Fabio Ciotti

LA RAPPRESENTAZIONE DIGITALE DEL TESTO: IL PARADIGMA DEL MARKUP E I SUOI SVILUPPI[1]

Introduzione

La rappresentazione digitale del testo ha costituito uno dei temi fondanti dell'informatica umanistica, sia dal punto di vista teorico, sia da quello pratico. Non è difficile individuare la ragione di tale centralità: i testi, nelle loro varie forme materiali e immateriali, sono tanto i principali oggetti quanto i principali strumenti di studio per buona parte delle discipline umanistiche. E oltre che farne oggetto della corrente attività di ricerca, la comunità scientifica deve anche garantire la tradizione dei testi nel tempo, come rileva Claus Huitfeldt:

> For humanities research it is therefore important not just to facilitate the exchange and reuse of the texts that record the results of research, but also to ensure that texts produced in very different contexts can be preserved in a form that will make them accessible also to research in the future. [2]

La riflessione in questo ambito – quasi sempre giustamente stimolata dalle pratiche concrete – ha evidenziato numerosi e complessi problemi teorici, oggetto di una vasta bibliografia. Il nodo della questione deriva dalla tensione che si stabilisce tra i due poli della rappresentazione: testo e rappresentazione digitale.

Il testo è un oggetto dalla ontologia complessa, a un tempo oggetto intellettuale, oggetto semiotico/linguistico e artefatto materiale; in quanto tale è in grado di veicolare o costruire significato (e di

[1] Questo lavoro estende emenda e rielabora il capitolo "La rappresentazione complessa di testi complessi" di F. Ciotti, *Il testo e l'automa*, Aracne, Roma 2007.

[2] C. Huitfeldt, *Scholarly Text Processing and Future Markup Systems*, "Jahrbuch fiir Computerphilologie", 5 (2003); <http://www.computerphilologie.uni-muenchen.de/jg03/huitfeldt.html>.

rivestire dunque interesse scientifico) su più livelli, e cosa ancora più rilevante attraverso l'instaurazione di molteplici relazioni tra più livelli. Come rileva Jerome McGann questa complessità viene esplicitamente tematizzata dai testi letterari e poetici, i quali, dunque, assumono un ruolo privilegiato nella nostra comprensione della comunicazione testuale:

> Considered strictly in term of bibliographical codes, then, poetical works epitomize a crucial expressive feature of textuality in general: that it can be seen to organize itself in terms of various relational segmentations and metasegmentations. Some elementary segmentations are sentences, paragraphs, chapters; in verse lines, inter- and intralinear forms (rhyme, for example, and metrical forms) stanzas, cantos; in the page, the opening, the book. These segmentations may be usefully traced to the level of the individual character and, in general, to font and typeface design. [3]

Quando parliamo di rappresentazione (e a maggior ragione di elaborazione) digitale d'altra parte, ci riferiamo a un insieme di sistemi o linguaggi di rappresentazione formali (nella tradizione dei formalismi logici e matematici da cui derivano almeno in parte) tra loro organizzati in una gerarchia di livelli perfettamente isomorfi (a partire dalla notazione binaria fino ai linguaggi e formalismi di livello alto), elaborabili mediante procedure automatiche. Naturalmente questa concezione astratta della rappresentazione digitale non esaurisce completamente le modalità attraverso cui un oggetto digitale si presenta al suo fruitore. Anzi, di norma, la natura formale di un oggetto digitale resta occultata dietro la raffigurazione o presentazione che lo trasforma in un oggetto percettivo; ma non si deve dimenticare che tale processo di fenomenizzazione è a sua volta una operazione formale, i cui dettagli tecnici sono del tutto invisibili agli utenti finali (per nostra fortuna, potremmo dire), che agisce sulla rappresentazione formale dell'oggetto digitale[4].

[3] J. McGann, *Radiant Textuality: Literature after the World Wide Web*, Palgrave, New York 2001, p. 183

[4] Sulla complicata natura del digitale si veda lo stimolante articolo di G. Rockwell, *Interrupting Digitization and Thinking about Text Or Digitization and the Form of Digital Text*, "Informatica Umanistica", 2 (2010); <http://www.ledonline.it/informaticaumanistica/Allegati/IU 02-09 Rockwell.pdf>.

In generale dunque la digitalizzazione del testo comporta la sua rappresentazione o meglio modellizzazione[5] in un qualche linguaggio formale. Il fatto è che non disponiamo di una teoria completa del testo, e nemmeno teorie di uno o più livelli testuali, che si possano definire in senso stretto formali (con le dovute eccezioni: alcune teorie linguistiche ad esempio, o alcune teorie meccaniciste della critica testuale). La storia della codifica informatica dei testi consiste nel tentativo di superare questo duplice difficoltà teorica e tecnica:

1. sviluppare teorie e modelli formali del testo (o di alcuni suoi livelli)
2. individuare formalismi atti a esprimerli in forma computazionalmente accettabile[6]

Alle questioni teoriche e metodologiche cui abbiamo accennato, si sono affiancate nel corso degli anni anche considerazioni di tipo socio-tecnologico e socio-organizzativo relative alla concreta praticabilità e sostenibilità dell'adozione di formalismi complessi e di difficile manutenzione, al loro impatto nelle pratiche e nelle attitudini delle comunità scientifiche umanistiche a cui sono rivolti, e alla capacità di preservare nel tempo le risorse testuali digitali prodotte per finalità scientifiche

I linguaggi di markup e XML: il paradigma vincente

L'evoluzione delle tecnologie informatiche ha portato allo sviluppo di diverse e sempre più evolute tecnologie di *text-processing*. Va tuttavia detto che nella gran parte queste tecnologie non sono state sviluppate tenendo conto delle esigenze di trattamento di dati testuali in ambito scientifico e umanistico in particolare, ma di quelle dell'applicazione dell'informatica documentale nei processi produtti-

5 Sulla centralità del concetto di modellizzazione oltre che ai nostri precedenti scritti, rimandiamo all'importante monografia di W. McCarty, *Humanities Computing*, Palgrave, London and New York 2005, in particolare al primo capitolo.

6 Si noti che per paradosso linguistico un formalismo (o un algoritmo) è tanto più utilizzabile quanto è meno complesso nel senso assunto dal termine nella teoria della complessità computazionale. Dove la complessità è una misura delle risorse temporali o spaziali (considerate indipendentemente dalla potenza e velocità di calcolo della macchina) necessarie a elaborarlo. Detto in termini semplici, è inutile avere formalismi la cui computazione potrebbe essere irragionevolmente lunga o esosa, ovvero intrattabile.

vi e gestionali. La famiglia di strumenti basati sul paradigma WYSIWYG, i *word processor* e i sistemi di *desktop publishing* – di gran lunga i più diffusi, anche in ambito scientifico, per la redazione di documenti – si sono ben presto rivelati inadeguati per un uso su vasta scala nelle iniziative di digitalizzazione del patrimonio testuale con fini scientifici e di conservazione, per diversi e fondati motivi.

Ben più interessante invece si è rivelato il paradigma basato sull'uso di *linguaggi di markup*, paradossalmente il più antico tra i sistemi di elaborazione testuale messi a punto nella storia dell'informatica[7]. In particolare l'incontro tra comunità umanistica e tecnologie documentali è stato favorito dalla introduzione dei linguaggi di markup di tipo descrittivo, il cui primo esemplare è stato SGML (*Standard Generalized Markup Language*). Il diretto successore di questo formalismo, XML (*Extensible Markup Language*), è allo stato (quasi) universalmente riconosciuto come la soluzione teoricamente più adeguata, e la più adottata nella pratica, dalla comunità scientifica e professionale dedita allo studio e alla conservazione (nonché all'accrescimento) del patrimonio testuale[8].

Il successo di XML presso la comunità scientifica umanistica è dovuto a diverse ragioni:

1. sufficiente semplicità di uso
2. apertura e standardizzazione
3. indipendenza da particolari piattaforme hardware e software, e di conseguenza maggiore portabilità e facilità di manutenzione e preservazione delle risorse digitali prodotte
4. disponibilità ampia di software e applicazioni, in massima parte *open source*, per la gestione ed elaborazione di dati in formato XML
5. flessibilità e adattabilità a diversi contesti e fini di utilizzo, anche di tipo scientifico e di ricerca, e al tempo stesso possibilità di definire in modo rigoroso schemi e vocabolari di

7 I primi sistemi e linguaggi di markup risalgono alla seconda metà degli anni 60 del secolo scorso.

8 La bibliografia su SGML, XML e sui linguaggi di markup è sterminata. Rimandiamo a F. Ciotti (ed.), *Il manuale TEI Lite*, Bonnard, Milano 2004 per una panoramica e una prima indicazione bibliografica su questi temi, nonché ovviamente sulla Text Encoding Initiative, per la quale si possono consultare i documenti disponibili sul sito del TEI Consortium (http://www.tei c.org).

codifica dei dati (mediante *schema language*) e procedure di validazione e controllo dei dati
6. fondatezza ed eleganza formale del linguaggio, del suo modello di dati e delle sue estensioni

Conseguenza e al tempo stesso motore del successo di XML in ambito umanistico sono stati anche l'istituzione e la diffusione della *Text Encoding Initiative*, un progetto scientifico internazionale volto a definire uno schema di codifica testuale comune e condiviso per le iniziative di ricerca in tale ambito. Una delle decisioni fondative del progetto fu proprio quella di adottare SGML come formalismo di base, scelta ribadita successivamente con il passaggio al successore XML. A distanza di venti anni dall'avvio del progetto si può dire che la TEI, superando non poche diffidenze, resistenze e persino ostracismi culturali, rappresenta un punto di riferimento ineludibile per tutte le imprese volte a digitalizzare risorse testuali in ambito umanistico per fini di ricerca e di conservazione digitale.

Tuttavia da più parti e con periodica insistenza l'affermazione di XML e in generale del paradigma del markup negli studi umanistici digitali ha sollevato dubbi, critiche e anche aspri rifiuti. Possiamo riassumere nei seguenti punti i nodi problematici principali che emergono da tali critiche:

- XML è inadeguato per rappresentare le nuove forme della testualità digitale, che sono ontologicamente non lineari e multimodali
- XML è semanticamente agnostico e dunque non permette di progettare applicazioni avanzate di analisi e gestione dei testi
- XML è intrinsecamente inadeguato a rappresentare le caratteristiche complesse del testo, quelle che rivestono vero interesse nella ricerca umanistica

Sul primo punto le critiche sono di fatto giustificate (cioè è vero che molte delle sperimentazioni testuali/letterarie con mezzi digitali ignorano XML e le sue potenzialità), ma non hanno fondamenti teorici. Dopo la sua formalizzazione da parte del Web Consortium[9],

[9] Il raggruppamento no profit di aziende, centri di ricerca e singoli studiosi promosso da Tim Berners Lee, responsabile dello sviluppo delle principali tecnologie che rendono possibile il funzionamento del World Wide Web (http://www.w3.org).

infatti, XML è stato affiancato da numerosi linguaggi e tecnologie di supporto che lo rendono perfettamente in grado di fornire tutti gli strumenti per rappresentare e gestire oggetti ipertestuali e multimodali: si pensi a XLink e XPointer per l'espressione di link e collegamenti; SMIL ed EMMA per la sincronizzazione e l'organizzazione di flussi e informazioni multimodali; RDF per la formalizzazione di complesse relazioni tra oggetti e risorse informative; SVG per la codifica di immagini vettoriali e così via. Anzi, va ricordato come di recente la *Electronic Literature Organization*[10] abbia avviato il progetto *Preserving, Archiving and Dissemination*, volto alla preservazione nel tempo dell'accesso alle nuove opere digitali, indicando nell'adozione di XML uno dei punti strategici per il perseguimento di questa finalità[11].

Venendo alla questione dei "limiti semantici" di XML, si tratta di un problema certamente giustificato tecnicamente e teoricamente, ma si deve osservare come anche in questo caso la sua natura di "problema" derivi da una sorta di incomprensione di fondo. Infatti XML di per sé non può garantire l'elaborazione e l'interoperabilità semantica dei dati sebbene spesso si senta affermare il contrario. Come afferma giustamente Robin Cover:

> XML is a poor language for data modeling if the goal is to represent information objects in the problem domain such that they correspond transparently ('one-to-one') to the user's conceptual model of objects in this domain.[12]

XML permette di esprimere semplici relazioni strutturali:

- gerarchia (A contiene B)
- adiacenza (A seguito da B)
- co-occorrenza (se A allora [anche/non] B)

10 La Electronic Literature Organization, fondata nel 1999, è una organizzazione no profit, che comprende scrittori, artisti, insegnanti, studiosi, e sviluppatori, il cui fine è promuovere la lettura, la scrittura, l'insegnamento e la comprensione della letteratura creata e disseminata in ambiente digitale (http://www.elo.org).

11 N. Montfort, N. Wardrip Fruin, *Acid Free Bits. Recommendations for Long Lasting Electronic Literature*, ELO, 2004; < http://www.eliterature.org/pad/afb.html>.

12 R. Cover, *XML and Semantic Transparency*, in Cover Pages, 1998; <http://xml.coverpages.org/xmlAndSemantics.html>.

Con l'introduzione degli *schema language* si è aggiunta la possibilità di "tipazione forte" dei valori di elementi e attributi. Ma XML non aggiunge senso ai dati (almeno non lo aggiunge in modo computazionalmente trattabile). L'errore deriva da una sorta di fallacia interpretazionale dovuta al fatto che i marcatori XML sono *human-readable* e che, di norma, il vocabolario dei linguaggi XML usa termini delle lingue naturali. Ma la semantica "naturale" di tale vocabolario è del tutto inaccessibile a un elaboratore XML. Per un *parser* i seguenti frammenti di markup sono entrambi perfettamente accettabili:

- <title>Il fu Mattia Pascal</title>
- <blob>Il fu Mattia Pascal</blob>

In definitiva, è la mente dell'agente umano che legge il documento XML a fornire un significato al markup[13].

D'altra parte questo indifferenza semantica può rappresentare un vantaggio in molte situazioni applicative: ad esempio garantisce la massima accoglienza per il formalismo nei contesti più diversi (la presenza di primitive semantiche forti rappresenterebbe infatti un fattore limitante). Normalmente il contesto semantico di un linguaggio di markup viene definito attraverso la stipulazione di modi di uso condivisi in specifiche comunità di uso di un determinato linguaggio di markup, eventualmente rafforzati dalla predisposizione di documentazione normativa in lingua naturale (si pensi ad esempio alle *Guidelines* della Text Encoding Initiative). Nondimeno la disponibilità di un qualche sistema formale per la definizione di vincoli semantici per un linguaggio XML sarebbe certamente utile, e in questa direzione si sono indirizzati di recente alcuni progetti di ricerca che hanno individuato nei linguaggi di programmazione logica come Prolog, o nei formalismi del Semantic Web (RDF e OWL) le possibili soluzioni tecniche da esplorare[14].

13 Cfr. T. Bray, *On Semantics and Markup*; <http://www.tbray.org/ongoing/When/-200x/2003/04/09/SemanticMarkup>.

14 Cfr. Y. Marcoux, É. Rizkallah, *Intertextual semantics: A semantics for information design*, "Journal of the American Society for Information Science & Technology", 60/9 (2009), pp. 1895-1906; <doi:10.1002/asi.21134>; D. Dubin, C. M. Sperberg McQueen, A. Renear, C. Huitfeldt, *A logic programming environment for document semantics and inference*, "Literary and Linguistic Computing", 18/2 (2003), pp. 225-233; G.

Il problema dei limiti di XML nella rappresentazione di strutture complesse, infine, ha fornito la linea di critica di gran lunga più importante contro la sua adozione in campo umanistico. Su questo aspetto dunque ci concentreremo nelle seguenti sezioni del presente lavoro.

I limiti rappresentazionali di XML

Le critiche volte a sostenere l'inadeguatezza rappresentazionale di XML hanno a loro volta assunto due punti di vista quasi opposti: secondo alcuni studiosi XML e SGML sono formalismi fortemente strutturati che forzano la natura intrinseca del testo, la sua essenza, con vincoli e strutture a essa estranei; secondo i fautori del punto di vista opposto XML è invece un formalismo troppo debole per poter rappresentare le complesse strutture del testo, e in particolare del testo letterario. Nella letteratura critica questi due punti di vista spesso coesistono e sono motivati a partire da prospettive teoriche le più varie.

A queste critiche la comunità di studiosi che più apertamente propende per l'adozione di XML – in particolare la comunità raccolta intorno alla TEI – ha risposto con una certa quantità di lavori volti a fornire giustificazioni teoriche per questa scelta. Tale attività teorica ha dato luogo a teorie metafisicamente impegnative, di cui la più nota è quella ormai nota come *OHCO* (*Ordered Hierarchy of Content Object*)[15]. Questa teoria alla domanda "che cosa è un testo veramente" risponde in modo assertorio che si tratta di un oggetto linguistico astratto organizzato secondo una struttura gerarchica ordinata di *oggetti di contenuto*.

Tummarello, C. Morbidoni, F. Kepler, F. Piazza, P. Puliti, *A novel Textual Encoding paradigm based on Semantic Web tools and semantics*, 5th International Conference on Language Resources and Evaluation, 2006; <http://www.sdjt.si/bib/lrec06/pdf/-225_pdf.pdf>.

15 La prima formulazione di questa teoria è in S.J. DeRose, Durand, D., Mylonas, E., Renear, A.H., *What is Text, Really*?, "Journal of Computing in Higher Education", 1/2 (1990), pp. 3-26; la sua revisione in A. Renear, Durand D., Mylonas E., *Refining Our Notion of What Text Really Is: The Problem of Overlapping Hierarchies*, in N. Ide, S. Hockey (eds.), *Research in Humanities Computing*, Oxford University Press, Oxford, 1996; versione preliminare 1992, <http://www.stg.brown.edu/resources/stg/-monographs/ohco.html>. Su questi temi rimandiamo anche a F. Ciotti, *Il testo e l'automa*, Aracne, Roma 2007.

Gli oggetti di contenuto a cui si fa riferimento sono sostanzialmente le strutture astratte di cui si compone un testo: capitoli, paragrafi, citazioni, enfasi, o poesie, strofe, versi, etc. Essi sono gerarchici poiché alcuni degli oggetti testuali contengono altri, e ordinati in quanto esiste sempre una relazione lineare di successione tra due oggetti posti sul medesimo livello gerarchico. La specificazione degli oggetti è determinata dal genere il testo. Reciprocamente un genere testuale è individuato dalla classe di oggetti di contenuto che contiene: il testo poetico è tale perché si articola in sequenze di elementi che chiamiamo «versi», le quali, per inciso, non hanno assolutamente nulla a che fare con la struttura verbale del testo. Se volessimo trascrivere un poema dovremmo essere in grado di segnalare questa struttura. Il medesimo discorso potrebbe essere fatto per un testo drammatico: la distinzione tra battute e didascalie è essenziale per comprendere il testo. Alcune caratteristiche sono comuni a tutti o a molti di questi tipi di testi, mentre altre sono assolutamente specifiche.

La fallacia essenzialista di questa teoria generale è stata ben presto riconosciuta dai suoi stessi estensori, che ne hanno proposto negli anni diverse revisioni più aperte e pluraliste, ma l'idea di una preminenza della struttura gerarchica nella testualità ha mantenuto un ruolo descrittivo ed esplicativo essenziale.

La ragione di tanto attaccamento all'idea di struttura gerarchica ovviamente non è immotivata. Il fatto è che XML può essere considerato sia un *formalismo* sia un *modello di dati* espresso da quel formalismo, e tale (meta)modello è appunto un *albero ordinato etichettato*[16]. La notazione di XML come noto permette di esprimere linearmente (*serializzare*) tale struttura mediante l'uso di coppie di marcatori parentetici annidati (in modo simile a quanto avviene nella notazione parentetica in algebra).

In altri termini XML considerato come linguaggio di modellizzazione può esprimere solo modelli la cui struttura è un albero. Di conseguenza tutte le applicazioni XML standard sono in grado di elaborare (validare, trasformare, formattare) strutture dati ad albero. É vero che come puro formalismo sintattico XML può essere utilizzato per codificare efficacemente strutture e modelli di dati non

16 Il Web Consortium ha rilasciato una Recommendation che formalizza il data model di XML: XML Information Set (Second Edition); <http://www.w3.org/TR/xml infoset/>.

strettamente gerarchici – e in questo di fatto risiede gran parte del suo successo come notazione standard per l'interoperabilità sintattica. Tuttavia se si adotta XML come puro linguaggio di serializzazione le possibilità di elaborazione fornite da applicazioni XML standard su tali dati sono assai ristrette (di fatto ristrette a ciò che può essere trattato come albero). Il prezzo costituito dall'adozione di un modello di dati così vincolante, d'altra parte, paga il vantaggio di potere validare in modo automatico ogni istanza di dati rispetto al modello mediante algoritmi generali ben conosciuti e computazionalmente trattabili, ciò che a sua volta consente di costruire sistemi di elaborazione degli stessi dati consistenti ed efficaci[17].

Naturalmente se si considerano le finalità applicative prioritarie per i quali sia SGML sia XML sono stati sviluppati, si può ben comprendere come nella comunità informatica non si sia avuta alcuna remora a considerare questo vantaggio un ben valido contraccambio per il prezzo pagato. In effetti nell'elaborazione informatica dei documenti in ambito commerciale e industriale (cioè volta a produrre, archiviare e disseminare in modo efficiente e sicuro documentazione in formato digitale), tale prezzo era in realtà nullo: ogni documento testuale può essere ragionevolmente descritto come un albero da un qualche punto di vista, basta adottare il livello giusto di segmentazione e astrazione.

Ma dobbiamo osservare che anche considerando testi complessi come quelli oggetto della ricerca umanistica è innegabile che il modello gerarchico si presti naturalmente a descrivere la struttura di numerosi livelli testuali scientificamente rilevanti: i livelli editoriale, morfosintattico, metrico, tanto per fare alcuni esempi, sono in linea generale descrivibili come strutture gerarchiche ordinate. Insomma in molti contesti e per numerose finalità di ricerca la rappresentazione del testo come una struttura gerarchica è, se non ontologicamente corretta, almeno metodologicamente accettabile.

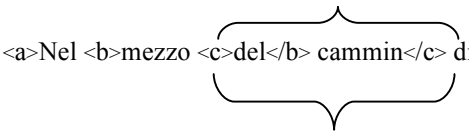
I problemi invece sono emersi quando la comunità degli studiosi di scienze del testo con tutto il loro bagaglio di teorie e interpretazioni del concetto di testo hanno pensato di eleggere XML a forma-

17 A dire il vero l'affermazione che XML possa modellizzare adeguatamente solo alberi può essere attenuata in virtù della capacità di esprimere relazioni non gerarchiche tra elementi mediante coppie di attributi ID/IDREF. Questi costrutti permettono sia di validare sia di elaborare relazioni strutturali non strettamente gerarchiche tra gli elementi di un documento XML. Tuttavia la capacità espressiva di questi costrutti è assai limitata.

lismo generale per la rappresentazione complessa dei loro oggetti di analisi. La manifestazione di queste difficoltà sono state comunemente rubricate come il *problema delle gerarchie sovrapposte* (*overlapping hierarchies*).

Il problema delle gerarchie sovrapposte

La prima formulazione del problema, corredata da alcuni casi esemplari si trova in un noto articolo di T. Barnard e altri intitolato *Hierarchical Encoding of Text: Technical Problems and SGML Solutions*[18]. In termini tecnici il problema delle gerarchie sovrapposte consiste nel fatto che, dati due oggetti logici presenti in un testo, le coppie di marcatori bilanciati che li rappresentano non si annidano propriamente ma si sovrappongono.



Questo corrisponde alla presenza di almeno due sotto-alberi (uno con il ramo a➝b e uno con quello a➝c) che descrivono contemporaneamente la struttura del documento. Tale situazione è sintatticamente e semanticamente vietata in XML.

Dal punto di vista concettuale il problema delle gerarchie sovrapposte è un sottoinsieme del più generale problema della complessità testuale, come a più riprese hanno fatto notare autori come Dino Buzzetti e Jerome McGann[19]:

> My own comparison is itself a kind of joke, of course, for an SGML model of the world of textualities pales in comprehensiveness before the Newtonian model of the physical world. But the outrageousness of the comparison in each case helps to clarify the situation. No autopoietic process or form can be simulated under the horizon of a structural model like SGML, not even topic maps. We see this very clearly when

[18] T. Barnard et al., *Hierarchical Encoding of Text: Technical Problems and SGML Solutions*, "Computers and the Humanities" , 29 (1988), pp. 211-231.

[19] Si veda D. Buzzetti, *Digital Representation and the Text Model*, "New Literary History", 33 (2002), pp. 61-88; D. Buzzetti, *Digital Edition and Text Processing*, in M. Deegan, K. Sutherland (eds.), *Text Editing, Print and the Digital World*, Ashgate, Aldershot 2009, pp. 45-61; D. Fiormonte, *Testo digitale, Semiotica, Rappresentazione*, "Informatica Umanistica", 3 (2010).

> we observe the inability of a derivative model like TEI to render the forms and functions of traditional textual documents. [20]

Il modello di dati di XML, infatti, presuppone che esista almeno un livello testuale per il quale si possa individuare e formalizzare una struttura chiara di elementi testuali e di relazioni formali tra questi elementi, o detto altrimenti, che esista un modello formale di quel livello testuale (ed eventualmente delle relazioni tra più livelli). Ma questo come già si notava è un presupposto irrinunciabile per adottare metodologie informatiche nella rappresentazione e analisi computazionale dei testi. Se esistono proprietà dei testi irriducibili a qualsiasi formalizzazione anche minimale, allora queste non possono per definizione essere rappresentate e trattate con metodi computazionali[21].

In realtà sotto l'etichetta di *gerarchie sovrapposte* sono stati collocati problemi rappresentazionali di natura e difficoltà diversa, anche se non sempre si tratta propriamente di fenomeni determinati dalla presenza di livelli gerarchici concorrenti.

Il tipo più semplice e comune è quello proprio, la compresenza di due o più strutture gerarchiche i cui elementi si sovrappongono. Rientrano in questo casistica la sovrapposizione della struttura fisica del documento con quella logica del testo, quella tra struttura logica del testo drammatico in verso e struttura metrica, quella tra struttura metrica e struttura sintattico/linguistica del discorso. Nell'ambito di questa classe è possibile individuare una scala crescente di complessi-

[20] J. McGann, *Marking Texts of Many Dimensions*, in S. Schreibman, R. Siemens, J. Unsworth (eds), *A Companion to Digital Humanities*, Blackwell, Oxford 2004; <http://www.digitalhumanities.org/companion/>.

[21] Si noti che questo vale anche per la creazione di facsimili digitali in formato immagine delle pagine di un libro. Infatti:

- nonostante le apparenze anche l'immagine digitale è un modello formale regolato da precise proprietà matematiche;
- tutto ciò che si può fare su una immagine digitale con metodologie strettamente computazionali deriva dalle proprietà formali del modello e dalla individuazione di algoritmi che possano manipolare tali proprietà.

In effetti i limiti attuali dell'elaborazione di immagini permettono di fare molto poco con tali facsimili: tutta l'elaborazione veramente interessante che si può fare su di esse viene eseguita dall'agente umano che vi accede attraverso un qualche dispositivo e supporto di output. Ovvero, l'immagine digitale di un manoscritto è veramente utilizzabile solo dallo studioso che la osserva e la interpreta, né più né meno che se osservasse l'originale o un facsimile su carta o su microfilm. Ovviamente questo è vero ora stante l'attuale livello di sviluppo delle tecnologie di image processing. Sul futuro è meglio non fare previsioni.

tà in ragione del numero di livelli descittivi compresenti e del numero di elementi di un dato livello che vengono attraversati da un elemento di un altro livello.

Un caso concettualmente più complicato è quello di un elemento appartenente a una gerarchia che si estende oltre i confini dell'elemento in cui inizia o persino di uno dei suoi predecessori (citazioni, annotazioni, fenomeni materiali nella trascrizioni di fonti primarie). In questo esempio (dalla *Commedia*) il discorso diretto si sovrappone con la segmentazione in versi

> Rispuosemi: "Non omo, omo già fui,
> e li parenti miei furon lombardi,
> mantoani per patrïa ambedui.

Tecnicamente simile ma concettualmente distinto il caso di elementi composti da segmenti discontinui e non contigui (di nuovo discorsi diretti, annotazioni, cancellature o aggiunte, fenomeni materiali), come nel seguente frammento dal *Turno* di Pirandello dove l'enunciato diretto di donna Rosa è inframmezzato al livello diegetico della narrazione:

> – In nome del Padre, del Figliuolo e dello Spirito Santo,
> – si lamentava intanto, in casa, la moglie
> del Ravì, la si- donna Rosa, accennando il segno della
> croce con un gesto che le era abituale e che ripeteva
> ogni qual volta si sentiva infastidita e urtata nella
> gravezza della sua gialla carne inerte: – Lasciatelo
> fare. Ciò che fa Marcantonio, per me, è ben fatto, –
> diceva ai parenti che sottovoce le facevan notare
> la mostruosità di quel progetto di nozze.

Infine il caso più complesso si verifica quando un dato fenomeno testuale può generare indefinite auto-sovrapposizioni degli elementi XML che lo rappresentano. Rientrano in questa casistica diverse tipologie di fenomeni di particolare interesse nelle applicazioni filologiche e critico-letterarie della codifica, come la creazione di edizioni diplomatiche e critiche o l'annotazione tematica. Ad esem-

pio nell'ottavo periodo del capitolo VIII del *De Principatibus* nell'edizione di Giorgio Inglese troviamo le seguenti varianti:

- lasciato parte della sua gente {(<alla [defesa) della> obsidione]}, con le
  - lasciato parte della sua gente <alla offesa delle> obsidione, con le
  - lasciato parte della sua gente alla [difesa di quella], con le
  - lasciato parte della sua gente (difesa) della obsidione, con le
  - gente {allobsidione}, con le

Le parentesi mostrano come il testo stabilito dal curatore vari rispetto alle diverse lezioni attestate nelle famiglie di testimoni adottati per la ricostruzione dello stesso: un ipotetico elemento XML `<variante>` usato per registrare tali varianti testuali dovrebbe necessariamente auto-sovrapporsi.

Il problema delle gerarchie sovrapposte, come si diceva prima, non esaurisce completamente le difficoltà della formalizzazione digitale del testo. Tuttavia esso rappresenta una consistente "minaccia teorica" interna al paradigma della codifica testuale basata sui linguaggi di markup strutturati, la cui rilevanza è ben chiara a uno dei massimi esponenti di questa scuola, Michael Sperberg-McQueen:

> It is an interesting problem because it is the biggest problem remaining in the residue. If we have a set of quantitative observations, and we try to fit a line to them, it is good practice to look systematically at the difference between the values predicted by our equation (our theory) and the values actually observed; the set of these differences is the residue …. In the context of SGML and XML, overlap is a residual problem. [22]

Non è un caso che negli ultimi dieci anni, proprio in parallelo con la diffusione della TEI nella comunità umanistica, si siano moltiplicati i tentativi di trovare delle soluzioni praticabili per questa difficoltà rappresentazionale, che si possono dividere in due classi: soluzioni interne al paradigma XML e soluzioni che portano al superamento di XML.

[22] C.M. Sperberg McQueen, *What matters?, Closing keynote. Extreme Markup Languages*, 2002; <http://www.w3.org/People/cmsmcq/2002/whatmatters.html>.

Nella classe delle soluzioni interne rientrano una serie di espedienti di codifica che preservano la conformità sintattica a XML (e dunque le proprietà di buona formazione e validità dei documenti prodotti). Tuttavia in molti casi essi sfruttano la possibilità di utilizzare XML come puro formalismo di serializzazione in grado di esprimere fenomeni e modelli di dati non gerarchici (fondamentalmente grafi orientati connessi, di cui gli alberi sono un sottoinsieme). Di conseguenza, sebbene le seguenti strategie siano tutte sintatticamente conformi a XML, non è detto che esse siano sempre logicamente trattabili con applicazioni XML standard[23]. Inoltre il difetto di queste soluzioni conservative, per così dire, è la estrema verbosità del markup che si ottiene applicandole, e una costante infrazione del principio del rasoio di Occam: gli elementi sintattici del linguaggio vengono infatti moltiplicati a dismisura rispetto ai soggiacenti fenomeni che intendono rappresentare. Ne derivano infine una onerosa applicazione e manutenzione.

La prima strategia di questo tipo è quella della *segmentazione:* un elemento logico che si sovrappone ai confini di un altro (o di più altri) viene diviso in due (o più) elementi XML dello stesso tipo correlati mediante apposti attributi. Ad esempio, in questo modo si potrebbe trattare il caso delle strutture editoriali e metriche concorrenti di un testo teatrale in versi (esempio tratto dall'*Antigone* di Alfieri):

```
<sp>
<speaker><emph>Argia</emph></speaker>
<l part="I">Una infelice io sono.</l>
</sp>
<sp>
<speaker><emph>Antigone</emph></speaker>
<l part="F">In queste soglie</l>
<l part="I">che fai? che cerchi in sì
tard'ora?</l>
</sp>
<sp><speaker><emph>Argia</emph></speaker>
<l part="F">Io... cerco...</l>
<l part="I">... d'Antigone...</l>
</sp>
<sp><speaker><emph>Antigone</emph></speaker>
<l part="F">Perché? — Ma tu, chi sei?</l>
```

23 Naturalmente nulla vieta di sviluppare procedure ad hoc, ovviamente rinunciando alla universalità e indipendenza dal software di cui gode XML.

```
<l>Antigone conosci? a lei se' nota?</l>
<l>che hai seco a far? che hai tu comun con
essa?</l>
</sp>
```

Questa soluzione permette di gestire i casi più semplici di sovrapposizioni tra due livelli e di discontinuità[24]. Tuttavia è impossibile da utilizzare se sussistono molteplici sovrapposizioni sulla stessa sequenza di caratteri. Inoltre costringe surrettiziamente ad eleggere una e una sola gerarchia come primaria.

Una evoluzione del sistema basato sulla segmentazione consiste nell'introduzione di elementi di congiunzione. Si tratta di elementi XML che hanno la funzione metatestuale di esprimere l'unicità semantica di un fenomeno testuale codificate mediante più elementi XML distinti (non necessariamente dello stesso tipo). Ad esempio il caso di un discorso diretto frammentario si potrebbe gestire in questo modo:

```
<p>- <q id="q1">In nome del Padre, del Figliuolo
e dello Spirito Santo,</q> - si lamentava
intanto, in casa, la moglie del Ravì, la si-donna
Rosa, accennando il segno della croce con un
gesto che le era abituale e che ripeteva ogni
qual volta si sentiva infastidita e urtata nella
gravezza della sua gialla carne inerte: - <q
id="q2">Lasciatelo
fare. Ciò che fa Marcantonio, per me, è ben
fatto,</q>- diceva ai parenti che sottovoce le
facevan notare la mostruosità di quel progetto di
nozze.</p>
…
<join targets="q1 q2" result="q"/>
```

Questa tecnica consente di trattare in linea teorica ogni caso di sovrapposizione e di non contiguità, di ordinamento inverso e di relazione n-aria tra oggetti testuali. Tuttavia si presenta anch'essa assai prolissa, specialmente se i livelli di sovrapposizione sono numerosi, e di difficile manutenzione. Essa inoltre presuppone che il testo

[24] Inoltre le soluzioni basate sulla segmentazione e l'uso di attributi di collegamento possono essere parzialmente validata mediante parser XML standard adottando oculatamente attributi ID/IDREF, dettaglio tecnico su cui in questa sede sorvoliamo.

sia segmentato esplicitamente in elementi base da collegare, ciò che potrebbe portare a una moltiplicazione dei marcatori da inserire nel testo con fini esclusivamente sintattici.

Le tecniche basate su elementi di congiunzione possono facilmente essere implementate mediante un approccio di *stand-off markup*. XML, come gran parte dei linguaggi di markup che lo hanno preceduto, presuppone che i marcatori siano inseriti all'interno della sequenza di caratteri che costituisce il contenuto linguistico del testo. La codifica insomma è un sistema di annotazione interna formalizzato. Grazie a questo è in grado di proiettare linearmente la struttura gerarchica del modello di dati. Questo approccio al markup, di gran lunga il più adottato, porta con sé dei vantaggi in termini di leggibilità e di robustezza gestionale dei dati. Tuttavia come è stato osservato è la fonte di gran parte dei limiti tecnici dell'XML per la rappresentazione di strutture complesse.

Nello stand-off markup i marcatori vengono totalmente o parzialmente separati dal file di testo. L'adozione di tecniche di markup esterne è compatibile con l'adozione di XML come linguaggio notazionale, se affiancato da un linguaggio per esprimere relazioni o collegamenti interdocumenali (come XLink/Xpointer). Questo approccio è di fatto identico a quelle basato su elementi di collegamento. La differenza consiste nel fatto che in questo caso tali elementi sono memorizzati in un documento XML esterno. Riprendendo l'esempio precedente, sempre basato sullo schema TEI, se il documento XML contenente il testo è denominato "base.xml" in un file separato avremmo i seguenti elementi che asseriscono la presenza di un legame tra i due elementi <q>:

```
<link xlink:type='extended'>
<anchor xlink:type='locator' xlink:role='quote'
href="doc(base.xml)id(q1)">
<anchor xlink:type='locator' xlink:role='quote'
href="doc(base.xml)id(q2)">
</link>
```

La soluzione per la gestione di gerarchie sovrapposte più "semplice" dal punto di vista sintattico (anche se non necessariamente meno prolissa) è quella basata sugli elementi *milestone*. Un elemento *milestone* (letteralmente elemento pietra miliare) è un elemento XML vuoto che segnala un punto monodimensionale in un documento XML. Questi elementi ovviamente non presentano problemi di sovrapposizione sintattica e si possono collocare liberamente in ogni

punto dell'albero in un documento XML (salvo restrizioni espresse da uno schema). In questo modo è possibile *segnalare* i confini di qualsiasi struttura documentale.

Questa strategia è ampiamente utilizzata nella TEI per veicolare le indicazioni sulla messa in pagina di un testo nel documento fonte da cui è stato memorizzato (si pensi a elementi come <pb>, <cb>, <lb> per indicare rispettivamente il salto di pagina, colonna e riga), come nel seguente esempio (*La notte*, da *I canti orfici* di Dino Campana):

```
<div2 type="poemetto">
<lb n="07.01°" id="NOTTEIB07.01°"/><head
id="NOTTEI">I.
<lb n="07.01b" id="NOTTEIB07.01b"/>LA
NOTTE</head>
<div3 type="sequenza">
<p>
<lb n="07.01" id="NOTTEIB07.01"/>
<milestone unit="notte.1.1"/>Ricordo una vecchia
citt&agrave;, rossa di mura
<lb n="07.02" id="NOTTEIB07.02"/>e turrita, arsa
su la pianura sterminata nel-
<lb n="07.03" id="NOTTEIB07.03"/>l'Agosto
torrido, con il lontano refrigerio di
<lb n="07.04" id="NOTTEIB07.04"/>colline verdi e
molli sullo sfondo. Archi enor-
<lb n="07.05" id="NOTTEIB07.05"/>
<milestone unit="notte.1.5"/>memente vuoti di
ponti sul fiume impaludato
<lb n="07.06" id="NOTTEIB07.06"/>in magre
stagnazioni plumbee: sagome nere
<lb n="07.07" id="NOTTEIB07.07"/>di zingari
mobili e silenziose sulla riva: tra
<lb n="07.08" id="NOTTEIB07.08"/>il barbaglio
lontano di un canneto lontane
<lb n="07.09" id="NOTTEIB07.09"/>forme ignude di
adolescenti e il profilo e la
<lb n="07.10" id="NOTTEIB07.10"/>
<milestone unit="notte.1.10"/>barba giudaica di
un vecchio: e a un tratto
<lb n="07.11" id="NOTTEIB07.11"/>dal mezzo
dell'acqua morta le zingare e un
<lb n="07.12" id="NOTTEIB07.12"/>canto, da la
palude afona una nenia primor-
<lb n="07.13" id="NOTTEIB07.13"/>diale monotona e
irritante: e del tempo fu
```

```
<lb n="07.14" id="NOTTEIB07.14"/>sospeso il
corso.
</p>
</div3>
<pb n="008"/>
```

Gli elementi *milestone* possono essere usati in coppie virtuali per segnalare i confini di segmenti arbitrari di testo che si sovrappongono agli elementi standard, e insieme a elementi di congiunzione o stand-off markup possono rappresentare virtualmente ogni genere di sovrapposizione, auto-sovrapposizione e segmentazione non contigua. Ad esempio la TEI consiglia questa strategia per il trattamento di varianti complesse, come le varianti del passo del Principe proposto poco sopra:

```
lasciato parte della sua gente
<anchor id=p1/>alla
<anchor id=p2/>defesa
<anchor id=p3/>della
<anchor id=p4/>obsidione
<anchor id=p5/>, con le
<-- ... -->
<-- altrove nel file -->
<app from=p1 to=p4>
<lemma wit="G L P W M U">alla defesa della</>
<rdg wit="D">alla offesa delle</>
</app>
<app from=p2 to=p5>
<lemma wit="G L P W M U">defesa della
obsidione</>
<rdg wit="R.Bla">difesa di quella</>
</app>
<app from=p1 to=p3>
<lemma wit="G L P W M U">alla defesa</>
<rdg wit="B">difesa</>
</app>
<app from=p1 to=p5>
<lemma wit="G L P W M U">alla defesa della
obsidione</>
<rdg wit="A E">allobsidione</>
</app>
```

Gli elementi *milestone* offrono la massima flessibilità sintattica senza costringere a separare markup e contenuto. Per contro hanno una scarsa possibilità di essere elaborati in modo automatico mediante software XML standard per finalità appena più complesse della

formattazione. Infatti un *parser* XML può validare la corretta collocazione di un elemento vuoto rispetto a un modello di contenuto, o verificare che due elementi vuoti siano stati correlati mediante coppie di attributi ID/IDREF, ma non può in alcun modo attribuire funzione strutturale alla sequenza di caratteri contenuta tra due elementi vuoti. Tale sequenza, per quanto vada considerata dal punto di vista logico un oggetto testuale non è accessibile come tale a una applicazione XML standard (e dunque anche a processori XSLT/XPATH o interpreti XQuery).

Oltre XML così come lo conosciamo

Le soluzioni al trattamento delle gerarchie sovrapposte viste finora assumono come criterio centrale la compatibilità sintattica con XML sebbene, come si è rilevato, solo parzialmente questa conformità si estenda anche sul piano della validazione strutturale e della possibilità di effettuare elaborazioni complesse mediante software conformi allo standard o a linguaggi basati strettamente sul modello di dati XML (come XSLT, XPath e XQuery). In generale tutte le soluzioni proposte finora hanno grossi limiti per quel che concerne la facilità di gestione e manutenzione. Si tratta insomma di artifici sintattici che non risolvono formalmente il problema bensì, potremmo dire, lo eludono.

Per questo negli ultimi anni la ricerca teorica sullo sviluppo di sistemi di markup non gerarchici ha avuto un notevole stimolo. Allo stato non esiste nessuna proposta che si possa ragionevolmente dire risolutiva. L'ostacolo maggiore consiste nella individuazione di un modello di dati e di un formalismo a esso associato che possa essere validato ed elaborato mediante algoritmi generali e computazionalmente trattabili come avviene per il modello ad albero di XML. Non possiamo in questa sede approfondire con il dovuto dettaglio tutte le possibili soluzioni proposte negli ultimi anni[25]. Ci limiteremo dunque a esaminare le soluzioni che presentano maggiore interesse.

[25] Il più recente intervento sul tema è stata la presentazione di C. Huidtfeldt, C.M. SperbergMcQuenn e Y. Marcoux, del progetto The MLCD Overlap Corpus (MOC) alla conferenza Digital Humanities 2010, <http://dh2010.cch.kcl.ac.uk/academic programme/abstracts/papers/html/ab 633.html>. Il progetto ha l'obiettivo di costituire un corpus di esempi di frammenti testuali con gerarchie sovrapposte codificati in vari linguaggi di markup.

Layered Markup and Annotation Language (LMNL, http://www.-lmnl.net) è un modello di dati, prima ancora che un formalismo di codifica, proposto da Wendell Piez, Jeni Tennison[26] e Paul Caton[27]. Sono disponibili due sintassi per serializzare questo modello: una standard e una proposta recentemente da Steve DeRose e basata interamente su elementi XML vuoti opportunamente caratterizzati mediante attributi, denominata CLIX (*Canonical LMNL in XML*), che ha il vantaggio di produrre documenti XML ben formati.

In LMNL la soluzione del problema delle gerarchie sovrapposte viene trovata uscendo definitivamente fuori dal paradigma gerarchico di XML. In questo modo è possibile rappresentare senza alcuna limitazione sovrapposizioni di qualsiasi tipo e complessità. I concetti fondamentali del *data model* di LMNL sono *range* e *layer*. I *range* sono sequenze arbitrarie di caratteri incluse tra coppie bilanciate di tag che possono sovrapporsi liberamente:

```
 [l}[w}One{w] [w}is{w] [w}be-{l]
[l}fore{w] [w}two{w].{l]
```

I *layer* sono i livelli gerarchici a cui appartengono un insieme di *range* o di *layer* subordinati. Anche essi possono sovrapporsi liberamente.

Allo stato LMNL non dispone ancora di efficaci strumenti di validazione, e tanto meno di elaborazione, e la sua radicale divergenza dal paradigma XML ha limitato molto l'interesse verso questa soluzione, sebbene i suoi ideatori siano ancora al lavoro per portarne avanti lo sviluppo.

Un seconda alternativa potenziale a XML è costituita dalla coppia TexMECS e GODDAG[28] che rappresentano probabilmente il

26 J. Tennison, W. Piez, *The Layered Markup and Annotation Language (LMNL)*, in Proceedings of Extreme Markup Languages 2002; <http://www.idealliance.org/papers/extreme /proceedings/html/2002/Tennison02/EML2002Tennison02.html>.

27 P. Caton, *LMNL Matters?*, in Proceedings of Extreme Markup Languages 2005; <http://www.idealliance.org/papers/extreme/proceedings/html/2005/Caton01/EML2005Caton01.html>

28 C. Huitfeldt, C.M Sperberg McQueen, *GODDAG: A Data Structure for Overlapping Hierarchies*, in "Lecture Notes in Computer Science, vol. 2003/2004", 2004, pp. 139-160. C. Huitfeldt, C.M. Sperberg McQueen, *Representation and processing of Goddag structures: implementation strategies and progress report*, in Proceedings of Extreme Markup Languages 2006, <http://www.idealliance.org/papers/extreme/proceedings/-html/2006/Huitfeldt01/EML2006Huitfeldt01.html>

più avanzato e completo tentativo di trovare una soluzione generale consistente e completa al problema della rappresentazione di strutture complesse nella codifica di documenti testuali, e quello che presenta la maggiore probabilità di avere una applicazione pratica nel prossimo futuro.

Lo sviluppo di questa tecnologia è condotto nell'ambito del progetto di ricerca *Markup Languages for Complex Documents* (MLCD), diretto da Claus Huitfeldt della University of Bergen, e con la collaborazione di Michael Sperberg-McQueen (attualmente al WorldWideWeb Consortium dopo anni di direzione della TEI), e di ricercatori della Graduate School of Library and Information Science (GSLIS) presso la University of Illinois a Urbana-Champaign.

Essa rappresenta l'evoluzione delle ricerche condotte a Bergen per la creazione del *Wittgenstein Archive* (l'edizione digitale delle carte del filosofo viennese) e in particolare della formalizzazione di un linguaggio di markup abbastanza espressivo per effettuare la complessa trascrizione dei manoscritti wittgensteiniani, battezzato MECS. A differenza degli altri formalismi visti finora, MECS è stato infatti utilizzato per la creazione di un grande archivio testuale, sebbene per lungo tempo non avesse alcuna struttura dati formale soggiacente e dunque non disponesse di strumenti per la validazione come avviene per XML.

Il progetto MLCD (http://teksttek.aksis.uib.no/projects/mlcd) ha in primo luogo potenziato la notazione originaria sviluppando TexMECS (*Trivially Extended MECS*), una notazione non molto dissimile da XML (provvede infatti anche strutture come gli attributi) la quale tuttavia permette di esprimere facilmente strutture sovrapposte, auto-sovrapposte e non contigue, come mostra il seguente esempio:

```
<act|<scene|
  <sp who="Åse"|
              <L|Peer, you're lying!|sp>
  <sp who="Peer"|<stage|without stopping.|stage>
        No, I am not!|L>|sp>
  <sp who="Åse"|<L|Well then, swear that
        it is true!|L>|sp>
  <sp who="Peer"|<L|Swear? Why should I?|sp>
  <sp who="Åse"|See, you dare not!|L>
        <L|It's a lie from first to last.|L>|sp>
|scene>|act>
```

Ma ancora più rilevante è stata la definizione di un modello di dati formale e lo studio di grammatiche formali dotate di algoritmi di

parsing trattabili per modello di dati. Il modello di dati è stato battezzato GODDAG (*Generalized Ordered-Descendant Directed Acyclic Graph*). Si tratta di un grafo orientato aciclico dotato delle seguenti caratteristiche:

- gli archi esprimono relazione padre/figlio
- i nodi figli di ogni nodo padre sono ordinati
- i nodi padre possono condividere i medesimi nodi figli, possono cioè avere una discendenza multipla

In linea teorica questo grafo può esprimere tutte le possibili relazioni tra oggetti testuali linearizzati sottoforma di stringhe di caratteri etichettate mediante markup, inclusi i più complessi casi di autosovrapposizione o di frammentazione non contigua e non linearmente ordinata. Esistono tuttavia ancora dei problemi per ricavare il grafo semanticamente corretto a partire da complesse espressioni sintattiche TexMECS che ammettono diverse formalizzazioni logiche.

L'ultimo tassello della proposta MLCD è la ricerca di una grammatica formale equivalente ai grafi GODDAG, in grado di validare univocamente una eventuale notazione seriale in gardo di esprimere tali grafi. Sono state studiate diverse alternative (come testimoniato dalla presentazione di Sperberg-McQueen alla conferenza *Extreme Markup Languages* del novembre 2006), come le *Duck-rabbit grammar*, o sistemi di validazione basati su vincoli espressi in sottoinsiemi completi di logica dei predicati (il cui dettaglio tecnico tuttavia esula dagli scopi della presente trattazione); tuttavia una soluzione solida e computazionalmente accettabile come quella fornita dalla validazione di schemi XML non è ancora stata individuata.

La difficoltà principale della proposta sviluppata dal gruppo MLCD consiste nel voler preservare il paradigma del markup esplicito interno al documento digitale linearizzato. Partendo da questa considerazione recentemente A. Di Iorio, S. Peroni e F. Vitali hanno avanzato una soluzione alla rappresentazione di strutture testuali complesse che, pur partendo dal medesimo modello di dati GODDAG[29], unisce una notazione basata sul cosiddetto *stand-off markup*, e

[29] In realtà l'articolo di Di Iorio et al. estende e generalizza il modello originale, proponendo un e-GODDAG.

tecnologie di *Semantic Web*: EARMARK (*Extreme Annotational RDF Markup*)[30].

Come già accennato sopra, nello stand-off markup il markup viene in parte o in tutto esternalizzato rispetto alla sequenza lineare dei caratteri. Naturalmente il problema in questo caso risiede nel modo in cui si esprime il collegamento tra gli asserti del markup e i brani di testo a cui sono applicati, senza adottare soluzioni o trucchi software *ad hoc* (come nei formati di file dei vari *word-processor* commerciali). In EARMARK questo problema viene affrontato inserendo un livello di astrazione. I costrutti principali sono infatti definiti in una ontologia formale in OWL[31], nel cui ambito le "Locazioni" testuali sono una classe astratta che si divide in sottoclassi che specificano diversi possibili sistemi di referenziazione:

- "CharNumberLocation" defines a location by counting characters. In that case, the string value of the "at" property must be an integer that identifies an unambiguous position in the character stream;
- "XPathLocation" defines a location as a node of an XML docuverse. In this case, the property "at" will be an XPath expression;
- "XPointerLocation" defines a precise point in a docuverse. In that case, the expression "xpointer(point(.42))", for instance, indicates the cursor in-between the 42nd and the 43rd character; with "xpointer-(point(/1/9.3))" we mean the cursor between the 3rd and the 4th character of the ninth node of the root, and so on. [32]

La descrizione delle strutture testuali diventa in questo modo un set di triple RDF sussunte a una ontologia OWL. L'articolo di Di Iorio et al. propone anche alcune possibili (ma incomplete) serializzazioni di parte del grafo RDF, sia in notazione XML standard (basandosi sulle tecniche per rappresentare strutture sovrapposte viste

30 A. Di Iorio, S. Peroni, F. Vitali. *Towards markup support for full GODDAGs and beyond: the EARMARK approach*, Presentato a Balisage: "The Markup Conference 2009". In Proceedings of Balisage: The Markup Conference 2009. Balisage Series on Markup Technologies, vol. 3 (2009); <doi:10.4242/BalisageVol3.Peroni01>.

31 OWL, Ontology Web Language, è lo standard promosso dal W3C per la definizione di ontologie formali nell'ambito Semantic Web. Su questo rimandiamo alla sezione del sito del W3C sul Semantic Web. Su questi temi in italiano segnaliamo E Della Valle., I . Celino, D. Cerizza, *Semantic Web. Modellare e condividere per innovare*, Pearson, Milano 2008.

32 A. Di Iorio et al., *Towards markup*…, cit.

sopra) sia in notazioni come RDFa o RDF XML, inserite all'interno di un documento XML come metadati strutturali interni.

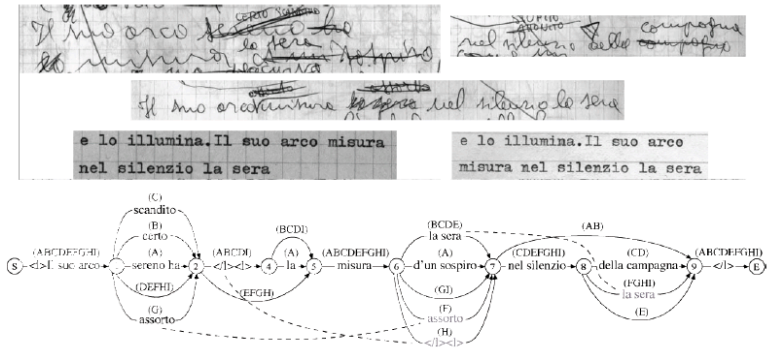
Un approccio concettualmente simile anche se tecnicamente distante è quello sviluppato da D. Schmidt, sulla base dell'esperienza condotta con D. Fiormonte nel progetto *Digital Variants*[33]. Schmidt, come Fiormonte[34], ha una posizione assai critica circa l'adeguatezza in generale di un approccio basato sul markup (e su linguaggi strutturati come XML in particolare) nella rappresentazione digitale di quelli che definisce c*ultural heritage texts*, (testi a stampa o manoscritti oggetto di studio in quanto testimonianze culturali). In particolare i limiti di XML sono evidenti quando si tratti di rappresentare testi soggetti a variazioni, siano esse introdotte dall'autore sui manoscritti (o su molteplici redazioni successive), siano determinate da una tradizione plurima e non uniforme del testo.

La soluzione alternativa che propone è denominata *Multi-Version Document* (MVD). MVD propriamente non è un formato di codifica ma un modello di dati per la rappresentazione di tutte le varianti di un insieme di testi. Si può raffigurare come un grafo diretto aciclico con un nodo iniziale e uno terminale; ogni arco del grafo è etichettato da una stringa opzionale (un dato frammento testuale) e da un insieme di indicatori di versioni in cui tale frammento è attestato. Per ogni possibile versione del testo esiste un cammino nel grafo che consente di ricostruirlo mettendo in serie i contenuti delle etichette stringa degli archi attraversati. L'immagine seguente mostra dei frammenti di testo e la loro rappresentazione in MVD[35].

33 Cfr. D. Schmidt, R. Colomb, *A data structure for representing multi version texts online*, "International Journal of Human Computer Studies", 67/6 (2009); D. Schmidt, *The inadequacy of embedded markup for cultural heritage texts*, "Literary and Linguistic Computing", Advanced Access April 16, 2010; <doi:10.1093/llc/fqq007>; su Digital Variants si veda D. Fiormonte, *Scrittura e filologia nell'era digitale*, Bollati Boringhieri, Torino 2003.

34 Cfr. D. Fiormonte, *Testo*…, cit.

35 L'immagine è tratta da D. Schmidt, R. Colomb, *A data structure*…, cit., e mostra un frammento di una poesia di Valerio Magrelli, Campagna Romana, le cui varianti sono ospitate nel progetto Digital Variants.



MVD può anche essere rappresentato come una lista di coppie formate da un frammento testuale e dall'elenco delle versioni in cui esso è attestato. Come appare evidente il modello di Schmidt è agnostico rispetto al formato in cui il contenuto di ciascuna versione viene codificato: al limite esso può essere anche espresso in XML, nel qual caso i frammenti di testo potranno essere composti da caratteri di testo propri e markup non necessariamente ben formato. In definitiva MVD è una sorta di meta modello di dati pensato per rendere conto di tutte le possibili variazioni testuali e per consentire di effettuare su di esse in modo algoritmico operazioni di ricerca, composizioni, confronto etc. Schmidt tuttavia ha anche sviluppato un formato di file MVD in codifica binaria e una serie di applicazioni software, proponendole come alternativa all'intero insieme di strumenti e linguaggi attualmente utilizzati dalla comunità informatica umanistica. Al di là di una verifica sistematica sulla correttezza e praticabilità del modello teorico in casi di variazione testuale reale e di dimensioni e complessità adeguata, proprio la scelta di proporre come alternativa a un formalismo aperto e standard quello che lo stesso autore definisce "il formato di una applicazione" creato e gestito da un solo sviluppatore solleva non poche perplessità (come testimoniato da una recente accesa discussione avviata da un messaggio dello stesso Schmidt su *Humanist*[36]). Inoltre ci permettiamo di rilevare come la gestione della varianza testuale sia solo un aspetto delle complesse esigenze di rappresentazione digitale dei testi della

36 La nota mailing list di tema umanistico e in particolare informatico umanistico. Per i messaggi a cui ci si riferisce suggeriamo di accedere al sito web che ospita l'archivio della mailing list (http://www.digitalhumanities.org/humanist) e di effettuare una ricerca con la stringa "inadequacies of markup"

tradizione culturale, esigenze tra le quali le questioni della preservazione nel tempo e della accessibilità non sono di secondario momento.

Conclusioni

La creazione di risorse testuali e linguistiche su supporto informatico condotta con finalità scientifiche e culturali ha ormai una storia ultradecennale. Il grande patrimonio di risorse creato rappresenta probabilmente il successo di gran lunga più importante della "disciplina inesistente" che ci ostiniamo chiamare Informatica umanistica. Presupposto e prodotto di questa vasta attività pratica è stata la continua riflessione circa i migliori metodi e strumenti formali per condurre il delicato compito di rappresentare quegli oggetti complessi, plurali e multiformi che sono i testi, soprattutto quelli che rientrano nella difficilmente definibile categoria dei testi letterari.

In questo articolo abbiamo visto come il paradigma dei *markup language* e di XML in particolare presenti indubitabilmente limiti e inadeguatezze. Tuttavia le alternative teoriche e tecniche proposte, allo stato, sono ancora ben distanti dal rappresentare delle alternative praticabili su vasta scala. Se nella valutazione di un formalismo di modellizzazione dei dati, infatti, il primo criterio da prendere in considerazione è ovviamente la completezza e congruenza della rappresentazione con i fenomeni strutturali rappresentati (isomorfismo del modello rispetto al sistema reale), questo criterio di congruità rappresentazionale preso di per sé è insufficiente. È infatti sempre possibile indebolire i vincoli di un formalismo fino a renderlo abbastanza "accogliente" da permettergli di modellizzare qualsiasi fenomeno testuale, ma in questo modo la sua efficacia euristica tende a degradare fino ad annullarsi.

Per questo alla completezza e congruenza va affiancata una serie di ulteriori criteri valutativi di natura teorica, tecnica e pragmatica:

- Leggibilità e facilità di comprensione da parte di un utente umano
- Ampia diffusione e condivisione nella comunità scientifica e oltre
- Facilità di manutenzione e modifica
- Possibilità di validazione e controllo formale dei dati
- Possibilità di presentazione ed elaborazione grafica

- Disponibilità di diverse e indipendenti implementazioni software (possibilmente *open source*)

Tenendo conto di questi criteri XML ha rappresentato e rappresenta ancora il migliore compromesso tra le complesse esigenze di adeguatezza della rappresentazione, quelle altrettanto ostiche della necessità di garantire la preservazione e la portabilità delle risorse e, ultime ma non ultime, quelle più prosaiche ma ineludibili della sostenibilità economica e organizzativa dei progetti di digitalizzazione di grande dimensione.

Naturalmente molto rimane ancora da fare. La ricerca di soluzioni più avanzate per fare meglio ciò che è stato fatto finora è all'ordine del giorno, e come abbiamo visto le possibilità aperte sono numerose e interessanti. A questo compito è chiamata tutta la vasta e variegata comunità delle scienze del testo. Anche coloro e soprattutto coloro che finora hanno avuto una posizione critica nei confronti della svolta digitale, poiché come scrive Jerome McGann:

> One of the great task lying ahead is the critical and editorial reconstitution of our inherited cultural archive in digital forms. We need to learn to do this because we don't as yet know how. Furthermore, we scholars need to learn because it is going to be done, if not by us the by others. We are the natural heirs to this task because it is we who know most about books.[37]

37 J. McGann, *Radiant*..., cit., p. 184.