Documenti Multiversione: una soluzione per gli artefatti testuali del patrimonio culturale

Multi-Version Documents

Desmond Schmidt
Domenico Fiormonte

SOMMARIO/ *ABSTRACT*

Gli artefatti del patrimonio culturale presentano due difficoltà per il codificatore: come rappresentare versioni modificate o distinte della stessa opera, e come codificare documenti stratificati usando i linguaggi di markup. Entrambe sono forme di variazione testuale e possono essere rappresentati in modo accurato usando un documento multi-versione, basato su un grafo diretto a ridondanza minima in grado di separare in modo preciso la variazione dal contenuto.

Textual cultural heritage artefacts present two serious problems for the encoder: how to record different or revised versions of the same work, and how to encode conflicting perspectives of the text using markup. Both are forms of textual variation, and can be accurately recorded using a multi-version document, based on a minimally redundant directed graph that cleanly separates variation from content.

Parole chiave: Digital archives, overlapping hierarchies, markup, cultural heritage.

1. Introduction

Literary and historical works comprise a wide range of cultural items, for example collections of papers by famous people, drafts of literary or philosophical works, letters and even audio recordings. Some of these items may be sufficiently important to warrant transcription in order to facilitate searching or to enhance readability for online presentation. Since the publication of the SGML standard in 1986 [19] generalised markup has been the method of choice for digitising textual artefacts of our cultural heritage. Unfortunately for almost as long the process of encoding historical texts in digital form has been fraught with the serious problem of how to represent overlapping structures, which naturally occur in such texts [1]. Markup, and nowadays this usually means XML [5], is ultimately derived from the computable formal languages developed by linguists in the 1950s [9]. The context-free grammars of SGML and the regular languages definable within XML [25] define a tree-structure used as a container of text. Computers are readily able to process such structures, but markup cannot accurately model the structure of paper-based texts. This failure of markup is concentrated in two areas:

1. Variation caused by corrections and alternatives by the writer, by redrafting or by copying
2. Loss of well-formedness caused by conflicting perspectives in the markup, or naturally overlapping structures in the text itself

We will examine each of these cases and show how the second case of overlap is entirely contained by the first.

2. Overlapping Structures

2.1 In Text

Since 1996 at the University of Edinburgh (Division of European Languages and Cultures) an online archive of literary artefacts by contemporary authors, called Digital Variants, has been available online [15,16]. The initial idea of the project was to provide a digital resource for the study of the literary writing process. With the advent of the computer fewer writers save the different versions of their texts, and from the point of view of textual criticism and the writing pedagogist this implies a loss for the knowledge of the work's textual genesis. Through the digitisation of these otherwise lost drafts, pre-texts, and writing sketches (in both image and text format), Digital Variants (DV) was the first 'digital window' opened into the writer's kitchen, showing the complex phenomena underlying the final version of a work.

In the last ten years the DV team has been experimenting with a number of instruments and tools for preserving the original material, and at the same time offering to the user the possibility of exploring the authors' writing process. We started in 1997 with HTML and Javascript for displaying both the images and transcriptions of autographs (Sanvitale, Cerami), and worked in parallel on SGML-TEI editions of the same variant texts [15]. Recent experiments include XML-TEI encoding [42] and XSLT visualisation of Vincenzo Cerami's multi-versionshort stories [37], and an interface realised in Flash, which tries to capture the fluidity of the composing process in Valerio Magrelli's poems [17].

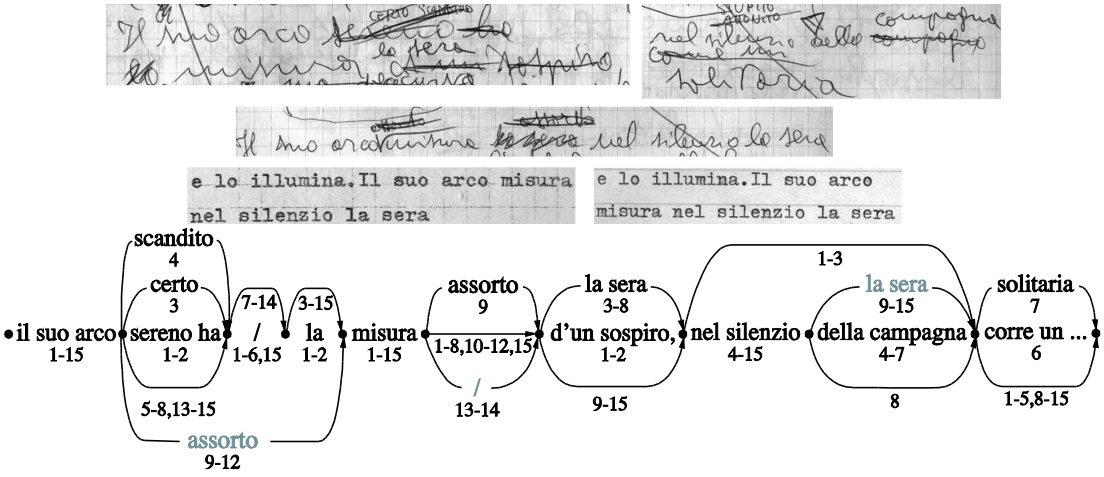

Figure 1

However, none of these solutions, due to the variety and complexity of textual phenomena involved, proved to be entirely satisfactory. Although XML seemed at least to guarantee preservation through time and provide possibilities for searching, it showed more than one limit both at the level of encoding and display/visualisation of the writing process. Similar difficulties have been reported elsewhere wherever markup is used to record variation in original documents [2, 30, 44, 45]. Variants are essentially non-linear texts, and their nature resists any kind of digital representation based on hierarchical modelling [45].

2.1 In Markup

The second half of this problem arises from the analysis of marked up texts that are not in themselves composed of different versions. Markup is usually regarded as an interpretation. An encoder selects one or more aspects of interest in a text and adds tags to represent them [38]. However, the markup for each of these aspects or perspectives may overlap. Potential sources of conflict in literary text include its physical structure (e.g. chapter, paragraph, line), graphical details such as underlining, erasure etc, or its metrical, syntactical, dramatic, prosodic, narrative, rhetorical or discourse structure [34, p.120]. Within each perspective elements might also overlap in ways that could not plausibly be teased out into separate hierarchies. Examples include annotations, variant readings and strikeouts [33]. Indeed, it is now generally recognised that literary texts frequently or even predominantly exhibit overlapping structures [6, 12, 22, 33].

Individual hierarchies or perspectives may also be completely separate but refer to the same text, or they may partially overlap, sharing some tags, or they may not even share all of the text. The problem seems insurmountable, indeed a great variety of methods for representing overlapping hierarchies have already been tried, but no one has yet produced a solution adequate for all situations [1, 3, 7, 10, 11, 13, 20, 21, 39, 40]. As Maas remarks, 'there is no silver bullet for the modelling of data in XML, when there is an overlap problem; the user must find the best solution that suits his or her particular needs [29, p.18].

In fact the entire overlapping hierarchies problem can be subsumed into the broader problem of variation described above. Since the tags are all part of the text, each hierarchy can be written out in full, even if they partly merge with one another. Hence if there are N overlapping hierarchies this can be easily turned into N variant texts, each copy sharing the content, or some part of it, and some tags with other versions.

We shall thus concentrate on the textual variation problem and describe a model that is adequate to represent that.

3. The Model

This description of the problem makes it clear that one should not proceed with the digitisation of cultural textual artefacts without first resolving the problem of overlap. In contrast to the traditional approaches described above, our model represents variant texts as a minimally redundant directed graph. Figure 1 shows a small section of one of the poems in the Digital Variants archive by the contemporary Italian poet Valerio Magrelli, and its representation in the model.

The process by which the text is converted into the graph is an editorial task, and requires the exercise of human judgment, in deciding which pieces of text belong to which versions [14]. The graph facilitates this process because it allows the encoder to record the natural structures of the text, although interpreting a complex document like this is still difficult.

We don't claim any originality for the structure of the graph - it is much the same as the original PERT/CPM graph that was once used to model workflows [8, 32]. It is also not unlike some structures that have been used in bioinformatics for representing multiple sequence alignments [27]. But we do claim originality for the labelling, the precise definition of this structure which we call a 'variant graph', and in particular for the method of compactly storing it. A variant graph is defined as follows:

1. A variant-graph is a directed acyclic graph with two special nodes called start and end. The unique start node has no incoming arcs and at least one outgoing arc. The unique end node has no outgoing arcs and at least one incoming arc. All other nodes have at least one incoming and one outgoing arc.
2. Each arc is labelled with a string, which may be empty, and with a set of versions, which may not.
3. For each version v_i represented by the graph there is a single path from start to end such that v_i is a subset

of the set of versions belonging to each arc in the path.

The graph is not otherwise constrained, and arcs may freely overlap. By following the appropriate path we can read off any version, e.g. version 7: 'Il suo arco/misura la sera nel silenzio della campagna solitaria'. The line-end in the middle of that quote is part of the text and is also subject to variation. Between the second and third drafts the line-break after `misura' is moved back after `arco'. Very often writers join lines together or split them apart as they edit a document, and to record that in markup is very difficult, because it violates well-formedness, but not in our model.

In the variant graph insertions and deletions are represented by empty arcs. Variant versions or exchanges are simply alternative paths. Transpositions are represented by a combination of insertion and deletion in which the inserted and deleted text is the same. In these cases the second time the text occurs it is referred to or pointed to rather than copied. For example, in the figure above 'la sera' is transposed around 'nel silenzio' in versions 9-15, and the second time it is drawn in grey to show that the text is only stored once.

3.1 Advantages of the Variant Graph

One of the most attractive characteristics of the variant graph is its clean separation of content and variation. The content of each version is expressed by the labels of each arc, while the variant information is encoded as the graph's structure and by the sets of versions that also label the arcs. This means that any technology can be used to encode the content, including XML. One advantage of this approach is that the markup, which no longer has to try to represent complex overlapping structures, can now be quite simple. The only extra functionality that would have to be added to a conventional editor to handle documents based on variant graphs, would be that instead of loading an entire document, it would need to request a single named version. All other complexities of the process could be hidden from the editor by an Application Programming Interface, as shown in Figure 2.

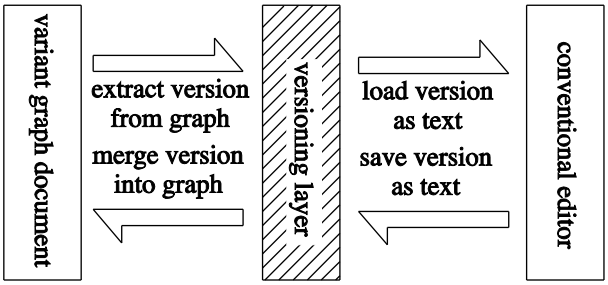

Figure 2

4. Representation of the Graph and its Algorithmic Properties

The standard means of storing graphs, the adjacency list and adjacency matrix representations [36, pp.418-422] fail to capture the particular nature of the variant graph as a non-linear text. However, a variant graph is not a general graph; it is strictly limited by the rules described above, and this enables it to be translated into a more compact format. This compact format is just a specially ordered list of the labels of each arc, namely a pair consisting of a fragment of perhaps empty text and a set of versions. This pairs-list representation is provably equivalent to the variant graph. It stores all of the graph's structure *by implication,* and hence it doesn't become overloaded even when a document consists of thousands of versions. To read any version all that is needed is to skip down the list, picking out pairs whose version-sets intersect with the version of interest. Figure 3 shows how part of the Magrelli example would look in this form. So reading a single version is O(N), where N is the number of pairs in the list.

Searching a pairs-list is almost as easy. Our multi-version document viewer application [35] uses the Karp-Rabin search [23], which calculates a rolling checksum that gets updated as each character is encountered, instead of the more familiar Boyer-Moore algorithm [36, p.286]. This is easier to implement in the variant-graph case and is also O(M+N) where M is the length of the pattern and N the length of the list of pairs. Each time the graph splits into two or more paths the current state of the search is split into separate threads or objects as required. Each search state is then run concurrently. Whenever the graph merges, the states also merge.

Comparison between versions is the main function of programs like MEDITE [18] and JUXTA [35], two desktop applications that display differences between two physically separate texts by calculating the differences, including transpositions, in real time. The two texts for comparison are displayed in adjacent windows and differences are indicated by highlighting etc. However, comparing two texts like this in real time can take up to an hour for long texts [4]. It is more efficient to perform any necessary calculations only once, store the result in a variant graph and then recover the differences by skimming the list for pairs belonging to one version that are not shared by another version. This is also O(N).

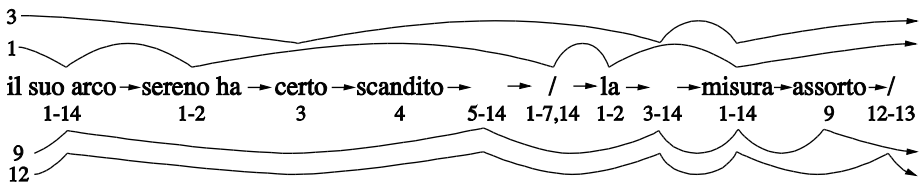

Figure 3

4.1 Creating and Editing a Variant Graph

Since the pairs-list representation is equivalent to the variant graph, the latter can be constructed by specifying a list of versions and a list of pairs, and loading them into computer memory. Viewing it only requires a standard editor and the version-reading algorithm of Figure 3.

A new version can be created by copying an existing version. Then any alterations to the text of the new version will become its unique text. When saving the new text any ordinary 'diff' program such as Ukonnen's [43] could be used to calculate new arcs where the text diverges from the parent version, as shown in Figure 4. This is O(ND), where D is the edit-distance between the two texts.

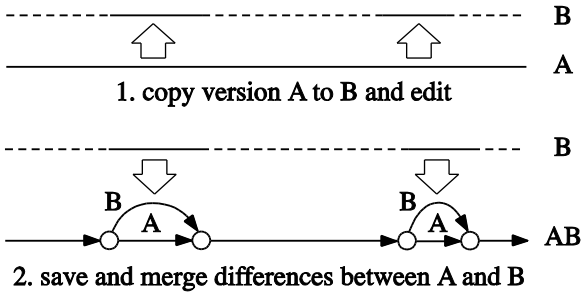

Figure 4

The same procedure could also be used to edit an existing version: differences between the old and new text of a version would be entered as new arcs for that version, and the similarities would simply remain as they were previously.

This almost manual approach, in which only two versions are merged at a time, is designed to avoid problems we have encountered in trying to automatically generate a variant graph from a set of N texts. This operation, called 'multiple sequence alignment' in biology, is frequently performed on sequences of amino acids or nucleotides [24,27]. However, when applied to cultural heritage texts, it appears to have two serious flaws:

1. Transpositions must be included, which makes the calculation NP-complete [28], necessitating a heuristic algorithm.
2. Automatic N-way alignment removes valuable human judgments about what is a variant of what and replaces them with a calculation based on edit-distance. That can only ever be an approximation of the real facts [26], which are already available from manual examination of the texts themselves.

Hence our future work will instead focus on developing a multi-version wiki that can be used in an ordinary web-browser, and which will use the variant graph only as a means of storing the resulting multi-version text. This will allow each version to be encoded using simple markup, as in a real wiki. In this way humans will retain control of the structure of the graph and the computational problem will remain tractable.

5. Conclusion

Multi-version documents correspond to the human notion of a 'work', representing it as a single, integrated digital entity. By cleanly separating content from variation it can leverage existing content-handling technologies such as XML. It has very good computational properties for reading, searching, comparing and editing multiple versions of a work in online presentation, and can accurately represent complex original documents of textual cultural heritage collections.

Bibliography

[1] D.T. Barnard, R. Hayter, M. Karababa, G. Logan, and J. McFadden. SGML-Based Markup for Literary Texts: Two Problems and Some Solutions. *Computers and the Humanities,* 22: 265-276,1988.

[2] P. Bart. Experimental markup in a TEI-conformant setting. *Digital Medievalist,* 2(1), 2006, http://www.digitalmedievalist.org/article.cfm?RecID=10.

[3] P.W. Berrie. Just In Time Markup for Electronic Editions. Apple University Consortium Conference in Wollongong Australia, http://auc.uow.edu.au/conf/conf00/papers/AUC2000_Berrie.pdf, 2000.

[4] J. Bourdaillet and J.G. Ganascia. Alignment of noisy unstructured text data. In *IJCAI Workshop on Analytics for Noisy Unstructured Text Data, Hyderabad, India*, pp. 139-146, http://research.ihost.com/and2007/cd/Proceedings_files/p139.pdf.

[5] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, F. Yergeau and J. Cowan (Eds). *Extensible Markup Language (XML) 1.1,* 2004, http://www.w3.org/TR/2004/REC-xml11-20040204/.

[6] D. Buzzetti. Digital Representation and the Text Model. *New Literary History*, 33(1): 61-88, 2002.

[7] J. Carletta, J. Kilgour, T.J. O'Donnell, S. Evert and H. Voormann. The NITE object model library for handling structured linguistic annotation on multimodal data sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web (NLPXML 2003)*. Budapest, Hungary, pp. 11-17, 2003.

[8] A.J. Catanese. Programming for Governmental Operations: The Critical Path Approach. *Public Administration Review,* 28(2): 155-167, 1968.

[9] N. Chomsky. On Certain Formal Properties of Grammars. *Information and Control,* 2: 137--167, 1959.

[10] A. Dekhtyar, I.E. Iacob, J.W. Jaromczyk, K. Kiernan, N. Moore and D.P. Carr. Support for XML Markup of Image-based Electronic Editions. *International Journal on Digital Libraries,* 6(1): 55--69, 2004.

[11] S. DeRose. Markup Overlap: A Review and a Horse. Extreme Markup Languages Conference, Montreal, http://conferences.idealliance.org/extreme/html/2004/DeRose01/EML2004DeRose01.html, 2004.

[12] D.G. Durand, E. Mylonas and S.J. DeRose. What Should Markup Really Be? Applying theories of text to the design of markup systems, ALLC/ACH Conference, Bergen, http://gandalf.aksis.uib.no/allc/durand1.pdf, 1996.

[13] P. Durusau and M.B. O'Donnell. Concurrent Markup for XML Documents. In *Proceedings XML Europe,* http://www.idealliance.org/papers/xmle02/dx_xmle02/papers/03-03-07/03-03-07.pdf, 2002.

[14] D. Ferrer. Hypertextual Representation of Literary Working Papers. *Literary and Linguistic Computing,* 10(2): 145-147, 1995.

[15] D. Fiormonte (Ed.). *Digital Variants,* 2007, http://www.digitalvariants.org, 2010.

[16] D. Fiormonte. Généalogie épistémologie de la variante: laGenetic Machine. *Genesis. Revue internationale de critique génétique,* 26: 173-176, 2006.

[17] D. Fiormonte and V. Martiradonna. La codifica digitale come atto ermeneutico e semiotico. Il caso di Valerio Magrelli. In C.B. Cazalé (Ed.*), Mémoire des textes - Textes de mémoire. Proceedings of the International conference, Université Paris X - Nanterre*. Centre de Recherches Italiannes and Presses Universitaires de Paris X, Nanterre, pp. 46-65, 2007.

[18] J.G. Ganascia, I. Fenoglio and J.L. Lebrave. Manuscrits, genèse et documents numérisés: EDITE : une étude informatisée du travail de l'écrivain. *Document numérique* 8(4): 91-110, 2004.

[19] C.F. Goldfarb. *The SGML Handbook,* Oxford University Press, Oxford, 1990.

[20] M. Hilbert. MuLaX - ein Modell zur Verarbeitung mehrfach XML-strukturierter Daten. Diploma Thesis, Universität Bielefeld, 2005; http://antareja.rvs.uni-bielefeld.de/mirco/pub/dipl/Diplomarbeit_2005-03-10.pdf, 2005.

[21] C. Huitfeldt. MECS - A Multi-Element Code System. *Working Papers from the Wittgenstein Archives at the University of Bergen* 3, Wittgenstein Archive, Bergen, 1993.

[22] C. Huitfeldt. Multi-Dimensional Texts in a One Dimensional Medium. *Computers and the Humanities,* 28: 235-241, 1995.

[23] R.M. Karp, and M.O. Rabin. Efficient Randomized Pattern-Matching Algorithms. *IBM Journal of Research Development*, 31(2): 249--260, 1987.

[24] K. Katoh, K. Misawa, K. Kuma and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on Fast Fourier transform. *Nucleic Acids Research*, 30(14): 3059-3066, 2002.

[25] P. Kilpelainen. SGML and XML Content Models. Markup Languages: Theory and Practice, 1(2): 53-76, 1999.

[26] T. Lassman and E.L.L. Sonnhammer. Quality assessment of multiple alignment programs. *FEBS Letters,* 529: 126-130, 2002.

[27] C. Lee, C. Grasso and M.F. Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics,* 18(3): 452-464, 2002.

[28] D. Lopresti and A. Tomkins. Block edit models for approximate string matching. *Theoretical Computer Science,* 181: 159-179, 1997.

[29] J.F. Maas. NEXUS: Vollautomatische Konvertierung mehrfach XML-annotierter Texte in das NITE-XML Austauschformat. Masters thesis, University of Bielefeld, Germany, 2003.

[30] J.J. McGann. *Radiant Textuality.* Palgrave McMillan, New York, 2001.

[31] J.J. McGann. Culture and Technology: The Way We Live Now, What Is to Be Done? *New Literary History,* 36(1): 71-82, 2005.

[32] PERT: Pert fundamentals: the fundamental concepts and techniques of Pert presented in a programmed instruction format. U.S.G.P.O., Washington, D.C., 1963.

[33] A. Renear, E. Mylonas and D. Durand. Refining our Notion of What Text Really Is: the Overlapping Hierarchies Problem. In N. Ide and S. Hockey (Eds.) *Research in Humanities Computing 4. Selected Papers from the ALLC/ACH Conference,* Christ Church Oxford, April 1992. Oxford University Press, Oxford, pp. 263-280, 1992.

[34] A. Renear. Out of Praxis: Three (Meta) Theories of Textuality. In K. Sutherland (Ed.), *Electronic Text.* Clarendon Press, Oxford, pp. 107-126, 1997.

[35] D.A. Schmidt and T. Wyeld. A novel user interface for online literary documents. ACM International Conference Proceeding Series, 122: 1-4, 2005.

[36] R. Sedgewick. Algorithms. Second Edition. Addison-Wesley, Reading, Massachusetts, 1988.

[37] D. Silvi and L. Geri. Le applicazioni di XML ai testi contemporanei: problemi di variantistica. In *Conference Abstracts Literatures, Languages and Cultural Heritage in a digital world.* King's College London, pp. 43-45, 2006.

[38] C.M. Sperberg-McQueen. Text in the Electronic Age. *Literary and Linguistic Computing* 1(6): 34-47, 1991.

[39] C.M. Sperberg-McQueen and C. Huitfeldt. GODDAG: A Data Structure for Overlapping Hierarchies. *Lecture Notes in Computer Science,* 2023: 139-160, 2004.

[40] C.M. Sperberg-McQueen. Rabbit/duck grammars: a validation method for overlapping structures. Extreme Markup Languages, Montréal, Quebec, http://conferences.idealliance.org/extreme/html/2006/SperbergMcQueen01/EML2006SperbergMcQueen01.html, 2006.

[41] C.M. Sperberg-McQueen and L. Burnard, (Eds).

[42] TEI P4: *Guidelines for Electronic Text Encoding and Interchange.* Text Encoding Initiative Consortium. XML Version, Oxford, Providence, Charlottesville, Bergen, 2002.

[43] E. Ukonnen. Algorithms for Approximate String Matching. *Information and Control,* 64: 100-118, 1985.

[44] E. Vanhoutte. Prose Fiction and Modern Manuscripts: Limitations and Possibilities of Text-Encoding for Electronic Editions. In L. Burnard, K. O'Brien O'Keefe and J. Unsworth (Eds.), *Electronic Textual Editing,* Modern Language Association, New York, pp. 161-180, 2006.

[45] L. Vetter and J. McDonald. Witnessing Dickinson's Witnesses. *Literary and Linguistic Computing,* 18: 151-165, 2003.

CONTACTS

Desmond Schmidt, Information Security Institute, Queensland University of Technology, 126 Margarent St., Brisbane, QLD, Australia. schmidda@qut.edu.au

Domenico Fiormonte, Dipartimento di Italianistica, Università Roma Tre, fiormont@uniroma3.it

BIOGRAPHY

Desmond Schmidt (PhD Cambridge, classical Greek literature) is a researcher at the Information Security Institute at the Queensland University of Technology in Brisbane, Australia.He recently completed a second PhD in Information Technology at the University of Queensland. Since 2002 he has been working with Domenico Fiormonte on ways to visualise and represent the complex stratified documents in the Digital Variants Archive.

Domenico Fiormonte has been working on digital texts since 1992 (PhD Edinburgh). He is currently Lecturer in Linguistics at the University of Roma Tre, where he teaches courses on Digital Editions, Digital Philology, and Writing for the New Media. He is author of Scrittura e filologia nell'era digitale, Turin, Bollati Boringhieri, 2003. His latest book, co-authored with Teresa Numerico and Francesca Tomasi, is L'umanista digitale, Il Mulino, Bologna, 2010.