

## LO STATO DELL'ARTE DELL'INNOVAZIONE TECNOLOGICA PER LE ARCHITETTURE WEB: PRESENTE E FUTURO PER «ARCHEOLOGIA E CALCOLATORI»

### 1. PREMESSA

L'oggetto di studio di questo scritto è lo stato dell'arte nell'innovazione tecnologica per le architetture web e le scelte adottate per il restyling strutturale del portale della rivista «Archeologia e Calcolatori» (d'ora in seguito A&C).

Il sito web di A&C necessitava di un restyling urgente a livello di programmazione, al fine di un aggiornamento della piattaforma ASP, nonostante questa per 15 anni si fosse occupata di eseguire script interattivi svolgendo egregiamente il proprio lavoro. Purtroppo il linguaggio ASP non è più supportato da Microsoft e le scelte infrastrutturali delle architetture web del CNR, che ospitano il portale di A&C, sono e saranno sempre più orientate verso il mondo dell'open source e verso architetture LAMP (Linux, Apache, MySQL, PHP/Phyton).

Per tale motivo una trasformazione non era solo suggerita ma necessaria. Si è pertanto deciso di rinnovare completamente il sito web, costituito da circa 70 pagine fisiche da riscrivere in linguaggio PHP. Il lavoro è stato impegnativo, ma la trasformazione verso l'open source ci ha aperto le porte alle tantissime nuove tecnologie che oggi stanno modificando le architetture web dell'intero WWW e che ci consentiranno di implementare in futuro servizi e funzioni sempre nuovi.

Nel lavoro di adattamento alle nuove tecnologie, particolare riguardo è stato rivolto ai fenomeni più significativi nel panorama innovativo tecnologico dei Sistemi Informativi su web: le Basi di Dati Semantiche e le nuove tecnologie RIA (Rich Internet Application), una splendida evoluzione, queste ultime, delle architetture web a lato client. Queste due soluzioni sono state utilizzate per la nuova architettura web del sito di A&C e i principali sforzi di restyling strutturale si sono concentrati lungo le seguenti tre direttrici principali:

- aggiornamento dell'architettura server-side (PHP, Node.js);
- costruzione di una base di dati “open source”, attraverso l'uso di un modello semantico di progettazione, e successivo popolamento (modello E-R, MySQL);
- utilizzo di architetture client-side (tecnologie per RIA: JavaScript, DOM, Ajax, JQuery, HTML5, XML) volto all'introduzione di primitive di interazione più potenti nell'architettura web, che offrono un insieme più ricco di eventi ed effetti rispetto ad una semplice architettura HTTP/HTML.

### 2. LE RAGIONI DI UN RESTYLING STRUTTURALE

La preferenza da parte dei Provider odierni, ormai a livello mondiale, per le tecnologie open source e la scelta delle soluzioni tecnologiche proprietarie

adottate in fase di sviluppo del portale di A&C hanno portato quest'ultimo a diventare velocemente e inevitabilmente obsoleto, cosa d'altronde del tutto naturale, se si pensa che un prodotto informatico può iniziare a dare segni di invecchiamento anche dopo un solo anno di vita.

Per quanto riguarda il sito, all'inizio del nuovo Millennio le linee guida sull'identità visuale erano state già ben definite, per cui le abbiamo lasciate sostanzialmente immutate; ma la parte grafica, che insieme ad altri elementi costituisce il cosiddetto "front end" del sito, è solo la punta dell'iceberg di un'architettura web ben più complessa e profonda (Fig. 1). La parte sostanziale è infatti costituita dagli script, che rendono le pagine dinamiche, e dal database, che alimenta di contenuti l'intera architettura. Gli script a loro volta possono operare sul server, come PHP, o direttamente sul client, come JavaScript; i database possono essere provvisti di aspetti semantici ed essere usati come strumenti per consentire la rappresentazione e gestione dei dati e la formulazione di interrogazioni ad alto livello, in cui si sfruttino conoscenze semantiche più ricche.

Con la progettazione e implementazione di un database estensionale è possibile esprimere interrogazioni con linguaggi formali, come Datalog e SPARQL, che sono di fatto linguaggi capaci di fare inferenze, calcolando i predicati definiti in modo intensionale a partire dal database estensionale, e che quindi si prestano alla descrizione di aspetti semantici, quali le ontologie o i vincoli di integrità.

### 3. L'ARCHITETTURA TECNOLOGICA DI PARTENZA E DI ARRIVO

Il portale di A&C è stato pensato e costruito, nei primi anni dello scorso decennio, come data provider nel framework di interoperabilità fornito dal protocollo OAI-PMH (Open Archives Initiative – Protocol for Metadata Harvesting). L'OAI è un'iniziativa, nata nel 1999, per sviluppare e promuovere standard di interoperabilità che mirano a facilitare la diffusione efficiente dei contenuti. Gli ambiti di applicazione sono principalmente gli archivi aperti e l'ampliamento e la facilitazione della gestione e dell'accesso alle risorse digitali per eScholarship, eLearning ed eScience. Dall'OAI sono scaturiti due progetti attualmente ancora operativi, l'OAI-PMH e l'OAI-ORE (Object Reuse and Exchange).

L'OAI-PMH, da noi adottato, in estrema sintesi, prevede l'esistenza di due provider: 1) un data provider, che espone i metadati delle risorse secondo standard descrittivi come Dublin Core, MARC e OAI-ORE (nel nostro caso Dublin Core) che ne facilitano la ricerca; 2) un service provider (harvester, letteralmente "mietitore" dei metadati) per gestire i servizi a valore aggiunto, per l'aggregazione e l'indicizzazione dei metadati (ricerca, scoperta, localizzazione degli oggetti digitali creati dai data provider) e per l'interrogazione

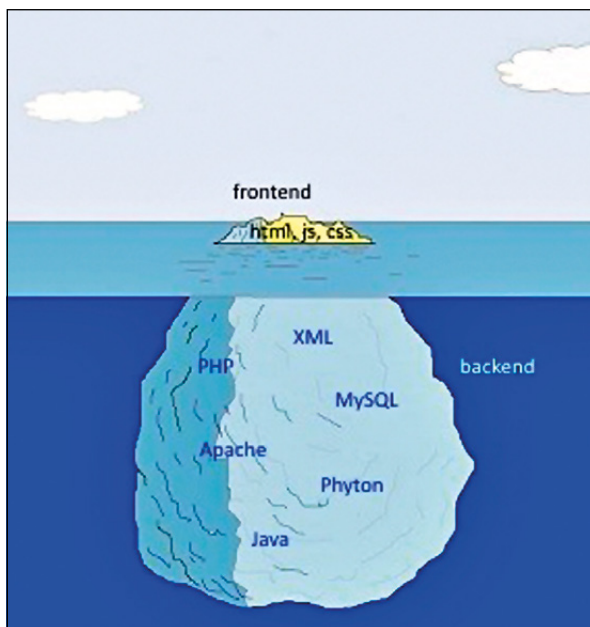


Fig. 1 – Frontend vs backend di un'architettura web.

degli archivi. Tra il data provider e il service provider c'è l'intermediazione di un server, definito gateway, gestito da una terza parte, che ha il compito di raccogliere le sottoscrizioni dei data provider e fornire supporto al protocollo OAI-PMH verso gli harvester (service provider) (BARCHESI 2005).

Dal 2005 A&C è quindi un data provider che rende disponibile il proprio repository digitale; un documento XML è generato da un database Microsoft Access con l'utilizzo del linguaggio di programmazione Visual Basic for Application (VBA); nel repository digitale le risorse condivise sono descritte attraverso il set di metadati Dublin Core. Sempre in quegli anni, l'implementazione web del portale di A&C è stata realizzata in HTML con l'uso del linguaggio di scripting server-side ASP, il tutto ospitato da un server web Microsoft IIS del CNR di Pisa.

A partire da queste premesse, fatte le dovute considerazioni e analisi dei requisiti e fermamente convinti di conservare le caratteristiche di data provider per il protocollo OAI-PMH, abbiamo stabilito le quattro direzioni principali per realizzare il restyling strutturale del portale:

- sostituzione di Microsoft IIS su server Windows con Apache su server Linux;
- sostituzione del linguaggio di scripting server-side ASP con PHP;
- sostituzione del database Access con MySQL;

– sostituzione del linguaggio VBA con PDO (PHP Data Object), un’estensione di PHP introdotta a partire dalla versione 5.1 del linguaggio con lo scopo di unificare le API di accesso ai database.

#### 4. SCELTA DEL LINGUAGGIO DI SCRIPTING

Il primo passo è stato quello di scegliere il linguaggio di scripting server-side da sostituire ad ASP, linguaggio che opera in ambiente Microsoft IIS; la scelta è ricaduta su PHP, uno dei linguaggi ancora più richiesti e usati per il web. Lo dimostra la classifica semestrale redatta da RedMonk sui linguaggi di programmazione più in voga, che analizza i codici pubblicati su GitHub e le richieste dei programmatori su Stack Overflow, un forum dove molti “developers” si incontrano per discutere di programmazione e ottimizzazione dei codici; Stephen O’Grady, analista e cofondatore di RedMonk, raccoglie infine i risultati in forma grafica (Fig. 2). Un vero vincitore non c’è, ma JavaScript e Java la fanno da padroni: il primo per il web con tutti i recenti framework come React.js, Node.js e JQuery, il secondo sia per applicazioni web che stand-alone. Ma qualsiasi classifica consultiamo, è il linguaggio Python ad

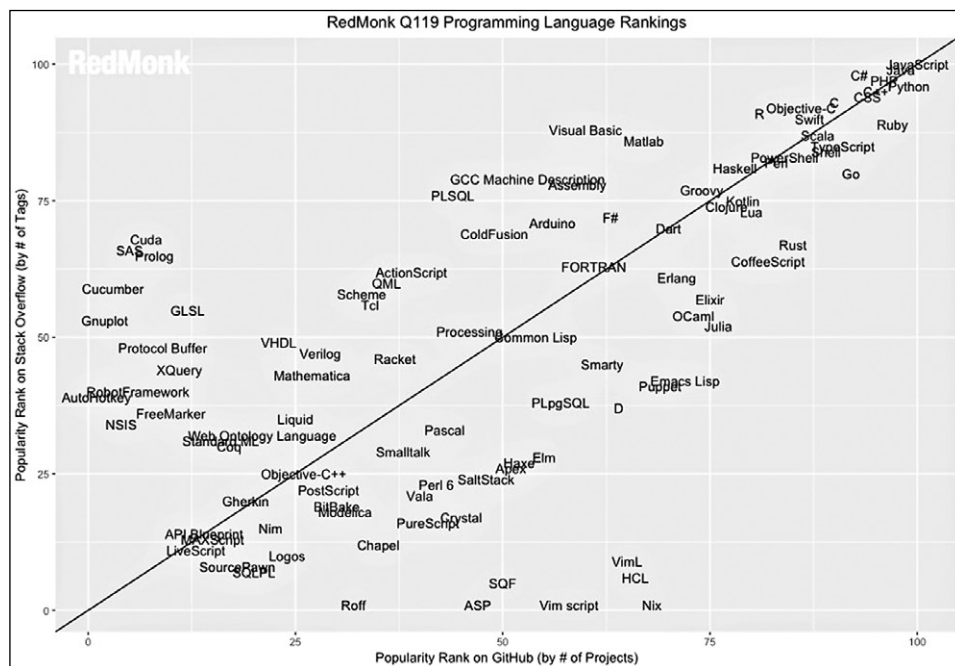


Fig. 2 – RedMonk Q119 Programming Language Rankings.

essere il vero elemento sorpresa, sicuramente il linguaggio del futuro, data la sua semplicità e trasversalità sia per il web che per applicazioni stand-alone e/o con GUI (Graphical User Interface).

Per quanto riguarda la scelta tra PHP e ASP, da un punto di vista tecnologico non c'è grande differenza, in quanto sono entrambi linguaggi validi. La differenza sostanziale è invece nella parte commerciale e strategica, che lega obbligatoriamente l'utilizzatore dello script ASP a un ambiente Microsoft e quindi ai suoi costi di acquisto e gestione; la scelta di utilizzare PHP in ambienti UNIX-like ha dato invece a molti Enti pubblici, compreso il CNR, la possibilità di scelte hosting più economiche e di contenere i costi di sviluppo e mantenimento. Ultima differenza, ma non meno importante, è il codice chiuso e non open source di ASP, codice che si presenta spesso sotto forma di file DLL che l'utente non può modificare. La scelta del codice open source PHP consente invece un'ampia libertà di modifica, una comunità di sviluppatori enorme e un'infinità di esempi pratici da prendere come casi d'uso descrittivi di determinate problematiche.

Il grafico in Fig. 2 mostra chiaramente come i linguaggi più richiesti siano, in ordine, JavaScript, Java, Python e PHP, mentre il codice ASP, pur essendo ancora molto diffuso (come possiamo vedere dai valori sulle ascisse), non sia quasi più richiesto (ordinate). Questo dimostra che PHP, forse proprio per la sua vocazione web, risulta ancora il linguaggio più usato e popolare. È probabile che nel futuro dovremo sostituirlo con Python e con il suo potentissimo Django (framework open source per Python, utilizzato per facilitare il processo di sviluppo di applicazioni web) o anche con il leggero ed estensibile Flask, ma ad oggi esso resta il candidato ideale per le nostre esigenze.

## 5. IMPLEMENTAZIONE AUTONOMA VS CMS

Il percorso di restyling che abbiamo intrapreso, dunque, parte dall'utilizzo di PHP e MySQL – che ci consente di conservare lo status di sito “dinamico” – per arrivare alla realizzazione di un CMS completo e funzionale (ROCHKIND 2013; WELLING, THOMSON 2016; BUTLER, YANK 2017; MELONI 2017; NIXON 2018; WEST, PRETTYMAN 2018). Come è noto, tutti i siti su Internet appartengono a due macro-tipologie: Statici e Dinamici. Nei siti Statici le pagine, pur non mancando di animazioni, non cambiano in aspetto e contenuti in funzione delle richieste dell'utente: i contenuti sono cioè di sola ed esclusiva lettura. Possiamo pensare un sito statico come un'immensa bacheca dove le informazioni sono solo quelle esposte. Nei siti Dinamici (come il nostro) le pagine cambiano in aspetto e contenuti, in genere a seguito di una richiesta specifica dell'utente “navigatore” al server web. Possiamo immaginare un sito dinamico come lo sportello di un grande ufficio, dal quale è possibile ottenere informazioni varie o richiederne di specifiche.

Un errore comune è quello di considerare dinamico un sito le cui pagine sono ricche di animazioni ed effetti speciali; la parola dinamico qui non si riferisce al movimento di un elemento grafico nella pagina ma solo alla manipolazione e successiva presentazione di un contenuto (che si trova sul server) a seguito di una richiesta del client. Nel nostro caso, ad esempio, non abbiamo animazioni ed effetti speciali, ma tramite il linguaggio PHP possiamo inviare richieste specifiche al server che ospita le nostre pagine e il nostro database e ottenere in risposta contenuti grafici e/o testuali tali da permettere la creazione dinamica di migliaia di pagine distinte e specifiche; sempre di più la metafora dell'iceberg diventa esplicativa, una piccola porzione visibile apre le porte ad un immenso archivio sommerso di informazioni.

Un sottoinsieme dei siti dinamici è costituito dai sempre più diffusi CMS (Content Management Systems), software che aiutano i proprietari dei siti a gestire autonomamente i contenuti e l'aspetto grafico delle proprie pagine e consentono agli utenti "navigatori" di interagire con il sito web, permettendo ad esempio di registrarsi come autori e di inviare i propri contenuti e commenti. I CMS contengono al loro interno tutti gli strumenti necessari per la creazione, la modifica e la cancellazione dei contenuti di un sito web, senza la necessità di conoscere il codice HTML o un linguaggio di programmazione.

Dal punto di vista tecnologico, un CMS è un'applicazione lato server che generalmente si appoggia su un database per l'archiviazione dei contenuti. Esso presenta sempre due interfacce:

- un backend di amministrazione costituito da tutto il complesso sistema di pannelli che compongono l'area riservata ad accesso esclusivo del webmaster o del proprietario del sito;
- un frontend di fruizione dei contenuti rappresentato dal sito pubblico e visibile ai navigatori.

La difficoltà nella realizzazione di un CMS è tanto maggiore quanto diversa è la varietà dei dati che questo software deve immagazzinare, e quanto è complessa l'interazione che questi dati devono avere nel frontend al visitatore. L'utilizzo di questi sistemi, che impongono comunque un contenitore standard, comporta sempre l'accettazione di un compromesso da parte del proprietario del sito, tanto più accettabile quanto più il CMS è personalizzato sulla base delle complesse specifiche del sito.

Per quanto riguarda il sito di A&C, visto il passaggio da Access a MySQL, si è pensato di iniziare a progettare un backend di amministrazione con la specifica funzione di popolamento del database; è stata cioè creata un'area riservata ad accesso esclusivo, con funzioni di inserimento, modifica e cancellazione dei dati relativi al database. Nel futuro l'obiettivo sarà ovviamente quello di rendere sempre più complessa la struttura di questa plancia

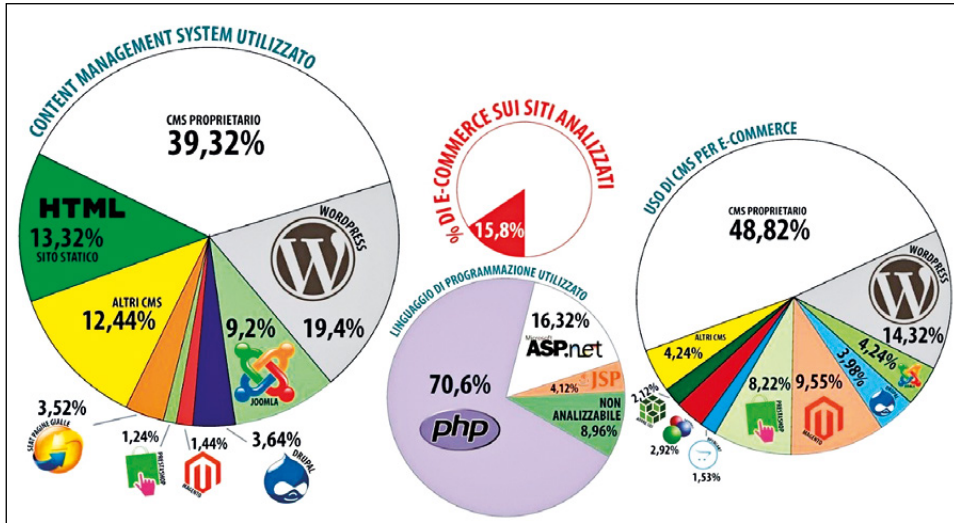


Fig. 3 – Infografica delle tecnologie dei siti Internet italiani (<http://www.posizionamento-seo.com/>, 2015).

di comando del sistema, per una completa gestione anche delle pagine del frontend oltre che del database MySQL.

Esistono centinaia di CMS sia con codice proprietario, sia open source, come mostra la Fig. 3, un'infografica dettagliata e ben costruita che, benché datata, offre ancora un'idea di massima di quali siano le tecnologie più usate sul web in Italia e in quali ambiti queste vengano utilizzate. La differenza principale fra un CMS proprietario e un CMS open source sta nel fatto che il primo si adatta alle esigenze del proprietario e contiene soltanto gli strumenti specifici e personali per cui è stato progettato; il secondo, invece, essendo concepito per una vasta platea di utenti diversi, contiene spesso più funzioni di quelle necessarie e impone un'architettura standard cui invece il proprietario del sito si deve adattare.

D'altro canto, un CMS proprietario richiede tempi e risorse molto alti mentre uno open source è gratis e implementabile con pochi passaggi. Come abbiamo potuto verificare anche per il nostro sito, il problema è che un CMS open source non sempre risponde a qualsiasi esigenza e, quando vi risponde, spesso non lo fa nel modo desiderato.

## 6. IMPLEMENTAZIONE DI UN BACKEND PER A&C

In fase di raccolta delle specifiche per il restyling del sito web di A&C ci si era interrogati se costruire un CMS proprietario o scegliere un CMS

open source. Fra questi ultimi la nostra scelta era inizialmente ricaduta su Drupal, oggi uno dei migliori a parere di chi scrive; siti web realizzati in Drupal sono ad esempio quello della Casa Bianca (<https://www.whitehouse.gov/>), della NASA (<https://www.nasa.gov/>) e dello stesso CNR (<https://www.cnr.it/>). Drupal è dotato di una particolare composizione modulare: di base ha a disposizione un design essenziale, ma il grado di personalizzazione che si può raggiungere con l'aggiunta di nuovi moduli è decisamente più elevato rispetto ai più diffusi WordPress e Joomla.

Se si hanno esigenze specifiche, come nel caso di A&C per quanto riguarda l'uso del protocollo OAI-PMH, il fatto che Drupal sia sviluppato in linguaggio PHP avrebbe semplificato di molto un'eventuale programmazione proprietaria per un modulo dedicato. Tuttavia, programmare moduli aggiuntivi per dotare un CMS open source di funzioni non nativamente previste richiede grande esperienza di programmazione. Si ritiene, infatti, che quando la proporzione tra il codice da creare e quello già "pronto" a corredo del CMS open source supera una certa percentuale, la creazione di un CMS proprietario diventa di gran lunga la scelta migliore.

Abbiamo pertanto deciso di optare per un CMS proprietario e abbiamo così iniziato la costruzione di un backend di amministrazione dei contenuti: l'"Admin backend dashboard" è stata progettata in questa fase iniziale per le funzioni che riguardano il popolamento del database MySQL. Poiché il vantaggio principale di un CMS proprietario è la possibilità di integrazione ed espansione nel tempo delle funzioni implementate, in un futuro prossimo, oltre alla gestione del database, il backend dovrà essere organizzato in una struttura che preveda anche la gestione delle pagine, dei menù, degli utenti, dei media e del protocollo OAI-PMH.

## 7. PROGETTAZIONE SEMANTICA DI UN DATABASE PER A&C

Per la creazione del nostro database ci siamo avvalsi doverosamente della metodologia di progettazione più largamente diffusa nell'ambito delle basi di dati. Metodologia di riferimento che è articolata in tre fasi principali da effettuare in cascata e si fonda su un principio dell'ingegneria semplice ma molto efficace: separare in maniera netta le decisioni relative a "cosa" rappresentare in una base di dati (prima fase) da quelle relative a "come" farlo (seconda e terza fase) (ATZENI *et al.* 2018).

### 7.1 *Progettazione concettuale*

Come strumento per la progettazione concettuale dei dati è stato utilizzato il modello Entità-Relazione; si tratta di un precursore (CHEN 1976) di un vastissimo numero di modelli che, per le loro caratteristiche, sono classificati come "modelli semantici". Attraverso una serie di raffinamenti successivi, il



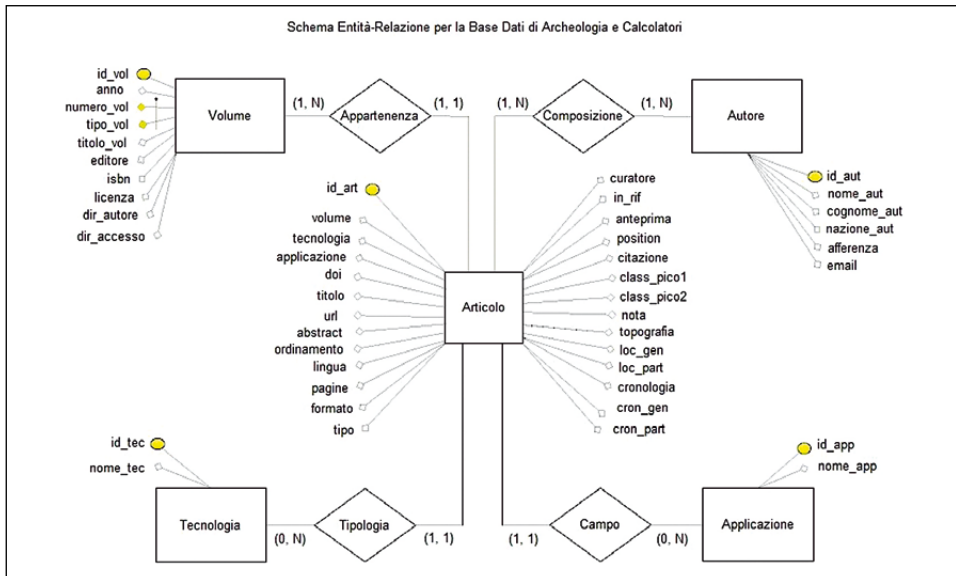


Fig. 4 – Schema Entità-Relazione per la base dati di A&C.

modello semantico Entità-Relazione ci ha consentito di costruire lo schema concettuale dei dati, sul quale l'intera metodologia si fonda, e di rappresentare il contenuto informativo della base di dati di A&C.

Come primo passo abbiamo individuato cinque Entità, cioè classi di oggetti della nostra realtà di interesse che hanno proprietà comuni ed esistenza "autonoma"; le cinque entità sono state collegate logicamente con quattro Relazioni, vere e proprie relazioni matematiche tra le occorrenze delle entità coinvolte. In uno schema E-R, Entità e Relazioni sono rappresentate graficamente rispettivamente da rettangoli e rombi; nella scelta dei nomi delle Relazioni si è preferito utilizzare sostantivi invece che verbi, in maniera da non indurre ad assegnare un verso alla Relazione.

Concettualmente le nostre inferenze logiche sono tutte derivate dalle seguenti considerazioni:

- ogni Volume della rivista è formato da più Articoli;
- ogni Articolo tratta di una Tecnologia;
- ogni Articolo ha un Campo d'Applicazione;
- ogni Articolo ha uno o più Autori.

In Fig. 4 è illustrato lo schema concettuale finale cui siamo pervenuti, con gli Attributi individuati per ogni Entità e le cardinalità di ogni legame logico fra Entità e Relazioni.

## 7.2 Progettazione logica

Per passare a una rappresentazione ancora indipendente da dettagli fisici ma concretamente realizzabile, lo schema concettuale è stato tradotto in uno schema logico basato su criteri di ottimizzazione e di semplificazione. Lo schema logico ottenuto vede nello schema relazionale la realizzazione della sola associazione molti a molti (Composizione), mentre le altre tre associazioni (Appartenenza, Tipologia e Campo) vengono inglobate dall'entità (Articolo) coinvolta nella relazione con cardinalità (1, 1). Di seguito lo schema logico realizzato:

- VOLUME (id\_vol, anno, numero\_vol, tipo\_vol, titolo\_vol, editore, isbn, licenza, dir\_autore, dir\_accesso);
- ARTICOLO (id\_art, volume, tecnologia, applicazione, doi, titolo, url, abstract, ordinamento, lingua, pagine, formato, tipo, curatore, in\_rif, antepri- ma, position, citazione, class\_pico1, class\_pico2, nota, topografia, loc\_gen, loc\_part, cronologia, cron\_gen, cron\_part);
- AUTORE (id\_aut, nome\_aut, cognome\_aut, nazione\_aut, afferenza, email);
- COMPOSIZIONE (id articolo, id autore);
- TECNOLOGIA (id\_tec, nome\_tec);
- APPLICAZIONE (id\_app, nome\_app);

## 7.3 Progettazione fisica

In questa fase lo schema logico è stato completato con la specifica dei parametri fisici di memorizzazione dei dati. Per la realizzazione si è utilizzato PhpMyAdmin, uno dei più diffusi pannelli di gestione via browser per MySQL. Scritto interamente in PHP, è un software open source e consente un accesso multilivello al database server; permette di creare, modificare e rimuovere facilmente interi database, tabelle, record e dati. Insostituibile la possibilità di esportare e importare rapidamente strutture e dati.

## 8. UNO SGUARDO AGGIORNATO SULLE NUOVE ARCHITETTURE DEI SISTEMI INFORMATIVI SU WEB

Il lavoro di restyling del sito web di A&C ha necessariamente richiesto da parte nostra un aggiornamento sulle nuove architetture dei Sistemi Informativi su web. Nella sua accezione originale, il World Wide Web può essere visto come un sistema su scala planetaria di distribuzione e accesso a documenti ipertestuali e multimediali, codificati in HTML, che viaggiano attraverso un canale di comunicazione client-server grazie al supporto del protocollo HTTP. Con il passare degli anni il web è diventato una complessa piattaforma per lo sviluppo di sistemi informativi eterogenei, accessibili attraverso i browser dai più svariati tipi di terminali; il browser è diventato così un client universale

in grado di collegarsi a qualsiasi applicazione che pubblichi la propria interfaccia utente sotto forma di pagine web. I sistemi informativi su web (Web Information Systems, WIS) condividono con i Sistemi Informativi tradizionali (SI) la capacità di gestire scambi di dati; poiché i dati sono gestiti per lo più da DBMS, è evidente l'importanza di poter integrare le tecnologie delle basi di dati e quelle del web.

## 8.1 WIS vs SI

Rispetto ai SI tradizionali, i WIS gestiscono informazioni con strutture flessibili e irregolari (immagini, audio, video); si pensi ad esempio ai dati semi-strutturati in un file XML. I WIS, inoltre, offrono agli utenti una ricca varietà di primitive per l'interazione con i dati in lettura. Infine, le interfacce utente devono permettere l'uso dei WIS con dispositivi e sistemi operativi molto diversi fra loro: un WIS deve poter operare su computer, smartphone, iPad, etc. Queste differenze fra WIS e SI spiegano gli sforzi fatti in questi ultimi anni per un ripensamento totale del processo di sviluppo delle applicazioni web, per quanto riguarda sia le componenti da realizzare ex-novo, sia l'integrazione di moduli software preesistenti.

Possiamo pensare globalmente al WWW come a un'architettura multi-livello (multi-tier) in cui le funzioni di presentazione (client), di elaborazione delle applicazioni (business logic) e di gestione (database) dei dati sono logicamente separate. Una tipica architettura three-tier prevede, ad esempio, un personal computer per l'interfaccia utente, un server per la business logic e un server per la gestione dei dati. Questo schema generale può anche essere esteso ipotizzando ad esempio che il livello intermedio sia a sua volta "stratificato"; in questo caso si giungerebbe ad un'architettura multi-tier o n-tier.

Il vantaggio di queste architetture modulari è che le applicazioni software sui diversi livelli sono sviluppate e mantenute come moduli indipendenti, nella maggior parte dei casi distribuiti su piattaforme separate e quindi su nodi diversi di una rete anche eterogenea. In questo modo, ciascuno dei moduli può essere modificato o sostituito indipendentemente dagli altri, conferendo scalabilità e manutenibilità all'architettura. Tali moduli sono intesi ad interagire fra loro secondo le linee generali del paradigma client-server: l'interfaccia è cliente della business logic, e questa è cliente del modulo di gestione dei dati persistenti.

### 8.1.1 Architetture lato server (three-tier con PHP)

L'architettura di A&C è esattamente una three-tier architecture (Fig. 5), con il client (client tier) realizzato con un browser standard, il server dei dati (database tier) con il DBMS MySQL, e il livello intermedio (middle tier) che ospita un web server Apache opportunamente esteso oggi con un PHP processor (ieri era un processore ASP) per l'esecuzione di programmi (oggi script PHP) per l'estrazione dei dati e la costruzione dinamica di pagine web.

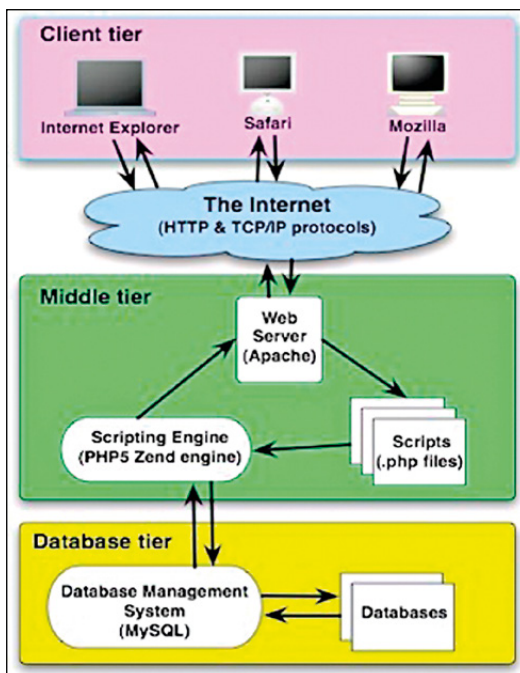


Fig. 5 – Esempio di architettura a tre livelli.

Esistono tante tecnologie utilizzate nelle architetture a tre livelli per estendere le funzioni del web server in modo da poter consentire la costruzione dinamica delle pagine web a partire da contenuti estratti dalle basi di dati. Di seguito elenchiamo le più diffuse:

- Common Gateway Interface (CGI): il client invia parametri ad uno script nel middle tier che provvede a costruire le relative risposte da inviare al client.
- Java Servlet: ambiente di esecuzione ad oggetti.
- “Template di pagina” e “server side scripting” (JSP, Java Server Pages): separazioni parti fisse e dinamiche di una pagina.
- Model-View-Controller (MVC): ripartizione modulare del codice nel middle tier.

Per l’architettura server-side di A&C abbiamo utilizzato il meccanismo delle “query string” (<http://www.archcalc.cnr.it/journal/idyear.php?IDyear=2001-01-01>), anche utilizzato nelle CGI, per l’invio di parametri ad uno script scritto nel nostro caso in PHP, la cui esecuzione permette la costruzione dinamica delle pagine. Con l’obiettivo di modularizzare la programmazione per renderla più scalabile e manutenibile anche in A&C è stata introdotta la

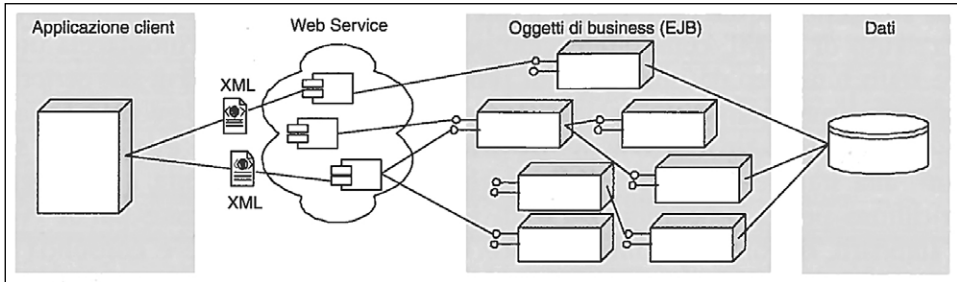


Fig. 6 – Architettura n-tier a servizi (ATZENI *et al.* 2018).

tecnica dei “template di pagina” che ci ha consentito una migliore separazione nelle pagine tra le parti fisse e le istruzioni per il calcolo delle porzioni dinamiche.

### 8.1.2 Architetture lato server multi-livello (n-tier)

La necessità di migliorare le prestazioni e la scalabilità del middle tier ha spinto ad un ulteriore potenziamento dell'architettura a tre livelli, con l'aggiunta di un nuovo elemento tra il middle tier e la base di dati, chiamato application server. Tale architettura, detta multi-livello, esporta il meglio della tecnologia delle basi di dati verso il mondo delle applicazioni, offrendo cioè alle applicazioni client una serie di servizi che solitamente è garantita da un DBMS.

### 8.1.3 Architetture n-tier a servizi

Il progetto OAI-PMH di A&C è un'architettura multilivello n-tier piuttosto che una semplice implementazione del paradigma client-server, e precisamente un'architettura a servizi, il cui middle tier ha per componenti un server web esteso, un server gateway e molti service provider; l'approccio è cioè molto simile a quello di una Service-Oriented Architecture. L'aggiunta di un web service nel middle tier (Fig. 6) offre un maggior grado di disaccoppiamento tra client e server, uno strato di messaggi in formato XML rendono trasparenti al client i dettagli implementativi delle funzioni server, l'XML funge cioè da lingua franca che permette le comunicazioni in un ambiente eterogeneo.

Negli anni la definizione di Web Service si è ampliata con l'inclusione di uno stile di realizzazione di servizi distribuiti in rete (REST – Representational State Transfer), basato più sulla rappresentazione delle risorse oggetto del servizio che sulle operazioni da questo offerte. In un'architettura a servizi, la maggiore indipendenza fra l'applicazione client e il fornitore di servizi si ottiene a discapito delle prestazioni, a causa del sovraccarico di elaborazione dovuto alla codifica e decodifica dei messaggi. Una soluzione a questa problematica può essere l'adozione di JSON e JavaScript anche a lato server (Node.js); l'uso di questa tecnologia contribuisce ad alleviare l'onere di codifica e decodifica

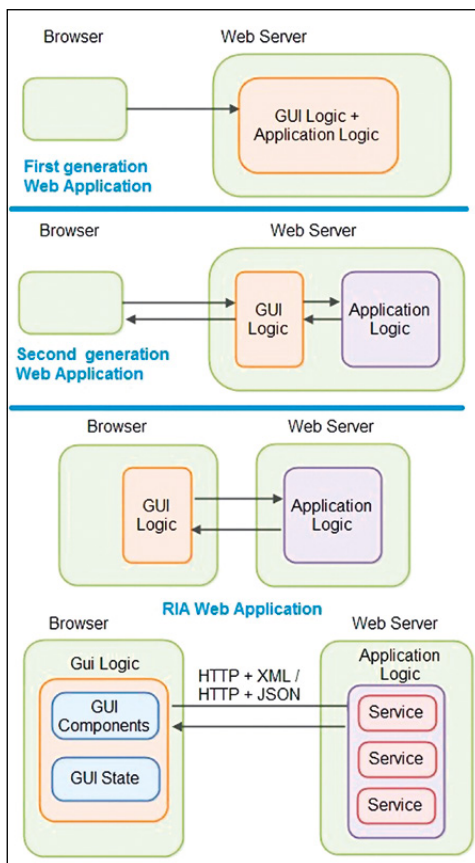


Fig. 7 – Rich Internet Application (JENKOV 2014).

dei messaggi e permette di realizzare architetture per i Web Service efficienti e scalabili; l'adozione di questa soluzione è oggi molto diffusa.

#### 8.1.4 Architetture lato client. Tecnologie per le RIA

L'ambizione finale del nostro processo di restyling è stata quella di trasformare A&C in un'applicazione di terza generazione, anche detta RIA Web Application (Fig. 7; JENKOV 2014). Tramite l'uso di tecnologie RIA come il DOM (Document Object Model), JavaScript, e JQuery abbiamo spostato quanto più possibile la logica di presentazione (GUI Logic) dal server al client, lasciando sul server solo l'Application Logic, la parte di costruzione dinamica delle informazioni che fornisce le pagine richieste dal client. Il risultato si traduce in un risparmio di risorse per il server ed in enormi vantaggi per

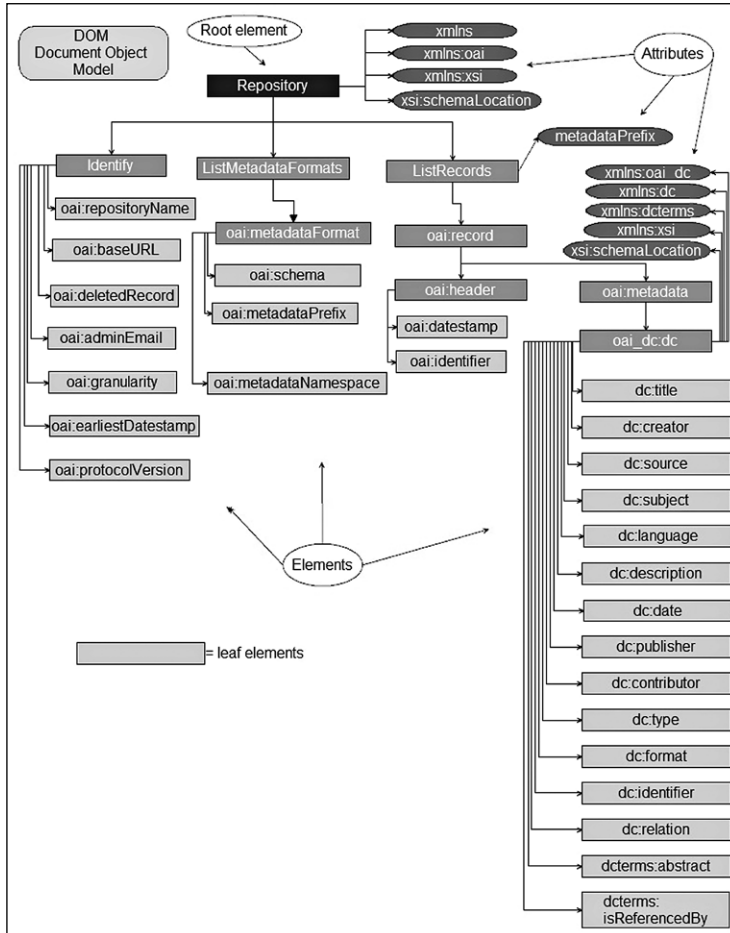


Fig. 8 – Modello DOM del repository XML di A&C. Schema costruito con DiagramDesigner.

l'utente, che ha a sua disposizione funzioni altamente interattive (drag & drop, completamento automatico dei campi, paginazione delle tabelle, etc.).

Le tecnologie RIA in genere comunicano con il server web scambiando dati, messaggi, non con il codice usato per la GUI Logic (HTML, CSS e JavaScript). Lo scambio di dati avviene in genere tramite file XML o JSON, inviati tramite protocollo HTTP. Questa metodica cambia concettualmente il server da fornitore di pagine a fornitore di servizi come abbiamo visto avvenire nei Web Service. Lato client le RIA presentano un'interfaccia "ricca", come nelle applicazioni desktop, anche per contenuti di tipo multimediale (GUI

components), progettate e sviluppate all'inizio in JavaFX e GWT (GIUDICI 2018), oggi sempre più in HTML5, CSS3 e JavaScript (ZAKAS 2012); nelle applicazioni tradizionali per avere questo tipo di supporto si doveva ricorrere all'uso di estensioni del browser o di applicazioni plug-in.

Le RIA inoltre permettono la memorizzazione persistente di dati strutturati sul client (GUI State) evitando i continui e costosi cicli di comunicazione client-server. La programmazione lato client con JavaScript, utilizzato prevalentemente per arricchire le interfacce web, ad esempio con animazioni, eventi e controlli dell'input dell'utente, richiede un modello a oggetti della pagina che permetta al programmatore di accedere ai diversi elementi per leggerne e modificarne le proprietà. Tale modello a oggetti esiste ed è uno standard internazionale, promosso dal W3C, noto come Document Object Model (DOM). In Fig. 8 è illustrato il Modello DOM del repository XML di A&C.

Il DOM è un'interfaccia di programmazione ad oggetti che vale sia per documenti HTML che per documenti XML; lo standard in vigore (DOM Level 3) specifica una serie di oggetti e metodi per ispezionare e manipolare il contenuto di un documento. Per ispezionare e manipolare i nodi di un DOM è possibile utilizzare JavaScript o qualsiasi linguaggio di programmazione che implementi il modello, come nel nostro caso il PHP, utilizzandone proprietà e metodi.

Le RIA non avrebbero avuto successo senza l'abbinamento delle capacità di JavaScript con quelle di un altro oggetto fornito dall'ambiente di esecuzione: AJAX (Asynchronous JavaScript and XML), che consente un aggiornamento selettivo di una regione dell'interfaccia, richiedendo la porzione di dati necessari solo a quella regione ed evitando in questo modo il continuo refresh della pagina ad ogni interazione dell'utente con relativo sovraccarico di comunicazione fra client e server.

Un altro oggetto che ha determinato il successo delle RIA è l'uso di JQuery, una libreria JavaScript compatta e leggera. Per la progettazione del backend di amministrazione di A&C, nella visualizzazione di tabelle di dati con paginazione e funzioni di ordinamento su ogni campo, l'utilizzo delle librerie JQuery.js e DataTables.js ha dato risultati eccezionali in termini di velocità di caricamento e semplicità di codifica.

#### 8.1.5 XML come anello di congiunzione fra il web e le basi di dati

Un aspetto fondamentale di XML è il fatto di rappresentare istanze di documenti contenenti dati e markup, in maniera simile a quanto rappresentato da un file web in HTML. D'altro canto la caratteristica particolare dei dati in XML è il fatto di avere un grado variabile di struttura (dati semi-strutturati); un documento XML dotato di schema, come il nostro Repository, può essere facilmente elaborato in modo automatico da un programma scritto in un linguaggio di programmazione come ASP o PHP, al fine di creare con





base di dati qualsiasi e, con l'ausilio di un foglio di stile (nel nostro caso un file XSL) sono state costruite le pagine dinamiche. Il linguaggio di scripting PHP, per nostra fortuna, implementa "nativamente" il modello DOM, per cui ne abbiamo utilizzato le proprietà e i metodi per accedere e manipolare i documenti DOM rispettivamente del Repository (XML) e del foglio di stile (XSL). Il risultato è stato uno script in PHP di sole 9 righe di codice (Fig. 9), che con l'utilizzo di opportuni parametri (query string) nelle richieste dà vita a migliaia di pagine dinamiche (Fig. 10).

## 9. CONCLUSIONI

Il restyling tecnologico del sito web di A&C è stato per noi un volo attraverso l'evoluzione del web, nel tentativo di inquadrarne lo "stato dell'arte" e di catturarne le novità. Ci siamo così resi conto che il sito web di A&C, creato alla metà degli anni Novanta in occasione del III International Symposium of Computing and Archaeology (MOSCATI 1996) e allora ospitato presso il CISADU dell'Università di Roma la Sapienza, ha seguito nel tempo le principali tappe evolutive della rete (Fig. 11). Qui tutto è iniziato nel lontano 1989, quando Tim Berners-Lee inventò un sistema che permetteva la consultazione collettiva via Internet di pagine web raggiungibili con hyperlink. Gli anni dal 1990 al 2004 sono stati gli anni del World Wide Web (Web 1.0), secondo il paradigma dell'"only read web": il Web of content, l'Internet dei contenuti, statici e in HTML, che potevano essere gestiti solo dai webmaster. Il termine Web 2.0 venne coniato da Tim O'Reilly nel 2004, aprendo l'era del Social Web e della fase del "read-write web": con l'introduzione dei linguaggi di programmazione dinamici gli utenti diventavano parte integrante del web, interagendo con i contenuti dei siti (social network, blog, forum, wiki, etc.).

Con velocità sorprendente, già nel 2006 si incomincia a parlare di Web 3.0: è l'epoca del Semantic Web e si entra nella fase del "read-write-execute web". Le parole d'ordine sono Dati e Semantica, la rete viene vista sempre più come un enorme database i cui contenuti possono essere messi in connessione in modo automatico; i motori di ricerca semantici non si basano solo sulla semplice frequenza di un termine nel testo, ma su un'analisi del significato del testo e degli ambiti tematici a cui fa riferimento. Il WWW continua a correre velocissimo e già nel 2015 si parla di Web 4.0 e il Symbiotic Web entra nella fase del "read-write-execute-concurrency". Le parole d'ordine sono Big Data, Realtà Aumentata, Domotica, alter ego digitali che si mettono in connessione con le "cose" attraverso l'esplosione delle tecnologie wireless; è il Web of Things, un web ubiquitous che integra le due fasi precedenti mettendo in connessione non solo i contenuti ma anche le persone e le cose. Negli anni dal 2020 al 2030 si prevede la nascita del Web 5.0: sarà l'epoca dell'Emotional Web, del Web of Thought, che metterà in connessione contenuti, cose e pensieri; gli input

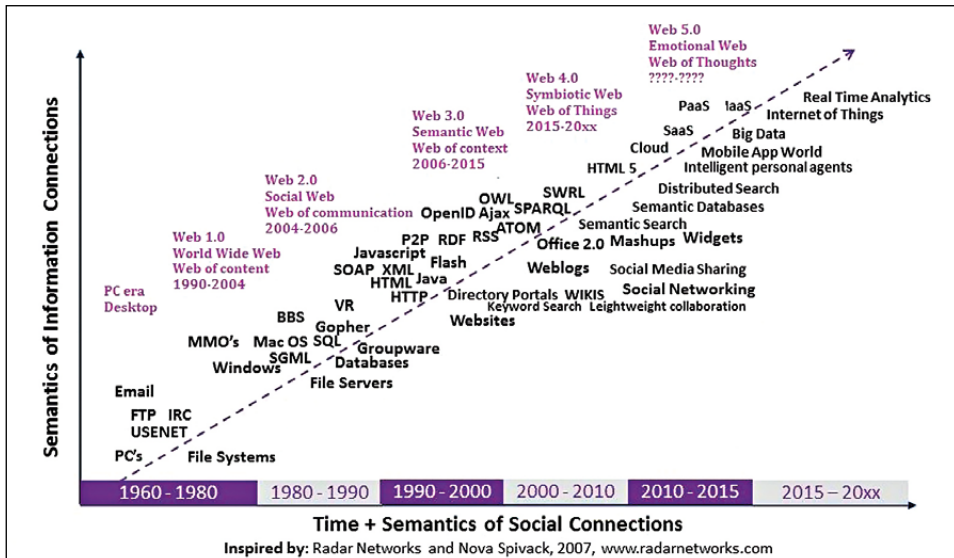


Fig. 11 – L'evoluzione del web verso il Web of Thoughts (<https://novaspivack.typepad.com/RadarNetworksTowardsAWebOS.jpg>).

partiranno da protesi cerebrali e il web si adatterà finanche agli umori degli utenti. Forse è fantascienza ma la storia ci ha insegnato che l'impossibile diventa spesso realtà, in particolare se parliamo di informatica (BENITO-OSORIO *et al.* 2013; PATEL 2013; KHANZODE, SARODE 2016; JAFAR 2018).

In questa evoluzione il web diventa sempre più uno strumento di servizio, per lo sfruttamento di risorse non solo statiche, ma sempre più intelligenti. Ciò che è certo è che la programmazione dei Sistemi Informativi su web richiede oggi la conoscenza di una varietà ampia di linguaggi di programmazione (PHP, Python, Javascript), di formati di rappresentazione dei dati e dei documenti (SQL, XML, JSON) e di schemi progettuali (E-R, DOM).

Per quanto concerne il sito web di A&C, come è stato sempre nel corso della vita trentennale della rivista, abbiamo cercato di dare una risposta veloce agli aspetti che ci sembravano più stringenti e importanti per un allineamento all'evoluzione tecnologica degli ultimi anni. I prossimi step ci vedranno impegnati a volgere lo sguardo verso le architetture a servizi e le tecnologie RIA, per un client sempre più "ricco" e potente e un server sempre più orientato ai servizi.

SALVATORE FIORINO

Istituto di Scienze del Patrimonio Culturale – CNR  
salvatore.fiorino@cnr.it

## BIBLIOGRAFIA

- ATZENI P., CERI S., FRATERNALI P., PARABOSCHI S., TORLONE R. 2018, *Basi di Dati*, Milano, McGraw-Hill Education, 5<sup>th</sup> ed.
- BARCHESI C. 2005, "Archeologia e Calcolatori": Nuove strategie per la diffusione di contenuti in rete sulla base dell'ÖAI-PMH", «Archeologia e Calcolatori», 16, 225-241 ([http://www.archcalc.cnr.it/indice/PDF16/Barchesi\\_16\\_2005.pdf](http://www.archcalc.cnr.it/indice/PDF16/Barchesi_16_2005.pdf)).
- BENITO-OSORIO D., PERIS-ORTIZ M., RUEDA ARMENGOT C.R., COLINO A. 2013, *Web 5.0: The future of emotional competences in higher education*, «Global Business Perspectives», 1, 3, 274 287 (<https://doi.org/10.1007/s40196-013-0016-5>).
- BUTLER T., YANK K. 2017, *PHP & MySQL: Novice to Ninja*, Collingwood, SitePoint Pty, 6<sup>th</sup> Ed.
- CHEN P. 1976, *The Entity-Relationship model: Toward a unified view of data*, «ACM Transactions on Database Systems», 1, 9-36.
- GIUDICI F. 2018, *Ascesa e declino di JavaFX* (<http://www.mokabyte.it/2018/04/javafx-1/>).
- JAAFAR A.F. 2018, *Evolution of the Web: Web 1.0, Web 2.0, Web 3.0, Web 4.0, Web 5.0 and Beyond* (<http://ahmadfaizar.blogspot.com/2018/08/evolution-of-web-web-10-web-20-web-30.html>).
- JENKOV J. 2014, *RIA Architecture*, Jenkov Aps (<http://tutorials.jenkov.com/software-architecture/ria-architecture.html>).
- KHANZODE C.A., SARODE R.D. 2016, *Evolution of the World Wide Web: From Web 1.0 to 6.0*, «International Journal of Digital Library Services», 6, 2 (<http://www.ijodls.in/uploads/3/6/0/3/3603729/ijodls121.pdf>).
- MELONI J.C. 2017, *PHP, MySQL & JavaScript, All in One*, Indianapolis, Sams publishing, 6<sup>th</sup> Ed.
- MOSCATI P. (ed.) 1996, *III International Symposium on Computing and Archaeology (Roma 1995)*, «Archeologia e Calcolatori», 7 (<http://www.archcalc.cnr.it/journal/idyear.php?IDyear=1996-01-01>).
- NIXON R. 2018, *Learning PHP, MySQL & JavaScript, with jQuery, CSS & HTML5*, O'Reilly Media, 5<sup>th</sup> Ed.
- PATEL K. 2013, *Incremental Journey for World Wide Web: Introduced with Web 1.0 to Recent Web 5.0 – A Survey Paper*, «International Journal of Advanced Research in Computer Science and Software Engineering», 3, 10, 410-417 (<https://pdfs.semanticscholar.org/41ad/b8d825d83680803851756bef93516e5a8208.pdf>).
- ROCHKIND M. 2013, *Expert PHP and MySQL: Application Design and Development*, Berkeley, CA, Apress.
- WELLING L., THOMSON L. 2016, *PHP and MySQL Web Development*, Addison-Wesley, 5<sup>th</sup> Ed.
- WEST A.W., PRETTYMAN S. 2018, *Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites*, 2<sup>nd</sup> Ed., Apress.
- ZAKAS N.C. 2012, *Professional JavaScript for Web Developers*, 3<sup>th</sup> Ed., Wrox, An imprint of Wiley.

## ABSTRACT

The paper illustrates the recent evolution of web architectures and the choices made for the web portal of «Archeologia e Calcolatori». The website needed an urgent restyling to update its ASP platform, even though for 15 years it had performed its interactive work very well. Today, the ASP language is no longer supported by Microsoft and the infrastructural choices of the CNR are and will be increasingly oriented towards the world of open source and LAMP architectures (Linux, Apache, MySQL, PHP/Phyton). For this reason, a transformation was not only suggested, but vital. It was therefore decided to completely renew the website, a task consisting of about 70 physical pages to be rewritten in PHP. This work transformation has helped to pave the way for new technologies that today are modifying the web architecture of the entire WWW and that will soon allow us to implement new services and functions. Particular attention was paid to the most significant phenomena in the innovative technological panorama of Web Information Systems, with particular emphasis on the Semantic Databases and the new RIA (Rich Internet Application) technologies, the latter a splendid evolution of client-side web architectures.