (e.g. [21], [22], [23]) concept of Common Ground arises from those studies.

This concept originates from linguistics. It is a model of conversation, based on the consideration that collaborative work can be achieved even though the coactors do not share a common comprehension, or representation [8], of its object, be it at the beginning or at the end of the interactive process. The explanation is that action is possible if there is a *feeling of* mutual understanding, "sufficient for current purpose" [6].

An interesting reformulation of the concept can be found in [4]. This paper deals with multidisciplinary intellectual work. Multidisciplinarity implies divergence of perspectives and epistemic styles. Coactors, consequently, never share a common understanding of the object of their task. Interaction becomes "processes of confrontation between different structures of knowledge" – thus, divergence is seen as a driver for interaction.

2.3 Bidirectional Transformations

The concept of Common Ground will drive us forward to define a cyclic editorial process in which each editor is associated to the real-time update of the data structure (see 3.1). This process will lead to the coexistence of several schemas sharing patterns and, consequently, of instances of those schemas that will also share items, so technical solutions to maintain the consistency of those instances will be needed.

Bidirectional transformations are precisely mechanisms that allow to maintain the consistency of two structured sources of information, denoted A and B hereafter, that share items. There are three main approaches in the field of bidirectional transformation: Lenses, Triple Graph Grammars (TGGs) and UNQL+. Lenses [10] are transformations capable of propagating an edit on one structure into an appropriate edit on the other: if a set A is connected to a set B by a lens, updates on A will be mapped to updates on B, and conversely. Unfortunately, the use of lenses is limited to very simply structured data, since they only work on trees. TGGs are grammars that generate languages of graph triples which consist in two related graphs A and B plus a graph that serves as a bridge between them [14]. This only works if the pattern-to-pattern correspondence between the related graphs is well known and stable over time, which can not be guaranteed in our context.

UnQL+ [11] is a graph algebra enriched with bidirectional semantics. It is based on the UnQL/UnCAL algebra [5], whose graph model is a rooted, directed and cyclic graph with labelled edges, and optionally marked and indexed nodes.

In UnQL+, whenever a (forward) transformation[6] Ff is performed on A, giving B, a corresponding (backward) transformation Bf is *automatically* defined, so that any update on B can be propagated to A.

[6]Transformations are defined as the composition of graph constructors and a recursion operator that allows structural recursion on graphs (see [9]).

2.4 Models of Annotation

We have seen that all bidirectional transformations do not work on all kinds of structured data (for instance, Lenses do not handle anything but hierarchical data), which impels us to determine the shape editorial data take, and to look for a proper data model for the instances, that is, in general terms, for an annotated corpus, regardless of the editorial project, for the sake of genericness.

Indeed, several generic theoretical models have also been proposed (and, lately, implemented). The directed graph-based models E-mu [25], LAF-GraF [24] and the more popular Annotation Graphs (AGs) [3] are such propositions. The first claims to be "widely equivalent" to the last one; the second conforms to the graph model chosen by the designers of the AGs to represent the annotated data.

AGs are directed acyclic graphs with edges that can be labelled with fielded records. The content of the *annotations* (that enrich the *annotated contents*) is contained in the edges' labels. Optionally, nodes can be labelled with indexes that can be used as references to the annotated content. Prefixes can be added to labels in order to group annotations into classes. Also, by means of suffixes, labels can reference to other labels for the specification of N-P relationships between notes.

The model is versatile enough: most of the existing annotation formats (E-mu, XML/TEI, etc.) can be translated into AGs. It is also worth mentioning that the graph model of AGs is very similar to the one at work in UnQL+.

3. ONGOING WORK

3.1 An interpretation of the Common Ground

In the light of the above considerations, one may consider that the definition and the renegotiation of data structures should be regarded as collaborative tasks *per se*. However, to our knowledge, there is no existing tool dedicated to such tasks, provided one (following the editors of the four editorial projects) considers schemas more than a hierarchy of types. Consequently, we will try to give shape to a data structure both product of and support to collaborative work. Such a data structure should reflect a consistent editorial policy, and at the same time meet the expressivity needs of the editors as individuals.

To solve this paradox, we developed an new interpretation of the concept of Common Ground. In our context, a data structure can be regarded as a representation of the edition to be made. Literature on the Common Ground indicates that no unique representation of the edition will arise; on the contrary, new perspectives may develop from the confrontation of diverse representations. However, editors may agree on an ephemeral feeling of mutual understanding, based on the use of a basic, common annotation language, or upon the confidence that one of them can lead an expert editorial project, in the frame of the common project.

We can rephrase this more concretely. An editorial data structure can be composite. It can be made of an evolutionary core structure and temporary peripheral structures. The core structure is made of types and links upon which the whole team of editors agreed at an instant t. This agreement could be based upon the fact that they share the impression that they are able to implement it. Peripheral structures are proposed by any editor, and are defined as modifications of the core structure.

Such peripheral structures are not meant to coexist independently. A typical scenario follows.

1. A publisher instantiates S, which is the core structure.
2. While annotating, she notices that one of the types in S is not adequate for the content to be annotated. She transforms S into a peripheral structure S', in which she defines a new pattern of types in place of the former one.
3. She argues in favour of S' before the other editors, through the edition tool, by showing use-cases and instance samples – the other editors reply.
4. S' is either accepted or rejected by the community of editors. In case of acceptance, S' becomes the new core structure.

This scenario raises technical and practical challenges. In particular, technically speaking, when two structures are defined, we want to have means to transform the instances of each of those structures so as to make them match the other structure. In other terms, it means that a bidirectional transformation must be weaved between the two competing instances of CS and PS respectively. Practically speaking, when defining a peripheral structure by modifying the core one, an editor shall be given ways to preview the effects of his structural modification over the existing annotated data.

Meeting those challenges would open promising perspectives. Editors would be given ways to fine-tune the existing core structure, or to propose new peripheral structures to enrich the initial editorial project and to experiment on those structures. More fundamentally, if we had ways to translate structured data from one structure to another by the means of a bidirectional transformation, then even if the editors were working on peripheral projects, data from those side projects would be converted into a shape compatible with the core structure; thus the collective edition, validated by the core structure, would keep progressing. Eventually, if a peripheral structure was accepted and the core structure updated, editors would be given the possibility to update the *data* instantiating the obsolete core structure; otherwise, the work done by the proposing editor would still be preserved, by being translated into another shape, respectful of the collective editorial policy.

Those challenges could be met by modeling the annotated data by means of an annotation model enriched with bidirectional semantics. Since we want to stick to an existing model of annotation, the goal for us is to bidirectionalize Annotation Graphs.

3.2 Bidirectionalizing AGs

We have already mentioned the fact that in terms of graph models, AGs and UnQL+ are very similar, which makes UnQL+ a privileged candidate for bidirectionalizing AGs. We insist here on the fact that the way UnQL+ works is highly compatible with the challenges we listed in section 3.1: Be S the core structure, I_S data instantiating S. An editor defines a peripheral structure S' by a transformation g on S [7], and instantiates S' in the shape of $I_{S'}$. Let us

[7] The editor will not have to define g directly: g will be obtained by composition of all the consecutive interactions of the editor with the data structure, mediated by the HCI.

imagine that we are able to determine Ff/Bf from g, so that $Ff(I_S) = I_{S'}$ and $Bf(I_{S'}) = I_S$. Once the bidirectional transformation Ff/Bf available, any update on the instances of S' will be propagated on the instances of S. This way, for instance, when an editor is annotating a part of the corpus according to the data structure S', he is actually, automatically, structuring the same part of the corpus according to the core structure S, because Bf propagates on I_S the modifications the editor is performing on $I_{S'}$.

The problem is : how to determine Ff/Bf from g?

3.3 The notion of (bi)simulation

So far, we only have clues about how to answer this question. Our intuition is to try to represent data structures and structured data in a "similar" way, so that g and Ff be as close as possible. It will then be possible to derive Bf from Ff [11]. The "similarity" between the representations of structures and instances we want to experiment on is called simulation.

It happens that UnQL/UnCAL is based upon a notion of extended bisimulation, an equivalence relation that comes from state transition systems (STS). Be $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ two STS. G_2 simulates G_1 if there is $S \in V_1 \times V_2$ so that if $(u_1, u_2) \in S \wedge (u_1, \epsilon^*.a, v_1) \in E_1$, $a \neq \epsilon$, then there is a node v_2 so that $(v_1, v_2) \in S \wedge (u_2, \epsilon^*.a, v_2) \in E_2$. Here we denote $\epsilon^*.a$ a path made of ϵ-edge and an edge labelled a. The relation "G_1 simulates G_2" will be denoted $G_1 \hookrightarrow G_2$ hereafter.

A bisimulation is a simulation S so that S^{-1} is a simulation as well. Such a relation is denoted $\equiv$ hereafter.

An important property of (bi)simulation is that the multiplicity of the edges of a given label outing from a given node is indifferent – see figure 1. This suggests that we may be able to represent a structure and instances of it (that may contain several instances of a given type from the structure) so well so that the graph representing the structure simulates the graph representing the data.

Another important property is that any composition f of graph constructors and recursive transformations is bisimulation generic (i.e. preserves bisimulation) [5]:

$$\forall \{G_1, G_2, ...\} \text{ and } \{G'_1, G'_2, ...\} \mid \forall i, G_i \equiv G'_i,$$
$$f(G_1, ..., G_N) \equiv f(G'_1, ..., G'_N).$$

Any transformation operated by the editors will be bisimulation generic (see note 6). Note that bisimulation generic functions are simulation generic.

Based on those properties, we are currently formalizing a representation of data structures and instances that are indeed connected by a relation of simulation, the structure simulating the instance. Accurate introduction to this mode of representation is beyond the scope of this paper; besides, it is an ongoing work that needs testing. The principle of this new representation is based on two facts:

1. AGs can be seen as enriched STS;

2. according to [14], a data structure defines a language whose words are paths of its own instances.

This indicates that a data structure and instances of this data structure can equally be represented as STS, so well so that any path in any instance of some structure is a possible execution of the structure-automaton. See figure 2 for a basic illustration.

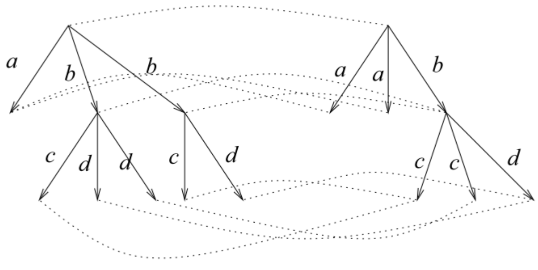

Figure 1: Two bisimilar graphs (taken from [5]).

This is only possible if the graphs representing structured data do not include values that are instance-specific, i.e. that do not appear in the graph representing the data structure. This is compatible with AGs, since annotated contents are only *referred to* by indexes placed on nodes – and simulation does not see values on nodes.

With such a representation, we are in the following situation: given a core structure S and instances I_S, a peripheral structure S' obtained *via* transformation g and instances $I_{S'}$,

$$\begin{array}{ccc} S & \overset{g}{\rightarrow} & S' \\ \hookrightarrow & & \hookrightarrow \\ I_S & \overset{Ff/Bf?}{\leftrightarrow} & I_{S'} \end{array} \qquad \text{and } (g.S = S') \hookrightarrow g.I_S.$$

It only suggests that in such a configuration, g and Ff might indeed be "close" one from the other – or even equal, in some situations. But this is sheer speculation at this stage of our work.

4. RESEARCH PLAN AND CONCLUSION

In the future, we want to formalize the automaton-like representation of data structures and instances we just give a draft of here.

We want to implement a prototype tool in order to test structural updating *in-situ*. This may require to tweak the UnQL+ algebra, firstly in order to allow for the inclusion of ϵ-edges in graphs representing data structures (so far, ϵ-edges are eliminated whenever a transformation is performed), and more importantly because UnQL+ has an asymmetric behaviour, which means that when two domains are connected via a bidirectional transformation, one is considered the very source of information and the other one a (restricted) view over the source. This does not correspond to the expected situation in our setting: a PS could be "richer", in terms of information, than the CS, and conversely.

We hope to contribute by providing editors with a structural update support tool, based on a versatile annotation model, that is Annotation Graphs. Such a tool may give back to the editors the means to master the expressivity of the critical apparatus they are in charge of, to experiment on new enrichments while contributing to a coherent, collective project, and to fine-tune the core structure validating the collective product, that is the digital edition itself.

5. ACKNOWLEDGMENTS

This work is supported by the ARC5 program of the Rhône-Alpes region, France.

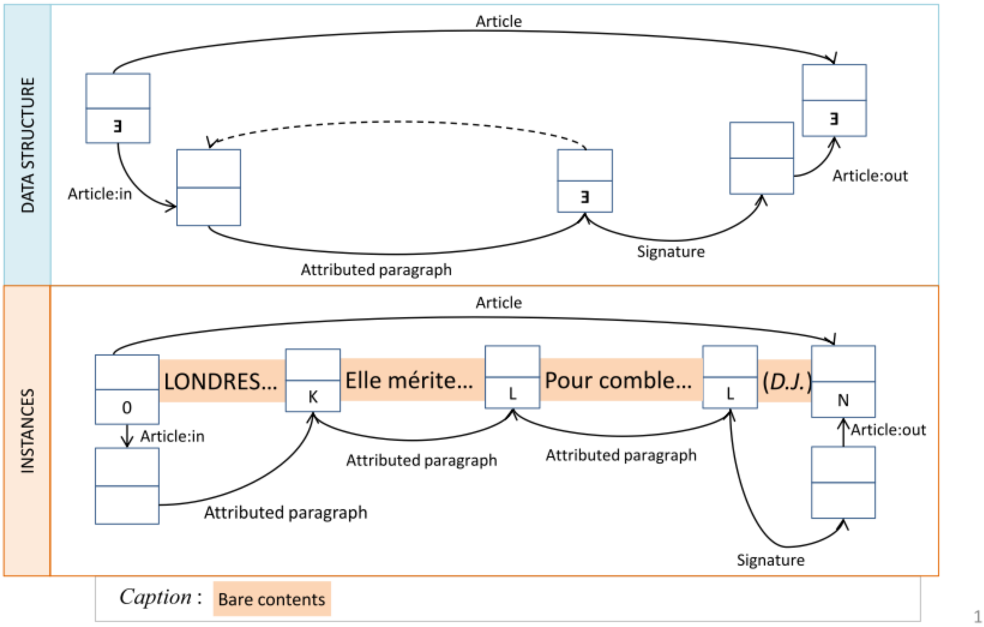

Figure 2: The structure states that an article contains one or more "Attributed paragraph", and one "Signature". The ∃ symbol indicates that a corresponding node in the instances should be indexed. An instance of that is illustrated underneath. The bare contents are not included in the instance graph, since contents are referred to by the indexes on the nodes of the graph.

6. REFERENCES

[1] C. Andrews and C. North. Analyst's workspace: An embodied sensemaking environment for large, high-resolution displays. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 123–131. IEEE, 2012.

[2] N. Audenaert, G. Lucchese, and R. Furuta. Critspace: a workspace for critical engagement within cultural heritage digital libraries. In *Research and Advanced Technology for Digital Libraries*, pages 307–314. Springer, 2010.

[3] S. Bird and M. Liberman. A formal framework for linguistic annotation. *Speech communication*, 33(1):23–60, 2001.

[4] R. Bromme. Beyond one's own perspective: The psychology of cognitive interdisciplinarity. *Practicing interdisciplinarity*, pages 115–133, 2000.

[5] P. Buneman, M. Fernandez, and D. Suciu. Unql: a query language and algebra for semistructured data based on structural recursion. *The VLDB Journal-The International Journal on Very Large Data Bases*, 9(1):76–110, 2000.

[6] H. H. Clark and S. E. Brennan. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149, 1991.

[7] P. D'Iorio and M. Barbera. Scholarsource: A digital infrastructure for the humanities. *Switching Codes. Thinking through New Technology in the Humanities and the Arts*, pages 61–87, 2011.

[8] S. Ehlinger. Les représentations partagées au sein des organisations: entre mythe et réalité. *Actes du 8ème congrès de l'AIMS*, 1998.

[9] M. Felleisen. *How to design programs: an introduction to programming and computing*. MIT Press, 2001.

[10] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt. Combinators for bi-directional tree transformations: a linguistic approach to the view update problem. In *ACM SIGPLAN Notices*, volume 40, pages 233–246. ACM, 2005.

[11] S. Hidaka, Z. Hu, K. Inaba, H. Kato, K. Matsuda, and K. Nakano. Bidirectionalizing graph transformations. In *ACM Sigplan Notices*, volume 45, pages 205–216. ACM, 2010.

[12] P. Robinson. What Digital Humanists don't know about Scholarly Editing. *SDSE2013*, 2013.

[13] R. Sanderson, B. Albritton, R. Schwemmer, and H. Van de Sompel. Sharedcanvas: a collaborative model for medieval manuscript layout dissemination. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 175–184. ACM, 2011.

[14] A. Schürr and F. Klar. 15 years of triple graph grammars. In *Graph Transformations*, pages 411–425. Springer, 2008.

[15] F. M. Shipman III and C. C. Marshall. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW)*, 8(4):333–352, 1999.

[16] A. Widlöcher and Y. Mathet. The glozz platform: a corpus annotation and mining tool. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 171–180. ACM, 2012.

[17] J. Gayoso-Cabada, A. Sanz-Cabrerizo and J. L. Sierra

@note: an electronic tool for academic readings. In *Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: metadata, vocabularies and techniques in the Digital Humanities* pages 17–21, ACM, 2013.

[18] F. Henri and K. Lundgren-Cayrol. *Apprentissage collaboratif à distance*. Presses de l'Université du Québec, 2001.

[19] M. Agosti and N. Ferro. A formal model of annotations of digital content. In *ACM Transactions on Information Systems* (TOIS), 26(1), 3.

[20] P.-E. Portier and S. Calabretto. Towards a user-centered and dynamic construction of dimensional hypertexts structures. *Hypertext*, 2011.

[21] G. Convertino, et al. Supporting content and process common ground in computer-supported teamwork. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2339– 2348. ACM, 2009.

[22] P. A Kirschner et al. Coercing shared knowledge in collaborative learning environments. In *Computers in human behavior*, 24(2), pages 403–420, 2008.

[23] K. Makitalo et al. Mechanisms of common ground in case-based web discussions in teacher education. In *The Internet and Higher Education*, 5(3), pages 247–265, 2002.

[24] N. Ide, L. Romary and E. de la Clergerie. International standard for a linguistic annotation framework. In *Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems*, 8, pages 25–30, 2003.

[25] S. Cassidy and J. Harrington. Multi-level annotation in the Emu speech database management system. In *Speech Communication*, 33(1), pages 61–77, Elsevier, 2001.

[26] C. Gonano et al. Converting the Zeri photo archive in Linked Open Data: formalizing the conceptual model. ACM/IEEE Joint Conference on Digital Libraries, 2014.

[27] F. Tomasi et al. Towards a taxonomy of suspected forgery in authorship attribution field: a case: Montale's *Diario postumo*. In *Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: metadata, vocabularies and techniques in the Digital Humanities*, 2013.