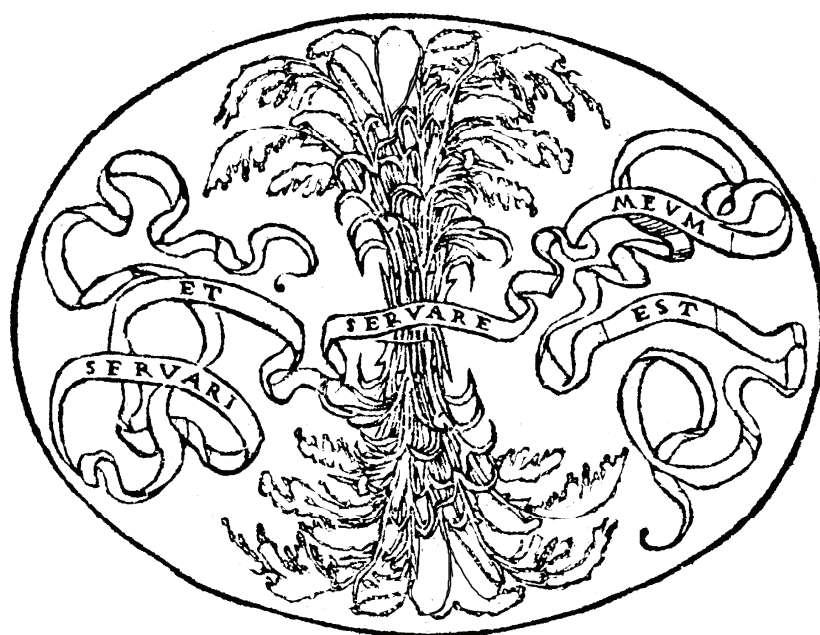


STUDI  
DI  
**MEMOFONTE**

*Rivista on-line semestrale*

6/2011



FONDAZIONE MEMOFONTE

*Studio per l'elaborazione informatica delle fonti storico-artistiche*

[www.memofonte.it](http://www.memofonte.it)

## COMITATO REDAZIONALE

*Proprietario*

Fondazione Memofonte onlus

*Direzione scientifica*

Paola Barocchi

Miriam Fileti Mazza

*Cura redazionale*

Claudio Brunetti, Irene Calloud, Elena Miraglio

*Segreteria di redazione*

Fondazione Memofonte onlus, Lungarno Guicciardini 9r, 50125 Firenze

[info@memofonte.it](mailto:info@memofonte.it)

ISSN 2038-0488

## INDICE

M. Fileti Mazza, <i>Editoriale</i>	p. 1
D. Levi, «Perdonate alle ripetizioni»: <i>elaborazione di una tecnica descrittiva nelle carte private di G.B. Cavalcaselle</i>	p. 3
E. Pellegrini, <i>I taccuini di Adolfo Venturi</i>	p. 13
E. Federighi, <i>Adolfo Venturi e la città di Budapest</i>	p. 39
I. Calloud, <i>Ugo Ojetti e le esposizioni; un'anagrafe digitale dal Fondo della Biblioteca Nazionale Centrale di Firenze</i>	p. 53
E. Miraglio, <i>Seicento, Settecento, Ottocento e via dicendo: Ojetti e l'arte figurativa italiana</i>	p. 63
M. Dei, <i>Ojetti e l'Exposition de l'art Italien de Cimabue à Tiepolo di Parigi</i>	p. 81
A. De Santis, G. Marotta, <i>Cesare Brandi: cronache e recensioni delle attività espositive tra 1932 e 1986. Aspetti e metodologie</i>	p. 91
C. Gamba, <i>L'Archivio privato di Giulio Carlo Argan. Introduzione alla schedatura della corrispondenza ed esempi di materiali epistolari</i>	p. 121
K. Quinci, <i>L'interesse di Giulio Carlo Argan per la fruizione pubblica delle collezioni private degli artisti: il caso del lascito di Lucio Fontana</i>	p. 133
A. Del Bimbo, A. Ferracani, V. Lepera, G. Serra, <i>Da Cavalcaselle ad Argan: un'applicazione web per la fruizione di testimonianze di cultura artistica e letteraria</i>	p. 159



## DA CAVALCASELLE AD ARGAN: UN'APPLICAZIONE WEB PER LA FRUIZIONE DI TESTIMONIANZE DI CULTURA ARTISTICA E LETTERARIA

### *Introduzione*

*Da Cavalcaselle ad Argan: archivio per la cultura artistica e letteraria* è un portale web di ricerca e *browsing* innovativo su documenti testuali, manoscritti o a stampa, relativi ad alcune delle figure principali della storiografia e della critica d'arte italiana dei secoli XIX e XX: Giovan Battista Cavalcaselle (1819-1897), Adolfo Venturi (1856-1941), Ugo Ojetti (1871-1946), Giulio Carlo Argan (1909-1992) e Cesare Brandi (1906-1988).

Il lavoro preliminare di ricerca dei materiali attualmente presenti nel portale, ha portato al ritrovamento di testi per lo più inediti quali taccuini, saggi, corrispondenze, note ecc. Al contempo l'attività di ricerca ha contribuito alla riorganizzazione, l'inventariazione e la catalogazione di tali fonti.

Gli strumenti offerti dal portale forniscono agli utenti ed agli studiosi una visione complessiva dell'ambiente e delle relazioni culturali, nonché del mondo editoriale dell'epoca in cui i cinque critici svolsero la loro attività.

I materiali principali presenti nel sistema sono costituiti da fonti ed eventi che riguardano documenti relativi a mostre, viaggi, eventi culturali, cui parteciparono tali personaggi, e la relativa corrispondenza pubblica o privata, e ne descrivono alcune caratteristiche rilevanti quali luoghi, date, produttori, curatori ecc.

L'applicazione web consente di cercare e filtrare tali documenti attraverso i metadati associati oltre che per mezzo di vocabolari controllati che sono stati definiti ed inclusi nel sistema (luoghi, mostre, movimenti e termini artistici, opere, persone e testi), selezionando i termini da vocabolari e thesauri nazionali ed internazionali o aggiungendone di specifici.

Il sistema comprende, infatti, un'interfaccia per la ricerca semplice ed una per la ricerca avanzata, un *content management system* per la gestione dei contenuti ed un sistema automatico per l'annotazione semantica dei testi e la visualizzazione avanzata dei documenti collocati nel tempo e nello spazio.

### *Applicazione Web*

L'archivio web si basa su un'architettura *Model View Controller* ed utilizza il *framework Symfony*<sup>1</sup>.

L'applicazione è attualmente pubblicata all'indirizzo <http://www.docart900.memofonte.it>.

Il sistema è composto da tre parti: presentazione generale del progetto e delle sue finalità, *content management system* per la pubblicazione e la modifica dei documenti, motore di ricerca semplice ed avanzato per la ricerca e la visualizzazione dei dati.

Il *frontend* dell'applicazione (Fig. 1) offre una presentazione sintetica degli archivi dei cinque critici su cui si incentra il *corpus* dei documenti attraverso una biografia essenziale, una galleria d'immagini ed una bibliografia generale. Gli schedatori, dopo una fase di autenticazione, possono accedere al *backend* del sistema, costituito da una serie di moduli che forniscono funzionalità di creazione, reperimento, modifica e cancellazione dei documenti (secondo il paradigma di persistenza *CRUD: Create, Retrieve, Update, Delete*).

Le due principali entità d'informazione inseribili e reperibili nel sistema sono le Fonti e gli Eventi. L'applicazione consente inoltre di inserire e associare immagini e foto sia alle Fonti (ad esempio la riproduzione digitale di un taccuino) che agli Eventi.

---

<sup>1</sup> Symfony, web PHP Framework, <http://www.symfony-project.org/>.

Il modulo di ricerca è composto da due componenti: la ricerca *full-text* e la ricerca avanzata. La ricerca *full-text* consente all'utente di ricercare quanto desiderato fornendo esclusivamente una chiave di ricerca. Esempi famosi di motori di ricerca *full-text* sono Google e Yahoo!.



Figura 1. Il *frontend* dell'applicazione.

Il motore di indicizzazione dei dati utilizzato all'interno dell'applicazione web per la ricerca *full-text* è basato su *Lucene*<sup>2</sup>. *Lucene* è una libreria (API) gratuita e *open source* estremamente flessibile ed adattabile ad ogni esigenza di ricerca. L'interfaccia di ricerca *full-text* (Fig. 2, in alto) offre la possibilità di scegliere se effettuare la ricerca su tutti o solo alcuni degli archivi documentali (Fonti ed Eventi di Cavalcaselle, Venturi, Ogetti, Argan e Brandi) e presenta una semplice casella di testo nella quale l'utente può inserire uno o più termini.



Figura 2. L'interfaccia di ricerca *full-text* e l'elenco dei risultati.

Nel caso l'utente voglia effettuare ricerche complesse può avvalersi di una sintassi completa per la creazione di *query*: può per esempio ricercare un'intera frase inserendo il testo fra virgolette (ad esempio: "Esposizione nazionale del Correggio"), oppure, all'interno delle

<sup>2</sup> MCCANDLESS-HATCHER-GOSPODNETIĆ 2010.

*query* di ricerca, può utilizzare operatori booleani (AND, OR, NOT) e le classiche *wildcard*? (un solo carattere) e \* (*n* caratteri).

I risultati della ricerca vengono raggruppati prima per archivio e poi per tipologia, in modo da restituire una visione d'insieme sul contenuto del database riguardo alla specifica *query* (Figura 2, in basso).

La ricerca avanzata invece fornisce la possibilità di effettuare interrogazioni applicando uno o più *set* di filtri generici e specifici relativi alle due macrocategorie Fonte ed Evento (Fig. 3).

Figura 3. L'interfaccia di ricerca avanzata.

I filtri generici permettono di filtrare i risultati per archivio di riferimento ed intervallo temporale.

I filtri specifici per la ricerca di tipo Fonte consentono di filtrare i documenti per tipologia e produttore. Al contempo è possibile effettuare una ricerca testuale nei campi titolo, regesto e/o trascrizione. Possono inoltre essere selezionati uno o più argomenti scelti da vocabolari controllati gestiti da un sistema tassonomico ed utilizzati dai ricercatori in fase di catalogazione. Un esempio di ricerca per fonte potrebbe essere quello di individuare tutte le recensioni scritte da Brandi sulle mostre di Picasso.

I filtri specifici per la ricerca Evento, invece, permettono di filtrare i documenti per tipologia, denominazione e luogo, per esempio, è possibile cercare tutte le mostre svoltesi a Firenze negli anni Trenta, sempre legate ai protagonisti della ricerca.

È presente infine una terza modalità, Fonti ed Eventi in relazione, che consente di incrociare i filtri in modo da realizzare interrogazioni più complesse. Ad esempio, è possibile cercare le recensioni scritte da Cesare Brandi che trattano di mostre svoltesi a Firenze.

L'interfaccia di ricerca avanzata è stata sviluppata secondo il paradigma delle *Rich Internet Applications*. Selezionando il valore di una qualsiasi casella, infatti, vengono aggiornate le opzioni di tutti gli altri elementi della maschera in base ai valori immessi fino a quel momento e coerentemente con il contenuto del database senza ricaricare la pagina stessa. Adottando

questa soluzione si risparmiano all'utente i tempi morti causati da ricerche che non producano risultati.

La visualizzazione dei risultati mostra per ogni risorsa trovata l'archivio, la tipologia della fonte, la segnatura, il titolo, la data, i produttori ed i destinatari, e permette all'utente di capire velocemente il contenuto delle risorse identificate dal motore di ricerca.

I documenti vengono visualizzati con una impaginazione tabulare divisa in sezioni (Fig. 4). Di particolare interesse è la sezione relazioni, che presenta i collegamenti alle fonti ed agli eventi associati al documento in esame.

Nella parte superiore di ogni scheda, infine, sono presenti lo strumento di navigazione detto a briciole di pane (*breadcrumbs*) che consente di scorrere la lista dei risultati e la numerazione per la paginazione dei stessi.

The screenshot shows a web interface with a breadcrumb trail at the top: "sei in: lista archivi > Brandi > Mostra della pittura riminese del Trecento". To the right of the breadcrumbs is a link "ritorna ai risultati della ricerca" and a "stamp" button. Below this is a table with the following data:

Dati Evento	
Tipologia Evento	mostra
Denominazione	Mostra della pittura riminese del Trecento
Luogo/Sede	Rimini, Palazzo dell'Arengo
Data	20 giugno 1935 - 30 settembre 1935
Comitato	<b>Comitato generale:</b> De Vecchi Cesare Maria, D'Andrea Massimiliano, Borri Dino, Scozzoli Vincenzo, Venturi Adolfo, Baiocchi Pio, Tricarico Pietro, Onesti Calzecchi Carlo <b>Membri:</b> Bianchini Eugenio, Cassoli Carlo, Lucchesi Carlo <b>Comitato esecutivo. Presidenza:</b> Mattioli Guido Alberto, Onesti Calzecchi Carlo <b>Direzione della mostra:</b> Brandi Cesare
Pubblicazione	Brandi C., Mostra della pittura riminese del Trecento, Stabilimento Tipografico Garattoni, Rimini 1935

Below the table is a section titled "Relazioni" with a sub-section "Fonti" containing three links:

- Brandi, La Critica d'Arte, 1937
- Brandi, Mostra della pittura riminese del Trecento, 1935
- Brandi, Pantheon, 1935

Figura 4. Visualizzazione di un documento.

#### *Analisi automatica del testo*

L'applicazione fa uso di un algoritmo per l'annotazione automatica dei contenuti dei documenti inseriti. Lo sviluppo di tale metodologia permette di aiutare lo schedatore nella fase di soggettazione dei contenuti delle fonti; fase estremamente lunga soprattutto per tipologie quali lettere o taccuini.

L'algoritmo ha come obiettivo quello di fornire alcune rappresentazioni semantiche del contenuto della collezione di testi oggetto di studio (*corpus*) e di estrarre da essi l'informazione significativa, ovvero i lessemi caratteristici del contenuto espresso nel *corpus* ed in particolare dei campi regesto e trascrizione di ciascun documento che costituiscono la maggior parte dell'informazione testuale.

Poiché in fase di schedatura dei documenti, i campi regesto e trascrizione vengono inseriti nel database tramite un *editor* visuale avanzato integrato nell'applicazione che consente di formattare il testo ed inserire oggetti multimediali convertendo gli *input* dello schedatore in *tag HTML*, il primo passo prima di procedere al processo di estrazione consiste, pertanto, nell'analisi dei documenti allo scopo di rilevare e rimuovere eventuali *tag HTML* che non presentino elementi semantici rilevanti. Il compito è delegato alla libreria java *open source Jericho Parser*<sup>3</sup> che permette l'analisi, la manipolazione e l'eliminazione del codice *HTML* non

<sup>3</sup> Jericho Parser, <http://jericho.htmlparser.net/docs/index.html>.



necessario, lasciando inalterati il testo semplice e il codice non riconosciuto. La libreria è in grado di identificare con alta precisione un gran numero di linguaggi (*HTML*, *ASP*, *JSP*, *PSP*, *PHP* etc.) pur rimanendo computazionalmente efficiente. Tra le funzionalità di *Jericho* si annoverano inoltre metodi appositamente studiati per assicurare la compatibilità con i più comuni motori di ricerca testuale.

Dopo che il testo è stato ‘ripulito’, l’algoritmo procede all’identificazione delle *keywords*. Non tutte le parole hanno naturalmente la stessa importanza; e non è la frequenza l’unico elemento a determinare il peso di un termine in un testo. Anche le parole che occorrono una sola volta possono avere un alto valore semantico. Molte fra le parole più frequenti in un testo però sono ‘parole vuote’ quali articoli, preposizioni e congiunzioni (per esempio <e>, <di>, <da>, <il> ecc.), dette anche *stop word*. L’algoritmo quindi elimina tali parole utilizzando la libreria Java open source *Lucene*, che, nata come indicizzatore per la creazione di motori di ricerca testuale, offre svariati metodi per l’analisi e la manipolazione del testo. Nello specifico l’algoritmo suddivide il testo in *tokens*, che vengono poi confrontati con una lista predefinita di parole non significative tipiche della lingua italiana.

A questa fase segue l’operazione di *stemming*: si tratta di un processo che consente la riduzione della forma flessa di una parola alla sua forma radice, detta tema. Il tema non corrisponde necessariamente alla radice morfologica (lemma) della parola: normalmente è sufficiente che le parole correlate siano mappate allo stesso tema, ad esempio ‘andare’, ‘andai’ e ‘andò’ mappano al tema ‘and’, anche se quest’ultimo non è una valida radice per la parola.

L’algoritmo applica poi al testo il così detto *PoS Tagging*, ovvero un processo di *tagging* grammaticale (*part of speech tagging*), diffuso in linguistica computazionale, che consente di identificare e disambiguare la categoria di appartenenza delle parole in base al contesto in cui esse si trovano (parole adiacenti e collegate), riuscendo a classificarle quali nomi, verbi, aggettivi, avverbi etc. L’analisi viene svolta dalla libreria *TreeTagger*<sup>4</sup>, rilasciata gratuitamente dall’Istituto di Linguistica Computazionale dell’Università di Stoccarda e basata sul metodo del *Decision Tree*. Il *tool*, usato con successo per l’analisi in diverse lingue (Tedesco, Inglese, Francese, Italiano, Spagnolo ecc.), integra al suo interno uno *stemmer* ed un analizzatore grammaticale ed è stato utilizzato al fine di eliminare dai testi tutti i verbi, non ritenuti rilevanti per l’annotazione dei contenuti.

Una volta eseguite le procedure di eliminazione delle *stop word*, lo *stemming* ed il *PoS Tagging*, l’algoritmo estrae le *keywords* con un approccio *corpus based* che è noto come *Term Frequency - Inverse Document Frequency (TFIDF)*. Tale approccio consente di calcolare un peso per ogni termine facente parte del vettore delle *keywords* del documento e fornisce informazioni sulla rilevanza di ogni parola basandosi su due assunti: a) tanto più un termine occorre in un documento tanto più è rappresentativo del suo contenuto; b) tanti più documenti contengono un termine, tanto meno questo è discriminante. Si tratta dei concetti di peso locale e peso globale del *tag*. Il peso locale indica la rilevanza del termine rispetto al testo contenuto nella singola fonte. Oltre a calcolare la frequenza di ogni elemento all’interno del documento di riferimento (*TF*, *Term Frequency*), l’algoritmo la relaziona con la frequenza dell’elemento nell’intero universo dei documenti (*IDF*, *Inverse Document Frequency*) assegnando, in questo modo, un peso globale al termine. Ciò permette di ovviare alle difficoltà nell’individuazione delle parole semanticamente rilevanti all’interno del contesto complessivo del *corpus* dei documenti e quindi anche nei singoli testi. Tali difficoltà sono dovute ad un principio ben noto in linguistica secondo il quale, all’interno di un documento, sono presenti poche parole con alta frequenza e molte parole con bassa frequenza<sup>5</sup>. Tale principio, detto ‘del minimo

---

<sup>4</sup> SCHMID 1994.

<sup>5</sup> LI 1992.

sforzo' afferma che, in una comunicazione, scritta o parlata, è più facile ripetere parole di uso comune piuttosto che usarne di nuove<sup>6</sup>.

Si ottiene dunque una regola di distribuzione dei pesi per cui il peso più alto viene assegnato a parole con un'occorrenza molto elevata all'interno di un ristretto numero di documenti mentre il peso decresce proporzionalmente alla presenza del termine in un documento ed al numero di documenti in cui il termine appare. L'implementazione dell'algoritmo si basa ancora una volta sulle librerie di *Lucene*, che forniscono metodi per l'indicizzazione e il calcolo dei pesi locali e globali.

#### *Semantica e visualizzazioni avanzate*

Il modello dell'archivio web, compresi i risultati ottenuti nel processo di estrazione delle *keywords*, è costituito da un database relazionale che non possiede di per se espressività semantica. Allo scopo di fornire un corrispettivo semantico del *knowledgebase* dell'applicazione è stata utilizzata la piattaforma *D2RQ*<sup>7</sup>. *D2RQ* consente di mappare i campi di un *database* relazionale ai termini di una ontologia (*RDFS/ OWL*)<sup>8</sup>, consentendo in questo modo all'applicazione di accedere ad una vista *RDF* del modello attraverso le *API* di *Jena*<sup>9</sup>, oppure via web tramite *query SPARQL*<sup>10</sup>. Un'ontologia consta di un insieme di concetti e proprietà e descrive le loro relazioni fornendo così una descrizione formale di un dominio di conoscenza utilizzabile da un programma<sup>11</sup>. Al momento l'implementazione integrata nel sistema attinge ad un *set* di dati generati *offline* sul *database* attraverso un file di *mapping* di cui si fornisce un estratto:

```
# Table evento
```

```
map:Evento a d2rq:ClassMap ;
d2rq:dataStorage map:database ;
d2rq:uriPattern "evento/@@evento.oggetto id@@";
d2rq:class iswc:Event ;
d2rq:classDefinitionLabel "evento";
d2rq:classDefinitionComment "Un evento";
```

```
Map:evento nome a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Evento ;
d2rq:property iswc:eventTitle ;
d2rq:property rdfs:label ;
d2rq:column "evento.nome";
d2rq:propertyDefinitionLabel "label";
```

L'esempio mostra l'istanza di una classe (*d2rq:ClassMap*) di tipo evento (*map:Evento*) mappata ai singoli *record* della tabella "evento" e la proprietà "label" di ciascuno di essi associata al campo "evento.nome" della tabella<sup>12</sup>.

---

<sup>6</sup> ZIPF 1949.

<sup>7</sup> BIZER 2004.

<sup>8</sup> Resource Description Framework (RDF), <http://www.w3.org/RDF/>; Web Ontology Language (OWL), <http://www.w3.org/TR/owl-features/>.

<sup>9</sup> Jena. A Semantic Web Framework for Java, <http://jena.sourceforge.net/>.

<sup>10</sup> SPARQL: Query Language for RDF - <http://www.w3.org/TR/rdf-sparql-query/>.

<sup>11</sup> GRUBER 1995.

<sup>12</sup> D2RQ, <http://www4.wiwiw.fu-berlin.de/bizer/d2rq/>.

Un'interrogazione *SPARQL* al modello *RDF* permette quindi di generare i dati per la visualizzazione in un formato di interscambio utilizzabile in ambiente *javascript*. Nello specifico i dati sono convertiti alla notazione *JSON* (*JavaScript Object Notation*).

I formalismi semantici hanno come scopo l'interazione tra macchine. Uno schema *RDF*, infatti, non offre una leggibilità adeguata alla comprensione umana. Per tale motivo sono stati sviluppati diversi strumenti in grado di generare visualizzazioni avanzate come *timelines*, mappe interattive, tabelle e gallerie a partire da una base di conoscenza semantica.

Il progetto *SIMILE* (*Semantic Interoperability of Metadata and Information in unLike Environments*)<sup>13</sup>, nato per iniziativa del *Massachusetts Institute of Technology* (*MIT*), raggruppa un insieme di tecnologie *opensource* di supporto allo sviluppo del web semantico.

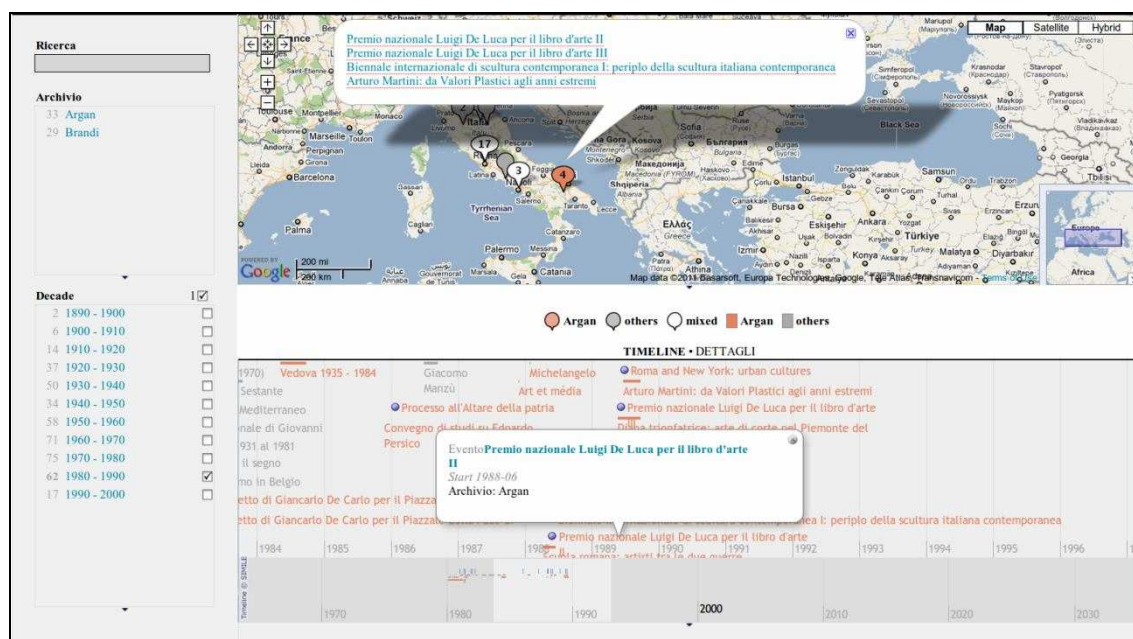


Figura 5. Una visualizzazione avanzata che fa uso di Exhibit.

L'applicazione utilizza il *framework Exhibit* appartenente al progetto *SIMILE* per l'implementazione di una vista avanzata sul modello dei dati. *Exhibit* è un *framework AJAX* che consente di realizzare visualizzazioni dinamiche di dati strutturati semanticamente. Il processo di generazione delle viste è totalmente a carico del *browser*. Le funzionalità del *framework* vengono infatti rese disponibili all'interno di una pagina *web* attraverso l'inclusione di *files javascript* (*API*). Le direttive di visualizzazione vengono impartite tramite specifici attributi assegnati agli elementi *HTML*.

La vista presenta una localizzazione geografica (mappa) e temporale (*timeline*) degli eventi. È possibile filtrare gli eventi da visualizzare in base all'archivio per intervalli temporali di durata decennale.

Tali visualizzazioni (di cui si fornisce un esempio alla Fig. 5) consentono agli studiosi di ricostruire i movimenti sul territorio e l'andamento cronologico dei percorsi critici per ognuno dei protagonisti i cui archivi documentali costituiscono la banca dati.

<sup>13</sup> SIMILE, <http://simile.mit.edu/>.

### *Conclusioni*

In questo articolo abbiamo presentato un sistema *web* per la gestione e la ricerca su archivi documentali manoscritti di natura artistica e letteraria del XIX e XX secolo scritti da alcuni noti critici dell'arte: Giovan Battista Cavalcaselle, Adolfo Venturi, Ugo Ojetti, Giulio Carlo Argan e Cesare Brandi. L'applicazione mette a disposizione un motore di ricerca semplice ed avanzato; un sistema di analisi automatica dei testi ai fini dello sviluppo di un sistema di annotazione semi-automatica, anche su base semantica; ed una vista avanzata tipica delle *Rich Internet Applications* che consente di collocare le singole unità di significato sia temporalmente che geograficamente e di metterle in relazione fra loro per mezzo di un modello dati mappato ad ontologie.

### **BIBLIOGRAFIA**

BIZER 2004

C. BIZER, *D2RQ-treating non-RDF databases as virtual RDF graphs*, Proceedings of the International Semantic Web Conference, Hiroshima 2004.

GRUBER 1995

T. GRUBER, *Principles for the design of ontologies used for knowledge sarin*, in *International Journal of Human-Computer Studies*, 43, 5-6, 1995, pp. 907-928.

LI 1992

W. LI, *Random texts exhibit zipf's-law-like word frequency distribution*, in *IEEE Transactions on Information Theory*, 38, 6, 1992, pp. 1842-1845.

MCCANDLESS-HATCHER-GOSPODNETIĆ 2010

M. McCandless, E. Hatcher, O. Gospodnetić, *Lucene in Action*, Manning 2010.

SCHMID 1994

H. SCHMID, *Probabilistic Part-of-Speech tagging using decision trees*, Proceedings of the International Conference on New Methods in Language Processing, Manchester 1994.

ZIPF 1949

G.K. ZIPF, *Human Behavior and the Principle of Least Effort*, Cambridge 1949.