

Progettazione e implementazione di nuove funzionalità per EVT 2: lo stato attuale dello sviluppo

Chiara Di Pietro,¹ Chiara Martignano,² Roberto Rosselli Del Turco³

¹M.E.T.A. S.r.l. (Pisa)

²Università di Firenze

³Università di Torino

¹dipi.chiara@gmail.com

²chiaramartignano@gmail.com

³roberto.rossellidelturco@unito.it

Abstract

EVT (Edition Visualization Technology) is a software tool to publish Digital Scholarly Editions on the Web. Its development dates back to the start of this decade, and it can boast quite an interesting development history: it was started as a Project-Based Learning initiative within the scope of a single edition project, but it has evolved in a versatile tool valued and used by many researchers. The current development version is based on the beta that has been released in October 2017: it surely is more feature-rich if compared to the first alpha version released in July 2016, however it is still very much critical edition-oriented. A full version supporting diplomatic / interpretative editions, complete with a new, powerful image viewer and a search engine, will be released in the second half of 2019. This paper will report on EVT progress and it will also propose some remarks about its development and the research questions that it entails, namely: the problems that derive from designing an User Interface that must take into account a great number of elements interacting on different layers; the new features that will be introduced in the first stable version and the impact that they will have on the final users; the soundness of the open source development model and the relationship with other projects.

EVT (Edition Visualization Technology) è un software per la pubblicazione di Edizioni Scientifiche Digitali sul Web. Il suo sviluppo risale agli inizi di questa decade, e può vantare una storia alquanto interessante: nato come una iniziativa di tipo didattico nell'ambito di uno specifico progetto di edizione digitale, nel corso del tempo si è evoluto in uno strumento flessibile apprezzato e utilizzato da molti ricercatori. La versione di sviluppo attuale si basa sulla beta che è stata pubblicata nell'ottobre del 2017: pur essendo molto più ricca di funzionalità rispetto alla prima versione alpha (rilasciata nel luglio 2016), è ancora incentrata sul supporto per edizioni critiche. Una versione completa, con il supporto per edizioni di tipo diplomatico e diplomatico-interpretativo sarà pubblicata nella prima metà del 2019. Questo contributo

intende fornire un aggiornamento sui progressi più recenti e anche proporre alcune considerazioni riguardo lo sviluppo di tale strumento, e in particolare i problemi che si incontrano nel progettare un'Interfaccia Utente: l'organizzazione in uno spazio bi-dimensionale di un gran numero di elementi posti su livelli diversi; le nuove funzionalità che saranno introdotte nella prima versione stabile e l'impatto che avranno sugli utenti; la validità del metodo di sviluppo open source e i rapporti con altri progetti.

Keywords: digital philology, digital scholarly edition, digital textual criticism, markup, TEI, critical edition, web-edition

Introduzione: breve storia di EVT

Edition Visualization Technology¹ (EVT d'ora in poi) è un software per la visualizzazione di edizioni digitali nato per rispondere ai bisogni del progetto Digital Vercelli Book (DVB d'ora in poi) [18]. Quest'ultimo risale al 2003, data in cui fu effettuata la scansione completa del manoscritto, e aveva come obiettivo la creazione di un'edizione diplomatica completa, oltre che di un facsimile digitale, del manoscritto noto come Vercelli Book (o *Codex Vercellensis*: Vercelli, Biblioteca e Archivio Capitolare, ms CXVII). Al momento in cui circa metà dei testi dell'edizione erano stati codificati secondo lo standard XML/TEI (v. le *Guidelines* TEI, [23]), i ricercatori del progetto hanno avviato un'indagine per appurare quale fosse lo stato dell'arte riguardo ai software di visualizzazione. Da questa analisi, purtroppo, è emerso che gli strumenti disponibili in quel periodo (nel 2010 circa) erano inadatti allo scopo perché di tipo proprietario, o dipendenti da uno specifico software e/o sistema operativo, o troppo complessi da installare e usare, o, ancora, così legati a un progetto specifico che adattarli alle necessità del DVB avrebbe richiesto troppo tempo e risorse.² Di fatto, svariati software presentavano più di uno degli svantaggi elencati sopra, ed è questo il motivo per cui il team al lavoro sul DVB ha deciso di creare *ex novo* uno strumento concepito appositamente per il progetto.³ Avendo ben presenti i limiti riscontrati da questo tipo di software nel corso della succitata ricognizione, lo strumento da implementare è stato impostato su alcuni di principi fondamentali: distribuito come software open source, basato sugli standard del Web (HTML, CSS, JavaScript, etc.) in modo da renderlo multi-piattaforma e indipendente da uno specifico sistema operativo, quanto più semplice e intuitivo possibile per l'utente finale e sufficientemente flessibile da essere modificato per future esigenze del progetto DVB, o per altri progetti di edizioni digitali.

- 1 Pagina home del progetto: <http://evt.labcd.unipi.it/>. Il codice è liberamente accessibile su GitHub: <https://github.com/evt-project/evt-viewer>.
- 2 Un buon esempio in proposito è il software sviluppato per l'edizione digitale dell'Exeter Book ([10]) che, pur essendo particolarmente avanzato ed efficace al momento della sua creazione, non ha avuto seguito in quanto era sia un software di tipo proprietario, sia troppo strettamente dipendente da una specifica versione di uno specifico navigatore Web (Microsoft Internet Explorer 5.5) per funzionare. Pur avendo avuto un impatto importante per quanto riguarda gli studi in merito a tale manoscritto (a tal riguardo v. *The Exeter Book's Digital Decade* <http://theheroicage.blogspot.com/2015/08/the-exeter-books-digital-decade.html>), come conseguenza di queste scelte di sviluppo le edizioni digitali basate su questo strumento non sono più utilizzabili se non per mezzo di un software di emulazione [19].
- 3 Si veda la nota precedente e [17] per ulteriori dettagli.

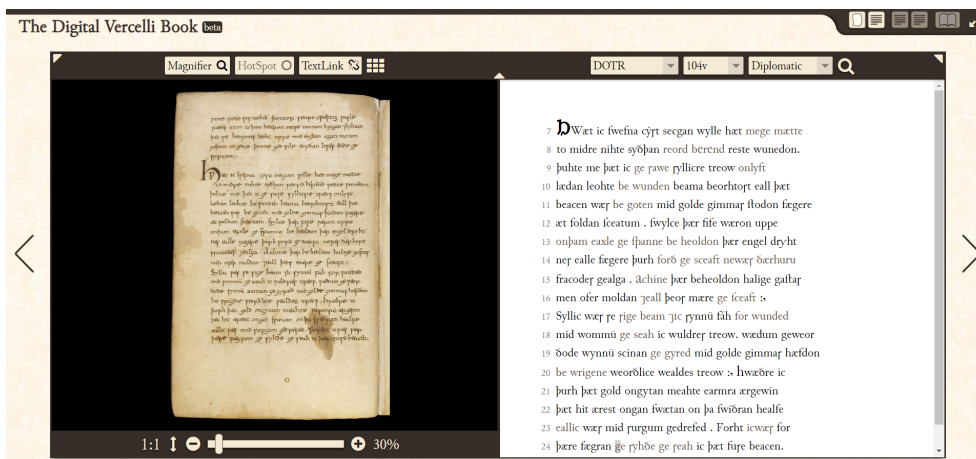


Figura 1: Il *Digital Vercelli Book*

Un primo prototipo sufficientemente completo, poi migliorato e usato per pubblicare una versione preliminare del Vercelli Book, è stato rilasciato nel dicembre 2013.⁴ In questa versione EVT si basa su una catena di trasformazioni XSLT 2.0 che, prendendo in input uno o più documenti nel formato XML/TEI versione P5, producono come risultato finale un'applicazione Web con architettura di tipo *client-only*. Questo approccio permette di copiare la cartella contenente l'edizione digitale su un qualunque server e di condividerla immediatamente, non richiedendo l'installazione di alcun tipo di software a supporto della stessa.

Collaborazione con altri progetti

Per quanto l'obiettivo iniziale fosse limitato alla pubblicazione di edizioni diplomatiche sulla base di trascrizioni codificate secondo lo standard XML/TEI e delle scansioni dei relativi manoscritti, EVT è stato progettato come uno strumento d'uso generale, in modo da poter essere utile anche in altri progetti. Di conseguenza, poco dopo la pubblicazione della prima edizione sperimentale del Vercelli Book alcuni studiosi impegnati in altri progetti di edizione digitale hanno manifestato il loro interesse verso questo strumento. Fra le varie sperimentazioni che hanno avuto luogo con questa prima versione, solo parzialmente completa, il progetto Codice Pelavicino Digitale (CPD d'ora in poi), diretto da Enrica Salvatori dell'Università di

⁴ Le prime versioni sufficientemente stabili da essere distribuite per un uso generale sono apparse nel 2014 (0.1.60-62), ma la versione 1.0, molto più completa e affidabile, risale al 2016. Al momento è disponibile la 1.2 e, entro la fine del 2019, sarà pubblicata la versione 1.3 che sarà anche l'ultima per EVT 1. Per quanto riguarda il DVB, la prima versione completa del facsimile digitale è stata pubblicata nel 2017 [18].

Pisa ([20]) e in corso di completamento (progetto avviato nel 2014), è sicuramente la collaborazione più importante.

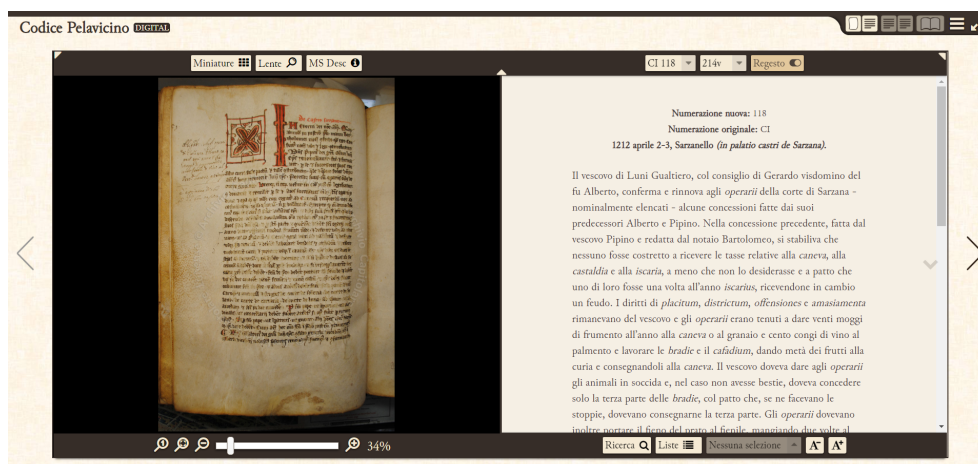


Figura 2: Il *Codex Pelavicino Digitale*

Le richieste relative a questa edizione erano solo in parte sovrapponibili a quelle del Vercelli Book Digitale: a una base comune (la visualizzazione del testo dell'edizione accanto alle immagini del manoscritto, il collegamento testo-immagine, l'inclusione delle informazioni contenute nell'elemento `<msDesc>` come paratesto, etc.) si aggiungeva un numero piuttosto cospicuo di nuove richieste. La presentazione del regesto relativo a ogni documento in primo luogo, ma anche caratteristiche importanti come la gestione delle *named entities* e il supporto a note critiche⁵ e di commento, oltre alla gestione dei riferimenti bibliografici, trasformati in collegamenti ipertestuali a una raccolta bibliografica valida per tutto il corpus di documenti.

Il lavoro portato a termine per il CPD costituisce sicuramente il progresso più significativo in questa fase dell'evoluzione di EVT. La collaborazione con altri progetti di edizione, in generale, si è rivelata l'impulso più importante per l'ampliamento delle funzionalità iniziali, soprattutto se finanziate da tali progetti, avendo come esito l'introduzione di nuove caratteristiche così lontane dagli obiettivi del VBD da non essere state prese in considerazione nella fase di progettazione. Questo è particolarmente vero per quanto riguarda, ad esempio, l'introduzione di un supporto alle edizioni genetiche ([8]) o la pubblicazione di documenti risalenti alla storia recente ([3]).

Proprio il fatto che tali caratteristiche non erano state in alcun modo contemplate al momento di stabilire il framework di programmazione di EVT, tuttavia, ha portato alla luce i limiti di flessibilità ed espandibilità di quest'ultimo.

5 Di fatto il CPD è un'edizione critica, per quanto basata su un unico testimone.

EVT 2: una nuova architettura e nuovi obiettivi

Una delle nuove caratteristiche da implementare in EVT 1, forse la più importante essendo fondamentale sia per il DVB, sia per altri studiosi che hanno avanzato richieste in tal senso, era il supporto per edizioni critiche complesse, basate su tradizioni testuali caratterizzate da due o più testimoni. Anche in questo caso, per la codifica dei testi era previsto il ricorso agli schemi TEI, pur con qualche esitazione dovuta al modulo *Critical Apparatus* ancora non del tutto definito nel 2015, e successivamente migliorato con modifiche apportate agli schemi di codifica TEI rilasciati negli ultimi anni.

Come accennato nella sezione precedente, al momento di definire questa nuova caratteristica eravamo già arrivati alla conclusione che il framework di programmazione esistente era inadatto per l'aggiunta di questa importante funzionalità. La base di codice XSLT 2 che costituiva il nucleo centrale di EVT, infatti, non solo era cresciuta in maniera considerevole, ma era divenuta complicata e piuttosto difficile da capire e modificare, soprattutto per i nuovi collaboratori. Il numero di elementi grafici da gestire era ugualmente aumentato, rendendo il *layout* generale dell'applicazione complesso da configurare e difficile da gestire per quanto riguarda l'interazione fra i diversi oggetti della GUI (Graphical User Interface). In nessun modo avremmo potuto aggiungere questo tipo di funzionalità alla base di codice esistente, e un intero set di nuovi oggetti al *layout* dell'applicazione Web, senza incorrere in complicazioni di ogni tipo e, soprattutto, senza rischiare una generale instabilità del sistema. Avevamo raggiunto il punto in cui sarebbe stato necessario disporre di una flessibilità e modularità molto maggiori per poter sviluppare nuove funzionalità e inserirle accanto a quelle già esistenti.

Per questi motivi decidemmo di rimodellare l'intera struttura del software, con l'obiettivo di renderlo più leggero, più facilmente gestibile grazie all'eliminazione della fase di *building* (la catena di trasformazioni XSLT 2 che deve essere avviata dall'editore), e soprattutto più flessibile e modulare. Per raggiungere questi obiettivi abbiamo scelto di basare il nuovo framework di sviluppo sull'architettura Model View Controller (MVC), un design pattern molto apprezzato nella programmazione a oggetti, da implementare grazie al framework AngularJS (<https://angularjs.org/>), uno dei più utilizzati in quegli anni nell'ambito dello sviluppo di applicazioni web a pagina singola (*Single Page Application* o SPA).⁶

Dopo una gestazione piuttosto lunga, comprensiva di molte discussioni sulle possibili soluzioni di presentazione dell'apparato critico, una prima versione ancora incompleta⁷ ma già in grado di mostrare le potenzialità del nuovo framework e di supportare edizioni critiche multi-testimoniali, è stata rilasciata nel luglio 2016 grazie al lavoro di Chiara Di Pietro ([5]). Una versione più avanzata, che presenta un insieme di caratteristiche per il supporto all'edizione critica più completo e sofisticato, è stata resa disponibile nell'ottobre 2017 grazie al lavoro di Chiara Martignano ([9]).

⁶ Per maggiori informazioni su questa architettura e sul framework AngularJS si veda [5].

⁷ Non soltanto per quanto riguarda il supporto alle edizioni critiche, ma soprattutto per il fatto che tutte le caratteristiche di EVT 1, in particolare il supporto per le immagini e il motore di ricerca, non erano state ancora implementate nel nuovo framework.



Figura 3: La prima versione sperimentale (alpha) di EVT 2

La versione in corso di sviluppo è quasi completa rispetto agli obiettivi iniziali, e il processo di implementazione di tutte le caratteristiche presenti in EVT 1 (motore di ricerca, supporto per trascrizioni diplomatiche, visualizzatore di immagini e supporto per il collegamento testo-immagine, supporto per le *named entities* e liste di vario tipo, etc.) è quasi concluso. Per la prossima release l'obiettivo è rendere EVT 2 uno strumento completo, in grado di pubblicare non solo edizioni diplomatiche ed edizioni critiche, ma anche un tipo di edizione "integrata", attraverso una visualizzazione sinottica e una interconnessione delle singole componenti, che faccia dialogare queste due diverse metodologie ecdotiche. I principi di funzionamento restano invariati, dato che anche EVT 2 accetta come input edizioni codificate usando lo standard XML/TEI, ma la struttura del software è stata progettata in maniera tale da poter affiancare al *parser* di documenti XML/TEI un *parser* equivalente per edizioni preparate usando altri linguaggi di markup. Per quanto riguarda la flessibilità e modularità del software, basterebbe questa caratteristica a rendere EVT 2 migliore rispetto alla versione precedente, condizionata dai limiti di XSLT.

Molte delle caratteristiche già implementate o in corso di sviluppo sono dovute al feedback e alla collaborazione con il progetto PhiBor - Philosophy on the Border of Civilizations ([11]), infatti l'esempio predefinito usato nella versione beta (si veda la Figura 4) si basa su un testo preparato come esperimento di codifica in preparazione dell'edizione digitale della traduzione della *Metafisica* di Aristotele fatta da Avicenna. Attualmente allo sviluppo di EVT 2 sta contribuendo anche il progetto *Marco Polo Digitale*, dell'università Ca' Foscari Venezia, che ha come obiettivo la pubblicazione del *Le Devisement dou monde* (o *Milione*, secondo la tradizione italiana) di Marco Polo. Come per il Codice Pelavicino Digitale, la collaborazione ravvicinata con progetti che mirano a pubblicare edizioni digitali complesse, sofisticate e tendenzialmente molto differenti fra loro, si è dimostrata essenziale per migliorare e aumentare l'insieme di funzionalità che EVT può offrire alla comunità accademica.

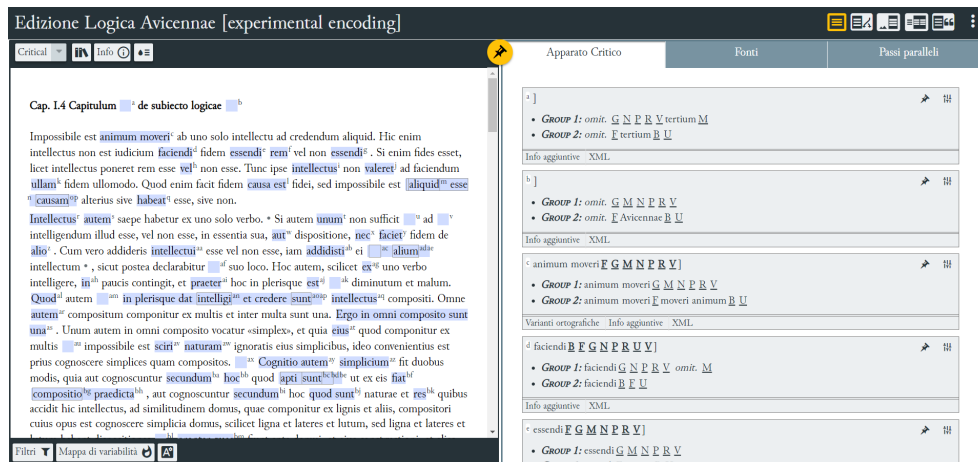


Figura 4: La versione beta di EVT 2

Le prime versioni

Per quanto incompleta sotto molti punti di vista, la prima versione alpha (luglio 2016) presentava già un supporto soddisfacente per le edizioni critiche e soprattutto mostrava come la scelta di riscrivere da zero l'intera base di codice, e di introdurre importanti novità anche per quanto riguarda il *layout* dell'interfaccia utente, fosse giustificata. In questa versione il documento XML/TEI che contiene l'edizione critica non viene trasformato in pagine HTML statiche da navigare, come avveniva in EVT 1: i dati dell'edizione sono invece salvati in una struttura JSON volatile che li rende disponibili al momento in cui devono essere visualizzati. Come per EVT 1, l'aspetto finale del testo viene gestito per mezzo di fogli di stile CSS. Altre caratteristiche importanti presenti nella versione alpha:

- *supporto per edizione critica*: le voci di apparato critico sono mostrate *inline* attraverso box espandibili e presentano contenuti aggiuntivi rispetto al tradizionale apparato; inoltre, è resa possibile l'applicazione di filtri, costruiti automaticamente in base alla tipologia delle voci stesse definita dall'editore e ad altri metadati ad esse associati; infine, lo strumento *heat map* permette di visualizzare più velocemente i punti del testo critico in cui si concentra una maggior frequenza di varianti;
- *modalità collazione*: è prevista una modalità di visualizzazione (ingl. *view*) gestita graficamente in maniera dinamica e *responsive*, ma impostando una larghezza minima per lo spazio destinato al testo critico e ai singoli testimoni; si basa sul metodo *parallel segmentation* della TEI per generare il testo di tutti i testimoni sulla base del documento dell'edizione critica;
- *strumento segnalibro*: per facilitare la citabilità e lo scambio di informazioni fra studiosi, questo strumento permette di salvare la posizione precisa in cui si trova il lettore (documento visualizzato, modalità di visualizzazione, pagina e livello di

edizione, voci di apparato aperte ed eventuali testimoni nella modalità collazione) in modo da poterlo salvare e inviare come segnalibro;

- *personalizzazione*: sia il *layout* generale dell'applicazione Web, sia l'aspetto del testo possono essere personalizzati grazie a un file di configurazione ricco di opzioni e alla possibilità di scrivere propri fogli di stile CSS per variare l'aspetto del testo.

Particolare attenzione è stata data al metodo di visualizzazione delle voci di apparato. Una volta rilasciata la versione alpha, è stato possibile raccogliere i primi riscontri sull'implementazione dell'apparato critico e riflettere su quali migliorie e nuove funzionalità realizzare all'interno del software. La collaborazione con il progetto PhiBor, menzionato sopra, è stata fondamentale, in quanto il progetto ha come scopo la realizzazione di un'edizione critica digitale molto complessa, ovvero un'edizione completa e multilingue del "Libro della cura" del filosofo arabo medievale Avicenna. L'edizione di questo testo particolarmente articolato ha permesso di individuare come funzionalità da sviluppare in EVT:

- il supporto di apparati critici completi di passi paralleli e apparato delle fonti;
- la gestione di casi di "doppia (o multipla) recensione";
- la presentazione di testi provenienti da diversi file XML all'interno di un'unica interfaccia; questa funzionalità in particolare è stata sfruttata per mettere a confronto il testo critico dell'edizione con il testo delle relative fonti;
- la presentazione di tutte le diverse entrate d'apparato nel loro insieme.

L'ultimo punto è emerso da una riflessione più generale, iniziata in maniera autonoma, come proseguimento degli studi HCI portati a termine per la versione alpha, e in seguito condotta dal gruppo di EVT insieme con i filologi del progetto PhiBor. Si è palesata, infatti, l'utilità di consultare tutte le entrate d'apparato nel loro insieme, di renderle più facilmente accessibili all'interno dell'interfaccia, e, in generale, l'opportunità di presentare la sostanza essenziale di qualunque edizione critica, ovvero le voci dell'apparato, in maniera "tradizionale". In altre parole, si trattava di rivedere alcune scelte grafiche dell'interfaccia, per consentire all'utente di leggere le edizioni in EVT in modo analogo a come leggerebbe un'edizione stampata. Le voci d'apparato sono state collocate in un'area del *layout* generale dell'edizione immediatamente riconoscibile, esattamente come nel caso delle fasce di apparato in una edizione a stampa. Sfruttando lo spazio presente ai lati della modalità "solo testo critico" della versione alpha, è stato aggiunto un pannello all'interno del quale trovano posto sia la fascia di apparato critico propriamente detto, sia l'apparato delle fonti (se presente) e anche passi paralleli o fonti dirette (anche queste ove presenti) (cfr. Figura 5). Il formato della voce di apparato segue le convenzioni editoriali classiche, ed è più visibile rispetto ad altri elementi della voce (note critiche, informazioni varie, il codice sorgente della stessa) che sono accessibili al click sul pulsante relativo. La struttura della voce di apparato resta immutata nella modalità collazione, quando il pannello per gli apparati non è più disponibile per esigenze di spazio, ma è presentata *inline* come pannello a scomparsa, attivabile cliccando sulla variante evidenziata nel testo critico.

Le modifiche apportate al software nella versione beta avevano, dunque, come principali

obiettivi quello di rendere EVT uno strumento completo e adatto anche per edizioni critiche complesse e di presentare alcuni elementi dell'interfaccia legati all'apparato critico in un modo più vicino alle convenzioni tipografiche delle edizioni critiche a stampa.⁸ Queste modifiche possono essere riassunte come segue:

- 1) nella modalità solo testo, la creazione di un pannello, posizionato accanto al testo critico, che presenta tre diverse tipologie di apparato (apparato delle varianti, passi paralleli e apparato delle fonti) nella loro interezza;
- 2) la possibilità di presentare le voci d'apparato in due o più gruppi distinti in base a raggruppamenti di testimoni stabiliti dall'editore all'interno del file di configurazione;
- 3) la lettura in parallelo e la gestione di versioni multiple del testo all'interno di una nuova apposita modalità di visualizzazione;
- 4) la lettura in parallelo in una nuova apposita modalità di visualizzazione del testo critico a confronto con le proprie fonti.

Una funzionalità introdotta nella versione alpha e migliorata nella beta è quella del cosiddetto *pin frame*: si tratta di un pannello a scomparsa visibile nella modalità "testo critico e apparati" che raccoglie voci di apparato (v. alpha) e anche altre componenti testuali, in particolare occorrenze di *named entities* (v. beta). L'utente può selezionare gli oggetti da aggiungere al *pin frame* grazie all'icona a forma di spillo (ingl. *pin*) visibile anche in altre modalità di visualizzazione; per accedere alla raccolta deve tuttavia espandere il pannello accessibile nella modalità sopra citata. Al momento lo scopo di questo pannello è quello di permettere un rapido accesso a punti degli apparati o del testo ritenuti interessanti dal lettore, e quindi si configura come una raccolta di segnalibri che possono rendere più rapido e semplice l'accesso al testo.

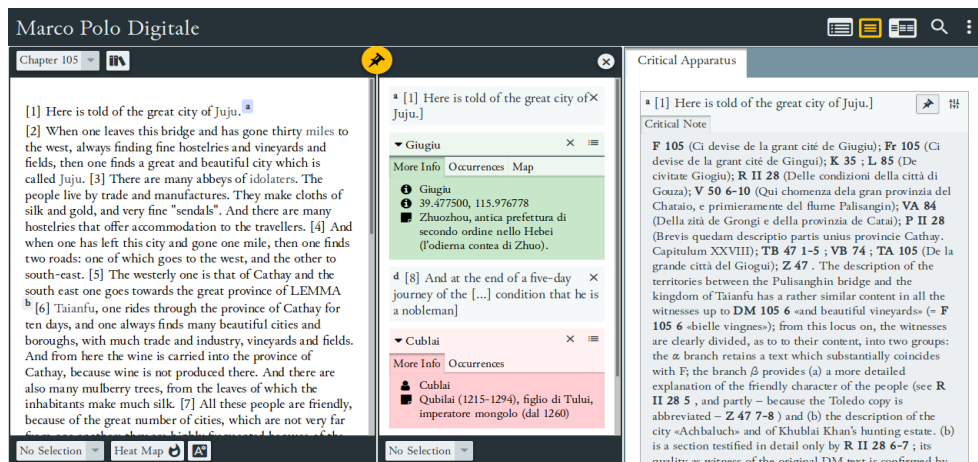


Figura 5: Il pin frame: sono visibili sia voci di apparato, in forma di nota critica, sia alcune named entities

⁸ Per un approfondimento relativo a questo aspetto si veda [6].

Lavori in corso

Il ritardo nella pubblicazione dell'ultima beta, oltre alla scarsità di risorse adeguate per uno sviluppo continuato nel tempo, è dovuto alla natura stessa del processo di implementazione di nuove funzionalità per EVT: molte di queste, in particolare quelle meno impegnative (ma ci sono eccezioni importanti, come le modifiche richieste da progetti di edizione o singoli studiosi), sono sviluppate come tirocini curriculari e tesi triennali presso il corso di laurea in Informatica Umanistica dell'Università di Pisa. Per loro stessa natura, queste forme di tutoraggio degli studenti richiedono molto tempo per essere avviate e poi seguite fino al completamento, perché lo sviluppo con Angular è più impegnativo rispetto alla semplice programmazione in JavaScript. A volte, purtroppo, l'inserimento dello studente nel team di sviluppo non ha successo, e alcune di queste collaborazioni sono state interrotte prematuramente. Nella media, tuttavia, abbiamo avuto studenti in grado di trarre profitto dal lavoro svolto, arricchendo in maniera considerevole le proprie competenze e fornendo un contributo significativo allo sviluppo di EVT.

Le funzionalità sviluppate grazie alla collaborazione con progetti quali PhiBor e il Marco Polo Digitale, viceversa, hanno tempi di esecuzione più definiti, ma qualunque nuova caratteristica deve essere integrata con le altre, operazione che richiede tempo e pazienza perché il momento in cui il codice di sviluppo viene allineato ai singoli "rami", importando le porzioni che contengono tali funzionalità, è anche il momento in cui si possono verificare conflitti fra rami diversi, a volte particolarmente problematici da risolvere.

Come si può constatare dalla lista proposta qui di seguito, nei mesi scorsi il team di sviluppo ha lavorato a molte *feature* importanti, tali da migliorare notevolmente l'utilità di EVT per lo studioso.

- *Navigazione*. Il supporto per edizioni diplomatiche con immagini digitalizzate a fronte è un passo importante per raggiungere la *feature parity* con la versione 1, al momento in cui scriviamo questo articolo è stato ultimato, ma necessita ancora di rifinitura, ad esempio per quanto riguarda la navigazione di immagini e documenti: per far questo è stata aggiunta una barra con frecce di navigazione standard e uno *slider*, ma sono previsti ulteriori miglioramenti (in particolare pulsanti di navigazione simili a quelli di EVT 1, e il supporto per un indice dei contenuti navigabile).
- *Motore di ricerca*. Altro importante tassello mancante nelle prime release di EVT 2, è stato implementato *ex novo* grazie al lavoro di Giulia Cacioli ([4]) che è riuscita a migliorare la funzionalità equivalente in EVT 1, aggiungendo nuove opzioni quali la ricerca per parola esatta, e a preparare il terreno per future espansioni.

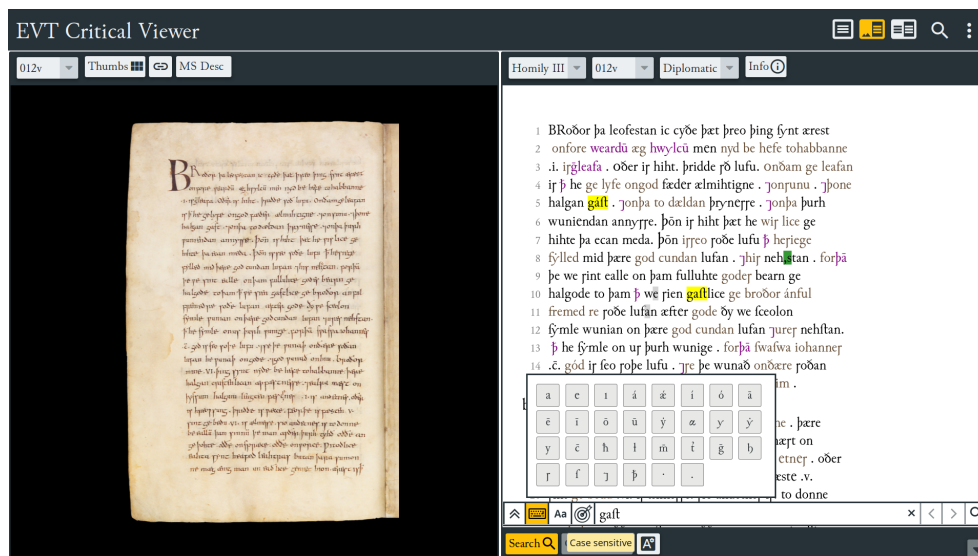


Figura 6: Il nuovo motore di ricerca in azione con in evidenza la tastiera virtuale

- *Integrazione di VisColl.* VisColl è un software che permette di visualizzare e navigare graficamente la struttura della fascicolazione di un manoscritto descritta in un documento XML; essendo basato su fogli di stile XSLT 2 è stato necessario aggiungere un *parser* appropriato (Saxon-JS)⁹ in quanto EVT 2 ha abbandonato completamente il supporto per questo linguaggio.

9) Pagina home di Saxon-JS: <http://www.saxonica.com/saxon-js/index.xml>. La versione testata nella prima implementazione di questa funzionalità ha manifestato alcune problematiche che ci hanno indotto a preferire, come soluzione temporanea, l'uso del processore XSLT 2 incluso con Oxygen XML Editor per la generazione dei diagrammi SVG. Recentemente è stata pubblicata una versione aggiornata di Saxon-JS (v. 1.2.0) pertanto sarà possibile effettuare nuovi test in modo da implementare questa trasformazione come previsto, senza richiedere l'uso di Oxygen.

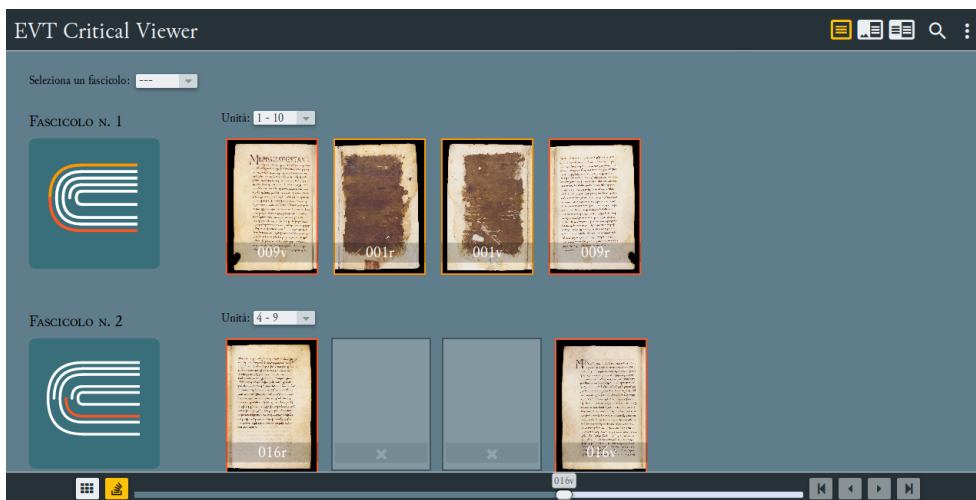


Figura 7: La struttura fascicolare del Vercelli Book visualizzata con VisColl in EVT 2, visibile anche la nuova barra di navigazione

- *Web app di configurazione.* Data la crescente complessità, per numero e funzionalità collegate, delle opzioni di configurazione è stata creata un'applicazione Web sperimentale in modo da permettere al responsabile dell'edizione di curare la configurazione della stessa grazie un'interfaccia semplice e intuitiva. Questa applicazione è disponibile all'URL <http://evt.labcd.unipi.it/evt2-config/> e la sua implementazione procede di pari passo con l'aggiunta di nuove opzioni di configurazione.
- *Supporto per la geo-localizzazione e la cronologia.* Il collegamento fra le *named entities* presenti in un documento e la visualizzazione dei luoghi su server quali Google Maps e OpenStreetMap,¹⁰ come pure la creazione di una cronologia generale degli eventi che metta insieme personaggi storici, date e luoghi, sono caratteristiche utilissime per ricerche mirate all'interno del testo di documenti TEI, in particolare di genere storico. Questa funzionalità, pur se limitatamente alla localizzazione spaziale per il momento, è in corso di sviluppo per il progetto Marco Polo Digitale.

Per garantire la stabilità del framework ciascuna di queste caratteristiche costituisce un sotto-progetto separato e viene sviluppata in un *branch* separato sul repository GitHub: questo permette di sperimentare liberamente senza il timore di provocare danni alla versione stabile. Quando ciascun *branch* raggiunge una maturità sufficiente, e i test confermano che gli obiettivi desiderati sono stati raggiunti e che eventuali bug sono stati corretti, si procede a un "riallineamento" di tutti i *branch* facendoli confluire in quello stabile. Questa operazione è alquanto delicata e può provocare conflitti tra versioni diverse del software, conflitti che devono essere necessariamente risolti altrimenti non sarà possibile fare un *merge* di tutte le funzionalità

10 Pagina home: <https://www.openstreetmap.org/>.

sviluppate.

Sviluppi futuri

Se le funzionalità descritte nella sezione precedente sono in fase di completamento, comunque in una fase avanzata di implementazione, altre sono invece allo stadio di progettazione, e altre ancora sono al momento dei semplici desiderata, o possibili percorsi di sviluppo che non si è ancora deciso se percorrere fino in fondo. In ordine crescente di complessità tecnica, queste sono le principali linee di sviluppo previste:

- *Annotazione e commenti dell'utente.* Nel panorama del software attualmente disponibile non vi è carenza di programmi che permettano di prendere appunti riguardo il testo o le immagini che si stanno consultando, da semplici editor ai programmi di elaborazione testi più noti; per quanto efficaci, sarebbe interessante affiancare a tali strumenti una funzionalità di annotazione interna a EVT, approfittando anche dell'attuale *pin frame* per fornire all'utente una vera e propria area di lavoro, e se possibile anche l'opzione di esportare le proprie note in un formato riutilizzabile in un *word processor*, ma che mantenga i collegamenti agli oggetti annotati. Nella forma più semplice questo strumento potrebbe salvare le annotazioni dell'utente su disco (come documento da scaricare) o nella *cache* del navigatore; opzioni più sofisticate e complesse, come la possibilità di salvare le proprie note sul server che ospita l'edizione digitale, e condividerle con altri utenti, richiederebbero invece una architettura *client-server* (v. infra).¹¹
- **Edizione integrata.** Già nel 2007, descrivendo il processo editoriale che aveva portato alla pubblicazione del *De Monarchia* di Dante Alighieri a cura di Prue Shaw ([22]), Peter Robinson aveva teorizzato il concetto di *full digital edition* ([13], [14]), ovvero un'edizione digitale basata sulla trascrizione diplomatica di tutti i principali testimoni di un'opera, sulla scansione dei manoscritti corrispondenti e sull'integrazione di questi dati con un'edizione critica. Una volta preparati tutti i dati, il testo critico, in particolare per quanto riguarda le varianti, può essere messo in collegamento con le singole edizioni diplomatiche e, come passo ulteriore, con le immagini corrispondenti. Questo sviluppo del concetto di edizione digitale è particolarmente interessante a livello teorico, perché permetterebbe non solo di gettare un ponte fra seguaci della *new philology* e studiosi di tendenza neo-lachmanniana,¹² ma anche di seguire il percorso dell'editore fino in fondo, verificando le scelte filologiche direttamente nel contesto del singolo testimone e, se necessario, visualizzando le scansioni del

11 Si noti che in caso di annotazioni condivise sul Web il formato dei dati diventa rilevante, in tal caso la scelta più opportuna sarebbe lo standard definito dal consorzio W3C: si veda il sito <https://www.w3.org/annotation/>.

12 Non solo, renderebbe anche possibile immaginare un'edizione "aperta", un ambiente di editing collaborativo all'interno del quale uno o più filologi si dedica a inserire nuovi testi e nuovo materiale.

manoscritto corrispondente.¹³

- *Migrazione su Angular 2+*. Una caratteristica del mondo informatico che invariabilmente stupisce chi viene da quello umanistico è il ritmo incredibilmente rapido con cui si susseguono le innovazioni tecnologiche: un framework come AngularJS è stato creato, reso disponibile per gli sviluppatori Web e si è diffuso rapidamente, ma nel giro di pochissimi anni gli autori hanno fatto seguito con una versione che non solo espande le funzionalità iniziali, ma corregge alcuni difetti iniziali di progettazione. Il prezzo da pagare per questa evoluzione naturale del software è l'incompatibilità fra la versione iniziale, ancora disponibile con il nome AngularJS, e le versioni successive. Anche il nome stesso del framework è cambiato, dalla v. 2 in poi è semplicemente Angular (<https://angular.io/>).
- *Nuova architettura client-server*. Il fatto che EVT sia basato su un'architettura *client-only* costituisce un indubbio punto di forza, soprattutto per quanto riguarda la facilità con la quale si può pubblicare un'edizione sul Web: basta infatti copiare la cartella di EVT su un qualsiasi server e accedere al file `index.html` per avviare la navigazione. Dal punto di vista dell'implementazione, tuttavia, costituisce un punto debole, in quanto l'intera elaborazione dei dati è affidata al *client*, ovvero al navigatore Web utilizzato, e questo può causare sia rallentamenti dovuti a un sovraccarico di operazioni, sia limitazioni alle funzionalità desiderate, ad esempio quelle di annotazione e ricerca. La migrazione sul framework Angular 2+ potrebbe essere l'occasione più appropriata per introdurre una funzionalità server in EVT, accoppiando dei servizi di tipo RESTful al framework, piuttosto che aggiungendo un server vero e proprio (anche in forma ridotta), in maniera tale da mantenere il software leggero e veloce (oltre che semplice da installare e configurare) e da ottenere una separazione effettiva fra i dati dell'edizione e l'interfaccia utente che li presenta. Da questa versione sarebbe inoltre semplice derivare, se così desiderato o necessario, una versione basata sul precedente modello *client-only*.

Il lavoro sull'edizione integrata è già in corso e questa nuova caratteristica potrebbe essere introdotta nella versione successiva alla seconda beta. Inoltre è cominciato il lavoro esplorativo sulle annotazioni dell'utente e per quanto riguarda la futura migrazione su Angular 2+. La nuova architettura *client-server* basata su servizi RESTful, viceversa, è una prospettiva più lontana, ma comunque centrale nei futuri piani di sviluppo di EVT.

Conclusioni

Lo sviluppo di software che superi una determinata soglia di complessità è sempre un compito impegnativo, sia per l'inevitabile accumulo di piccoli conflitti fra moduli diversi (per non

¹³ Niente può sostituire un esame autoptico di un manoscritto per apprezzare determinati dettagli, ma un restauro virtuale fatto bene ci si avvicina moltissimo, e l'evidenziazione di aree del manoscritto grazie a scansioni con illuminazioni particolari (UV, Infrarossi) possono portare alla luce testo non altrimenti visibile a occhio nudo.

parlare dei bug), sia pure, nel caso specifico, per l'estrema "mobilità" ed eterogeneità dei dati da gestire. Diviene inoltre fondamentale una gestione ottimizzata delle risorse, che non riguarda soltanto (e neanche in prevalenza) l'aspetto economico, in modo da far fruttare al meglio le competenze individuali e i finanziamenti mirati allo sviluppo di specifiche funzionalità. Ultimo, ma non per importanza, la costante ricerca del miglior compromesso possibile per quanto riguarda l'interfaccia utente: come si può arguire da quanto discusso sopra, presentare informazioni così eterogenee e tuttavia strettamente interconnesse fra loro rappresenta un compito assai difficile, soprattutto se si considerano i limiti di spazio nei quali è possibile inserirle.

Malgrado queste difficoltà lo sviluppo di EVT prosegue alacremente e il percorso delineato è ricco di promesse: grazie a un team di sviluppatori appassionati e di grande esperienza faremo del nostro meglio per mantenerle. Il feedback dei nostri utenti e le proposte di collaborazione (oltre che lo sviluppo autonomo nella direzione desiderata, il codice è disponibile per possibili *fork* sul repository GitHub) da parte di altri progetti rappresentano, come sempre, la miglior motivazione possibile per la continuazione di questo progetto.

References

- [1] Angular. 'Angular'. <https://angular.io/>.
- [2] AngularJS. 'AngularJS'. <https://angularjs.org/>.
- [3] Armaselu, Florentina. 2015. 'Towards an XML-TEI Edition of Diplomatic Documents – Part 2. EVT Visualisation'. Billet. *Digital Humanities LAB at CVCE* (blog). <https://cvcedhlab.hypotheses.org/149>.
- [4] Cacioli, Giulia. 2019. 'Filologia e Information Retrieval: Progettazione e Sviluppo Di Un Motore Di Ricerca per EV'. Master's Thesis, Università di Pisa. ETD - Electronic theses and dissertations repository. <http://etd.adm.unipi.it/theses/available/etd-01152019-191842/>.
- [5] Di Pietro, Chiara. 2016. 'EVT per Le Edizioni Critiche Digitali: Progettazione e Sviluppo Di Una Nuova GUI Basata Sullo Schema Progettuale MVC'. Master's Thesis, Università di Pisa. ETD - Electronic theses and dissertations repository. <https://etd.adm.unipi.it/theses/available/etd-03292016-203644/>.
- [6] Di Pietro, Chiara, and Roberto Rosselli Del Turco. 2018. 'Between Innovation and Conservation: The Narrow Path of UI Design for the DSE'. In *Digital Scholarly Editions as Interfaces*, edited by Roman Bleier, Martina Bürgermeister, Helmut W. Klug, Frederike Neuber, and Gerlinde Schneider, Vol. 12. Schriften Des Instituts Für Dokumentologie Und Editorik. Norderstedt: Books on Demand.
- [7] *EVT - Edition Visualization Technology*. Home page: <http://evt.labcd.unipi.it/>. Development blog: <http://visualizationtechnology.wordpress.com/>. Repository per la

- distribuzione: . Codice sorgente: <https://github.com/evt-project/evt-viewer>.
- [8] Leclerc, Elise, and Samantha Saïdi. 2017. 'Élaborer La Version Numérique d'une Édition Analytique de l'exorde de La Storia d'Italia de Francesco Guicciardini'. In *Journée d'étude "Philologie Numérique et Méthodes Informatiques d'analyse de Textes"*. Lyon, France: ENS Lyon, Laboratoire Triangle. <https://hal.archives-ouvertes.fr/hal-02020479>.
- [9] Martignano, Chiara. 2017. 'Progettazione e Sviluppo Di Un Apparato Critico Modellato Sulla Tradizione a Stampa in EVT'. Master's Thesis, Università di Pisa. ETD - Electronic theses and dissertations repository.
- [10] Muir, Bernard J. 2006. *The Electronic Exeter Anthology of Old English Poetry: An Edition of Exeter Dean and Chapter MS 3501. DVD*. Exeter: University of Exeter Press.
- [11] 'Philosophy on the Border of Civilizations and Intellectual Endeavours: Towards a Critical Edition of the Metaphysics (Ilāhiyyāt of Kitāb al-Šifā') of Avicenna (Ibn Sīnā)', Project. <http://www.avicennaproject.eu/>.
- [12] Porter, Dot. *VisColl*. XSLT. <https://github.com/leoba/VisColl>.
- [13] Robinson, Peter. 2007. 'Electronic Editions Which We Have Made and Which We Want to Make'. In *Digital Philology and Medieval Texts*, edited by Arianna Ciula and Francesco Stella, 1–12. Pisa: Pacini.
- [14] Robinson, Peter. 2013. 'Towards a Theory of Digital Editions'. *Variants* 10: 105–32.
- [15] Rosselli Del Turco, Roberto. 2014a. 'EVT Development: An Update (and Quite a Bit of History)'. *Edition Visualization Technology Blog*. <https://visualizationtechnology.wordpress.com/2014/01/26/evt-development-an-update-and-quite-a-bit-of-history/>.
- [16] Rosselli Del Turco, Roberto. 2014b. 'EVT development: an update (and quite a bit of history)'. *Edition Visualization Technology* (blog), 26 January 2014. <https://visualizationtechnology.wordpress.com/2014/01/26/evt-development-an-update-and-quite-a-bit-of-history/>.
- [17] Rosselli Del Turco Roberto, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti, and Jacopo Pugliese. 2015. 'Edition Visualization Technology: A Simple Tool to Visualize TEI-Based Digital Editions'. *Journal of the Text Encoding Initiative* 8. <https://doi.org/10.4000/jtei.1077>.
- [18] Rosselli Del Turco, Roberto. 2017. *The Digital Vercelli Book*. <https://www.collane.unito.it/oa/items/show/111#?c=0&m=0&s=0&cv=0>.
- [19] Rosselli Del Turco, Roberto. 2018. 'Il Manoscritto e Le Edizioni dell'Exeter Book'. In *Atti Del XVII Seminario Avanzato in Filologia Germanica. Lettura Del CODEX EXONIENSIS: Le Elegie Anglosassoni*. Alessandria: Dell'Orso Editore.

- [20] Salvatori, Enrica, Edilio Riccardini, Laura Balletto, Roberto Rosselli Del Turco, Chiara Alzetta, Chiara Di Pietro, Chiara Mannari, Raffaele Masotti, and Alessio Miaschi. 2014. *Codice Pelavicino. Edizione Digitale*. <http://pelavicino.labcd.unipi.it/>.
- [21] 'Saxon-JS'. <https://www.saxonica.com/saxon-js/index.xml>.
- [22] Shaw, Prue. 2006. *Dante Alighieri: Monarchia. An Electronic Edition on DVD-Rom*. Leicester: Scholarly Digital Editions; Società Dantesca Italiana.
- [23] TEI Consortium. 2019. 'TEI P5: Guidelines for Electronic Text Encoding and Interchange [3.6.0]'. <https://tei-c.org/Guidelines/P5/>.

Ultimo accesso URLs: 19th October 2019.