

## Un modello concettuale per favorire lo sviluppo e il riutilizzo di strumenti software per edizioni digitali

Chiara Martignano

Università di Firenze, Italia  
chiara.martignano@unifi.it

### Abstract

Nel campo della filologia digitale sono stati realizzati negli anni numerosi applicativi per la visualizzazione di edizioni digitali, prodotti informatici sofisticati e innovativi. Il loro riutilizzo per progetti di edizione diversi si è rivelato essere parecchio complesso, nonostante si tratti talvolta di strumenti concepiti come generici. Riguardo a questo problema, in un articolo sul futuro delle edizioni digitali, Pierazzo sostiene che a mancare è la proposta di un modello scientifico fondato sulla teoria e sull'editing filologico ([18]). Nel presente articolo si analizzano le basi teoriche e le implementazioni tecniche mediante le quali si potrebbe realizzare un modello concettuale che consenta di facilitare lo sviluppo e il riuso di applicativi per edizioni digitali. Dall'analisi emerge la proposta di fondare il modello sulla descrizione delle diverse metodologie filologiche e di formalizzarlo sotto forma di ontologia, in modo che risulti adattabile e interoperabile tra edizioni diverse sia su un piano teorico sia su un piano tecnico. L'articolo si conclude con una rassegna degli aspetti da indagare ulteriormente, affinché il modello concettuale possa diventare una soluzione concreta che favorisca la generalizzazione e il riuso degli applicativi per le edizioni digitali, in un'ottica di maggiore sostenibilità e di consolidamento delle sperimentazioni condotte finora.

In the field of scholarly digital editing, many innovative applications have been produced to visualise scholarly editions online. Even if some of these applications are specifically meant to be used for different editions, it can be really complicated to reuse these products for different edition projects. In an article about the future of digital editions, Pierazzo wrote that “what is missing is the proposal of a scholarly model which is theoretically and editorially based” ([18]). In this article I analyse the theoretical basis and the technical implementations through which it could be possible to create a model to facilitate the development and reuse of applications for digital editions. The outcome of the analysis is the proposal of founding the model on the description of the different philological methodologies and then formalizing the model as an ontology, so that it is interoperable and applicable to different editions. Finally, I illustrate the further steps of the research that are necessary in order to transform the abstract model proposed

into an effective and pragmatic tool for the reuse and the generalization of applications for digital editions.

## Introduzione

Negli ultimi anni buona parte della ricerca nell'ambito delle *Digital Humanities* si è concentrata su come le edizioni digitali debbano essere presentate sul web<sup>1</sup>. Il risultato è stato un rapido e costante proliferare di software per la visualizzazione di edizioni digitali: alcuni nati in seno a specifici progetti, altri, invece, concepiti come strumenti general-purpose. Questi software, essendo complessi e costosi da mantenere, sono soggetti a una rapida obsolescenza e, al tempo stesso, il loro riutilizzo per diverse edizioni digitali è tutt'altro che scontato. L'attuale panorama delle edizioni digitali, da un punto di vista dello sviluppo software, è eterogeneo, complesso e non sostenibile.

Lo scopo del presente articolo è quello di mettere a fuoco il problema sopra descritto (sezione 1), individuandone le principali cause, e di proporre una soluzione che favorisca lo sviluppo di software per la visualizzazione di edizioni digitali che sia maggiormente riutilizzabile e sostenibile (sezione 2). L'articolo trae origine dall'analisi preliminare che ho condotto nel contesto del mio dottorato di ricerca, e pertanto si tratta di una proposta che necessita di essere elaborata ulteriormente.

Questo articolo si rivolge agli sviluppatori di software per le edizioni digitali, con la speranza di ricevere dei riscontri tecnici sulle difficoltà legate allo sviluppo di questo tipo di applicativi e su quale sia la soluzione più praticabile. Al contempo è rivolto ai filologi, "digitali" e non, con lo scopo di stimolare un dibattito sulle premesse teoriche su cui ho fondato la mia proposta di soluzione.

## 1. L'eterogeneo panorama delle edizioni digitali

Al giorno d'oggi la pubblicazione di edizioni scientifiche digitali è completamente affidata al web, sotto forma di siti o applicazioni. Questi applicativi possono essere "haute couture" o "prêt-à-porter" ([18]), ovvero possono essere stati creati *ad hoc* per un determinato progetto di edizione digitale oppure essere degli strumenti generici per la pubblicazione di più edizioni.

Negli anni sono stati realizzati sia strumenti generici, alcuni dei quali sono largamente adottati all'interno della comunità scientifica, sia applicativi per specifici progetti di ricerca. Gli applicativi

---

<sup>1</sup> Cfr.: «Visualization tools are so largely used in Digital Humanities that they can represent a distinctive mark of the discipline. The importance of a certain presentation of data is (or should be) particularly clear in the field of scholarly editions: textual scholars are used to devote attention to how, where and why a certain text is displayed on the page and in the book. Moreover, the shift from a printed book to a digital screen has stimulated reflections on how to present data» ([2]).

“prêt-à-porter” offrono svariati vantaggi rispetto a quelli “haute-couture”. Consentono di realizzare edizioni digitali a un costo minore e la cui manutenzione risulta più semplice sul lungo periodo, in quanto spesso il software viene aggiornato direttamente dai suoi sviluppatori. Tuttavia, un gran numero di edizioni digitali continua a essere realizzato attraverso software su misura.

Il continuo proliferare di applicativi “haute couture” è dovuto prima di tutto al desiderio da parte della comunità scientifica di sperimentare nuove metodologie filologiche e modalità di visualizzazione innovative. A ciò si aggiunge che spesso le alternative “prêt-à-porter” presenti sul mercato sono reputate non del tutto adeguate o soddisfacenti per il tipo di edizione che si intende condurre. Riguardo a questi strumenti Mancinelli e Pierazzo scrivono che: «agiscono come collane editoriali, vale a dire che offrono una formula comune con minimi margini di flessibilità» ([12]). Sono proprio i margini ridotti di flessibilità che spingono i team con cospicui finanziamenti a progettare nuovi applicativi. In merito agli applicativi “haute couture” Mancinelli e Pierazzo affermano:

le edizioni scientifiche che dispongono di ingenti finanziamenti di ricerca e che sviluppano il proprio software di gestione e di pubblicazione rappresentano in un certo senso un laboratorio di sperimentazione e innovazione, testando sul campo soluzioni innovative che potrebbero poi essere generalizzate a beneficio della comunità scientifica. ([12])

Tuttavia, il processo di generalizzazione di un applicativo per visualizzare edizioni digitali è di per sé complesso da un punto di vista tecnico e teorico. Questa complessità motiva come mai:

- le soluzioni innovative siano difficilmente riutilizzabili;
- le soluzioni generiche faticino ad adattarsi a diversi progetti di ricerca;
- il panorama delle edizioni digitali continui ad arricchirsi di esperimenti isolati, che non vengono poi mantenuti né riutilizzati.

Un consolidamento delle pratiche della filologia digitale è fondamentale per far fruttare le sperimentazioni condotte finora in un’ottica più sostenibile. Mancinelli e Pierazzo sostengono che il rapporto dialettico tra le due tipologie di edizione «può rappresentare infatti il meccanismo grazie al quale i prodotti di ricerca più avanzati si acquisiscono, si stabilizzano e diventano usufruibili da un numero elevato di ricercatori, dando allo stesso tempo un ruolo anche etico all’esistenza di una ricerca scientifica digitale costosa ed elitaria» ([12]). Il rapporto dialettico tra i due fronti dell’editing filologico, quello “haute-couture” e quello “prêt-à-porter”, per quanto sia in grado di stimolare la ricerca, non è però sufficiente per generalizzare le soluzioni informatiche esistenti.

### ***1.1 La generalizzazione degli applicativi per la visualizzazione di edizioni digitali***

Ogni applicativo è legato a doppio filo a un modello per la gestione dei dati al suo interno. Semplificando le pratiche dell’ingegneria del software, si può dire che in fase di progettazione gli sviluppatori stilano una specifica dei requisiti, analizzando lo schema di un formato di dati per

le edizioni digitali, primo fra tutti l'XML-TEI, oppure dialogando con gli editori dell'edizione digitale. A partire dalla specifica dei requisiti, gli sviluppatori elaborano un modello per la gestione dei dati, che viene utilizzato nel codice e riflesso nell'interfaccia grafica del software, determinando il modo in cui i dati vengono organizzati e presentati all'utente finale. Per cui ogni software dedicato alla visualizzazione di edizioni digitali dipende fortemente da un modello per la gestione dei dati, che può essere definito "modello di edizione".

A sua volta ogni edizione digitale è caratterizzata da un proprio modello, che può essere diviso su due piani:

1. un piano astratto, che comprende i componenti logici che costituiscono l'edizione e che solitamente è determinato dall'approccio filologico e dalle finalità scientifiche dell'editore. Per esempio, un'edizione critica prevede la presenza di componenti quali l'elenco dei manoscritti, dell'apparato critico per documentare le scelte editoriali, ecc.;
2. un piano più tecnico, che dipende dal formato in cui l'edizione digitale viene creata. Trattandosi per la maggior parte delle volte di edizioni codificate in XML-TEI, il modello dell'edizione viene espresso dalla codifica TEI.

Generalizzare o adattare uno strumento software a edizioni digitali diverse è un'operazione molto complicata a causa delle incompatibilità che si generano tra il modello di edizione sottostante l'applicativo e il modello delle singole edizioni.

I software confezionati su misura per un particolare progetto di edizione sono difficilmente riutilizzabili non solo per via della specificità del modello di edizione sottostante, ma anche perché molto spesso il modello deve essere evinto tramite l'analisi della codifica TEI dell'edizione oppure del codice del software (se disponibile in open source). La mancanza di una descrizione formale del modello di edizione si riscontra anche negli strumenti generici e costituisce un problema sia per il riuso degli applicativi sia per la loro manutenzione. Se per ogni strumento "prêt-à-porter" fosse disponibile il relativo modello di edizione, in una veste formale facilmente consultabile da altri sviluppatori informatici ed editori digitali, il software sarebbe più facilmente riutilizzabile. Inoltre gli editori e gli sviluppatori riuscirebbero a valutare meglio a priori l'adeguatezza di un determinato strumento per i propri scopi scientifici.

Anche la creazione di un software generico è impresa ardua, in quanto richiede di stabilire che tipo di dati l'applicativo necessita in input. Il vantaggio di basare il modello di edizione di un applicativo su un particolare formato è poter disporre di schemi e altri strumenti formali che guidino lo sviluppo su un piano logico. Anche se il modello di edizione di un applicativo generico, una volta plasmato su alcuni determinati formati, risulta più difficile da adattare a edizioni digitali allestite in formati diversi da quelli previsti. Nel caso in cui si tratti di uno strumento nato all'interno di un progetto e generalizzato in un secondo momento, c'è il rischio che il software mantenga delle rigidità, dovute all'essere ancorato a uno specifico modello di edizione. Qualora, invece, si tratti di uno strumento nato come generalista si verifica il problema opposto: il modello di edizione è troppo vasto e complesso da gestire all'interno del codice in modo efficace. Nel caso delle edizioni marcate in XML-TEI, occorre sottolineare come la TEI consenta un notevole grado di flessibilità, per cui uno stesso concetto o dato, può essere codificato

in modi diversi. Si consideri, per esempio, l'apparato critico di un'edizione. L'apparato può essere marcato mediante il metodo della parallel-segmentation all'interno del testo critico oppure, giusto per fornire un esempio di marcatura alternativa, in un apposito elemento <listApp>, annidato nell'elemento <back>, e collegato al testo critico attraverso il metodo del double-end-point-attachment.<sup>2</sup> Perché uno strumento generico si adegui al maggior numero di edizioni in TEI possibile, il modello di edizione sottostante dovrà essere progettato tenendo conto di tutte le marcature possibili, o perlomeno di quelle più comuni.

Per risolvere le rigidità e i limiti degli strumenti generici, è spesso necessario un intervento diretto sul codice sorgente, operazione che richiede approfondite competenze informatiche. Diversi progetti di edizione continuano a prediligere lo sviluppo di soluzioni informatiche su misura, a causa della difficile personalizzazione degli applicativi generici. Gli editori, infatti, necessitano di un pieno controllo sul modo in cui la propria edizione viene visualizzata sul web, perché la resa grafica influisce sulla fruibilità e sulla qualità dell'edizione stessa.

Al momento, nel campo della filologia digitale non ci sono dei modelli che guidino lo sviluppo di applicativi per la visualizzazione. Per questo motivo, ogni volta che ci si appresta a realizzare un software, generico o specifico, è necessario progettare sostanzialmente da zero l'architettura e l'interfaccia grafica. Per realizzare da sé l'applicazione o per fornire in modo strutturato delle indicazioni sullo sviluppo al supporto informatico, sono necessarie delle competenze che esulano dalla formazione tradizionale del filologo. La realizzazione e la manutenzione nel tempo di un applicativo risulterebbero meno costose se gli sviluppatori potessero contare su dei modelli di edizione pronti all'uso, che siano fondati su una riflessione teorica condivisa dalla comunità scientifica.

I principali problemi legati alla generalizzazione possono essere sintetizzati nel seguente modo:

- gli applicativi sono ancorati a dei modelli per la gestione dei dati, che costituiscono dei “modelli di edizione”;
- questi modelli, la maggior parte delle volte, rimangono impliciti all'interno del codice del software;
- molto spesso ci sono delle discrepanze, delle incompatibilità tra i modelli di edizione degli applicativi e i modelli delle edizioni;
- i modelli di edizione vengono prodotti da zero ogni volta che si realizza un software per le edizioni digitali, in quanto non ci sono modelli condivisi.

### ***1.2 EVT: un esempio di applicativo generalizzato***

[EVT](#) (Edition Visualization Technology) è un software open source per la visualizzazione di edizioni digitali in XML-TEI, sviluppato da un team di studenti dell'Università di Pisa guidato da Roberto Rosselli Del Turco. La prima versione del software, EVT 1, ideata per poter pubblicare l'edizione del [Vercelli Book](#), è stata trasformata in un secondo momento in uno

---

<sup>2</sup> Si rimanda al modulo dedicato all'apparato critico delle [linee guida TEI](#).

strumento general-purpose per la visualizzazione di edizioni diplomatiche. Successivamente è stata sviluppata EVT 2 che, rispetto alla prima versione, è stata progettata *ad hoc* per le edizioni critiche ed è stata concepita fin dall’inizio come strumento generico.<sup>3</sup> EVT 2 può essere configurata dai propri utenti-editor<sup>4</sup> mediante un foglio di stile CSS, che consente di aggiungere regole personalizzate, e un file JSON. Questo secondo file contiene un unico oggetto JSON, in cui ogni proprietà è associata a un parametro di configurazione dell’interfaccia grafica o del modo in cui le funzioni di parsing dell’applicativo recuperano i dati dai file TEI in input.

Nel Frammento di codice 1 si forniscono degli esempi di parametri che l’utente-editor può impostare per modificare l’interfaccia grafica del software. In EVT i dati di base dell’edizione (il testo critico, le immagini dei manoscritti, le trascrizioni dei manoscritti, ecc.) vengono organizzati in “view” (viste) e, al caricamento dell’applicazione, viene inizializzata una determinata view. L’utente-editor può selezionare quale view caricare di default, modificando il valore della proprietà “defaultViewMode”, e quali view mettere a disposizione del lettore, impostandone la visibilità su true o false all’interno della proprietà “availableViewModes”.

```
"defaultViewMode": "readingTxt",  
"availableViewModes": [  
  {  
    "label": "READING_TEXT",  
    "icon": "reading-txt",
```

Frammento di codice 1: esempio di parametri di configurazione di EVT 2 che determinano l’aspetto dell’interfaccia grafica.

Le view impostate come visibili nel file di configurazione corrispondono a quelle attivabili con i rispettivi bottoni presenti nella barra in alto a destra (evidenziate in rosso nella Figura 1).

---

<sup>3</sup> Per la storia del software EVT si rimanda a 21. Al momento è in corso di sviluppo una terza versione del software, EVT 3. La realizzazione di una nuova versione è stata indotta dalla ricerca di una maggiore flessibilità e modularità del codice, nonché dal passaggio da AngularJS ad Angular come framework per lo sviluppo.

<sup>4</sup> “Utente-editor” è una definizione coniata dal team di sviluppatori di EVT per distinguere gli utenti che si servono del software per visualizzare e pubblicare online un’edizione digitale dagli utenti che consultano un’edizione digitale pubblicata con EVT.

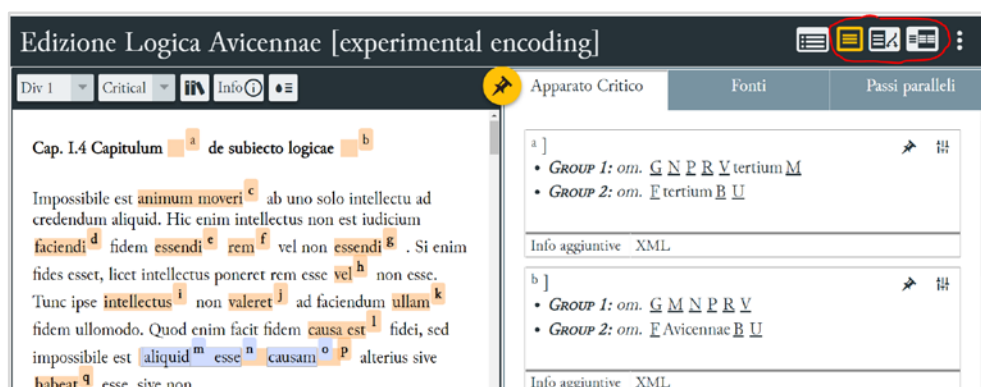


Figura 1: screenshot dell'interfaccia di EVT 2 in cui compaiono evidenziati i bottoni per attivare le view disponibili.

Come accennato, nel file JSON possono essere configurate anche delle proprietà che determinano il modo in cui le funzioni di parsing del software recuperano i dati dai file TEI. Più precisamente, queste proprietà servono a mappare alcuni elementi dell'edizione digitale, di cui EVT può gestire la presentazione grafica e l'interazione, con il modo in cui sono stati marcati in TEI. Nel Frammento di codice 2, si riporta come esempio il parametro "notSignificantVariant", che consente all'utente-editor di esplicitare come ha marcato le varianti ortografiche o, in generale, non significative, così che i parser possano riconoscerle e classificarle come tali.

```
"notSignificantVariant": "<orig>, <sic>, [type=orthographic]"
```

Frammento di codice 2: esempio di parametro di configurazione di EVT 2 che determina il modo in cui i parser recuperano i dati dai file TEI.

Per la presente analisi l'aspetto più interessante di questo sistema di configurazione è che l'oggetto JSON è, in un certo senso, una formalizzazione del modello di edizione di EVT. Non si tratta di una formalizzazione completa, in quanto nell'oggetto JSON mancano delle proprietà che corrispondono ad altri componenti dell'interfaccia grafica e ad altri elementi interattivi dell'edizione. Tuttavia, un sistema di generalizzazione simile può aiutare a superare le incompatibilità tra il modello di edizione del software e il modello dell'edizione digitale espresso dalla codifica TEI, creando una mappatura personalizzata del modo in cui i dati devono essere recuperati.

L'altro aspetto da evidenziare è che il modello di edizione di EVT presenta ad ogni modo i limiti delineati nella sezione precedente: non è stato formalizzato in una veste facilmente condivisibile e riutilizzabile; non è completo, per quanto sia stato elaborato sulla base delle linee guida TEI, studiando le pratiche di marcatura più diffuse e mediante la collaborazione con diversi progetti di edizioni, che negli anni hanno fornito delle codifiche di prova oppure hanno sviluppato nuove funzionalità del codice.

## 2. Un modello concettuale di edizione condiviso

Una possibile soluzione alla difficoltà di generalizzare e riutilizzare applicativi per edizioni digitali è quella di creare un modello concettuale di edizione, che possa essere preso a riferimento sia dagli sviluppatori sia dagli editori. In altre parole, un modello che da un lato sia a disposizione degli sviluppatori per progettare i propri software e dall'altro sia in grado di rappresentare le pratiche filologiche e le esigenze scientifiche degli editori.

Si consideri, come esempio, il sistema di configurazione del software EVT descritto nella sezione precedente e la definizione da parte dei creatori del software del parametro "notSignificantVariant". Se fosse disponibile un modello concettuale condiviso, cui poter fare riferimento, il parametro potrebbe essere associato direttamente a un elemento del modello concettuale. Al contempo gli editori potrebbero marcare liberamente le edizioni in XML-TEI, modellandole, su un piano logico, sulla base del modello concettuale.

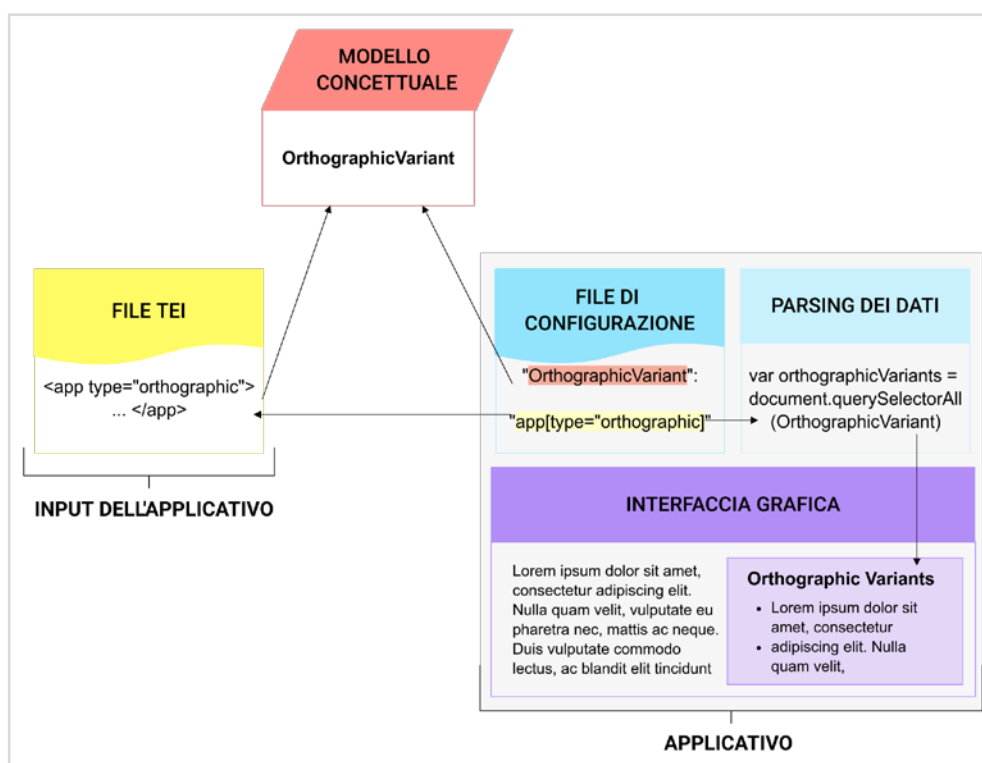


Figura 2: schema del rapporto tra il modello concettuale, le edizioni digitali e gli strumenti di visualizzazione.

Lo scopo di questo modello, dunque, è di fungere da ponte tra le edizioni digitali e gli strumenti di visualizzazione, a prescindere dai formati delle prime e dalle specificità tecniche dei secondi. Perché ciò sia possibile, il modello deve essere: a) esplicito, b) condivisibile e c) in grado di



adattarsi a più edizioni di diverso tipo. Ciò significa che il modello deve essere descritto formalmente con linguaggi che consentano l'interoperabilità tra sistemi e tecnologie diverse, e che al tempo stesso non deve vincolare la scelta del formato dei dati di partenza per l'allestimento delle edizioni digitali. In secondo luogo, come sostiene Pierazzo: «what is missing is the proposal of a scholarly model which is theoretically and editorially based» ([18]), per cui il modello deve essere fondato su una riflessione teorica sulla pratica filologica, affinché si adatti a più edizioni e a differenti tipologie di edizione.

Lo sviluppo di un simile modello deve essere condotto su due livelli:

- un primo livello concettuale, che funga da base teorica e che si concentri sulla descrizione del concetto di edizione;
- un secondo livello formale, che sia la traduzione pragmatica del livello concettuale, e che funga da guida nello sviluppo di applicativi per le edizioni digitali.

La suddivisione in due livelli è stata pensata anche per garantire una maggiore longevità del modello. Il livello concettuale, se ben studiato, potrebbe rimanere valido ed efficace nel tempo. Il livello pragmatico, invece, può essere declinato di volta in volta in forme diverse, a seconda delle esigenze tecniche e delle tendenze nel campo dello sviluppo e della progettazione web.

### ***2.1 I diversi approcci filologici come base concettuale del modello***

Ogni edizione è frutto di un'operazione di modellazione che il filologo conduce a partire dai materiali e dai contenuti di cui dispone. Si tratta di un'operazione che viene condotta seguendo uno dei diversi approcci filologici che si sono consolidati negli anni. Un'edizione critica, per esempio, è diversa negli intenti da un'edizione diplomatica: se la prima prevede come principale obiettivo la ricostruzione del testo in una veste il più simile possibile all'originale, la seconda, invece, è contraddistinta da una finalità più documentaria. L'applicazione di un certo approccio filologico è chiaramente flessibile e adattabile sui contenuti dell'edizione, ma funge da denominatore comune tra edizioni diverse. Per questo motivo si propone di costruire il modello concettuale basandosi sui diversi approcci filologici che sono stati ideati dall'alba della filologia a oggi.<sup>5</sup>

Più concretamente, nel modello concettuale per ogni approccio filologico dovrebbero essere inserite le definizioni:

1. degli elementi individuati all'interno dei materiali di partenza;
2. delle operazioni del lavoro filologico;
3. degli attori del lavoro filologico.

---

<sup>5</sup> Cfr. «In order to be able to offer a stable infrastructure for digital editions it is therefore necessary to reflect on the features that the scholarly community will consider essential to a particular type of text or scholarly problem and to agree on some essential models which take into account the new affordances offered by the digital» ([18]).

Si prenda per esempio un modello che rappresenti le edizioni prodotte con il metodo (neo-)lachmanniano. Nel modello sarebbero contenuti i concetti di “testimone” e “variante”, che rientrano nella categoria 1, e i concetti di “restitutio textus” e “curatore”, che rientrano rispettivamente nelle categorie 2 e 3.

Il modello concettuale potrebbe comprendere anche una quarta categoria di componenti, ovvero gli elementi che vengono prodotti dal lavoro filologico. Proseguendo l'esempio sull'approccio (neo-)lachmanniano, in questa categoria rientrerebbe il concetto di “apparato critico”. L'inclusione nel modello di questa quarta categoria di elementi è da valutare, in quanto si tratta di elementi la cui definizione dipende fortemente dal modo in cui vengono presentati al lettore. Nelle edizioni a stampa l'apparato critico viene allestito mediante un ricco insieme di abbreviazioni e di soluzioni tipografiche, che ne condensano il significato, per via dei costi di stampa e dei limitati spazi disponibili. In un'edizione digitale queste soluzioni tipografiche non sono necessarie, per cui nel modello non conviene includere il concetto di apparato critico in una forma legata alla sua forma tradizionale. In generale, l'inserimento nel modello concettuale della quarta categoria di elementi rischierebbe di ancorare il modello al paradigma dell'edizione a stampa ([19]) e di limitare la possibilità di applicare il modello anche a visualizzazioni innovative. Concentrandosi sulle operazioni e sugli scopi scientifici dell'ecdotica, invece che sulla resa finale dell'edizione, è possibile astrarre e modellare il concetto di edizione in un modo che potrebbe adattarsi tanto alle edizioni stampate quanto agli applicativi web. Per concludere l'esempio, al posto dell'apparato critico si possono modellare due operazioni, o meglio, due obiettivi a esso equivalenti: documentare il lavoro filologico svolto e offrire al lettore un rapido confronto tra le lezioni alternative ritenute più interessanti. Per cui si potrebbero aggiungere al modello gli obiettivi del lavoro filologico invece dei prodotti.

Tra gli approcci filologici da tenere in considerazione ci sono *in primis* quelli “tradizionali”, come appunto il metodo lachmanniano, bedieriano, ecc. Il passaggio dalla carta al web non ha completamente rivoluzionato le metodologie filologiche. Sul web vengono pubblicate tante edizioni, definite dai rispettivi curatori in modo tradizionale come “edizione critica”, “edizione diplomatica”, ecc. Inoltre, le metodologie filologiche tradizionali fungono da riferimento teorico per valutare la qualità anche delle edizioni pubblicate online. A tal proposito è rappresentativa l'iniziativa dell'Osservatorio Permanente sulle Edizioni Digitali di autori Italiani (OPEDIt), ideata da Michelangelo Zaccarello, che tra i propri principali obiettivi ha proprio quello “di assicurare che la metodologia dell'ecdotica tradizionale sia tenuta in considerazione anche nel caso delle edizioni digitali o digitalizzate presenti in rete” ([23]).

Il modello concettuale per essere completo, però, deve anche tener conto delle metodologie nate negli ultimi anni dall'incontro della filologia con le nuove tecnologie. In più il modello deve essere espandibile, in modo da poter includere anche le nuove metodologie che potrebbero essere sviluppate in futuro. In altre parole, il modello deve costituire un punto di partenza e non di arrivo per la comunità scientifica. Da un lato deve garantire una continuità con il passato, dall'altro non deve porre limiti all'innovazione, ma incoraggiare un consolidamento e una valorizzazione di quanto viene sperimentato con le nuove tecnologie.

## ***2.2 Una formalizzazione object-oriented del modello***

Gli strumenti informatici per la visualizzazione di edizioni digitali, oltre a dividersi tra generici e personalizzati, possono essere distinti in base a due diversi approcci al modo in cui trattano ed elaborano le edizioni da visualizzare. Un approccio è più conservativo, cioè mira a preservare il più possibile la struttura originale del documento (o dei molteplici documenti) in input, concentrandosi per lo più sull'aggiunta di regole di stile che modifichino l'aspetto grafico e di funzioni per rendere interattiva la consultazione. Il vantaggio di questo approccio è che solitamente permette di realizzare strumenti con linguaggi standard, senza dipendere da librerie esterne, che complicano la manutenzione a lungo termine del software. Un esempio di strumento conservativo per lo sviluppo di software è [CETELcean](#). Si tratta di una libreria JavaScript che, come TEIBoilerplate, offre la possibilità di visualizzare le edizioni marcate in TEI in una forma molto simile all'originale, trasformando però gli elementi XML-TEI in [Custom Elements HTML](#). Lo sviluppatore può personalizzare il risultato finale aggiungendo regole di stile e funzioni JavaScript *ad hoc*.

Il secondo approccio, invece, è più trasformativo e prevede la realizzazione di interfacce web in cui i dati dell'edizione digitale vengono presentati in modi che si allontanano di molto dal formato-documento. L'approccio trasformativo viene adottato sia da strumenti "haute-couture" sia da quelli generici, quali EVT, in quanto consente di sperimentare nuove modalità di visualizzazione.

Da un punto di vista architetturale i software "trasformativi" tendono a essere molto complessi. Nel campo dell'ingegneria del software, quando un applicativo deve rappresentare un dominio (un settore della realtà) particolarmente vasto e articolato, come quello della critica del testo, si tende ad applicare il paradigma di programmazione orientato agli oggetti. Mediante la creazione di classi e oggetti, il codice risulta più robusto e le sue parti possono essere riutilizzate.

All'interno di software conservativi l'utilizzo di un modello concettuale che faccia da ponte con le edizioni digitali rischia di essere ridondante. Per gli scopi di questa categoria di applicativi, infatti, il modello espresso dalla codifica TEI costituisce di per sé un modello di edizione sufficiente per una efficace progettazione dell'applicativo. Lo sviluppo dei software trasformativi, invece, può essere facilitato dall'utilizzo di un modello di riferimento che sia "orientato agli oggetti": ovvero (1) che sia costituito da un insieme di componenti astratti, (2) che fornisca per ciascuno una descrizione dei requisiti funzionali ed (3) espliciti i rapporti reciproci tra i componenti.

## **3. Domande aperte e possibili sviluppi di ricerca**

La principale domanda di ricerca che al momento rimane aperta è la seguente: in che forma e con quale linguaggio deve essere elaborato il livello pragmatico del modello?

Lo scopo principale del modello sarebbe idealmente quello di poter essere integrato nelle procedure di sviluppo software. Al contempo, è fondamentale che il modello sia accessibile anche

agli editori, digitali e non: il modello, infatti, deve essere usato per inquadrare su un piano logico/organizzativo le edizioni, altrimenti la sua efficacia come “ponte” è compromessa.

In un intervento dal titolo “The importance of being... object-oriented” ([6]), Del Grosso, Giovannetti e Marchi presentano un progetto di modello per la critica del testo, che ben si sposa alle premesse teoriche della presente ricerca, e che ha lo scopo di sopperire alla mancanza di specifiche formali, di modelli astratti condivisi e di piani per il riuso di software nel campo della filologia digitale ([10]). Il progetto consiste sostanzialmente nell’individuare e definire Domain Specific Abstract Data Types (DS-ADTs), ovvero delle categorie astratte per determinare i dati all’interno dell’applicativo. La creazione di tipi astratti favorisce lo sviluppo di componenti software che possono essere riutilizzati e condivisi anche mediante API (Application Programming Interface). La proposta di Del Grosso et al. è molto promettente su un piano di sviluppo software. L’unico difetto è che questi tipi di dati sono pensati per rimanere nascosti all’utente del software, quando invece il modello dovrebbe essere condivisibile anche con chi realizza le edizioni.

A mio avviso si profilano due possibili strategie per la formalizzazione del modello. La prima è quella di seguire la direzione fornita in Del Grosso et al. ([6]), ovvero individuare e descrivere degli elementi astratti, che possano essere utilizzati per lo sviluppo del software e che siano presentati in modo schematico, per esempio in UML. Questa soluzione, però, deve prevedere anche la stesura di linee guida o raccomandazioni, che descrivano dettagliatamente i tipi astratti individuati (e che corrisponderebbero al piano concettuale del modello), fornendo agli editori anche degli esempi di come questi tipi possono essere applicati in diversi contesti di studio e di editing filologico.

La seconda strategia è quella di creare un’ontologia, che più facilmente risulterebbe comprensibile sia agli sviluppatori sia agli editori e che inoltre porterebbe a degli sviluppi di ricerca molto interessanti. Rimane, però, da valutare quanto un’ontologia sia effettivamente usabile per gli sviluppatori e se sia facilmente integrabile o meno nel flusso di lavoro per lo sviluppo di software.

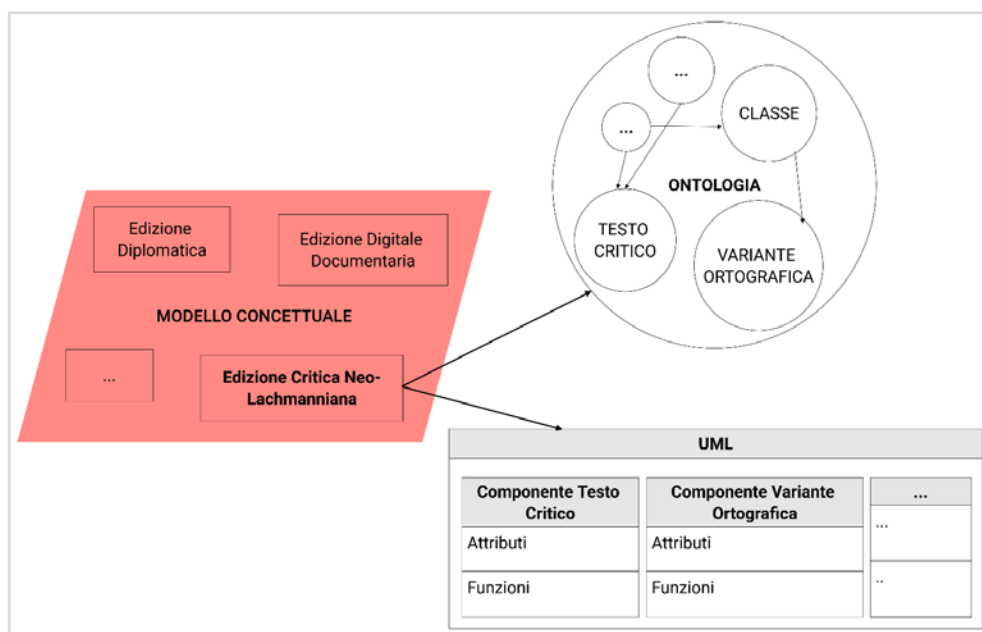


Figura 3: schema delle possibili strategie di formalizzazione del modello concettuale.

Per quanto molti aspetti della proposta delineata rimangano da valutare ulteriormente, la suddivisione dell'elaborazione del modello su due piani fa sì che il livello pragmatico possa essere concretizzato in svariate forme, per perseguire diversi obiettivi scientifici. Tra i più interessanti, uno è quello di creare un quadro di riferimento comune per valutare la qualità delle edizioni digitali che vengono pubblicate in rete. Il modello rappresenterebbe una sintesi schematica della pratica filologica e delle metodologie che si sono consolidate nell'ambito delle edizioni digitali. Un secondo possibile obiettivo, direttamente collegato al precedente, consisterebbe nel categorizzare gli strumenti di visualizzazione in base alla loro conformità con il modello. Uno studioso disporrebbe di un riferimento preciso per valutare a priori l'adeguatezza degli strumenti disponibili e la loro compatibilità con le proprie edizioni.

### ***3.1 Vantaggi e limiti di un'ontologia per la formalizzazione del modello***

In linguaggio informatico un'ontologia è la rappresentazione di un dominio della realtà mediante un insieme di classi e proprietà. Come affermato da Ciotti e Tomasi, «the practice of ontological modeling is a good operationalist translation of the common definition of model» ([4]). Un modello espresso sotto forma di ontologia potrebbe risultare chiaro a chi abbia familiarità con le prassi dell'ecdotica, soprattutto nell'esplicitare le relazioni tra le diverse parti del modello.

Le ontologie possiedono altre caratteristiche interessanti:

- sono tecnologie consolidate, per le quali esistono linguaggi standard supportati dal W3C;<sup>6</sup>
- un'ontologia, per poter essere usata, non è necessario che descriva il dominio di pertinenza in modo esaustivo. L'ontologia può essere incompleta e venire espansa in futuro. Questa proprietà permette di usare la stessa tecnologia per modellare le edizioni digitali su livelli di astrazione diversi<sup>7</sup> e, come affermato da Ciotti e Tomasi, «is very helpful when the domain is very complex and subject to different points of view, and its modeling is conceived as a work in progress» ([4]);
- ontologie diverse possono essere unite insieme. Questo consente da un lato il riuso di ontologie esistenti,<sup>8</sup> dall'altro un approccio di modellazione collaborativo. Il campo della filologia digitale è quanto mai complesso e destinato a cambiamenti nei prossimi anni, per cui la scelta di un approccio di modellazione flessibile e adattabile sul lungo periodo è fondamentale.

L'utilizzo delle ontologie, inoltre, permetterebbe alla presente ricerca di essere espansa nella direzione dei linked open data, dell'utilizzo di vocabolari quali [Schema.org](https://schema.org/) e di altri strumenti del web semantico. Un'ontologia simile, permetterebbe di marcare semanticamente le edizioni digitali, sia a livello di codifica TEI che a livello di allestimento delle pagine in HTML, e di effettuare delle ricerche incrociate sui dati derivanti da diverse edizioni digitali pubblicate in rete. Una funzionalità simile sarebbe molto utile per effettuare studi che beneficiano di vasti corpora di dati, come per esempio lo studio delle concordanze.

L'aspetto che rimane da valutare è se un modello espresso sotto forma di ontologia possa essere integrato in modo efficace nel procedimento di sviluppo di software per le edizioni digitali. Le premesse teoriche per condurre delle sperimentazioni in questo senso ci sono. Nel campo dell'ingegneria del software, infatti, è stato ideato un approccio di sviluppo detto *Ontology-Driven Software Development (ODSD)*, che ha come obiettivo l'integrazione delle ontologie nelle diverse fasi di produzione del software. L'idea alla base di questo paradigma è quella di sfruttare la potenza espressiva e le funzioni di reasoning automatico delle ontologie sia in fase di modellazione che in fase di scrittura del codice.

Ontologies allow for defining a shared “language” about the domain for which we develop some software. Ontologies also offer a formal representation about relationships between concepts of our domain language. This way we automatically check if our “language” is

---

<sup>6</sup> Si rimanda alla [pagina web](#) del W3C.

<sup>7</sup> Le ontologie possono essere usate per modellare i diversi tipi di dati che riguardano un'edizione. Oltre alla metodologia filologica con cui l'edizione viene prodotta, con un'ontologia si possono modellare, per esempio, i riferimenti a luoghi e persone presenti all'interno del testo.

<sup>8</sup> L'ontologia “[Scholarly Editing](#)” è un esempio di ontologia esistente che descrive le pratiche della filologia per la realizzazione di edizioni e che, eventualmente, costituirebbe un punto di partenza per la creazione dell'ontologia del modello descritto.

consistently used and if all our sentences when we talk to other humans or computers are correctly written. ([15])

Tuttavia, il lavoro informatico richiesto per questo tipo di implementazione software richiede delle competenze molto specifiche e avanzate. Competenze che all'interno di un progetto di edizione digitale necessiterebbero dell'intervento di un supporto informatico specialistico. L'applicazione di questo approccio potrebbe, inoltre, essere troppo specifico e vincolante per sviluppatori che sono abituati a lavorare seguendo altre metodologie dell'ingegneria del software. Infine, dalla ricognizione dello stato dell'arte che ho condotto, non è emersa alcuna sperimentazione esistente sull'applicazione dell'ODSD per lo sviluppo di software per le edizioni digitali.

## Conclusioni

Il problema messo a fuoco è stato dibattuto negli ultimi anni dalla comunità scientifica, per cui non stupisce che alcuni studiosi e sviluppatori informatici abbiano già elaborato dei tentativi di soluzione. Tuttavia, l'impressione che si ricava da un'analisi delle sperimentazioni condotte finora è che, nell'ambito della sostenibilità dei software di visualizzazione, tardino ad arrivare degli effettivi progressi. Ci sono diversi fattori che influiscono sul raggiungimento di soluzioni concrete e percorribili. Prima di tutto la complessità tecnica: l'applicazione di soluzioni richiede delle solide competenze sia informatiche sia filologiche, che non sono affatto scontate. Questa difficoltà potrebbe essere ovviata all'interno di progetti di edizione mediante un approccio fortemente interdisciplinare e servendosi anche di figure ibride, che possiedano entrambe le tipologie di competenze. Un secondo fattore è la difficoltà di realizzazione di un modello che soddisfi i requisiti scientifici dell'editing filologico. Il coinvolgimento dei filologi, "digitali" e non, è fondamentale per ideare delle soluzioni tecniche che siano fondate su delle solide basi teoriche.

Il modello concettuale presentato nell'articolo vuole essere sì una proposta di soluzione, ma anche e soprattutto un'occasione di dibattito su come si possa elaborare un modello "theoretically and editorially based" ([18]). Ci sono diversi aspetti del modello e del presente progetto di ricerca che vanno elaborati ulteriormente. La prima domanda, cui occorrerà rispondere, è chi deve realizzare il modello. Trattandosi di un modello ambizioso per le dimensioni del dominio che vuole rappresentare, potrebbe essere più corretto immaginare, invece di un unico modello generale, più modelli generati con la stessa metodologia. L'operazione di modellazione potrebbe essere condotta collaborativamente da diversi gruppi di studiosi, sfruttando eventualmente anche le proprietà delle ontologie. Ogni progetto di edizione digitale contribuirebbe alla definizione di una particolare metodologia filologica, sulla base delle proprie sperimentazioni e dell'esperienza nel proprio campo di studio. La ricerca, dunque, dovrà mettere a punto una strategia condivisa, che consenta la modellazione dei diversi approcci filologici e che possa essere applicata dalla comunità scientifica. Per riuscire a mettere a fuoco quanto il modello concettuale proposto possa essere una soluzione efficace, sarà necessario da un lato un confronto con le esperienze di sviluppo

software di altri progetti di edizione digitale, dall'altro avviare una fase sperimentale della ricerca, realizzando un primo modello concettuale per le edizioni critiche di stampo (neo-)lachmanniano.

## References

- [1] «5-Star Open Data». s.d. Consultato 11 gennaio 2021. <http://5stardata.info/en/>.
- [2] Barabucci, Gioele, Elena Spadini, e Magdalena Turska. 2017. «Data vs. Presentation. What Is the Core of a Scholarly Digital Edition?» In *Advances in Digital Scholarly Editing*, 37–46. Sidestone Press. [https://serval.unil.ch/notice/serval:BIB\\_09C6C598108A](https://serval.unil.ch/notice/serval:BIB_09C6C598108A).
- [3] Ciotti, Fabio. 2018. «A Formal Ontology for the Text Encoding Initiative». *Umanistica Digitale*, n. 3 (novembre). <https://doi.org/10.6092/issn.2532-8816/8174>.
- [4] Ciotti, Fabio, e Francesca Tomasi. 2016. «Formal Ontologies, Linked Data, and TEI Semantics». *Journal of the Text Encoding Initiative*, n. Issue 9 (settembre). <https://doi.org/10.4000/jtei.1480>.
- [5] Ciula, Arianna, e Øyvind Eide. 2014. «Reflections on cultural heritage and digital humanities: modelling in practice and theory». In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, 35–41. DATeCH '14. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2595188.2595207>.
- [6] Del Grosso, Angelo Mario, Emiliano Giovannetti, e Simone Marchi. 2017. «The Importance of Being... Object-Oriented: Old Means for New Perspectives in Digital Textual Scholarship». *Advances in Digital Scholarly Editing* 1: 269–74.
- [7] Doerr, Martin. 2003. «The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata». *AI Magazine* 24 (3): 75–75. <https://doi.org/10.1609/aimag.v24i3.1720>.
- [8] Eide, Øyvind. 2014. «Ontologies, Data Modeling, and TEI». *Journal of the Text Encoding Initiative*, n. Issue 8 (dicembre). <https://doi.org/10.4000/jtei.1191>.
- [9] Gonzalez-Perez, Cesar, e Patricia Martín-Rodilla. 2017. «Teaching Conceptual Modelling in Cultural Heritage». *Revista de Humanidades Digitales* 1: 408. <https://doi.org/10.5944/rhd.vol.1.2017.16128>.
- [10] Guarino, Nicola, Daniel Oberle, e Steffen Staab. 2009. «What Is an Ontology?» In *Handbook on Ontologies*, a cura di Steffen Staab e Rudi Studer, 1–17. International Handbooks on Information Systems. Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-540-92673-3\\_0](https://doi.org/10.1007/978-3-540-92673-3_0).
- [11] Ionov, Christian Chiarcos and Max. s.d. «Linking the TEI: Approaches, Limitations, Use Cases». Text. Consultato 16 dicembre 2020. <https://dev.clariah.nl/files/dh2019/boa/0910.html>.



- [12] Mancinelli, Tiziana, e Elena Pierazzo. 2020. *Che cos'è un'edizione scientifica digitale*. Carocci.
- [13] Nussbaumer, Philipp, e Bernhard Haslhofer. 2007. «Putting the CIDOC CRM into Practice - Experiences and Challenges». Technical Report. University of Vienna. settembre 2007. <https://eprints.cs.univie.ac.at/404/>.
- [14] Ore, Christian-Emil, e Øyvind Eide. 2009. «TEI and cultural heritage ontologies: Exchange of information?» *Literary and Linguistic Computing* 24 (2): 161–72. <https://doi.org/10.1093/lc/fqp010>.
- [15] Pan, Jeff Z., Steffen Staab, Uwe Aßmann, Jürgen Ebert, e Yuting Zhao, a c. di. 2013. *Ontology-Driven Software Development*. Berlin Heidelberg: Springer-Verlag. <https://doi.org/10.1007/978-3-642-31226-7>.
- [16] Peroni, Silvio, David Shotton, e Fabio Vitali. 2012. «Scholarly publishing and linked data: describing roles, statuses, temporal and contextual extents». In *Proceedings of the 8th International Conference on Semantic Systems*, 9–16. I-SEMANTICS '12. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2362499.2362502>.
- [17] Pierazzo, Elena. 2014. *Digital Scholarly Editing: Theories, Models and Methods*. <http://hal.univ-grenoble-alpes.fr/hal-01182162>.
- [18] Pierazzo, Elena. 2019. «What Future for Digital Scholarly Editions? From Haute Couture to Prêt-à-Porter». *International Journal of Digital Humanities*, maggio. <https://doi.org/10.1007/s42803-019-00019-3>.
- [19] Sahle, Patrick. 2016. «What is a Scholarly Digital Edition?» In *Digital Scholarly Editing*, a cura di Matthew James Driscoll e Elena Pierazzo, 1ª ed., 4:19–40. Theories and Practices. Open Book Publishers. <https://www.jstor.org/stable/j.ctt1fzhh6v.6>.
- [20] Turco, Roberto Rosselli Del, Giancarlo Buomprisco, Chiara Di Pietro, Julia Kenny, Raffaele Masotti, e Jacopo Pugliese. 2014. «Edition Visualization Technology: A Simple Tool to Visualize TEI-Based Digital Editions». *Journal of the Text Encoding Initiative*, n. Issue 8 (dicembre). <https://doi.org/10.4000/jtei.1077>.
- [21] Turco, Roberto Rosselli Del, Chiara Di Pietro, e Chiara Martignano. 2019. «Progettazione e implementazione di nuove funzionalità per EVT 2: lo stato attuale dello sviluppo». *Umanistica Digitale* 3 (7). <https://doi.org/10.6092/issn.2532-8816/9322>.
- [22] Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. «The FAIR Guiding Principles for Scientific Data Management and Stewardship». *Scientific Data* 3 (1): 160018. <https://doi.org/10.1038/sdata.2016.18>.
- [23] Zaccarello, Michelangelo. 2018. «Progetto Di Un Osservatorio Permanente Sulle Edizioni Digitali Di Autori Italiani (OPEDIt). Prime Indagini Sulle Pratiche Di Digitalizzazione e Sull'autorevolezza Dell'edizione Di Testi Letterari Italiani in Formato Elettronico - BOLOGNA». *Prassi Ecdotiche Della Modernità Letteraria*, n. 3. <https://doi.org/10.13130/2499-6637/9491>.

