

LA VISUALIZZAZIONE DI EDIZIONI DIGITALI CON EVT: UNA SOLUZIONE PER EDIZIONI DIPLOMATICHE E CRITICHE

ROBERTO ROSSELLI DEL TURCO, CHIARA DI PIETRO

1. Introduzione

Questo articolo ha come obiettivo l'approfondimento delle tematiche relative alla preparazione e visualizzazione di edizioni scientifiche digitali, con particolare riguardo per le edizioni critiche. Il contesto non è solo quello della metodologia e degli strumenti propri della filologia digitale, ma anche quello delle strategie di sviluppo più efficaci per tali strumenti, con l'obiettivo di mettere lo studioso in grado di preparare e pubblicare una DSE (*Digital Scholarly Edition*, 'edizione scientifica digitale') nella maniera più semplice ed efficace possibile. EVT (Edition Visualization Technology)¹ è un software *open source* appositamente concepito per la creazione di edizioni pubblicabili sul Web a partire da file codificati nel formato XML/TEI P5.² Questo strumento nasce per la visualizzazione del manoscritto noto come Vercelli Book (progetto Digital Vercelli Book,³ d'ora in poi DVB) per mostrare le scansioni del codice affiancate da una trascrizione diplomatica e fornire semplici strumenti di studio come, ad esempio, la ricerca testuale e la visualizzazione di dettagli frutto di un restauro virtuale del manoscritto.

¹ Pagina home: <http://evt.labcd.unipi.it/>. Blog di sviluppo: <http://visualizationtechnology.wordpress.com/>. Codice sorgente: <https://github.com/evt-project/evt-viewer>.

² Consorzio TEI (Text Encoding Initiative): <http://www.tei-c.org/>. Per le *Guidelines* si veda TEI Consortium (a cura di). 2019. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. [3.6.0]. [2019-07-16]. TEI Consortium. URL: <http://www.tei-c.org/Guidelines/P5/>.

³ Si veda in proposito R. Rosselli Del Turco, *The Digital Vercelli Book*, trascrizione e codifica a cura di R. Rosselli Del Turco, R. Cioffi, F. Gorla; software EVT creato da C. Di Pietro, J. Kenny, R. Masotti, R. Rosselli Del Turco; Torino, Collane@unito.it, 2017, URL: <http://www.collane.unito.it/oa/items/show/11>. Al momento in cui scriviamo l'edizione è accessibile direttamente a questo URL: <http://vbd.humnet.unipi.it/beta2/>.

Al momento in cui la codifica dei testi era quasi ultimata (2010 ca.) i programmi disponibili per la pubblicazione erano tendenzialmente troppo complicati da utilizzare, o, al contrario, semplici ma limitati in termini di funzionalità, oppure legati a specifici software e/o sistemi operativi, o ancora proprietari, quindi difficilmente adattabili a specifiche esigenze. La scelta è pertanto ricaduta sulla creazione *ex novo* di uno strumento che non fosse strettamente legato al progetto di origine, ma fosse *open source*, per condividerne lo sviluppo, e il più possibile flessibile e quindi adattabile a diverse tipologie di edizione in vista di possibili nuovi progetti.⁴

Lo sviluppo di un'applicazione di ambito così specifico e ristretto, certo scarsamente utilizzabile al di fuori dell'ambito d'uso accademico, pone problemi che vanno al di là della semplice implementazione tecnica in base alle metodologie della filologia digitale e agli 'standard' di presentazione dei dati di un'edizione digitale:⁵

Rappresentazione del testo. Com'è noto, il linguaggio XML è basato su una 'teoria del testo' di natura prettamente gerarchica, scelta implementativa che costituisce un punto di forza, soprattutto per quanto riguarda la possibilità di elaborare dati salvati in tale formato, ma anche di debolezza per i ben noti problemi che si presentano al momento in cui si desidera individuare nel testo gerarchie parallele.⁶ Oltre a questa caratteristica generale, comune a ogni applicazione basata su XML, lo

⁴ La genesi di questo strumento è già stata descritta in numerosi articoli e non è dunque necessario riproporla in questa sede: per una storia completa di EVT si veda R. Rosselli Del Turco et al., «Edition Visualization Technology: A Simple Tool to Visualize TEI-Based Digital Editions», *Journal of the Text Encoding Initiative*, 8 (2015), URL: <http://jtei.revues.org/1077>; C. Di Pietro, R. Rosselli Del Turco, «Between innovation and conservation: The narrow path of UI design for the DSE», in R. Bleier, M. Bürgermeister, H. Klug, F. Neuber, e G. Schneider (a cura di), *Digital Scholarly Editions as Interfaces. Schriften des Instituts für Dokumentologie und Editorik* 12, Norderstedt, Books on Demand, 2018, pp. 133-163; C. Di Pietro, C. Martignano, R. Rosselli Del Turco, «Progettazione e implementazione di nuove funzionalità per EVT 2: lo stato attuale dello sviluppo», in *Umanistica Digitale*, 7 (2019), pp. 5-21, URL: <https://umanisticadigitale.unibo.it/article/view/9322>.

⁵ Stante la grande eterogeneità delle edizioni digitali pubblicate fino a questo momento, le somiglianze fra edizioni di diversa provenienza sono dovute più alle caratteristiche intrinseche dello sviluppo sul Web e a fenomeni di evoluzione parallela che alla ricerca di un paradigma comune per la visualizzazione delle DSE.

⁶ Esistono più metodi per ovviare a questa limitazione, si veda in proposito il capitolo 20 *Non-hierarchical Structures* delle *Guidelines* TEI (<https://www.tei-c.org/release/doc/tei-p5-doc/en/html/NH.html>).

standard TEI presenta alcune problematiche dovute in parte all'elevato numero di elementi che mette a disposizione, in parte a una certa ridondanza delle soluzioni di codifica.⁷ Si tratta pertanto di uno strumento potente ed efficace, ma non privo di limiti, e di questo deve essere ben conscio il filologo che ne faccia uso.⁸

Elaborazione e riproduzione del testo. La flessibilità ed espressività degli schemi TEI permette di marcare un testo nei dettagli più minuti, ma per conseguire un risultato finale valido sia sul piano tecnico, sia, soprattutto, su quello filologico, è fondamentale una seria pianificazione delle strategie di codifica. Una caratteristica molto interessante, ad esempio, è la possibilità di creare più livelli di edizione (diplomatica, semi-diplomatica, normalizzata) sulla base di un unico documento TEI, selezionando gli elementi più pertinenti per ogni livello. Risulta evidente come questo tipo di metodo editoriale esiga non solo ottime competenze sul piano filologico, ma anche su quello della codifica: la qualità stessa del risultato si basa su questo delicato connubio e sulla capacità del software di visualizzazione di rendere giustizia al lavoro editoriale del filologo.

Presentazione dei dati. Fase finale e particolarmente delicata del lavoro di preparazione di una DSE, anche perché si tratta dell'unica parte effettivamente visibile all'utente finale, è vincolata ai limiti e alle caratteristiche dello strumento di pubblicazione. Questa considerazione riguarda sia il Web come *framework* generale per la condivisione di DSE, sia il software di navigazione, sia, infine, i limiti del dispositivo fisico di visualizzazione, uno schermo bidimensionale dalle dimensioni quanto mai variabili.⁹

La storia dello sviluppo di EVT è soprattutto la storia di come questi problemi sono stati affrontati e risolti in maniera iterativa, facendo tesoro delle lezioni apprese da versione a versione del software.

⁷ Spesso, con l'obiettivo di rendere la marcatura più flessibile e di non far sentire lo studioso costretto entro vincoli troppo rigidi, gli schemi TEI offrono più modi diversi per effettuare la codifica dello stesso fenomeno testuale. Questa libertà d'azione risulta in alcuni casi eccessiva perché costituisce un fattore di potenziale confusione per l'utente, e certamente rappresenta una complicazione per chi deve implementare software basato sul *parsing* di documenti XML/TEI. Si noti che questa stessa considerazione è alla base di iniziative quali TEI Simple (<https://github.com/TEIC/TEI-Simple>).

⁸ Sulle implicazioni riguardo lo sviluppo di uno strumento come EVT cf. *infra* la sezione 6 *Problemi e prospettive*.

⁹ Rimandiamo di nuovo alla sezione 6 per un approfondimento di questo aspetto.

2. Le DSE basate sulla codifica XML/TEI

I metodi per la preparazione di una DSE presentano una tipologia relativamente ampia, che si riflette successivamente in una notevole eterogeneità tipica del prodotto finale. Se il caso più semplice, almeno dal punto di vista tecnico, è quello della raccolta di testi all'interno di una biblioteca digitale,¹⁰ un'altra opzione è quella di edizioni ipertestuali, particolarmente adatta alle edizioni critiche. Le edizioni codificate in HTML e pubblicate direttamente sul Web, infatti, costituiscono potenzialmente un buon prodotto¹¹ e sono sicuramente durevoli nel tempo, ma sono altresì limitate per quanto riguarda l'interscambio, l'annotazione semantica e la possibilità di aggiungere ulteriori livelli di codifica (prosopografie, Linked Open Data etc.).

Per quanto questi e altri metodi abbiano una loro validità, la codifica XML/TEI è il metodo più comune e apprezzato:

- XML è uno standard indipendente da piattaforme hardware e software, molto diffuso e impiegato per gli usi più diversi grazie alla sua flessibilità, e che permette la conservazione dei dati nel medio e lungo periodo;
- gli schemi XML/TEI costituiscono lo standard *de facto* per quanto riguarda la codifica di testi letterari in quanto si tratta di un formato maturo, potente ed espressivo;
- la comunità TEI è ampia e molto disponibile ad aiutare i neofiti;
- attorno agli schemi di codifica TEI è nato un ricco e vivace ecosistema software;
- oltre a garantire la conservazione a lungo termine, la marcatura TEI permette l'interscambio e il riuso dei dati, anche attraverso modalità distribuite sul Web.¹²

Questa breve lista è sufficiente a spiegare perché una percentuale molto alta di DSE siano prodotte grazie alla codifica TEI; si noti tuttavia

¹⁰ Come la biblioteca digitale ALIM (Archivio della Latinità Italiana del Medioevo), URL: <http://alim.unisi.it/>.

¹¹ Si veda ad esempio E. Burgio, A. Ghersetti, M. Buzzoni, *Giovanni Battista Ramusio. Dei viaggi di Messer Marco Polo. Filologie medievali e moderne* 5, a cura di S. Simion e E. Burgio, Università Ca' Foscari Venezia, 2015, DOI: <https://doi.org/10.14277/978-88-6969-00-06>.

¹² Come le specifiche e le API dei DTS (Distributed Text Services): <https://github.com/distributed-text-services/distributed-text-services.github.io>.

che una buona maggioranza di tali edizioni sono trascrizioni diplomatiche accompagnate dalle scansioni del manoscritto di riferimento. In questo modo, infatti, si uniscono i vantaggi di un facsimile digitale a quelli di una trascrizione diplomatica su più livelli di edizione, in altre parole si tratta dello strumento perfetto per la metodologia ecdotica nota come *new philology*.

Le edizioni critiche basate sul formato XML/TEI, viceversa, sono relativamente più rare per vari motivi:

- come accennato sopra, il formato HTML può essere sufficiente se gestito in maniera adeguata;
- le edizioni critiche presentano problemi di scalabilità per quanto riguarda tradizioni testuali complesse;¹³
- la mancanza di strumenti adeguati sia per la preparazione, sia per la visualizzazione di edizioni critiche basate su tale formato rende meno allettante questa modalità di lavoro.

Al di là di quelle che erano le esigenze iniziali del progetto che ha originato EVT, dunque, e per soddisfare richieste simili avanzate da altri studiosi, abbiamo ritenuto fondamentale che EVT acquisisse la funzionalità di visualizzazione di edizione critiche basate sul formato XML/TEI.

3. Il modulo *Critical Apparatus* della TEI

Strumento fondamentale per la creazione di edizioni critiche digitali secondo gli schemi TEI, e alla base della nuova implementazione di EVT, è il modulo *Critical Apparatus*,¹⁴ che raccoglie gli elementi dedicati alla gestione delle informazioni riguardanti l'evoluzione del testo all'interno di una tradizione multi-testimoniale.

Le regole di codifica formalizzate in questo modulo nascono dalla riproposizione di un'impostazione tradizionale della filologia testuale nell'ambito del mondo digitale e permettono di codificare la variabilità del testo all'interno dei testimoni che ne compongono la tradizione, documentando e argomentando i passaggi interpretativi del processo ecdotico, con l'obiettivo di creare un'edizione critica completa in base al metodo stemmatico.

¹³ Si noti che questo non è un problema che riguarda unicamente la TEI.

¹⁴ TEI *Critical Apparatus*: <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/TC.html>.

Gli elementi principali messi a disposizione dal modulo CA sono i seguenti:

- `<listWit>` e `<witness>`: il primo contiene una lista dei testimoni della tradizione testuale di riferimento; ogni testimone viene descritto in un elemento `<witness>` e viene identificato in maniera univoca facendo uso dell'attributo `xml:id`;
- `<app>`: per codificare il segmento di testo per il quale è stata registrata una variazione e raggruppare quindi tutte le lezioni correlate;
- `<lem>`: per registrare le lezioni corrispondenti al testo base;
- `<rdg>`: per codificare ogni diversa lezione attestata in uno o più testimoni della tradizione testuale di riferimento;
- `<rdgGrp>`: per raggruppare più lezioni alternative e classificarle secondo criteri predefiniti.

Gli elementi `<lem>`, `<rdg>` e `<rdgGrp>` possono essere accompagnati da attributi che ne permettono una classificazione mediante una qualche tassonomia di riferimento (*type*), in base alla causa che ha scaturito tale variazione (*cause*), al responsabile di quella scelta editoriale (*resp*) oppure al grado di certezza dell'interpretazione (*cert*). `<lem>` e `<rdg>`, inoltre, possono essere collegati al testimone di riferimento mediante l'attributo *wit* che prende come valore proprio la lista delle sigle – ovvero degli identificativi – dei testimoni in questione.

Vi sono poi tre metodi messi a disposizione per il collegamento dell'apparato al testo critico: il metodo *parallel segmentation* prevede l'inserimento delle voci di apparato all'interno del flusso di testo laddove si verificano letture diverse nella tradizione testuale (*in-line markup*); il *location-referenced method* e il *double-end-point-attached method*, invece, permettono di registrare le variazioni anche all'esterno del corpo del testo, secondo il principio dello *stand-off markup*, in una sezione diversa del documento TEI o anche in un documento TEI separato. Ciascuno di questi tre metodi, descritti in dettaglio nel già citato capitolo delle *Guidelines TEI*,¹⁵ presenta vantaggi e svantaggi:

- il metodo *location-referenced* permette di collegare voci di apparato e note critiche a un testo ricostruito in maniera semplice ma poco pre-

¹⁵ Rimandiamo in particolare alla sezione <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/TC.html#TCAPLK> per un approfondimento delle peculiarità tipiche di ciascun metodo, in questa sede presentate necessariamente in forma sintetica.

cisa, grazie a riferimenti a segmenti generali del testo (paragrafi, versi, righe etc.): non permette quindi una successiva elaborazione dei dati codificati e pertanto sembra utile soltanto nel caso si voglia digitalizzare rapidamente un'edizione critica a stampa;

- il metodo *double-end-point-attached* permette di evitare ogni problema di sovrapposizione delle varianti (ingl. *variant overlap*), ma in questo caso la definizione dei segmenti di testo interessati da varianti è effettuata in maniera rigorosa grazie all'uso di elementi vuoti (<anchor/>), e questo permette una successiva elaborazione dei dati codificati, ad esempio per generare automaticamente il contenuto testuale di tutti i testimoni; l'inserimento di elementi vuoti e di riferimenti a questi ultimi negli elementi <app>, tuttavia, è un procedimento più complesso e quindi più suscettibile di provocare errori da parte del filologo, tanto che sin dal primo inserimento nelle *Guidelines* TEI si prevede che tale metodo sia usato con l'ausilio di un qualche software sviluppato a tal scopo;

- il metodo *parallel segmentation* è l'unico a prevedere la marcatura delle voci di apparato esclusivamente all'interno del testo, includendo le lezioni accettate dal filologo in <lem> e le varianti nei <rdg>,¹⁶ permettendo anche in questo caso successive elaborazioni del dato critico (come la già citata generazione automatica di tutti i testimoni); lo svantaggio principale di questo metodo è che non consente di gestire varianti sovrapposte (ma permette varianti annidate).

In base a questo breve sommario non stupirà apprendere che il metodo più usato sia il terzo, il *parallel segmentation*, sia perché piuttosto semplice e intuitivo, in quanto prevede l'inserimento della marcatura nel corpo del testo (codifica *inline* invece che *stand-off*), sia pure perché il suo principale svantaggio può comunque essere gestito includendo all'interno di <lem> e <reg>¹⁷ un unico segmento di testo per il quale esistono varianti sovrapposte.

¹⁶ Negli altri due metodi il testo critico è già definito come 'testo principale', esattamente come in una edizione a stampa.

¹⁷ Questa soluzione presenta lo svantaggio di aggiungere ridondanza alle voci di apparato, e anche di rendere meno comprensibili ed evidenti i fenomeni di variazione del testo, pertanto non è raccomandabile se non nel caso in cui questa tipologia di variazione sia nettamente minoritaria nella tradizione testuale di riferimento.

FIGURA 2

Il primo *capitulum* dell'Editto di Rotari con codifica delle varianti (metodo *parallel segmentation*)

```
<p n="1"><w>1</w>. <w>Si</w> <w>quis</w> <app>
  <lem><w>hominum</w></lem>
  <rdg wit="#B5"><w>homo</w></rdg>
  <rdg wit="#B3"/>
</app> <w>contra</w> <w>animam</w> <w>regis</w> <app>
  <lem><w>cogitaverit</w></lem>
  <rdg wit="#B5"><w>considerauerit</w></rdg>
</app> <app>
  <lem><w>aut</w> <app>
    <lem><w>consiliaverit</w></lem>
    <rdg wit="#B9"><w>consiliaberit</w></rdg>
    <rdg wit="#B11 #B12"><w>consiliatus</w> <w>fuerit</w></rdg>
  </app></lem>
  <rdg wit="#B2 #B5"/>
</app>, <w>animae</w> <w>suae</w> <w>incurrat</w> <w>periculum</w> <app>
  <lem><w>et</w></lem>
  <rdg wit="#B6"><w>aut</w></rdg>
</app> <w>res</w> <w>eius</w> <w>infiscentur</w>.</p>
```

Il modulo CA presuppone che la collazione dei testimoni sia già stata effettuata e che il filologo abbia a sua disposizione tutte le lezioni fra le quali scegliere quelle che ritiene più appropriate per la definizione di un testo critico. La codifica nel formato TEI, dunque, riguarda la sola fase della *restitutio textus* e ha come obiettivo la trasposizione in un linguaggio formale delle informazioni che costituiscono l'apparato critico.

Se consideriamo un'edizione critica come un'ipotesi di ricerca,¹⁸ è fondamentale che l'apparato critico sia in grado di rendere conto delle scelte editoriali, in modo che il lettore possa verificarle e, nel caso non ne fosse soddisfatto, abbia i mezzi per avanzare ipotesi alternative. In una DSE questo compito può essere svolto in maniera più efficace rispetto a un'edizione tradizionale perché essa offre un ambiente di ricerca molto più ricco di opzioni rispetto alla seconda. Una volta sottratti alla tirannia dello spazio disponibile nelle edizioni a stampa, infatti, è possibile presentare al lettore non solo molte più informazioni altrimenti escluse dalla pubblicazione (apparato positivo, varianti ortografiche, tipologia

¹⁸ «...un'edizione critica è, come ogni atto scientifico, una mera ipotesi di lavoro, la più soddisfacente (ossia economica) che colleghi in sistema i dati.», G. Contini, *Esercizi di lettura sopra autori contemporanei: con un'appendice sui testi non contemporanei. Nuova edizione aumentata di 'Un anno di letteratura'*, Torino, Einaudi, 1974, p. 369.

delle varianti etc.), ma anche gestire i dati dell'edizione in maniera dinamica. Ad esempio, l'apparato può essere presentato in forma diversa a seconda del contesto: più esteso e ricco di informazioni quando si visualizza il solo testo critico, più sintetico ma ugualmente valido al momento in cui si confronta il testo ricostruito con i singoli testimoni.¹⁹ La navigazione stessa dell'edizione, oltre a permettere una flessibilità di gran lunga maggiore rispetto a un testo a stampa, non è confinata all'oggetto libro, perché può estendersi e collegare il testo ad altre risorse sul Web.

Considerato il ruolo centrale dell'apparato al fine di raggiungere tutti questi obiettivi, la distribuzione di caratteristiche utili in ben tre metodi diversi per il collegamento dell'apparato al testo critico ha avuto come conseguenza problemi di cui solo adesso si inizia a intravedere soluzione.

Parte di tali problemi è dovuta al fatto che si tratta di uno dei moduli più anziani della TEI, praticamente rimasto immutato anche riguardo alle innovazioni introdotte nelle versioni più recenti degli schemi e Guidelines TEI, in particolare la P5 (pubblicata nel 2007 e successivamente aggiornata almeno una volta l'anno). Creato negli anni '90 da studiosi di grande valore – fra tutti basti ricordare Peter Robinson – questo modulo è stato quindi usato nella sua forma originale finché nell'annuale conferenza TEI del 2009 venne avanzata l'idea di creare una *task force* per migliorarlo o riscriverlo *ex novo*. Era ormai evidente, infatti, che i limiti presenti nella forma originale costituivano un disincentivo all'utilizzo, contribuendo così al fenomeno per cui le edizioni critiche digitali, nella fattispecie edizioni basate sullo standard TEI, rappresentano solo una piccola percentuale del totale delle DSE. Sulla scorta di queste considerazioni è stato creato un gruppo di lavoro all'interno del SIG (Special Interest Group) TEI dedicato ai manoscritti²⁰ con il compito di avanzare proposte per superare tali limiti. Purtroppo, dopo una fase di slancio iniziale, e in parte anche a causa di divergenti punti di vista sulle strategie da seguire, si è deciso di rinunciare all'obiettivo più ambizioso, ovvero una riscrittura totale del modulo. Parte delle proposte formulate dal gruppo di lavoro sono state dunque convertite in richieste di cambiamento relative a punti specifici, e i responsabili dello sviluppo degli schemi TEI hanno dato seguito a tali richieste implementando una serie di miglioramenti incrementali.²¹

¹⁹ Si veda la sezione 5: *Le caratteristiche della prossima release*.

²⁰ Pagina home del SIG: <https://tei-c.org/Activities/SIG/Manuscript/>.

²¹ Parte del lavoro svolto dal SIG è visibile sulla pagina dedicata all'interno del Wiki TEI: https://wiki.tei-c.org/index.php/Critical_Apparatus_Workgroup.

La situazione attuale per quanto riguarda il modulo CA della TEI può essere riassunta come segue: da un lato, gli strumenti di codifica necessari per creare la lista dei testimoni e le voci di apparato, e mettere in collegamento le une e gli altri, sono implementati in maniera soddisfacente e costituiscono una risorsa efficace per lo scopo prefisso; dall'altro, malgrado i miglioramenti introdotti di recente, la necessità di scegliere uno fra tre diversi metodi di collegamento dell'apparato al testo critico costituisce un problema non indifferente, perché come abbiamo visto il metodo più semplice e intuitivo (*parallel segmentation*) non è quello più potente, mentre quello più potente (*double-end-point-attached*) non è né semplice, né intuitivo. Stante la situazione attuale, un progresso significativo potrà verificarsi solo nel momento in cui si avvererà la 'profezia' della TEI e il filologo potrà avvalersi di uno strumento che gli consenta di preparare un'edizione critica nel formato XML/TEI usando un ambiente grafico (*authoring system*) piuttosto che la codifica diretta del testo. Una soluzione di questo tipo sarebbe preferibile alla tendenza attuale, che vede una netta preferenza per il metodo *parallel segmentation*, a patto che tutto il processo sia trasparente e la codifica TEI sia sempre disponibile per eventuali controlli o modifiche da effettuare a mano.²² Per raggiungere questo obiettivo non è indispensabile che lo strumento di *authoring* lavori direttamente sui documenti TEI: potrebbe, al contrario, lavorare su un formato interno²³ per la gestione dei dati inseriti dall'utente e importare / esportare in formato TEI il prodotto finale.²⁴

Un problema più generale, che non dipende dal metodo prescelto o dagli attuali strumenti di codifica disponibili, è la scalabilità di una edizione critica prodotta con marcatura nel formato TEI. Molto dipende dalla ricchezza e complessità della traduzione testuale relativa all'opera oggetto di edizione, ma se la *constitutio textus* sulla base di una dozzina di testimoni non dovrebbe, in circostanze normali, presentare preoccupazioni particolari per il filologo, non possiamo fare a meno di interrogarci su cosa succede quando i testimoni sono cento o più. È probabile che anche in questo frangente possa risultare prezioso uno strumento

²² L'uso di altri formati, proprietari o meno, al momento sembra decisamente sconsigliabile dato il ruolo giocato dallo standard TEI nelle buone pratiche di filologia digitale.

²³ Come il formato JSON, cf. *infra*.

²⁴ Uno strumento promettente, ma il cui sviluppo purtroppo è momentaneamente interrotto, è il software CEED (Critical Edition EDitor) presentato in occasione della conferenza AIUCD 2018 (abstract disponibile nel *Book of Abstracts* della conferenza: <http://amsacta.unibo.it/5997/>).

di *authoring*, sia ad esempio per la gestione di trascrizioni diplomatiche separate e/o il risultato di operazioni di collazione (semi)automatica, sia pure per la possibilità di organizzare il lavoro su base collaborativa e distribuita, con l'eventuale controllo e sincronizzazione (RCS) degli interventi editoriali effettuati da più studiosi contemporaneamente.

Ultima, ma non per importanza, la questione relativa alla visualizzazione di edizioni critiche digitali: per quanto riguarda il metodo *double-end-point-attached*, non solo manca uno strumento grafico per preparare una DSE usando tale modalità di codifica, ma anche per visualizzarla al termine del lavoro, fatto ancora più increscioso. Come si potrà arguire da quanto segue, riteniamo che EVT possa dare un contributo sostanziale per raggiungere questo obiettivo, in modo da rendere più semplice e rapida almeno la fase finale della creazione di una edizione critica digitale, ovvero la sua pubblicazione sul Web.

4. EVT per le edizioni critiche

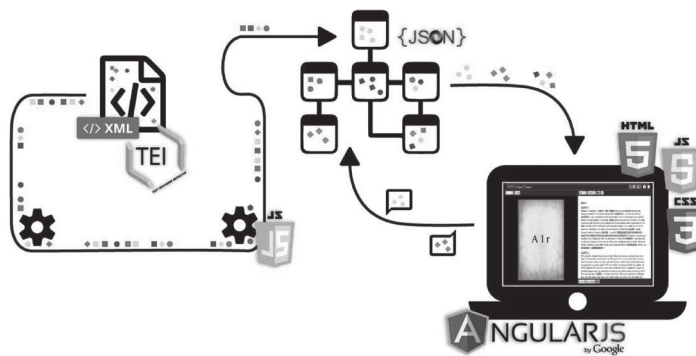
Come accennato nell'*Introduzione*, fin dalla sua origine EVT è stato progettato e sviluppato come strumento generico per la pubblicazione di edizioni digitali pur avendo come primo obiettivo la pubblicazione di una specifica tipologia di edizione, il facsimile digitale accompagnato da trascrizione diplomatico-interpretativa e da alcuni strumenti utili in tale contesto (ad esempio il collegamento testo-immagine). Nel corso degli anni, la collaborazione con progetti esterni è stata il principale motore di evoluzione: ogni nuova funzionalità aggiunta per uno specifico progetto di edizione è stata integrata ed è divenuta utilizzabile da tutti gli altri progetti pubblicati con EVT.²⁵

La versione 2 di EVT nasce appositamente per permettere la visualizzazione di edizioni critiche codificate con XML/TEI, obiettivo fondamentale per il Digital Vercelli Book, ma anche richiesta pervenuta da numerosi altri studiosi. Questa nuova versione ha comportato non solo una profonda revisione dell'interfaccia utente, per la quale – a differenza di quanto avviene nelle edizioni tradizionali per le quali esiste un *layout* familiare e condiviso – non esisteva (e ancora non esiste) uno standard

²⁵ Tra le collaborazioni più importanti realizzate grazie a questa prima versione troviamo quella con il Codice Pelavicino digitale (<http://pelavicino.labcd.unipi.it/>), con il progetto Tarsian / HumaRec (<https://humarec.org/>) e con il progetto PhiBor (<http://www.avicennaproject.eu/>).

riconosciuto,²⁶ ma anche una altrettanto complessa riprogettazione e riscrittura di tutto il software. Pur rimanendo il testo XML il punto di partenza dal quale vengono ricavati i dati per generare l'edizione digitale, abbiamo infatti rinunciato al *builder* di EVT 1 basato su una catena di trasformazioni XSLT 2 che si occupavano di riorganizzare i dati di partenza in appositi file HTML da dare poi in *input* all'applicazione Web per la visualizzazione. In EVT 2 invece, pur rimanendo legati a tecnologie standard dello sviluppo Web quali HTML5 e CSS3, il *viewer* è stato sviluppato utilizzando il *framework* JavaScript AngularJS,²⁷ progettato appositamente per la realizzazione di applicazioni Web a pagina singola (Single Page Application o SPA): le informazioni vengono recuperate (*parsing*) tramite un'unica chiamata AJAX direttamente dal file XML, senza alcun passaggio intermedio; sono poi archiviate in strutture dati e collezioni virtuali, organizzate in maniera tale da permetterne un rapido accesso in caso di necessità.²⁸ Si è infine scelto di non mantenere in memoria i dati direttamente nel formato originale XML, bensì di utilizzare apposite strutture JSON (JavaScript Object Notation), più facilmente accessibili e gestibili da un linguaggio come JavaScript.

FIGURA 3
Schema illustrativo della nuova architettura di EVT 2.



²⁶ Per maggiori dettagli sulla progettazione dell'interfaccia utente di EVT 2 si veda Di Pietro, Rosselli Del Turco, «Between innovation and conservation».

²⁷ AngularJS – Superheroic JavaScript MVW Framework: <https://angularjs.org/>.

²⁸ La migrazione dell'architettura originale è descritta in maniera particolareggiata in Di Pietro, Rosselli Del Turco, «Between innovation and conservation», e Di Pietro, Martignano, Rosselli Del Turco, «Progettazione e implementazione di nuove funzionalità per EVT 2».

Più difficile e tortuoso è stato il percorso per arrivare a definire l'interfaccia utente finale. È infatti importante ricordare che, per quanto complesso e raffinato sia un software a livello di funzionalità offerte, se non si presenta come intuitivo, facile da usare e accessibile, la sua diffusione e utilità risulterà limitata. Ciò è ancora più importante nel campo delle edizioni digitali, dove non esistono ancora degli standard ufficialmente riconosciuti. Pertanto la definizione dell'interfaccia utente presentata con le versioni *alpha*²⁹ e *beta*³⁰ di EVT2 è stata preceduta da un'attenta analisi dello stato dell'arte e delle problematiche che alcune delle edizioni digitali pubblicate ad oggi presentano, da un approfondito studio degli elementi caratteristici di un'edizione critica tradizionale e dall'identificazione delle nuove funzionalità ritenute utili strumenti aggiuntivi da integrare nell'edizione finale.

Il risultato finale è uno strumento che permette una veloce pubblicazione di un'edizione digitale codificata in XML/TEI. L'utente non dovrà far altro che impostare le opzioni di configurazione appropriate (a cominciare dal nome del documento XML dell'edizione, che può essere sia un file locale sia una risorsa on-line) e aprire il file `index.html` su un navigatore.³¹ L'applicazione AngularJS si occuperà quindi di leggere il file di configurazione, aggiornare sulla base di tali impostazioni gli appositi parametri nel modello interno, avviare una richiesta AJAX e recuperare quindi il contenuto del file indicato dall'editore. Se non si verificano errori durante la lettura del file, il sistema procede affidando i dati letti alle apposite funzioni di *parsing* che si occupano sia di salvare una copia del contenuto originale all'interno del modello, sia di recuperare e archiviare tutte le informazioni necessarie alla corretta visualizzazione richiesta (ad esempio, l'elenco delle pagine, delle *named entities* o dei testimoni della tradizione). Come indicato precedentemente, ogni *parser*, una volta recuperati i dati necessari, li riorganizza in opportuni oggetti JSON e li salva all'interno del modello dei dati che viene poi utilizzato dall'applicazione per l'inizializzazione e la gestione delle varie viste.

²⁹ C. Di Pietro, *EVT per le edizioni critiche digitali: progettazione e sviluppo di una nuova GUI basata sullo schema progettuale MVC*, Pisa, ETD - Electronic theses and dissertations repository, Università di Pisa, 2016, URL: <https://etd.adm.unipi.it/theses/available/etd-03292016-203644/>.

³⁰ C. Martignano, *Progettazione e sviluppo di un apparato critico modellato sulla tradizione a stampa in EVT*, Pisa, ETD - Electronic theses and dissertations repository, Università di Pisa, 2017, URL: <https://etd.adm.unipi.it/theses/available/etd-06132017-202934/>.

³¹ Come per EVT 1.0, se si utilizza il software localmente senza alcun server d'appoggio, l'accesso ai file mediante AJAX sarà possibile soltanto dopo aver disabilitato l'opzione che blocca l'accesso ai file locali o per mezzo di un server eseguito in locale.

Oltre alle funzioni per l'estrazione di dati specifici (come l'elenco dei testimoni collazionati, o le voci d'apparato codificate), è stata implementata anche una funzione generica che trasforma i nodi XML/TEI in elementi HTML aventi come classe principale il nome del nodo stesso per permettere una più facile personalizzazione di stile di elementi semplici; inoltre, ogni attributo del nodo di partenza viene rimappato con un apposito *data-attribute*³² che permette quindi di affinare eventuali regole di stile per uno specifico nodo.

Non appena le informazioni di base sono state recuperate, inizia il *rendering* su schermo dell'edizione: essa viene costruita in maniera semi-automatica, tenendo conto sia della configurazione impostata dall'editore nell'apposito file, sia dei dati effettivamente recuperati dal file sorgente.

Di seguito verranno elencate e descritte le diverse funzionalità presenti nell'ultima versione di EVT, legate in particolare alle edizioni critiche.

Per quanto riguarda l'interfaccia, l'utente si trova davanti uno strumento *user-friendly*, pulito ed elegante che cerca di ottimizzare lo spazio disponibile grazie a una precisa organizzazione in pannelli distinti. Nella palette di colori usata per gli elementi strutturali dominano i toni freddi, tra il blu e il grigio, in modo da mettere più in risalto gli elementi interattivi con colori caldi e vivaci. Proprio ai colori è stato affidato il compito informativo della variabilità testuale: sia nel caso di apparato *inline*, sia in quello di apparato presentato in un pannello separato (si veda sotto), infatti, l'utente è in grado di riconoscere a colpo d'occhio la presenza di elementi d'interesse grazie al colore di sfondo diverso dal bianco. Talvolta le varianti sono rappresentate con colori differenti, ad indicare fenomeni filologici diversi, come ad esempio il cambio di mano scribale, la causa che ha generato la variante, la tipologia di variazione testuale etc.; un'apposita legenda esplicita il significato associato ad ogni colore.

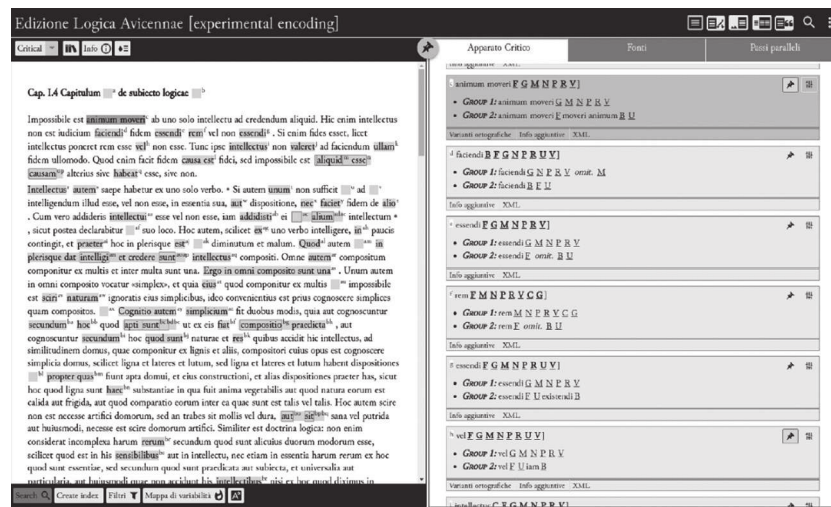
L'utente ha, in ogni momento, la possibilità di accedere all'apparato critico legato a una determinata lezione. Tale apparato, come anticipato, si può presentare in due modalità differenti, in base alle scelte di configurazione dell'editore, ma anche alle esigenze di spazio dell'interfaccia utente. L'apparato *inline* non è altro che un contenitore, accodato alla lezione di riferimento, che suddivide il testo in due parti e si distingue da quest'ultimo grazie a uno sfondo differente. Il pannello separato, invece, è un elemento più simile a quanto presente nelle edizioni a stampa: si tratta di una sezione affiancata al testo che raggruppa sia tutte le voci di apparato critico, sia l'apparato delle fonti, i passi paralleli e le note di commento;

³² HTML Global data-* Attributes: http://www.w3schools.com/tags/att_global_data.asp.

ogni elemento delle diverse fasce d'apparato rimane sempre strettamente collegato al testo, tanto che basta un clic su una singola voce per riallineare il testo critico al punto corrispondente nel testimone. In entrambi i casi, l'apparato è organizzato in due sezioni principali: la prima presenta in modo sintetico le informazioni principali, tradizionalmente inserite all'interno di una voce critica, ovvero il lemma, seguito dall'elenco delle sigle dei testimoni che lo attestano, e l'elenco delle varianti, ciascuna corredata dell'elenco dei testimoni in cui sono attestate; la seconda sezione, invece, ospita dei contenuti di approfondimento, a loro volta organizzati in sezioni secondarie, quali note critiche codificate dall'editore all'interno della voce d'apparato, le varianti ortografiche normalmente ritenute meno significative e quindi escluse dall'apparato critico, una lista dei metadati associati ai singoli contenuti della voce d'apparato e, infine, la codifica XML corrispondente alla voce di apparato selezionata.

FIGURA 5

Il layout generale di EVT nella vista riservata al solo testo critico.



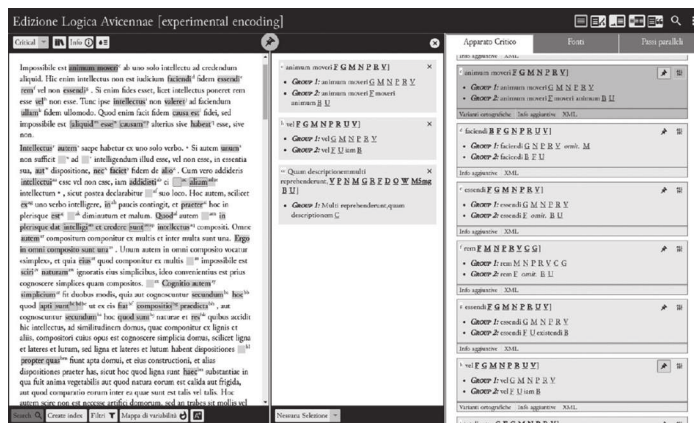
All'interno del contenitore delle voci d'apparato, in base alle opzioni di configurazione fornite dall'editore, è possibile inoltre presentare le varianti suddivise in due o più gruppi, sulla base dei testimoni in cui esse sono state attestate e della tipologia predefinita.

Per arricchire l'interazione dell'utente con le varianti si è poi scelto di implementare tre strumenti specifici:

- la *heatmap*: permette di avere a colpo d'occhio un quadro generale della variabilità testuale, più una lezione è soggetta a variabilità all'interno della tradizione codificata, più intenso sarà il suo colore di sfondo.
- il filtro di varianti: questo selettore rende selezionabili le tipologie di varianti da mantenere evidenziate, in modo da permettere uno studio più mirato di determinati fenomeni all'interno del testo.
- lo strumento *pin frame*: permette di selezionare e salvare³³ alcune tipologie di elementi (in particolare voci d'apparato e *named entities*) in modo da potervi accedere in maniera molto rapida.

FIGURA 7

La vista del testo critico con le fasce di apparato e il *pin frame* attivo.



A partire dalla versione beta di EVT 2, è possibile affiancare e arricchire l'apparato critico con l'apparato delle fonti e dei passi paralleli: il primo raccoglie in forma sintetica i riferimenti bibliografici che un certo passo del testo cita esplicitamente o implicitamente; il secondo, invece, si presenta sotto forma di elenco di riferimenti a passi di altri testi, vicini semanticamente o linguisticamente ad alcune porzioni del testo critico, risultando utili per lo studioso come strumento per un'analisi comparata dei testi. Per quanto destinati a scopi differenti, da un punto di vista

³³ Dal momento che EVT si presenta come strumento che lavora in modalità *client only*, e quindi non si ha a disposizione un server d'appoggio sul quale registrare i dati, non viene effettuato un vero e proprio salvataggio. Per simulare tale salvataggio viene sfruttato l'oggetto *localStorage* del navigatore che permette di registrare informazioni in maniera da renderle disponibili anche dopo un riavvio del computer.

prettamente strutturale essi trovano diverse analogie anche con le voci di apparato, in particolare il fatto di far riferimento ad un segmento del testo critico e di presentare contenuti facilmente suddivisibili in sezioni differenti. Per questo motivo si è scelto di utilizzare lo stesso modello grafico utilizzato per presentare le voci di apparato, con l'obiettivo di creare omogeneità all'interno dell'interfaccia utente. Il contenitore di una singola voce d'apparato è dunque composto da

- un'intestazione, che presenta la citazione o il passo parallelo e consente di navigare all'interno di una fascia di apparato per cercare la lezione desiderata, senza dover consultare il testo critico;
- un elenco delle opere citate o dei testi paralleli sotto forma di riferimenti bibliografici sintetici;
- una sezione dedicata alle informazioni aggiuntive riguardanti la fonte selezionata, opportunamente organizzate in schede; ad es. il passo della fonte che corrisponde al passo del testo critico; il collegamento a una risorsa esterna o il collegamento interno alla vista testo-fonte; il riferimento bibliografico completo della fonte; la codifica XML sia del passo del testo critico, sia del riferimento bibliografico della fonte.

Altre funzionalità importanti riguardano le diverse viste messe a disposizione e quindi i contenuti che l'utente può visualizzare:

Vista di collazione: particolare modalità di visualizzazione dell'applicazione in cui uno o più testimoni vengono affiancati l'uno all'altro e messi a confronto con il testo critico; è l'utente che sceglie quanti e quali testimoni affiancare al testo critico. Non si tratta di una collazione automatica di più testi, bensì di un'estrazione delle informazioni circa le varianti attestate nei diversi testimoni e opportunamente codificate nel file XML di partenza, procedimento che porta alla ricostruzione automatica del testo degli stessi. A livello prettamente grafico, per riuscire a gestire un numero ipoteticamente infinito di testimoni da confrontare mantenendo l'interfaccia ben strutturata, si è scelto di utilizzare una soluzione di compromesso che solo in parte sfrutta i principi del cosiddetto *responsive design*. Infatti, se fino a due testimoni aggiunti si ha un ridimensionamento automatico dei pannelli visibili, che vanno ad occupare ognuno circa un terzo dello spazio disponibile, dal terzo testimone in poi il ridimensionamento non è più proporzionale allo spazio disponibile, ma è guidato da una larghezza minima, e i nuovi pannelli vengono semplicemente affiancati ai precedenti senza essere ridimen-

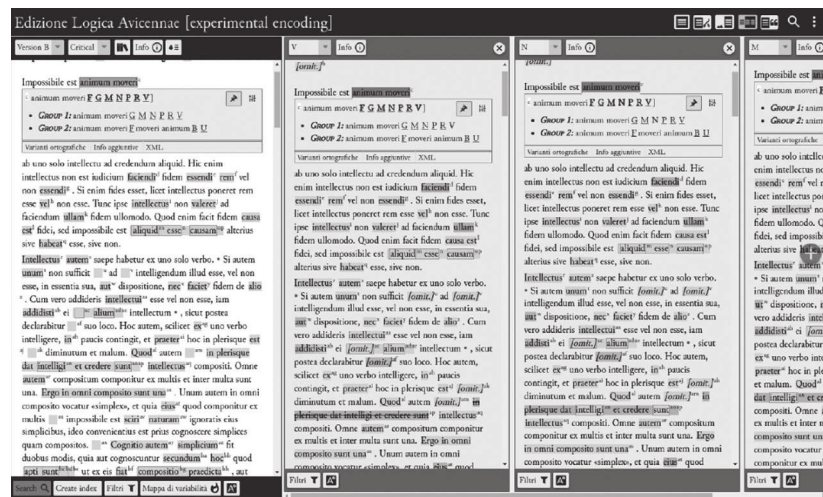
sionati. In questo modo, l'interfaccia rimane stabile, il testo dei singoli testimoni è facilmente leggibile e l'accesso a quelli non immediatamente visibili è reso possibile da una barra di scorrimento orizzontale.

Vista di recensioni multiple: altra modalità di visualizzazione che similmente alla precedente affianca al testo critico principale le altre diverse redazioni che l'editore ha scelto di presentare integralmente al lettore.

Vista per l'apparato delle fonti: il testo critico viene messo a confronto con i testi delle sue fonti, permettendo uno studio dell'opera da nuovi punti di vista. Per esempio, analizzando il testo da una prospettiva genetica, è possibile sfruttare questa vista per osservare meglio quanto certe fonti abbiano influito sulla stesura finale dell'autore, oppure, nel caso di testo argomentativo o scientifico, è possibile studiare il dialogo dell'autore con altri scrittori autorevoli del suo tempo.

Segnalibro (bookmark): ultimo, ma non per importanza, lo strumento permette di ottenere un riferimento diretto ad una determinata vista o sezione del testo, tenendo traccia di diversi parametri quali la modalità di visualizzazione, il documento, la pagina, il livello di edizione, i testimoni visualizzati e l'eventuale voce d'apparato selezionata. Tale segnalibro virtuale è stato pensato non solo per permettere all'utente di riprendere lo studio del testo dal punto in cui era stato interrotto, ma anche per facilitare la citabilità e lo scambio di informazioni fra studiosi.

FIGURA 8
La vista collazione.



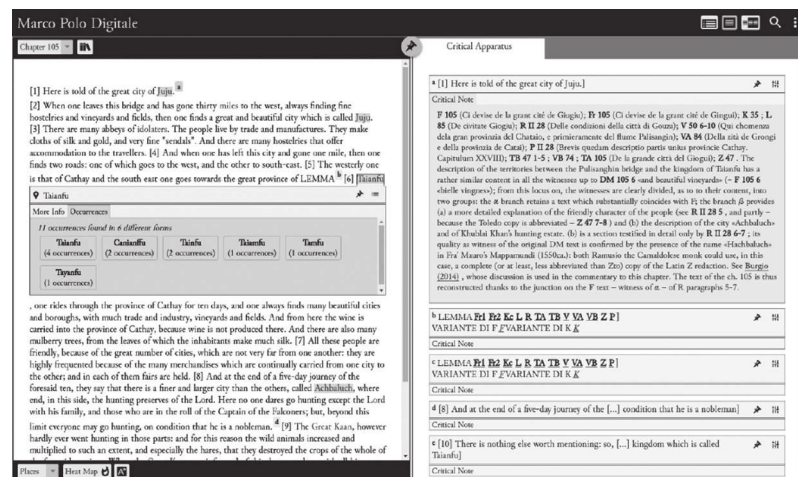
5. Le caratteristiche della prossima versione

Nella versione prevista per il 2020 saranno introdotti miglioramenti relativi alle funzionalità già esistenti, in particolare al supporto per le edizioni critiche, ma soprattutto saranno aggiunte nuove funzionalità, parte delle quali necessarie per raggiungere la *feature parity* con la versione precedente (EVT 1).

Per quanto riguarda il supporto alle edizioni critiche, oltre alla correzione di *bug* e miglioramenti a livello di interfaccia utente, sono previste due novità molto importanti: il supporto per il metodo *double-end-point-attached* e il supporto per parte o tutti i testimoni come trascrizioni diplomatiche separate (quindi non generati automaticamente sulla base del documento TEI). A queste caratteristiche si aggiungerà anche la gestione di versioni diverse di un testo, in modo da poter avere traduzioni e rifacimenti come integrazione e confronto del testo critico. Si tratta di funzionalità complesse che saranno introdotte nella seconda beta a titolo sperimentale, per ricevere quanto più *feedback* possibile dagli utenti di EVT, e che potrebbero cambiare, anche sensibilmente, nella versione finale.

FIGURA 9

Il supporto per il metodo *double-end-point-attached* e per le *named entities*.³⁴



³⁴ Testo e codifica tratti da una versione preliminare e sperimentale del progetto *Marco Polo Digitale*, a cura di M. Buzzoni e E. Burgio.

Le nuove caratteristiche del tutto assenti nella prima beta, invece, sono *in primis* l'introduzione di un supporto completo per le edizioni diplomatiche, al momento presente solo in forma embrionale, e soprattutto l'introduzione di un nuovo visualizzatore di immagini particolarmente potente e flessibile,³⁵ e di un nuovo motore di ricerca.³⁶ Una funzionalità del tutto nuova è il supporto per VisColl,³⁷ un software per la visualizzazione della struttura fascicolare di un manoscritto.

FIGURA 11

La struttura fascicolare del Vercelli Book usando VisColl.



Questa *release* rappresenta dunque un momento cruciale nello sviluppo di EVT, e pone anche problemi di tipo tecnico per quanto riguarda la ricomposizione nella base di codice generale di molte funzionalità significative sviluppate separatamente.³⁸ Si tratta di un momento deli-

³⁵ OpenSeaDragon (<http://openseadragon.github.io/>), particolarmente adatto alla gestione di immagini di grandi dimensioni e compatibile con il protocollo IIIF (<https://iiif.io/>) per la condivisione di immagini sul Web.

³⁶ G. Cacioli, *Filologia e Information Retrieval: progettazione e sviluppo di un motore di ricerca per EVT*, Pisa, ETD - Electronic theses and dissertations repository, Università di Pisa, 2019, URL: <http://etd.adm.unipi.it/theses/available/etd-01152019-191842/>.

³⁷ Pagina home: <https://github.com/leoba/viscoll>.

³⁸ Il codice sorgente di EVT è disponibile sul sito GitHub (<https://github.com/evt-project/evt-viewer/>) ed è organizzato in 'rami' separati che duplicano una base comune

cato, anche perché impone una valutazione dello stato globale di avanzamento del programma, e in questa fase questo significa valutare quanto sia efficace questo incontro di funzionalità diverse e quali conseguenze abbia sul piano dell'interfaccia utente. La progettazione di un'interfaccia utente che risulti adeguata per compiti complessi quali la navigazione fra edizione critica e testimoni in trascrizione diplomatica, e che consenta l'integrazione delle nuove caratteristiche senza appesantire troppo l'interfaccia attuale, risulta infatti la sfida più impegnativa da affrontare. Un altro aspetto importante, inoltre, è proprio quello dei problemi a livello di interfaccia utente riscontrati nelle versioni sperimentali, quando per la prima volta si può verificare il funzionamento effettivo delle nuove caratteristiche e il loro grado di interazione nel *framework* esistente. Abbiamo infatti notato la 'collisione'³⁹ fra strumenti di evidenziazione per elementi TEI fondamentali (come abbreviazioni, normalizzazioni, *named entities*), in generale un uso eccessivo di colori a tal fine e un affollamento di evidenziazioni simili sulla pagina, con la conseguenza che gli elementi di intervento editoriale non sono messi in risalto a sufficienza.

6. Problemi e prospettive

A prescindere dai problemi generali che hanno caratterizzato questa fase dello sviluppo di EVT, sul futuro a medio/lungo termine del progetto pesa una battuta d'arresto piuttosto significativa, ovvero la necessità di migrare nuovamente il *framework* di sviluppo. Come descritto brevemente nella sezione *EVT per edizioni critiche*, si è passati da un *framework* basato sul linguaggio XSLT 2.0 a una nuova struttura interamente basata sul *framework* AngularJS, con un *parser* scritto in tale linguaggio che analizza il documento TEI e lo visualizza al volo (EVT 2).

AngularJS, sviluppato da Google, è un *framework* che ha avuto un enorme successo fra i programmatori di siti Web, e tale successo ha convinto i responsabili del progetto dell'opportunità di migliorarlo ulteriormente. Al termine di questo nuovo ciclo di sviluppo, tuttavia, la quantità e qualità di modifiche apportate ha avuto come effetto collaterale l'incompatibilità con la versione precedente, tanto che gli sviluppatori di

per il programma e permettono di sviluppare in maniera indipendente caratteristiche separate.

³⁹ Nel senso di uso dello stesso metodo di evidenziazione del testo per fenomeni differenti.

Google hanno deciso di cambiarne il nome, da AngularJS ad Angular, proprio per sottolineare il salto evolutivo. Quanto è stato implementato usando il vecchio *framework* potrà continuare a funzionare senza problemi, dato che si tratta di codice JavaScript del tutto compatibile con gli standard del Web, ed EVT potrebbe continuare a basarsi su di esso, ma questo significherebbe rinunciare ai numerosi miglioramenti introdotti in Angular⁴⁰ e ad altri che potranno essere aggiunti in futuro. Le versioni di Angular che si sono finora susseguite, infatti, sono compatibili ciascuna con le precedenti, e questo significa che eventuali nuove caratteristiche potranno essere applicate sulla base di codice esistente senza problemi. Per questo motivo da tempo è stato progettato, ed è in corso di attuazione, il *porting* del codice su Angular: un'operazione estremamente delicata, resa ancora più complessa dal gran numero di nuove caratteristiche aggiunte a EVT. Si tratta di una complicazione inaspettata, anche se non paragonabile al passaggio da XSLT a JavaScript che ha caratterizzato la genesi di EVT 2. La rapida evoluzione degli strumenti informatici di uso generale, purtroppo, tende a 'lasciare indietro' quei software che richiedono uno sviluppo meditato e un *testing* prolungato da parte degli utenti, come è il caso per EVT.

Come descritto sopra,⁴¹ malgrado le modifiche apportate nelle versioni più recenti degli schemi TEI, il modulo Critical Apparatus presenta ancora tre metodi diversi per il collegamento dell'apparato al testo critico. Si tratta di un problema al livello della marcatura che si traduce anche in una complicazione al livello implementativo: infatti solo in tempi recenti è stato aggiunto a EVT il supporto al metodo *double-end-point-attached* oltre al *parallel segmentation*. Un problema simile è costituito dalla già menzionata ridondanza della TEI, in quanto dobbiamo prevedere possibili combinazioni degli elementi per la codifica di fenomeni testuali di vario tipo, e questo impone uno sforzo supplementare a livello teorico, oltre a rendere più complessa la programmazione dei *parser* che devono analizzare ed elaborare i documenti TEI.

Usare il linguaggio JavaScript per fare il *parsing* di documenti XML, inoltre, per quanto valido in termini generali, presenta alcuni problemi di efficienza dell'elaborazione quando si supera un certo numero di elementi, problema che in EVT può essere collegato al metodo usato per convertire XML in JSON / HTML. Questo aspetto, inoltre, è esacerbato dal

⁴⁰ Non solo nuove funzionalità, ma anche una ottimizzazione del codice, la semplificazione di alcune caratteristiche e una curva di apprendimento più agevole.

⁴¹ Sezione 3: *Il modulo Critical Apparatus della TEI*.

fatto che, per mantenere l'approccio *client only*, tutti i dati necessari alla visualizzazione devono essere generati dal navigatore: ogni elemento TEI diventa un elemento HTML generico e, a seconda della densità della marcatura, questo può comportare rallentamenti anche marcati.⁴² Una possibile soluzione sarebbe il passaggio a un'architettura di tipo *client-server*, dove è il software in esecuzione sul server ad effettuare almeno parte dell'elaborazione dati, alleggerendo il compito del navigatore, ma una soluzione tradizionale in tal senso, oltre a rappresentare un ulteriore pesantissimo carico di lavoro, significherebbe allontanarsi in maniera radicale dalle scelte progettuali iniziali del progetto, e avrebbe come conseguenza immediata delle modalità di installazione, configurazione e uso di EVT assai più complesse rispetto alla situazione attuale. Al momento stiamo prendendo in considerazione l'ipotesi di aggiungere servizi di tipo RESTful⁴³ al *framework* per ottenere funzionalità di tipo server senza dover basare EVT su un server vero e proprio,⁴⁴ ma si tratta di una prospettiva strettamente collegata alla migrazione su Angular e quindi di lungo periodo.

Tornando ai problemi di progettazione dell'interfaccia utente cui si accennava all'inizio dell'articolo, non possiamo non riconoscere di essere in presenza di un paradosso: da un lato, la disponibilità di uno spazio virtualmente illimitato per i dati dell'edizione sembra garantire un distacco definitivo dal cosiddetto paradigma della pagina stampata; dall'altro, lo spazio visivo in cui mostrare tali dati è tutt'altro che infinito. Al contrario, volendo prendere in considerazione anche dispositivi di tipo mobile (per lo meno i tablet), potrebbe essere anche inferiore per quanto riguarda le dimensioni assolute rispetto a una pagina nel classico formato di un'edizione a stampa. La densità delle informazioni che è possibile visualizzare, inoltre, è tale da porre problemi non solo di *layout* generale, ma anche di navigazione: dal testo critico alle fonti o alla traduzione o al singolo testimone, da quest'ultimo al manoscritto che lo conserva, dalle scansioni del foglio di pergamena alle immagini ottenute

⁴² I navigatori moderni sono potenti, in particolare gli interpreti JavaScript che incorporano sono costantemente affinati per renderli sempre più efficienti, ma hanno comunque dei limiti, e la nostra scelta a livello di architettura generale (architettura *client only*) ha come conseguenza che il 100% dell'elaborazione è affidato all'interprete del navigatore, con il rischio di appesantirne notevolmente il funzionamento.

⁴³ Si veda in proposito l'introduzione disponibile all'URL https://access.redhat.com/documentation/en-us/red_hat_jboss_fuse/6.0/html/developing_restful_web_services/restintro.

⁴⁴ Su questo punto si veda Di Pietro, Martignano, Rosselli Del Turco, «Progettazione e implementazione di nuove funzionalità per EVT 2».

grazie al restauro virtuale; viceversa, restringendo la visuale alla singola lezione del testo critico, possiamo mostrare anche le varianti ortografiche oltre a quelle lessicografiche, possiamo collegare le singole parole a un glossario interno o a un dizionario *online*, possiamo evidenziare le *named entities* e organizzarle in liste, e molto altro ancora.

In passato sono state proposte interfacce sperimentali in 3D,⁴⁵ ma nessuna di queste ha avuto successo, pertanto è stata adottata quella che è la strategia più diffusa ed efficace, ovvero il ricorso a più 'strati' (ingl. *layer*) di componenti grafici per ottenere un effetto 3D virtuale, nascondendo parti dell'interfaccia finché l'utente non decide di attivarle. Combinando le metafore più note, come pannelli scorrevoli, schede, finestre modali etc., con una navigazione basata su 'viste' (ingl. *view*) differenti del testo e di altri dati dell'edizione, si è riusciti a collocare su uno spazio bidimensionale ridotto una grande quantità di informazioni diverse, collegate per mezzo di strumenti di navigazione intuitivi. Sarebbe interessante provvedere a una sistematizzazione, anche a livello concettuale, di quello che finora è stato soprattutto un approccio empirico.

Per quanto riguarda l'evoluzione dell'apparato critico in EVT, la funzionalità indubbiamente cruciale è l'integrazione del testo critico con le singole edizioni diplomatiche di testimoni selezionati, prevista come fase finale per il progetto delle *Leges Langobardorum*.⁴⁶ Si tratta di una funzionalità molto importante anche dal punto di vista metodologico perché offre allo studioso un insieme di strumenti di ricerca completo, permettendo di studiare un'opera sotto molti punti di vista diversi, spaziando dalla paleografia all'esame delle singole lezioni nel loro contesto. In questo modo l'edizione digitale potrà porsi come strumento di ricerca per eccellenza, offrendo la possibilità di verificare direttamente sui manoscritti le lezioni respinte o accettate. Oltre a questo e agli altri vantaggi descritti in precedenza, un'edizione che potremmo definire 'integrata' consentirebbe di superare il dissidio tra *new philology* e neo-/post-lachmannismo,⁴⁷ facilitando la collaborazione fra studiosi che prediligono l'una o l'altra di queste importanti metodologie ecdotiche.

⁴⁵ Ad esempio il progetto *Looking Glass* di Sun Microsystems, mostrato al pubblico per la prima volta nel 2003 e abbandonato già nel 2006. L'unica documentazione oggi disponibile è la pagina di Wikipedia https://en.wikipedia.org/wiki/Project_Looking_Glass.

⁴⁶ Si veda in proposito M. Buzzoni, R. Rosselli Del Turco, «Verso un'edizione digitale dell'Editto di Rotari», in *Atti del XV Seminario Avanzato in Filologia Germanica. I Longobardi in Italia: lingua e cultura*, Alessandria, Edizioni Dell'Orso, 2015, pp. 37-85.

⁴⁷ Si veda in proposito P. Monella, «L'edizione critica digitale: la critica del testo nella storia della tradizione», in M. Rosellini, E. Spangenberg Yanes (a cura di), *Textual*

7. Conclusioni

Volendo citare il titolo di uno degli articoli più recenti riguardo lo sviluppo di EVT,⁴⁸ il sentiero che conduce a uno strumento di visualizzazione maturo, flessibile e in grado di rendere onore al lavoro editoriale di un filologo, è davvero stretto e impervio. Non solo, infatti, le variabili dei problemi da risolvere sono particolarmente numerose, ma le difficoltà maggiori vengono dall'incontro di limitazioni tecniche con esigenze della critica testuale, ad esempio per quanto riguarda la capacità del formato TEI di rappresentare adeguatamente fenomeni di tipo filologico, oppure quando la necessità di gestire molte informazioni di tipo diverso in quello che resta uno spazio bidimensionale comunque finito pone seri problemi di progettazione dell'interfaccia utente. In generale, tutti i fattori di cui abbiamo discusso in questo articolo, dalla modellizzazione del testo alla sua visualizzazione, hanno un ruolo fondamentale nello sviluppo di uno strumento come EVT: dato che tali aspetti sono non solo complessi da affrontare singolarmente, ma a volte interagiscono fra loro in maniera imprevedibile e persino contraddittoria, l'obiettivo consiste nel raggiungere il miglior compromesso possibile per mezzo di una progettazione accurata, privilegiando l'aspetto filologico rispetto a tutti gli altri.

ABSTRACT

This contribution presents a publication tool for digital edition, EVT – Edition Visualization Technology, developed by the University of Pisa. Born as a visualizer for a diplomatic edition to be visualized with the reproduction of the manuscript, the Digital Vercelli Book, EVT has subsequently evolved in a powerful and flexible software, which is able to publish digital critical editions as well.

In addition, briefly describing the technical specifications of EVT, the article focuses on the issues of modelling a text and the XML/TEI coding, that is the standard for digital critical editions. Then it presents the way in which the different parts of a critical edition are processed and visualized by the software.

Philology Facing Liquid Modernity, numero speciale di *Storie e Linguaggi*, 5 (2019), pp. 61-82; e G. Chiarini, «Prospettive translachmanniane dell'eclotica», in *Eclotica e testi spagnoli: Atti del Convegno di Verona, 18-19-20 giugno 1981*, Padova, Università degli Studi di Padova, 1982, pp. 45-66.

⁴⁸ Di Pietro, Rosselli Del Turco, «Between innovation and conservation: The narrow path of UI design for the DSE».

In the last part some comments about the latest specifications are proposed, such as the visualisation of the collating of a manuscript, the integration between diplomatic and critical editions, and the future perspectives of the project.

Keywords

Textual criticism, Digital philology, Digital scholarly editions, critical apparatus, XML/TEI markup, software development.

RIASSUNTO

Questo articolo presenta uno strumento per la pubblicazione di edizioni digitali, EVT - Edition Visualization Technology, sviluppato presso l'Università di Pisa. Nato come visualizzatore per un progetto di edizione diplomatica accompagnata dalle scansioni del manoscritto, il Digital Vercelli Book, si è successivamente evoluto in un software potente e flessibile, in grado di coprire anche le esigenze relative alla pubblicazione di edizioni critiche. Oltre a descrivere brevemente le caratteristiche tecniche di EVT, l'articolo si sofferma in particolare su problemi di modellizzazione del testo e di gestione della codifica XML/TEI, lo standard di fatto per le edizioni critiche digitali, illustrando il modo in cui le diverse componenti di una edizione critica sono elaborate e visualizzate dal software. Nell'ultima parte sono proposte alcune considerazioni sulle caratteristiche più recenti, come la visualizzazione della fascicolazione di un manoscritto e l'integrazione fra edizioni diplomatiche e critiche, e sulle prospettive future del progetto.

Parole-chiave

Critica testuale, filologia digitale, edizioni scientifiche digitali, apparato critico, markup XML/TEI, sviluppo software.