

# Codex[ml]

**Matteo Perelli, Carla Zini**

*CILEA, Segrate*

## *Abstract*

Il sistema integrato Codex[ml] è stato sviluppato dal CILEA per la gestione, conservazione e fruizione via web di risorse digitali. La piattaforma consente a ogni istituzione culturale, biblioteca o archivio che possieda informazioni cartacee digitalizzate, la relativa fruizione attraverso la navigazione via web, sfruttando l'analisi dei diversi standard di metadati, in linguaggio XML. Il sistema si presenta modulare e scalabile: ogni modulo è un elemento indipendente che interagisce con gli altri per garantire la possibilità di creare, conservare, distribuire immagini corredate da metadati. L'utente può navigare all'interno della risorsa digitale visualizzandola con un'interfaccia web, sfruttando la struttura descritta nei metadati, codificati in linguaggio XML, secondo le specifiche previste dagli standard METS, MAG, etc., ma predisposta anche per nuovi standard non ancora scritti.

Codex[ml] integrated system has been developed by CILEA with the aim of meeting management, storage and fruition requirements of digital objects. Using Codex[ml] every agency and every library that deals with organized information can offer the fruition of its own works, text or archives, through an on-line navigation based on different xml metadata standards. Codex[ml] performs import, creation, long-term storage, delivery and fruition of xml documents that follow administrative standards for data management. It is an open system, which is able to manage METS, MAG and other standards, even not yet developed (adding new metadata interface modules). This is a modular and scalar software: every component is developed over a common framework to guarantee full interoperability.

*Keywords:* pHp, digitization, XML, storage, MAG, integrated system.

Il sistema integrato Codex[ml] [1] è stato sviluppato dal CILEA nell'ambito del "Piano Territoriale di potenziamento del Servizio Bibliotecario Nazionale - Regione Lombardia", finanziato con i fondi UMTS, per risolvere le esigenze di gestione, conservazione e fruizione via web dei documenti cartacei.

Codex[ml] trae il suo nome dall'accostamento tra il termine latino CODEX (manoscritto, codice) e il termine informatico XML (Extensible Markup Language). La piattaforma consente a ogni ente, biblioteca o struttura, che possiede informazioni cartacee organizzate, di proporre la fruizione delle proprie opere, testi o archivi attraverso la navigazione on-line, sfruttando l'analisi dei diversi standard di metadati (XML).

## **Il Progetto**

Il progetto nasce dall'esigenza di gestire i processi di digitalizzazione, emersa nell'ambito dei cicli di corsi formativi tenutisi al CILEA sui vari aspetti della digitalizzazione. I responsabili di biblioteca delle università consorziate, della Biblioteca Universitaria di Pavia e della Biblioteca

Nazionale Braidense sono giunti a esprimere la necessità di disporre di una piattaforma integrata per la gestione delle varie fasi che portano alla fruizione via web dei documenti cartacei. L'analisi di mercato, effettuata dal CILEA per rispondere a queste esigenze, ha evidenziato un panorama povero dei necessari investimenti per lo sviluppo e una produzione di software assai contratta e poco affidabile. La poca dimestichezza dei responsabili degli enti riguardo a uno standard tutto italiano, quasi contrapposto, almeno in una prima fase, allo standard internazionale METS, ha condizionato fino a ora la realizzazione di ben strutturati progetti di digitalizzazione, la loro fruizione e, soprattutto, la loro conservazione nel tempo. Il CILEA ha proposto alle biblioteche appartenenti ai due poli MIL e USM di strutturare un sistema di fruizione collegabile agli OPAC già esistenti, interoperabile con il portale Internet Culturale [2] e in grado di gestire la conservazione delle risorse. Non avendo comunque a disposizione fondi sufficienti, il CILEA ha investito in proprio nel progetto e il gruppo di sviluppo ha deciso di

sfruttare a pieno le potenzialità degli *open source* disponibili in rete, in modo da abbassare i costi di avvio e di gestione. L'obiettivo è stato la creazione di un prodotto modulare, scalabile, dinamico ed elastico di fronte agli standard e alle esigenze sui metadati gestionali e amministrativi.

Si è quindi voluto che la piattaforma fosse in grado di supportare i prevedibili cambiamenti che questi strumenti, ancora poco utilizzati e quindi poco consolidati, potranno subire nei prossimi anni. I moduli che compongono la piattaforma sono strutturati al fine di permettere l'utilizzo di diversi modelli di metadati, oltre che il passaggio dall'uno all'altro sulla base delle mappature che via via dovranno garantire la massima interoperabilità. Ora la piattaforma è in grado di caricare metadati costruiti all'esterno ma, grazie a un impegno e a un ulteriore investimento del CILEA, sarà disponibile anche un modulo di creazione e gestione (modifica e cancellazione) dei metadati e degli oggetti digitali correlati.

Attualmente, la piattaforma viene utilizzata dalla Biblioteca di Facoltà di Lettere Francesco Petrarca dell'Università di Pavia. Infatti è stato digitalizzato e corredato di metadati il Fondo Fraccaro Rari di Storia locale [3] della biblioteca che ora è a disposizione degli studiosi sia che accedano direttamente dall'opac riservato (<http://armida.unipv.it>) sia che effettuino una ricerca sull'OPAC di Ateneo dell'Università (<http://opac.unipv.it>).

Si può vedere un esempio di fruizione facendo una ricerca per liste con il titolo 'forche caudine' e, una volta trovata la notizia, accedere alla risorsa digitale tramite l'apposito pulsante (fig. 1 e 2). Per dare maggiore rilievo a questa, che è la prima collezione digitale che l'Università di Pavia rende disponibile on line, è stato anche creato un apposito catalogo on line che interessa soltanto il Fondo Fraccaro Rari di Storia locale. L'OPAC è interrogabile al sito URL: <http://armida.unipv.it> e consente la visualizzazione diretta delle opere in formato digitale (fig.3).

The screenshot shows the OPAC - Catalogo On Line interface. At the top, it says 'Università degli Studi di Pavia' and 'Sistema Bibliotecario'. The main title is 'OPAC - Catalogo On Line'. Below this, there are tabs for 'Modalità di ricerca' (Liste, Semplice, Avanzata, Esperti, E-Journal) and 'Info e servizi' (Stampa, Help, Contatti). The search expression is '(TI=FORCHE AND TI=CAUDINE)' and the result count is 1. There are options for 'Formato ridotto' and 'Export'. Below that, it says 'Seleziona uno o più documenti da richiedere in prestito' with buttons for 'esegui' and 'annulla'. A checkbox is checked for 'Selezionare per richiedere in prestito'. The main content area shows a search result for 'Daniele, Francesco 1740-1812'. The result includes a description: 'Le Forche Caudine illustrate con due appendici . - In Napoli : appresso Angelo Trani, 1811 (In Napoli : nella stamperia di Angelo Trani, 1812). - XIII, [3], 102, [6] p., [1] c. di tav. ripieg. : ill. ; fol'. It also lists the author 'Trani, Angelo [tipografo]'. There are buttons for 'Accedi' (highlighted with a red box) and 'Richiedi in prestito'. At the bottom, there is a table with columns: Biblioteca, Collocazione, Inventario, Note, and Dettagli.

Biblioteca	Collocazione	Inventario	Note	Dettagli
Interdip. unificata Francesco Petrarca	F.SPECIALI Fraccaro Rari St.Loc. 5.	X 20003 C	1 v.	

Fig. 1 – Risultato della ricerca sull'OPAC d'Ateneo

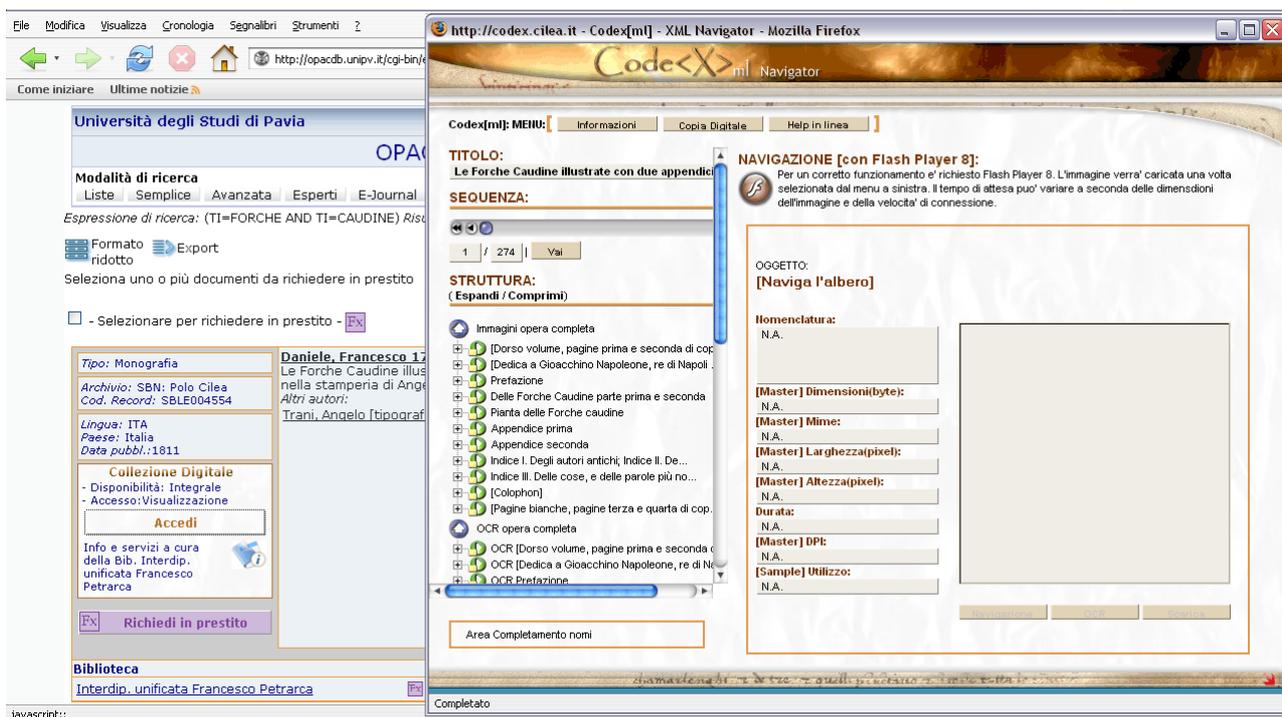


Fig. 2 – Esempio di chiamata a Codex[ml]



Fig. 3 – Esempio di ricerca sul catalogo on line del Fondo Fraccaro Rari di storia locale

## Codex[ml] in dettaglio

Da un punto di vista tecnico, Codex[ml] può essere considerato un insieme di applicativi collegati tra loro da un *framework* che racchiude tutte le funzionalità di base condivise. Il sistema è stato sviluppato interamente in ottica *object oriented*, con pHP 5.x [4], utilizzando molte estensioni PEAR [5] (PHP Extension and Application Repository) e librerie grafiche rese disponibili di recente. Per Codex[ml]: ImageServer (vedi avanti) è stato utilizzato anche Adobe Macromedia Flash 8, per rendere più intuitiva e adeguata all'utenza l'esperienza di navigazione delle immagini. Codex[ml] è affiancato da PostgreSQL 7.4, un database relazionale open source di libero utilizzo estremamente potente e caratterizzato da affidabilità e stabilità [6] elevate. Infine, per lo storage di file XML si è scelto di utilizzare eXist [7], un database XML nativo interamente realizzato in Java, che rende le operazioni di archiviazione e interrogazione di qualunque XML molto semplice ed efficiente. Riportiamo di seguito la configurazione software utilizzata per la macchina di produzione attualmente attiva al CILEA.

Tecnologie	Version
pHp	5.1.6
Java 2 SDK	1.5.0_09-b03

WebServer	Version
Apache http webserver	2.00.54

Servlet Container	Version
Apache Tomcat	5.05.20

pHp Libs	Version
GD Lib	2.00.28
Mcript Module	2.05.07
SOAP extension	n.a.

PEAR packages	Version	Release
Auth	1.04.00	stable
Cache_Lite	1.07.02	stable
Config	1.10.07	stable
DB	1.07.06	stable
HTML_AJAX	0.05.00	beta
HTML_Template_IT	1.02.01	stable
Mail	1.01.14	stable
Mail_Mime	1.03.01	stable
SOAP	0.09.04	beta
XML_Beautifier	1.01	stable
XML_Parser	1.02.07	stable
XML_Util	1.01.01	stable

Codex[ml] viene installato solamente su sistemi operativi Linux.

eXist (versione 1.1.1 newcore) è stato installato come pacchetto Java WAR e opera sotto Tomcat 5.

Passiamo ora ad analizzare i singoli elementi di Codex[ml], partendo dal suo cuore, il Framework.

## Il Codex[ml]: Framework

È possibile suddividere le funzioni del Framework in 5 aree principali:

- controllo delle variabili globali;
- configurazione del sistema;
- scelta del linguaggio (Italiano, Inglese, etc...);
- controllo delle risorse (dischi, nastri, database);
- scelta dei moduli per poter leggere i vari formati di metadati.

I primi due punti riguardano il *set-up* base di tutto il sistema, cioè la creazione di tutte le variabili e/o costanti necessarie al corretto funzionamento di Codex[ml] (compreso lo stesso framework). Il terzo punto riguarda la lingua in cui i contenuti e i messaggi verranno presentati all'utente.

Le parti più complesse del framework riguardano gli ultimi due punti. Tutti gli applicativi Codex[ml] utilizzano classi comuni per recuperare file su file-system (FS), dati relazionali da PostgreSQL, XML da eXist. In più è presente un sistema in grado di capire come leggere il contenuto dei file XML e come presentarlo. Codex[ml] è infatti in grado di utilizzare XML costruiti secondo diversi standard (come per esempio il MAG), a patto che esista il rispettivo modulo in grado di dire al sistema come interpretare i dati. Il framework, quindi, su richiesta di un applicativo (per esempio XML Navigator) fornisce sia il modo per accedere a una determinata risorsa, sia quello per utilizzarla.

All'interno di un processo attivo, le classi principali che compongono il framework sono Struttura e Architettura, rispettivamente la configurazione e il cuore del framework. Struttura contiene in sé tutte le variabili globali leggibili dagli applicativi (e in alcuni casi modificabili), mentre Architettura fornisce tutte le classi e i metodi condivisi. All'interno di quest'ultima citiamo le principali funzionalità elencate di seguito con una breve descrizione:

- Time\_Related: fornisce il timer a Codex[ml];
- Ecl\_Log: Classe che esegue il Log;
- Validations: Classe di validazione degli input;
- Input\_Detector: Classe di gestione dell'input;

- Error\_Exception: fornisce la gestione degli errori;
  - Configuration: esegue il set-up della configurazione globale e specifica per l'applicativo;
  - Message\_Handling: gestisce i messaggi di Codex[ml] (anche la lingua);
  - Page\_Builder: assembla i template HTML per Codex[ml];
- Db\_IF ed Exist\_IF: rappresentano le interfacce verso i database;
- Fs\_IF: fornisce l'interfaccia verso il File-System.

Di seguito, rappresentato in figura 4, osserviamo l'ordine di caricamento dei file.

Il primo file richiamato è index.php, quello a cui gli utenti fanno riferimento entrando in un sito internet; questo a sua volta richiama includes\_X.php (X = nome a scelta). I file che contengono una parte del nome variabile sono file che vengono replicati per ogni applicativo Codex[ml] (per esempio, esisteranno includes\_navigator.php, includes\_shifter.php, etc.); includes\_X.php si occupa di includere tutti i file necessari al corretto funzionamento dell'applicativo. Legge quindi includes.php, contenente i link verso tutti i file necessari e comuni, dopodiché richiama init\_X.php. Se osserviamo il grafico, init\_X.php ha un funzionamento analogo a includes\_X.php, ma si occupa della configurazione di base.

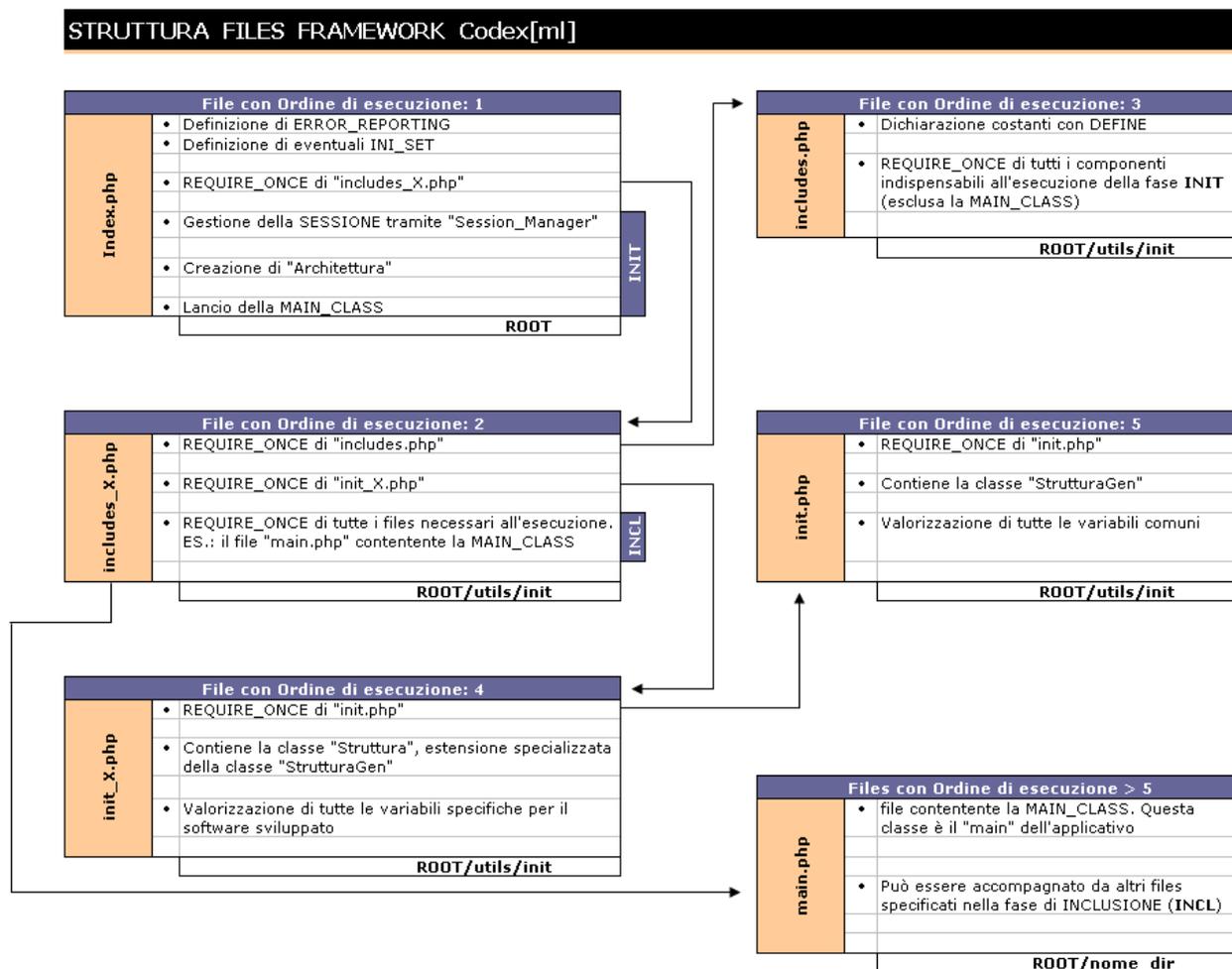


Fig. 4 – Struttura files del framework

Il file main.php contiene il codice dell'applicativo Codex[ml] ed è quindi presente in diverse copie, una per applicativo; viene letto da includes\_X.php come ultimo passo. A questo punto si torna a index.php, che istanzia le classi ed esegue il metodo start, presente in tutte le classi Main\_Class nei diversi main.php. Per fare un confronto, si può dire che Altre\_classi e Main\_Class del grafico precedente vengono lette e quindi incluse nel processo attivo da includes\_X.php e istanziate alla fine di index.php.

Il framework si occupa anche del dialogo con i database (fig.5). eXist fornisce innumerevoli funzionalità indispensabili al funzionamento di Codex[ml]. Innanzitutto è in grado di salvare qualunque XML indipendentemente dal suo contenuto, inoltre è in grado di interrogare i dati utilizzando X-Query (sfruttando anche gli X-Path). Poiché la velocità di reperimento dei dati, senza conoscere in modo preciso la posizione del documento all'interno di eXist, è molto bassa, gli è stato affiancato PostgreSQL, che

assume la funzione di indice per eXist, salvando al suo interno tutti i percorsi X-Path a cui si trovano gli Xml. PostgreSQL è un database relazionale molto veloce, per cui l'associazione con eXist e pHP / PEAR Cache\_Lite ha reso Codex[ml] estremamente veloce, anche nel caso si debbano gestire quantità di dati molto grandi (per esempio, gli XML MAG possono arrivare anche a 2 MB di dimensioni). Un ultimo e importante aspetto di eXist sono le trasformazioni (XSLT) che vengono utilizzate in modo esteso dal framework. eXist permette di salvare al suo interno file XSL (a tutti gli effetti degli XML) e utilizzarli per generare output direttamente trasformato, con prestazioni molto buone. La classe Exist\_IF si avvale di un client SOAP per inoltrare e ricevere dati da eXist.

Avendo spostato la complessità dello sviluppo sul framework, si è resa più semplice la programmazione di tutti gli applicativi che compongono Codex[ml], illustrati qui di seguito.

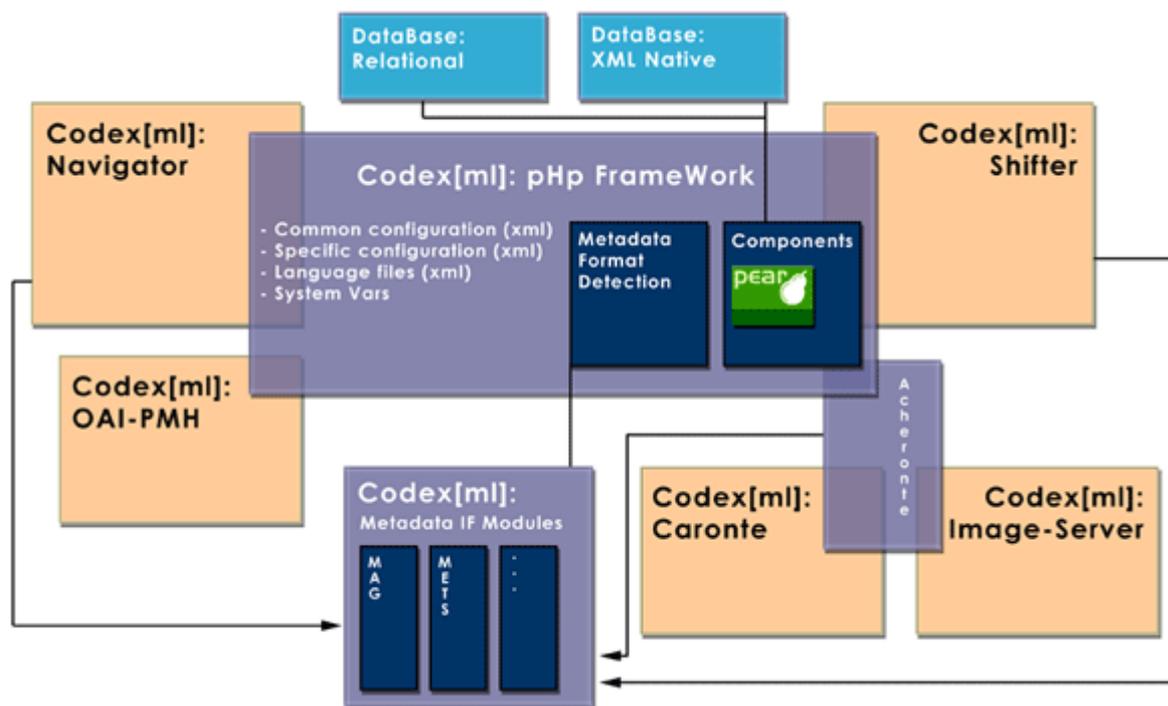


Fig. 5 – Relazioni fra gli applicativi che compongono Codex[ml]

## Il Codex[ml]: Container

Un gradino sopra il framework si trova Codex[ml] Container. Questo applicativo controlla l'archivio di conservazione di tutte le risorse digitali prodotte. In pratica crea, copia, muove, legge e cancella i file che si accompagnano ai metadati e che necessariamente devono essere immagazzinati in modo preciso, efficiente e ordinato. Controlla due tipi di deposito: Light Container, utilizzato per l'archiviazione delle risorse fruibili on-line, con capacità disco ridotta, ma dotato di una serie di hard-disk in grado di dare risposte rapide, adeguate alla consultazione via Web; Heavy Container, utilizzato per la conservazione, di capacità molto elevate, le cui peculiarità sono la capienza e la durata nel tempo dei supporti di memorizzazione. Di norma all'interno di Light Container sono presenti copie a risoluzione ridotta (attenzione! risoluzione ridotta non significa piccoli) dei file presenti in Heavy Container. Per sfruttare meglio il FileSystem, i file vengono salvati in una struttura di directory e subdirectory molto articolata, ma semplice dal punto di vista concettuale. Il file passato viene rinominato con l'aggiunta di un codice univoco e il nome così composto viene passato a una funzione di hashing (md5). I primi sette caratteri dell'hash vengono utilizzati per determinare il path del file su disco. Per esempio, il file 001.jpg viene rinominato 12\_001.jpg, il cui hash è:

```
24c85efec06eda9a77e12b6072a69f3a.
```

Il percorso finale sarà:

```
/storage/2/4/c/8/5/e/f/12_001.jpg.
```

Prendendo i primi 7 caratteri è possibile salvare  $16^7$  files.

## Il Codex[ml]: Caronte

Codex[ml] Caronte si occupa di traghettare le risorse presenti all'interno dello storage (Container) all'applicazione richiedente (per esempio: un applicativo Codex[ml], un browser, oppure un'applicazione realizzata da terze parti), sempre che la diffusione ne sia consentita. Può essere utilizzato indipendentemente per consentire all'utente di scaricare direttamente il contenuto di Codex[ml]: Container e permette a Codex[ml] ImageServer di recuperare le immagini da visualizzare (vedi avanti). In termini informatici produce un flusso di dati binari utilizzabili da altro software. Nella versione attuale è in grado di leggere solo i file salvati su disco e non su nastro; in Codex[ml] 2 sarà anche in grado di gestire download in differita neces-

sari quando si parla di lunghi tempi di recupero dati dovuti a jukebox di nastri.

## Il Codex[ml]: Image Server

Codex[ml] ImageServer ha il compito di ricampionare e visualizzare gli oggetti digitali di tipo immagine. Riceve, tramite Codex[ml] Caronte, il flusso di dati binario che compone il file sorgente e, a seconda della richiesta, ne ricalcola le dimensioni in tempo reale, in modo che Codex[ml]: Navigator visualizzi le *thumbnail* (miniature) nell'interfaccia principale e ne consenta la fruizione vera e propria attraverso Adobe Macromedia Flash 8. Codex[ml] ImageServer è interamente realizzato in pHp, ma il suo client è stato sviluppato con Adobe Macromedia Flash 8, utilizzando in buona parte il sistema di scripting Action Script. Il client di Codex[ml] ImageServer è richiamabile solamente da Codex[ml] Navigator (vedi avanti). Nell'attuale versione possiede una semplice finestra di navigazione, che racchiude le opzioni permesse. Dal punto di vista dell'utente, l'ImageServer è uno strumento semplice, in cui le azioni permesse sono di Zoom In/Out per ridurre o ingrandire l'immagine e di spostamento sulla superficie (fig.6).

Per avere un archivio semplice, in cui non fossero coinvolti formati proprietari di immagini, si è optato per un maggior carico sulle CPU e formati JPG, GIF, PNG come immagini compatibili, fino a 1,8 MB di dimensioni. Se da un lato tutto ciò può apparire un limite, si consideri che un'immagine in formato jpg di 700 KB RGB può avere dimensioni di 2300 x 1600 pixel con una compressione del 75% (3.6 megapixel) – cioè un'immagine enorme da visualizzare a video! La funzione di Codex[ml] ImageServer è dunque quella di ridimensionare immagini così grandi, mantenendo al contempo una buona qualità visiva; in caso contrario, l'esperienza di fruizione risulterebbe compromessa.

## Il Codex[ml]: Navigator

Codex[ml] Navigator è lo strumento che permette di visualizzare e navigare i documenti XML presenti a sistema. Rappresenta un po' il volto di Codex[ml], in quanto chiunque sia dotato di una connessione a internet può accedervi. Navigator ricostruisce la struttura dei metadati (definita nel lavoro di catalogazione dall'operatore addetto) con un menù ad albero e consente la fruizione dettagliata degli oggetti digitali contenuti al suo interno (fig.7).



Fig. 6 – Interfaccia Codex[ml] Image Server

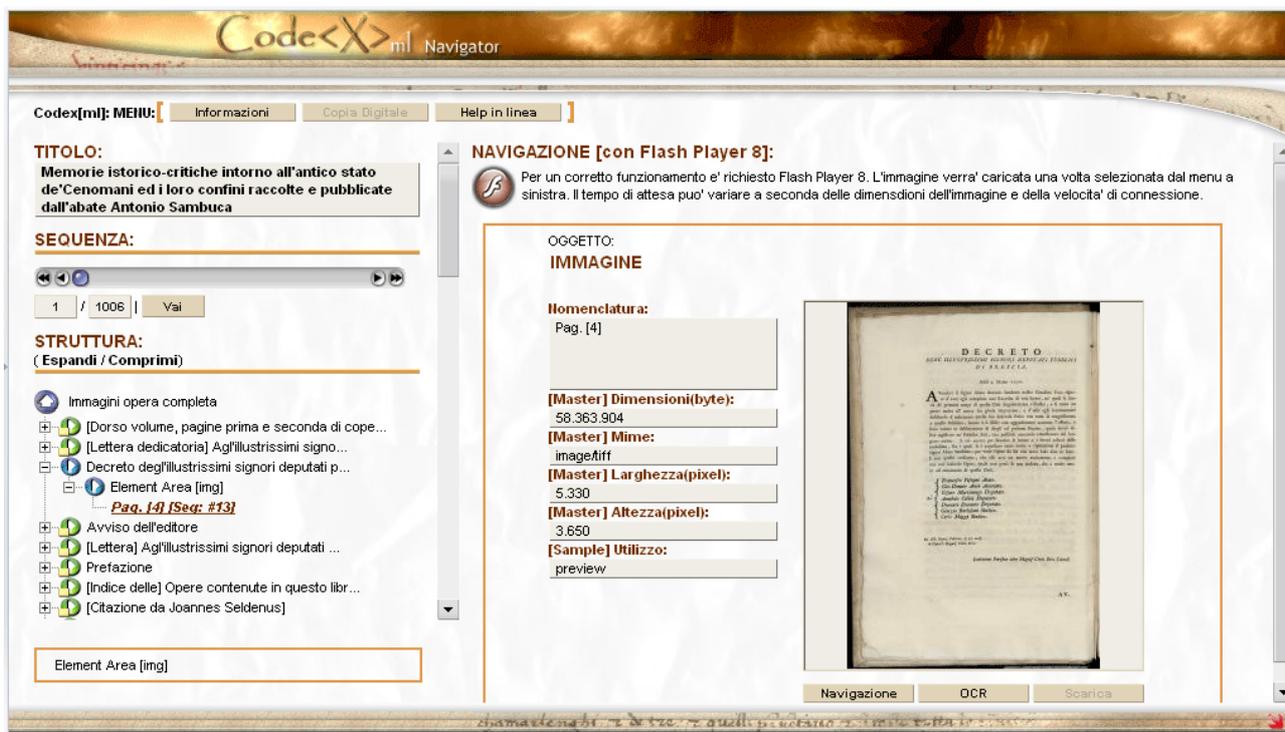


Fig. 7 – Interfaccia Codex[ml] Navigator

Permette di sfogliare l'oggetto digitale via web, bilanciando il carico di lavoro tra client e server. Richiama, inoltre, le funzionalità di Codex[ml] ImageServer e permette di navigare le immagini conservate in Codex[ml] Container.

Le funzionalità di ricerca più estese sono demandate agli OPAC; questo significa che se non si possiedono tutti i dati richiesti per l'interrogazione, non si otterrà risultato. La parte sinistra dell'interfaccia di Codex[ml] Navigator è dedicata all'XML, mentre la parte destra alle risorse descritte dall'XML. Questo significa che a sinistra troveremo l'albero espandibile, i riferimenti bibliografici e le informazioni sull'ente che possiede il corrispettivo oggetto fisico, mentre a destra avremo un'anteprima della risorsa in esame e i controlli per scaricarla (se possibile) oppure per navigare al suo interno. Tutte le informazioni vengono reperite in tempo reale direttamente dagli XML in eXist; i dati sono interpretati a partire dall'output trasformato dal database XML nativo. Le immagini vengono invece fornite a seguito di una chiamata a Codex[ml] ImageServer e anch'esse sono ricampionate in tempo reale (con l'ausilio di una cache). Il flusso dei dati è illustrato nella figura 8.

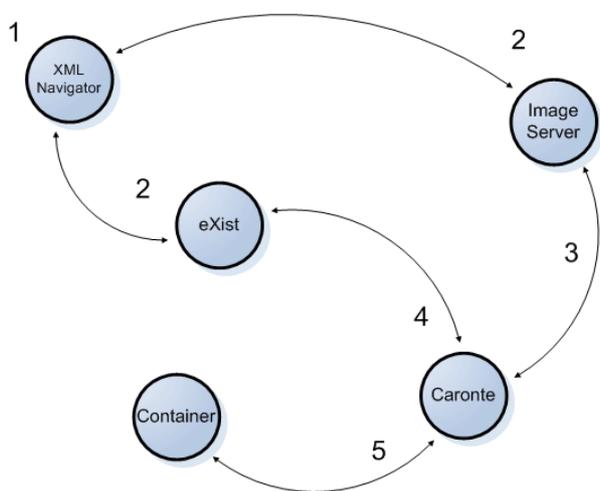


Fig. 8 – Flusso dei dati per la fruizione delle immagini

Da XML Navigator partono due richieste, una verso eXist e l'altra verso l'ImageServer. eXist risponde con XML trasformati, mentre

l'ImageServer chiede a Caronte la risorsa da ricampionare.

Caronte verifica se si possiedono i diritti per visualizzare l'immagine dialogando con eXist; se la risposta è positiva manda la richiesta a Container, che restituisce la risorsa. A questo punto, l'ImageServer esegue il ricampionamento e restituisce il tutto a Navigator che potrà quindi mostrare la risorsa.

### Il Codex[ml]: OAI - PMH

Codex[ml] OAI-PMH è un disseminatore/data provider OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting). OAI-PMH è uno standard internazionale utilizzato per la diffusione di metadati provenienti da archivi aperti e altri software compatibili. Si occupa, quindi, della sola ricerca e visualizzazione dei metadati. Questa tipologia di software risponde utilizzando solo XML, per cui è uno strumento adatto a utilizzatori più esperti o ad altri applicativi (per esempio NTC). Può essere utile se si vuole esaminare l'XML originale, di una particolare risorsa, contenuto negli archivi di Codex[ml], lo stesso che interpreta Codex[ml] Navigator per presentare un contenuto facilmente navigabile. Dalle pagine dello stesso Navigator viene sempre creato un link dinamico verso Codex[ml] OAI-PMH.

### Il Codex[ml]: Creator

Codex[ml] Creator sarà il modulo di data entry, aggiornamento e gestione degli standard XML. Verrà implementato con Codex[ml] 2, già in fase di sviluppo.

### Il Codex[ml]: Shifter

Codex[ml] Shifter è un importatore di metadati. Consente di caricare le risorse all'interno di Codex[ml] dai documenti XML che descrivono un oggetto digitale ai testi, alle immagini, ai video etc. Contestualmente si occupa della validazione dei documenti XML a seconda dello standard scelto: se i metadati descrittivi delle risorse non sono validi, l'applicativo non esegue l'Import.

Una volta selezionati i dati, viene preparato un XML Driver, al cui interno sono specificati la posizione degli XML da importare, le opzioni relative al progetto di appartenenza e all'archivio OAI-PMH, quindi si lancia lo Shifter che, seguendo queste istruzioni, esegue in batch tutto il lavoro. A causa della consistente quantità di dati che vengono gestite dallo Shifter, la procedura di import può essere molto lunga, sia per

le dimensioni delle risorse descritte dagli XML da importare (anche 60 MB per file, fino a 600 file) che per le alterazioni a cui necessariamente devono essere sottoposti gli stessi XML. La principale modifica consiste nel cambiamento dei link alle risorse descritte, per esempio le URL/URI verso i file all'interno dell'XML. Una volta che i documenti sono spostati in Codex[ml] Container (Light/Heavy), questi ultimi possono essere recuperati solo tramite Codex[ml] Caronte, per cui i vecchi link vengono sostituiti con quelli diretti verso Caronte, codificati con gli ID univoci delle immagini. Vista la possibile durata e la complessità della procedura, la gestione degli errori è stata strutturata in modo da permettere il recupero di tutte le operazioni già effettuate anche in caso di errore bloccante. Codex[ml] Shifter, infatti, non solo scrive un *log* dettagliato delle operazioni, ma in caso di errore produce un *dump*, che una volta risolto il problema, può essere utilizzato in una seconda esecuzione dello Shifter, dando la possibilità di proseguire l'import dal punto di interruzione; invece in caso di errore sul singolo XML, crea il dump per quell'XML e prosegue con il successivo.

### **Il Codex[ml]: prospettive future**

Come già accennato, è attualmente in fase di sviluppo Codex[ml] 2. Questa nuova versione manterrà tutte le caratteristiche funzionali di Codex[ml] 1 e ne aggiungerà molte altre, prime tra tutte le nuove funzionalità di Codex[ml] Creator. Verrà inoltre migliorata l'esperienza di fruizione con Codex[ml] Navigator, implementando una nuova interfaccia Flash (client di Codex[ml] ImageServer) e verranno estese le funzionalità di Codex[ml] Shifter.

### **Bibliografia**

- [1] URL: <http://codex.cilea.it>
- [2] URL: <http://www.internetculturale.it>
- [3] URL: <http://petrarca.unipv.it/digit/fraccaro.htm>
- [4] URL: <http://www.php.net>
- [5] URL: <http://pear.php.net>
- [6] URL: <http://www.postgresql.org>
- [7] URL: <http://www.exist-db.org>