

# Knossos: un database di scavo open source per l'archeologia

Damiano Lotto<sup>\*</sup>, Francesco Biscani<sup>†</sup>, Sebastiano Tibolla<sup>‡</sup>

**SOMMARIO.** Il contributo ha lo scopo di presentare dal punto di vista sia metodologico che funzionale il processo di creazione di un ideale database open source indirizzato specificatamente all'impiego archeologico.

Il programma presentato è basato su formati e standard open source: si tratta di un'applicazione programmata in C++ utilizzando le librerie Qt (versione 4), multiplatforma, con capacità di gestione e connessione ai più diffusi database open source (tra i quali MySQL, PostgreSQL, SQLite).

**ABSTRACT.** *This paper shows the development process of a ideal open source database specifically oriented to archaeological work, with the focus centered on the metodological approach and on the software functionality. Knossos is based on open source formats; it's developed in C++ with Qt libs (version 4) and is studied to be multiplatform, with the feature to connect to the most used open source databases, like MySQL, PostgreSQL, SQLite.*

## 1. Una premessa

Quando questo contributo venne presentato il focus della discussione era incentrato su “come fare un buon database archeologico?”. Ma ora, a distanza di anni, al momento invece della sua pubblicazione, il focus è molto diverso: cosa di un'esperienza purtroppo fallimentare<sup>1</sup>, può essere utile al presente?

Riproponiamo dunque le riflessioni che ci avevano mosso nella costruzione di “Knossos”, così come le avevamo pensate allora: non più come propedeutica di presentazione a un “prodotto”, ma come parole in libertà sui processi che portano a comprendere come funzionano questi sistemi “delicati”. Un esercizio non di stile, ma utile come riflessione scatenante; spesso nel mondo degli archeologi anche spunti minimi come questo contributo (che parte, si può dire, dalle “basi” del problema) possono risultare molto fruttuosi.

## 2. I problemi

La presenza e l'importanza di un database informatizzato nella pratica archeologica, specie quella di scavo, sono principi per i quali non occorre alcuna discussione: fondamentale invece discutere la metodologia di approccio.

Due punti devono essere tenuti presenti: la struttura del database, che deve rispettare i canoni e le esigenze della ricerca e salvare contemporaneamente la funzionalità, e la portabilità dei dati. Per “portabilità” si intende la possibilità di far circolare, confrontare, recuperare ed analizzare in ogni ambiente (software) e momento i propri dati.

Già da qualche tempo è divenuto sempre più evidente come in Archeologia, come pure nella prassi di ricerca scientifica in genere, la scelta dell'impianto software utile non possa che ricadere su strumenti Open Source: questi infatti garantiscono (con un risvolto economico non indifferente) il soddisfacimento dei due punti precedenti, offrendo ben pochi punti problematici, e anzi, “per loro stessa natura” sono portati all'apertura e allo scambio (di formato e di dati).

Tuttavia, sia dal punto di vista generale, cioè della difficoltà di gestire un database come tale, sia dal punto di vista più particolare, delle problematiche maggiormente collegate alla scelta del software da implementare, cioè software libero, esistono diversi problemi.

Parlando di database in senso stretto, esiste l'esigenza di trovare soluzioni standard, sia dal punto di vista dei formati che della struttura dei dati. Se per *i formati* la soluzione dovrebbe essere intrinseca nell'utilizzo di prodotti open source, risultando comunque necessario fornire un'apertura anche verso formati chiusi, per quanto riguarda *la struttura* il primo problema nasce proprio in seno all'utilizzo finale del database per fini archeologici; infatti, nella realtà pratica ogni scavo ha il suo database, adattato alle specifiche esigenze di quel contesto particolare e alle specifiche “forme mentali” dell'operatore che lo implementa. A questo si aggiungono i problemi relativi alle competenze dell'utente finale, ovvero chi deve adoperare il database, sia nel momento dell'immissione dei dati, sia nelle fasi successive, come l'elaborazione, recupero ecc. degli stessi.

Spesso i prodotti open source vengono tacciati di un giudizio che va dal cauto “sono complessi” da usare, allo sfiduciato “sono solo per esperti”. Senza entrare nel merito di questa questione, si dovrà cercare una soluzione finale che si riveli essere semplice e immediata, a qualsiasi livello di utilizzo (e di utilizzatore), mantenendo salva la potenza di analisi.

Quello che viene richiesto, insomma, è un database potente, semplice, adattabile; Knossos vuole essere allora: potente, utilizzando come database MySQL; semplice, grazie a un'interfaccia semplice e intuitiva; adattabile, perchè esportabile su qualsiasi database Sql compatibile.

### 3. In pratica

Occorre qui fare qualche premessa: caratteristica principale di un database come MySQL è la distinzione tra la parte server e quella client del programma; un server è una componente informatica che fornisce servizi ad altre componenti (tipicamente chiamate appunto client) attraverso, solitamente, una rete. Questa distinzione, non presente nei software commerciali, offre diversi vantaggi, il maggiore dei quali si evidenzia proprio in fase di definizione della struttura del database stesso. Infatti nel processo di strutturazione della banca dati utilizzando un programma commerciale è gioco-forza impostare il lavoro partendo dalla maschera (il client, in questo caso), o perlomeno considerare come secondaria la struttura delle tabelle. Questo comporta che le relazioni tra gli elementi del database potranno essere non ottimizzate, rindondanti o addirittura errate, tanto da impedire la funzione principale della banca dati stessa: il recupero e l'analizzabilità dei dati. Invece l'approccio che parte dal server e si occupa poi di costruire su di esso il client, consente di definire da subito relazioni coerenti e l'ottimizzazione del database.

Un altro vantaggio di questo sistema è che diversi client possono connettersi a un singolo server, il che si traduce in diversi operatori che contemporaneamente possono lavorare su un unico database, accedendo da diversi terminali. Questi terminali non necessariamente devono essere vicini al server, ovviamente, ma possono essere anche connessi via internet, senza necessità di installare nessun altro software sui computer client, di importare su diverse macchine il database e di non poter sincronizzare in tempo reale e in sicurezza la banca dati.

Infine, la distinzione client-server consente un'operazione impossibile con i vari software commerciali: poter scegliere se utilizzare un certo client (maschera) piuttosto che un altro, oppure connettere il medesimo client a un diverso server, ovvero poter disporre di quella facoltà di scelta che sta alla base del concetto di open source. Infatti, se Knossos fosse un software chiuso, l'utente dovrebbe accontentarsi dell'interfaccia che è stata già scelta per lui, o di una struttura del database che non corrisponde alle sue esigenze. Invece in questa maniera l'utente potrà tenere quello che di Knossos ritiene valido, che sia l'interfaccia o il database, e sostituire l'uno o l'altro con altre soluzioni, o anche con una propria.

Parte centrale di Knossos è quindi il database: ma in che maniera è stato realizzato? Nel campo archeologico, come già detto più sopra, un rischio sempre presente è la "personalizzazione" della struttura del database: processo che porta alla non standardizzazione.

Diversi approcci possono essere tentati per risolvere questo problema: un primo approccio poteva essere dunque quello massimalista, cioè di inserire tutti i campi possibili. Ma naturalmente, vista l'estrema variabilità della casistica possibile, questo sarebbe risultato ingestibile. Oppure si sarebbe potuto scegliere l'altro estremo, ovvero non programmare nessuna struttura, e proporre un "semplice" gestore di database. Questo avrebbe portato però all'errore iniziale.

La sola soluzione praticabile è parsa dunque quella di procedere alla realizzazione di un database preformattato. Nella fase iniziale, orientativa, di strutturazione del database, ci si è rivolti a cercare innanzitutto se esistessero altre soluzioni precedenti e in verità queste non mancano, soprattutto nei paesi anglosassoni; ma anche in Italia sono stati prodotti software o database di scavo, che condividono però la stessa sorte dei prodotti stranieri: la maggior parte di queste esperienze non hanno avuto seguito o sono rimaste legate a programmi ormai obsoleti, oppure sono state realizzate su piattaforme non libere. Causa principale dell'abbandono di questi progetti è apparsa essere ancora la mancanza di standardizzazione. Un progetto iniziato da pochi non trova consensi presso gli altri utenti, i quali preferiscono sempre ricorrere a soluzioni personali.

In Italia esistono diverse schede utilizzate per registrare i dati di scavo, la maggior parte delle quali sono versioni "alleggerite" delle schede ministeriali. L'ICCD (l'Istituto Centrale per la Catalogazione e Documentazione) ha prodotto diversi tipi di schede, per il Saggio Archeologico, Sito Archeologico, Reperti, ecc. Queste schede tuttavia propongono una struttura dei dati che è quella del supporto cartaceo, ovvero con ripetizione degli stessi elementi di scheda in scheda, ridondanze, specificazioni particolareggiate in certi punti e assolutamente insufficienti in altre, campi non generalizzabili né astraibili. Soprattutto la struttura di queste schede, al di là dei campi, che siano utili o meno, o integrabili, non può assolutamente essere riportata così com'è in quella di un database informatico, pena la distruzione della coerenza interna del database stesso.

Non da ultimo, è vero che lo "standard" indicato dal ministero vuole appunto presentarsi come *standard*, a cui ci si dovrebbe uniformare per avere dati uniformi su tutto il terreno nazionale. Tuttavia per come è strutturato lo schema dei dati ministeriale, ci si trova davanti a quanto di meno standard sia possibile incontrare: infatti questa struttura dati ottiene il risultato inverso a quello che si propone di fare: rende inconfrontabili i dati. Questo fatto discende proprio dalla struttura delle schede, che danno adito a diverse interpretazioni (ovvero, aumentano la possibilità di errori e di non-uniformità), sono troppo varie e soggettive e hanno una struttura non informatica.

In più, nell'ottica di rendere Knossos il più possibile "compatibile", e quindi "esportabile" anche all'estero, non si può far certo riferimento esclusivo alla normativa italiana<sup>2</sup>: l'obiettivo sarebbe quello di enucleare entità logiche minime confrontabili il più possibile generali.

Partendo quindi comunque dal set di informazioni richieste a livello ministeriale da riversare nel database e tenendo anche conto delle esigenze di raccolta dei dati del lavoro da campo, il database di Knossos è stato concepito avendo

sempre bene in mente come requisito fondamentale di *mantenere minimo il numero di tabelle*. Questo risulta spesso difficile quando soprattutto si ha a che fare con una classificazione e un'astrazione difficile come queste che si sono tentate nel razionalizzare la complessità del record archeologico. Si tratta infatti di individuare entità logiche minime condivisibili, che siano accettabili sia dal punto di vista della coerenza del database che dell'utente finale (l'archeologo). Il tentativo operato con Knossos non si pone naturalmente come un punto di arrivo, ma bensì di partenza, laddove il fruttuoso apporto degli utenti e degli utilizzatori porti il suo prezioso contributo e la sua esperienza.

Il problema principale nell'astratizzare il record archeologico è stato individuato nella gestione e nella comprensione delle relazioni presenti, o postulabili, tra le varie entità. Un gran numero di relazioni, infatti, è del tipo "molti a molti" (n:n); ciò rende impossibile avere campi univoci. Molte relazioni conducono a produrre molte tabelle, rendendo pesante il database e le operazioni a suo carico. Una possibile soluzione che abbiamo progettato è stata quella di inserire tutti i dati di questo tipo in un singolo campo, separando i vari valori tramite una stringa arbitraria (come ";"). Il client, e non il database, è a questo punto costretto a portare tutto il carico dell'elaborazione dei dati e delle query: non è possibile per il database gestire una query con questo sistema, e solo a livello software si possono distinguere tra i dati veri e propri e la stringa adoperata ("passando" solo i dati). Tuttavia creare una query solo via client (con l'implementazione di questo sistema con separatore arbitrario) può essere problematico; stiamo tuttora valutando altre alternative tramite le quali mantenere basso il numero di tabelle (e mantenere una struttura interna ottimizzata e coerente).

Per non rompere la compatibilità con i vari altri database e garantire la portabilità dei dati, si è deciso di implementare le varie funzioni non a livello del database, ma di client/programma. Anche la *consistenza* delle relazioni è mantenuta tramite la gestione delle relazioni non a livello server, ma a livello client: c'è però da dire che affidando tutto il carico del lavoro al client possono sorgere alcuni problemi. Infatti, passando attraverso il client per ogni operazione, si possono moltiplicare gli errori (ci sono da considerare anche gli errori a livello del codice client) e rallentare le operazioni di analisi (c'è un passaggio in più da compiere, infatti, visto che non è il database a compiere le query direttamente, ma il client che interroga il database). Si sta perciò procedendo parallelamente allo sviluppo del software con questo sistema anche a un processo di ottimizzazione del database, per provare a eliminare i problemi strutturali che hanno condotto a cercare la soluzione sopra esposta.

Infine le etichette dei campi sono state tutte tradotte in inglese: sia perchè in inglese non esistono accenti e certi caratteri speciali presenti in italiano, evitando così certi banali errori e incompatibilità con database diversi (nel caso si vogliano esportare i dati immessi con Knossos in un altro database, come PostgreSQL, SQLite o altro), in secondo luogo perchè l'internazionalizzazione del software è uno dei primi passi da compiere se si vuole arrivare a proporre uno standard applicativo.

#### 4. L'interfaccia

Diversamente da altri tipi di database, il progetto di Knossos prevede un'interfaccia disarticolata dall'impostazione tradizionale che procede per tabelle. Nel senso che, sebbene in molti prodotti commerciali e non la parte che comprende le tabelle sia mascherata all'utente, comunque a livello di interfaccia viene ripresa una struttura "rigida", tabellare e schematica. Ogni scheda con i dati è su di una pagina a sè stante. L'idea di Knossos è invece quella di presentare in una sola finestra tutte le tabelle e i dati.

Il risultato finale dovrebbe essere quello di avere delle schede a "linguetta" estraibili, cliccando sulle quali si apre il menù relativo alla scheda (ad esempio: la scheda SITO); in questo modo ogni punto del database è raggiungibile dalla pagina principale. In più ogni scheda presenterà una barra per la ricerca rapida all'interno di quella maschera/tabella, con un wizard per le query o con la possibilità di digitarne una direttamente a mano.

Ogni "linguetta" mostrerà in forma di elenco i dati (ad esempio: la scheda Us presenterà l'elenco delle Us) su ciascuna delle quali sarà possibile cliccare per aprire la scheda completa (nell'esempio, i dati per la singola Us). Trovare la scheda richiesta sarà possibile, come detto, grazie alla barra per la ricerca rapida.

Questo metodo ci pare particolarmente semplice e utile in quanto mantiene valida la funzione basilare del database, al di là delle possibilità di analisi, cioè quella di rendere recuperabili i dati. Una grande parte dei dati utili, specie nel doposcavo, sono quelli relativi alla documentazione: documentazione bibliografica, documentazione grafica (disegni di scavo, fotopiani, vettoriali dei rilievi, fotografie dei reperti, ecc.). Altra caratteristica importante la gestione delle fotografie. Questa sarà implementata tramite un vero e proprio "gestore di album di foto", con tutta la semplicità e fruibilità che questa soluzione comporta.

La possibilità di poter accedere facilmente e direttamente a questi dati e di averli, per così dire, sempre "sottomano", ci sembra una funzione fondamentale di un database.

Insieme con lo sviluppo di questa interfaccia, che, come risulta evidente, è di tipo client si stava pensando anche allo sviluppo di un'interfaccia server-side utile per accedere al database anche da PC su cui non è stato installato Knossos. Questo obiettivo potrebbe essere realizzato o tramite un server remoto con PHP o Python o tramite un applicativo server vero e proprio in C++ dotato di interfaccia web.

#### 5. Aggiornamento sullo stato del progetto al 2012

Al momento della presentazione del progetto, nel 2007, il database Knossos pareva destinato a un rapido sviluppo; esistevano le risorse, esisteva un gruppo di lavoro ben affiatato. Ma come nel testo sopra ci si lamentava dei troppi pro-

getti per un database archeologico (o dei “beni culturali”) andati a finire male, anche in questo caso ci si deve arrendere all’evidenza di una (forse) “bella” idea che non ha purtroppo avuto seguito.

Nel testo si additava, come causa del fallimento dei progetti, principalmente il problema del non standardizzare: si propongono certe strutture dei dati, ma queste non vengono accolte. In parte è stato così anche per Knossos. La sua struttura dati non ha incontrato il favore di chi avrebbe poi dovuto impiegarlo, tanto da portare al congelamento del progetto, prima, e del suo definitivo cancellamento, poi, quando sono intervenuti altri problemi (come lo scioglimento del gruppo di lavoro, la mancanza di finanziamenti).

Qual è stato il tallone d’Achille di Knossos? Ci sentiamo forse di ricercarlo nell’eccessiva ricerca di standardizzazione. Si voleva un progetto che fosse utilizzabile in ogni contesto, sia nella struttura dei dati, sia nel vocabolario con il quale i dati avrebbero dovuto essere trattati. Per molti un database dovrebbe essere invece “costruito ad hoc”, di esigenza in esigenza. Anche questo è vero, ma “costruito ad hoc” non vuol dire di rinunciare del tutto a minimi concetti atomici che sottendano all’intero impianto della ricerca (in questo caso archeologica).

Lo standard ICCD, pur non essendo, come detto sopra, affatto uno standard, è di fatto l’unico punto di riferimento in materia; tuttavia nemmeno esso viene preso in considerazione da tutti (anzi, da molto pochi). Se potessimo riprendere in mano il progetto, incominceremmo (e in parte, dobbiamo dire che già nel 2008, prima del congelamento, eravamo partiti da qui) proprio dall’ICCD. Tuttavia non credo che rinunceremmo nemmeno questa volta a proporre una struttura dati solida e ben ragionata, una struttura dati che cerchi di rendere conto degli atomi minimi di cui sono composti i processi della ricerca archeologica.

## Notes

\* Università degli Studi di Padova, Dottorato di Ricerca in Scienze Archeologiche.

† Università degli Studi di Padova, Dottorato di Ricerca in Scienze Astronomiche, collaboratore a contratto E.S.A.

‡ Database, Web Designer, sviluppatore software.

<sup>1</sup> Vedi l’ultima sezione.

<sup>2</sup> In questo senso, c’è da tenere conto per esempio che nelle schede ministeriali sono presenti numerosi campi che contengono dati relativi alla posizione amministrativa del sito sul territorio nazionale, ovvero comune, diocesi, ecc., divisioni che negli altri paesi sono diverse. In questo caso i campi sono stati pensati e organizzati per coprire il più possibile le diverse e più diffuse realtà amministrative esistenti.