



UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

Il *Flos* di Leonardo Pisano:

edizione elettronica con note matematiche

Candidato: *Alessandra Giammalva*

Relatore: *prof. Enrica Salvatori*

Co-Relatore: *prof. Roberto Rosselli Del Turco*

Anno Accademico 2007-2008

INDICE

INTRODUZIONE	4
IL CODICE E75 P. SUP.....	5
1.1 - Storia del manoscritto	5
1.2 - Descrizione fisica	6
1.3 - Descrizione del contenuto	8
1.4 - Vita e opere di Leonardo Pisano	10
ELABORAZIONE DEL PROGETTO	15
2.1 - Fasi preliminari	15
2.1.1 - <i>Acquisizione del materiale</i>	15
2.1.2 - <i>Stesura del percorso</i>	16
2.2 - La codifica del testo	16
2.2.1 - <i>Che cosa è XML</i>	17
2.2.2 - <i>Che cosa è XSL</i>	18
2.2.3 - <i>La TEI</i>	21
2.2.5 - <i>La codifica del testo</i>	24
2.3 - Le note ai problemi.....	34
2.3.1 - <i>Creazione delle note</i>	34
2.3.2 - <i>Visualizzazione dei problemi</i>	37
2.4 - EVT.....	38
2.4.1 - <i>Che cosa è EVT</i>	38
2.4.2 - <i>Caricamento dei file in EVT</i>	39
CONCLUSIONI	46

BIBLIOGRAFIA	47
SITOGRAFIA	48
Articoli e notizie storiografiche	48
Codifica ed esempi di annotazione di testi antichi...	50
Software	51

INTRODUZIONE

Ho scritto questa tesi per illustrare i passaggi che mi hanno condotto alla realizzazione dell'edizione elettronica del *Flos* di Leonardo Pisano, detto Fibonacci. Il *Flos* (il fiore) è un opuscolo contenente le soluzioni ai quesiti che il matematico della corte di Federico II, maestro Giovanni da Palermo, aveva posto a Fibonacci durante un loro incontro a Pisa, all'incirca nel 1225.

Il testo dell'opuscolo è oggi conservato in un unico manoscritto del XV secolo, archiviato presso la Veneranda Biblioteca Ambrosiana di Milano, e catalogato come E75 P. Sup. (parte superiore).

Nel primo capitolo della tesi fornirò una descrizione dettagliata del manoscritto e la ricostruzione della biografia dell'autore. Passerò quindi nel secondo capitolo ad illustrare le varie fasi del progetto, introducendo, dove necessario, delle note tecniche agli strumenti che ho utilizzato.

Infine, nella parte finale, descriverò le scelte stilistiche per il layout del progetto, e il programma EVT (*Edition Visualization Technology*) un *software* per la presentazione dell'edizione digitale del manoscritto.

CAPITOLO 1

IL CODICE E75 P. SUP.

1.1 - Storia del manoscritto

Il codice E75 P. Sup. faceva parte della preziosa biblioteca del patrizio genovese Giovanni Vincenzo Pinelli, amico del cardinale Federico Borromeo. Alla sua morte, nel 1601, Pinelli lasciò la sua biblioteca al nipote duca Cosmo, che abitava a Napoli ma, prima che le casse piene di libri potessero partire per Napoli, il Senato veneto sequestrò 200 volumi di scritti politici con il pretesto che fossero stati copiati dagli archivi del governo veneziano.

In seguito le casse furono caricate su tre navi per essere finalmente trasportate a Napoli all'erede, ma una di queste navi fu catturata dai pirati turchi a largo di Ancona. I pirati avevano sperato di trovare nelle casse un ricco bottino, ma trovandovi soltanto libri, gettarono i codici in mare.

I pescatori di Fermo, trovando tanti fogli di pergamena nelle loro reti, li usarono per le finestre delle loro case, finché un decreto del prefetto pontificio della provincia non li obbligò alla restituzione.

Vennero così recuperare ventidue casse, mentre undici andarono perdute; una volta giunti i libri a Napoli e morto l'erede, vennero messi in vendita.

Il cardinale Borromeo in quegli anni aveva maturato l'idea di costruire a Milano una ricca biblioteca che raccogliesse principalmente testi di storia del Cattolicesimo. È il periodo della Controriforma e il cardinale intendeva creare uno strumento per combattere l'avanzare del Protestantismo.

Essendo venuto a conoscenza della vendita della ricca collezione pinelliana, mandò un suo emissario a Napoli per curarne l'acquisizione.

L'emissario, Grazio Maria Grazi, nel gennaio 1607 scrisse a Federico di aver visto i libri e per quanto danneggiati dal lungo viaggio erano tali che “*non si pagheranno mai tanto che non vaglino sempre due volte tanto di più*” (Paredi 1981).

La celebre libreria pinelliana venne infine acquistata per conto del cardinale Borromeo per 3050 ducati, e con essa il codice E75 P. Sup., all'interno della cui copertina si legge (vedi Figura 1):

“*Hic Codex fuit Vincentij Pinelli V.Cl. a cuius heredibus tota eiusdem bibliotheca empta fuit anno 1609*”.

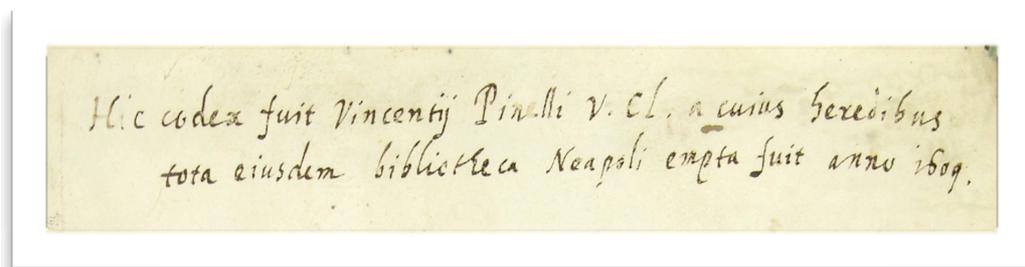


Figura 1 - Particolare dell'interno della copertina

Il codice giacque dimenticato per oltre due secoli nella Biblioteca Ambrosiana, finché non venne rintracciato da Baldassarre Boncompagni agli inizi del 1853, che lo pubblicò l'anno successivo a Firenze con il titolo *Tre scritti inediti di Leonardo Pisano*. Nel 1854 ne pubblicò una seconda edizione, rivista e corretta, dal titolo *Opuscoli di Leonardo Pisano* (Picutti 1983).

1.2 - Descrizione fisica

Il codice E 75 P. Sup. risale al XV secolo, è in pergamena, di dimensioni 21,7 cm per 14 cm ed è composto da 42 carte, tutte numerate nel recto, tranne la prima, coi numeri da 1 a 41.

Il testo, in latino, è in caratteri goticeggianti, scritto con inchiostro nero, mentre alcuni titoli, sottolineature e capilettura sono scritti con inchiostro rosso.

Le carte 1r e 19r, che rappresentano le prime pagine dei due testi contenuti nel manoscritto, presentano i capilettura in oro, decorati con miniature in rosso, verde, blu con applicazioni in oro.



Figura 2 - Particolare del foglio 1r

Il testo si presenta disposto su un'unica colonna, allineato in modo giustificato, tranne alcuni capilettura che fuoriescono verso sinistra. I fogli sono raggruppati in fascicoli e alla fine di ognuno il copista ha trascritto, nell'angolo in basso a destra, le prime parole del fascicolo seguente, in verticale.

Il copista ha utilizzato numerose abbreviazioni caratteristiche del latino medievale, di cui si presenteranno alcuni esempi nel capitolo relativo alla codifica testuale.

A margine di alcune pagine sono presenti disegni geometrici, relativi al contenuto delle pagine in questione; gli spazi a bordo pagina sono utilizzati anche per aggiunte, correzioni e specchietti.

1.3 - Descrizione del contenuto

Il codice E 75 P. Sup. contiene due testi di Leonardo Pisano, il *Flos*, fino alla carta 18v e il *Liber Quadratorum* dalla 19r in poi.

All'interno del *Flos* si possono individuare due sezioni: dalla carta 1r alle prime quattro righe della 15r è contenuto il *Flos Leonardi bigòlli pisani super solutionibus quarundam questionum ad numerum et geometriam vel ad utrumque pertinentium*, mentre le carte dalla 15r alla 18v contengono *Epistola scripti Leonardi ad Magistrum Theodorum phylosophum domini Imperatoris*, ovvero una lettera a maestro Teodoro, filosofo della corte di Federico II.

Nella prima trascrizione moderna del codice E 75 P.Sup. risalente al 1853, ad opera di Baldassarre Boncompagni, queste due sezioni sono considerate separate, infatti Boncompagni intitola la sua opera *Tre scritti inediti di Leonardo Pisano* (il terzo è ovviamente il *Liber Quadratorum*).

Già nella seconda edizione, Boncompagni rivede la sua posizione, intendendo la Lettera come una parte del *Flos*, e difatti modifica anche il titolo in *Opuscoli di Leonardo Pisano*.

A mio avviso è anche la scelta più giusta, dal momento che non c'è una netta divisione grafica tra *Flos* e Lettera, come invece tra *Flos* e *Liber Quadratorum*, e inoltre entrambe le sezioni affrontano lo stesso tipo di problemi matematici.

Il *Flos*, che è poi l'oggetto del mio lavoro, contiene sedici problemi sia di contenuto geometrico, sia di contenuto algebrico. La maggior parte è di carattere pratico, principalmente economico, ad esempio riguardo a divisioni di denaro tra più uomini e ad acquisto di merce.

Per ogni problema Fibonacci fornisce la sua soluzione e il procedimento con cui vi è arrivato.

Non avendo a disposizione un procedimento algebrico standard, Leonardo applica il suo ingegno e la sua intuizione in ogni problema, arrivando a risultati eccellenti quanto astrusi per un matematico moderno e riuscendo addirittura ad estrarre regole generiche da problemi specifici.

Un esempio è il problema che si trova a metà della carta 15r:

“De avibus emendis secundum proportionem datam.

Quidam emit passeris 3 pro uno denario et turtures 2 pro uno denario et columbam 1 pro denarios 2 et ex his tribus generibus avium habuit aves 30 pro denarijs 30. Queritur quot aves emit ex uno quoque genere”

Il quesito riguarda l’acquisto di 30 uccelli di tre specie diverse per 30 denari, sapendo che tre passeri costano un denaro, due tortore un denaro e una colomba due denari. Come primo passaggio Leonardo propone di acquistare 30 passeri, che sono gli uccelli meno costosi, arrivando così a spendere 10 denari e avendone ancora 20. Non resta quindi che sostituire iterativamente un passero con un uccello di un’altra specie, fino ad aver speso tutti i 20 denari restanti.

Ponendo $y =$ tortore e $z =$ colombe, la sostituzione degli uccelli altro non è che la risoluzione di questa equazione:

$$y/6 + 10z/6 = 20$$

quindi:

$$y + 10z = 120$$

Le soluzioni del problema sono quindi: 9 passeri, 10 tortore, 11 colombe.

Al di là delle specifiche soluzioni, Leonardo ha individuato un metodo risolutivo corretto e applicabile a qualsiasi problema di questo genere e ne dà una dimostrazione concreta fornendone altri esempi nel *Flos*.

All’interno del testo sono presenti alcune dediche e riferimenti ad illustri personaggi contemporanei a Fibonacci.

Nella carta 1r, ovvero nel prologo, viene citato il cardinale R, da intendersi come Raniero Capocci Cardinale di Santa Maria in Cosmedin (Boncompagni 1854), uno tra gli uomini di maggior talento vissuti presso la Curia romana nel XIII secolo. Si intuisce che il Cardinale Raniero, apprezzando il lavoro di

Fibonacci, gli aveva richiesto una copia delle sue opere e il matematico, molto lusingato, gli stava mandando proprio una copia del *Flos*, che conteneva alcuni problemi propostigli dai matematici della corte di Federico II.

L'imperatore è appunto citato nella stessa carta, così come, in altre parti del testo, i suoi studiosi di corte maestro Giovanni da Palermo, maestro Teodoro e il donzello Robertino.

1.4 - Vita e opere di Leonardo Pisano

Mi pare doveroso fornire alcune notizie biografiche dell'autore del *Flos*, perché è proprio dalla sua vita che si comprende come sia diventato il più grande matematico occidentale del Medioevo, nonché il precursore della matematica moderna.

Leonardo Pisano, detto Fibonacci da filius Bonaccii (Bonaini 1858), nacque a Pisa nel 1170. Il padre, Guglielmo Bonacci, era un funzionario pubblico della Repubblica pisana, inviato alla Dogana di una grande città portuale sul Mediterraneo: Bugia, più tardi chiamata Bougie e ora Bejaia, nel nord-est dell'Algeria. Questo porto era una piazza molto importante nel quadro degli scambi commerciali con la Barberia orientale, eccellente



Figura 3 - Leonardo Pisano

produttrice di cereali oltre che mercato interessante per l'acquisto di cuoio, lana, cera, allume e scorze tanniche, importanti per la concia (La Navigation du Savoir, pagina *Pisa*). Il lavoro di Guglielmo consisteva nel rappresentare i

mercanti della Repubblica di Pisa, che commerciavano a Bugia. Il padre desiderava che Leonardo diventasse un mercante, così nel 1180 si fece raggiungere presso Bugia, affinché, viaggiando, potesse studiare i procedimenti aritmetici che gli Arabi usavano, e che si rivelavano molto utili nel commercio. Leonardo venne così in contatto con i testi dei più grandi matematici musulmani, come Al-Khawarizmi (l'inventore dell'algebra) e Abu Kamil.

Conobbe l'uso della numerazione indù o araba, cioè l' "*ars per novem figure indorum*" che con il "*signo 0 quod arabice zephirum appellatur*" permetteva di risolvere rapidamente qualsiasi operazione aritmetica (Fibonacci, *Liber Abbaci*, 1202).

Questo metodo di numerazione era già noto in alcuni ambienti dell'Occidente, ma era usato solo per numerare o datare; era visto con scetticismo perché era invenzione di infedeli, e questo poteva offendere la religione dominante, ma anche perché si pensava che le cifre fossero facilmente alterabili, molto sconveniente per le operazioni economiche (Arrighi 1994).

La numerazione usata comunemente era quella romana, molto scomoda e poco pratica per fare i conti, perché non era posizionale come quella araba ed era indispensabile l'uso dell'abaco per fare le operazioni di somma e sottrazione.

Nell'alto medioevo l'opera dei matematici antichi si presentava come qualcosa di prodigioso, che i moderni dovevano necessariamente limitarsi a imitare ciecamente. È chiaro che in un simile contesto, al rientro a Pisa intorno al 1200, la formazione matematica di Leonardo era di gran lunga superiore a quella di molti colleghi europei.

La prima opera di Fibonacci è il *Liber Abbaci*, la cui prima edizione è del 1202 mentre la seconda, rivista e dedicata a Michele Scoto, astrologo della corte di Federico II, è del 1228.

È scritta in latino così che i dotti dell'occidente potessero leggerla agevolmente e contiene una vasta gamma di argomenti interessanti per gli operatori economici (Arrighi 1994). Il titolo del libro non si riferisce propriamente all'abaco, ma si deve più precisamente tradurre come "libro di calcolo".

I modelli per la stesura di quest'opera, ma anche delle successive, sono stati la completezza delle dimostrazioni degli *Elementi* di Euclide, da cui riprende

anche la divisione in 15 capitoli, e il rigore delle opere di Archimede (La Navigation du Savoir, pagina *Pisa*).

Tra i vari problemi proposti da Fibonacci, uno è di fondamentale importanza perché è all'origine della famosissima *Successione di Fibonacci*.

Il quesito è questo:

“Un certo uomo mette una coppia di conigli in un posto circondato su tutti i lati da un muro. Quante coppie di conigli possono essere prodotte da quella coppia in un anno, se si suppone che ogni mese ogni coppia genera una nuova coppia, che dal secondo mese in avanti diventa produttiva?”

Fibonacci propone la seguente risoluzione:

si suppone di avere una coppia di conigli (maschio e femmina). I conigli sono in grado di riprodursi già all'età di 2 mesi, per cui al secondo mese la prima coppia ne genera un'altra; nel recinto abbiamo dunque 2 coppie. Al terzo mese la prima coppia ne genera un'altra mentre la seconda non ha ancora l'età per procreare; nel recinto abbiamo quindi 3 coppie di conigli.

Passa un altro mese: le prima 2 coppie generano altre 2 coppie mentre la terza non può ancora generare; nel recinto ci sono 5 coppie di conigli, e così via di mese in mese.

In pratica la soluzione è data dalla successione numerica:

1 1 2 3 5 8 13 21 34 55 89 144 233...

dove ogni numero si ottiene come somma dei due precedenti secondo lo schema:

$$F_n = F_{n-1} + F_{n-2}$$

Questo è il primo esempio in Europa di successione ricorsiva (ovvero una successione di numeri nella quale ogni termine si ottiene compiendo una qualche operazione sui termini che lo precedono), nonché uno dei primi modelli matematici di crescita di una popolazione, completa di ipotesi restrittive (è

chiaro che nella realtà i conigli prima o poi muoiono e non sempre riescono a riprodurre un maschio e una femmina) e carattere predittivo.

Il *Liber Abbaci* ebbe un impatto fortissimo in un'Europa all'epoca poco interessata al sapere scientifico e la fama di Leonardo arrivò fino all'Imperatore Federico II di Svevia, incoronato re della Germania nel 1212 e nel 1220 nominato a Roma, nella chiesa di San Pietro, Imperatore del Sacro Romano Impero.

L'imperatore aiutò Pisa nel suo conflitto con Genova in mare e con Lucca e Firenze per via terra, e trascorse gli anni successivi al 1227 a consolidare il suo potere in Italia.

In un suo soggiorno a Pisa, forse del 1225, Federico II volle incontrare Fibonacci, poiché molti intellettuali della sua Corte gliene avevano elogiato l'opera.

Leonardo intratteneva infatti contatti epistolari, fin dal suo rientro a Pisa, con Michele Scoto, l'astrologo di corte a cui dedicò il *Liber Abbaci*, col filosofo Teodoro e con Domenico Ispano (*Stupor Mundi*, pagina *Fibonacci*).

Il matematico di corte, Maestro Giovanni da Palermo, durante l'incontro tra l'imperatore e Leonardo, sottopose a quest'ultimo alcuni problemi, risolvibili con equazioni quadratiche e cubiche, le cui soluzioni furono riportate nell'opuscolo *Flos*.

La seconda opera di Fibonacci è la *Practica Geometriae*, del 1220, dedicata a Domenico Ispano, della corte di Federico II.

Tratta, in 8 capitoli, della geometria euclidea, con dimostrazioni rigorose di teoremi e problemi applicativi. Inoltre il libro include informazioni pratiche per i controllori, incluso un capitolo su come calcolare l'altezza di oggetti elevati, usando i triangoli simili.

L'ultimo capitolo presenta ciò che Fibonacci chiama sottigliezze geometriche, tra cui il calcolo dei lati di un pentagono e di un decagono dal diametro di circonferenze circoscritte e inscritte.

Una terza opera, il *Liber Quadratorum*, forse la più elevata per i contenuti, tratta delle proprietà delle terne pitagoriche. In quest'opera Fibonacci dimostra, tra l'altro che non possono esistere coppie di numeri interi x, y tali che x^2+y^2 e x^2-y^2

siano entrambi quadrati, e che $x^4 - y^4$ non può essere un quadrato. Egli definì inoltre il concetto di *congruum*, un numero della forma $ab(a+b)(a-b)$, se $a+b$ è pari, e quattro volte questo, se $a+b$ è dispari.

L'ultimo documento conosciuto su Fibonacci è un decreto del governo pisano del 1240, che attribuisce un onorario annuale di 20 denari pisani, più le spese, al maestro Leonardo Bigollo, per i servizi resi insegnando aritmetica e contabilità ai cittadini. Morì pochi anni dopo il 1240 (University of Evansville, articolo del *Prof. Kimberling*).

Fibonacci esercitò un'influenza diretta solo attraverso quelle parti del *Liber abbaci* e del *Practica Geometriae* che servirono ad introdurre i numeri indo-arabici e i metodi che contribuirono al dominio dei problemi della vita quotidiana. Per quanto riguarda la teoria dei numeri, il lavoro di Fibonacci restò praticamente sconosciuto per il resto del Medioevo e gli stessi risultati appariranno nell'opera del frate Maurolico, nel 1520.

CAPITOLO 2

ELABORAZIONE DEL PROGETTO

2.1 - Fasi preliminari

2.1.1 - Acquisizione del materiale

Per poter iniziare la stesura del progetto, mi sono dovuta procurare le immagini digitali delle pagine del *Flos*. Queste mi sono state fornite dalla Veneranda Biblioteca Ambrosiana di Milano, che possiede l'unico esemplare esistente del *Flos*, e sono state realizzate appositamente per il mio progetto dai tecnici del suo archivio.

Il *copyright* sulle immagini rimane esclusivo dell'Ambrosiana e la mia edizione digitale del testo verrà pubblicata sul sito web www.ambrosiana.it, nonché collocata nell'archivio digitale della biblioteca. Mi sono state fornite immagini *jpg*, di circa 4-5 *megabyte* di dimensione.

Per quanto riguarda la trascrizione del testo, mi sono avvalsa delle due edizioni del Boncompagni, che ho acquisito attraverso l'uso di un *software* OCR, **Readiris Pro** versione 11.5 per Macintosh.

Ho poi corretto manualmente gli inevitabili e numerosi errori creatisi durante l'acquisizione.

2.1.2 - Stesura del percorso

Una volta raccolto il materiale, è iniziato il lavoro per arrivare allo scopo finale che mi ero prefissata: un'edizione digitale del *Flos*, sia in versione diplomatica che critica, consultabile attraverso il *software* EVT, che permette di visualizzare contemporaneamente l'immagine e la trascrizione del manoscritto.

Ho ritenuto utile, inoltre, creare delle note per ogni problema in modo tale che, attraverso l'uso di un linguaggio più moderno, i procedimenti utilizzati da Leonardo Pisano per risolvere i quesiti potessero risultare più comprensibili per il *target* di persone per cui stavo creando questa edizione del *Flos*, ovvero, in linea di massima, studiosi di Storia Medievale.

Quindi, il progetto si è svolto attraverso queste fasi:

- studio del tipo di codifica da effettuare;
- codifica del testo;
- note ai problemi;
- caricamento dei file in EVT
- scelte grafiche e di presentazione (soprattutto per quanto riguarda le note).

2.2 - La codifica del testo

Prima di scendere nel dettaglio della codifica del mio testo, ritengo essenziale spiegare che cosa vuol dire codificare un testo, qual è lo scopo e quali gli standard.

2.2.1 - Che cosa è XML

Un testo è un messaggio linguistico di senso compiuto e interpretarlo significa estrarne le informazioni che ne caratterizzano la struttura, la presentazione, la natura linguistica e il suo contenuto informativo. Affinché un computer sia in grado di interpretare un testo, ovvero, affinché le informazioni riguardanti un testo siano *machine readable*, è necessario renderne esplicita l'interpretazione, attraverso una codifica.

Il testo infatti è una fonte di dati linguistici, correlati secondo diversi piani di organizzazione (struttura del testo, del contesto ecc.), ma se questi dati non sono inseriti in uno schema che li organizza e li trasforma in informazione, non hanno nessun significato intrinseco per la macchina.

Codificare un testo significa quindi “tradurlo” in un formato comprensibile per il computer, usando un linguaggio di codifica digitale, in particolare un linguaggio di marcatura (*mark-up language*) come XML.

XML (eXtensible Mark-up Language) nasce nel 1996 come evoluzione e semplificazione di SGML, per rispondere alla necessità di un linguaggio di *mark-up* con maggiore libertà nella definizione delle etichette (*tag*) pur rimanendo nell'ambito del rispetto di uno standard.

Nel 1998 XML diventa uno standard del W3C, il consorzio internazionale per la standardizzazione di linguaggi e strumenti per il web; XML è indipendente dal sistema operativo ed è estensibile, ovvero non è composto da un numero fisso di elementi.

Un documento XML è un file di testo che contiene una serie di elementi, attributi e testo secondo delle regole sintattiche stabilite. Gli elementi rappresentano i blocchi costitutivi in cui si articola un testo e sono individuati inserendo un delimitatore di apertura e di chiusura del tipo:

<etichetta>testo</etichetta>

Gli elementi possono essere annidati tra loro e in questo modo il testo può essere rappresentato attraverso strutture gerarchiche di varia profondità; inoltre gli elementi possono essere arricchiti con attributi che ne specifichino determinate caratteristiche.

Un documento XML deve essere ben formato (*well formed*), ad esempio deve esistere un solo nodo radice, ogni nodo figlio deve essere contenuto completamente nel nodo padre ecc. Può inoltre essere validato, sulla base del relativo schema di codifica, definito attraverso il linguaggio DTD (*Document Type Definition*) o Schema.

2.2.2 - Che cosa è XSL

XSL (*eXtensible Stylesheet Language*) è un linguaggio di programmazione per la presentazione e la trasformazione dei documenti XML ed è uno standard del W3C.

Comprende tre diversi linguaggi:

- XSL-T (*XSL Transformation*): per la trasformazione di un documento XML;
- XSL-FO (*XSL Formatting Objects*): per formattare un oggetto trasformato;
- XPath (*XML Path Language*): definisce espressioni e metodi per accedere ai nodi di un documento XML;

Ai fini del mio progetto, è necessario e sufficiente spiegare come funziona XSL-T, che in seguito chiamerò semplicemente XSL.

XSL segue la sintassi di XML, quindi un file .xsl è validabile. Con un foglio di stile XSL si può trasformare un documento XML in un altro documento XML o in un formato più adatto alla visualizzazione, come HTML.

Le prime righe di un foglio di stile devono essere necessariamente queste (almeno secondo le ultime specifiche):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tei="http://www.tei-c.org/ns/1.0">
```

XSL è basato su regole di trasformazione dichiarative e non necessariamente sequenziali. Il processore XSL-T riceve in input un documento XML, ne percorre i nodi, confrontandoli con le regole di trasformazione; quando trova una regola corrispondente ad un determinato nodo, la applica e produce l'output, proseguendo nella sua ricerca fino alla fine del documento. Il risultato finale è un nuovo albero.

Esistono vari tipi di istruzioni, tra cui:

- **xsl:template**, genera un *template*;
- **xsl:apply-templates**, applica le regole definite in un *template*;
- **xsl:attribute**, crea un attributo;
- **xsl:element**, crea un nuovo elemento;
- **xsl:if**, applica una regola se sono soddisfatte determinate condizioni;
- **xsl:value-of**, restituisce il valore di un elemento o di un attributo.

Per fornire un esempio pratico, analizzerò ora uno dei fogli di stile che ho creato per il mio progetto (Tabella1).

Istruzione	Risultato
<code><xsl:output method="html"/></code>	L'output sarà un documento html
<code><xsl:template match="/"> <html> <head> <title>Flos</title> </head> <body> <xsl:apply-templates /> </body> </html> </xsl:template></code>	È l'istruzione per costruire l'ossatura dell'output. Il processore XSL legge il nodo radice (indicato con "/"), che è il documento stesso, e restituisce <code><html><head><title>Flos</title></head><body></code> , poi avvia l'elaborazione dei nodi figli e solo alla fine restituisce <code></body></html></code>
<code><xsl:template match="tei:teiHeader"> </xsl:template></code>	Quando il processore trova il nodo <code>teiHeader</code> , non restituisce nulla.
<code><xsl:template match="tei:text"> <xsl:apply-templates/> </xsl:template></code>	Quando il processore trova il nodo <code>text</code> , avvia l'elaborazione sui nodi figli.
<code><xsl:template match="tei:front"> </xsl:template></code>	Quando il processore trova il nodo <code>front</code> , non restituisce nulla.
<code><xsl:template match="tei:lb">
<b style="color:green" > <xsl:value-of select="@n"/>&#160;&#160; </xsl:template></code>	Quando il processore trova un elemento <code></code> , crea un'interruzione di riga, restituisce il valore dell'attributo "n" di <code><lb></code> in grassetto verde, seguito da due spazi.
<code><xsl:template match="tei:pb">

 <b style="color:green">foglio&#160; <xsl:value-of select="@xml:id"/> </xsl:template></code>	Quando il processore trova un elemento <code><pb></code> , crea due interruzioni di riga, restituisce il valore dell'attributo "xml:id" di <code><pb></code> in grassetto verde, preceduto dalla stringa "foglio" e uno spazio.
<code><xsl:template match="*[@rend='red']"> <xsl:apply-templates /> </xsl:template></code>	Quando il processore trova "red" come valore dell'attributo "rend", in qualsiasi elemento, restituisce in rosso il testo contenuto in tale elemento, all'interno di un contenitore di tipo <code></code> .
<code><xsl:template match="tei:choice"> <xsl:apply-templates select="tei:abbr"/> <xsl:apply-templates select="tei:sic"/> </xsl:template></code>	Quando il processore trova il nodo "choice", restituisce il contenuto dell'elemento "abbr", o dell'elemento "sic"
<code><xsl:template match="tei:back"> </xsl:template></code>	Quando il processore trova il nodo <code>back</code> , non restituisce nulla.

Tabella 1 - Esempi di trasformazione

2.2.3 - La TEI

Con il fine di realizzare uno standard per la codifica digitale dei testi letterari, in particolare dei testi del passato, e di normalizzare i formati di memorizzazione dell'informazione testuale, nel 1987 è stato creato il progetto di ricerca TEI (*Text Encoding Initiative*), che ha portato alla stesura di linee guida (*Guidelines*). La prima versione delle *Guidelines*, chiamata TEI P1, risale al 1990, mentre l'ultima versione, TEI P5, è stata pubblicata in versione definitiva alla fine del 2007.

Le *Guidelines* forniscono uno strumento per rendere esplicite determinate caratteristiche di un testo, con il fine di facilitarne l'elaborazione mediante programmi informatici basati su diverse piattaforme. Per questo sono chiare, di semplice utilizzazione per i ricercatori, consentono di creare estensioni definite dall'utente e sono, ovviamente, conformi agli standard esistenti.

Lo schema di codifica della TEI è basato sulla sintassi di XML (SGML nelle prime versioni) e consente di rappresentare la struttura di varie tipologie di testo e le caratteristiche testuali rilevanti per diverse aree di ricerca. Nelle *Guidelines* è raccolto un elevato numero di elementi strutturali e semantici, definiti per mezzo di DTD o Schema, caratterizzati da una forte modularità, in particolare nella versione P5. L'elevata modularità permette di creare degli schemi di codifica ad hoc, scegliendo solo i moduli necessari e modificando gli elementi secondo le proprie necessità.

I moduli di base, che contengono gli elementi utili per tutti i documenti da codificare, sono:

- **tei:** definisce le classi di elementi, le macro e i *datatype* che verranno usati per tutti i moduli
- **header:** l'intestazione contenente i metadati relativi al documento TEI XML
- **textstructure:** elementi strutturali per qualsiasi tipo di testo
- **core:** elementi utili in qualsiasi tipo di documento

A seconda del tipo di documento da codificare, si possono aggiungere moduli specifici, tra cui:

- **msdescription:** metadati relativi a manoscritti
- **transcr:** trascrizione di fonti primarie
- **textcrit:** apparato critico

Il sito del consorzio TEI (<http://www.tei-c.org/>) offre un ottimo supporto agli utenti, in particolare mette a disposizione uno script, TEI Roma, per creare schemi di codifica *ad hoc*, selezionando i moduli di interesse tra quelli a scelta e modificando e inserendo elementi.

Con il fine di facilitare l'applicazione della TEI da parte degli utenti, senza richiedere la conoscenza dell'intera DTD e della relativa documentazione, ne è stato creato un sottoinsieme, denominato *TEI Lite*, che permette di creare documenti TEI-compatibili in maniera rapida e semplice, e che riesce a soddisfare il 90% delle esigenze del 90% degli utenti.

2.2.4 - Creazione della DTD

Per la composizione dello schema di codifica per il mio testo, ho utilizzato lo strumento TEI Roma, reperibile alla pagina <http://www.tei-c.org/Roma/>

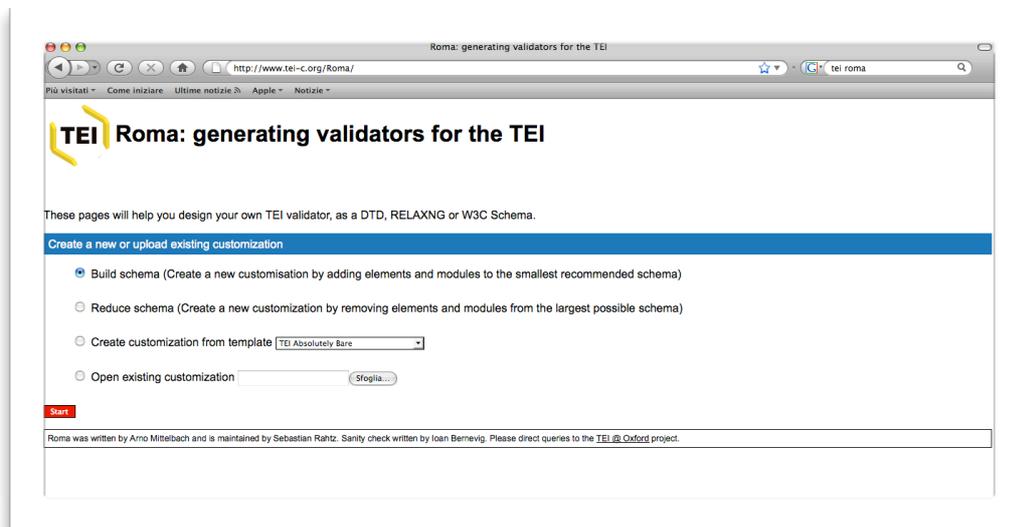


Figura 4 - Home Page TEI Roma

Come si vede nella figura 4, TEI Roma permette di creare uno schema a partire dai moduli base, oppure eliminare moduli a partire dallo schema completo o modificare uno schema già esistente.

Una volta deciso da quale cominciare, si può scegliere la lingua, il nome e il formato in cui salvare.

Io ho scelto di utilizzare uno schema di codifica nel formato DTD, ma è consigliato salvare anche in formato ODD in modo da poterlo eventualmente caricare di nuovo su Roma e modificarlo.

Per la codifica del mio testo, ho deciso di partire dallo schema base, composto dai moduli “**tei**”, “**core**”, “**textstructure**” e “**header**”, e di aggiungere i moduli “**transcr**” e “**msdescription**”.

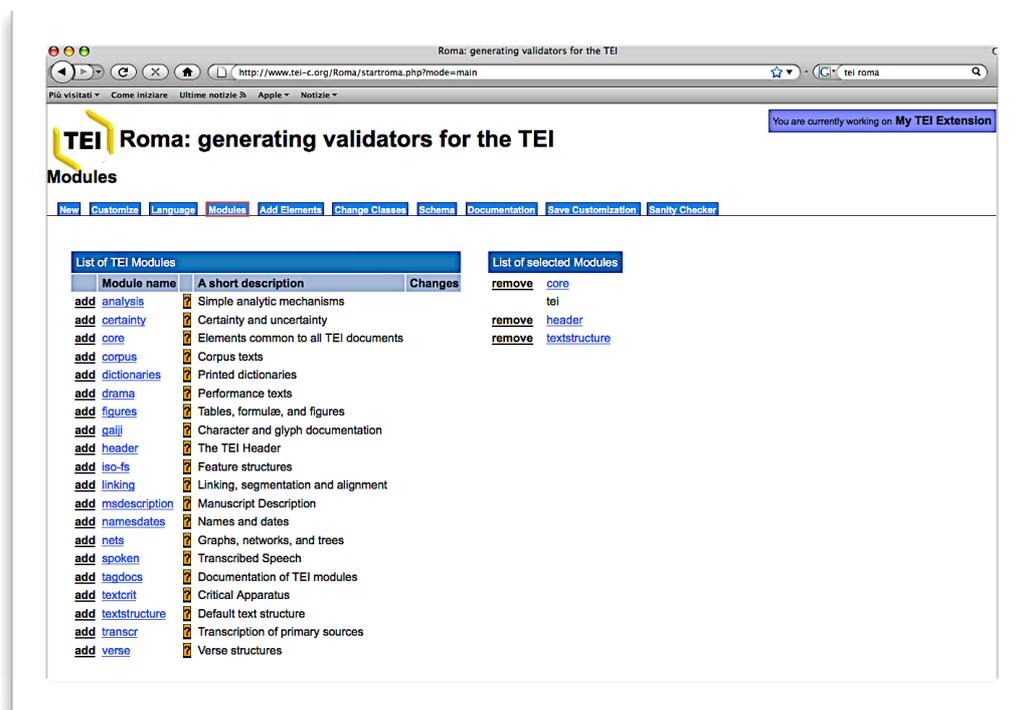


Figura 5 - TEI Roma: scelta dei moduli

Il modulo “**transcr**” contiene gli elementi utili per la rappresentazione delle fonti primarie, come un manoscritto. Permette di associare le immagini alla trascrizione e di registrare errori, alterazioni, correzioni, cambi di mano, in generale tutti quei fenomeni testuali dei quali si ha la necessità di prender nota. Il modulo “**msdescription**” contiene una gamma di elementi con cui annotare

dettagliate informazioni descrittive sulle fonti primarie manoscritte. L'elemento principale di questo modulo è

<msDesc></msDesc>

e appare all'interno della *teiHeader*. L'elemento **<msDesc>** può contenere una grande quantità di altri elementi che marcano diversi tipo di informazioni.

Ad esempio:

- **<msIdentifier>**: contiene le informazioni utili per l'identificazione del manoscritto
- **<physDesc>**: contiene la descrizione fisica del documento
- **<history>**: contiene tutti gli elementi per la descrizione dell'origine e della storia del codice.

Oltre a questi due moduli, ho aggiunto il modulo "**namesdate**", che ho ritenuto utile per la marcatura dei nomi propri, e, considerato che il *Flos* ha un contenuto matematico, il modulo "**figures**" per codificare numeri e formule (ma in realtà non ne ho avuto grande necessità).

2.2.5 - La codifica del testo

Per annotare il testo ho utilizzato il programma **<oXygen/> XML Editor** versione 10.0 per Macintosh, scaricabile dal sito <http://www.oxygenxml.com> ed utilizzabile gratuitamente con una licenza valida 30 giorni.

Il *software* aiuta nella codifica di un testo perché verifica costantemente che il file XML sia *well-formed* e, una volta associata una DTD, controlla anche la validità, segnalando eventuali errori.

Inoltre permette di associare un foglio di stile XSL e visualizzare il risultato delle trasformazioni in una pagina HTML. Come primo passaggio della codifica, ho creato lo scheletro del file XML in questo modo:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TEI SYSTEM "myTEI.dtd">
<TEI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <teiHeader>
    </teiHeader>
    <text>
    </text>
</TEI>

```

Sono passata quindi a sviluppare la `teiHeader`, inserendo l'elemento `<fileDesc>` e i suoi figli, in questo modo:

```

<teiHeader>
    <fileDesc>
        <titleStmt>
        </titleStmt>
        <publicationStmt>
        </publicationStmt>
        <sourceDesc>
        </sourceDesc>
    </fileDesc>
</teiHeader>

```

All'interno di `<titleStmt>` ho aggiunto gli elementi per l'individuazione di titolo, autore e responsabile della codifica:

```

<titleStmt>
    <title>Flos super solutionibus quarumdam questionum ad numerum
        et geometriam vel ad utrumque pertinentium.</title>
    <author>Leonardo Pisano</author>
    <respStmt>
        <resp>Edizione elettronica a cura di </resp>
    </respStmt>
</titleStmt>

```

```
<name>Alessandra Giammalva </name>
</respStmt>
</titleStmt>
```

All'interno di **<publicationStmt>** ho inserito l'informazione riguardo all'anno di pubblicazione:

```
<publicationStmt>
  <date>2009</date>
</publicationStmt>
```

Infine ho completato l'elemento **<sourceDesc>** con **<msDesc>**, che fa parte del modulo "msdescription".

In **<msDesc>** ho inserito informazioni riguardo all'identificazione del manoscritto:

```
<msIdentifier>
  <country>Italia</country>
  <settlement>Milano</settlement>
  <repository>Veneranda Biblioteca Ambrosiana</repository>
  <idno>E.75 P.SUP</idno>
</msIdentifier>
```

e riguardo al contenuto, individuando da subito i due testi che compongono il manoscritto:

```
<msContents>
  <msItem n="1"></msItem>
  <msItem n="2"></msItem>
</msContents>
```

All'interno di ciascun **<msItem>** ho dato informazioni riguardo al titolo, al numero di pagine e alla struttura del testo.

Ho inoltre inserito, all'interno di **<msDesc>**, l'elemento **<physDesc>** che contiene le informazioni riguardanti l'aspetto esteriore del manoscritto:

```
<physDesc>  
  <objectDesc form="codex">  
    <supportDesc material="perg">  
      <support>Pergamena</support>  
      <extent>42 carte, numerate nel recto, tranne la prima, coi numeri  
        1-41  
      <dimensions unit="cm">  
        <height>21.7</height>  
        <width>14</width>  
      </dimensions>  
    </extent>  
  </supportDesc>  
  <layoutDesc>  
    <layout>Il testo si presenta...</layout>  
  </layoutDesc>  
</objectDesc>  
  <typeDesc>  
    <p>Il codice è scritto...</p>  
  </typeDesc>  
  <decoDesc>  
    <p>Il testo è scritto con inchiostro nero...</p>  
  </decoDesc>  
</physDesc>
```

e l'elemento **<history>** che raccoglie le notizie sulle origini e la storia del manoscritto:

```

<history>
  <origin>
    <p>Il codice risale al XV secolo, probabilmente alla prima metà.</p>
  </origin>
  <provenance>
    <p>Il codice faceva parte ...</p>
  </provenance>
  <acquisition>
    <p>Il cardinale Borromeo mandò a...</p>
  </acquisition>
</history>

```

Una volta completata la `teiHeader`, sono passata alla codifica vera e propria del testo. All'interno del tag `<text>` ho creato tre nuovi elementi:

- `<front>`: contiene ulteriori informazioni riguardo a titolo, autore ecc.
- `<body>`: contiene la trascrizione annotata del *Flos*.
- `<back>`: contiene le note ai problemi matematici.

Analizzando il manoscritto e le sezioni in cui è diviso il testo, ho deciso di creare all'interno dell'elemento `<body>`, due elementi `<div1>`: uno contenente i fogli da 1r alla prima metà di 15r, ovvero il *Flos* vero e proprio, l'altro contenente i restanti fogli fino a 18v, ovvero la lettera a Maestro Teodoro.

```

<body>
  <div1 xml:id="part1">
  </div1>
  <div1 xml:id="part2">
  </div1>
</body>

```

Ho poi creato all'interno di `<div1>`, più `<div2>`: il primo `<div2>` del *Flos* marca l'introduzione del testo, con la dedica al destinatario, mentre il primo `<div2>` della lettera a Teodoro contiene la parte introduttiva dell'epistola.

I restanti elementi `<div2>` marcano ognuno un problema matematico, per questo ho loro assegnato un attributo di tipo `xml:id`, per individuare in maniera univoca ogni singolo quesito.

L'ossatura della trascrizione viene quindi ad essere:

```
<body>
  <div1 xml:id="part1">
    <div2 xml:id="intro1">
    </div2>
    <div2 xml:id="pr1">
    </div2>
    .
    .
    .
    <div2 xml:id="pr10">
    </div2>
  </div1>
  <div1 xml:id="part2">
    <div2 xml:id="intro2">
    </div2>
    <div2 xml:id="pr11">
    </div2>
    .
    .
    .
    <div2 xml:id="pr16">
    </div2>
  </div1>
</body>
```

Sono passata quindi ad evidenziare le interruzioni di pagina, inserendo all'inizio di ognuna il tag `<pb>` con l'aggiunta dell'attributo `xml:id` per indicare il numero di foglio. Ad esempio il marcatore per individuare la prima pagina, ovvero il foglio 1r è:

`<pb xml:id="r1"/>`

Per indicare le interruzioni di riga ho aggiunto l'elemento `<lb>` con l'attributo `n` che ne identifica il numero:

`<lb n="1"/>`

L'unico problema che ho incontrato in questo punto del mio lavoro di annotazione, riguarda alcuni titoli, che non seguono il normale flusso di scrittura sulla pagina.

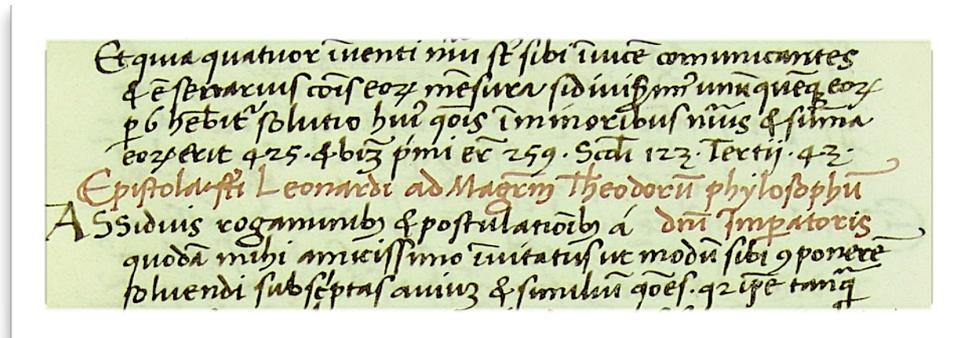


Figura 6 - Particolare del foglio 15r

Come si vede nella figura 6, il titolo della *Epistola ad Magistrum Theodorum*, in rosso, arrivato alla fine della riga, non prosegue in quella successiva da sinistra verso destra, ma occupa la parte finale di tale riga, mentre nella parte iniziale si trovano le prime parole del corpo della lettera.

Nella codifica di questo passaggio (così come nei passaggi analoghi), ho preferito considerare il titolo come un tutt'uno, facente parte della stessa riga,

per agevolare la comprensione del testo, a discapito della totale aderenza della trascrizione all'immagine per quanto riguarda la resa visiva.

Il risultato è questo:

```
<lb n="1000"/><head rend="red">Epistola scripti Leopardi ad Magistrum  
Theodorum phylosophum domini Imperatoris</head>  
<lb n="1001"/><p>Assiduis rogaminibus et postulationibus a  
<lb n="1002"/>quodam mihi amicissimo invitatus ut modum sibi  
conponerem  
. .  
</p>
```

Il passaggio successivo è stato prendere nota di tutti i nomi propri che compaiono nel testo, inserendoli nell'elemento `<name>` in previsione della possibilità di creare un elenco. (in realtà non ce n'è stata necessità visto il limitato numero di nomi citati).

A questo punto è iniziata la parte più difficile e minuziosa della codifica: individuare ed annotare le abbreviazioni, scegliendo il modo migliore per renderle correttamente dal punto di vista grafico e, contemporaneamente, fornirne la versione espansa.

Ho marcato ciascuna abbreviazione con il *tag* `<choice>`, contenente i *tag* `<abbr>` per la forma abbreviata e `<expan>` per quella "sciolta", con questo risultato:

```
<choice>  
  <abbr type="abbr">abbreviazione</abbr>  
  <expan>forma espansa</expan>  
</choice>
```

Per non appesantire il file XML, ho creato un'entità per ogni possibile <choice> e l'ho richiamata nel testo. Ad esempio:

```
<!ENTITY qd '<choice> <abbr type="abbr">qd</abbr> <expan>quod  
</expan> <choice>'
```

L'elenco delle entità è cospicuo, in quanto il copista ha usato molti tipi di abbreviazione, e più di un tipo per la stessa forma espansa.

Per la resa grafica dei simboli ho consultato il sito <http://www.mufl.info/>.

MUFI (*Medieval Unicode Font Initiative*) è un gruppo di lavoro, formato da studiosi che ricercano una soluzione ai problemi di codifica e resa grafica dei caratteri nei testi medievali.

“Medieval Unicode Font Initiative is a non-profit workgroup of scholars and font designers who would like to see a common solution to a problem felt by many medieval scholars: the encoding and display of special characters in Medieval texts written in the Latin alphabet.”

MUFI Medieval Unicode Font Initiative

Disclaimer: This site is managed by scholars in Medieval studies with the aim of establishing a consensus on the use of Unicode among medievalists. It is not affiliated with or endorsed by Unicode.

Background

The Medieval Unicode Font Initiative is a non-profit workgroup of scholars and font designers who would like to see a common solution to a problem felt by many medieval scholars: the encoding and display of special characters in Medieval texts written in the Latin alphabet.

MUFI was founded in July 2001 by a workgroup consisting of Odd Einar Haugen (Bergen), Alec McAllister (Leeds) and Tarrin Willis (Sydney/Aberdeen). The members of the workgroup communicates primarily by e-mail, but have occasionally met in Leeds (July 2002 and 2003). The first MUFI group meeting was held in Bergen (30-31 August 2003), the second in Lisboa, (10-12 March 2005), the third in Bonn (12-13 June 2006) and the fourth in Mainz (23 June 2008). As of August 2006, MUFI has a board of four members (listed in the right column of this page).

Any scholars or font designers who would like to join the workgroup should contact one of the board members listed on this page. Over the last six years the MUFI group has had regular contributors from (in alphabetical order) Australia, Denmark, Iceland, Ireland, Germany, Great Britain, Norway, Portugal, Spain, Switzerland and USA.

Why Unicode?

Unicode is the new international font standard. In version 5.0, it covers approx. 100,000 characters in living as well as historical scripts. It is fully supported by computer platforms like Linux, Mac and Windows. There is simply no alternative to Unicode.

Many characters needed by medieval scholars are already defined in Unicode, but a great number of other characters, and especially abbreviation marks, are missing. The Private Use Area in Unicode may be used for encoding missing characters, but we would like to see as many special characters defined in the official area as possible.

For this reason, the MUFI groups is pursuing two lines, (a) coordinating the allocation of medieval characters in the Private Use Area and (b) proposing missing medieval characters to Unicode.

NEW! In the Unicode Standard v. 5.1, which was released 4 April 2008, 152 medieval (and classical) characters have been accepted. The new characters are located in the charts *Combining Diacritical Marks Supplement* (26 chars.), *Latin Extended Additional* (10 chars.), *Supplemental Punctuation* (15 chars.), *Ancient Symbols* (12 chars.), and especially *Latin Extended-D* (89 chars.). Cf. the section on medievalist proposals to Unicode below.

Board

Odd Einar Haugen, Bergen (Chair)
▶ E-mail
▶ [Medieval Nordic Text Archive](#)

Andreas Stötzner, Leipzig (Deputy Chair)
▶ E-mail
▶ [SIGNA](#) (Signographical research)
▶ [Andron](#) (Unicode font)

Alec McAllister, Leeds
▶ E-mail
▶ [Home page](#) (with the font LeedsUni)

Tarrin Willis, Sydney/Aberdeen
▶ E-mail
▶ [Home page](#)
▶ [Skaldic poetry project](#)

Smart font technology

David J. Perry has published a survey of present font technology, covering the smart fonts on Windows (OpenType), Mac (AAT) and Linux. The survey is a work-in-progress and comments are welcome:

▶ [Cutting-edge text processing](#)

Follow the link under "recent updates".

Other fonts & projects

Peter S. Baker, University of Virginia
▶ E-mail
▶ [Junicode](#) (a Unicode font)
▶ [Old English](#) at Univ. of Virginia

Figura 7 - Home Page del sito MUFI

MUFI, tra le altre cose, mette a disposizione alcuni font, compatibili con Unicode, che contengono tutti i caratteri descritti nella *MUFI recommendation*. Ho scelto il font Junicode, che nella versione attuale è compatibile anche con le specifiche MUFI 2.0. Ovviamente, per una corretta visualizzazione del mio progetto, dovrà essere installato anche nella libreria font degli utenti.

Di seguito mostro alcune delle abbreviazioni più significative (i caratteri in verde sono i nomi delle entità, quelli in rosso sono le abbreviazioni e quelli in blu le espansioni):

<!ENTITY am '<abbr type="abbr">ǎ</abbr><expan>am</expan></choice>'>

<!ENTITY biz '<abbr type="abbr">biž</abbr><expan>bizant</expan></choice>'>

<!ENTITY ergo '<abbr type="abbr">ġ</abbr><expan>ergo</expan></choice>'>

<!ENTITY hoc '<abbr type="abbr">ĥ</abbr><expan>hoc</expan></choice>'>

<!ENTITY mihi '<abbr type="abbr">ĥ</abbr><expan>mihi</expan></choice>'>

<!ENTITY pro '<abbr type="abbr">ṑ</abbr><expan>pro</expan></choice>'>

<!ENTITY quam '<abbr type="abbr">ṑ</abbr><expan>quam</expan></choice>'>

<!ENTITY quoque '<abbr type="abbr">ṑ</abbr><expan>quoque</expan></choice>'>

<!ENTITY rum '<abbr type="abbr">ṑ</abbr><expan>rum</expan></choice>'>

<!ENTITY us '<abbr type="abbr">ṑ</abbr><expan>us</expan></choice>'>

Ettore Picutti, nell'articolo *Il Flos di Leonardo Pisano (Physis, numero 25, fascicolo 2, anno 1983)*, suggerisce delle correzioni al testo, riguardanti principalmente l'errata trascrizione di alcune cifre. Una volta verificata la validità delle osservazioni di Picutti, ho annotato tali errori in questo modo:

<choice> <sic>forma errata</sic> <corr>forma corretta</corr> </choice>

Ho segnalato quindi le porzioni di testo aggiunte, con il tag `<add>` e l'attributo `place` per indicarne la posizione nella pagina, e quelle cancellate, con il tag `` e il relativo attributo `rend` per indicare il modo in cui è avvenuta la cancellazione.

Ad esempio:

```
<add place="supralinear">testo inserito nell'interlinea superiore</add>
```

```
<add place="margin-right">testo inserito nel margine destro</add>
```

```
<del rend="strikethrough">testo cancellato con una linea</del>
```

Per l'ultima "rifinitura" della codifica, ho marcato le porzioni di testo rosso, per prendere nota del cambio di inchiostro, in questo modo:

```
<rs rend="red">testo in rosso</rs>
```

2.3 - Le note ai problemi

2.3.1 - Creazione delle note

Come accennato nei paragrafi precedenti, ho deciso di corredare i problemi matematici con note esplicative, al fine di rendere più comprensibili i passaggi che Fibonacci ha eseguito per arrivare alla soluzione. La difficoltà nell'approccio ad un problema del *Flos* non sta tanto nel contenuto, quanto nella forma in cui è scritto, lontana anni luce da quella moderna.

Per ogni quesito ho percorso questi passaggi:

- Lettura e comprensione dell'intestazione, per capire a quale domanda cerca di rispondere Fibonacci.

- Lettura e comprensione del testo, per capire quali sono i passaggi risolutivi.
- Prima redazione della nota.
- Confronto con la soluzione proposta da Ettore Picutti nell'articolo citato nel paragrafo precedente e correzione di eventuali mie imprecisioni.

Fornisco ora, a titolo di esempio, per evidenziare quale sia per me l'utilità di queste note, il problema *De quatuor hominibus qui invenerunt bizantios* che si trova nel foglio 12v.

Quattro uomini possiedono una certa quantità di bisanti:

a = primo uomo

b = secondo uomo

c = terzo uomo

d = quarto uomo

x = bisanti posseduti da a

y = bisanti posseduti da b

z = bisanti posseduti da c

w = bisanti posseduti da d

T = somma dei bisanti posseduti dai quattro uomini

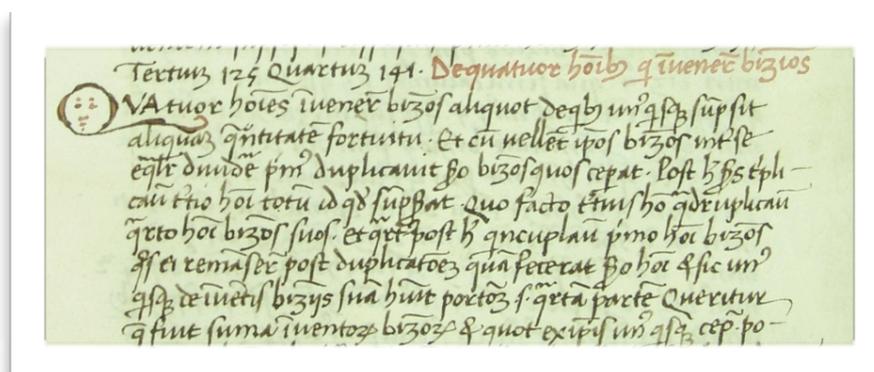


Figura 8 - Intestazione problema, foglio 12v

Ogni uomo dà ad un altro una certa somma, con questi risultati:

a dà y a **b** \rightarrow **a** possiede $(x-y)$ **b** possiede $2y$

b dà $2z$ a **c** \rightarrow **b** possiede $2(y-z)$ **c** possiede $3z$

c dà $3w$ a **d** → **c** possiede $3(z-w)$ **d** possiede $4w$

d dà $4(x-y)$ a **a** → **d** possiede $4(w-x+y)$ **a** possiede $5(x-y)$

Alla fine di questi “scambi”, ognuno possiede $T/4$ ovvero:

$$5(x-y) = 2(y-z) = 3(z-w) = 4(w-x+y) = T/4$$

A questo punto Leonardo Pisano esegue una serie di passaggi che portano fino alla soluzione:

1) **b** inizialmente possiede y , **a** dà y a **b** e quindi **b** possiede $2y$, **a** rimane con $(x-y)$. Poiché $5(x-y) = T/4$ allora $(x-y) = T/20$

2) **d** dà ad **a** $4(x-y) = 5(x-y) - (x-y) = T/4 - T/20 = T/5$

d, prima di dare ad **a**, possedeva $T/4 + T/5 = 9T/20$

Prima di ricevere $3w$ da **c**, **d** possedeva w , ovvero $(9T/20)/4 = 9T/80$

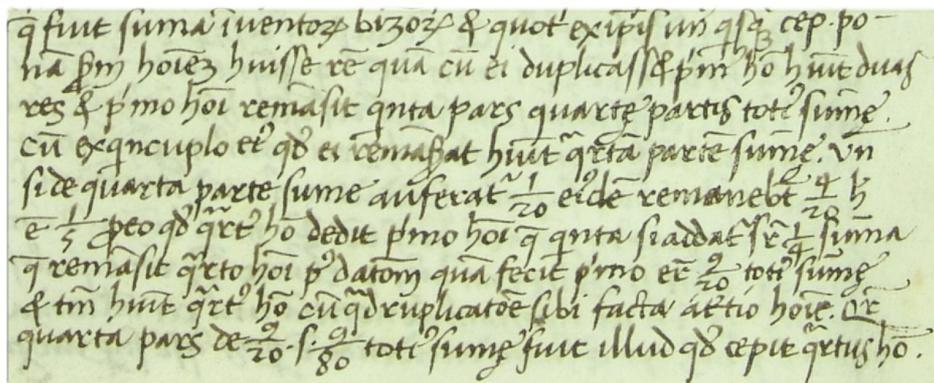


Figura 9 - Testo del problema, foglio 12v

3) **c** dà a **d** $3w = 3(9T/80) = 27T/80$

c, prima di dare a **d**, possedeva $T/4 + 27T/80 = 47T/80$

Prima di ricevere $2z$ da **b**, **c** possedeva z ovvero $(47T/80)/3 = 47T/240$

4) **b** dà a **c** $2z = 94T/240$

b, prima di dare a **c**, possedeva $T/4 + 94T/240 = 154T/240$

Prima di ricevere y da **a**, **b** possedeva y , ovvero $(154T/240)/2 = 77T/240$

5) prima di dare a **b**, **a** possedeva x

Dal primo passaggio si ricava che $x = T/20 + y = T/20 + 77T/240 = 89T/240$

A questo punto Leonardo pone $T = 240$, risolvendo le equazioni:

$$x = 89 \quad y = 77 \quad z = 47 \quad w = 27$$

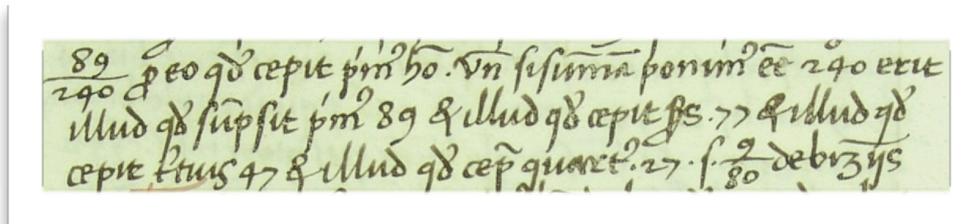


Figura 10 - Conclusione del problema, foglio 12v

2.3.2 - Visualizzazione dei problemi

Ho scelto di visualizzare ogni singola nota ai problemi in una pagina html, collegata ad EVT, come spiegherò nel capitolo successivo.

Per una maggiore comprensibilità, ho usato colori diversi per le variabili e ho riprodotto il *layout* di una pagina di quaderno a quadretti, per richiamare l'idea dello svolgimento di un problema matematico. Il risultato finale è quello che si vede nella figura 11.

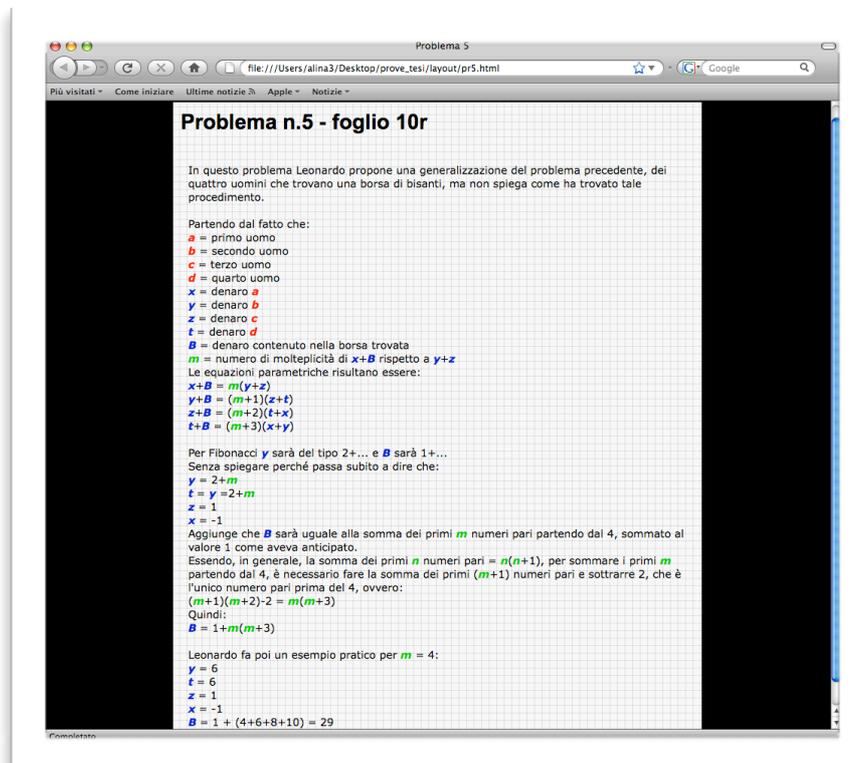


Figura 11 - Layout problema n.5

2.4 - EVT

2.4.1 - Che cosa è EVT

EVT è un *software* per la consultazione e lo studio di edizioni digitali, in particolare di manoscritti antichi e medievali. È stato ideato dal professor Rosselli Del Turco e implementato dalla dottoressa Fiorentini, all'interno del “*Vercelli Book Digitale*”, un progetto per la creazione di una versione digitale del *Codex Vercellensis*, manoscritto del X secolo contenente una miscellanea di opere a carattere religioso, in versi e in prosa.

Il codice del *software* è stato interamente scritto utilizzando il linguaggio di *mark-up* HTML, separando contenuti e grafica attraverso l'uso di fogli di stile CSS e completando la parte di programmazione con script Javascript, sfruttando una libreria *open source*, *Ext JS*.

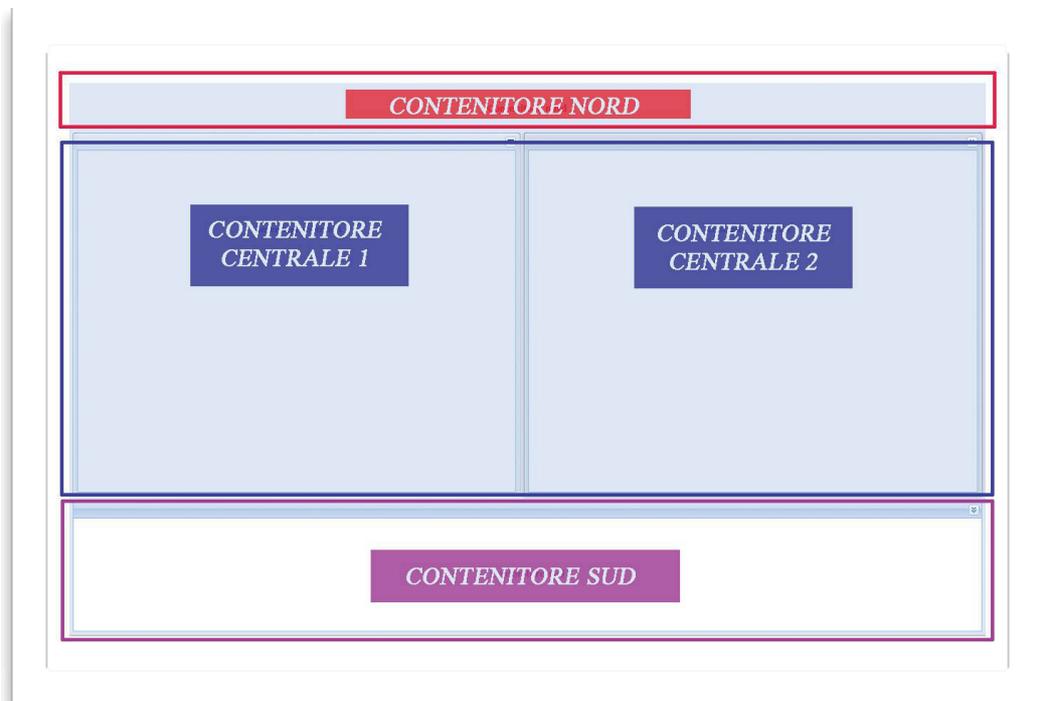


Figura 12 - Struttura dell'interfaccia

Come si vede nella Figura 12, l'interfaccia è composto da quattro regioni:

- area Nord: corrisponde alla sezione orizzontale che occupa l'*header* della pagina ed ospita il titolo e la barra dei pulsanti;
- area Centrale: suddivisa in due sotto-aree, corrisponde alla sezione centrale della schermata, dove vengono visualizzati i contenuti principali, ovvero immagini e trascrizione;
- area Sud: corrisponde alla sezione orizzontale che occupa il *footer* della pagina ed ospita le miniature delle immagini;

2.4.2 - Caricamento dei file in EVT

Nell'inserire i miei file in EVT ho effettuato alcune modifiche, tenendo presente che, a differenza del manoscritto per cui era stato creato, io avevo la necessità di visualizzare le immagini, la trascrizione del testo, sia in edizione diplomatica che critica, alcune note introduttive e i link alle pagine html contenenti le spiegazioni dei problemi matematici.

Per caricare i file è necessario copiarli nelle cartelle apposite e accedere al codice per modificare eventuali nomi e percorsi. Inoltre la versione di EVT che ho utilizzato non permette di accedere all'intero file XML che contiene la trascrizione e visualizzarne solamente le parti relative all'immagine caricata in quel momento, quindi ho dovuto creare un file XML per ogni pagina di manoscritto.

Per prima cosa mi sono occupata delle immagini, fornitemi dalla Biblioteca Ambrosiana: dopo averle salvate nei formati di dimensione richiesta dal programma, le ho copiate nelle cartelle e ho inserito nel codice javascript i nomi corretti.

La stessa cosa ho fatto per i file XML relativi alla trascrizione in edizione diplomatica ed ho modificato il foglio di stile XSL per adattarlo ai miei contenuti.

Il risultato, dopo queste modifiche è quello nella Figura 13:

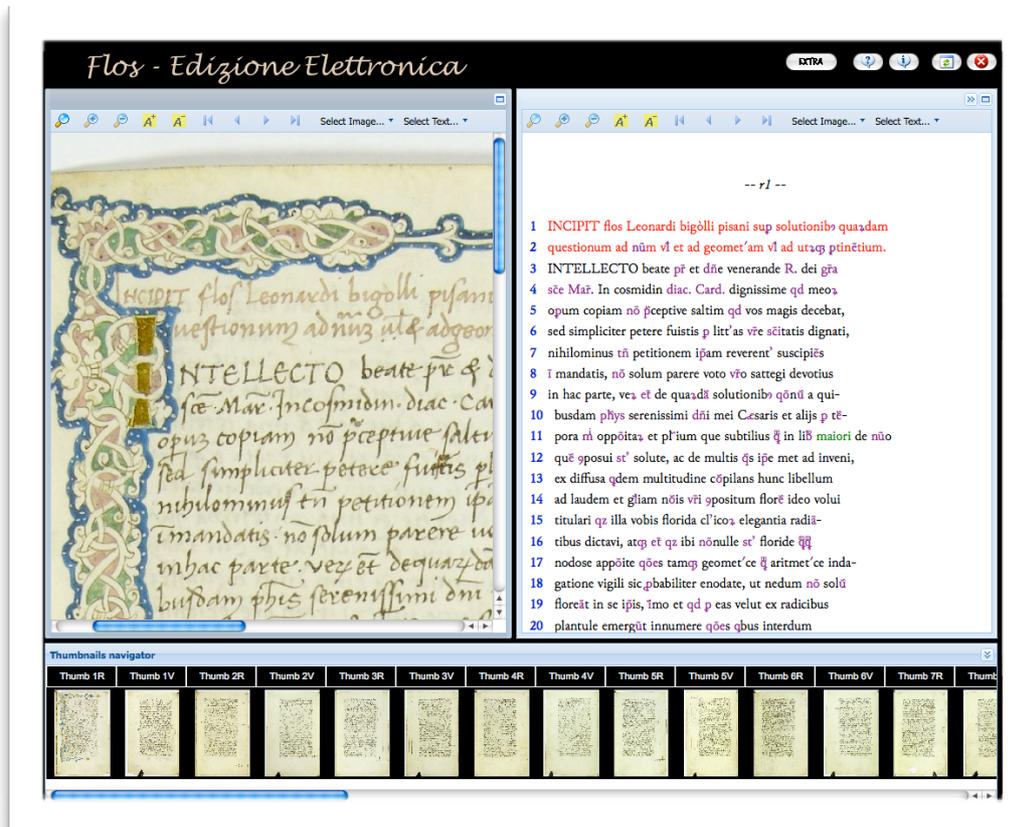


Figura 13 - Risultato del caricamento dei primi file in EVT

Nell'area in basso vengono mostrate le miniature delle immagini, che permettono la navigazione: selezionando una miniatura appare l'immagine grande a sinistra e la relativa trascrizione a destra.

Come si vede nella figura 13, le abbreviazioni paleografiche sono evidenziate in viola, mentre in verde le parti di testo aggiunte a margine o nell'interlinea. Per visualizzare correttamente i caratteri utilizzati nelle abbreviazioni è indispensabile aver salvato nella libreria font del sistema operativo in uso il font Junicode.

Nella barra degli strumenti posta sopra ciascun'area centrale, ci sono due pulsanti, "Select Image" e "Select Text", che aprono un menù a tendina che permette la navigazione dei file caricati in EVT, offrendo anche la possibilità di vedere contemporaneamente due testi o due immagini.

Ho deciso di sfruttare questa caratteristica per visualizzare i file dell'edizione critica, con le correzioni e le abbreviazioni sciolte, consentendo all'utente di selezionare dal menù a tendina (vedi Figura 14) il file dell'edizione critica corrispondente al testo visualizzato nell'area di destra.

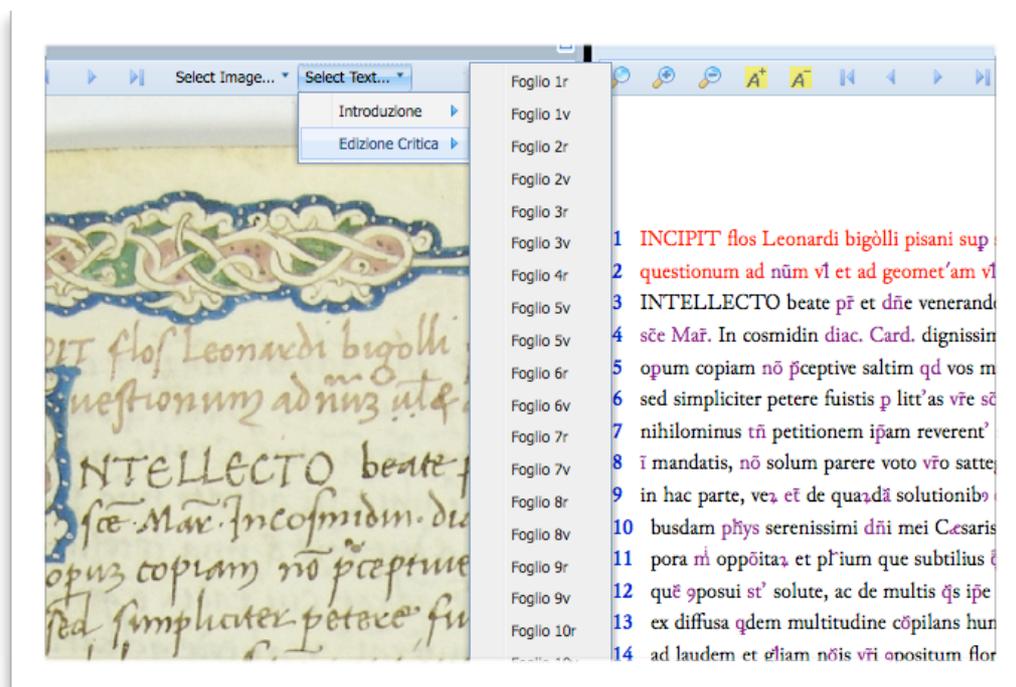


Figura 14 - Modifica del menù a tendina

Per ottenere questo risultato ho dovuto modificare la parte del codice relativa al menù e creare un nuovo foglio di stile XSL. Rispetto a quello descritto nel paragrafo 2.2.2 ho modificato la seguente trasformazione:

```
<xsl:template match="tei:choice">
<xsl:apply-templates select="tei:abbr"/>
<xsl:apply-templates select="tei:sic"/>
</xsl:template>
in
<xsl:template match="tei:choice">
<xsl:apply-templates select="tei:expan"/>
<xsl:apply-templates select="tei:corr"/>
</xsl:template>
```

A questo punto, selezionando la voce di menù relativa al foglio 1r, la cui versione diplomatica è già visibile sulla destra, si potranno visualizzare le due trascrizioni, come nella figura 15.

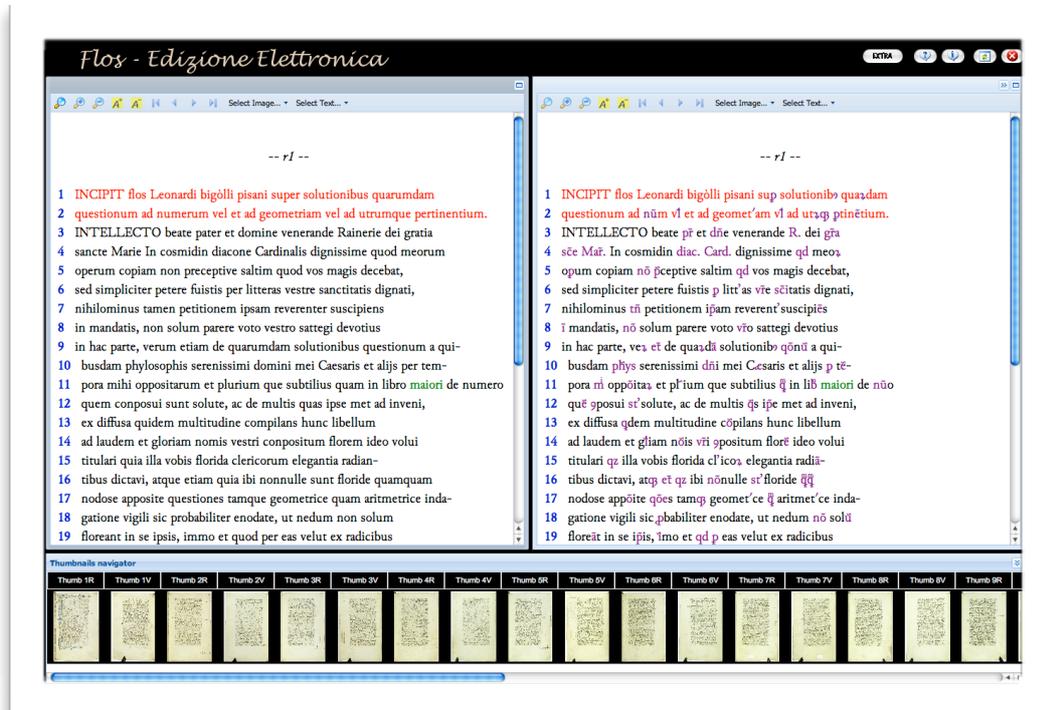


Figura 15 - Le due edizioni visibili contemporaneamente

Come si vede nella figura 14, dal pulsante “Select Text” si può anche selezionare la voce “Introduzione”, per visualizzare i file XML relativi alla descrizione della storia, dell’aspetto e del contenuto del manoscritto.

Nell’area Nord, a destra del titolo ci sono una serie di pulsanti (vedi figura 16) con diverse funzioni.



Figura 16 - Barra degli strumenti

Gli ultimi due bottoni servono per aggiornare o chiudere la finestra del *browser* contenente EVT.

Il pulsante “EXTRA” apre una finestra contenente dei link ai contenuti speciali.

Ho sfruttato questa funzione per creare i riferimenti alle note matematiche.

Cliccando infatti su uno dei link di tipo “Problema” si apre una nuova scheda del *browser* con il quesito scelto (per il *layout* vedi il paragrafo 2.3.2).

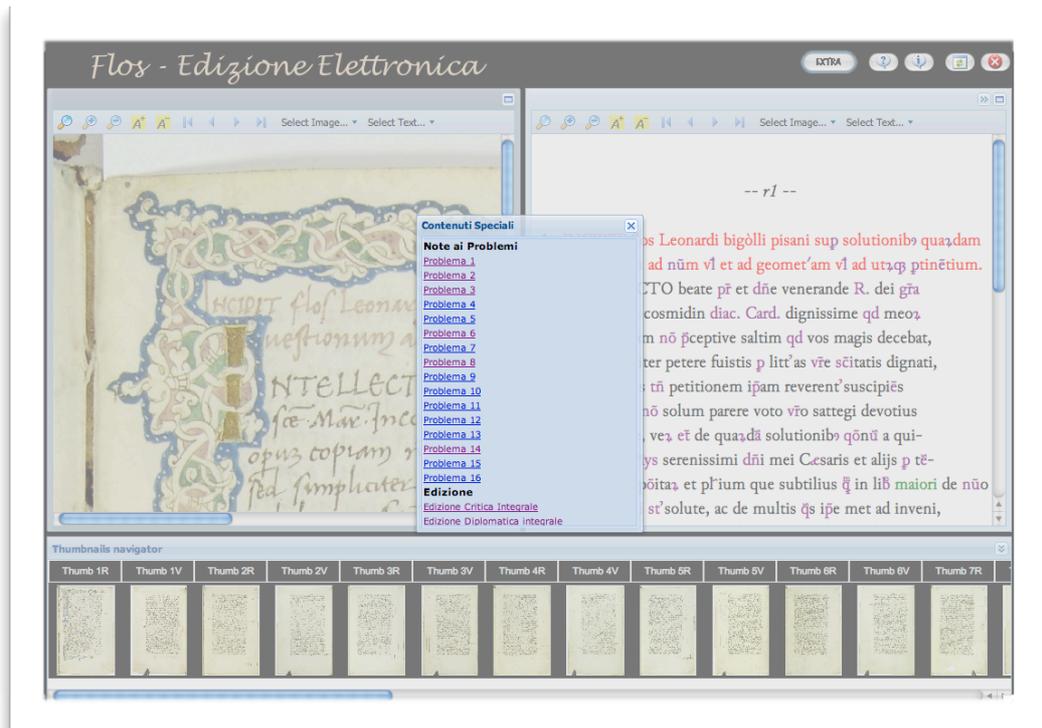


Figura 17 - Menu

Tra i contenuti speciali ho inserito anche la trascrizione intera del *Flos*, sia in edizione critica che diplomatica, in modo che si possa avere anche una visione d’insieme del testo: cliccando su “Edizione Critica Integrale” o “Edizione Diplomatica Integrale”, si apre una nuova scheda del *browser* con il testo selezionato, come si può vedere nelle figure 18 e 19, nella pagina successiva.

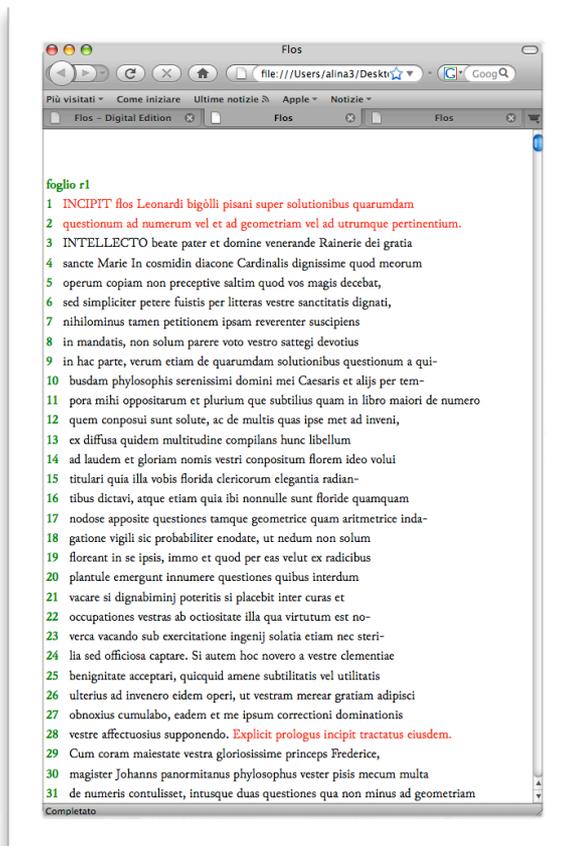


Figura 18 - Edizione critica

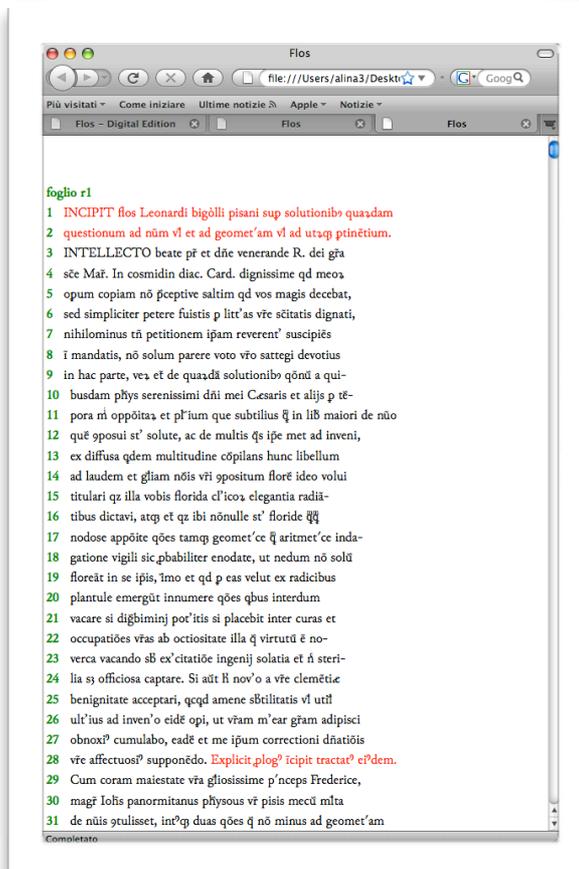


Figura 19 - Edizione diplomatica

Il secondo pulsante apre una finestra con informazioni riguardo allo scopo del progetto e alle istruzioni per la navigazione del *Flos*, in particolare per l'utilizzo della lente ingrandimento e della barra delle miniature.

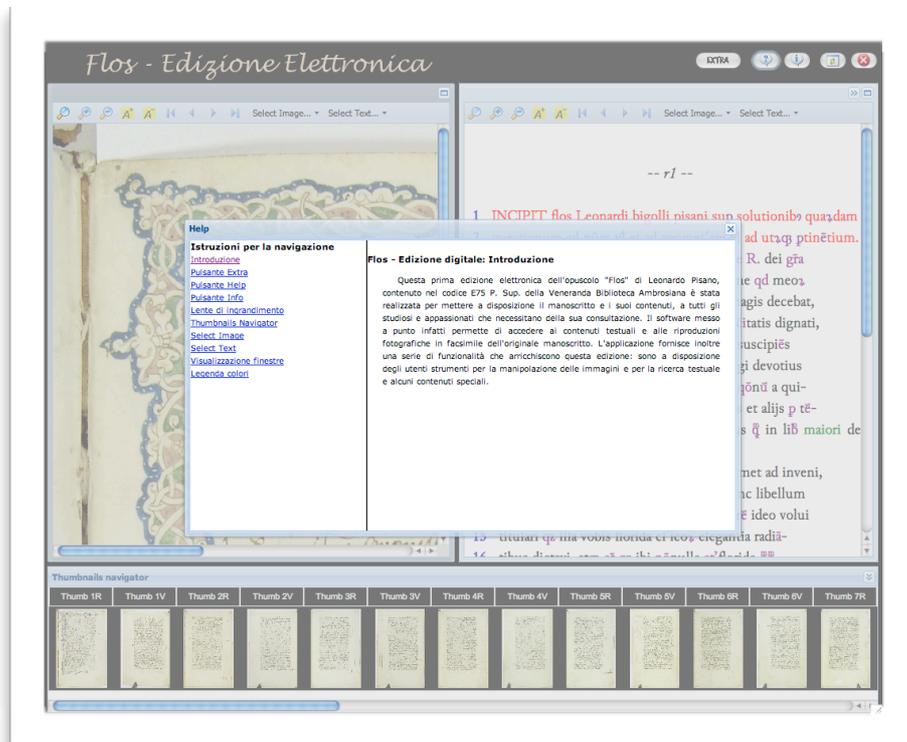


Figura 20 - Finestra Help

Il terzo bottone apre una piccola finestra contenente i dati del *Flos*, del manoscritto che lo contiene e dell'edizione digitale (autore, *designer*, *publisher* ecc.) come si può vedere nella figura 21.



Figura 21 - Finestra Info

CONCLUSIONI

Tirando le somme alla fine di questo progetto lungo e minuzioso, non posso che essere soddisfatta del risultato.

Prima di tutto ho creato l'edizione di un manoscritto che, storicamente, è stato maneggiato da pochissime persone, a differenza del ben più conosciuto *Liber Abbaci*, ma che, nonostante la scarsa diffusione, contiene delle novità algebriche così importanti che hanno lasciato un segno nella storia.

Sono venuta in contatto con un linguaggio matematico lontano anni luce da quello moderno, ma proprio per questo l'essere arrivata alla comprensione dei quesiti è stata una grande soddisfazione.

Un altro particolare importante è che sento di aver racchiuso in questo progetto quello che è il senso e lo scopo di Informatica Umanistica, ovvero riuscire a padroneggiare strumenti informatici per la gestione di qualsiasi tipo di contenuti umanistici; in particolare ho utilizzato le conoscenze informatiche acquisite durante questi tre anni per la fruizione, la conservazione e la divulgazione di un testo antico.

Senza dubbio, la soddisfazione più grande è che il mio lavoro non finirà dimenticato in un cassetto, ma avrà un'illustre collocazione: sarà infatti pubblicato sul sito della Veneranda Biblioteca Ambrosiana (dopo alcune necessarie migliorie a EVT) e sarà accolto nell'archivio della stessa, in modo tale che, chiunque chiederà in futuro di consultare il *Flos*, potrà sfogliare il mio elaborato elettronico. In questo modo si limiterà il contatto diretto tra il manoscritto, unico per i documenti che contiene, e le maggiori cause di deterioramento (polvere, contatto fisico, luce ecc.), garantendone la conservazione nel tempo.

Non posso che ringraziare l'Ambrosiana, in particolare il dottor Francesco Braschi, per avermi dato questa opportunità e per avermi fornito le immagini a titolo gratuito (Aut. G0018/08).

BIBLIOGRAFIA

Andrei, Giovanni. 1832. *Dell'origine, progressi e stato attuale di ogni letteratura*. Venezia, Giuseppe Antonelli Editore.

Antoni, Tito. 1994. *Leonardo Pisano detto il Fibonacci e lo sviluppo della contabilità mercantile del Duecento*. In “Leonardo Fibonacci: il tempo, le opere, l’eredità scientifica. Atti del Congresso internazionale di Pisa”. Pisa, Ed. Pacini.

Arrighi, Gino. 1994. *Considerazioni sul Liber Abaci di Leonardo Pisano*. In “Leonardo Fibonacci: il tempo, le opere, l’eredità scientifica. Atti del Congresso internazionale di Pisa”. Pisa, Ed. Pacini.

Bonaini, Francesco. 1858. *Memoria unica sincrona di Leonardo Fibonacci*, Pisa.

Boncompagni, Baldassarre. 1852. *Della vita e delle opere di Leonardo Pisano*. Roma, Tipografia delle Belle Arti.

Boncompagni, Baldassarre. 1854. *Intorno ad alcune opere di Leonardo Pisano*. Roma, Tipografia delle Belle Arti.

Boncompagni, Baldassarre. 1854. *Tre scritti inediti di Leonardo Pisano*. Firenze, Tipografia Galileiana.

Boncompagni, Baldassarre. 1856. *Opuscoli di Leonardo Pisano*. Firenze, Tipografia Galileiana.

Lucas, Par M. Edouard Lucas. 1877. *Recherches sur plusieurs ouvrages de Léonard de Pise et sur diverses questions d'arithmétique supérieure*. “Bollettino

di bibliografia e di storia delle scienze matematiche e fisiche” - tomo X. Roma, Tipografia delle scienze matematiche e fisiche.

Mazzatinti, Giuseppe (a cura di). 1894. *Inventari dei manoscritti delle biblioteche d'Italia*. Forlì.

Paredi, Angelo. 1981. *Storia dell'Ambrosiana*. Vicenza, Neri Pozza Editore.

Picutti, Ettore. 1983. *Il “Flos” di Leonardo Pisano*. “Physis”, 25, fasc. 2, p. 293-387.

TEI Consortium. 2008. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, Oxford – Providence – Charlottesville – Nancy.

1790. *Memorie istoriche di più uomini illustri pisani*, tomo I. Pisa, Tipografia Ranieri Prosperi.

SITOGRAFIA

(data ultima visita ai siti: 2 febbraio 2009)

Articoli e notizie storiografiche

Biblioteca Pinacoteca Accademia Ambrosiana

<http://www.ambrosiana.it/>

La Navigation du Savoir

http://www.navigationdusavoir.net/PortalPisa/seafarers/medievale_saperi.html

University of Evansville

<http://faculty.evansville.edu/ck6/bstud/fibo.html>

Storia Moderna

http://www.stmoderna.it/rassegna_quotidiani_dettaglio.asp?id=48

Stupor Mundi

<http://www.stupormundi.it/Fibonacci.htm>

Il giardino di Archimede - Un Museo per la Matematica

http://web.math.unifi.it/archimede/archimede/note_storia/numeri/numeri1/node5.html

<http://neo.math.unifi.it/archimede/archimede/fibonacci/catalogo/pepe.php>

BASE CINQUE - Appunti di Matematica Ricreativa

<http://utenti.quipo.it/base5/fibonacci/fibonacci.htm>

Plus Magazine

<http://plus.maths.org/issue3/fibonacci/index.html>

Vialattea.net - Divulgazione Scientifica

<http://www.vialattea.net/esperti/mat/fibonacci/>

No-Thing.it - Il Portale del Sapere in Podcasting

<http://www.no-thing.it/leonardo-fibonacci.php>

Medioevo

<http://www.medioevo.com/index.php?>

[option=com_medioevocontent&task=view&id=278&Itemid=35&limit=1&limitstart=3&lang=it](http://www.medioevo.com/index.php?option=com_medioevocontent&task=view&id=278&Itemid=35&limit=1&limitstart=3&lang=it)

Codifica ed esempi di annotazione di testi antichi

Consorzio TEI

<http://www.tei-c.org/index.xml>

Griseldaonline - Portale di Letteratura

http://www.griseldaonline.it/informatica/manuale_parte6_3.htm

Vercelli Book Digitale

<http://islp.di.unipi.it/bifrost/vbd/>

Biblioteca Italiana

<http://www.bibliotecaitaliana.it/>

Digital Medievalist

<http://www.digitalmedievalist.org/>

Medieval Unicode Font Initiative

<http://www.mufi.info/>

Junicode

<http://junicode.sourceforge.net/>

World Wide Web Consortium

<http://www.w3.org/>

The Unicode Consortium

<http://www.unicode.org/>

Scholarly Digital Editions

<http://www.sd-editions.com/>

Evellum.com - Medieval Multimedia - Digitizing the Middle Ages

<http://www.evellum.com/>

RHC - Collaboratory for Research in Computing for Humanities

<http://www.rch.uky.edu/29r/>

A.Cappelli - Dizionario di Abbreviature Latine ed Italiane

<http://www.hist.msu.ru/Departments/Medieval/Cappelli/index.html>

XSLT Tutorial

<http://www.w3schools.com/xsl/>

Software

Oxygen XML Editor

<http://www.oxygenxml.com/>

Editix XML Editor

<http://www.editix.com/>

I.R.I.S. - OCR software and Document Management Solutions

<http://www.irislink.com/>

NeoOffice

<http://www.neooffice.org/neojava/it/index.php>

GIMP - the GNU Image Manipulation Program

<http://www.gimp.org/>

iWork09 - Pages

<http://www.apple.com/it/iwork/pages/>