



UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

Relazione

**Sviluppo della funzionalità di confronto tra
manoscritti diversi in EVT 3**

Candidato: *Sara Maenza*

Relatore: *Prof. Paolo Milazzo*

Correlatore: *Prof. Roberto Rosselli del Turco*

Anno Accademico 2020-2021

Indice

| | |
|--|-----------|
| Introduzione | 3 |
| 1. Il facsimile digitale: diffusione e vantaggi | 5 |
| 1.1 Il lavoro degli amanuensi | 5 |
| 1.2 Le discipline che studiano la scrittura | 6 |
| 1.3 Il facsimile fotografico | 9 |
| 1.3.1 L'importanza dei manoscritti | 9 |
| 1.3.2 Difficoltà di analisi | 10 |
| 1.3.3 Diffusione del facsimile fotografico | 10 |
| 1.3.4 Il problema dei costi | 12 |
| 1.4 Il facsimile digitale | 13 |
| 1.4.1 Strumenti utili per lo studio | 14 |
| 2. EVT e il facsimile digitale | 16 |
| 2.1 EVT: <i>Edition Visualization Technology</i> | 16 |
| 2.2 EVT 1: <i>view image-text, bookreader</i> | 17 |
| 2.3 EVT 2: <i>view image-text</i> | 18 |
| 2.4 EVT 3 | 20 |
| 2.4.1 Estensione della funzionalità di <msDesc> multipli | 21 |
| 2.4.2 Estensione della funzionalità <i>view image-image</i> | 21 |
| 3. Stato dell'arte: recensione di siti e strumenti che permettono di visualizzare manoscritti | 23 |
| 3.1 <i>E-codices - Virtual Manuscript Library of Switzerland</i> | 23 |
| 3.2 <i>The Shelley-Goldwin Archive</i> | 25 |
| 3.3 <i>Vincent van Gogh - The Letters</i> | 28 |
| 3.4 <i>Emily Dickinson Archive</i> | 30 |
| 3.5 <i>Mark Twain Project Online</i> | 32 |
| 4. Progettazione e implementazione | 35 |
| 4.1 Funzionalità desiderate | 35 |

| | |
|--------------------------------------|-----------|
| 4.2 Angular | 36 |
| 4.3 Progettazione interfaccia utente | 38 |
| 4.4 Implementazione | 41 |
| 4.4.1 Modello dati | 41 |
| 4.4.2 Scrittura dei parser | 43 |
| 4.4.3 Implementazione dei componenti | 44 |
| Conclusioni | 51 |
| Bibliografia | 52 |
| Sitografia | 54 |

Introduzione

Lo sviluppo di nuove funzionalità per il confronto di manoscritti diversi all'interno del progetto EVT 3 nasce dall'esigenza di recuperare sia una delle *feature parity* con EVT 2, ovvero la visualizzazione della *view* per la descrizione del manoscritto (*Manuscript description*), sia per l'esigenza di rimediare alla mancanza nelle versioni attuali di EVT (EVT 1.3 e EVT 2) di viste che permettono il confronto tra descrizioni multiple e il confronto tra immagini diverse. Entrambe le versioni esistenti di EVT, infatti, permettono la visualizzazione di una trascrizione diplomatica con a fianco le scansioni del manoscritto, ma prevedono la presenza di un unico manoscritto e sono quindi limitate alla classica vista *image-text*, che non permette di confrontare due immagini diverse o due versioni diverse della stessa immagine. Si è reso quindi indispensabile sviluppare una vista *image-image*, oltre che *text-text* per il caso in cui ci sono le trascrizioni di testimoni diversi da confrontare, in modo da rendere più semplice i confronti tra diverse versioni dello stesso manoscritto (ad esempio nel caso in cui ci sia un secondo set di immagini derivate da restauro virtuale) e tra manoscritti diversi (testimoni separati di una stessa opera, o anche tra originale e copia successiva), e permettere quindi una migliore consultazione di questo aspetto di una edizione digitale.

All'interno di questa tesi l'obiettivo è quindi quello di continuare lo sviluppo di EVT 3, integrando nuove funzionalità utili allo studio e al confronto di edizioni digitali. Nel presente elaborato sono state esposte le fasi per la realizzazione di tale obiettivo.

Nel capitolo 1, si trova esposta una sintesi della storia dei facsimili e la loro diffusione nel tempo. In particolare, viene descritto come le opere che precedono l'invenzione della stampa sono state conservate fino a giungere a noi attraverso copie manuali, fotografiche e, infine, con copie digitali, evidenziando l'importanza che tali documenti hanno avuto, sia come mezzo per la creazione di edizioni affidabili che come oggetti meritevoli di studio in sé.

Nel capitolo 2, viene invece introdotto il progetto digitale *Edition Visualization Technology*, andando a definire le sue origini, i suoi cambiamenti nel tempo e le modalità di funzionamento della versione attuale. Dopodiché vengono

descritte le nuove funzionalità che verranno implementate all'interno del progetto di tesi.

Nel capitolo 3, viene effettuata un'indagine di studio dello stato dell'arte dei siti e degli strumenti per la visualizzazione di edizioni digitali, al fine di comprendere al meglio il contesto in cui il progetto EVT si colloca.

Infine, nel capitolo 4, vengono illustrate le fasi di progettazione e di implementazioni delle nuove funzionalità, illustrando le decisioni prese in merito alla definizione dell'interfaccia utente, una descrizione del *framework* Angular e la creazione del software vero e proprio, dalla creazione dei modelli dati fino alla implementazione della visualizzazione dei dati sulla piattaforma.

1. Il facsimile digitale: diffusione e vantaggi

1.1 Il lavoro degli amanuensi

Prima dell'invenzione della stampa il compito di trascrizione e conservazione delle opere letterarie (e non) era affidato principalmente ai monaci amanuensi che preservavano la cultura manoscritta, ma anche a botteghe artigiane e, a partire dal basso Medioevo, alle università.

Il loro minuzioso lavoro ha permesso ad opere vecchie centinaia di anni di giungere a noi e di preservare buona parte della cultura classica e medievale.

Nonostante queste figure fossero legate all'ambiente ecclesiastico, fortunatamente non hanno rivolto i propri sforzi solo verso la conservazione di opere sacre, ma si sono dedicate anche a testi dell'antichità greca e latina, inclusi anche molti libri profani.

La parola "amanuense" deriva dal latino *servus a manu*, letteralmente traducibile come "schiavo che ha l'incarico di copiare a mano", ed era colui che all'interno del monastero svolgeva l'incarico di copiatura scrupolosa dei testi.

L'editto di Costantino del 313 d.C., che proclamava la libertà di culto per i cristiani, diede origine a questa nuova figura. La principale conseguenza dell'atto, infatti, fu l'aumento della richiesta di libri per una maggiore volontà di diffondere la dottrina cristiana, e dato che la stampa non era stata ancora inventata, tali testi dovevano essere scritti a mano.

I monasteri divennero quindi centro di trasmissione e conservazione culturale: spesso vennero costruite delle vere e proprie biblioteche che riuscirono a preservare opere letterarie e altri documenti preziosi nel tempo salvandole da invasioni, saccheggi ed eventi naturali. Il lavoro svolto al loro interno è stato di fondamentale importanza storico-culturale in quanto ha permesso a testi antichi di sopravvivere negli anni giungendo fino a noi, rendendoli ancora oggi reperibili e facilmente consultabili.

Nonostante l'eccellente operosità dei monaci amanuensi si deve tener conto di alcuni aspetti chiave, in particolare il fatto che il lavoro venisse svolto a mano da singoli individui.

Il primo aspetto riguarda la scelta dei testi da copiare: essa veniva effettuata secondo rigorosi standard in linea con il pensiero ecclesiastico del tempo.

Il secondo invece è l'approccio critico ai testi, praticamente assente nei monaci amanuensi, che ha portato a numerosi errori nelle riproduzioni. Tali errori possono essere di vari tipi, più o meno involontari: errori di distrazione, di ortografia (spesso a causa della diversa pronuncia del dialetto del copista rispetto alla lingua originale del testo), omissioni, trasposizioni, errori di interpolazione (capitava quando un amanuense non si limitava a copiare, ma correggeva quelli che a suo parere erano degli errori e a volte aggiungeva anche brani di altra provenienza), incomprendimento del testo e così via.

Il terzo riguarda la fonte dei documenti: spesso il lavoro di copiatura veniva svolto non sull'originale, ma su manoscritti già trascritti più volte che potevano presentare gli errori appena citati, andando a creare un risultato finale sempre più incorretto e infedele, alterando totalmente l'opera originale.

1.2 Le discipline che studiano la scrittura

I manoscritti, medievali e non, sono oggetti meritevoli di studio in sé per varie discipline specialistiche: la paleografia e la filologia.

La paleografia (dal latino *palaeographia*) studia la storia della scrittura, in particolare di quella manuale, senza quindi l'uso di mezzi meccanici. Questa disciplina ne analizza il sistema e le tecniche, la loro evoluzione nel tempo, il processo di produzione, l'aspetto grafico dell'oggetto fisico e il supporto sul quale è stato creato (carta, pergamena, papiro). Approfondisce inoltre il contesto socioculturale in cui si colloca cercando di dare una datazione e una locazione all'opera. Ogni scrittura ha la sua paleografia specifica, quindi esisterà per esempio la paleografia latina, la paleografia greca, araba e così via.

La filologia (dal latino *philologiā*, dal greco *φιλολογία*, ovvero «amore della parola») è la disciplina che si occupa della cura del testo attraverso la critica testuale, ricostruendo e trasmettendo l'opera nella forma voluta all'origine dall'autore.

Il termine filologia, tuttavia, racchiude vari significati, una definizione relativa a un ambito applicativo generico ci viene proposta da Paolo Chiesa:

Ambito degli studi umanistici che mira più generalmente a un'esatta comprensione dei testi letterari, e anche non letterari, prodotti in una determinata epoca (dal punto di vista linguistico, metrico, stilistico, ecc.)

Possiamo comparare il lavoro del filologo a quello di un restauratore di quadri, come dice Lucia Cesarini Martinelli in *La filologia*¹: entrambi, infatti, cercano di riparare ai danni del tempo riportando l'opera all'antico splendore.

Ovviamente il filologo per poter operare al meglio deve essere a conoscenza di tutte le informazioni fondamentali del testo: l'epoca in cui nasce l'opera, l'autore, la lingua utilizzata, il contesto socioculturale, le ideologie dell'epoca ecc.

Il suo lavoro, tuttavia, in quanto frutto di pensieri e ragionamenti soggettivi, potrebbe essere condizionato, in maniera anche del tutto involontaria, da quelli che sono i suoi ideali talvolta andando ad intaccare l'opera con giudizi personali.

La filologia, nel suo significato più stretto, coincide con il concetto di ecdotica o critica testuale. L'ecdotica (dal greco ἔκδοσις «pubblicazione») è una branca della filologia che si occupa di riportare alla forma originale (o quasi) un'opera letteraria in maniera scientificamente motivata, nella forma voluta dall'autore. Il metodo più utilizzato per ricostruire un testo critico è il metodo di Lachmann (o metodo stemmatico), dal nome del filologo tedesco Karl Lachmann. L'obiettivo di tale metodo è quello di consentire la pubblicazione di un'edizione critica affidabile di una determinata opera, riducendo al minimo ogni aspetto soggettivo. Il metodo si compone di più fasi:

- *Recensio*: valutazione dei testimoni diretti e indiretti che trasmettono il testo dell'opera. Si divide in una serie di operazioni:
 - Ricognizione dei testimoni: ricerca e analisi dei testimoni che costituiscono la tradizione di un'opera. Viene effettuata una ricerca bibliografica in modo da censire tutti i documenti esistenti. Può rivelarsi un'operazione lunga e complessa, soprattutto se l'opera è assai diffusa.

¹ Lucia Cesarini Martinelli 1984, p. 11-12.

- *Collatio*: confronto tra i vari testimoni dell'opera per individuare e registrare tutte le differenze. Si prende a modello un testo-base, scelto con criteri abbastanza arbitrari, e lo si confronta con gli altri testimoni al fine di registrare tutte le lezioni² nelle quali essi differiscono dall'esemplare di collazione.

È in questa fase di *recensio* che vengono effettuate due operazioni che semplificano il lavoro di ricostruzione del testo. La prima è la creazione dell'albero genealogico dei manoscritti (*stemma codicum*), che ricostruisce i rapporti esistenti fra i vari documenti. La seconda operazione è l'esclusione dallo studio dei testimoni ritenuti inutili in quanto derivati esclusivamente da un'altro testimone già conservato (operazione di *eliminatio codicum descriptorum*).

- *Constitutio textus*: costruzione meccanica del testo originale basandosi sui dati della *recensio*. I testimoni rimanenti dopo la fase di *eliminatio codicum descriptorum* sono quelli collocati nei livelli più alti dello *stemma codicum*, chiamati subarchetipi³. Se le lezioni dei subarchetipi concordano allora si è trovato l'originale o l'archetipo⁴, altrimenti si procede con l'operazione di *selectio*.
 - *Selectio*: bisogna distinguere tra tradizione multipartita (i subarchetipi sono tre o più) e tradizione bipartita (gli subarchetipi sono due). Nel primo caso si procede in modo automatico adottando la legge della maggioranza: se la maggioranza numerica degli archetipi riportano la stessa lezione allora sarà quella originale. Nel secondo caso non si agisce in maniera automatica, ma viene scelta la lezione di qualità più alta. Se le lezioni sono entrambi accettabili, allora vengono fatte delle

² Lezione: «dal latino *lectio*, indica ciascuna delle varianti testuali esistenti di un documento antico nei testimoni che lo hanno tramandato», Wikipedia, Lezione (filologia), [https://it.wikipedia.org/wiki/Lezione_\(filologia\)](https://it.wikipedia.org/wiki/Lezione_(filologia)).

³ Subarchetipi: «primi discendenti dell'archetipo», Paolo Chiesa 2002, p. 81.

⁴ Archetipo: «indica il più antico esemplare, distinto dall'originale, da cui discendono tutti i testimoni superstiti di un testo», Wikipedia, Archetipo (filologia), [https://it.wikipedia.org/wiki/Archetipo_\(filologia\)](https://it.wikipedia.org/wiki/Archetipo_(filologia)).

congetture sulla base di determinati criteri: *lectio difficilior*⁵, *usus scribendi*⁶, *loci paralleli*⁷ o comportamento dei subarchetipi⁸.

Una volta trovato l'archetipo, si procede con *l'emendatio*:

- *Emendatio*: correzione del testo mediante congettura. E' sicuramente la fase più creativa della critica testuale: il filologo cerca di ripristinare il testo in base alle proprie conoscenze dell'autore, della lingua, dell'opera, e in base alle proprie intuizioni.

1.3 Il facsimile fotografico

1.3.1 L'importanza dei manoscritti

Per poter ricostruire l'integrità di un testo secondo le volontà dell'autore, sono di importanza fondamentale i manoscritti e la comparazione dei testimoni che conservano il testo attraverso l'operazione di collazione. Sono infatti esigui i casi in cui si è in possesso del documento originale, mentre ben più numerosi quelli in cui sono presenti molteplici documenti (testimoni) che trasmettono il testo. Nel secondo caso i vari testimoni vengono confrontati parola per parola al fine di comprendere il grado di parentela fra i vari documenti (rappresentato dallo *stemma codicum*) e di ricostruire il testo originale (applicando quando possibile la legge di maggioranza); questo è realizzabile mediante l'analisi della natura degli errori presenti, in particolare di quelli poligenetici (fatti dai copisti indipendentemente l'uno dall'altro) e di quelli monogenetici o errori guida (errori commessi in una determinata condizione, come per esempio copiandolo dalla stessa fonte). Attraverso il confronto sarà quindi possibile andare a evidenziare quali sono effettivamente le problematiche⁹ riscontrate all'interno delle varie trascrizioni, che devono quindi

⁵ *Lectio difficilior*: «fra due lezioni concorrenti, viene considerata più probabilmente originaria quella che appare più difficile rispetto all'altra», Paolo Chiesa 2002, p. 90.

⁶ *Usus scribendi*: «fra due varianti concorrenti, quella originaria sarà quella che risponde meglio alle abitudini dello scrittore», Paolo Chiesa 2002, p. 90.

⁷ *Loci paralleli*: confrontare il passo in questione «con passi di altri autori o di altre opere dove ricorrano espressioni simili a quella indagata», Paolo Chiesa 2002, p. 92.

⁸ Comportamento dei subarchetipi: fra due subarchetipi α e β è più probabile che quello che riporta innovazioni sia l'archetipo α in quanto redatto da un copista più incline a modificare il testo, Paolo Chiesa 2002, p. 93.

⁹ Come quelli già descritti all'interno della sezione 1.1.

essere risolte e quali invece possono essere considerate delle varianti, ovvero dei testi alternativi che si allineano con la grammatica e il senso del resto del testo. Il risultato finale di tali operazioni sarà il testo critico del documento, che poi sarà base di successivi studi.

1.3.2 Difficoltà di analisi

Una delle più grandi difficoltà che può riscontrare il filologo è quello di riuscire ad analizzare i manoscritti originali: la loro fragilità e unicità rende a volte problematica la loro consultazione. Da considerare fattori negativi anche la chiusura e l'inaccessibilità a biblioteche e archivi, il divieto di consultazione o comunque la difficoltà di accesso a determinati testi particolarmente preziosi, inventari introvabili o incompleti, fonti rovinate da operazioni di legatura o di restauro non eseguite correttamente oppure difficoltà nel raggiungere l'ubicazione della fonte in quanto situato, per esempio, in un'altro continente. Altri problemi possono essere riscontrabili a causa della mancanza di fondi per la cultura che rende quindi impossibile avere una gestione efficace del patrimonio librario, problema diffuso soprattutto nei piccoli-medi centri o comuni.

Si riscontrano in alcune circostanze dei problemi legati al documento stesso, come per esempio la sua non integrità a causa di danni e deterioramenti dovuti al tempo o alla cattiva conservazione (strappi, buchi, inchiostro sbiadito ecc.). In questi casi è molto più difficoltoso, se non impossibile, per uno studioso riuscire a recuperare le informazioni mancanti. C'è infine la possibilità che un determinato manoscritto sia particolarmente sensibile ad alcuni ambienti e luci, necessitando quindi l'uso di specifici strumenti e cure che non sempre possono essere reperibili e utilizzabili.

1.3.3 Diffusione del facsimile fotografico

Un'importante rivoluzione del XIX secolo è stata l'invenzione della fotografia. Nel 1839 la nascita del dagherrotipo di Louis Mandé Daguerre ha

permesso la riproduzione di una natura morta di grande dimensioni. Questa scoperta è stata presentata all'Accademia di Francia e diffusa in tutto il mondo, accompagnandola con un manuale descrittivo delle istruzioni e dei processi tecnici. Daguerre però non è stato il pioniere della fotografia come la intendiamo modernamente, bensì lo è stato William Henry Fox Talbot, un fisico inglese, che nel 1833 ha condotto alcuni esperimenti che presero il nome di "calotipia", usati nell'estate dello stesso anno per produrre delle piccole immagini di qualche centimetro. Le prime fotografie sono state accolte con stupore dal pubblico che non ne comprendeva alla perfezione le tecniche di creazione, rimanendo estasiato dalla definizione dei dettagli immortalati. La nuova tecnica si è affiancata, e in molti casi ha sostituito, vecchie usanze come la rappresentazione dei paesaggi mediante la pittura, in quanto più veloce e meno dispendiosa. La diffusione e l'applicazione della fotografia in tutti gli ambiti di studio è stata esponenziale, soprattutto per quanto riguarda la creazione di facsimili.

Grazie, infatti, a tecniche come la litografia¹⁰, in un primo momento, ed alla collotipia¹¹, in un secondo, nacque la possibilità di creare documenti che riproducevano con un grado di fedeltà mediamente buono l'opera originale, eliminando tutti i possibili e frequenti errori dovuti alla copiatura manuale. Dalla datazione dei documenti è possibile stabilire che tali tecniche sono state utilizzate per la creazione di copie fino alla metà del XX secolo. La loro diffusione ha avuto effetti positivi come una migliore conservazione e una maggiore accessibilità a libri e manoscritti antichi o raramente trovabili. Tuttavia, queste tecniche comportano un costo molto elevato e le stampe si presentavano solo in toni di grigio, perdendo i colori del documento originale.

Al termine della Seconda guerra mondiale sono state importate dagli Stati Uniti in Europa delle moderne tecniche di stampa che permettevano anche l'uso dei colori: questo ebbe un forte impatto sulla cultura e sugli sviluppi della creazione dei facsimili. Negli anni '60 una nuova metodologia proveniente sempre dagli USA

¹⁰ Litografia: (dal greco "scrivere su pietra") tecnica di stampa basata sull'incompatibilità tra l'inchiostro grasso e l'acqua. Il disegno viene tracciato direttamente su una lastra di pietra piatta o su un materiale simile con una matita grassa, Bregeaud e Senefelder, 1826, p. 1-14.

¹¹ Collotipia: tecnica di riproduzione fotomeccanica delle immagini. Il procedimento si basa sulle proprietà della gelatina e del bicromato di potassio che vengono sparsi su una lastra di metallo o di vetro e una volta esposta alla luce si crea una reazione chimica, Denis Defibaugh 1997, p. 2-3.

trova la sua fortuna: la stampa offset. Essa consiste in una tecnica di stampa indiretta dove il disegno viene trasferito da una lastra matrice al foglio, o un altro materiale, attraverso dei cilindri di gomma.

Il motivo dell'ampia diffusione di queste tecniche di riproduzione delle immagini è legato principalmente al numero di persone raggiungibili. Grazie alla produzione di un maggior numero di testi ad un costo inferiore, infatti, molti più interessati potevano accedere alle informazioni più facilmente. Con tali metodologie le copie prodotte di un documento erano diventate un numero ragguardevole, maggiore che nei decenni precedenti, più facili e veloci da realizzare, a differenza delle coperture effettuate manualmente.

1.3.4 Il problema dei costi

I facsimili fotografici, in particolari quelli a colori realizzati con le tecniche che si sono diffuse nella seconda metà del '900, presentano alcuni svantaggi. Il più importante fra essi è il costo della loro realizzazione. Analizzando in particolare la stampa offset si nota che essa risulta essere estremamente costosa e quindi poco conveniente. I fattori che incidono sull'alto costo sono:

- i costi di installazione: necessita di strumenti specifici e dispendiosi come pellicole e matrici;
- il costo è inversamente proporzionale al numero di copie stampate: non è adatto quindi per le piccole tirature;
- i costi di avviamento: la preparazione della matrice comporta un costo elevato e la lastra una volta pronta non è più facilmente modificabile;
- i costi in termini di tempo: tempi lunghi per la preparazione;

Al giorno d'oggi è preferibile quindi utilizzare altri mezzi più economici e veloci per permettere la conservazione e la diffusione dei testi antichi e preziosi: grazie allo sviluppo tecnologico è possibile, infatti, creare facsimili digitali.

1.4 Il facsimile digitale

Il facsimile digitale è una riproduzione fedele di un manoscritto o di una stampa ottenuta tramite un'operazione di scansione digitale (ad esempio il *Caxton's Canterbury Tales: The Prologue*, British Library¹² o il *Frankenstein*, The Shelley-Godwin Archive¹³). Si tratta pertanto di una replica del documento sotto forma di immagine digitale: la sua qualità varia a seconda di fattori come la risoluzione e il formato del file. Un facsimile digitale nel caso più semplice consiste in una raccolta di immagini con annessi dei metadati utili per poter identificare e descrivere la fonte e il suo materiale. Lo sviluppo naturale del facsimile digitale è quello di affiancare l'immagine alla sua trascrizione con la possibilità di stabilire un collegamento fra i due, rendendo così possibile, anche se in maniera indiretta, effettuare ricerche ed elaborazioni all'interno dell'immagine.

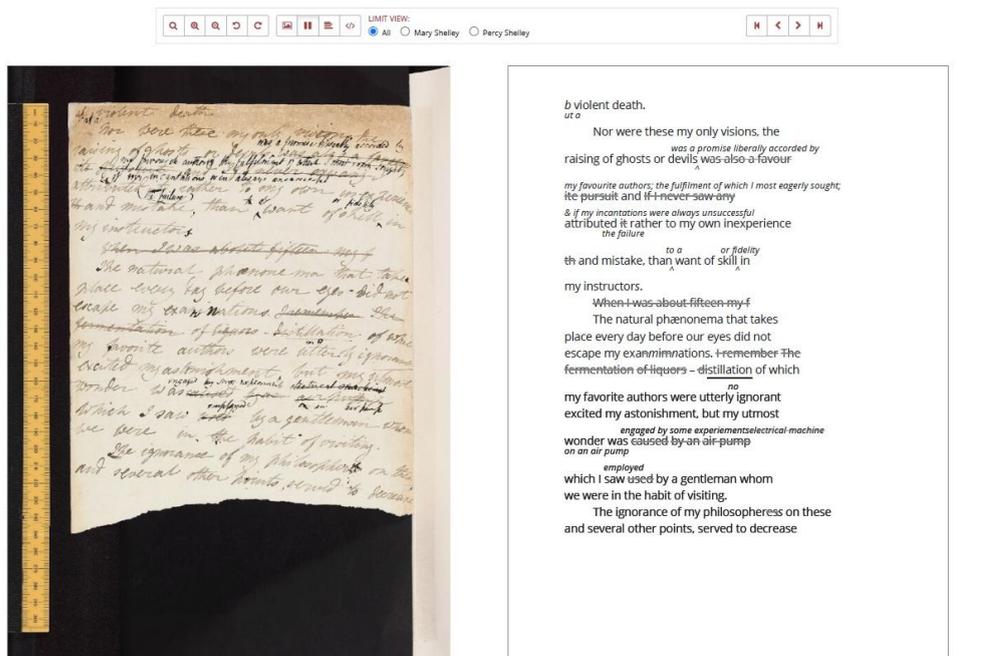


Figura 1: Shelley, M. W. *Frankenstein, Volume I*, in The Shelley-Godwin Archive.

Al giorno d'oggi l'uso dei facsimili digitali è in continua espansione e si sono resi indispensabili per lo studio e la ricerca. Il tutto è merito di una serie di vantaggi:

- economico: il prezzo per la sua creazione è relativamente basso;

¹² *Caxton's Canterbury Tales: The Prologue*: <https://www.bl.uk/treasures/caxton/record.asp?LHPage=1v&LHvol=&LHCopy=1&RHPage=2r&RHvol=&RHCopy=1&disp=s&Linked=0#DispTop>.

¹³ *Frankenstein*: <http://shelleygodwinarchive.org/contents/frankenstein/>.

- facilità di pubblicazione sul web: tali risorse possono essere condivise in rete e rese accessibili a un pubblico estremamente ampio;
- facilità di accesso: i documenti digitali sono facilmente accessibili a chiunque sia interessato con l'unico requisito di essere in possesso di un dispositivo elettronico dotato di connessione, eliminando così tutte le problematiche legate alla sua ubicazione e alle difficoltà di consultazione;
- facilità di conservazione e salvaguardia dell'opera originale: i manoscritti sono testi rari e fragili, l'uso della digitalizzazione permette di conservare e studiare i documenti evitando ulteriori deterioramenti del documento;
- maggiore leggibilità del documento: è possibile confrontare l'immagine direttamente con la sua trascrizione o codifica oppure con altri documenti, permette di poter osservare anche i più piccoli dei dettagli, semplicemente zoomando l'immagine, il tutto senza l'irreversibile rischio di rovinare l'opera originale.
- funzionalità aggiuntive: spesso i facsimili digitali sono integrati con strumenti in grado di fare ricerche ed elaborazioni all'interno dell'immagine;
- possibilità di creare edizioni diplomatiche¹⁴ con immagini a fronte.

1.4.1 Strumenti utili per lo studio

Nel processo di analisi di un documento, manoscritto originale o trascrizioni successive, l'attenzione deve essere focalizzata sull'identificazione degli errori scribali presenti. È fondamentale la loro interpretazione che è possibile solamente mediante la creazione di strumenti e di ambienti appositi. Per eseguire al meglio una operazione di ricognizione e di studio, il soggetto deve avere a disposizione dei *tool* che lo aiutino a sviluppare nelle migliori condizioni possibili il suo lavoro. Tra questi, di fondamentale importanza, troviamo la possibilità di ingrandire le immagini mediante lo *zoom*, lo spostamento all'interno di esse con il puntatore ed il fermo-immagine sulle aree in analisi. Oltre ad essi è importante citare la moderna

¹⁴ Edizione diplomatica: «edizione che trascrive il testo con criteri di fedeltà diplomatica, senza che abbia luogo, da parte dell'editore, il benché minimo intervento, né per sanare lacune ed errori anche manifesti, né per regolarizzare la divisione delle parole e alterare l'usus grafico-fonetico, compresi particolari anche minimi come segni d'interpunzione, capoversi, e simili», https://www.filologiasarda.eu//didattica/glossario_manca.pdf.

tecnica di restauro virtuale che permette di recuperare l'integrità dell'opera e le informazioni del documento perse col tempo. Grazie ad essa il soggetto che analizza il documento è in grado di avere un'idea il più possibile veritiera e affidabile del manoscritto originale di cui sta effettuando la trascrizione.

Il lettore è quindi in grado, con gli strumenti forniti, di appurare come è stato svolto il lavoro del filologo seguendo il ragionamento che lo ha condotto all'analisi finale. Per rendere possibile ciò è fondamentale che l'edizione digitale fornita all'utente sia il più possibile dettagliata e commentata. Inoltre, mettendo a disposizione le fonti utilizzate, il materiale di studio e le metodologie di analisi, si avrà la possibilità di condividere con il pubblico tutti i processi cognitivi del filologo.

2. EVT e il facsimile digitale

2.1 EVT: *Edition Visualization Technology*

EVT¹⁵ è un software utilizzato per la pubblicazione e la consultazione di edizioni digitali, sviluppato da un gruppo di ricerca dell'Università di Pisa. Esso prende vita circa nel 2003 nell'ambito del progetto didattico *Digital Vercelli Book*¹⁶, data in cui il manoscritto viene completamente scansionato. L'idea nasce dal fatto che all'epoca (2010), dopo aver codificato i testi e l'interesse si è rivolto alla ricerca di un software per la visualizzazione adatto, non fosse disponibile una piattaforma in grado di consentire la navigazione e la consultazione online del manoscritto: gli strumenti disponibili al tempo infatti erano o troppo complessi da installare e usare, o software di tipo proprietari, oppure legati a un determinato programma/sistema operativo o a un progetto specifico. Dato che tali software presentavano molteplici svantaggi, venne deciso di crearne uno appositamente per il progetto. Esso avrebbe dovuto seguire alcuni principi:

- essere distribuito con licenza *open source*;
- basarsi su tecnologie web standard come HTML¹⁷, CSS¹⁸, JavaScript¹⁹, jQuery²⁰, in modo da renderlo *multi-platform* e indipendente da specifici sistemi operativi;
- fare uso di un'interfaccia interattiva e *user friendly*;
- essere flessibile in modo da consentire lo sviluppo di nuovi progetti di edizioni digitali, soddisfacendo molteplici casi d'uso.

Il risultato ottenuto è un software in grado di ricevere in input uno o più documenti XML²¹ nel formato TEI²² P5²³ e trasformarli in un'applicazione web

¹⁵ The Digital Vercelli Book: <http://vbd.humnet.unipi.it>.

¹⁶ Edition Visualization Technology: <http://evt.labcd.unipi.it/>.

¹⁷ HTML (*HyperText Markup Language*), Sito ufficiale W3C, <https://www.w3.org/TR/html52/>.

¹⁸ CSS (*Cascading Style Sheets*), Sito ufficiale W3C, <https://www.w3.org/Style/CSS/Overview.en.html>.

¹⁹ Javascript, <https://www.javascript.com/>.

²⁰ JQuery, <https://jquery.com/>.

²¹ XML (*eXtensible Markup Language*), Sito ufficiale W3C, <https://www.w3.org/XML/>.

²² TEI (*Text Encoding Initiative*), <https://tei-c.org/>.

²³ TEI P5 Guidelines, <https://tei-c.org/guidelines/p5/>.

client-only. Grazie alla sua flessibilità, l'applicazione trovò diffusione e apprezzamento nella comunità di ricerca di tutto il mondo in quanto strumento utile alla pubblicazione del loro lavoro, a prescindere dal fatto che avessero buone competenze informatiche.

Ad oggi sono tre le versioni esistenti di EVT:

- EVT 1: offre il supporto per edizioni diplomatiche/interpretative e l'uso di un testimone unico (*codex unicus*). Si basa su una catena di trasformazioni XSLT²⁴ 2.0 per realizzare un'applicazione web. La versione finale EVT 1.3 è stata pubblicata a fine 2019;
- EVT 2: pubblicato nel giugno 2020, offre il supporto per edizioni critiche e la raccolta di testimoni utilizzando Angular JS²⁵.
- EVT 3: basato su Angular 9. Il rilascio della versione Alpha è previsto nel corso del 2021.

2.2 EVT 1: *view image-text, bookreader*

Nella prima versione di EVT è prevista la creazione di edizioni con due possibili livelli: diplomatico e diplomatico-interpretativo. Il layout della pagina web è molto semplice: di default la modalità di visualizzazione è quella *image-text* dove sul lato sinistro della schermata è situata l'immagine, mentre sul lato destro c'è la trascrizione corrispondente del manoscritto. È possibile però cambiare vista attraverso la barra degli strumenti posta in alto a destra, dove sono visualizzate le icone di tutte e tre le possibili modalità di visualizzazione:

- *view image-text*: vista predefinita dove vengono visualizzati sia la scansione che il testo dell'edizione. Il livello di edizione può essere cambiato attraverso un menù a tendina nella barra degli strumenti posta sopra il testo;
- *view text-text*: vista utile per poter confrontare diversi livelli di edizione;

²⁴ XSLT (*Extensible Stylesheet Language Transformations*), Sito ufficiale W3C, <https://www.w3.org/TR/xslt-30/>.

²⁵ Angular JS, Superheroic JavaScript MVW Framework, <https://angularjs.org>.

- *bookreader*: vista dove l'immagine si espande e permette di poter sfogliare le pagine del manoscritto, assumendo una modalità di navigazione a doppia facciata (viene mostrato il verso del folio e il recto del successivo).

L'applicazione inoltre permette di analizzare in maniera più approfondita i dettagli delle varie immagini attraverso strumenti come la lente di ingrandimento e lo *zoom*.



Figura 2: vista *bookreader* in EVT 1.3.

2.3 EVT 2: *view image-text*

La seconda versione di EVT, sviluppata dal 2016 al 2020, è stata implementata anzitutto per risolvere dei problemi e dei difetti che si erano riscontrati nella prima versione tra cui: la complessità del codice XSLT 2.0, diventato difficile da modificare, e la difficoltà nel gestire il layout dato il numero crescente di *widget* presenti. Queste problematiche hanno portato al bisogno di un rinnovamento verso una nuova infrastruttura più flessibile. Viene quindi abbandonato l'XSLT a favore di un'architettura MVC²⁶ (*Model View Controller*) e del framework Angular JS in grado di interpretare direttamente i documenti XML. Lo stile invece viene affidato a fogli di stile CSS, grazie ai quali diventa più facile la personalizzazione del sito.

²⁶ MVC: «pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti e in applicazioni web, in grado di separare la logica di presentazione dei dati dalla logica di business», MVC – Wikipedia <https://it.wikipedia.org/wiki/Model-view-controller>.

EVT 2 possiede una nuova architettura basata appunto sul *design pattern* MVC, che permette una migliore flessibilità nella gestione dei dati. Inoltre non esiste più una fase di trasformazione come nella versione precedente. Per il suo funzionamento, il documento XML viene caricato e parsato direttamente da EVT: i dati in input vengono trasformati in un set di dati JSON²⁷ che vengono poi trasformati in HTML per la visualizzazione finale.

Dal punto di vista della visualizzazione, notiamo alcuni cambiamenti rispetto alla versione precedente come:

- la scomparsa della modalità di vista *Bookreader*, che permetteva di sfogliare le pagine del manoscritto a doppia faccia;
- l'aggiunta della vista *Reading Text* che consente di visualizzare solamente il testo dell'edizione;
- l'aggiunta della vista *Collation* che permette la lettura in parallelo tra uno o più testimoni e il testo critico;
- l'aggiunta della vista *Multiple recensions*, che permette il confronto tra versioni (o recensioni) multiple del testo.

Rimane invariato invece la tradizionale vista *image-text* (facsimile digitale sulla sinistra e trascrizione sulla destra dello schermo) e la vista *text-text*.

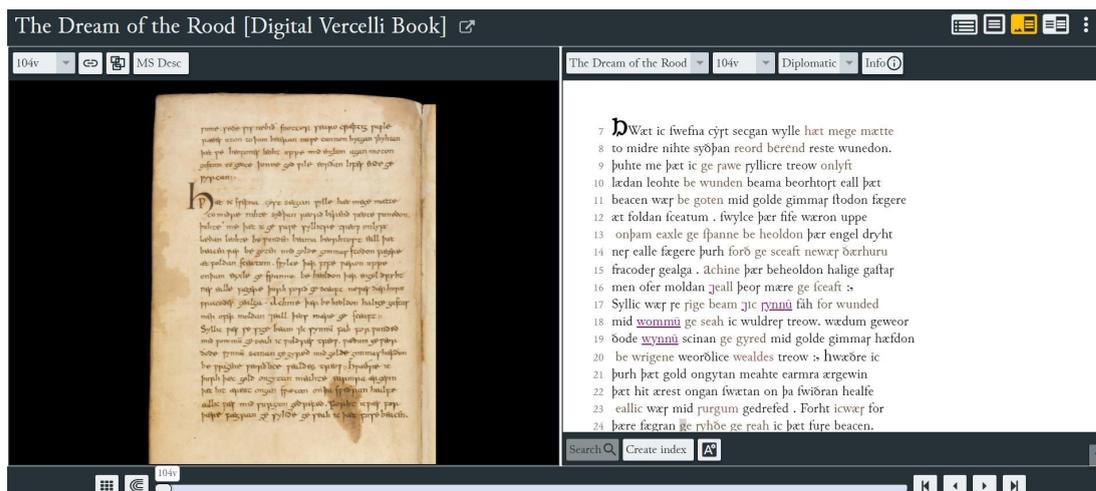


Figura 3: vista *image-text* in EVT 2.

²⁷ JSON, <https://www.json.org/json-en.html>.

2.4 EVT 3

Novità importante apportata con lo sviluppo della nuova versione di EVT è sicuramente il passaggio da Angular JS ad Angular 9 per il bisogno di maggiore flessibilità e facilità di manutenzione nel lungo periodo. Il cambiamento di *framework*, inoltre, è stato “imposto” da Google stesso: la nuova versione è stata talmente migliorata da risultare incompatibile con quella vecchia, inoltre il supporto per AngularJS terminerà a luglio 2021.

L’obiettivo principale per la versione Alpha di EVT 3 è il raggiungimento delle *feature parity* con EVT 2 e il miglioramento degli aspetti legati in particolare all’edizione critica, all’edizione diplomatica con più livelli di edizioni, alle *named entities* e al motore di ricerca.

Il funzionamento di EVT 3, similmente alla versione precedente ma di più facile applicazione, è caratterizzato da un *parser* di input e da un *viewer* per l’output. I dati in input provengono da un documento nel formato XML TEI. Essi vengono poi parsati e salvati in un file intermedio nel formato JSON, che a sua volta viene elaborato e trasformato in un file HTML da poter visualizzare (fig. 4).

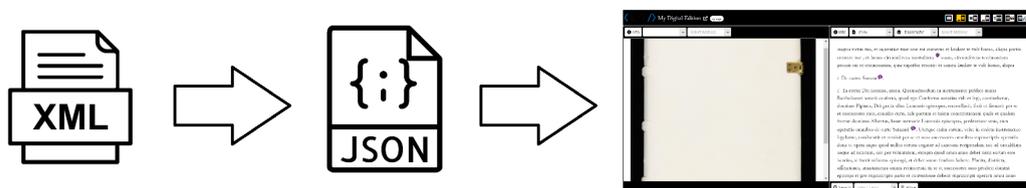


Figura 4: modalità di funzionamento di EVT 3.

L’architettura è di tipo *client-only*, ciò significa che non è necessario dover installare un software a supporto dell’applicazione sul server (che richiederebbe manutenzione costante), ma per pubblicare un’edizione digitale è sufficiente scompattare la cartella sul desktop e accedere attraverso il file `index.html` per poter avviare la navigazione. Questo approccio prevede una maggiore semplicità di condivisione sul web.

Tra le nuove funzionalità che verranno sviluppate è importante citare quelle legate alla modalità di visualizzazione e confronto dei dati: sarà infatti implementato

il supporto per descrizioni di manoscritti multiple, migliorando la già esistente vista *text-text*, e una nuova vista *image-image*, utile per il confronto tra facsimili diversi.

2.4.1 Estensione della funzionalità di <msDesc> multipli

La prima delle nuove funzionalità che si aggiungerà a EVT 3 è sicuramente lo sviluppo del supporto all'elemento TEI <msDesc> e la possibilità di poter confrontare <msDesc> multipli. Questo ultimo caso in particolare è una funzionalità del tutto nuova, che non si trova in nessuna delle precedenti versioni di EVT.

Tale risorsa si rivelerebbe essere assai utile in vari casi d'uso. È possibile infatti che all'interno di un file XML siano presenti più descrizioni del manoscritto, permettendo così al lettore la possibilità di scegliere quale tra essi visualizzare. Altri casi in cui si renderebbe utile una tale implementazione è per la possibilità di confrontare più manoscritti differenti con trascrizioni diverse dello medesimo testo, o il confronto tra <msDesc> di più testimoni.

2.4.2 Estensione della funzionalità *view image-image*

Ulteriore nuova funzionalità prevista per EVT 3 è la creazione di una nuova vista *image-image*. Essa sarà di fondamentale importanza per il confronto e l'analisi in parallelo di immagini diverse dello stesso manoscritto o di manoscritti diversi²⁸. I casi in cui è utile avere una comparazione di questo genere sono i casi in cui si analizzano:

- una stessa immagine ma con risoluzioni differenti;
- le immagini che testimoniano il prima e il dopo di un restauro del documento, sia virtuale sia fisico;

²⁸ Per il Vercelli Book, ad esempio, è disponibile una trascrizione completa del manoscritto effettuata da C. Maier nel 1833, la scansione di questo documento è già stata effettuata e sarebbe molto interessante poterla confrontare con il manoscritto originale perché Maier ha potuto leggere e trascrivere alcune parole che adesso non sono più leggibili.

- le scansioni multispettrali o con una illuminazione particolare;
- i facsimili dei vari testimoni di una stessa opera;
- il facsimile del manoscritto originale con quelli dei suoi testimoni.

3. Stato dell'arte: recensione di siti e strumenti che permettono di visualizzare manoscritti

Al giorno d'oggi le edizioni digitali sono sempre più diffuse proprio grazie alla loro flessibilità, alla loro funzione di conservazione e diffusione, e alla possibilità di sfruttare strumenti sofisticati e innovativi per l'analisi dei documenti. È interessante andare ad esaminare alcuni di questi progetti disponibili in rete per capirne il funzionamento e la gestione dell'interfaccia grafica, andando a studiare quello che è lo stato dell'arte attuale in questo ambito. Tale analisi non si ferma ai soli siti ospitanti facsimili digitali, ma si espande andando a studiare il funzionamento di edizioni e biblioteche digitali che mettono in relazione tra loro facsimili, descrizioni del manoscritto e la relativa trascrizione (diplomatica) o l'edizione dei testi ivi contenuti. Questo perché, in prospettiva con EVT, l'interesse è di non limitare la comparazione e il confronto tra testi e immagini, ma stabilire un vero e proprio rapporto tra tutte le componenti dell'edizione.

3.1 E-codices - Virtual Manuscript Library of Switzerland

La biblioteca digitale *e-codices*²⁹ è un progetto dell'università di Friburgo che nasce nel 2005 con l'obiettivo iniziale di digitalizzare 130 manoscritti della biblioteca dell'Abbazia di San Gallo. Grazie ai contributi e al sostegno di varie fondazioni, *e-codices* si è evoluto fino a diventare una biblioteca virtuale che ospita, e rende disponibili gratuitamente per ricercatori e studiosi, manoscritti svizzeri medievali e moderni in forma digitale. I metadati testuali relativi a ciascun manoscritto sono codificati nel formato XML secondo lo standard TEI P5, consentendo la possibilità di scambiare informazioni con altri progetti digitali e l'utilizzo di motori di ricerca che permettono di ricercare i testi in base a criteri come nome dell'autore, titolo dell'opera, incipit ecc.

²⁹ *E-codices*, <https://www.e-codices.unifr.ch/it>.

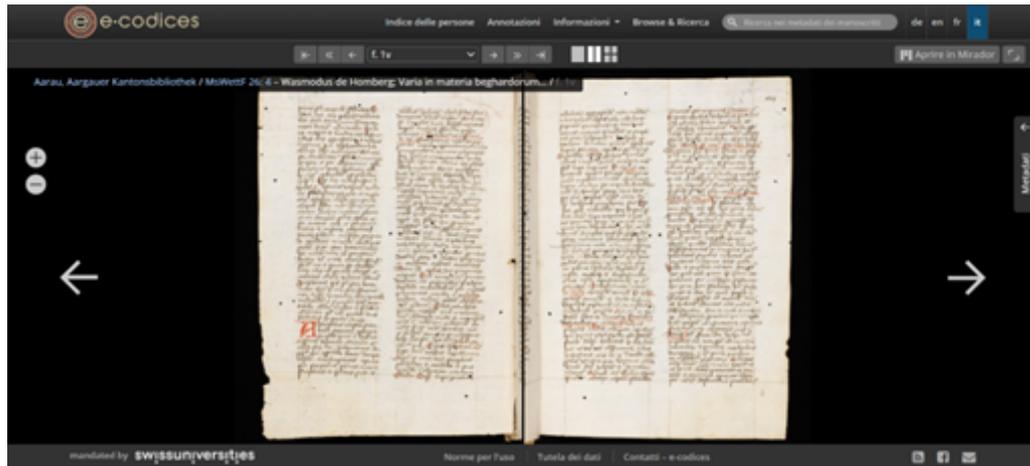


Figura 5: e-codices, *Single page view*.

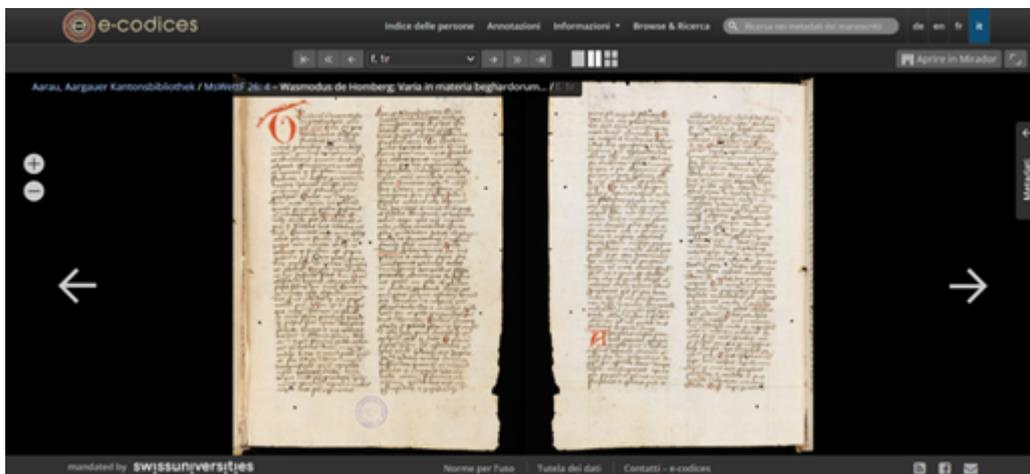


Figura 6: e-codices, *Double page view*.

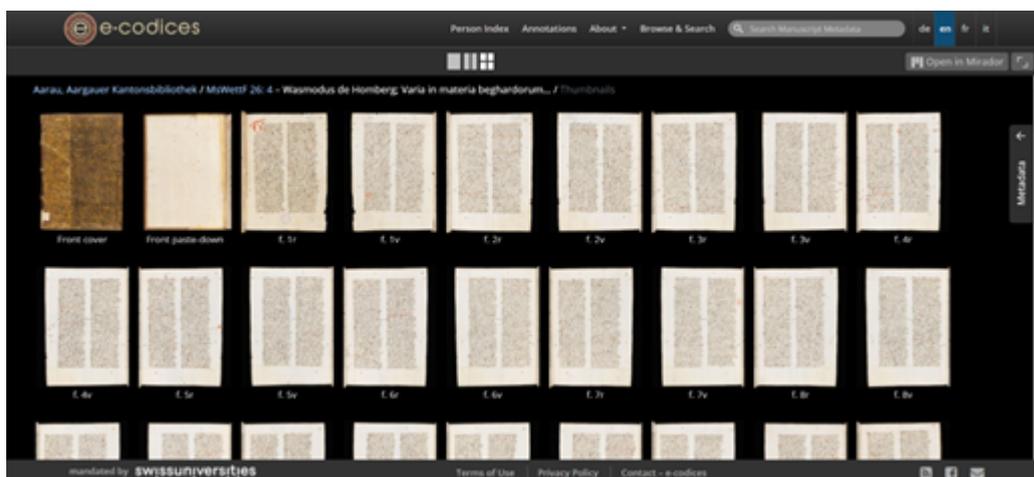


Figura 7: e-codices, *Thumbnails*.

Le possibili modalità di fruizione del contenuto del manoscritto sono tre: *Single page* (Fig. 5), in cui viene visualizzata una singola pagina del documento, *Double page* (Fig. 6), dove vengono visualizzate in parallelo la pagina corrente e la successiva, e *Thumbnails* (Fig. 7), in cui si visualizza l'elenco di tutte le miniature

del manoscritto. In tutte queste viste è sempre possibile usufruire di strumenti aggiuntivi come lo *zoom*, dei *tool* per capovolgere le immagini a piacimento, e frecce e menù a tendina che permettono di selezionare la pagina desiderata. Importante la presenza del bottone *Metadata* sul lato destro dello schermo: con un click è possibile visualizzare tutte le informazioni e le descrizioni riguardanti il manoscritto. C'è inoltre la possibilità di passare alla visione d'insieme (Fig. 8), dove facsimile e metadati sono affiancati uno accanto all'altro, permettendo una migliore consultazione del testo.

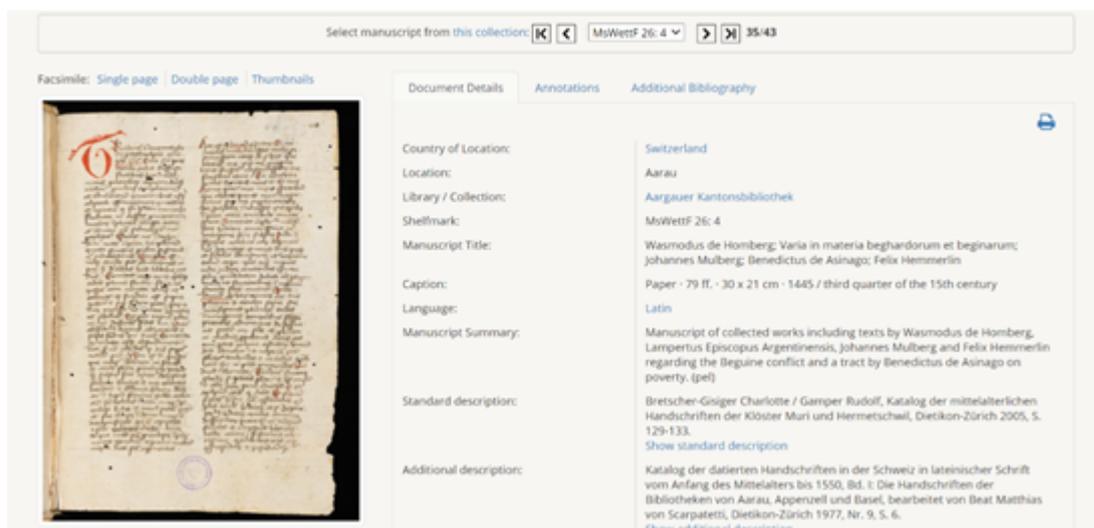


Figura 8: *e-codices*, visione d'insieme.

Questa modalità di edizione digitale presenta però un grosso svantaggio: riguarda solo le scansioni e i metadati, non esiste la possibilità di consultare la trascrizione dei manoscritti. Di conseguenza diventa difficoltosa la lettura e la comprensione del testo attraverso i soli facsimili, soprattutto per i documenti più antichi. Non essendo prevista una modalità di visualizzazione *image-text* che permetta una migliore consultazione dei documenti, inoltre, non esiste alcun tipo di collegamento testo-immagine.

3.2 *The Shelley-Goldwin Archive*

L'archivio *Shelley-Goldwin*³⁰ nasce e si sviluppa con l'obiettivo di pubblicare edizioni digitali delle opere di Percy Bysshe Shelley, Mary Wollstonecraft Shelley,

³⁰ *The Shelley-Godwin Archive*, <http://shelleygodwinarchive.org/>.

William Godwin e Mary Wollstonecraft. Il progetto è stato creato grazie alla collaborazione tra il *Maryland Institute for Technology in the Humanities* e le biblioteche *New York Public Library* (New York, USA), *Bodleian Library* (Oxford, Uk), *Huntington Library* (San Marino, USA), *Houghton Library* (Cambridge, USA) e il *Victoria and Albert Museum* (Londra, Uk).

Il modello dati utilizzato nella infrastruttura tecnica è lo *Shared Canvas*, ideato dall'università di Stanford, che facilita la descrizione e la presentazione di artefatti, mentre i testi sono stati codificati secondo il linguaggio XML TEI P5. Su Github³¹ sono presenti a libero accesso le varie trascrizioni dell'archivio, oltre che le applicazioni software e le librerie, il tutto disponibile con licenze aperte (*Apache 2.0* e *Creative Commons Attribution*).

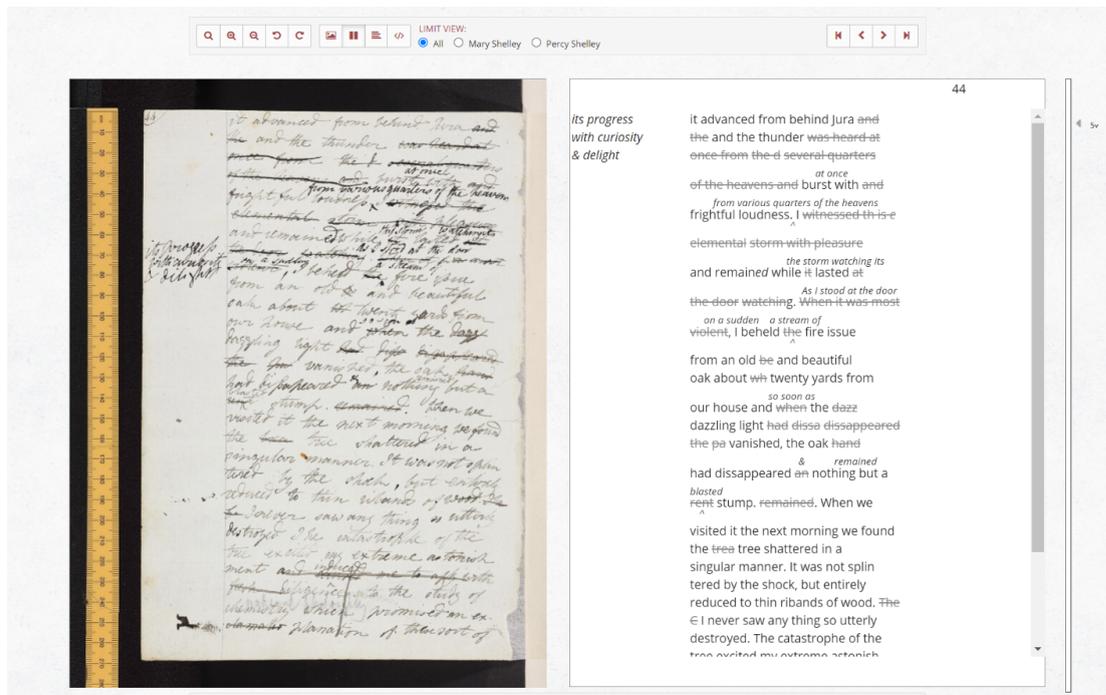


Figura 9: *The Shelley-Goldwin Archive, Standard View.*

³¹ <https://github.com/umdmith/sga>.

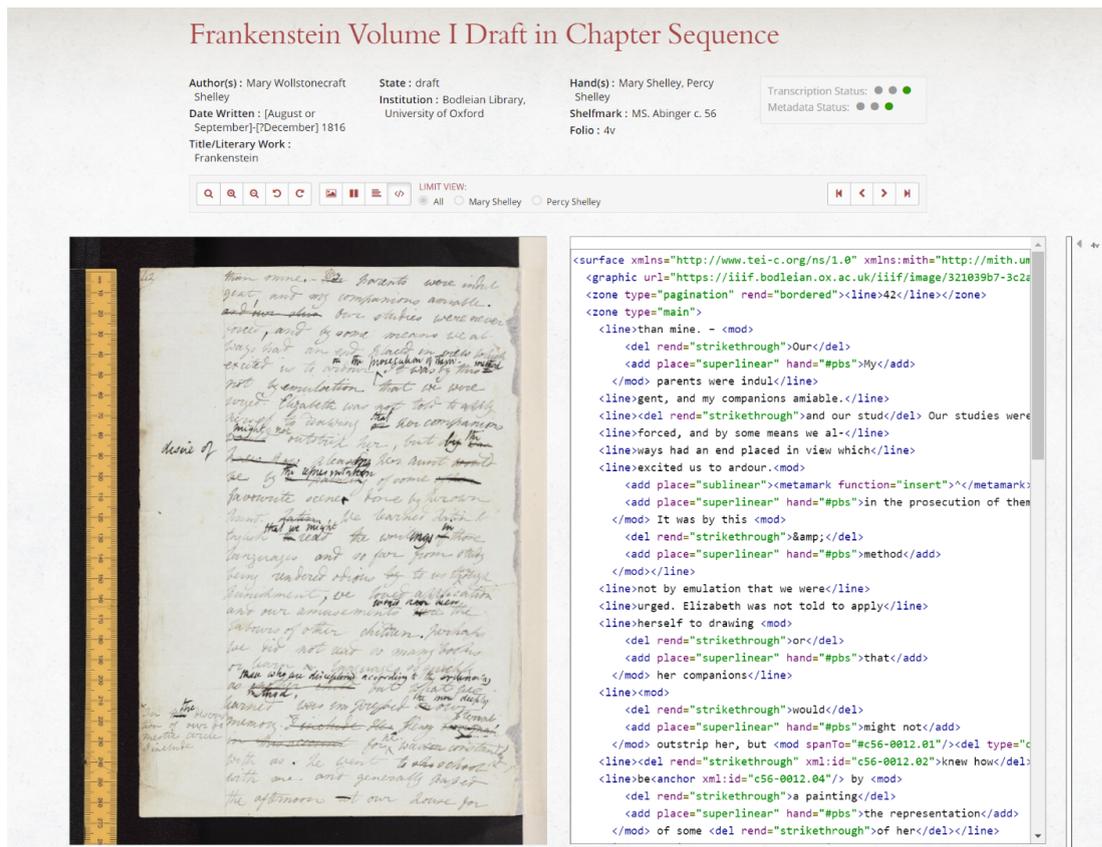


Figura 10: *The Shelley-Goldwin Archive, View TEI.*

Dal punto di vista dell'interfaccia utente, ogni manoscritto è accompagnato nella visualizzazione dal suo facsimile ad alta risoluzione, dai suoi metadati (autore/i, titolo, data ecc.), dalla trascrizione (se presente), dallo stato della trascrizione e dai pulsanti di controllo (come le frecce per cambiare pagina, lo *zoom* e la rotazione dell'immagine). La visualizzazione predefinita è la *Standard view* (Fig. 9), in cui si visualizza due sezioni in parallelo sullo schermo: quella di sinistra mostra il facsimile, quella di destra mostra la trascrizione diplomatica. Sono presenti altre tre viste, a cui si può accedere tramite le icone presenti nella barra degli strumenti: *View Image Only*, in cui si visualizza l'immagine del manoscritto a tutto schermo, *View Reading Text*, che contiene il testo dell'edizione, e *View Tei* (Fig. 10), in cui si visualizza la trascrizione XML TEI del documento. Una delle funzioni più interessanti è sicuramente quella del *Limit View* (Fig. 11), che permette all'utente finale di filtrare il testo visualizzando solamente quello scritto da una certa mano.



Figura 11: *The Shelley-Goldwin Archive, Limit View.*

In conclusione, il sito ha molto potenziale ed è ben progettato. Le informazioni sono fornite in modo semplice ed essenziale, e l'interfaccia grafica si presenta in maniera molto tradizionale. La piattaforma potrebbe però essere arricchita maggiormente sfruttando appieno le potenzialità del mezzo elettronico: gli strumenti messi a disposizione, infatti, sono abbastanza semplici, e non permettono una consultazione e un'analisi più approfondita da parte del lettore finale.

3.3 *Vincent van Gogh - The Letters*

Il progetto *Vincent van Gogh – The Letters*³² è finalizzato alla pubblicazione in digitale di tutta la corrispondenza scritta e ricevuta ancora esistente del pittore olandese Vincent van Gogh (30 marzo 1853 - 29 luglio 1890), precisamente di 903 lettere e 25 manoscritti correlati. Si tratta di un'edizione finalizzata alla ricerca, ideata principalmente per gli studiosi di Van Gogh, studenti di scuole d'arte, di letteratura o comunque a persone con uno sfondo accademico, anche se, essendo pubblicato online, tutti coloro che sono interessati possono consultare l'edizione. I testi sono codificati nel formato XML sia secondo gli standard TEI P5 sia con l'uso di alcuni tag non standard presi dal progetto DALF (*Digital Archive of Letters in Flanders*³³), e uno schema in formato DTD (*Document Type Definition*) creato appositamente per il progetto. Il sito che ospita l'edizione è stato creato con un moderno linguaggio di programmazione di alto livello, chiamato Ruby³⁴, e con tecnologie standard quali HTML e Javascript.

³² Vincent van Gogh – The Letters, <http://vangoghletters.org/vg/>.

³³ DALF, <https://ctb.kantl.be/project/dalf/>.

³⁴ Ruby, <https://www.ruby-lang.org/en/>.



Figura 12: *The Vincent van Gogh - The Letters: facsimile, original text e translation*

Ogni lettera è affiancata ai relativi metadati, e sono proposte diverse modalità di visualizzazione. La pagina è divisa in due frame, ognuno dei quali presenta una barra degli strumenti che permette di cambiarne il contenuto; in particolare, le varie opzioni sono: *original text*, con la trascrizione del testo in lingua originale, + *line ending*, dove viene mostrato il testo con le sue interruzioni di riga originali e numerate, *facsimile*, *translation*, con la traduzione del testo in inglese, *note*, che mostra le varie note esplicative del testo (Fig. 12), e *artworks* (Fig. 13), che mostra le opere d'arte che vengono citate all'interno della lettera. Inoltre, se lo schermo è abbastanza ampio, è possibile aggiungere fino a due frame oltre a quelli già presenti, ognuno dei quali può contenere una delle viste appena citate.

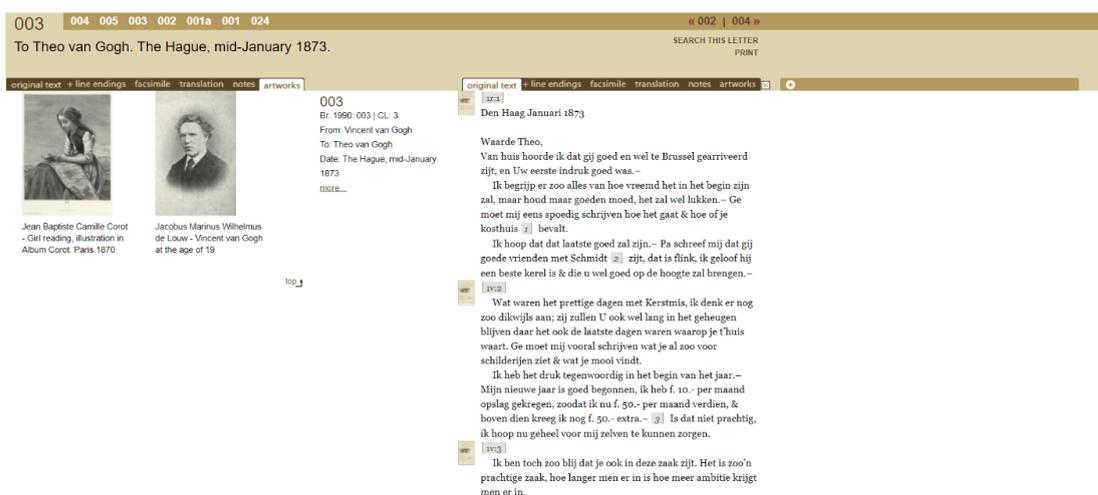


Figura 13: *The Vincent van Gogh - The Letters, artworks e original text*

Nel complesso, questo progetto di edizione si è rivelato essere una risorsa web funzionale e completa. Grazie all'aggiunta di nuove schede, la consultazione e il

confronto fra le varie informazioni risultano ottimali, permettendo all'utente finale una grande libertà di organizzazione delle risorse. Una debolezza da segnalare è la mancanza di collegamento fra l'immagine, il testo e le informazioni associate, che rende quindi più difficoltosa l'analisi del documento.

3.4 *Emily Dickinson Archive*

L'*Emily Dickinson Archive*³⁵ è un archivio *open-access* che si occupa di rendere disponibile al pubblico immagini e trascrizioni di manoscritti delle poesie di E. Dickinson. L'obiettivo del progetto è quello di mettere a disposizione agli studiosi di letteratura tutto il materiale utile e fondamentale per lo studio della poetessa, in un unico luogo virtuale facilmente consultabile. L'archivio segue le volontà dell'autrice, ovvero di far leggere al pubblico le proprie poesie in forma scritta a mano: la piattaforma consente infatti la visualizzazione dei facsimili delle opere ad alta risoluzione, oltre alla relativa trascrizione del testo.

Le trascrizioni e le note delle edizioni sono state codificate con il markup XML TEI. Ogni edizione è stata codificata separatamente in modo da permettere la visualizzazione e il confronto della loro cronologia, e la possibilità di creare un'edizione nuova (possibile solo se viene effettuato l'accesso al sito da parte dell'utente). Le trascrizioni sono accessibili tramite un sito web creato secondo gli standard HTML, CSS e Javascript.

³⁵ Emily Dickinson Archive: <https://www.edickinson.org/>.



Figura 14: *Emily Dickinson Archive, Reading View*

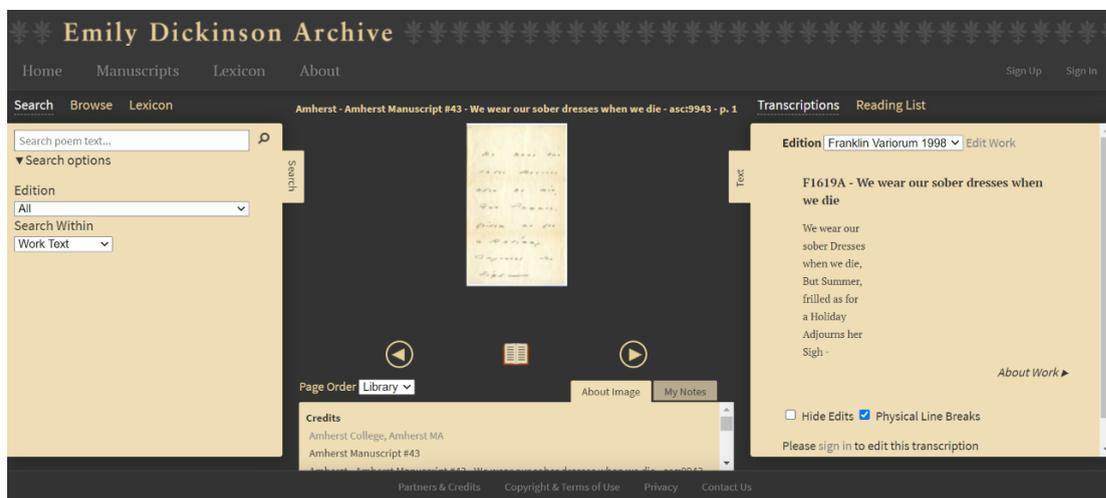


Figura 15: *Emily Dickinson Archive, Manuscript View*

L'edizione presenta una interfaccia ricca e semplice. Le modalità di visualizzazione sono due. La prima è la *Reading View* (Fig. 14), che permette di sfogliare le pagine del manoscritto mostrando contemporaneamente il verso della pagina e il recto della successiva. La seconda vista è la *Manuscript View* (Fig. 15), che permette di analizzare meglio l'immagine, di vederne la trascrizione, i metadati e le opzioni di ricerca. Al centro del *Manuscript View* è posto il facsimile del manoscritto con annessi strumenti per ispezionare meglio il contenuto (Fig. 16): *Zoom in* che ingrandisce l'immagine, *Zoom out* che la riduce, *Go home* che torna alla dimensione predefinita, e *Toggle full page* che allarga l'immagine a tutto schermo.



Figura 16: *Emily Dickinson Archive*, strumenti per l'analisi dell'immagine

Sulla destra si trova la scheda *Text*, che contiene la trascrizione del testo dell'edizione scelta, che può essere cambiata dal menù apposito. È inoltre possibile filtrare il contenuto visualizzato attraverso i bottoni *Hide edits* e *Physical Line Breaks*. Il primo nasconde le note testuali dell'edizione selezionata. Il secondo mostra la trascrizione con le stesse interruzioni di riga presenti nel manoscritto originale. Con un click su *About Work* è inoltre possibile visualizzare i metadati del documento e scaricarne la codifica. Sotto l'immagine è posta la scheda dei *Credits* dell'immagine, mentre sulla sinistra sono presenti tre frame: *Search*, un piccolo motore per la ricerca dei manoscritti, *Browse* che sfoglia in ordine alfabetico le immagini del manoscritto in base a criteri come la prima riga, l'edizione o il destinatario, e *Lexicon*, un dizionario completo sfogliabile che ospita più di 9.000 parole della poesia di Dickinson.

Per concludere, il progetto è interessante e nonostante le funzioni e gli strumenti messi a disposizione siano complessi, il loro utilizzo si rivela essere molto intuitivo. Altro punto di forza molto interessante è il fatto di poter creare e aggiungere all'edizione delle proprie note, dei commenti e delle modifiche, oltre alla possibilità di creare un'edizione digitale ex novo. Punto di debolezza è invece l'assenza delle note che arricchiscono il testo con informazioni essenziali per la sua comprensione.

3.5 *Mark Twain Project Online*

Il *Mark Twain Project Online*³⁶, rilasciato al pubblico nel 2007, è un progetto che ha come scopo la pubblicazione di edizioni critiche digitali delle opere scritte di Mark Twain. Il formato della codifica usato per i testi è l'XML secondo gli standard *Text Encoding Initiative P4*.

³⁶ Mark Twain Project Online, <https://www.marktwainproject.org/homepage.html>.

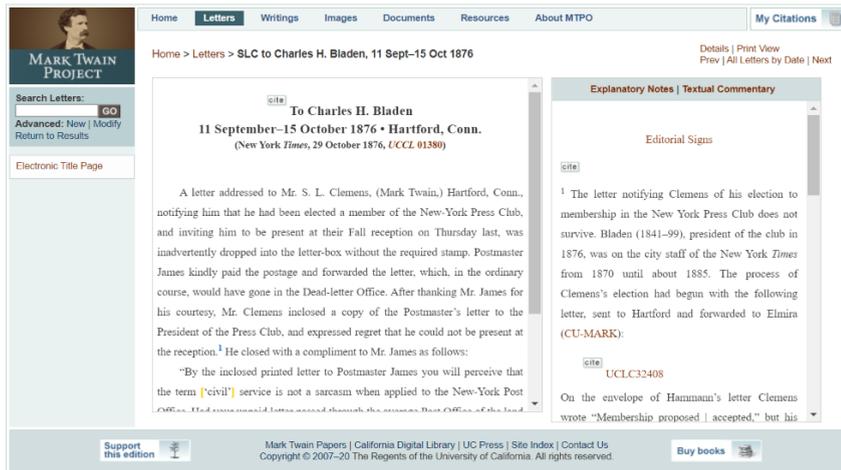


Figura 17: *Mark Twain Project Online*, vista predefinita

Esistono diverse modalità di visualizzazione. Quella predefinita (Fig. 17) consiste in due schede poste in parallelo sulla schermata che consentono di visualizzare contemporaneamente la trascrizione sulla sinistra e i le note esplicative (come i segni editoriali, la descrizione del testo e il processo di revisione) sulla destra. Molto interessante è la funzione *click-to-locate*, in cui si può cliccare o passare il mouse sul testo evidenziato del frame di sinistra e accedere alle informazioni associate sulla destra; oppure, viceversa, fare click o passare il mouse su una nota nel frame di destra evidenziando la corrispondente parte nel riquadro di sinistra. Il contenuto del frame di sinistra può essere cambiato per poter visualizzare il facsimile (Fig. 18) del manoscritto (se presente): esso è provvisto di strumenti come lo *zoom* per ingrandire ad alta risoluzione l'immagine. Infine, è presente la *Print View*, dove viene visualizzato il testo e le note in un unico documento.

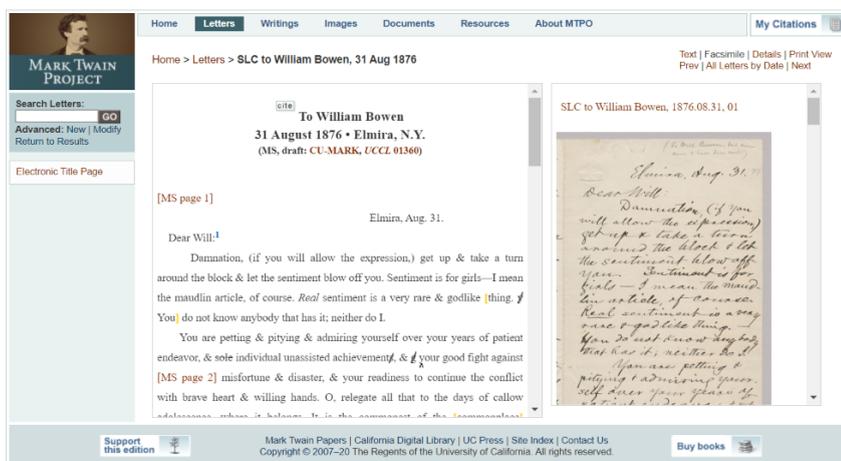


Figura 18: *Mark Twain Project Online*, Facsimile

Il progetto presenta quindi un'interfaccia grafica abbastanza intuitiva e semplice. Le funzioni e gli strumenti fruibili sono molto pratici, rendendo pertanto la consultazione dei testi un'operazione assai efficiente.

4. Progettazione e implementazione

4.1 Funzionalità desiderate

Il progetto di tesi riguardante EVT 3 è sviluppato in modo da affrontare e sviluppare varie funzionalità che hanno in comune la facilitazione dei confronti fra testi e immagini di manoscritti.

Di principale importanza è stata l'implementazione di una nuova vista legata all'elemento TEI <msDesc> (*manuscript description*) e lo sviluppo di <msDesc> multipli, volto a migliorare la già esistente vista *text-text* aggiungendo la possibilità di visualizzare in parallelo la descrizione dei testi mostrati. Altra novità importante apportata all'interno del progetto è la costruzione di un'ulteriore vista: la *view image-image*.

Nello specifico, tale lavoro di implementazione è divisibile in due fasi distinte:

- Progettazione dell'interfaccia utente, sviluppata attraverso la discussione con il resto del team di sviluppo EVT e la creazione di mockup che raffigurano la resa finale dell'applicazione, in modo da rendere il sistema il più *user-friendly*³⁷ possibile.
- Implementazione del software, mediante la scrittura del codice e la realizzazione pratica delle funzionalità desiderate.

Tali implementazioni hanno tuttavia richiesto uno studio preliminare sul linguaggio adottato da EVT 3, in questo caso il *framework* Angular 9, un'analisi sui dati da manipolare con particolare attenzione alle linee guida TEI, e lo studio dell'infrastruttura EVT, in modo da capire come rendere coerente il lavoro da me svolto con quello già presente.

³⁷ *User-friendly*, <https://www.wikibit.it/u/cosa-significa-user-friendly-1844/>.

4.2 Angular

Per l'implementazione di EVT 3 è stato utilizzato Angular³⁸. Si tratta di una piattaforma di sviluppo *open source* scritta con il linguaggio di programmazione Typescript³⁹ e HTML. È stato sviluppata da Google, che ha rilasciato la prima versione nel corso del 2016. Essa include: un *framework* per la creazione di applicazioni web dinamiche basate su componenti, una raccolta di librerie per l'aggiunta di funzionalità e un insieme di strumenti utili per costruire e testare il codice.

Angular implementa il pattern architettonico MVC che consente la separazione delle competenze. Il pattern si divide in tre gruppi principali di componenti:

- *Model*: gestisce i dati dell'applicazione;
- *View*: rappresenta quello che l'utente finale visualizza, l'interfaccia grafica creata a partire dal *template* HTML.
- *Controller*: si occupa delle interazioni tra il *View* e il *Model*.

Grazie alla separazione logica è possibile una maggiore indipendenza tra i componenti, ma anche una più facile manutenibilità dell'applicazione, con la possibilità quindi di apportare modifiche senza che si vada ad intaccare gli altri componenti. Questi vantaggi sono fondamentali in applicazioni di grandi dimensioni quale è EVT, poiché permettono un più facile sviluppo e *debugging* delle varie componenti.

Componenti

Fra gli elementi fondamentali per la costruzione di applicazioni Angular ci sono sicuramente i *components*. Essi si occupano di gestire la logica di interazione fra i dati e l'utente. Ogni *component* è costituito da:

- un selettore che definisce come il componente viene utilizzato in un modello,
- una classe Typescript che definisce il comportamento del componente,

³⁸ Home page: <https://angular.io/>.

³⁹ Typescript, <https://www.typescriptlang.org/>.

- un *template* HTML che specifica l'interfaccia utente,
- uno o più stili CSS per definire l'aspetto grafico.

Un'applicazione Angular può avere molteplici componenti, ognuno dei quali si occupa di gestire una determinata porzione dell'interfaccia utente.

L'uso dei componenti risulta essere estremamente vantaggioso grazie alla possibilità di riutilizzare il codice: usando infatti un insieme comune di componenti è possibile assemblare applicazioni diverse, favorendone anche la leggibilità e i tempi di sviluppo.

Template

Un *template* è una porzione di codice HTML con l'aggiunta di *markup* Angular. Ogni componente ha il proprio *template* per definire come esso dovrà essere visualizzato nell'applicazione. Come già accennato, oltre all'HTML, è ammessa una sintassi speciale per poter sfruttare le funzionalità di Angular all'interno del modello, permettendo così la possibilità di inserire dei valori dinamici. Angular consente inoltre l'aggiornamento in tempo reale del DOM attraverso il meccanismo di *binding* bidirezionale.

Il *binding* bidirezionale gestisce la sincronizzazione tra DOM e modello, quindi ogni modifica effettuata nel modello dati si riflette immediatamente nella *View*, e viceversa (Fig. 19).

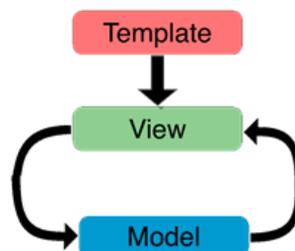


Figura 19: Angular: *Two-Way Data Binding*

Servizi e *dependency injection*

Altro costituente fondamentale per la costruzione di un'applicazione Angular sono i servizi. Essi hanno il compito di gestire dati e informazioni utilizzabili all'interno dell'applicazione, gestendoli in entità esterne ai componenti. Ogni servizio svolge un compito ben preciso ed è possibile averne molteplici all'interno dell'applicazione. Una volta creati, i servizi possono essere “iniettati” all'interno di un componente grazie alla *dependency injection*.

La *dependency injection* è un design pattern che si occupa di gestire le dipendenze e di fornire, quando richieste, informazioni recuperate da entità esterne. Essa permette di semplificare lo sviluppo e la testabilità di applicazioni di grandi dimensioni.

4.3 Progettazione interfaccia utente

L'interfaccia utente, o UI (*User Interface*), è il componente percepibile di un sistema interattivo, ciò che consente all'utente finale di interagire e comunicare con l'applicazione: è in sintesi il layout grafico, ciò che il lettore finale vede. La scelta della UI da adottare ha preso ispirazione da entrambe le precedenti versioni di EVT.

L'interfaccia proposta si compone di un nuovo selettore nello *header* del riquadro contenente il testo del documento (fig. 20), che permette di poter scegliere quale *manuscript description* visualizzare. Cliccando nel menù a tendina verrà proposto un elenco degli <msDesc> presenti rappresentati dal loro ID. Una volta aperto, sarà possibile chiudere la *view* attraverso l'apposito pulsante “x” presente nel selettore (fig. 21).



Figura 20: *mockup* per il nuovo selettore 'MSDesc'



Figura 21: *mockup* per la vista msDesc

Scelto l'<msDesc> da visualizzare, vengono mostrate le informazioni riguardanti la descrizione del manoscritto (Fig. 22). All'interno del *manuscript description*, ogni sezione è segnalata da titolo sottolineato, seguito da vari *label* con la propria informazione associata. Particolare attenzione è stata rivolta alla gestione delle informazioni contenute all'interno del *manuscript contents* (<msContents>): l'elemento infatti può avere al suo interno vari <msItem>, quindi varie parti del manoscritto, ognuno con un proprio titolo, autore, locus, incipit, explicit ecc. Per gestire al meglio la visualizzazione di tale parte, ogni *item* viene individuato con il proprio <locus> in grassetto, posizionato sulla colonna sinistra del *frame*, mentre sulla destra vengono elencate tutte le relative informazioni. C'è poi la possibilità di

un annidamento di `<msItem>`: il `<locus>` dell'*item* annidato verrà posto alla stessa altezza delle informazioni dell'`<msItem>` padre. Nel caso in cui il `<locus>` sia assente all'interno dell'*item*, le informazioni vengono allineate nella colonna sinistra dello schermo (Fig. 22).

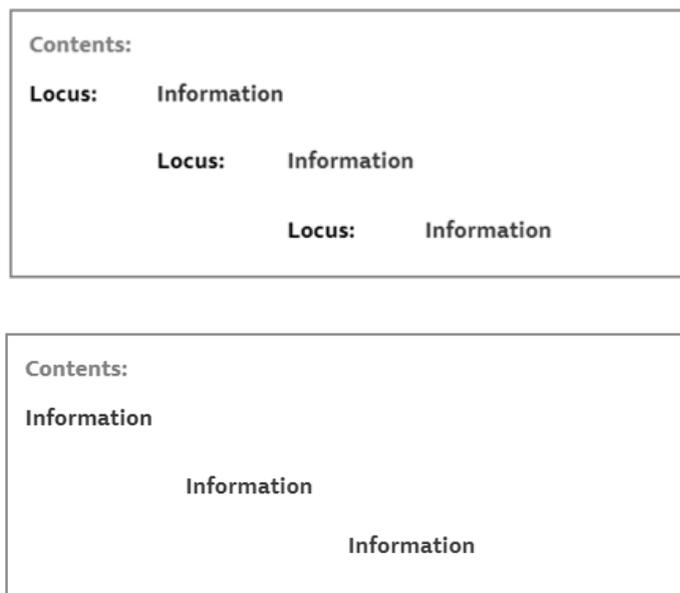


Figura 22: *mockup* per l'annidamento di `msItem`, con e senza `<locus>`

In merito alla visualizzazione di `<msDesc>` multipli, l'idea è quella di implementare all'interno della vista *text-text* la possibilità di visualizzare per ogni manoscritto la descrizione corrispondente scelta dall'apposito selettore (Fig. 23).

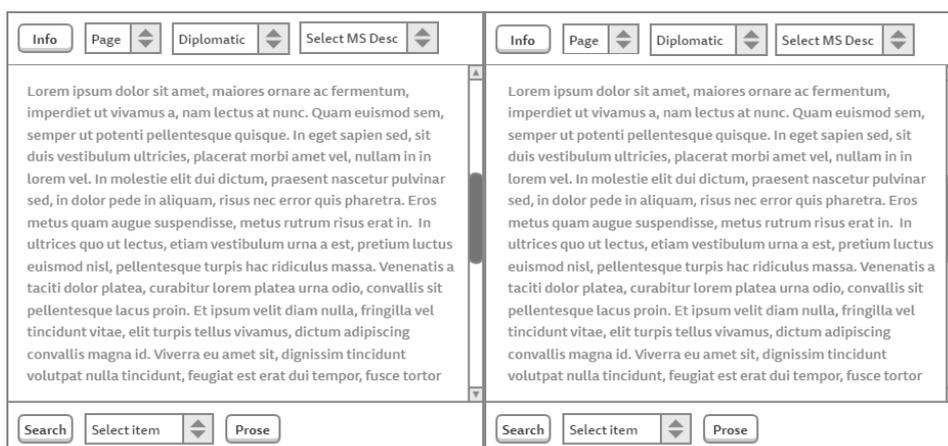


Figura 23: *mockup* della *view text-text* aggiornata

Infine, l'interfaccia proposta per la visualizzazione della view *image-image* si compone di due immagini affiancate in parallelo nella stessa schermata, ognuna arricchita da una barra degli strumenti in modo da facilitare il confronto (Fig. 24).

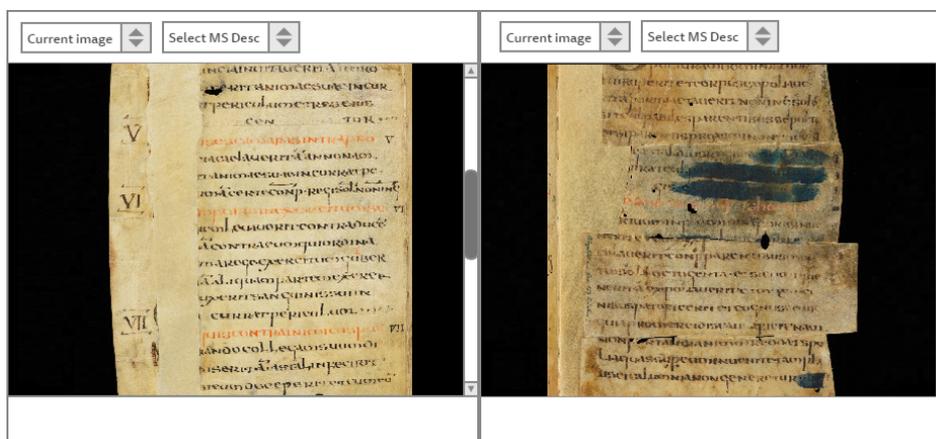


Figura 24: mockup della nuova view *image-image*

Per quanto riguarda lo stile, è stato preferito rimanere sui toni del bianco e del grigio, adattandosi ai colori del resto dell'applicazione.

4.4 Implementazione

Dopo aver progettato l'interfaccia grafica del prototipo e aver approfondito le caratteristiche del *framework* da utilizzare, il passo successivo è stato l'implementazione vera e propria del software⁴⁰. Essa a sua volta è divisibile in fasi: realizzazione del modello dati, scrittura dei *parser* e creazione/modifica dei componenti. Vedremo nel corso del capitolo alcuni dei punti più interessanti di tali fasi. Il codice completo delle nuove funzionalità è disponibile su GitHub nel repository EVT 3⁴¹.

4.4.1 Modello dati

Lo scopo della prima fase è stato quello di definire dei modelli dati che rappresentino al meglio gli elementi XML/TEI necessari per lo sviluppo. Prima della

⁴⁰ La modalità di funzionamento di EVT 3 è stata già descritta nella sezione 2.4.

⁴¹ <https://github.com/evt-project/evt-viewer-angular/tree/develop>.

loro implementazione vera e propria, è stato fondamentale lo studio dei tagName coinvolti (in questo caso gli elementi appartenenti al modulo *manuscript description*) attraverso le linee guida TEI. La *Text Encoding Initiative*, infatti, rappresenta un punto di riferimento per la digitalizzazione delle fonti testuali in ambito umanistico, e ha fra i propri obiettivi quello di fornire delle direttive per la gestione dei dati presenti in tali risorse. Le informazioni che in particolare sono servite per la creazione dei *data models* sono state quelle riguardo le relazioni che ogni elemento XML/TEI intraprende con gli altri (gli attributi che può avere, gli elementi in cui può essere contenuto e quali possono contenere a sua volta).

Alla fase preliminare di studio è seguita la fase di scrittura vera e propria dei modelli di dati, traducendo le informazioni recuperate nel *framework*. I vari modelli dati vengono definiti come delle interfacce che estendono il `GenericElement` (classe che dichiara le caratteristiche comuni a tutti gli elementi XML/TEI). Ogni tagName è stato descritto come un array associativo (dizionario) che sfrutta il meccanismo chiave/valore. Al suo interno possiamo trovare dichiarati sia gli attributi rilevanti dell'elemento, con specificato il tipo (es: stringa, numero, booleano), sia i possibili elementi figli, andando a creare dei collegamenti diretti con le relative classi, se esistenti. Nel caso contrario, l'elemento è stato dichiarato come `Array<ParseResult<GenericElement>>`, affiancandolo a un commento `TODO` che ricordi, a chi ne avrà il compito in futuro, di gestire quell'elemento specifico.

Tali interfacce vengono dichiarate all'interno dell'EVT Model, file dove vengono descritti tutti i modelli dei vari elementi XML/TEI. Un esempio di modello dati appartenente al modulo *Manuscript description* è il seguente:

```
export class Seal extends GenericElement {
  contemporary: boolean;
  sealType: string;
  n: string;
  decoNote: DecoNote;
  pEl: Paragraph[];
  ab: Array<ParseResult<GenericElement>>; // TODO: Add specific
type when ab is handled
}
```

L'elemento sopra dichiarato è l'elemento XML/TEI `<seal>`. Grazie al modello dati possiamo comprendere che tale `tagName` può contenere gli attributi `@contemporary`, con valore `True` o `False`, `@sealType` e `@n`, entrambi con valore stringa, e può inoltre avere come nodi figli gli elementi `<decoNote>`, `<p>` e `<ab>`.

4.4.2 Scrittura dei *parser*

Una volta creati i vari modelli dati degli elementi appartenenti al modulo *manuscript description*, è stato implementato per ognuno di essi un *parser*. Esso ha lo scopo di estrarre i dati dal documento XML, che verranno poi gestiti e manipolati successivamente dai relativi componenti. I *parser* vengono definiti in una mappa che associa un *parser* a ogni `tagName` supportato.

Per ogni elemento dichiarato nel modello dati viene creata una classe che:

- implementa il `Parser<XMLElement>`,
- possiede le proprietà `type`, con valore l'interfaccia stessa,
- possieda una costante dove vengono salvati tutti gli attributi del nodo,
- ritorni i valori degli attributi e dei figli dell'elemento.

Per lo sviluppo delle nuove funzionalità, i vari *parser* appartenenti al modulo *manuscript description* sono stati dichiarati all'interno di un unico file. Di seguito un esempio:

```
@xmlParser('seal', SealParser)
export class SealParser extends GenericElemParser implements
Parser<XMLElement> {
  parse(xml: XMLElement): Seal {
    const genericElem = super.parse(xml);
    const { n, sealType } = genericElem.attributes;
    return {
      ...genericElem,
      type: Seal,
      contemporary: isBoolString(xml.getAttribute
('contemporary')),
```

```

        decoNote: queryAndParseElement(xml, 'decoNote'),
        sealType,
        n: getDefaultN(n),
        pEl: queryAndParseElements(xml, 'p'),
        ab: unhandledElement(xml, 'ab', this.genericParse),
    };
}
}

```

4.4.3 Implementazione dei componenti

La terza fase dell’implementazione prevede la creazione dei *components*, o la modifica di quelli già esistenti, per la gestione della visualizzazione dei dati.

View <msDesc> e <msDesc> multipli

La prima funzionalità implementata è stata la nuova vista <msDesc> per la visualizzazione della descrizione del manoscritto. Sono stati inizialmente realizzati, nel seguente ordine, i componenti relativi agli elementi figli di <msDesc>: <msIdentifier>, <msContents>, <history>, <physDesc>, <additional>, <msPart> e <msFrag>.

I *template* di tali componenti seguono a grandi linee la stessa struttura: all’interno del <div> principale si trova in primo luogo il titolo della sezione visualizzata (es: “*Manuscript identifier*”), seguito da un <div class=“flex-container”> contenente varie porzioni, ognuna distinta dal tag <ng-container> e specifica per una determinata informazione. Ciascuna di queste porzioni presenta al suo interno due contenitori: il primo è caratterizzato dalla classe “flex-item-label” contenente l’etichetta dell’informazione analizzata, il secondo è caratterizzato dalla classe “flex-item-information”, al cui interno si trova il valore dell’informazione. Gli stili sono stati definiti attraverso il relativo foglio di stile CSS. Di seguito è visibile un frammento di codice che mostra la struttura generale di questi primi componenti.

```

<div class="msIdentifier">
  <evt-ms-desc-section [label]='msIdentifier'
[underline]=true></evt-ms-desc-section>
  <div class="flex-container">
    [. . .]
    <ng-container *ngIf="data?.msName.length>0">
      <div class="flex-item-label">
        <evt-ms-desc-section [label]='msName'
[additionalClass]='msName'
[inlineLabel]="true"></evt-ms-desc-section>
      </div>
      <div class="flex-item-information">
        <evt-content-viewer *ngFor="let m of
data.msName" [content]="m"
class="projectInfo-block"></evt-content-viewer>
      </div>
    </ng-container>
    [. . .]
  </div>
</div>

```

Ogni titolo e ogni *label* sono stati localizzati e dichiarati in appositi file JSON: in questo modo è stata offerta una soluzione per il cambio di lingua senza la necessità di ricaricare completamente l'applicazione. Esiste un file JSON per ogni traduzione esistente (al momento è presente uno per l'italiano e uno per l'inglese). Il file è organizzato come un array associativo con coppie chiave-valore:

```

{
  [...]
  "msIdentifier": "Manuscript identifier",
  [...]
}

```

La coppia presenta sia una chiave da utilizzare all'interno dei *template* HTML, sia il valore della traduzione da visualizzare. All'interno dei *template* quindi, ogni qual volta è necessario recuperare un testo tradotto, viene inserita la *pipe translate* preceduta dalla chiave.

Dopo aver realizzato i componenti per la visualizzazione dei figli del *Manuscript Description*, la fase successiva è stata la creazione del componente

dell'elemento principale `<msDesc>`. Nel suo relativo *template* sono stati elencati tutti i figli con i relativi selettori.

```
[ ... ]
  <evt-ms-contents *ngIf="data?.msContents"
[data]="data.msContents"></evt-ms-contents>
  <evt-phys-desc *ngIf="data?.physDesc"
[data]="data.physDesc"></evt-phys-desc>
  <evt-history *ngIf="data?.history"
[data]="data.history"></evt-history>
  <evt-additional *ngIf="data?.additional"
[data]="data.additional"></evt-additional>
[ ... ]
```

Una volta terminata la gestione della visualizzazione dei tag XML/TEI *Manuscript Description*, è stato necessario realizzare un selettore. Esso è stato inserito all'interno della barra degli strumenti e permetta di scegliere quale `<msDesc>` visualizzare tra quelli presenti nel file XML, integrando così la visualizzazione di `<msDesc>` multipli.

Per ottenere questo risultato è stato modificato un componente già esistente, il `text-panel-component`, che si occupa di far visualizzare il frame relativo al testo e agli strumenti per poterlo analizzare. Nel relativo *template* è stato necessario aggiungere il selettore attraverso il tag Angular `<ng-select>`: esso è in grado di creare dinamicamente degli elenchi di dati che possono essere selezionati da parte degli utenti tramite un selettore a discesa. In questo modo vengono visualizzati tutti gli `<msDesc>` presenti senza alcun vincolo sulla quantità.

```
<ng-select
  class="mr-1"
  [closeOnSelect]="true"
  [searchable]="false"
  bindLabel="id"
  bindValue="id"
  [items]="msDesc$ | async"
  [clearable]="true"
  (clear)="resetMsDesc();"
  [placeholder]=" 'selectMsDesc' | translate"
```

```

        (change)="msDescID = $event.id;
openMsDescContent () ">
    <ng-template
        ng-id-tmp
        let-item="item">
        {{ item.id }}
    </ng-template>
</ng-select>

```

La direttiva `<ng-select>` richiama al suo interno due funzioni: `openMsDescContent()` e `resetMsDesc()`. La funzione `openMsDescContent()` rende le variabili booleane `showSecondaryContent` e `msDescOpen` come `True`: la prima permette di far apparire un pannello che si sovrappone al frame di testo presente, la seconda consente di poter visualizzare i dati dell'`<msDesc>` nell'interfaccia utente; la funzione `resetMsDesc()` fa esattamente il contrario: cliccando la "x" presente nel selettore il layout viene ripristinato.

Al fine di estrarre tutti i dati relativi ai diversi *Manuscript Description* presenti nell'XML, è stato implementato un nuovo *service* specifico per il *parser* `ms-desc`. Al suo interno è stata sviluppata la funzione `parseMsDesc()` in grado di restituire un array (fig. 25) per ogni `<msDesc>` presente all'interno dell'XML.

```

ms-desc-parser.service.ts:15
▼ (3) [{...}, {...}, {...}] ⓘ
  ▶ 0: {class: "msdesc", content: Array(13), attributes: {...}, additional: null, type: f, ...}
  ▶ 1: {class: "msdesc", content: Array(3), attributes: {...}, additional: null, type: f, ...}
  ▶ 2: {class: "msdesc", content: Array(3), attributes: {...}, additional: null, type: f, ...}
    length: 3
  ▶ __proto__: Array(0)

```

Figura 25: esempio di array restituiti dal *service* `ms-desc-service`

View image-image

Ulteriore funzionalità che è stata implementata all'interno del progetto è quella legata alla vista *image-image* per il confronto in parallelo di facsimili diversi.

A tale scopo, è stato creato un nuovo componente denominato `image-image`. All'interno del relativo file Typescript è stato dichiarato un layout a griglia, specificando misure e proprietà, sfruttando l'interfaccia `GridsterItem`. Nel *template* è stata invece sviluppata una griglia attraverso il tag `<gridster>` e i relativi figli `<gridster-item>`, uno per ogni *frame* della schermata. Ogni *item* (in questo caso due) contiene il selettore `<evt-image-panel>` per richiamare il componente specifico per la visualizzazione dell'immagine.

```
<gridster [options]="layoutOptions">
  <gridster-item [item]="imagePanel1Item">
    <evt-image-panel></evt-image-panel>
  </gridster-item>
  <gridster-item [item]="imagePanel2Item">
    <evt-image-panel></evt-image-panel>
  </gridster-item>
</gridster>
```

Per poter accedere alla vista, è stata aggiunta una nuova icona nella barra degli strumenti, affiancandola alle già esistenti icone *Reading Text*, *Image Text*, *Text Text*, *Collation*, *Text Sources* e *Text Versions*. Per fare ciò è stato aggiornato l'array `availableViewModes` all'interno del file `edition_config.json`, inserendo un nuovo oggetto specifico con associati i valori della nuova icona:

```
{
  "icon": "imgImg",
  "iconSet": "evt",
  "id": "imageImage",
  "label": "Image Image"
}
```

Per ottenere poi il collegamento tra l'icona e il relativo componente, è stato aggiunto il *component* `image-image` all'interno dell'array di oggetti `appRoutes`. Quest'ultimo serve a definire la configurazione della route. Così facendo è stato implementato il *path* della nuova *view* associando l'identificatore dell'icona (`imageImage` in questo caso) al relativo componente (`ImageImageComponent`). Cliccando sulla specifica icona è quindi possibile visualizzare facilmente la nuova vista.

```

const appRoutes: Routes = [
  { ... }
  { path: 'imageImage', component: ImageImageComponent },
  { ... }
];

```

Successivamente, sono stati modificati il già esistente *component* `image-panel`, il quale scopo è quello di visualizzare correttamente il *frame* legato all'immagine e ai relativi strumenti di analisi, e il *component* `osd`, che ha il compito di far visualizzare le immagini dei facsimili provenienti da un file JSON esterno. All'interno del componente `image-panel`, è stato implementato un nuovo selettore in grado di visualizzare dinamicamente, attraverso un menù a cascata, tutti i `label` delle immagini presenti nel file JSON, permettendo così all'utente la possibilità di scegliere il facsimile da visualizzare.

```

<ng-select
  class="mr-1"
  bindLabel="label"
  bindValue="label"
  [searchable]="true"
  [clearable]="false"
  [ngModel]="currentImg"
  (change)="toggleImg($event)">
  <ng-option [value]="item" *ngFor="let item of images;">
    {{ item.label }}
  </ng-option>
</ng-select>

```

L'array preso di riferimento all'interno di `<ng-select>` è stato costruito recuperando i dati dal server attraverso il metodo `get()`, con argomento il valore dell'URL del file JSON. È stato poi iterato l'oggetto recuperato al fine di estrarre e salvare in un array apposito, `images`, i valori di ogni singolo facsimile presente. Attraverso tale array è stato quindi possibile visualizzare correttamente tutti i `label` esistenti al click della freccia del selettore.

```

this.http.get(`${this.manifest}`).subscribe((item: any) => {
  item.sequences.forEach((value) => {
    this.images.push(...value.canvases);
  });
});

```

```
});
```

Cliccando uno dei valori elencati, il selettore richiama la funzione `toggleImg()` per recuperare l'indice dell'immagine scelta e inoltrarla al componente `osd`. All'interno di quest'ultimo, l'indice viene ripreso attraverso la direttiva `@Input`⁴², e con la proprietà `goToPage`, viene immediatamente mostrata nell'interfaccia finale il facsimile selezionato.

```
private _page: number;
@Input() set page(v: number) {
  if (v !== this._page) {
    this._page = v + 1;
    this.pageChange.next(this._page);
  }
}
get page() { return this._page; }
```

È stata infine inserita all'interno del Typescript del componente `image-image` la funzione `eventHandler()`. Il compito di questa funzione è di recuperare l'indice dell'immagine che in quel momento è visualizzata nella schermata: l'obiettivo è di visualizzare il relativo `label` all'interno del selettore così da sapere in qualsiasi momento quale pagina stiamo visualizzando. Per fare ciò, `eventHandler()` recupera il valore attraverso il parametro `event`, che recupera a sua volta l'informazione dall'attributo `this.data.emit(page)` presente nel component `osd`. Tale attributo viene aggiornato ogni qual volta la pagina visualizzata viene cambiata.

```
eventHandler(event: number) {
  this.page = event;
  this.currentImg = this.images[this.page].label;
  this.cdref.detectChanges();
}
```

⁴² `@Input`: «Decorator that marks a class field as an input property and supplies configuration metadata. The input property is bound to a DOM property in the template. During change detection, Angular automatically updates the data property with the DOM property's value», <https://angular.io/api/core/Input>.

Conclusioni

Attraverso il seguente elaborato, sono state esposte le varie fasi che hanno portato all'implementazione di nuove funzionalità legate al confronto di informazioni e di immagini all'interno del progetto digitale EVT 3. L'obiettivo era creare delle *feature* che fossero in grado di permettere uno studio più approfondito e preciso dell'edizione digitale attraverso il confronto di oggetti diversi, fornendo un prezioso strumento in grado di semplificare il lavoro di analisi e paragone a studiosi e appassionati. Tale obiettivo è stato raggiunto grazie all'implementazione delle seguenti funzionalità:

- lo sviluppo della *view manuscript description* e il supporto per descrizioni multiple, grazie alla quale è possibile visualizzare e comparare facilmente tutte le informazioni riguardanti le caratteristiche del manoscritto;
- lo sviluppo della *view image-image*, attraverso cui immagini diverse possono essere facilmente messe a confronto.

La collaborazione all'interno del progetto EVT, da un punto di vista personale, nonostante le difficoltà iniziali si è rivelato essere una preziosa esperienza che mi ha permesso di applicare le competenze acquisite durante il mio corso di studi in un contesto reale e di avvicinarmi al mondo della filologia e della critica testuale. Ho avuto l'occasione di imparare a utilizzare nuovi strumenti e linguaggi innovativi, in particolare il *framework* di sviluppo Angular, e di migliorare le mie competenze nel campo della progettazione e programmazione web.

Bibliografia

Mario Ferrara, Graziano Ramina, *Tecnologie dei processi di produzione*, Clitt Editore, 2016.

Mario Moiraghi, Maurizio Cali, *Scripta manent*, Scriptorium, 2003 URL: <http://www.scriptorium.it/scripta1.pdf>.

Fabio Macciò, *Introduzione allo studio della filologia classica*, Loescher Editore, Media Classica.

Armando Petrucci, *Breve storia della scrittura latina*, Roma: Bagatto Libri, 1989.

Maria Rosa Formentin, *Le discipline della scrittura*, URL: <http://www2.units.it/eussc/pdf11/Formentin.pdf>.

Lucia Cesarini Martinelli, *La filologia*, Roma: Editori Riuniti, 1984.

Salvatore Battaglia, *Grande dizionario della lingua italiana*, Torino: Unione tipografico-editrice torinese, 1961-2004.

Paolo Chiesa, *Elementi di critica testuale*, Patròn Editore, 2002.

Giorgio Ziffer, Paul Mass, *La critica del testo*, Roma: Edizioni di Storia e letteratura, 2017.

Louis Rémy Brégeaut, Alois Senefelder, *Manuale di litografia, o sia Istruzione teorico-pratica pel disegnatore e per lo stampatore litografo*, Milano: Coi tipi di Felice Rusconi, 1828.

Jasmine Elizabeth Burns, *Digital Facsimiles and the Modern Viewer: Medieval Manuscripts and Archival Practice in the Age of New Media*, *Art Documentation: Journal of the Art Libraries Society of North America*, vol. 33, no. 2, 2014.

Elena Pierazzo, *Digital Scholarly Editing: Theories, Models and Methods*, Farnham, Surrey: Ashgate, 2015.

Paolo Divizia, *Fenomenologia degli errori guida, Filologia e critica*, Roma: Salerno Editrice, 2011.

Giorgio Vedovati, *Introduzione alla filologia*, 2008, URL: http://www.liceoamaldi.it/?q=system/files/Introduzione_alla_filologia.pdf.

Manfred Kramer, *Imagination, Almanach 1986-1993*, Sammelheft, 2006, URL: <https://www.omifacsimiles.com/kramer.html>.

Denis Defibaugh, *The Collotype: History, Process & Photographic Documentation [Ma Thesis]*, Rochester: Rochester Institute of Technology, 1997.

Claudia Corfiati, *Elementi di filologia*, URL: <https://www.uniba.it/docenti/corfiati-claudia/attivita-didattica/Filologia.pdf>.

Roberto Rosselli Del Turco, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti, Jacopo Pugliese, *Edition Visualization Technology: A Simple Tool to Visualize TEI-based Digital Editions*, Journal of the Text Encoding Initiative Issue 8, 2018, URL: <http://jtei.revues.org/1077>.

Roberto Rosselli Del Turco, *Designing an Advanced Software Tool for Digital Scholarly Editions: The Inception and Development of EVT (Edition Visualization Technology)*, Textual Cultures, vol. 12, no. 2, 2019, URL: www.jstor.org/stable/26821538.

Roberto Rosselli Del Turco, Chiara Di Pietro, Chiara Martignano, *Progettazione e implementazione di nuove funzionalità per EVT 2: lo stato attuale dello sviluppo*, Umanistica Digitale, 2019, URL: <https://doi.org/10.6092/issn.2532-8816/9322>.

Brad Green, Shyam Seshadri, *AngularJS Up & Running*, O'Reilly Media, Inc., 2013.

Pablo Deeleman, *Learning Angular 2*, Packt Publishing, 2016.

Sitografia

Il punto quotidiano Albo Scuole,

<https://www.ilpuntoquotidiano.it/alboscuole/il-duro-lavoro-degli-amanuensi/>.

Amanuense.it, <http://www.amanuense.it/>.

Racconti di storia,

<https://www.raccontidistoria.com/storia-della-scrittura-e-del-libro/le-discipline-che-studiano-la-scrittura-e-i-libri/>.

Treccani, <https://www.treccani.it/enciclopedia/filologia/>.

Treccani, <https://www.treccani.it/vocabolario/eidotica/>.

Text Encoding Initiative (TEI), <http://www.tei-c.org/>.

Digital Humanities Workbench,

https://www2.fgw.vu.nl/werkbanken/dighum/digital_data/digital_text/types/facsimile.php.

Facsimile Finder, <https://www.facsimilefinder.com/articles/analog-to-digital/>.

Encyclopedia Britannica, <https://www.britannica.com/technology/offset-printing>.

Caxton's Canterbury Tales: The Prologue,

<https://www.bl.uk/treasures/caxton/record.asp?LHPage=1v&LHvol=&LHCopy=1&RHPage=2r&RHvol=&RHCopy=1&disp=s&Linked=0#DispTop>.

Frankenstein, <http://shelleygodwinarchive.org/contents/frankenstein/>.

Rosario Lentini, Come distruggere manoscritti e libri senza ricorrere alle fiamme,

<https://www.istitutoeuroarabo.it/DM/come-distruggere-manoscritti-e-libri-senza-ricorrere-alle-fiamme/>.

Stampa digitale e stampa offset, quali sono le differenze?,

<https://www.graziani.it/blog/stampa-digitale-e-stampa-offset-quali-sono-le-differenze>

.

Marco Rovere, Viaggio nella storia della fotografia,
<https://www.nikonschool.it/corso-breve-storia-fotografia/>.

EVT, <http://sourceforge.net/projects/evt-project/>.

The Digital Vercelli Book, <http://vbd.humnet.unipi.it>.

E-codices, <https://www.e-codices.unifr.ch/it>.

The Shelley-Godwin Archive, <http://shelleygodwinarchive.org/>.

Vincent van Gogh – The Letters, <http://vangoghletters.org/vg/>.

Emily Dickinson Archive: <https://www.edickinson.org/>.

Mark Twain Project Online, <https://www.marktwainproject.org/homepage.html>.